

FedCS: Coreset Selection for Federated Learning

Chenhe Hao¹, Weiyang Xie^{1,*}, Daixun Li¹, Haonan Qin¹, Hangyu Ye¹, Leyuan Fang², Yunsong Li¹
¹ Xidian University, ² Hunan University

{23011210919, ldx, wagyr}@stu.xidian.edu.cn, {wyxie, qinhaonan}@xidian.edu.cn
 fangleyuan@gmail.com, yslie@mail.xidian.edu.cn

Abstract

Federated Learning (FL) is an emerging direction in distributed machine learning that enables jointly training a model without sharing the data. However, as the size of datasets grows exponentially, computational costs of FL increase. In this paper, we propose the first Coreset Selection criterion for Federated Learning (FedCS) by exploring the Distance Contrast (DC) in feature space. Our FedCS is inspired by the discovery that DC can indicate the intrinsic properties inherent to samples regardless of the networks. Based on the observation, we develop a method that is mathematically formulated to prune samples with high DC. The principle behind our pruning is that high DC samples either contain less information or represent rare extreme cases, thus removal of them can enhance the aggregation performance. Besides, we experimentally show that samples with low DC usually contain substantial information and reflect the common features of samples within their classes, such that they are suitable for constructing coreset. With only two time of linear-logarithmic complexity operation, FedCS leads to significant improvements over the methods using whole dataset in terms of computational costs, with similar accuracies. For example, on the CIFAR-10 dataset with Dirichlet coefficient $\alpha = 0.1$, FedCS achieves 58.88% accuracy using only 44% of the entire dataset, whereas other methods require twice the data volume as FedCS for same performance.

1. Introduction

Federated Learning (FL), as a distributed machine learning paradigm, has been widely explored in fields such as computer vision [5], medical analysis [18], and autonomous driving [10]. However, low training efficiency is an inherent challenge in FL due to the high consumption of computational resources. Specifically, as datasets on clients explosively grow, learning from these large-scale datasets be-

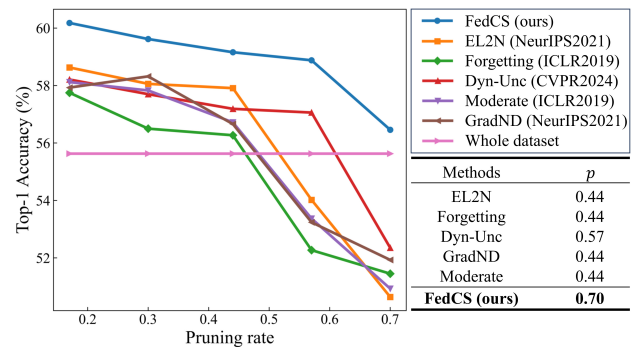


Figure 1. Performance comparison between FedCS and other coreset selection methods from low to high pruning rate on CIFAR-10 with ResNet-18 in FL with highly data distribution heterogeneous, where Dirichlet coefficient α is set to 0.1. The table on the right shows the maximum pruning rates (p) achieved by various methods without degrading model performance.

comes progressively time-consuming, leading to the training efficiency decrease of FL.

Currently, numerous works aim to reduce the computational costs by employing network pruning [14] in FL. Typical approaches generate client-specific subnetworks to decrease network size in local training [16, 29]. While network pruning effectively reduces computational costs, it also introduces structural differences among client networks, which posing challenges for network aggregation and coordination. In contrast, coreset selection methods can avoid these issues by removing data from the dataset without altering the network structure [27, 30]. Due to the consistency of data distribution, a significant portion of data is redundant [27, 30], meaning it can be excluded from training without sacrificing performance. However, the coreset selection work in FL lacks an effective lossless pruning criterion. As a result, applying existing gradient-matching-based coreset selection frameworks or coreset selection criterion to FL will lead to (1) insufficient privacy protection and (2) exacerbation of data distribution imbalance. On the one hand, existing coreset selection framework in FL re-

*Corresponding author

quires setting up a dataset on the server, which is difficult and privacy-intrusive. For example, GCFL [36] removes samples with noise on clients by matching the gradients generated from server dataset. On the other hand, most SOTA coreset selection methods select hard samples to construct coresets, such as Forgetting [39] and GradND [31]. Since these samples always include rare extreme cases, remaining them will only increase data distribution differences among clients, ultimately harming the performance of the global network (as shown in Figure 1).

In this paper, we propose a **Coreset Selection** method for **Federated Learning (FedCS)**, which constructs efficient and unbiased coresets for subsequent training, aiming to reduce computational costs while maintaining model performance. We employ the Distance Contrast (DC) between a sample’s distance to corresponding class center and to the nearest other class center for coreset selection because DC indicates the intrinsic properties inherent to samples. Our proposed solution selects decision boundary samples with low DC while removing redundant and rare extreme samples with high DC to avoid exacerbation of data distribution imbalance, thereby preventing the degradation of the global network’s aggregation performance. It makes FedCS a potential and high-efficiency method applicable to various FL scenarios. Specifically, we first extract the hidden representations of each sample after pre-training, then compute local class centers and aggregate them to obtain the global class centers. Class center aggregation avoids important samples from overlooking caused by discrepancy between client sample distributions and global sample distribution. Next, each client calculates DC for each sample as its score. Finally, we perform high-ratio pruning on high DC samples from classes with large capacities, followed by low-ratio pruning on high DC samples from remaining samples. Double pruning further mitigates the imbalance of exacerbating data distribution during pruning process. Convergence analysis and extensive experiments have demonstrated the better convergence and superior generalization capability of the proposed FedCS, enabling it to outperform other comparison methods across different pruning settings on various datasets and networks.

Our main contributions are summarized as:

- We propose FedCS, a novel coreset selection method to reduce the computational costs for FL. To our best knowledge, our criterion is the first work that can keep performing well under high pruning rate in FL.
- We pioneeringly construct coreset with DC in FL which makes a trade-off between reducing data distribution imbalance and retaining sample information in the feature space, thereby greatly enhancing training efficiency while maintaining the model’s aggregation capability.
- Extensive experiments and mathematical proof have demonstrated our higher efficiency and better perfor-

mance over SOTA methods. Besides, our designed ablation experiments also show effectiveness in class center aggregation and double pruning strategy.

2. Related Works

2.1. Federated Learning

Low training efficiency in FL is an inherent issue. Previous researches have addressed it from 3 perspectives: reducing communication costs, decreasing computational costs and using asynchronous algorithms. Mainstream methods for reducing communication costs include gradient compression, network quantization and network pruning [1, 4, 22, 23, 40, 44]. To reduce computational costs, some researches typically use hard pruning on network to shrink the local network size [16, 24, 29]. Konečný *et al.* [17] perform low-rank decomposition on the network to reduce computational costs by updating only part of the network. DWNP [11] proposes a method for device-level subnetwork pruning in FL using supernet networks and embedding layers. Although these methods have made progress in reducing FL’s computational overhead, they increase the risk of privacy leakage. Moreover, these methods require complex aggregation strategies to coordinate and aggregate the pruned local networks, which will introduce additional and unnecessary computational costs. FL asynchronous algorithms enable clients to participate in training without synchronisation, thus reducing communication delays and waiting time [25, 35, 42, 45, 48]. While these methods have improved the training efficiency of FL from different aspects, efficient methods for reducing computational costs in FL are still lacking. We believe that coreset selection method will be an important tool to reduce computational costs in FL.

2.2. Coreset Selection

Coreset Selection, also known as Dataset Pruning, aims to reduce dataset size by selecting significant samples based on predefined criteria [2, 8, 12, 20, 26, 30, 46]. Entropy [7] investigates certainty and decision-making boundaries via predicted outputs. MoSo [38] selects coresets based on loss values. ACS [15] uses adaptive coreset selection guided by error and divergence scores to identify informative samples. TDDS [47] uses KL divergence to assess sample importance. InfoBatch [33] accelerates training by pruning less informative samples based on loss distribution. However, these methods tend to construct coreset with hard samples, which will exaggerate the imbalance of data distribution. Though there are some researches focusing on other samples, they still have some limitations. For example, Moderate [43] calculates distances in feature space and selects samples near median, but median is not always accurate for capturing sample’s importance. AUM [32] identifies mislabeled samples by exploiting training dynamics differences,

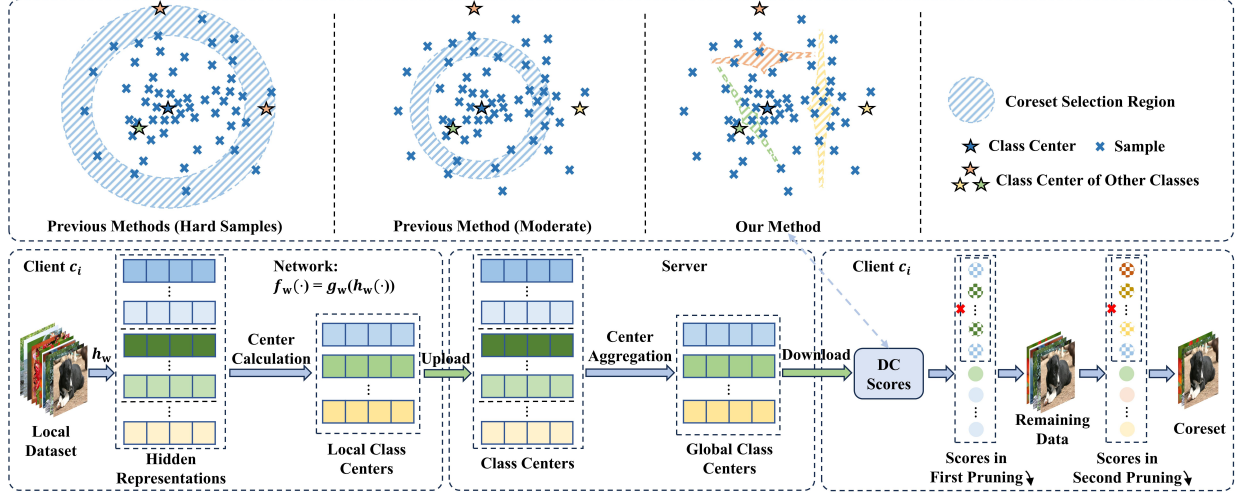


Figure 2. Overview of FedCS. The top section shows representation structures achieved by different coreset selection ways, and the bottom section illustrates the FedCS framework, where different colors represent different classes. Each client calculates local class centers and sends them to the server. Server aggregates the class centers by using median and sends the global class centers back to the clients. Clients calculate the DC score for each sample, then apply double pruning.

but directly using logits overlooks unique information beyond the predicted class, which will further decrease generalization ability of networks. Currently, there is a lack of effective criterion for coreset selection in FL. In this paper, we will design an efficient coreset selection criterion for FL based on the characteristics of data distribution.

3. Methodology

3.1. Preliminaries

In the sequel, vectors, matrices, and tuples are denoted by bold-faced letters. We use $\|\cdot\|_p$ as the ℓ_p norm of vectors or matrices. Let $[n] = 1, \dots, n$, $\mathbb{I}(\mathcal{B})$ be the indicator of the event $\mathcal{B}$. In our FL set up, a group of N clients are represented as set $\mathcal{C} = \{c_1, c_2, \dots, c_N\}$. The training dataset $D = \bigcup_{i=1}^N D_i$ is divided among the clients in a disjoint and non-IID way, and each client c_i has a data chunk D_i consisted of $n_i = \sum_{k \in [\mathcal{K}]} n_i^k$ samples $\{(\mathbf{x}_{ij}, y_{ij})\}_{j=1}^{n_i}$, here $\mathbf{x}_{ij} \in \mathcal{X}$ denotes the j^{th} data point on c_i client, $y_{ij} \in [\mathcal{K}]$ represents its corresponding target, and n_i^k represents the number of samples belonging to target k on client c_i .

We denote a global neural network parameterized with weight vector $\mathbf{w}$ as $f_{\mathbf{w}}$. The optimization objective of $f_{\mathbf{w}}$ on D is to minimize empirical risk:

$$F_i(\mathbf{w}_i, D_i) = \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(f_{\mathbf{w}_i}(\mathbf{x}_{ij}), y_{ij}), \quad (1)$$

$$F(\mathbf{w}, D) = \sum_{i=1}^N \frac{n_i}{n} F_i(\mathbf{w}_i, D_i) \quad (2)$$

where $f_{\mathbf{w}}(\mathbf{x}_{ij}) \in R$ is the predicted probabilities of $\mathbf{x}_{ij}$. Typically, global network is updated as follow

$$\mathbf{w}_i^{(t+1)} = \mathbf{w}_i^{(t)} - \eta_t \nabla F_i(\mathbf{w}_i^{(t)}, D_i), \quad (3)$$

$$\mathbf{w}^{(t+1)} = \frac{1}{m} \sum_{i \in \mathcal{S}} \mathbf{w}_i^{(t+1)}, \quad (4)$$

where η_t is the learning rate, $\mathbf{w}_i^{(t)}$ is the local weight vector on client c_i at epoch t , $\mathbf{w}^{(t)}$ is the global weight vector on server at epoch t , m is the number of clients participating in aggregation, $\mathcal{S}$ is the set of clients participating in aggregation, and $\nabla F_i(\mathbf{w}_i^{(t)}, D_i)$ is defined as the gradient calculated over the dataset D_i on client c_i at epoch t .

Our aim is to enhance training efficiency by constructing a coreset $D^* = \bigcup_{i=1}^N D_i^* = \bigcup_{i=1}^N \{(\mathbf{x}_{ij}^*, y_{ij}^*)\}_{j=1}^{n_i^{\text{re}}}, D_i^* \subset D_i$, where n_i^{re} is the number of samples included in client c_i after pruning. $(\mathbf{x}_{ij}^*, y_{ij}^*)$ is a selection of data from the original dataset which can represent the complete dataset. The pruning rate on each client is $p_i = 1 - (n_i^{\text{re}}/n_i)$.

3.2. Produce Description

Representation Extraction. Given a network after several rounds of pre-training denoted by $f_{\mathbf{w}}(\cdot) = g_{\mathbf{w}}(h_{\mathbf{w}}(\cdot))$, where $h_{\mathbf{w}}(\cdot)$ denotes the part of the network mapping input data to hidden representations at the penultimate layer and $g_{\mathbf{w}}(\cdot)$ is the part mapping such hidden representations to the output $f_{\mathbf{w}}(\cdot)$ for classification. Namely, for a data point $(\mathbf{x}_{ij}, y_{ij})$, its hidden representation is $h_{\mathbf{w}}(\mathbf{x}_{ij})$. Therefore, with the pre-trained deep network $f_{\mathbf{w}}(\cdot)$ and full training data on each client $D_i = \{(\mathbf{x}_{ij}, y_{ij})\}_{j=1}^{n_i}$, the hidden representations of all data points on each client are acquired

as $\{\mathbf{z}_{ij} = h_{\mathbf{w}}(\mathbf{x}_{ij})\}_{j=1}^{n_i}$. At the representational level, the local class center of each class is:

$$\left\{ \mathbf{z}_i^k = \frac{\sum_{j=1}^{n_i} \mathbb{I}(y_{ij} = k) \mathbf{z}_{ij}}{\sum_{j=1}^{n_i} \mathbb{I}(y_{ij} = k)} \right\}_{k \in [\mathcal{K}]}, \quad (5)$$

where the mean of the representations from one class on each client is calculated as the mean of every single dimension in the representations. Although calculating class centers by median is more effective on eliminating bias from extremely rare samples, the enormous dataset on the client would result in a higher computational costs during median calculation. Therefore, we use the geometric center as the local class center to reduce computational costs.

Classes Center Aggregation. Each client sends local class centers to server after calculating with their full datasets D_i . In this process, each client only transmits the geometrically averaged results of the hidden representations generated by the penultimate layer. Since the hidden representations do not carry original information related to samples, and averaging operations also obscure the unique information of individual samples, data privacy can be effectively safeguarded. The reason for not using the logits which is more correlated with classification information, is that the hidden representations generated by the penultimate layer contain a little valuable intermediate feature information that the logits lack. This additional information enhances the generalization ability of the coreset. We use the median to aggregate these local class centers in order that it eliminates the bias of global centers brought by rare extreme data in clients. In this way, we get the global class center

$$\{\mathbf{z}^k = \text{median}(\mathbf{z}_i^k)_{i=1}^N\}_{k \in [\mathcal{K}]}, \quad (6)$$

where the representations median from one class on each client is calculated as the median of every single dimension in the representations. Then, server sends these global class centers to each client. Since the amount of clients is relatively small compared to the large volume of samples on each client, median aggregation is used to obtain global class centers to prevent coreset selection bias caused by differences in dataset distribution among the clients.

DC Score for Data Selection. With hidden representations $\mathbf{z}_{ij}$ and representational global class centers $\{\mathbf{z}^k\}_{k \in [\mathcal{K}]}$, the Euclidean distance from each representation to the each global class center can be calculated as

$$d_{ij}^k = \|\mathbf{z}_{ij} - \mathbf{z}^k\|_2. \quad (7)$$

Samples that are farther away from the corresponding class center usually have lower posterior probabilities on their own class, we usually consider them outliers or harder to train. Samples that are closer to the corresponding class center usually have a higher posterior probability on corresponding class, we usually consider them redundant. On

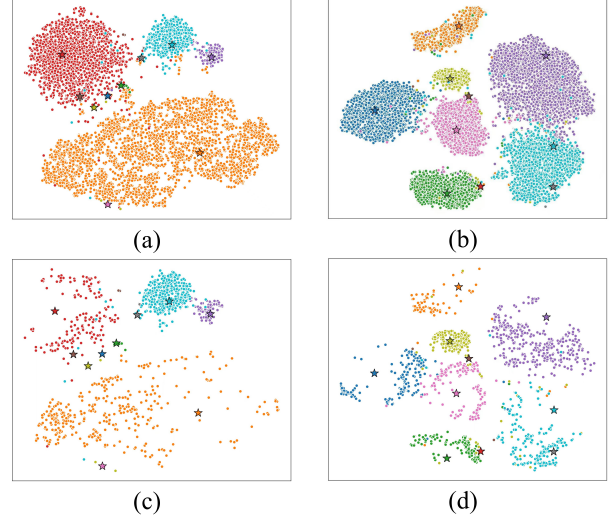


Figure 3. Comparison of t-SNE effects between before and after pruning on MNIST. (a) and (b) show sample distributions before pruning, while (c) and (d) show distributions after pruning.

the one hand, some works have suggested that the network’s performance during the early training stages may be weak, therefore when the coreset is predominantly composed of hard samples, it may result in noisy gradients [27, 34]. On the other hand, hard samples on each client may represent extreme or rare cases within that client’s dataset. When each client retains only hard samples, the distribution differences between clients will increase, leading to greater overall data distribution heterogeneity. These existing issues in training process will ultimately make the network converge to sub-optimal optimum or even fail to converge. Thus it is not appropriate to select hard samples with massive information only to construct the coreset in FL. Additionally, when a sample is located in a dense region of class center, using distance alone is insufficient to capture the sample’s intrinsic characteristics. Therefore, we choose Distance Contrast (DC), which better represents the intrinsic features of the samples than distance, to measure sample importance, and specifically select samples around decision boundary.

On each client, we calculate the distance $\{d_{ij}^k\}_{k \in [\mathcal{K}]}$ between each sample $\mathbf{z}_{ij}$ and the global class center $\{\mathbf{z}^k\}_{k \in [\mathcal{K}]}$ of each class, then choose two distances, the one which is relevant to the corresponding category $d_{ij}^{y_{ij}}$, and the other which is the minimum one $d_{ij}^{\min}$ except $d_{ij}^{y_{ij}}$

$$d_{ij}^{\min} = \min_{k \neq y_{ij}} \{d_{ij}^k\}_{k \in [\mathcal{K}]} . \quad (8)$$

We derive a score for samples by calculating the absolute value of DC between these two distances

$$s_{ij} = |d_{ij}^{\min} - d_{ij}^{y_{ij}}|. \quad (9)$$

Assumption 4.2 (μ -strongly Convex). $F_1, \dots, F_N$ are all μ -strongly convex, i.e., for all $\mathbf{v}$ and $\mathbf{w}$, we have $F_i(\mathbf{v}) \geq F_i(\mathbf{w}) + (\mathbf{v} - \mathbf{w})^T \nabla F_i(\mathbf{w}) + \frac{\mu}{2} \|\mathbf{v} - \mathbf{w}\|_2^2$.

Assumption 4.3 (Bounded Variance of Gradients). For the mini-batch ξ_i uniformly sampled at random from $\mathcal{D}_i$, the resulting gradient is unbiased, that is, $\mathbb{E}[\mathbf{g}_i(\mathbf{w}_i, \xi_i)] = \nabla F_i(\mathbf{w}_i, D_i)$. Also, the variance of stochastic gradients are bounded: $\mathbb{E}\|\mathbf{g}_i(\mathbf{w}_i, \xi_i) - \nabla F_i(\mathbf{w}_i, D_i)\|^2 \leq \sigma^2$ for $i = 1, \dots, N$. $\mathbf{g}_i(\mathbf{w}_i, \xi_i) = \nabla F_i(\mathbf{w}_i, \xi_i)$.

Assumption 4.4 (Bounded Stochastic Gradient). The expected squared norm of stochastic gradients is uniformly bounded, i.e., $\mathbb{E}\|\mathbf{g}_i(\mathbf{w}_i, \xi_i)\|^2 \leq G^2$ for $i = 1, \dots, N$.

These assumptions are common in related literature [3, 6, 21, 37], then we introduce two core metrics.

Definition 4.1 (Degree of Non-iid). After pruning, for the global optimum of $\mathbf{w}^* = \arg\min_{\mathbf{w}} F(\mathbf{w})$ and local optimum $\mathbf{w}_i^* = \arg\min_{\mathbf{w}} F_i(\mathbf{w})$, we define the degree of non-iid as:

$$\Gamma \triangleq F^* - \sum_{i=1}^N \mathcal{P}_i F_i^* = \sum_{i=1}^N \mathcal{P}_i (F_i(\mathbf{w}^*, D_i^*) - F_i(\mathbf{w}_i^*, D_i^*)). \quad (13)$$

This definition was first introduced by Li *et al.* [21]. Γ reveals the degree of heterogeneity in the data distribution after pruning. Usually, a larger Γ implies higher heterogeneity. If $\Gamma = 0$, it indicates that the data distribution between clients is balanced.

Definition 4.2 (Client Selection Bias). After pruning,

$$\rho(\mathcal{S}, \mathbf{w}') = \frac{\mathbb{E}_{\mathcal{S}}[\frac{1}{m} \sum_{i \in \mathcal{S}} (F_i(\mathbf{w}', D_i^*) - F_i^*)]}{\sum_{i=1}^N \mathcal{P}_i (F_i(\mathbf{w}', D_i^*) - F_i(\mathbf{w}_i^*, D_i^*))}, \quad (14)$$

$\mathcal{S}$ represents a set of clients who participate in aggregation at each epoch. Then, we define two related metrics that are independent of $\mathbf{w}'$:

$$\bar{\rho} \triangleq \min_{\mathbf{w}'} \rho(\mathcal{S}, \mathbf{w}'), \quad (15)$$

$$\tilde{\rho} \triangleq \rho(\mathcal{S}, \mathbf{w}^*). \quad (16)$$

The definition was introduced by Cho *et al.* [6]. All $\rho(\mathcal{S}, \mathbf{w}')$ are only related to the client selection strategy, independent of dataset pruning, and $\bar{\rho} \leq \tilde{\rho}$.

4.2. Main Convergence Result

We present the convergence for any coreset selection strategy in Fedavg with partial device participation in terms of Γ and client selection bias $\bar{\rho}, \tilde{\rho}$.

Theorem 4.1 (Convergence with Decaying Learning Rate). Under Assumptions 4.1 to 4.4, for learning rate $\eta = \frac{1}{\mu(t+\gamma)}$ and $\gamma = \frac{4L}{\mu}$, with any coreset selection strategy, after T iterations of Fedavg with partial device participation, we have the convergence as:

$$\mathbb{E}[F(\mathbf{w}^{(T)}, D^*)] - F^* \leq \frac{1}{(T + \gamma)} \left[\frac{4L(32\tau^2 G^2 + \sigma^2/m)}{3\mu^2 \bar{\rho}} + \frac{8L^2 \Gamma}{\mu^2} + \frac{L\gamma \|\bar{\mathbf{w}}^{(0)} - \mathbf{w}^*\|^2}{2} \right] + \frac{8L\Gamma}{3\mu} \left(\frac{\tilde{\rho}}{\bar{\rho}} - 1 \right). \quad (17)$$

The proof for Theorem 4.1 is presented in Supplementary. A key insight from Theorem 4.1 is that a smaller Γ results in faster convergence at the rate $\mathcal{O}(\frac{\Gamma}{T})$. It theoretically proves that alleviating data distribution heterogeneity through Double Pruning and DC Score can improve the convergence speed of Fedavg with partial device participation. The discussion for the influence of dataset pruning methods on data distribution is presented in Supplementary. The other key insight from Theorem 4.1 is that a smaller Γ results in a lower final error. Therefore, compared to other methods which select hard samples, FedCS accelerates convergence and reduces final solution with lower error by mitigating data distribution heterogeneity.

5. Experiments and Analysis

5.1. Experimental Settings

Networks and Datasets. Following previous studies, we compare the performance of different state-of-the-art coreset selection algorithms on CIFAR-10 [19], CIFAR-100 [19], CINIC-10 [9] and TinyImageNet. For CIFAR-10 and CINIC-10, we apply joint learning on 10 clients, the updates will be conducted across whole 10 clients. For CIFAR-100 and TinyImageNet, we apply joint learning on 100 clients, the updates will be a random selection of 10 clients out of a total of 100. We use ResNet-18 for CIFAR-10, ViT-base-patch16-224 (ViT-B-16) for CIFAR-100, VGG-11 for CINIC-10, and ResNet-50 for TinyImageNet.

Evaluation Measures. We keep the total number of epochs same in different methods and use Top-1 accuracy to compare the performance of various pruning methods and FedAvg [28] on the same network and dataset. The savings in computational and storage costs can be evaluated based on the actual pruning rate. For instance, if we prune 70% of the whole dataset, we save approximately 70% of the computational resources. The computational costs produced during the pruning process are negligible comparing to the savings achieved through pruning.

Baselines and Implementation. We conduct subsequent experiments by using various coreset selection methods on

Table 1. The Top-1 accuracy (%) of coreset selection methods with ResNet-18 on CIFAR-10.

Methods	$\alpha = 0.1, p_i^l = 0.1$					$\alpha = 1.0, p_i^l = 0.1$				
	$p_i^f = 0.1$	$p_i^f = 0.3$	$p_i^f = 0.5$	$p_i^f = 0.7$	$p_i^f = 0.9$	$p_i^f = 0.1$	$p_i^f = 0.3$	$p_i^f = 0.5$	$p_i^f = 0.7$	$p_i^f = 0.9$
EL2N	58.63 $\pm$ 0.79	58.06 $\pm$ 0.93	57.91 $\pm$ 1.27	54.02 $\pm$ 1.33	50.64 $\pm$ 1.43	66.55 $\pm$ 0.61	64.28 $\pm$ 0.71	60.57 $\pm$ 0.73	55.72 $\pm$ 0.80	53.07 $\pm$ 0.94
Moderate	58.12 $\pm$ 0.63	57.83 $\pm$ 0.68	56.72 $\pm$ 0.72	53.37 $\pm$ 0.75	50.93 $\pm$ 0.82	66.18 $\pm$ 0.37	64.78 $\pm$ 0.48	63.40 $\pm$ 0.54	60.35 $\pm$ 0.60	56.65 $\pm$ 0.67
GM	58.48 $\pm$ 0.51	57.97 $\pm$ 0.58	56.82 $\pm$ 0.61	57.53 $\pm$ 0.66	55.52 $\pm$ 0.71	66.40 $\pm$ 0.25	65.17 $\pm$ 0.31	63.59 $\pm$ 0.34	60.79 $\pm$ 0.44	57.06 $\pm$ 0.54
GradND	57.93 $\pm$ 0.70	58.32 $\pm$ 1.03	56.66 $\pm$ 1.14	53.24 $\pm$ 1.22	51.93 $\pm$ 1.27	66.29 $\pm$ 0.54	64.21 $\pm$ 0.55	60.83 $\pm$ 0.77	56.20 $\pm$ 0.87	52.06 $\pm$ 0.91
Forgetting	57.75 $\pm$ 0.74	56.50 $\pm$ 0.97	56.27 $\pm$ 1.25	52.27 $\pm$ 1.33	51.45 $\pm$ 1.46	66.18 $\pm$ 0.64	62.56 $\pm$ 0.88	62.28 $\pm$ 0.98	59.15 $\pm$ 1.03	54.76 $\pm$ 1.11
Random	58.24 $\pm$ 0.86	57.80 $\pm$ 1.07	55.46 $\pm$ 1.44	56.01 $\pm$ 1.64	54.38 $\pm$ 2.04	66.52 $\pm$ 0.67	65.45 $\pm$ 0.78	62.17 $\pm$ 0.79	61.02 $\pm$ 1.15	58.18 $\pm$ 1.48
FedCS(ours)	60.18$\pm$0.73	59.62$\pm$0.79	59.16$\pm$0.81	58.88$\pm$0.86	56.46$\pm$0.88	66.98$\pm$0.44	66.11$\pm$0.49	63.82$\pm$0.60	61.64$\pm$0.71	58.42$\pm$0.76
Whole Dataset	55.63 $\pm$ 0.68	55.63 $\pm$ 0.68	55.63 $\pm$ 0.68	55.63 $\pm$ 0.68	55.63 $\pm$ 0.68	66.04 $\pm$ 0.24	66.04 $\pm$ 0.24	66.04 $\pm$ 0.24	66.04 $\pm$ 0.24	66.04 $\pm$ 0.24

Table 2. The Top-1 accuracy (%) of coreset selection methods with Vit-B-16 on CIFAR-100.

Methods	$\alpha = 0.1, p_i^l = 0.1$			
	$p_i^f = 0.3$	$p_i^f = 0.5$	$p_i^f = 0.7$	$p_i^f = 0.9$
EL2N	76.46 $\pm$ 0.62	76.04 $\pm$ 0.64	75.83 $\pm$ 0.87	74.87 $\pm$ 0.92
Moderate	75.95 $\pm$ 0.53	75.22 $\pm$ 0.59	75.54 $\pm$ 0.74	74.38 $\pm$ 0.79
GM	76.59 $\pm$ 0.45	75.37 $\pm$ 0.52	75.79 $\pm$ 0.56	74.54 $\pm$ 0.68
GradND	76.04 $\pm$ 0.68	75.84 $\pm$ 0.83	74.62 $\pm$ 0.94	74.33 $\pm$ 1.07
Forgetting	75.89 $\pm$ 0.98	75.59 $\pm$ 1.06	74.16 $\pm$ 1.13	73.21 $\pm$ 1.18
Random	76.31 $\pm$ 0.81	75.39 $\pm$ 0.92	75.40 $\pm$ 1.16	74.72 $\pm$ 1.39
FedCS(ours)	77.06$\pm$0.51	76.72$\pm$0.66	76.58$\pm$0.71	75.25$\pm$0.74
Whole Dataset	77.08 $\pm$ 0.43	77.08 $\pm$ 0.43	77.08 $\pm$ 0.43	77.08 $\pm$ 0.43

Table 3. The Top-1 accuracy (%) of coreset selection methods with ResNet-50 on TinyImageNet.

Methods	$\alpha = 0.1, p_i^l = 0.1$			
	$p_i^f = 0.3$	$p_i^f = 0.5$	$p_i^f = 0.7$	$p_i^f = 0.9$
EL2N	56.99 $\pm$ 0.97	56.98 $\pm$ 1.15	54.18 $\pm$ 1.34	55.13 $\pm$ 1.45
Moderate	57.53 $\pm$ 0.88	57.02 $\pm$ 0.92	54.83 $\pm$ 1.01	55.23 $\pm$ 1.08
GM	57.85 $\pm$ 0.73	57.36 $\pm$ 0.89	55.81 $\pm$ 0.91	55.66 $\pm$ 0.94
GradND	57.16 $\pm$ 0.93	56.53 $\pm$ 1.12	53.87 $\pm$ 1.24	55.01 $\pm$ 1.43
Forgetting	57.09 $\pm$ 1.16	56.87 $\pm$ 1.41	54.69 $\pm$ 1.56	55.21 $\pm$ 1.73
Random	57.26 $\pm$ 1.34	57.13 $\pm$ 1.59	54.12 $\pm$ 1.77	55.44 $\pm$ 2.12
FedCS(ours)	58.67$\pm$0.81	58.26$\pm$0.83	58.15$\pm$0.96	56.83$\pm$1.18
Whole Dataset	57.61 $\pm$ 0.74	57.61 $\pm$ 0.74	57.61 $\pm$ 0.74	57.61 $\pm$ 0.74

FedAvg [28]. We compare ours against several state-of-the-art coreset selection methods: EL2N [31], Forgetting [39], Moderate [43], GradND [31], random. For the Moderate, we add class center aggregation from our approach to de-

sign a new method Global Moderate (GM) in the comparative experiments. For all comparison methods, we use the double pruning strategy to keep all methods in same pruning rate. Following recent studies [41], we fix synchronization interval $I = 5$. For non-IID dataset partitioning over clients, we use Dirichlet- α (abbreviated as Dir(α)) sampling [13], where the coefficient α measures the heterogeneity. The smaller the Dirichlet- α , the more imbalanced the data distribution. In the experiments on CIFAR-10, we select the Dirichlet coefficient α from $\{0.1, 1.0\}$ and accordingly select β from $\{0.5, 0.65\}$. In the experiments on other dataset, we set the Dirichlet coefficient α to 0.1, β to 0.65. For CIFAR-10, we choose 5 different values of p_i^f and p_i^l for each method and each α , and for other dataset, we choose 4 different values of p_i^f and p_i^l for each method. The SGD optimizer uses a cosine decay learning rate with an initial value $l_r = 0.01$. We set the communication round $T = 200$ and pre-training communication round $T_P=4$ for CIFAR-10 and CINIC-10 datasets, $T = 500$ and $T_P=40$ for CIFAR-100 and TinyImagenet datasets. The mean values for the average accuracy over the last 100 epochs are calculated from 10 experiments using 5 different random seeds and can be found in Table 1, Table 2, Table 3 and Table 5 (presented in Supplementary). For easy comparison, each table includes FedAvg results on the full dataset, with the best pruning method results bolded.

5.2. Performance Analysis

ResNet-18 on CIFAR-10. As shown in Table 1. When using the entire dataset for training, overfitting occurs due to the imbalance in data distribution. Double pruning can effectively mitigate overfitting by easing the imbalance of data distribution. Therefore, the model’s performance after pruning still improves even comparing with those whose whole dataset are already trained. Our evaluation indicates that FedCS has achieved highest performance at all pruning rates, especially under high pruning rates. For instance,

Table 4. Top-1 Accuracy (%) of Ablation experiments with ResNet18 network on CIFAR-10 dataset.

Methods	CA	DP	$\alpha = 0.1, p_i^l = 0.1$				$\alpha = 1.0, p_i^l = 0.1$			
			$p_i^f = 0.3$	$p_i^f = 0.5$	$p_i^f = 0.7$	$p_i^f = 0.9$	$p_i^f = 0.3$	$p_i^f = 0.5$	$p_i^f = 0.7$	$p_i^f = 0.9$
			$p_i = 0.30$	$p_i = 0.44$	$p_i = 0.57$	$p_i = 0.70$	$p_i = 0.32$	$p_i = 0.47$	$p_i = 0.61$	$p_i = 0.76$
FedCS w/o CA, DP			57.41 $\pm$ 1.01	56.56 $\pm$ 1.25	52.76 $\pm$ 1.45	51.55 $\pm$ 1.57	63.80 $\pm$ 0.72	61.28 $\pm$ 0.86	59.32 $\pm$ 1.01	55.14 $\pm$ 1.12
FedCS w/o DP	✓		57.77 $\pm$ 1.09	57.43 $\pm$ 1.16	55.37 $\pm$ 1.30	53.16 $\pm$ 1.45	64.23 $\pm$ 0.64	61.67 $\pm$ 0.75	60.01 $\pm$ 0.81	55.75 $\pm$ 0.93
FedCS w/o CA		✓	58.22 $\pm$ 0.93	58.11 $\pm$ 1.02	56.26 $\pm$ 1.11	53.78 $\pm$ 1.19	64.90 $\pm$ 0.57	63.37 $\pm$ 0.64	61.28 $\pm$ 0.73	57.51 $\pm$ 0.82
FedCS	✓	✓	59.62$\pm$0.79	59.16$\pm$0.81	58.88$\pm$0.86	56.46$\pm$0.88	66.11$\pm$0.49	63.82$\pm$0.60	61.64$\pm$0.71	58.42$\pm$0.76
Whole Dataset			55.63 $\pm$ 0.68	55.63 $\pm$ 0.68	55.63 $\pm$ 0.68	55.63 $\pm$ 0.68	66.04 $\pm$ 0.24	66.04 $\pm$ 0.24	66.04 $\pm$ 0.24	66.04 $\pm$ 0.24

when $\alpha = 0.1$, $p_i^f = 0.9$ and $p_i^l = 0.1$, FedCS achieves 56.46% accuracy after pruning, outperforming other methods by 0.94% to 5.82%, and even 0.83% higher than using the entire dataset. It confirms our conclusion that, selecting hard samples in FL with high degree of non-IID can lead to a suboptimal coreset. The comparison under 2 different α values demonstrates that our method performs better with severe data distribution heterogeneity.

ViT-B-16 on CIFAR-100. In this experiment, we load with the ViT-B-16 pre-trained weights. As shown in Table 2, FedCS demonstrates the best performance across various pruning rates with an accuracy improvement of 0.75%-2.42% over other methods. When we use the entire dataset for training, overfitting does not occur, however, FedCS consistently achieves performance closer to that of using the full dataset under all pruning rates. It further confirms that our proposed method not only improves training efficiency but also pays attention to data distribution.

ResNet-50 on TinyImageNet. In this experiment, we load with the ResNet-50 pre-trained weights provided by PyTorch. As shown in Table 2 and 3, FedCS also achieves the best performance across experiments with all pruning rates and a large number of sample label types. When there are many categories but few samples per category, some clients may lack samples for certain categories. Experiments show that FedCS uses class center aggregation to enable clients to access information from missing categories, aiding in coreset selection. Our method performs exceptionally well even not all clients are involved in the updates.

5.3. Ablation Study

Experimental Settings. For the ablation study, we conduct three experiments to validate each design component. First, we conduct FedCS w/o CA, DP which uses DC scores without class center aggregation (CA) and double pruning (DP). Then, we create FedCS w/o DP and FedCS w/o CA by sequentially adding CA and DP respectively. We train ResNet-18 on CIFAR-10 using these methods across 10 clients, with updates performed on all clients, by setting Dirichlet coefficient α to 0.1 and 1.0, β to 0.5 and 0.65,

communication round $T = 200$, pre-training communication round $T_P = 4$ and synchronization interval $I = 5$. For methods without double pruning, we first calculate the actual pruning rate p_i , then prune based on the p_i to ensure consistency in the pruning ratio. For each α value, we set 4 different p_i^f and p_i^l to get 4 values of p_i . We found that, for the same α , the actual pruning rate remains nearly identical across trials. To ensure fairness, we conduct multiple experiments and report the average accuracy. The mean accuracy over the last 100 epochs is averaged from 10 experiments with 5 random seeds and can be found in Table 4. For easy comparison, each table includes FedAvg results on the full dataset, with the best pruning method results bolded.

Evaluation of Ablation Study Findings. As shown in Table 4, when $\alpha = 0.1$, accuracy decreases by 1.15%-2.68% without CA and 1.73%-3.52% without DP, highlighting the effectiveness of each. Removing both CA and DP leads to a decrease of 2.21%-6.12%, further emphasizing their importance. Similar trends are observed When $\alpha = 1.0$. These results confirm the effectiveness of our framework, particularly under high pruning rates. Additionally, in Tables 1, 2, 3 and 5 (in Supplementary), the GM method also perform better than the Moderate, further validating the role of CA.

6. Conclusion

In this paper, we propose FedCS as an efficient coreset selection method in FL. After calculating the DC between a sample's distance to the corresponding class center and to the nearest other class center, we select samples with low DC scores to construct the coreset. Sufficient theoretical proof has demonstrated the rationality of FedCS, and extensive experiments have demonstrated the superiority of FedCS over other methods in FL, especially under high pruning rates and high heterogeneity. Additionally, we design ablation experiments to demonstrate the effectiveness of each module in FedCS. These findings highlight the potential of FedCS to significantly reduce computational costs in FL and provide a powerful solution in achieving lossless network performance while improving efficiency.

Acknowledgement

This work was supported by National Natural Science Foundation of China under Grants (62322117, 62371365, U24B20136, and U22B2014) and Fundamental Research Funds for the Central Universities and Innovation Fund of Xidian University (YJSJ25007).

References

- [1] Lusine Abrahamyan, Yiming Chen, Giannis Bekoulis, and Nikos Deligiannis. Learned gradient compression for distributed deep learning. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 33(12):7330–7344, 2021. 2
- [2] Fadhel Ayed and Soufiane Hayou. Data pruning and neural scaling laws: Fundamental limitations of score-based algorithms. *arXiv preprint arXiv:2302.06960*, 2023. 2
- [3] Debraj Basu, Deepesh Data, Can Karakus, and Suhas Digavi. Qsparse-local-sgd: Distributed sgd with quantization, sparsification and local computations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 6
- [4] Chia-Yu Chen, Jungwook Choi, Daniel Brand, Ankur Agrawal, Wei Zhang, and Kailash Gopalakrishnan. Ada-comp: Adaptive residual gradient compression for data-parallel distributed training. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 2827–2835, 2018. 2
- [5] Junming Chen, Meirui Jiang, Qi Dou, and Qifeng Chen. Federated domain generalization for image recognition via cross-client style transfer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 361–370, 2023. 1
- [6] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Towards understanding biased client selection in federated learning. In *Artificial Intelligence and Statistics Conference (AISTATS)*, 2022. 6
- [7] C Coleman, C Yeh, S Mussmann, B Mirzsoleiman, P Bailis, P Liang, J Leskovec, and M Zaharia. Selection via proxy: Efficient data selection for deep learning. In *Proceeding of the International Conference on Learning Representations (ICLR)*, 2020. 2
- [8] Xiyang Dai, Dongdong Chen, Mengchen Liu, Yinpeng Chen, and Lu Yuan. Da-nas: Data adapted pruning for efficient neural architecture search. In *European Conference on Computer Vision (ECCV)*, pages 584–600, 2020. 2
- [9] Luke N Darlow, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey. Cinic-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505*, 2018. 6
- [10] Eros Fani, Marco Ciccone, and Barbara Caputo. Feddrive v2: An analysis of the impact of label skewness in federated semantic segmentation for autonomous driving. *arXiv preprint arXiv:2309.13336*, 2023. 1
- [11] Shangqian Gao, Junyi Li, Zeyu Zhang, Yanfu Zhang, Weidong Cai, and Heng Huang. Device-wise federated network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12342–12352, 2024. 2
- [12] Muyang He, Shuo Yang, Tiejun Huang, and Bo Zhao. Large-scale dataset pruning with dynamic uncertainty. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7713–7722, 2024. 2
- [13] 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. 7
- [14] Hong Huang, Weiming Zhuang, Chen Chen, and Lingjuan Lyu. Fedmef: Towards memory-efficient federated dynamic pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27548–27557, 2024. 1
- [15] Xijie Huang, Zechun Liu, Shih-Yang Liu, and Kwang-Ting Cheng. Efficient quantization-aware training with adaptive coreset selection. *arXiv preprint arXiv:2306.07215*, 2023. 2
- [16] Yuang Jiang, Shiqiang Wang, Victor Valls, Bong Jun Ko, Wei-Han Lee, Kin K Leung, and Leandros Tassioulas. Model pruning enables efficient federated learning on edge devices. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 34(12):10374–10386, 2022. 1, 2
- [17] Jakub Konečný, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016. 2
- [18] Nikolas Koutsoubis, Yasin Yilmaz, Ravi P. Ramachandran, Matthew Schabath, and Ghulam Rasool. Privacy preserving federated learning in medical imaging with uncertainty estimation. *arXiv e-prints*, art. arXiv:2406.12815, 2024. 1
- [19] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6
- [20] Hojun Lee, Suyoung Kim, Junhoo Lee, Jaeyoung Yoo, and Nojun Kwak. Coreset selection for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7682–7691, 2024. 2
- [21] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations (ICLR)*. 6
- [22] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and Bill Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. In *Proceeding of the International Conference on Learning Representations (ICLR)*, 2018. 2
- [23] Qianyu Long, Christos Anagnostopoulos, Shameem Puthiya Parambath, and Daning Bi. Feddip: Federated learning with extreme dynamic pruning and incremental regularization. In *IEEE International Conference on Data Mining (ICDM)*, pages 1187–1192, 2023. 2
- [24] Chengjie Ma. Federated learning based on pruning and recovery. *arXiv preprint arXiv:2403.15439*, 2024. 2
- [25] Jeffrey Ma, Alan Tu, Yiling Chen, and Vijay Janapa Reddi. Fedstaleweight: Buffered asynchronous federated learning with fair aggregation via staleness reweighting. *arXiv preprint arXiv:2406.02877*, 2024. 2
- [26] Adyasha Maharana, Prateek Yadav, and Mohit Bansal. D2 pruning: Message passing for balancing diversity and dif-

- difficulty in data pruning. *arXiv preprint arXiv:2310.07931*, 2023. 2
- [27] Max Marion, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. When less is more: Investigating data pruning for pretraining llms at scale. *arXiv preprint arXiv:2309.04564*, 2023. 1, 4
- [28] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics Conference (AISTATS)*, pages 1273–1282, 2017. 6, 7
- [29] Mark Huasong Meng, Sin G Teo, Guangdong Bai, Kailong Wang, and Jin Song Dong. Enhancing federated learning robustness using data-agnostic model pruning. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pages 441–453, 2023. 1, 2
- [30] Rishabh Patra, Ramya Hebbalaguppe, Tirtharaj Dash, Gautam Shroff, and Lovekesh Vig. Calibrating deep neural networks using explicit regularisation and dynamic data pruning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1541–1549, 2023. 1, 2
- [31] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 20596–20607, 2021. 2, 7
- [32] Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger. Identifying mislabeled data using the area under the margin ranking. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 17044–17056, 2020. 2
- [33] Ziheng Qin, Kai Wang, Zangwei Zheng, Jianyang Gu, Xiangyu Peng, Daquan Zhou, Lei Shang, Baigui Sun, Xunsong Xie, Yang You, et al. Infobatch: Lossless training speed up by unbiased dynamic data pruning. In *Proceeding of the International Conference on Learning Representations (ICLR)*, 2023. 2
- [34] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015. 4
- [35] Donald Shenaj, Marco Toldo, Alberto Rigon, and Pietro Zanuttigh. Asynchronous federated continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5055–5063, 2023. 2
- [36] Durga Sivasubramanian, Lokesh Nagalapatti, Rishabh Iyer, and Ganesh Ramakrishnan. Gradient coreset for federated learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2648–2657, 2024. 2
- [37] Sebastian Urban Stich. Local sgd converges fast and communicates little. In *Proceeding of the International Conference on Learning Representations (ICLR)*, 2019. 6
- [38] Haoru Tan, Sitong Wu, Fei Du, Yukang Chen, Zhibin Wang, Fan Wang, and Xiaojuan Qi. Data pruning via moving-one-sample-out. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2
- [39] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018. 2, 7
- [40] Thijs Vogels, Sai Praneeth Karimireddy, and Martin Jaggi. Powersgd: Practical low-rank gradient compression for distributed optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2
- [41] Hengyi Wang, Weiyang Xie, Jitao Ma, Daixun Li, and Yunsong Li. Fedsls: Exploring federated aggregation in saliency latent space. In *Proceedings of the 32nd ACM International Conference on Multimedia (ACMMM)*, pages 7182–7190, 2024. 7
- [42] Qiyuan Wang, Qianqian Yang, Shibo He, Zhiguo Shi, and Jiming Chen. Asyncfeded: Asynchronous federated learning with euclidean distance based adaptive weight aggregation. *arXiv preprint arXiv:2205.13797*, 2022. 2
- [43] Xiaobo Xia, Jiale Liu, Jun Yu, Xu Shen, Bo Han, and Tongliang Liu. Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *Proceeding of the International Conference on Learning Representations (ICLR)*, 2022. 2, 7
- [44] Weiyang Xie, Haowei Li, Jitao Ma, Yunsong Li, Jie Lei, Donglai Liu, and Leyuan Fang. Jointsq: Joint sparsification-quantization for distributed learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5778–5787, 2024. 2
- [45] Chenhao Xu, Youyang Qu, Yong Xiang, and Longxiang Gao. Asynchronous federated learning on heterogeneous devices: A survey. *Computer Science Review*, 50:100595, 2023. 2
- [46] Shuo Yang, Zeke Xie, Hanyu Peng, Min Xu, Mingming Sun, and Ping Li. Dataset pruning: Reducing training data by examining generalization influence. *arXiv preprint arXiv:2205.09329*, 2022. 2
- [47] Xin Zhang, Jiawei Du, Yunsong Li, Weiyang Xie, and Joey Tianyi Zhou. Spanning training progress: Temporal dual-depth scoring (tdds) for enhanced dataset pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26223–26232, 2024. 2
- [48] Ligeng Zhu, Hongzhou Lin, Yao Lu, Yujun Lin, and Song Han. Delayed gradient averaging: Tolerate the communication latency for federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 29995–30007, 2021. 2