frontiers | Frontiers in Neuroinformatics

TYPE Original Research
PUBLISHED 09 September 2024
DOI 10.3389/fninf.2024.1430987

OPEN ACCESS

EDITED BY
Rashid Ibrahim Mehmood,
Islamic University of Madinah, Saudi Arabia

REVIEWED BY
Paola Galdi,
University of Edinburgh, United Kingdom
Aftab Khan,
Bristol Research and Innovation Laboratory (BRIL), United Kingdom

*CORRESPONDENCE
Bishal Thapaliya
✉ bthapaliya16@gmail.com

[†]These authors have contributed equally to this work and share first authorship

RECEIVED 10 May 2024
ACCEPTED 08 August 2024
PUBLISHED 09 September 2024

CITATION
Thapaliya B, Ohib R, Geenjaar E, Liu J, Calhoun V and Plis SM (2024) Efficient federated learning for distributed neuroimaging data. *Front. Neuroinform.* 18:1430987. doi: 10.3389/fninf.2024.1430987

Efficient federated learning for distributed neuroimaging data

Bishal Thapaliya[1,2*†], Riyasat Ohib[1,3†], Eloy Geenjaar[1,3], Jingyu Liu[1,2], Vince Calhoun[1,2,3] and Sergey M. Plis[1,2]

[1]Translational Research In Neuroimaging and Data Science Center, Atlanta, GA, United States, [2]Department of Computer Science, Georgia State University, Atlanta, GA, United States, [3]School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, United States

Recent advancements in neuroimaging have led to greater data sharing among the scientific community. However, institutions frequently maintain control over their data, citing concerns related to research culture, privacy, and accountability. This creates a demand for innovative tools capable of analyzing amalgamated datasets without the need to transfer actual data between entities. To address this challenge, we propose a decentralized sparse federated learning (FL) strategy. This approach emphasizes local training of sparse models to facilitate efficient communication within such frameworks. By capitalizing on model sparsity and selectively sharing parameters between client sites during the training phase, our method significantly lowers communication overheads. This advantage becomes increasingly pronounced when dealing with larger models and accommodating the diverse resource capabilities of various sites. We demonstrate the effectiveness of our approach through the application to the Adolescent Brain Cognitive Development (ABCD) dataset.

KEYWORDS

efficient federated learning, neuroimaging, sparse models, communication efficiency, sparsity

1 Introduction

Deep learning has transformed fields like computer vision, natural language processing, and is also starting to transform the field of neuroimaging. As deep learning models grow, distributed and collaborative training becomes essential, especially when sensitive data is spread across distant sites. Collaborative MRI data analysis offers profound insights, allowing researchers to utilize data beyond a study's original scope. As MRI scans are often preserved, vast amounts of data accumulate across decentralized research sites. Training models on more data, while preserving data privacy is thus crucial. Aggregating data from different sources to a central server for training can however expose this sensitive information, raising ethical concerns. Federated Learning (FL), an emerging paradigm in machine learning aims to leverage this distributed data while maintaining privacy. It achieves this by enabling devices or organizations to train models locally and share only aggregated training updates instead of raw data.

In FL, a central server coordinates training, and client sites communicate only model parameters, keeping local data private. In the decentralized setting, the server usually doesn't exist and clients train a model collaboratively among themselves. However, challenges arise due to data's statistical heterogeneity, limited communication bandwidth, and computational costs. Various methods have been proposed to address the high communication and computational costs of federated learning. Inspired by the findings from the lottery ticket hypothesis (Frankle and Carbin, 2019) which discovered that there exists *sub-networks* (a subset of network parameters within the larger complete neural network) which can be trained in isolation to almost full accuracy, many methods were proposed to train and update only a sub-network in the client sites (Dai et al., 2022; Huang H. et al., 2022). However, finding these sub-networks in the traditional method (Frankle and Carbin, 2019) is extremely computationally intensive and thus FL methods that rely on them (Huang H. et al., 2022) also share the same issues. Although initiating the federated training process from a random sub-network and updating the network in later work (Dai et al., 2022) brought about the benefits of both computational and communication efficiency, it came at the cost of performance due to starting the FL training process with random sub-networks. In this work we aim to solve this issue of starting from random sub-networks for the sparse FL process, targeted toward neuroimaging data.

We introduce *Sparse Federated Learning for NeuroImaging* or NeuroSFL a communication efficient federated learning method that identifies salient sub-networks at each client sites and trains sparse local models, greatly reducing the communication bandwidth. A notable difference of our method in contrast to competiting methods such as DisPFL (Dai et al., 2022) is that, NeuroSFL enjoys the benefits of sparse models at local cites such as faster inference (Dey et al., 2019) on top of the communication efficiency of sparse communications methods (Vahidian et al., 2021; Dai et al., 2022; Isik et al., 2022).

1.1 Contributions

NeuroSFL is a sparse federated learning method that discovers a common sub-network from the available data distributed across local sites and trains sparse local models leveraging the distributed data. Our key contributions are as follows:

1. We introduce NeuroSFL, a communication efficient federated learning approach geared toward training on distributed neuroimaging data in different client sites.
2. Our method identifies a global common sub-network at initialization and keeps this sub-network static throughout the federated learning process. Consequently, it only needs to share the sub-network masks *only once* before training begins, and never again, significantly reducing the communication overhead during training.
3. NeuroSFL does not need to share dense model parameters or masks during the training phase as it starts with a common initialization and only transmit sparse parameters each communication round depending on the chosen sparsity level.
4. We validate our method in a neuroimaging task and demonstrate its efficacy compared to competing methods.
5. Finally, unlike most competiting methods, to test the effectiveness of NeuroSFL, we also deploy and evaluate it in a real-world federated learning framework called COINSTAC (Plis et al., 2016) that trains neuroimaging models and report wall-clock time speed up.

2 Backgrounds and related works

In this section, we provide the necessary background for this work by introducing the federated learning problem in Section 2.1. We then discuss the related works in Section 2.3.

2.1 Federated learning

Federated Learning (FL) (McMahan et al., 2017) represents a novel approach in machine learning, facilitating model training across numerous decentralized devices or servers that hold local data samples without needing to exchange them. This contrasts sharply with traditional distributed learning methods, which centralize data and distribute computations. FL prioritizes privacy preservation, efficient communication, and resilience in diverse, heterogeneous environments. It diverges from conventional distributed learning paradigms, due to its distinct characteristics, some of which we detail below:

1. Non-IID data: the training data across different clients are not identically distributed, which means that the data at each local site may not accurately represent the overall population distribution.
2. Unbalanced data: the amount of data varies significantly across clients, leading to imbalances in data representation.
3. Massive distribution: often, the number of clients exceeds the average number of samples per client, illustrating the scale of distribution.
4. Limited communication: communication is infrequent, either among clients in a decentralized setting or between clients and the server in a centralized setting, due to slow and expensive connections.
5. Heterogeneous devices: clients in FL may have diverse computational capabilities, ranging from powerful servers to resource-constrained mobile devices.
6. Privacy preservation: FL is designed to ensure that raw data never leaves the clients' devices, preserving user privacy. Instead of sharing data, only model updates are shared. Although more sophisticated techniques have been proposed to both break the privacy guaranteed by vanilla FL (Geiping et al., 2020) and preserve the privacy (Zhang et al., 2023).
7. Local training: each client performs local training on its own data and only shares updates (e.g., weights or gradients) with the central server, which then aggregates these updates to improve the global model.
8. Client availability: clients may be intermittently available due to power constraints, connectivity issues, or user activities, requiring the system to be robust to varying participation.
9. Scalability: FL frameworks are designed to handle a large number of clients, scaling from hundreds to potentially millions of devices.

One of the main focuses of this work is to reduce the communication costs between the server and clients in a centralized setting or among clients in a decentralized setting when dealing with non-IID and unbalanced data. This is achieved by identifying a sub-network based on the data distributions at each local site and transmitting only the parameters of this sub-network in each communication round r. In each round, a fixed set of $\tilde{K}$ clients is sampled from all K clients, and federated training continues on the selected sub-network of those clients. The general federated optimization problem encountered is detailed next.

2.2 Federated optimization problem

In the general federated learning (FL) setting, a central server tries to find a global statistical model by periodically communicating with a set of clients. The federated averaging algorithm proposed by Konečnỳ et al. (2016), McMahan et al. (2017), and Bonawitz et al. (2019) is applicable to any finite sum objective of the form

$$\min_{\theta \in \mathbb{R}^d} f(\theta), \quad \text{where} \quad f(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta). \tag{1}$$

In a typical machine learning problem, the objective function $f_i(\theta) = \ell(x_i, y_i; \theta)$ is encountered, where the i^{th} term in the sum is the loss of the network prediction on a sample (x_i, y_i) made by a model with parameter θ. We assume that the data is partitioned over a total of K clients, with $\mathcal{P}_k$ denoting the set of indices of the samples on client k, and $n_k = |\mathcal{P}_k|$. The total number of samples n is given by $n = \sum_{k=1}^{K} n_k$. Thus, the objective in Equation 1 can be re-written as follows in Equation 2

$$f(\theta) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(\theta), \quad \text{where } F_k(\theta) = \frac{1}{n_k} \sum_{i \in \mathcal{P}_k} f_i(\theta). \tag{2}$$

In the typical distributed optimization setting, the IID assumption is made, which says the following: if the partition $\mathcal{P}_k$ was created by distributing the training data over the set of clients uniformly at random, then we would have $\mathbb{E}_{\mathcal{P}_k}[F_k(\theta)] = f(\theta)$, where the expectation is over the set of examples assigned to a fixed client k. In this work, we consider the non-IID setting where this does not hold and F_k could be an arbitrarily bad approximation to f.

When designing an FL training paradigm, a set of core considerations have to be made to maintain data privacy and address *statistical* or *objective* heterogeneity due to the differences in client data and resource constraints at the client sites. A range of work tries to address the issue of heterogeneous non-IID data (McMahan et al., 2016; Kulkarni et al., 2020), however, some research also suggests that deterioration in accuracy in the FL non-IID setting is almost inevitable (Zhao et al., 2018).

2.3 Related works

In this section, we discuss the relevant literature in relation to NeuroSFL. First, in Section 2.3.1, we describe the role of federated learning in neuroimaging and discuss the relevant literature. Second, in Section 2.3.2, we introduce key works on model pruning and sparsity in deep learning, findings from which we leverage for formulating NeuroSFL. Third, in Section 2.3.3, we describe applications of model pruning and sparsity in the FL setting for efficient FL. Finally, in Section 2.3.4, we briefly discuss privacy in the FL setting.

2.3.1 Federated learning in neuroimaging

Over the past decade, the field of neuroimaging has strongly embraced data sharing, open-source software, and collaboration across multiple sites. This shift is largely driven by the need to offset the high costs and time demands associated with neuroimaging data collection (Landis et al., 2016; Rootes-Murdy et al., 2022). By pooling data from different sources, researchers can explore findings that extend beyond the initial scope of individual studies (Poldrack et al., 2013). The practice of sharing data enhances the robustness of research through larger sample sizes and the replication of results, offering significant benefits for neuroimaging studies. Even though data pooling and sharing data is embraced, there are significant challenges related to data privacy, security, and governance that limit the extent to which data can be shared. This is where FL becomes crucial as it enables collaborative model training across multiple institutions without the need to directly share sensitive data. Moreover, with FL collaborative training, sample size also plays a crucial role, where increasing the sample size not only makes predictions more reliable but also ensures the reliability and validity of research findings, thereby preventing data manipulation and fabrication (Tenopir et al., 2011; Ming et al., 2017). Furthermore, aggregating data can lead to a more diverse sample by combining otherwise similar datasets, thus reflecting a broader range of social health determinants for more comprehensive results (Laird, 2021). Additionally, reusing data can significantly reduce research costs (Milham et al., 2018).

FL is increasingly recognized as a transformative approach in healthcare and neuroimaging. In the realm of biomedical imaging, FL has been applied to a variety of tasks. These include whole-brain segmentation from MRI T1 scans (Roy et al., 2019), segmentation of brain tumors (Li et al., 2019; Sheller et al., 2019), multi-site fMRI classification, and the identification of disease biomarkers (Li X. et al., 2020). COINSTAC (Plis et al., 2016) offers a privacy-focused distributed data processing framework specifically designed for brain imaging showcasing FL's role in enhancing privacy and efficiency in healthcare data analysis. Additionally, it has been utilized in discovering brain structural relationships across various diseases and clinical cohorts through federated dimensionality reduction from shape features (Silva et al., 2019).

2.3.2 Role of model pruning in reducing computational demands

The primary objective of *model pruning* is to identify sub-networks within larger architectures by selectively removing connections. This technique holds considerable appeal for various reasons, particularly for real-time applications on resource-constrained edge devices, which are prevalent in federated learning (FL) and collaborative learning scenarios. Pruning large networks can significantly alleviate the computational demands of inference (Elsen et al., 2020) or hardware tailored to exploit sparsity (Cerebras, 2019; Pool et al., 2021). More recently, the *lottery ticket hypothesis* has emerged (Frankle and Carbin, 2019), suggesting the existence of sub-networks within densely connected networks.

These sub-networks, when trained independently from scratch, can attain comparable accuracy to fully trained dense networks (Frankle and Carbin, 2019), revitalizing the field of sparse deep learning (Chen et al., 2020; Renda et al., 2020). This resurgence of interest has also extended into sparse reinforcement learning (RL) (Arnob et al., 2021; Sokar et al., 2021). Pruning techniques in deep learning can broadly be categorized into three groups: methods that induce sparsity before training and during initialization (Lee et al., 2018; Tanaka et al., 2020; Wang et al., 2020; Ohib et al., 2023), during training (Zhu and Gupta, 2018; Ma et al., 2019; Yang et al., 2019; Ohib et al., 2022), and post-training (Han et al., 2015; Frankle et al., 2021). In this work, we leverage findings from methods that induce sparsity at initialization, specifically parameter saliency metrics, to formulate NeuroSFL.

2.3.3 Efficiency in federated learning

For pruning in the FL setting, using a *Lottery Ticket* like approach would result in immense inefficiency in communication. Such methods (Frankle and Carbin, 2019; Bibikar et al., 2022) usually require costly pruning and retraining cycles, often training and pruning multiple times to achieve the desired accuracy vs sparsity trade-off. Relatively few research have leveraged pruning in the FL paradigm (Li A. et al., 2020, 2021; Jiang et al., 2022). In particular, with LotteryFL (Li A. et al., 2020) and PruneFL (Jiang et al., 2022), clients need to send the full model to the server regularly resulting in higher bandwidth usage. Moreover, in Li A. et al. (2020), each client trains a personalized mask to maximize the performance only on the local data. A few recent works (Li A. et al., 2020; Bibikar et al., 2022; Huang T. et al., 2022; Qiu et al., 2022) also attempted to leverage sparse training within the FL setting as well. In particular, Li A. et al. (2020) implemented randomly initialized sparse mask, FedDST (Bibikar et al., 2022) built on the idea of RigL (Evci et al., 2020) which is a prune and re-grow technique, and mostly focussed on magnitude pruning on the server-side resulting in similar constraints and (Ohib et al., 2023) uses sparse gradients to efficiently train in a federated learning setting. In this work, we try to alleviate these limitations which we discuss in the following section.

2.3.4 Privacy in federated learning

Even without sharing raw data, FL can still be vulnerable to privacy attacks such as gradient inversion attacks (Geiping et al., 2020), which can sometimes compromise privacy. Traditional FL algorithms, like federated stochastic gradient descent, are particularly susceptible to these attacks, although methods like Federated Averaging (FedAvg) (McMahan et al., 2017) mitigate this vulnerability to some extent (Geiping et al., 2020; Dimitrov et al., 2022).

Recent research has explored various privacy-preserving techniques in FL. Differential privacy has been proposed to add noise to the model updates to provide strong privacy guarantees (Abadi et al., 2016). Secure aggregation methods ensure that aggregated updates are protected against eavesdropping and manipulation during transmission (Bonawitz et al., 2017). Furthermore, advancements in cryptographic techniques, such as homomorphic encryption and secure multiparty computation, offer promising solutions for preserving privacy in FL settings (Mohassel and Zhang, 2017; Juvekar et al., 2018).

These approaches aim to enhance the robustness of Federated Learning against privacy threats while enabling collaborative model training across distributed data sources. In this work, we primarily focus on improving communication efficiency in FL systems. Although we do not explicitly address privacy, our method can be used in conjunction with other privacy-preservation techniques.

3 Method description

In this section we present our proposed method. We first describe the process of discovering a sub-network $f(\boldsymbol{\theta} \odot \mathbf{m})$ within the full network $f(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$, the mask $\mathbf{m} \in \mathbb{R}^d$, with $\|\mathbf{m}\|_0 < d$. To discover a performant sub-network an importance scoring metric is required, which we describe in Section 3.1.1. Finally, we delineate our proposed method in Section 3.2.

3.1 Sub-network discovery

Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ at a site k, the training of a neural network f parameterized by $\theta \in \mathbb{R}^d$ can be written as minimizing the following empirical risk as in Equation 3:

$$\underset{\theta}{\operatorname{argmin}} \frac{1}{n} \sum_i \mathcal{L}(f(\boldsymbol{\theta}; \boldsymbol{x}_i), y_i) \quad \text{s.t. } \boldsymbol{\theta} \in \mathcal{H} \tag{3}$$

where $\boldsymbol{\theta} \in \mathbb{R}^d$ and $\mathcal{L}$ and $\mathcal{H}$ are the loss function and the constraint set respectively.

In general, in unconstrained (standard) training the set of possible hypotheses is considered to be $\mathcal{H} = \mathbb{R}^d$, where d is the model dimension. The objective is to minimize the empirical risk $\mathcal{L}$ given a training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^n \sim \mathcal{D}$ at the local client site k. Given access to the gradients of the empirical risk on a batch-wise basis, an optimization algorithm such as Stochastic Gradient Descent (SGD) is typically employed to achieve the specified objective. This process generates a series of parameter estimates, $\{\theta_i\}_{i=0}^T$, where θ_0 represents the initial parameters and θ_T the final optimal parameters. A sub-network within this network is defined as a sparse version of this network with a mask $\mathbf{m} \in \{0,1\}^{|\boldsymbol{\theta}|}$ that results in a masked network $f(\boldsymbol{\theta} \odot \mathbf{m}; \boldsymbol{x}_i)$. When aiming for a target sparsity level where $k < d$, the parameter pruning challenge entails ensuring that the final optimal parameters, θ_T, have at most k non-zero elements, as denoted by the constraint $\|\theta_T\|_0 \leq k$. In many works, this sparsity constraint applies only to the final parameters and not to any intermediate parameter estimates. However, in this work we maintain this sparsity constraint throughout the entire training phase, that is throughout the entire evolution of $\boldsymbol{\theta}$ from $\boldsymbol{\theta}_0$ to $\boldsymbol{\theta}_T$.

The goal of discovering sub-networks at initialization introduces additional constraints to the previously described framework by requiring that all parameter iterations fall within a predetermined subspace of $\mathcal{H}$. Specifically, the constraints seek to identify an initial set of parameters, θ_0, that has no more than k_1 non-zero elements ($\|\theta_0\|_0 \leq k_1$), and ensure that all intermediate parameter sets, θ_i, belong to a subspace $\bar{\mathcal{H}} \subset \mathcal{H}$ for all i in

$\{1, \ldots, T\}$, where $\bar{\mathcal{H}}$ is the subspace of $\mathbb{R}^d$ spanned by the natural basis vectors $\{e_j\}_{j \in \text{supp}(\theta_0)}$. Here, $\text{supp}(\theta_0)$ represents the support of θ_0, or the set of indices corresponding to its non-zero entries. This approach not only specifies a sub-network at initialization with k parameters but also maintains its structure consistently throughout the training.

3.1.1 Connection importance criterion

Lee et al. (2018) introduced a technique for estimating the importance of a connection in a deep learning network inspired by the saliency criterion originally proposed by Mozer and Smolensky (1988). They contributed an important insight, demonstrating that this criterion is remarkably effective in predicting the significance of each connection in a neural network at the initialization phase. The core concept revolves around retaining those parameters that, when altered, would have the most substantial effect on the loss function. This is operationalized by considering a binary vector $c \in \{0,1\}^m$ and utilizing the Hadamard product $\odot$. Consequently, SNIP calculates the sensitivity of connections based on this approach as following:

$$s(\boldsymbol{\theta}; \mathcal{D}) := \left.\frac{\partial \mathcal{L}(\boldsymbol{\theta} \odot \mathbf{c})}{\partial \mathbf{c}}\right|_{\mathbf{c}=1} = \frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \odot \boldsymbol{\theta} \tag{4}$$

After determining $s(\boldsymbol{\theta})$, the parameters associated with the highest k magnitudes of $|s(\boldsymbol{\theta}; \mathcal{D})_i|$ are retained, where i corresponds to the indices of the selected parameters. Essentially, SNIP calculates the importance score of each parameter as its product with the incoming gradient. It prioritizes weights that, regardless of their direction, are distant from the origin and yield large gradient values. It's noteworthy that the objective of SNIP can be reformulated as noted by De Jorge et al. (2020) and Frankle et al. (2021):

$$\max_{c} S(\boldsymbol{\theta}, \mathbf{c}) := \sum_{i \in \text{supp}(\mathbf{c})} |\theta_i \nabla L(\boldsymbol{\theta})_i| \quad \text{s.t.} \quad \mathbf{c} \in \{0,1\}^m, \|\mathbf{c}\|_0 = q. \tag{5}$$

where S is defined to be the saliency scores. It is trivial to note that the optimal solution to the above problem can be obtained by selecting the indices corresponding to the top-q values of $s_i = |\theta_i \nabla \mathcal{L}(\theta)_i|$.

3.1.2 Iterative connection importance criterion

In this section, we test the effectiveness of iterative-SNIP (De Jorge et al., 2020), which is an iterative version of the application of saliency criterion in Equation 4. We briefly describe the iterative-SNIP next. We assume q to be the number of parameters to be preserved post pruning. Given that we have some pruning schedule (similar to learning rate schedule: linear, exponential etc.) to divide q into a set of natural numbers $\{k_t\}_{t=1}^T$ such that $q_t > q_{t+1}$ and $q_T = q$. Now, given the binary masking variable $\boldsymbol{c}_t$ corresponding to q_t, the formulation of pruning from q_t to q_{t+1} can be made using the connection sensitivity (4) similar to De Jorge et al. (2020) as:

$$\boldsymbol{c}_{t+1} = \underset{\hat{\theta}, c}{\text{argmax}}\ S(\bar{\boldsymbol{\theta}}, \boldsymbol{c}) \quad \text{s.t.}\ \ \boldsymbol{c} \in \{0,1\}^m,\ \ \|\boldsymbol{c}\|_0 = k_{q+1},\ \boldsymbol{c} \odot \boldsymbol{c}_t = \boldsymbol{c}, \tag{6}$$

where $\bar{\boldsymbol{\theta}} = \boldsymbol{\theta} \odot \boldsymbol{c}_t$. The constraint $\boldsymbol{c} \odot \boldsymbol{c}_t = \boldsymbol{c}$ is added to ensure that no previously pruning parameter is re-activated. Assuming that the pruning schedule ensures a smooth transition from one topology to another ($\|\boldsymbol{c}_t\|_0 \approx \|\boldsymbol{c}_{t+1}\|_0$) such that the *gradient approximation* $\left.\frac{\partial \mathcal{L}(\bar{\boldsymbol{\theta}})}{\partial \bar{\boldsymbol{\theta}}}\right|_{\boldsymbol{c}_t} \approx \left.\frac{\partial \mathcal{L}(\bar{\boldsymbol{\theta}})}{\partial \bar{\boldsymbol{\theta}}}\right|_{\boldsymbol{c}_{t+1}}$ is valid, Equation 6 can be approximated as solving Equation 5 at $\bar{\boldsymbol{\theta}}$. In the scenario where the schedule parameter is set to $T = 1$, the original SNIP saliency method is recovered. This is basically employing a *gradient approximation* approach between the initial dense network $\boldsymbol{c}_0 = \mathbf{1}$ and the resulting mask $\boldsymbol{c}$. We conduct experiments with IterativeSNIP in the federated neuroimaging setting and present our findings in Section 5.2.

3.2 Proposed method

We propose a novel method for *efficient distributed sub-network discovery* for distributed neuroimaging and propose a method for training such sparse models or subnetworks in a communication efficient manner called *Sparse Federated Learning for NeuroImaging* or NeuroSFL with the goal of tackling communication inefficiency during decentralized federated learning with non-IID data distribution in the context of distributed neuroimaging data. The proposed method initiates with the common initialization $\boldsymbol{\theta}_0$ at all the local client models. Next, importance scores s_j are calculated for each model parameter in the network based on the information from the imaging data available across all the client sites. At this stage, each client has a unique set of importance scores for their parameters in the local network f based on the local data available at that site similar to Lee et al. (2018) and De Jorge et al. (2020). As shown in Equation 7, all the clients transmit these scores to each other and a mask $\mathbf{m}$ is created corresponding to the top-q % of the aggregated saliency scores:

$$\mathbf{m} = T_q\left(\sum_{k=0}^{K-1} s_k\right) \tag{7}$$

where the T_q is the top-q operator that retains the top q percentage of the s_k values by magnitude and sets the rest to zero. This mask is then used to train the model $f_k(\boldsymbol{\theta} \odot \mathbf{m}; \boldsymbol{x})$ at site k on their local data $(\boldsymbol{x}, y) \sim \mathcal{D}_k$.

For the federated training among a total of K clients, the clients are trained locally, and at the end of local training they share their trained parameters which are then averaged; we call this a *communication round*. At the start of this local training, each site k starts with the same initial model weights $\boldsymbol{\theta}_0$ which at each site k is denoted as $\theta_{k,0}$ at training step $t = 0$ which are then masked with the generated saliency mask $\mathbf{m}$ to produce the common masked initialization $\theta_{k,0}^m$ as follows:

$$\theta_{k,0}^m = \theta_{k,0} \odot \mathbf{m}$$

Next these models at each site k are trained on their local dataset $(\boldsymbol{x}, y) \sim \mathcal{D}_k$.

The masked models $f(\theta_{k,0} \odot \mathbf{m})$ across all the sites are trained for a total of T communication rounds to arrive at the final weights $\theta_{k,T}$ at each local site. In each communication round t, only a random subset $\mathcal{F}' = \{f_1, f_2, ..., f_{K'}\}$ of K' clients where $\mathcal{F}' \subseteq \mathcal{F}$ the set of all clients, and $K' \leq K$ are trained on their local data. These K' clients are sampled uniformly at random without replacement in a given round but with replacement across different rounds. We sample a subset of clients uniformly instead of including all the clients in a single communication round because previous works have shown that it is computationally more efficient and including more clients in a single round leads to diminishing returns (McMahan et al., 2016). This approach is also a standard practice in the federated learning (FL) literature (Yang et al., 2018; Reddi et al., 2020; Sun et al., 2020; Dai et al., 2022). Since each client has an equal probability of being chosen for participation in a given communication round, over the course of enough communication rounds, all clients will eventually participate. In this work, we train our FL pipeline for a total of $T = 500$ communication rounds, similar to Dai et al. (2022).

At the end of local training on the random subset $\mathcal{F}'$, the updated weights of the selected clients are aggregated to get the new updated parameters $\hat{\boldsymbol{\theta}}^m_{k,t}$, which would be the starting weights for the next communication round. When sharing the updated weights only the weights corresponding to the 1's in the binary mask $\mathbf{m}$ are shared among the clients and with the server, as only these weights are being trained and the rest of the weights are *zero-ed* out. This results in the gains in communication efficiency. To efficiently share the model weights, the clients only share their sparse masked weights $\boldsymbol{\theta}^m_{\mathcal{F}'} = \boldsymbol{\theta}_{\mathcal{F}'} \odot \mathbf{m}$ among the selected clients in $\mathcal{F}'$ using the compressed sparse row (CSR) encoding. The algorithm for the training process is delineated in Algorithm 1.

```
Input: Total number of clients $K$; Total communication
    rounds $T$
Output: Sparse local models $\hat{\boldsymbol{\theta}}^{\mathcal{C}}_m$
Initialize local models with $\boldsymbol{\theta}_0$ and transmit to
    all clients.
$s_k \leftarrow S_k(\boldsymbol{\theta}_0, \mathcal{D}_k)$ # generate saliency scores at each
    site $k$ and share the scores to the server
$\mathbf{m} = T_k(\sum_{k=0}^{K-1} s_k)$ # generate a common global mask from
    importance scores at the server
Transmit m to all the sites $K$.
$\boldsymbol{\theta}^m_{k,0} \leftarrow \boldsymbol{\theta}_{k,0} \odot \mathbf{m}$ #apply the mask at all sites $k =$
    $1, 2, ..., K$
for $t = 0$ to $T - 1$ do
    $\{c_i\}_{i=1}^{\tilde{K}}$ # Sample a set of $\tilde{K}$ clients uniformly
    from the set of all clients
    for site $k$ in parallel for all $\tilde{K}$ clients do
        $\hat{\boldsymbol{\theta}}^m_{k,t} \leftarrow$ csr($\theta^m_{S,t}$); #Gather masked weights $\theta_{k,m}$
        from the server
        for $\tau = 0$ to $N - 1$ do
            Sample a batch of data $\delta_{k,t,\tau}$ from the
            local dataset.
            $\mathbf{g}^m_{k,t,\tau} \leftarrow \nabla_{\boldsymbol{\theta}} \mathcal{L}(\hat{\boldsymbol{\theta}}^m_{k,t,\tau}; \delta_{k,t,\tau}) \odot \mathbf{m}$ # calculate and
            mask gradients
            $\hat{\boldsymbol{\theta}}^m_{k,t,\tau+1} \leftarrow \hat{\boldsymbol{\theta}}^m_{k,t,\tau} - \eta \mathbf{g}^m_{k,t,\tau}$# take optimization
            step with masked gradients on masked
            weights
        end for
        transmit the non-zero elements of the
        updated model $\hat{\boldsymbol{\theta}}^m_{k,t,N-1}$ back to all clients.
    end for
    $\boldsymbol{\theta}^m_{S,t} \leftarrow \sum_{k \in \tilde{K}} \hat{\boldsymbol{\theta}}^m_{k,t,N-1}$ # Aggregate the masked
    non-zero weights in the server
end for
```

Algorithm 1. NeuroSFL.

4 Experiments

4.1 Dataset and non-IID partition

We evaluated NeuroSFL on the ABCD dataset. ABCD study is the largest long-term study of brain development and child health in the US. It recruited over 10 thousand children of 9 and 10 years old from 21 sites and followed them for 10 years with annual behavioral and cognitive assessments and biannual MRI scans (Garavan et al., 2018). Along with multi-session brain MRI scans for structure and function, the ABCD study also includes key demographic information including gender, racial information, socio-economic backgrounds, cognitive development, and mental and physical health assessments of the subjects. The ABCD open-source dataset can be found on the National Institute of Mental Health Data Archive (NDA) (https://nda.nih.gov/). In this study, we used data from the ABCD baseline, which contain 11,875 participants aged 9–10 years.

T1-weighted MRI images were preprocessed using the Statistical Parametric Mapping 12 (SPM12) software toolbox for registration, normalization, and tissue segmentation. Then the gray matter density maps were smoothed by a 6 mm^3 Gaussian kernel, creating images with the dimensionality of (121, 145, 121) of voxels at Montreal Neuroimaging Institute (MNI) space with each voxel having dimensions of $1.5 \times 1.5 \times 1.5^3$ mm.

We simulated the heterogeneous data distributions across federated clients through the adoption of two distinct data partitioning strategies. We outline these strategies for generating non-IID data partitions with a comprehensive discussion in Section 4.1.1.

4.1.1 Generating non-IID data partition with Dirichlet distribution

In contrast to centralized data-center training where data batches are often independent and identically distributed (IID), federated learning typically deals with non-IID data distributions across different clients. Hence, to evaluate novel federated learning methods it is crucial to not make the IID assumption to better reflect the real world setting and instead generate non-IID data among clients for evaluation (Hsu et al., 2019). In this section, we discuss the process of generating non-identical data distribution in the client sites using the Dirichlet Distribution, specifically for the context of federated learning.

4.1.1.1 Generating non-IID data from Dirichlet distribution

In this study, we assume that each client independently chooses training samples. These samples are classified into N distinct classes, with the distribution of class labels governed by a probability vector $\boldsymbol{q}$, which is non-negative and whose components sum to 1, that is, $q_i > 0, i \in [1, N]$ and $\|\boldsymbol{q}\|_1 = 1$. For generating a group of non-identical clients, $q \sim \text{Dir}(\alpha \boldsymbol{p})$ is drawn from the Dirichlet Distribution, with $\boldsymbol{p}$ characterizing a prior distribution over the N classes and α controls the degree of identicality among the existing clients and is known as the *concentration parameter*.

In this section, we generate a range of client data partitions from the Dirichlet distribution with a range of values for the concentration parameter α for exposition. In Figure 1, we generate a group of 10 balanced clients, each holding an equal number of total samples. Similar to Hsu et al. (2019) the prior distribution $\boldsymbol{p}$ is assumed to be uniform across all classes. For each client, given a concentration parameter α, we sample a $\boldsymbol{q}$ from $\text{Dir}(\alpha)$ and allocate the corresponding fraction of samples from each client to that client. Figure 1 illustrates the effect of the concentration parameter α on the class distribution drawn from the Dirichlet distribution on different clients, for the CIFAR-10 dataset. When $\alpha \rightarrow \infty$, identical class distribution is assigned to each classes. With decreasing α, more non-identicalness is introduced in the class distribution among the client population. At the other extreme with $\alpha \rightarrow 0$, each client only consists of one particular class. To create a more realistic FL scenario, we used the value of $\alpha = 0.3$ for all of our experiments.

4.2 Architecture, hyperparameters and experimental details

Here we provide a comprehensive overview of the architecture, hyperparameters, and the experimental setup we use to evaluate our proposed NeuroSFL method on the neuroimaging Adolescent Brain Cognitive Development (ABCD) data. Our study focuses on the task of classifying a participant's sex based on MRI scans, by employing a 3D variant of the well-known AlexNet model (Krizhevsky et al., 2012). The 3D variant was referenced from Abrol et al. (2021), which has a specific channel configuration for the convolutional layers set as: 64C-128C-192C-192C-128C, where "C" denotes channels.

We optimized the learning rate for this task through an exhaustive search ranging from LR $= 1 \times 10^{-3}$ to 1×10^{-6}, achieving a delicate balance between rapid convergence and fine-tuning during training. We employed a batch size of 32 and a learning rate decay factor of 0.998 was applied. We applied varying sparsity levels, ranging from 0%, 50%, 80%, 90%, and 95% to assess the overall performance. A random split of 80/20 was used for training and testing within each individual site. For nonIID setting, as we used alpha = 0.3 for Dirichlet distribution, we enforced an additional constraint of having at least five samples in each client, in order to perform stable training and perform random 80/20 split similar to IID setting. Our training consists of five epochs with 200 communication rounds.

4.3 Baselines

We compared our method with both centralized and decentralized baselines. Centralized baseline includes FedAvg (McMahan et al., 2017), FedAvg-FT (Cheng et al., 2021) which are the standard dense baselines, and for the decentralized FL setting, we take the sparse Dis-PFL (Dai et al., 2022).

In FedAvg (McMahan et al., 2017), each client trains its local model using its local data, and then these local models are aggregated or averaged to update the global model. On the other hand, FedAvg-FT (Cheng et al., 2021) extends the FedAvg algorithm by incorporating fine-tuning or transfer learning. Specifically, after the global model is trained using FedAvg, the global model is then fine-tuned or adapted using additional data from a central server or other external sources. This fine-tuning step allows the global model to adapt to new tasks or data distributions beyond what was initially learned from the federated learning process. We also compare with DisPFL (Dai et al., 2022) with varying sparsity levels. DisPFL is a new sparse FL technique that randomly prunes each layer similar to Evci et al. (2020) and uses the prune and regrow method from that work as well, resulting in a dynamically sparse method. The prune and regrow method involves periodically pruning a fraction of the network's weights to zero, and then regrowing new weights in their place, allowing the model to dynamically adjust its sparsity pattern during training.

In exploring the impact of using unique local masks instead of a global mask on the performance of FL, we established IndividualSNIP as a baseline, representing an approach where unique local masks are devised from the saliency criterion, and local models are trained based on these masks. Moreover, to isolate the impact of just using *global masking*, that is using the same random mask in all clients, instead of using different unique random masks at different sites we compare our method and competing methods against random global masking as well in Figure 2.

Additionally, with further experiment on how different methods of model pruning and selection impact the performance of our approach, we further experiment with other techniques named IterSNIP and WeightedSNIP. IterSNIP builds upon the traditional SNIP method (Lee et al., 2018) by incorporating multiple minibatches during the training process of mask generation. This approach aggregates saliency scores from these minibatches to generate a comprehensive and robust pruning mask. Conversely, WeightedSNIP adopts a different strategy, deriving a global mask through a weighted average of saliency scores based on the frequency of data at each site, and assigning importance levels to individual sites based on the amount of data at sites.

4.4 Experiments in real-world FL system

In order to demonstrate the use-case of the NeuroSFL in real world scenario, we further aim to perform extensive experiments in a real-world FL system by making use of

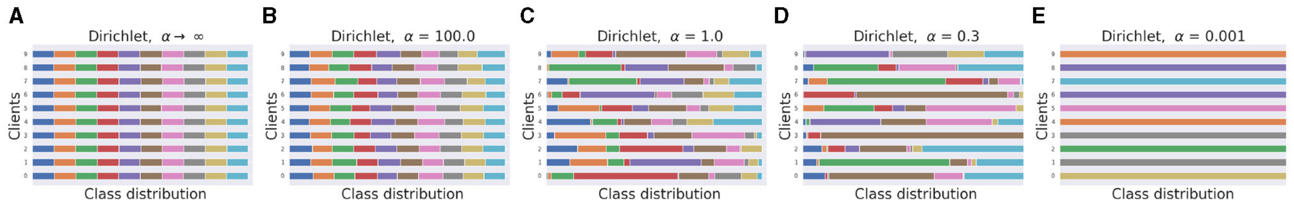

FIGURE 1
Generating non-identical client data partitions using the Dirichlet Distribution for the Cifar10 dataset among 10 clients. Distribution among classes is represented using different colors. **(A)** Dirichlet, $\alpha \rightarrow \infty$ results in identical clients **(B–D)** Client distributions generated from Dirichlet distributions with different concentration parameters α **(E)** Dirichlet, $\alpha \rightarrow 0.0$ results in each client being assigned only one particular class.

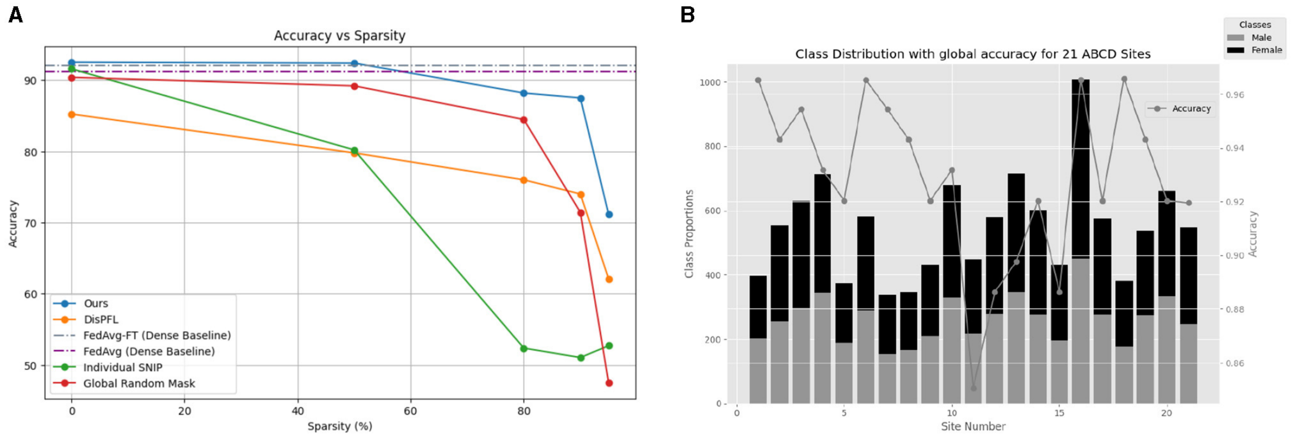

FIGURE 2
(A) Comparison of methods for gender classification using MRI Scans of ABCD dataset. **(B)** Gender differences in each of the 21 ABCD sites along with the performance of the global model on each site with 50% sparsity.

Coinstac (Plis et al., 2016), a cutting-edge open-source federated learning solution designed for collaborative neuroimaging endeavors at scale. Deployed in real-world scenarios, Coinstac embodies a paradigm shift in collaborative research, transcending traditional boundaries and fostering synergistic interactions among researchers worldwide. Coinstac's architecture facilitates decentralized computations across a distributed network of geographically dispersed client nodes, seamlessly integrating diverse computational tasks while safeguarding data privacy through state-of-the-art differential privacy mechanisms. Having said this, our experiment leverages Coinstac's robust infrastructure to benchmark our method against the standard dense FedAvg algorithm (McMahan et al., 2017) within a practical real-world context. Our evaluation spans five diverse client locations, spanning from North Virginia to Frankfurt, each representing a distinct geographical node within Coinstac's decentralized network. By meticulously assessing the mean communication timereflecting the duration for the server model to aggregate all client weights during each communication round, we demonstrate the efficiency of our algorithm in optimizing federated learning workflows. Our investigation encompasses five local client models, each featuring varying sizes or depths of ResNet architectures while maintaining a sparsity level of 90% across experiments.

5 Results and discussion

5.1 Effect of varying sparsity levels

We first explore the effect of sparsity on IID data in Section 5.1.1 and then explore the efficacy of NeuroSFL on non-IID data in Section 5.1.2.

5.1.1 Effect of varying sparsity levels on IID data

The performance of various methods across different sparsity levels was evaluated, as presented in Table 1, and visually presented in Figure 2. Sparse baselines, including Ours (*NeuroSFL*), IndividualSNIP, DisPFL (Dai et al., 2022), and Global Random Mask, were compared against dense baselines such as FedAvg-FT (Cheng et al., 2021) and FedAvg (McMahan et al., 2017). Notably, our proposed *NeuroSFL*, exhibited robust performance across varying sparsity levels, achieving an accuracy of 92.52% at 0% sparsity and maintaining high accuracy even at higher sparsity levels, with 71.18% accuracy at 95% sparsity. In comparison, IndividualSNIP demonstrated decreasing accuracy as sparsity increased, with a significant drop to 52.70% at 95% sparsity. This is in line with expectation as individual-SNIP only incorporates the saliency scores from a single site at random and

TABLE 1 Performance comparison of different methods and sparsity levels.

Method	Sparsity (%)				
	0%	50%	80%	90%	95%
Sparse baselines					
Ours (NeuroSFL)	92.52%	92.4%	88.19%	87.5%	71.18%
DisPFL	85.24%	79.78%	76.00%	74.01%	62.12%
IndividualSNIP	91.59%	80.20%	52.37%	51.04%	52.70%
Global Random Mask	90.39%	89.20%	84.48%	71.44%	47.53%
Dense baselines					
FedAvg-FT	92.1% (dense baseline)				
FedAvg	90.5% (dense baseline)				

Sparse baselines include Ours- (NeuroSFL), DistPFL, IndividualSNIP, and Global Random Mask. Dense baselines include FedAvg and FedAvg-FT.

does not incorporate information from the datasets at all the participating cites.

Moreover, in contrast to NeuroSFL, DisPFL, and Global Random Mask also showcased diminishing accuracy with increasing sparsity, highlighting the effectiveness of our proposed approach in mitigating the adverse effects of sparsity on model performance on neuroimaging data. Notably, Global Random Mask outperformed DisPFL on lower sparsities, suggesting that in general global random masks might be more suitable for federated applications compared to even targeted unique local masks which DisPFL employs.

Dense baselines, such as FedAvg-FT and FedAvg, even while being *not sparse* and using full communication achieved comparable performances to NeuroSFL in the non-extreme sparsity region. NeuroSFL even surpassed the performance of dense baselines at a sparsity level of 50%, highlighting the effectiveness of our proposed sparse method in optimizing model performance while reducing communication costs. Furthermore, Figure 2B illustrates that the single global model trained with *NeuroSFL* demonstrated excellent performance for data within each site, emphasizing the model's effectiveness in capturing site-specific characteristics while maintaining high accuracy.

Additionally, in Figure 3, it is observed that the performance of local models trained with *NeuroSFL* remains consistently robust across non-IID states of local data, indicating the model's versatility and reliability in various data distribution scenarios.

5.1.2 Effect of varying sparsity levels on non-IID data

In this section, we explore the influence of changing sparsity levels on our model's performance on non-IID data across different client configurations: 10, 20, and 30 clients with f1-score as a metric. For all configurations, we employ the Dirichlet Distribution with alpha = 0.3 across various sparsity levels.

5.1.2.1 10 Clients

We begin by examining the model's performance with 10 clients. Figure 4 provides a visual representation of the F1-score versus sparsity relationship, showcasing the consistent performance achieved across different sparsity levels (Figure 4A). Additionally, Figure 4B illustrates the class distribution with Dir(0.3) for the ABCD dataset for 10 clients and their final local test F1-score. The Dirichlet partition results in an uneven data distribution, as visually confirmed by Figure 4B.

Notably, our model demonstrated robust performance across different sparsity levels, ranging from 84.75 to 92.07%. These results underscore the resilience of our approach in maintaining high performance even under significant sparsity constraints. This resilience suggests that our model's effectiveness extends beyond homogeneous datasets, making it suitable for deployment in federated learning scenarios with diverse client characteristics.

5.1.2.2 20 Clients

Expanding our analysis to 20 clients, we investigate how varying sparsity levels impact our model's performance on non-IID data. Figure 5 provides a visual representation of the F1-score versus sparsity relationship, highlighting the consistent performance achieved across different sparsity levels (Figure 5A). Additionally, Figure 5B illustrates the class distribution with Dir(0.3) for the ABCD dataset for 20 clients and their final local test F1-score.

The findings indicate a similar pattern to the 10-client case, with the model maintaining notable F1-scores across varying sparsity levels, with the lowest performance being 79.21% under the highly sparse constraint of 95% sparsity. This highlights the model's ability to generalize effectively even with significant data sparsity.

5.1.2.3 30 Clients

Finally, we extend our analysis to 30 clients to further understand the impact of varying sparsity levels on non-IID data with larger number of clients. Figure 6 provides a visual representation of the F1-score versus sparsity relationship, showcasing the consistent performance achieved across different sparsity levels (Figure 6A). Additionally, Figure 6B illustrates the class distribution with Dir(0.3) for the ABCD dataset for 30 clients and their final local test F1-score.

The results reveal a similar trend to the 20-client scenario, with the model achieving notable F1-scores across varying sparsity levels. However, for the highly sparse condition of 95%, there is a slight drop in F1-score to 76.42%. This decline

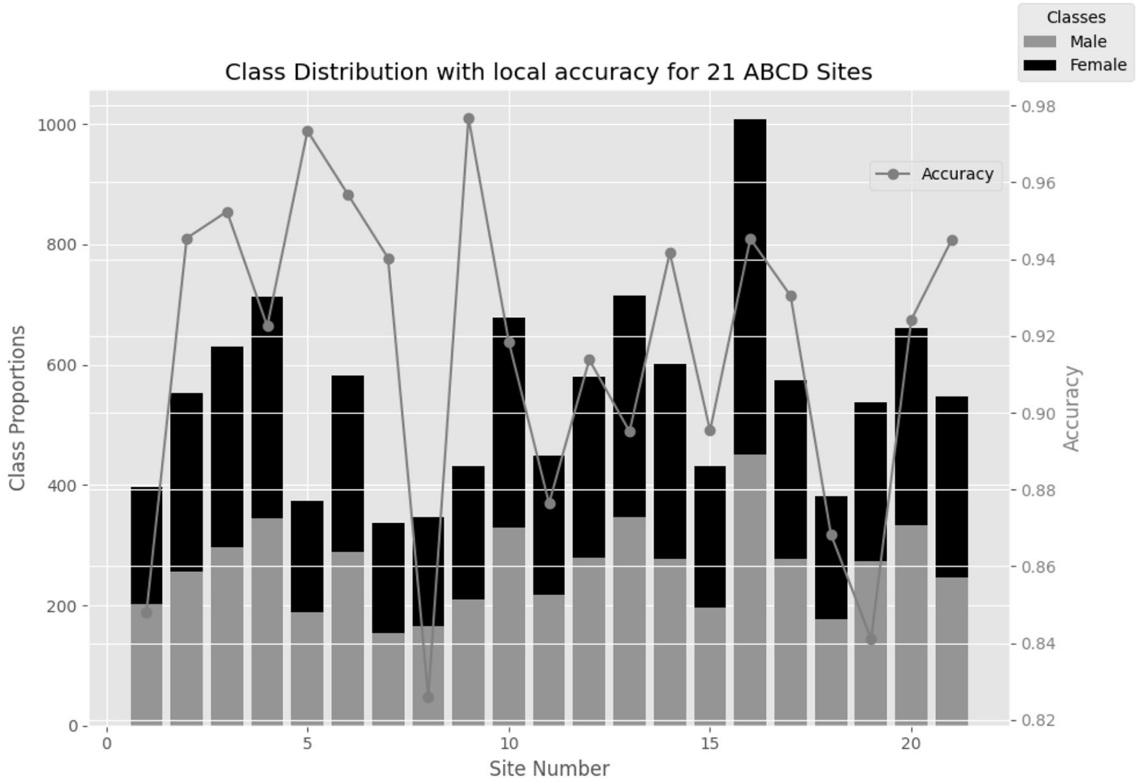

FIGURE 3
Gender differences in each of the 21 ABCD sites along with the performance of the local models on each site with 50% sparsity.

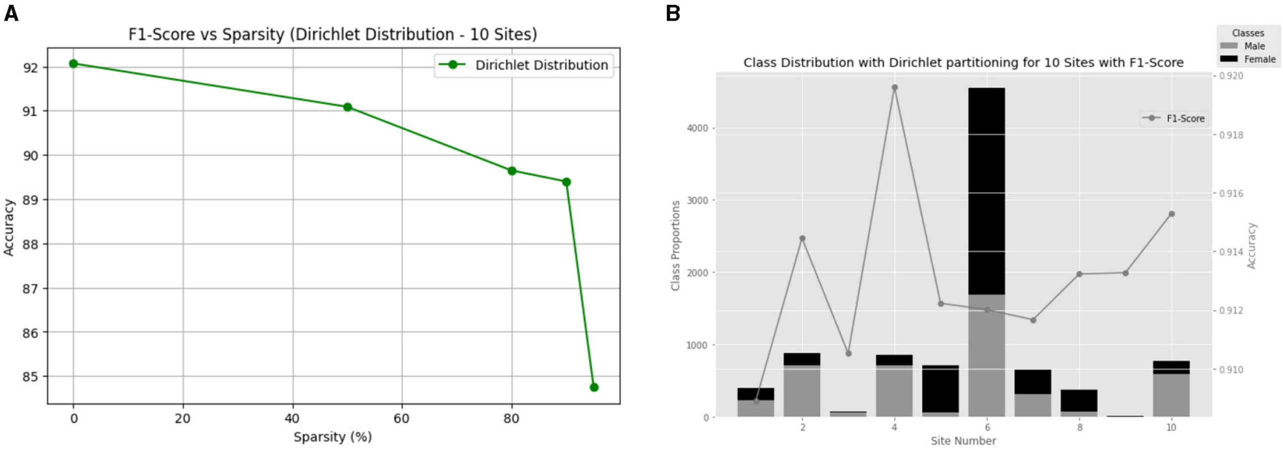

FIGURE 4
(A) Comparison of methods for gender classification using non-IID Dirichlet distribution with alpha = 0.3 and varying sparsity levels. **(B)** Gender differences in each of the 10 ABCD sites along with the performance of the model.

can be attributed to the increased difficulty for the model to generalize with such sparse data under the additional restriction of having larger clients. Despite this challenge, our model maintains its effectiveness across a diverse range of sparsity levels, indicating its potential for practical applications in federated learning scenarios with a larger number of client sites.

5.2 IterativeSNIP performance

We evaluate the performance of IterSNIP and WeightedSNIP to explore their efficacy in sparse FL scenarios. Table 2 summarizes the accuracy results obtained at 50% sparsity for different iterations of IterSNIP and WeightedSNIP. IterSNIP, with varying numbers of iterations (1, 10, and 20), demonstrated consistent performance

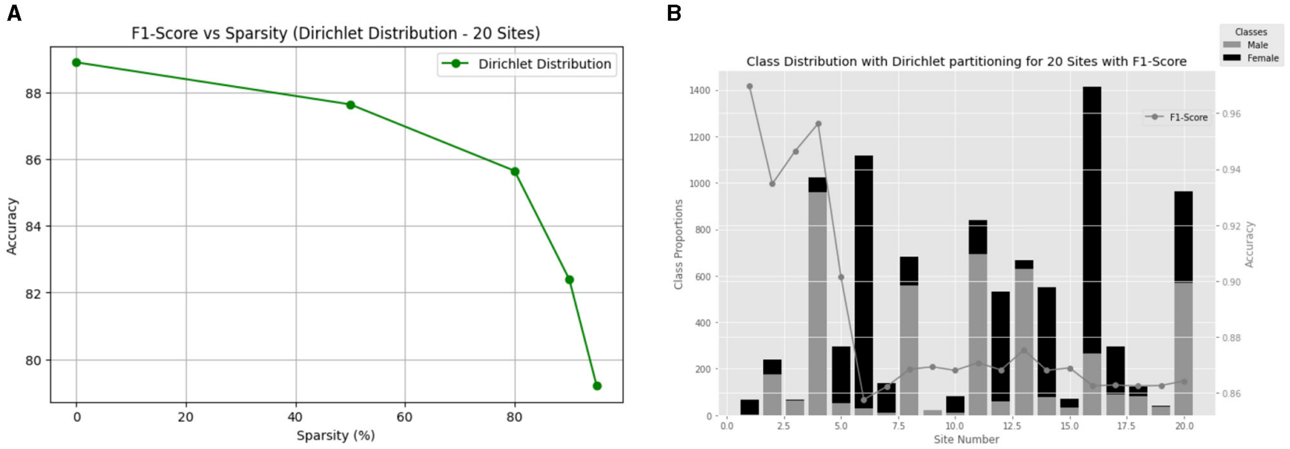

FIGURE 5
(A) Comparison of methods for gender classification using non-IID Dirichlet distribution with alpha = 0.3 and varying sparsity levels. **(B)** Gender differences in each of the 20 ABCD sites along with the performance of the model.

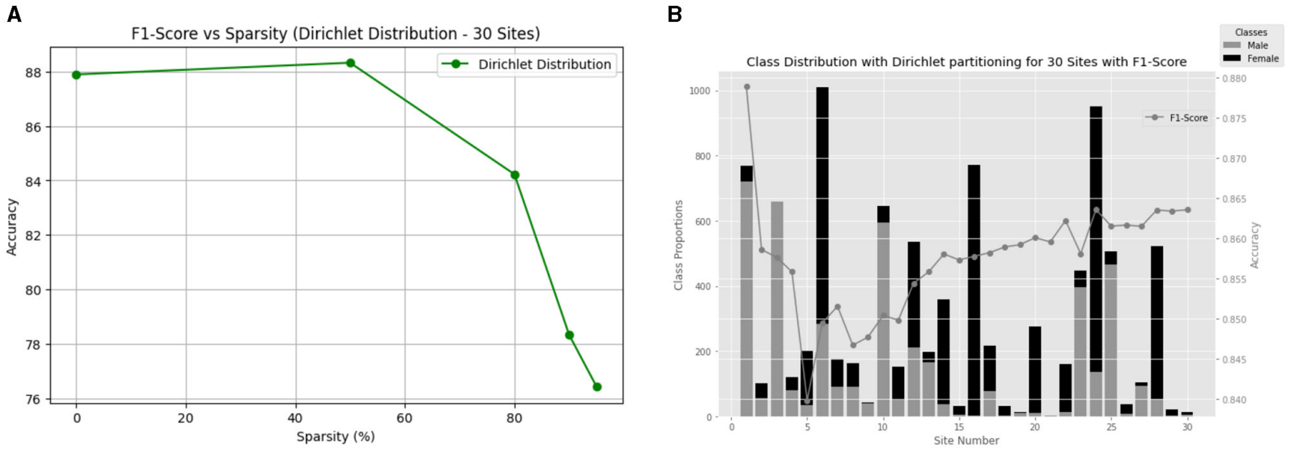

FIGURE 6
(A) Comparison of methods for gender classification using non-IID Dirichlet distribution with alpha = 0.3 and varying sparsity levels. **(B)** Gender differences in each of the 30 ABCD sites along with the performance of the model for 50% sparsity.

TABLE 2 **Performance comparison of IterSNIP with different iterations and WeightedSNIP in terms of accuracy at sparsity of 50%.**

Method	Iterations	Accuracy (50% sparsity)
IterSNIP	1 Iteration	92.40%
	10 Iteration	91.82%
	20 Iteration	92.67%
WeightedSNIP	1 Iteration	92.10%

with increasing iterations, achieving accuracies of 92.4%, 91.82%, and 92.67%, respectively. These results suggest that utilizing multiple iterations to obtain SNIP masks does not necessarily enhance model performance in scenarios with sparsity constraints, and especially when used on neuroimaging data. This is a departure from the single-node case on natural image datasets such as CIFAR10 or CIFAR100 (De Jorge et al., 2020). Similarly, WeightedSNIP, which incorporates weighted averages of saliency scores, achieved an accuracy of 92.10% and does not outperform the vanilla averaging technique. This proves that our model is robust enough to find a sparse mask, with minimal effect from the amount of data at each sites.

5.3 Wall-time efficiency gains in the real world COINSTAC system

The results of the ensuing comparative analysis as delineated in Table 3, demonstrate the tangible speed enhancements achieved by our proposed methodology NeuroSFL as compared to the standard FedAvg in a real-world setting. Importantly, our results indicate that our approach consistently outperforms FedAvg across

TABLE 3 Comparison of communication time between FedAvg and NeuroSFL on Cifar10 for ResNet architectures of different depth.

Architecture	Accuracy	Communication time (s)		Speed up
		FedAvg	NeuroSFL	
ResNet32	90.52%	0.285 ± 0.04	0.238 ± 0.02	1.20 *times*
ResNet44	89.65%	0.409 ± 0.06	0.328 ± 0.04	1.24 *times*
ResNet56	93.74%	0.531 ± 0.07	0.407 ± 0.06	1.30 *times*
ResNet110	93.25%	1.812 ± 0.33	0.781 ± 0.13	2.32 *times*

all ResNet architectures. For instance, in the case of ResNet32, our method achieves a communication time of 0.238 ± 0.02 s, compared to 0.285 ± 0.04 s for FedAvg, resulting in a speedup of 1.20*times*. This trend continues across deeper architectures, with our technique demonstrating significant improvements in communication efficiency. For instance, for ResNet110, our method achieves a remarkable speedup of 2.32x over FedAvg, showcasing its ability to handle complex models with greater efficiency. These empirical findings underscore the importance of sparse federated techniques like NeuroSFL, thereby propelling collaborative neuroimaging research to unprecedented heights.

5.4 Sparsity vs. accuracy performance comparison

In this section we analyze and interpret the results from Section 5. First, we probe the reasons behind the performance gains in comparison to a state of the art federated sparse learning method (Dai et al., 2022).

In a specific comparison with DistPFL, we can see that NeuroSFL consistently performs better than DisPFL in a range of sparsities in the selected tasks. This is probably due to a better choice of the initial sparse sub-network using the importance criterion. Another difference is that, in DisPFL different local clients have different levels of sparsity and a final model averaging is done, where the final model becomes denser due to the union of many sparse subnetworks. We however retain the same mask in all the clients and start from the same initialization in all the clients, result in equivalent sparsity in all the clients; this also leaves open the potential of keeping sparse global models in a centralized FL setting.

6 Conclusion and future work

In this work, we propose and analyze a novel communication-efficient FL method for neuroimaging called NeuroSFL. By extending a gradient-based parameter importance criterion to the FL setting, we achieve reduced communication costs and better bandwidth in decentralized training. Our method leverages the nature of local data distribution, resulting in a client data-aware global sparse mask. This leads to savings in communication time and bandwidth during sparse training. We tested our approach on the ABCD dataset and reported improved performance compared to contemporary methods. Overall, our sparse FL technique enhances communication time, making it suitable for bandwidth-limited settings without compromising accuracy.

However, more exploration is needed regarding privacy considerations and performance in more complex tasks. Although FL models inherently provide more privacy compared to other training pipelines, such as training with centralized data (Li Q. et al., 2021), they can still be susceptible to more sophisticated forms of attacks (Geiping et al., 2020). Sparse gradients can often result in more privacy-preserving methods (Zhang et al., 2023), hence it is likely our method would enjoy similar advantages. Moreover, our method should be easily extensible to incorporate differential privacy techniques (Ouadrhiri and Abdelhadi, 2022). We leave such explorations for future work.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: https://nda.nih.gov/.

Ethics statement

The studies involving humans were approved by Georgia State University (GSU) IRB. The studies were conducted in accordance with the local legislation and institutional requirements. Written informed consent for participation was not required from the participants or the participants' legal guardians/next of kin in accordance with the national legislation and institutional requirements.

Author contributions

BT: Software, Validation, Visualization, Writing – review & editing, Investigation. RO: Conceptualization, Formal analysis, Investigation, Methodology, Project administration, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. EG: Writing – review & editing. JL: Funding acquisition, Supervision, Writing – review & editing. VC: Conceptualization, Funding acquisition, Project administration, Resources, Supervision, Investigation, Writing – review & editing. SP: Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Validation, Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work was supported by R01DA040487 and R01DA049238 from NIH and 2112455 from NSF.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., et al. (2016). "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security* (New York, NY: ACM), 308–318. doi: 10.1145/2976749.2978318

Abrol, A., Fu, Z., Salman, M., Silva, R., Du, Y., Plis, S., et al. (2021). Deep learning encodes robust discriminative neuroimaging representations to outperform standard machine learning. *Nat. Commun.* 12:353. doi: 10.1038/s41467-020-20655-6

Arnob, S. Y., Ohib, R., Plis, S., and Precup, D. (2021). Single-shot pruning for offline reinforcement learning. *arXiv* [Preprint]. arXiv:2112.15579. doi: 10.48550/arXiv.2112.15579

Bibikar, S., Vikalo, H., Wang, Z., and Chen, X. (2022). Federated dynamic sparse training: computing less, communicating less, yet learning better. *Proc. AAAI Conf. Artif. Intell.* 36, 6080–6088. doi: 10.1609/aaai.v36i6.20555

Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., et al. (2019). Towards federated learning at scale: system design. *Proc. Mach. Learn. Syst.* 1, 374–388. doi: 10.48550/arXiv.1902.01046

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., et al. (2017). "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (New York, NY: ACM), 1175–1191. doi: 10.1145/3133956.3133982

Cerebras (2019). *Wafer Scale Engine: Why We Need Big Chips for Deep Learning*. Available at: https://cerebras.net/blog/cerebras-wafer-scale-engine-why-we-need-big-chips-for-deep-learning/ (accessed March 20, 2024).

Chen, T., Frankle, J., Chang, S., Liu, S., Zhang, Y., Wang, Z., et al. (2020). The lottery ticket hypothesis for pre-trained bert networks. *Adv. Neural Inf. Process. Syst.* 33, 15834–15846. doi: 10.48550/arXiv.2007.12223

Cheng, G., Chadha, K., and Duchi, J. (2021). Fine-tuning is fine in federated learning. *arXiv* [Preprint]. arXiv:2108.07313. doi: 10.48550/arXiv.2108.07313

Dai, R., Shen, L., He, F., Tian, X., and Tao, D. (2022). Dispfl: towards communication-efficient personalized federated learning via decentralized sparse training. *arXiv* [Preprint]. arXiv:2206.00187. doi: 10.48550/arXiv.2206.00187

De Jorge, P., Sanyal, A., Behl, H. S., Torr, P. H., Rogez, G., and Dokania, P. K. (2020). Progressive skeletonization: trimming more fat from a network at initialization. *arXiv* [Preprint]. arXiv:2006.09081. doi: 10.48550/arXiv.2006.09081

Dey, S., Huang, K.-W., Beerel, P. A., and Chugg, K. M. (2019). Pre-defined sparse neural networks with hardware acceleration. *IEEE J. Emerg. Sel. Top. Circuits Syst.* 9, 332–345. doi: 10.1109/JETCAS.2019.2910864

Dimitrov, D. I., Balunovic, M., Konstantinov, N., and Vechev, M. (2022). Data leakage in federated averaging. *Trans. Mach. Learn. Res.* doi: 10.48550/arXiv.2206.12395

Elsen, E., Dukhan, M., Gale, T., and Simonyan, K. (2020). "Fast sparse convnets," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (Seattle, WA: IEEE), 14629–14638. doi: 10.1109/CVPR42600.2020.01464

Evci, U., Gale, T., Menick, J., Castro, P. S., and Elsen, E. (2020). "Rigging the lottery: making all tickets winners," in *International Conference on Machine Learning* (Cambridge, MA: PMLR), 2943–2952.

Frankle, J., and Carbin, M. (2019). "The lottery ticket hypothesis: finding sparse, trainable neural networks," in *7th International Conference on Learning Representations, ICLR 2019* (New Orleans, LA).

Frankle, J., Dziugaite, G. K., Roy, D., and Carbin, M. (2021). "Pruning neural networks at initialization: why are we missing the mark?" in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Garavan, H., Bartsch, H., Conway, K., Decastro, A., Goldstein, R., Heeringa, S., et al. (2018). Recruiting the abcd sample: design considerations and procedures. *Dev. Cogn. Neurosci.* 32, 16–22. doi: 10.1016/j.dcn.2018.04.004

Geiping, J., Bauermeister, H., Dröge, H., and Moeller, M. (2020). Inverting gradients-how easy is it to break privacy in federated learning? *Adv. Neural Inf. Process. Syst.* 33, 16937–16947. doi: 10.48550/arXiv.2003.14053

Han, S., Mao, H., and Dally, W. J. (2015). Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv* [Preprint]. arXiv:1510.00149. doi: 10.48550/arXiv.1510.00149

Hsu, T.-M. H., Qi, H., and Brown, M. (2019). Measuring the effects of non-identical data distribution for federated visual classification. *arXiv* [Preprint]. arXiv:1909.06335. doi: 10.48550/arXiv.1909.06335

Huang, H., Zhang, L., Sun, C., Fang, R., Yuan, X., Wu, D., et al. (2022). Fedtiny: pruned federated learning towards specialized tiny models. *arXiv* [Preprint]. arXiv:2212.01977. doi: 10.48550/arXiv.2212.01977

Huang, T., Liu, S., Shen, L., He, F., Lin, W., Tao, D., et al. (2022). Achieving personalized federated learning with sparse local models. *arXiv* [Preprint]. arXiv:2201.11380. doi: 10.48550/arXiv.2201.11380

Isik, B., Pase, F., Gunduz, D., Weissman, T., and Zorzi, M. (2022). Sparse random networks for communication-efficient federated learning. *arXiv* [Preprint]. arXiv:2209.15328. doi: 10.48550/arXiv.2209.15328

Jiang, Y., Wang, S., Valls, V., Ko, B. J., Lee, W.-H., Leung, K. K., et al. (2022). Model pruning enables efficient federated learning on edge devices. *IEEE Trans. Neural Netw. Learn. Syst.* 10374–10386. doi: 10.1109/TNNLS.2022.3166101

Juvekar, C., Vaikuntanathan, V., and Chandrakasan, A. (2018). "$*GAZELLE*$: a low latency framework for secure neural network inferencem" in *27th USENIX security symposium (USENIX security 18)* (Baltimore, MD), 1651–1669.

Konečnỳ, J., McMahan, H. B., Ramage, D., Richtárik, P. (2016). Federated optimization: distributed machine learning for on-device intelligence. *arXiv* [Preprint]. arXiv:1610.02527. doi: 10.48550/arXiv.1610.02527

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems, Volume 25*, eds. F. Pereira, C. Burges, L. Bottou, and K. Weinberger (Red Hook, NY: Curran Associates, Inc).

Kulkarni, V., Kulkarni, M., and Pant, A. (2020). "Survey of personalization techniques for federated learning," in *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)* (London: IEEE), 794–797. doi: 10.1109/WorldS450073.2020.9210355

Laird, A. R. (2021). Large, open datasets for human connectomics research: considerations for reproducible and responsible data use. *Neuroimage* 244:118579. doi: 10.1016/j.neuroimage.2021.118579

Landis, D., Courtney, W., Dieringer, C., Kelly, R., King, M., Miller, B., et al. (2016). Coins data exchange: an open platform for compiling, curating, and disseminating neuroimaging data. *Neuroimage* 124, 1084–1088. doi: 10.1016/j.neuroimage.2015.05.049

Lee, N., Ajanthan, T., and Torr, P. H. (2018). Snip: single-shot network pruning based on connection sensitivity. *arXiv* [Preprint]. arXiv:1810.02340. doi: 10.48550/arXiv.1810.02340

Li, A., Sun, J., Wang, B., Duan, L., Li, S., Chen, Y., et al. (2020). Lotteryfl: personalized and communication-efficient federated learning with lottery ticket hypothesis on non-iid datasets. *arXiv* [Preprint]. arXiv:2008.03371. doi: 10.48550/arXiv.2008.03371

Li, A., Sun, J., Zeng, X., Zhang, M., Li, H., Chen, Y., et al. (2021). "Fedmask: joint computation and communication-efficient personalized federated learning via heterogeneous masking," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems* (New York, NY: ACM), 42–55. doi: 10.1145/3485730.3485929

Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li, Y., et al. (2021). A survey on federated learning systems: vision, hype and reality for data privacy and protection. *IEEE Trans. Knowl. Data Eng.* 35, 3347–3366. doi: 10.1109/TKDE.2021.3124599

Li, W., Milletarì, F., Xu, D., Rieke, N., Hancox, J., Zhu, W., Baust, M., Cheng, Y., Ourselin, S., Cardoso, M. J., et al. (2019). "Privacy-preserving federated brain tumour segmentation," in *Machine Learning in Medical Imaging: 10th International Workshop, MLMI 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13, 2019, Proceedings 10* (Cham: Springer), 133–141. doi: 10.1007/978-3-030-32692-0_16

Li, X., Gu, Y., Dvornek, N., Staib, L. H., Ventola, P., Duncan, J. S., et al. (2020). Multi-site fmri analysis using privacy-preserving federated learning and domain adaptation: abide results. *Med. Image Anal.* 65:101765. doi: 10.1016/j.media.2020.101765

Ma, R., Miao, J., Niu, L., and Zhang, P. (2019). Transformed ℓ_1 regularization for learning sparse deep neural networks. *Neural Netw.* 119, 286–298. doi: 10.1016/j.neunet.2019.08.015

McMahan, B., Moore, E., Ramage, D., Hampson, S., and Arcas, B. A. (2017). "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics* (Fort Lauderdale, FL: PMLR), 1273–1282.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and Arcas, B. A. (2016). "Communication-efficient learning of deep networks from decentralized data," in *International Conference on Artificial Intelligence and Statistics* (Fort Lauderdale, FL).

Milham, M. P., Craddock, R. C., Son, J. J., Fleischmann, M., Clucas, J., Xu, H., et al. (2018). Assessment of the impact of shared brain imaging data on the scientific literature. *Nat. Commun.* 9, 2818. doi: 10.1038/s41467-018-04976-1

Ming, J., Verner, E., Sarwate, A., Kelly, R., Reed, C., Kahleck, T., et al. (2017). Coinstac: decentralizing the future of brain imaging analysis. *F1000Res.* 6:1512. doi: 10.12688/f1000research.12353.1

Mohassel, P., and Zhang, Y. (2017). "Secureml: a system for scalable privacy-preserving machine learning," in *2017 IEEE symposium on security and privacy (SP)* (San Jose, CA: IEEE), 19–38. doi: 10.1109/SP.2017.12

Mozer, M. C., and Smolensky, P. (1988). Skeletonization: a technique for trimming the fat from a network via relevance assessment. *Adv. Neural Inf. Process. Syst.* 1.

Ohib, R., Gillis, N., Dalmasso, N., Shah, S., Potluru, V. K., Plis, S., et al. (2022). Explicit group sparse projection with applications to deep learning and NMF. *Trans. Mach. Learn. Res.* doi: 10.48550/arXiv.1912.03896

Ohib, R., Thapaliya, B., Gaggenapalli, P., Liu, J., Calhoun, V., Plis, S., et al. (2023). Salientgrads: sparse models for communication efficient and data aware distributed federated training. *arXiv* [Preprint]. arXiv:2304.07488. doi: 10.48550/arXiv.2304.07488

Ouadrhiri, A. E., and Abdelhadi, A. (2022). Differential privacy for deep and federated learning: a survey. *IEEE Access* 10, 22359–22380. doi: 10.1109/ACCESS.2022.3151670

Plis, S. M., Sarwate, A. D., Wood, D., Dieringer, C., Landis, D., Reed, C., et al. (2016). Coinstac: a privacy enabled model and prototype for leveraging and processing decentralized brain imaging data. *Front. Neurosci.* 10:204805. doi: 10.3389/fnins.2016.00365

Poldrack, R. A., Barch, D. M., Mitchell, J. P., Wager, T. D., Wagner, A. D., Devlin, J. T., et al. (2013). Toward open sharing of task-based fMRI data: the openfmri project. *Front. Neuroinform.* 7:12. doi: 10.3389/fninf.2013.00012

Pool, J., Sawarkar, A., and Rodge, J. (2021). *Accelerating Inference with Sparsity Using the NVIDIA Ampere Architecture and NVIDIA TensorRT*. Available at: https://developer.nvidia.com/blog/accelerating-inference-with-sparsity-using-ampere-and-tensorrt(accessed March 23, 2024).

Qiu, X., Fernandez-Marques, J., Gusmao, P. P., Gao, Y., Parcollet, T., Lane, N. D., et al. (2022). Zerofl: efficient on-device training for federated learning with local sparsity. *arXiv* [Preprint]. arXiv:2208.02507. doi: 10.48550/arXiv.2208.02507

Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., et al. (2020). Adaptive federated optimization. *arXiv* [Preprint]. arXiv:2003.00295. doi: 10.48550/arXiv.2003.00295

Renda, A., Frankle, J., and Carbin, M. (2020). Comparing rewinding and fine-tuning in neural network pruning. *arXiv* [Preprint]. arXiv:2003.02389. doi: 10.48550/arXiv.2003.02389

Rootes-Murdy, K., Gazula, H., Verner, E., Kelly, R., DeRamus, T., Plis, S., et al. (2022). Federated analysis of neuroimaging data: a review of the field. *Neuroinformatics* 20, 1–14. doi: 10.1007/s12021-021-09550-7

Roy, A. G., and Siddiqui, S. Pölsterl, S., Navab, N., Wachinger, C. (2019). Braintorrent: a peer-to-peer environment for decentralized federated learning. *arXiv* [Preprint]. arXiv:1905.06731. doi: 10.48550/arXiv.1905.06731

Sheller, M. J., Reina, G. A., Edwards, B., Martin, J., and Bakas, S. (2019). "Multi-institutional deep learning modeling without sharing patient data: a feasibility study on brain tumor segmentation," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part I 4* (Cham: Springer), 92–104. doi: 10.1007/978-3-030-11723-8_9

Silva, S., Gutman, B. A., Romero, E., Thompson, P. M., Altmann, A., Lorenzi, M., et al. (2019). "Federated learning in distributed medical databases: meta-analysis of large-scale subcortical brain data," in *2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019)* (Venice: IEEE), 270–274. doi: 10.1109/ISBI.2019.8759317

Sokar, G., Mocanu, E., Mocanu, D. C., Pechenizkiy, M., and Stone, P. (2021). Dynamic sparse training for deep reinforcement learning. *arXiv* [Preprint]. arXiv:2106.04217. doi: 10.48550/arXiv.2106.04217

Sun, L., Qian, J., and Chen, X. (2020). Ldp-fl: practical private aggregation in federated learning with local differential privacy. *arXiv* [Preprint]. arXiv:2007.15789. doi: 10.48550/arXiv.2007.15789

Tanaka, H., Kunin, D., Yamins, D. L., and Ganguli, S. (2020). "Pruning neural networks without any data by iteratively conserving synaptic flow," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, eds. H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin.

Tenopir, C., Allard, S., Douglass, K., Aydinoglu, A. U., Wu, L., Read, E., et al. (2011). Data sharing by scientists: practices and perceptions. *PLoS ONE* 6:e21101. doi: 10.1371/journal.pone.0021101

Vahidian, S., Morafah, M., and Lin, B. (2021). Personalized federated learning by structured and unstructured pruning under data heterogeneity. *arXiv* [Preprint]. arXiv:2105.00562. doi: 10.48550/arXiv.2105.00562

Wang, C., Zhang, G., and Grosse, R. B. (2020). "Picking winning tickets before training by preserving gradient flow," in *8th International Conference on Learning Representations, ICLR 2020* (Addis Ababa).

Yang, H., Wen, W., and Li, H. (2019). Deephoyer: learning sparser neural network with differentiable scale-invariant sparsity measures. *arXiv* [Preprint]. arXiv:1908.09979. doi: 10.48550/arXiv.1908.09979

Yang, T., Andrew, G., Eichner, H., Sun, H., Li, W., Kong, N., et al. (2018). Applied federated learning: improving google keyboard query suggestions. *arXiv* [Preprint]. arXiv:1812.02903. doi: 10.48550/arXiv.1812.02903

Zhang, Z., Tianqing, Z., Ren, W., Xiong, P., and Choo, K.-K. R. (2023). Preserving data privacy in federated learning through large gradient pruning. *Comput. Secur.* 125:103039. doi: 10.1016/j.cose.2022.103039

Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V., et al. (2018). Federated learning with non-iid data. *arXiv* [Preprint]. arXiv:1806.00582. doi: 10.48550/arXiv.1806.00582

Zhu, M., and Gupta, S. (2018). "To prune, or not to prune: exploring the efficacy of pruning for model compression," in *6th International Conference on Learning Representations, ICLR 2018*, Workshop Track Proceedings (Vancouver, BC).