Khan, Muhammad Irfan., Jafaritadi, Mojtaba., Alhoniemi, Esa., Kontio, Elina., Khan, Suleiman A. 2022. Adaptive Weight Aggregation in Federated Learning for Brain Tumor Segmentation. In: BrainLes 2021: Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. (15.7.2022).

Adaptive Weight Aggregation in Federated Learning for Brain Tumor Segmentation

Muhammad Irfan Khan[1(✉)], Mojtaba Jafaritadi[1,2], Esa Alhoniemi[1],
Elina Kontio[1], and Suleiman A. Khan[3]

[1] Turku University of Applied Sciences, 20520 Turku, Finland
{irfan.khan,mojtaba.jafaritadi,esa.alhoniemi,elina.kontio}@turkuamk.fi
[2] Stanford University, Stanford, CA 94305, USA
[3] University of Helsinki, 00100 Helsinki, Finland
suleimkh@amazon.com

Abstract. We introduce similarity weighted aggregation, a principled and efficient method for regularized weight aggregation in federated learning. Our method is adapted to non-IID collaborators and is simultaneously cost-efficient. This is the first method to propose a sliding-window to select the collaborators, to the best of our knowledge. We demonstrate our method on the federate training task of the FeTS 2021 challenge. We proposed two variations coined Similarity Weighted Aggregation (SimAgg) and Regularized Aggregation (RegAgg). SimAgg results on internal validation data demonstrate that the proposed method outperforms the baseline FedAvg. The method SimAgg by our team HT-TUAS won 2nd position on both leaderboards in FeTS2021 challenge. SimAgg is the only method to be among the top performing methods on both the leaderboards, making it robust and reliable to data variations. Our solution is open sourced at: https://github.com/dskhanirfan/FeTS2021

Keywords: Brain tumors · Cancer · Collaborative learning · Federated learning · FeTS challenge · Lesion segmentation · Weight aggregation

1 Introduction

Federated Learning (FL) can facilitate healthcare organizations to collaborate and share information without compromising patients privacy. This is in contrast to many medical imaging studies that use data stored in a centralized database, where the curation of image data, prepossessing, and model development are done with full access to the sensitive and delicate patient information. Moreover, by using secure FL infrastructures we can potentially eliminate tedious and time-consuming ethical permission process for using medical images.

A. Crimi and S. Bakas (Eds.): BrainLes 2021, LNCS 12963, pp. 455–469, 2022.
https://doi.org/10.1007/978-3-031-09002-8_40

Federated learning is a computational paradigm for distributed or decentralized machine learning where training data is shared via multiple collaborators and a central server learns a consensus model by aggregating locally-computed updates [14]. In other words, FL allows distributed adaption of AI development in a privacy-preserving fashion such that private data never leaves the local data storage (e.g. a medical device, academic research center, clinical trial site, and medical data repository). With the advent of strict regulations like GDPR (EU) and HIPAA (US) the usability spectrum of Federated Learning is diverse [1,20]. In FL, multiple collaborators – also referred to as devices or clients – contribute to a learning task. This approach allows clients to collaboratively train a shared inference model while holding all the training data on the local storage privately, decoupling the ability to do machine learning from the need to keep the data in one centralized location [18]. Hence, only certain model updates may leave the client's secure computational environment, enabling the aggregation of the learned parameters into a single generalized (global) model without disclosing the raw data to the third parties. The communication between the clients usually involves a central orchestrator that receives and aggregates client's updates [12].

Decentralized training of an inference model in a federated fashion is an iterative process, in which a subset of clients are selected to receive the current global model in each iteration. Each client runs several epochs, for example in a stochastic gradient descent optimization problem where a neural network is trained with certain mini-batches, and communicates its model update back to the server. The differences between the local models and the received global model are considered as model updates, for which the server aggregates them from the contributing clients to obtain an improved global model. This process continues to the next iteration until a desired performance is obtained [11]. Figure 1 shows a high-level schema of the federated learning framework for healthcare institutions.

In general, algorithms for FL face three main challenges: 1) statistical heterogeneity in weight aggregation, 2) communication efficiency, and 3) privacy with security [12,22]. An efficient aggregation strategy, i.e. combining the models of all clients, is essential for the successful implementation of FL in real-life applications. Numerous aggregation strategies have been studied, of which Federated Averaging (FedAvg) [14] is one of the most well-known FL methods. This approach considers the normalized number of non-Independent Identical Distribution (non-IID) data in each client to aggregate the models in the server. However, FedAvg does not address the weight divergence challenge due to the strongly skewed data distributions. FedProx [13] handles statistical heterogeneity in the network by constraining the local solvers so that they do not deviate significantly from the global model. This is achieved using a proximal weight term, however, FedProx works on the client side. Existing research on dealing with the statistical challenge of federated learning focuses on the ideas of inverse distance aggregation [23], temporal weighting [6], knowledge transfer [9], knowledge distillation and augmentation [10], multi-task learning [7], and meta-learning [5].

Communication efficiency in many FL settings is the primary bottleneck, which requires adequate cost management strategies such as decreasing the

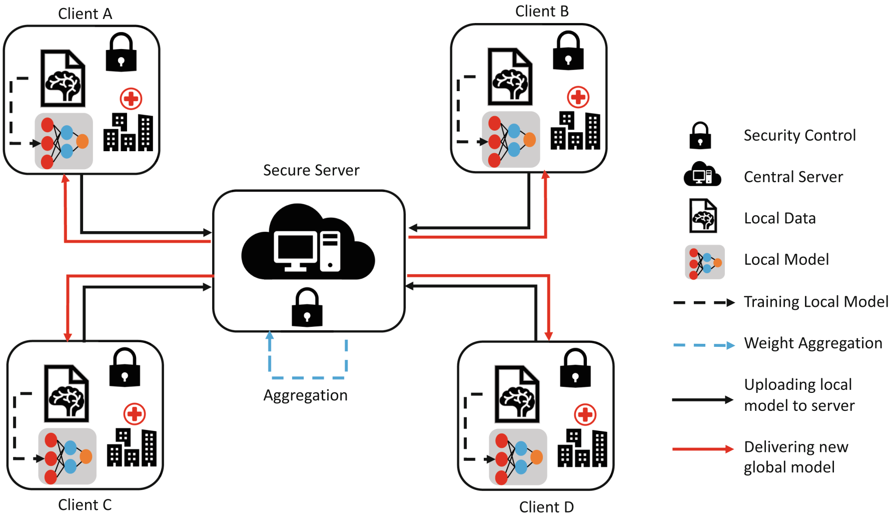


Fig. 1. General workflow of an FL-trained model and the key components in a federated learning setting [22]. Private clients A–D (e.g. healthcare institutions) communicate the local weight updates with a central secure server at regularly occurring intervals to learn a global model; the server aggregates the updates and sends back the parameters of the updated global model to the clients.

number of clients, reducing the update size, and reducing the number of updates. Hence, the existing research on communication-efficient FL is divided into four major categories: model compression, client selection, update reducing, and peer-to-peer learning [12,22].

From privacy-preserving point of view, it is also important to securely aggregate the model parameters or weights to avoid possibilities of leakage of information and vulnerability to adversarial inference and inversion [8,11,15]. Even well-generalized deep models can potentially leak a considerable amount of information about the input training data [15]. Even worse, certain neural networks trained on sensitive data (e.g., medical image data) can memorize the training data [8]. Secure aggregation protocols such as secure multiparty computation (SMC) and differential privacy (DP) have been proposed to alleviate the risk of adversarial attacks and further enhance privacy guarantees in FL [19,21]. We leave dealing with the privacy-preserving and security challenges for the future works.

Our main contributions in this paper are 1) the establishment of an efficient adaptive regularized weight aggregation approach on the FeTS 2021 multi-modal brain MRI data; 2) the implementation of a practical algorithm that can be applied to this setting; and 3) an extensive evaluation of the proposed weight aggregation approach. This paper is organized as follows: in Sect. 2, we describe the methodologies including our two FL weight aggregation strategies by our experiment setting. In Sect. 3, we describe FL experiments and evaluate the performance of the proposed methods quantitatively and in Sect. 4, we discuss about the presented work, potentials and limitations, and describe our future direction in FL. Finally, Sect. 5 concludes this work.

2 Methods

2.1 FeTS 2021 Challenge

Federated Tumor Segmentation (FeTS) Challenge 2021 focuses on federated learning in medical imaging, and intends to address efficient creation and evaluation of a consensus model for the segmentation of intrinsically heterogeneous brain tumors, namely gliomas. The FeTS 2021 challenge considers an ample multi-institutional multi-parametric Magnetic Resonance Imaging (mpMRI) scans of glioblastoma (GBM), the most common primary brain tumor, before any kind of resection surgery as the training and validation data. The datasets used in the FeTS 2021 challenge are the subset of GBM cases from the Brain Tumor Segmentation Challenge (BraTS) 2020 [2–4]. BraTS offers the largest fully annotated and publicly available database for AI model development for the objective of brain tumor segmentation methods.

The FeTS 2021 data release consists of a training set and two partitions each providing information for how to split the training data into non-IID institutional subsets[1]. The training dataset includes 341 subjects with High-Grade Gliomas (HGG) and Low-Grade Gliomas (LGG). All FeTS mpMRI scans, provided as NIfTI files (`.nii.gz`), had four $240 \times 240 \times 155$ structural MRI images including native (T1), post-contrast T1-weighted (T1Gd), T2-weighted (T2), and T2 FLuid Attenuated Inversion Recovery (FLAIR) volumes. A sample image is shown in Fig. 2. Annotations comprise the pathologically confirmed segmentation labels with similar volume size of $240 \times 240 \times 155$ including the GD-enhancing tumor (ET - label 4), the peritumoral edematous/invaded tissue (ED - label 2), and the necrotic tumor core (NCR - label 1). All these provided MRI scans were collected from multiple institutions and certain pre-processing steps such as rigid registration, brain extraction, alignment, $1 \times 1 \times 1$ mm resolution resampling, and skull stripping were applied as described in [2–4].

We deployed Intel Federated Learning (OpenFL) [17] framework for training brain tumor segmentation model—an encoder-decoder U-shape type of convolutional neural network provided by FeTS2021 challenge—using the data-private collaborative learning paradigm of FL. OpenFL considers two main components: 1) the collaborator which uses a local dataset to train the global model and 2) the aggregator which receives model updates from each collaborator and fuses them to form the global model. Our experiments were performed on a cluster workstation with running NVIDIA TITAN V100 GPU and 350 GB memory.

[1] https://github.com/FETS-AI/Challenge/tree/main/Task_1.

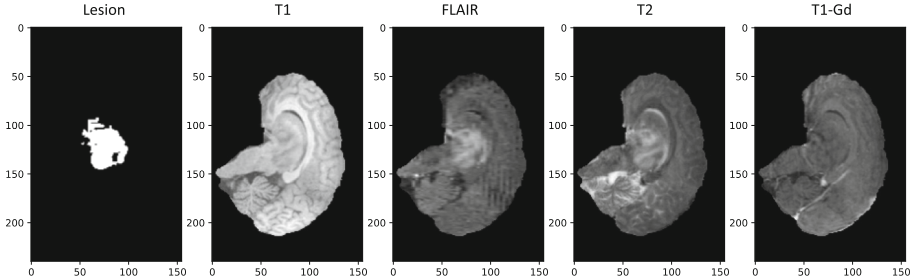


Fig. 2. Sample images from all MRI modalities with the corresponding GBM lesion.

2.2 Method 1: Similarity Weighted Aggregation (SimAgg)

We developed an adaptive machine-learning approach coined similarity weighted aggregation for efficient aggregation of model parameters at the server. Our approach is suitable for both IID as well as non-IID data. Specifically, our strategy is focused on collaborator selection and parameter aggregation policy.

Collaborator Selection. For the collaborator selection, we use a subset of the available collaborators (for example, 20%) in each round. To allow for systems heterogeneity where collaborators can contribute in a non-deterministic fashion, we simulate random selection of collaborators in each round. However, to ensure that the model sees all collaborators the same number of times at regular intervals, we implement a sliding window over the randomized collaborator index as shown in Fig. 3. In this setup, once all collaborators have participated in updates, a new randomized order is computed for better learning. We use a sliding window instead of random collaborator selection to ensure participation of all collaborators. We used a sliding-window size equal to 20% of the collaborators in each partition. In partition 1, the sliding-window size was set to three as the total number of collaborators was 17. In partition 2, the sliding-window size was set to four as the total number of collaborators was 22.

Weight Aggregation. A fundamental issue with non-IID data is that model parameters coming from the collaborators can diverge. To overcome such a scenario we use weighted aggregation of the collaborators at the server. The collaborators are weighted based on how similar they are to their non-weighted average. This simple yet effective mechanism can help in learning a master model that is representative of most of the collaborators at each round, see Algorithm 1.

Specifically, at round r, the parameters p_{C^r} of the participating collaborators C^r are collected at the server. The average of these parameters is calculated as:

$$\hat{p} = \frac{1}{|C^r|}\Sigma_{i \in C^r} p_i. \tag{1}$$

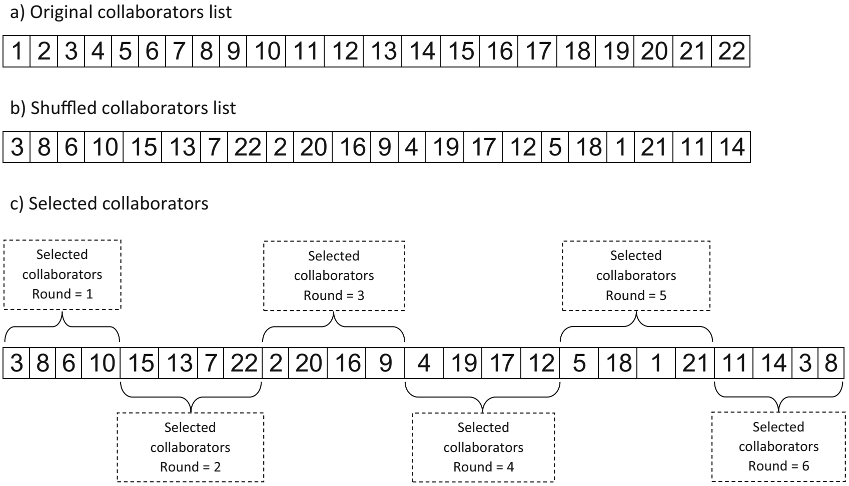

a) Original collaborators list
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
b) Shuffled collaborators list
3 8 6 10 15 13 7 22 2 20 16 9 4 19 17 12 5 18 1 21 11 14
c) Selected collaborators
Selected collaborators Round = 1
Selected collaborators Round = 3
Selected collaborators Round = 5
3 8 6 10 15 13 7 22 2 20 16 9 4 19 17 12 5 18 1 21 11 14 3 8
Selected collaborators Round = 2
Selected collaborators Round = 4
Selected collaborators Round = 6

Fig. 3. Collaborator selection strategy. In a) the model receives a list of initial collaborators, in b) collaborator order is randomized to help better learning, and in c) collaborators are selected for each round using a sliding window. Once the collaborator list is entirely used, it is shuffled again and the process starts again from step b).

We subsequently calculate the similarity of each collaborator $c \in C^r$ with the average parameter values from all collaborators as

$$sim_c = \frac{\Sigma_{i \in C^r} |p_i - \hat{p}|}{|p_c - \hat{p}| + \epsilon}, \tag{2}$$

where $\epsilon = 1e - 5$ (small positive constant), and normalize to obtain similarity weights as follows:

$$u_c = \frac{sim_c}{\Sigma_{i \in C^r} sim_i}. \tag{3}$$

The collaborators closer to the average receive a higher similarity score while those further away obtain a lower value. In the extreme case this approach can expel the diverging collaborator.

In order to adjust for the effect of varying number of samples in each collaborator $c \in C^r$, we use a second weighting factor that favors collaborators with larger sample sizes:

$$v_c = \frac{N_c}{\Sigma_{i \in C^r} N_i}, \tag{4}$$

where N_c is the number of examples in collaborator c.

Using the weights obtained using Eqs. 3 and 4, the similarity weighted parameter values (p^m) are computed as:

$$w_c = \frac{u_c + v_c}{\Sigma_{i \in C^r}(u_i + v_i)}. \tag{5}$$

Finally, the parameters are aggregated as follows:

$$p^m = \frac{1}{|C^r|} \cdot \Sigma_{i \in C^r}(w_i \cdot p_i). \tag{6}$$

The normalized aggregated parameters p^m are then dispatched to the next set of collaborators in the successive federation rounds.

Algorithm 1. SimAgg aggregation algorithm

```
procedure SIMILARITY WEIGHTED AGGREGATION($C^r$, $p_{C^r}$)
    $\epsilon \leftarrow 1e - 5$                              ▷ $C^r$ = set of collaborators (at round $r$)
    $\hat{p}$ = average($p_{C^r}$) using Eq. 1    ▷ $p_{C^r}$ = parameters of the collaborators in $C^r$
    for $c$ in $C^r$ do
        Compute similarity weights $u_c$ using Eqs. 2 and 3
        Compute sample weights $v_c$ using Eq. 4
    for $c$ in $C^r$ do
        Compute aggregation weights $w_c$ using Eq. 5
    Compute master model parameters $p^m$ using Eq. 6
    return $p^m$
```

2.3 Method 2: Regularized Aggregation (RegAgg)

We also developed a regularizing version of our aggregation approach. The method performs stronger penalization of diverging collaborators.

Collaborator Selection. The collaboration selection for regularized aggregation is the same as in Sect. 2.2.

Weight Aggregation. The weight aggregation methodology is adapted from Sect. 2.2 to compute the similarity and sample weights using Eqs. 3 and 4. We then compute the regularizing weights of each of the collaborator as:

$$w_c = \frac{u_c \cdot v_c}{\Sigma_{i \in C^r}(u_i \cdot v_i)}. \tag{7}$$

Finally, the master models parameters are computed using Eq. 6. The entire process is summarized in Algorithm 2.

Algorithm 2. RegAgg aggregation algorithm

```
procedure REGULARIZED AGGREGATION($C^r$, $p_{C^r}$)
    $\epsilon \leftarrow 1e-5$                                    ▷ $C^r$ = set of collaborators (at round $r$)
    $\hat{p}$ = average($p_{C^r}$) using **Eq. 1**   ▷ $p_{C^r}$ = parameters of the collaborators in $C^r$
    for $c$ in $C^r$ do
        Compute similarity weights $u_c$ using **Eqs. 2** and **3**
        Compute sample weights $v_c$ using **Eq. 4**
    for $c$ in $C^r$ do
        Compute aggregation weights $w_c$ using **Eq. 7**
    Compute master model parameters $p^m$ using **Eq. 6**
    return $p^m$
```

3 Experiments

3.1 Setup

The goal of task 1 is to improve the federation process by focusing on efficient aggregation, client selection, training-per-round, compression, and communication efficiency. We have developed an efficient method that aggregates the model updates trained on individual collaborators. A data set with total of 341 multi-institutional patients was available. Supplementary information indicates the division of patients in different partitions. Partition 1 and partition 2 have 17 collaborators and 22 collaborators, respectively. Partition 1 means institutional split, while partition 2 is further divided based on the tumor size. The experimental setup uses Intel's OpenFL platform for federation learning and a predefined 3D U-shape neural network for the semantic segmentation of whole tumor, tumor core, and enhancing tumor. The metrics computed in the aggregation rounds are binary DICE similarity (whole tumor, enhancing tumor, tumor core) and Hausdorff (95%) distance (whole tumor, enhancing tumor, tumor core) as described in [16].

The hyperparameters used are shown in Table 1. Collaborator selection for SimAgg and RegAgg are shown in Fig. 3.

Table 1. Hyperparameters used in aggregation algorithms.

Leaderboard	Hyperparameter	SimAgg	RegAgg	FedAvg
1	Learning rate	5e−5	5e−5	5e−5
1	Epochs per round	5.0	1.0	1.0
2	Learning rate	5e−5	5e−5	5e−5
2	Epochs per round	5.0	5.0	1.0

3.2 Results

In this section, results are summarized for leaderboards 1 and 2 (with partitions 1 & 2). The comparison of baseline FedAvg with default setting and our aggregation methods – namely regularized aggregation and similarity weighted aggregation – shows that both of our methods rapidly converge and are stable as the learning progresses across all the measured metrics. Moreover, our methods show significant improvement in the performance.

Model Training and Performance Using Internal Validation Data. Figure 4 shows the performance comparison of model training on internal validation for partition 2 for Leaderboard 1. Figure 5 contains the same comparison for both partitions 1 and 2 of Leaderboard 2.

In Leaderboard 1, SimAgg significantly outperforms RegAgg with approximately 10–15% improvement across all DICE and Hausdorff (95%) scores. In Leaderboard 2, SimAgg performs slightly better than RegAgg and FedAvg across all DICE and Hausdorff (95%) scores.

Model Performance Using External Validation Data. Prior to the official testing phase, the performance of both of our methods was assessed using unseen external validation data provided by challenge organizers, see Tables 2, 3, and 4. From the Tables 3 and 4, we can see that SimAgg resulted in higher performance across all DICE scores. Similarly, the Hausdorff (95%) distances obtained by SimAgg method were smaller than the RegAgg for the partitions 1 & 2.

Table 2. Leaderboard 1 experiments: Trained aggregation algorithms on partition 2 performance on validation data.

	SimAgg	RegAgg
Binary DICE WT	0.7774	0.6982
Binary DICE ET	0.6793	0.5856
Binary DICE TC	0.6682	0.5664
Hausdorff (95%) WT	34.2991	50.1060
Hausdorff (95%) ET	22.8250	42.5777
Hausdorff (95%) TC	29.6163	43.1602

Model Performance Using Fully Blinded Test Set. FeTS2021 challenge organizing committee permitted one algorithm per team for ranking in the official leaderboards. Therefore, we submitted SimAgg algorithm for the leaderboard ranking since SimAgg performed better on internal and external validation data in our experiments.

The SimAgg performance stats for team HT-TUAS on the fully blinded test set for Leaderboards 1 and 2 are shown in Tables 5 and 6, respectively. These

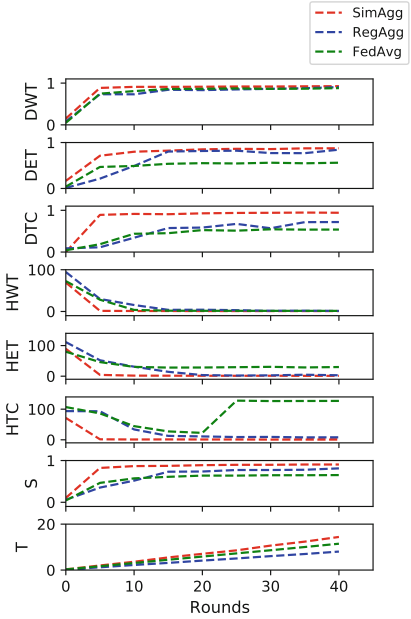
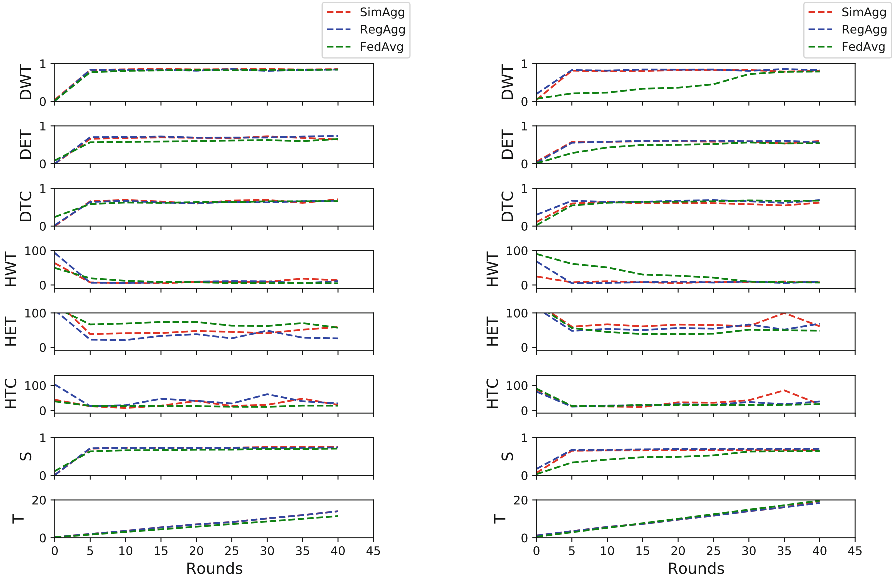


Fig. 4. Leaderboard 1 experiments: Performance metrics model training of SimAgg, RegAgg, and FedAvg for partition 2. The horizontal axis refers to the number of rounds and the vertical axis to the performance metrics. Metrics; DWT: DICE Whole Tumor, DET: DICE Enhancing Tumor, DTC: DICE Tumor Core, HWT: Hausdorff (95%) Whole Tumor, HET: Hausdorff (95%) Enhancing Tumor, HTC: Hausdorff (95%) Tumor Core, S: Projected Convergence Score, T: Simulation Time (Hours).

results ranked us as the top second team for the federated tumor segmentation challenge. In leaderboard 1, a significant discrepancy between the validation and testing datasets for the DICE and Hausdorff distance scores was visible. The discrepancy is because the wrapper function for data loader had a logical bug that the next collaborator is not selected. However, in leaderboard 2, the results on fully blind test set is better because model training is performed for 500 rounds by challenge organizers after the logical bug was removed.

Overall, SimAgg performs whole tumor segmentation better as compared to enhancing tumor segmentation and tumor core segmentation.



(a) Partition 2

(b) Partition 1

Fig. 5. Leaderboard 2 experiments: Performance metrics model training of SimAgg, RegAgg, and FedAvg for partition 2 (a) and partition 1 (b). The horizontal axis refers to the number of rounds and the vertical axis to the performance metrics. Metrics; DWT: DICE Whole Tumor, DET: DICE Enhancing Tumor, DTC: DICE Tumor Core, HWT: Hausdorff (95%) Whole Tumor, HET: Hausdorff (95%) Enhancing Tumor, HTC: Hausdorff (95%) Tumor Core, S: Projected Convergence Score, T: Simulation Time (Hours).

Table 3. Leaderboard 2 experiments: Trained aggregation algorithms on partition 2 performance on validation data.

	SimAgg	RegAgg
Binary DICE WT	0.8415	0.8387
Binary DICE ET	0.6993	0.6910
Binary DICE TC	0.7143	0.7110
Hausdorff (95%) WT	12.1612	12.8851
Hausdorff (95%) ET	17.2475	26.5882
Hausdorff (95%) TC	17.6554	26.3145

4 Discussion

Various methods have been proposed in the literature for federated aggregation. However, a limited set of methods work exclusively on the server side. To start with, we explored several alternatives including exponential smoothing aggre-

Table 4. Leaderboard 2 experiments: Trained aggregation algorithms on partition 1 performance on validation data.

	SimAgg	RegAgg
Binary DICE WT	0.8501	0.8265
Binary DICE ET	0.7087	0.6867
Binary DICE TC	0.7038	0.7160
Hausdorff (95%) WT	13.2122	14.1265
Hausdorff (95%) ET	15.2001	16.9239
Hausdorff (95%) TC	16.3441	18.1132

Table 5. SimAgg (HT-TUAS) test set performance on Leaderboard 1.

	Mean	Standard Deviation	Median	25quantile	75quantile
DICE WT	0.7076	0.2676	0.8259	0.5788	0.9066
DICE ET	0.6054	0.3172	0.7558	0.3633	0.8461
DICE TC	0.6502	0.3320	0.8144	0.4134	0.9082
Sensitivity ET	0.7845	0.2908	0.8967	0.7565	0.9542
Sensitivity WT	0.8471	0.1722	0.8997	0.8231	0.9410
Sensitivity TC	0.8104	0.2913	0.9328	0.8259	0.9729
Specificity WT	0.9942	0.0081	0.9985	0.9909	0.9994
Specificity ET	0.9973	0.0045	0.9993	0.9971	0.9997
Specificity TC	0.9964	0.0062	0.9993	0.9957	0.9998
Hausdorff (95%) WT	30.5343	29.3950	13.8515	4.3872	58.5597
Hausdorff (95%) ET	53.9195	98.8776	5.5649	1.4142	71.2686
Hausdorff (95%) TC	48.6906	80.4691	16.8320	3.0000	68.4579
Communication Cost	0.8562	0.8562	0.8562	0.8562	0.8562

gation and conditional threshold aggregation. However, both of these methods required user defined threshold parameters that needed tuning, hence, these approaches are not inherently generalizable to new and unseen data sets. Therefore, we designed similarity weighted aggregation and regularized aggregation that automatically adapt the weights. Unlike, our approach, FedProx [13] performs regularized weight aggregation on the client side by restricting the local solvers so that they do not deviate significantly from the global model. Our method works on the server-side by limiting the contribution of the diverging collaborators to learn the global model. Our approach has the additional advantage that it can be implemented only on the server-side so that clients with varying configurations can join the federation.

Several works have demonstrated that using a subset of random collaborators helps speed up the training of federated learning algorithms [24]. We extended

Table 6. SimAgg (HT-TUAS) test set performance on Leaderboard 2.

	Mean	Standard deviation	Median	25quantile	75quantile
DICE WT	0.8213	0.1797	0.8847	0.8055	0.9188
DICE ET	0.7438	0.2425	0.8174	0.7179	0.8868
DICE TC	0.7455	0.2662	0.859	0.6780	0.9119
Sensitivity ET	0.8423	0.2597	0.9427	0.8563	0.9820
Sensitivity WT	0.9070	0.1731	0.9619	0.9190	0.9866
Sensitivity TC	0.8510	0.2685	0.9607	0.8735	0.9881
Specificity WT	0.9979	0.0025	0.9984	0.9975	0.9991
Specificity ET	0.9993	0.0011	0.9995	0.9992	0.9998
Specificity TC	0.9988	0.0019	0.9994	0.9986	0.9998
Hausdorff (95%) WT	8.2904	10.7090	5.0990	3.0000	9.0415
Hausdorff (95%) ET	26.4082	88.3786	2.2361	1.4142	3.6056
Hausdorff (95%) TC	26.2290	74.0068	6.7082	2.4495	16.9027
Communication Cost	0.7937	0.7937	0.7937	0.7937	0.7937

the ideas here and formulated a sliding window strategy that ensures representation of all collaborators in the training process. It may be valuable to study the performance of sliding window alone without SimAgg or RegAgg aggregation in future. We used a sliding window size equal to 20% of the collaborators. The size of the sliding window is a hyper-parameter of the method and optimizing it will only further improve the model performance. A promising future work is to develop a strategy for optimizing the sliding-window size.

The FeTS 2021 data release consists of two partitions each providing information for how the training data is split into non-IID institutional and tumor size subsets. Therefore, the size and distribution of data in each collaborator can be different. Our method works well on both partitions, as the weighted aggregation approach helps learn a model that is representative of most of the collaborators at each round, with minimal impact from the outliers. While the model performs well in general when data has non-IID splits, it will be valuable to further investigate the performance on the outliers.

A limitation of this work is the small number of patients and collaborators. However, our approach has laid the groundwork for the refined development of an improved model that can be applied to newly generated data sets at scale. The aggregation algorithm can be used for generalizable ML model training for "real-world" clinical data in clinical practices and production environments on geographically distinct collaborators.

Our future research direction includes incorporation of our developed FL methods with diverse state-of-the-art privacy protection AI frameworks for data anonymization, augmentation, object detection and segmentation, and image translation. The widespread adoption of secure and private AI on medical image data still requires vigorous improvements to the generalization or personalization of the AI models. Decentralized data storage, efficient cryptographic and privacy primitives, and dedicated neural network operations are yet emerging to replace the current paradigm of data sharing and privacy preservation, enabling privacy-preserving cross-institutional research in a breadth of biomedical disciplines.

5 Conclusion

In this work, we proposed two novel weight aggregation schemes, regularized aggregation and similarity weighted aggregation, for aggregation of neural network models in a federated learning setting for brain tumor segmentation. Our extensive experiments on internal validation show that the proposed methods outperform FedAvg in terms of convergence score and communication costs. Our team HT-TUAS submitted SimAgg for ranking on official leaderboards and won 2$^{\text{nd}}$ position on both Leaderboards in FeTS2021 challenge. While our proposed strategies offer better aggregation benefits, providing stronger privacy guarantees, for example via differential privacy, secure multi-party computation, or a mixture of them is an interesting future research direction.

Acknowledgements. This work was supported by the Business Finland under Grant 33961/31/2020. We also acknowledge the support and computational resources facilitated by the CSC-Puhti super-computer, a non-profit state enterprise owned by the Finnish state and higher education institutions in Finland.

References

1. Annas, G.J.: HIPAA regulations-a new era of medical-record privacy? (2003)
2. Bakas, S., et al.: Segmentation labels and radiomic features for the pre-operative scans of the TCGA-GBM collection. The cancer imaging archive. Nat. Sci. Data **4**, 170117 (2017)
3. Bakas, S., et al.: Segmentation labels and radiomic features for the pre-operative scans of the TCGA-LGG collection. Cancer Imaging Archive **286** (2017)
4. Bakas, S., et al.: Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features. Sci. Data **4**(1), 1–13 (2017)
5. Beel, J.: Federated meta-learning: democratizing algorithm selection across disciplines and software libraries. Science (AICS) **210**, 219 (2018)
6. Chen, Y., Sun, X., Jin, Y.: Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation. IEEE Trans. Neural Netw. Learn. Syst. **31**(10), 4229–4238 (2019)
7. Corinzia, L., Beuret, A., Buhmann, J.M.: Variational federated multi-task learning. arXiv preprint arXiv:1906.06268 (2019)
8. Fung, C., Yoon, C.J., Beschastnikh, I.: Mitigating sybils in federated learning poisoning. arXiv preprint arXiv:1808.04866 (2018)
9. He, C., Annavaram, M., Avestimehr, S.: Group knowledge transfer: federated learning of large CNNs at the edge. arXiv preprint arXiv:2007.14513 (2020)

10. Jeong, E., Oh, S., Kim, H., Park, J., Bennis, M., Kim, S.L.: Communication-efficient on-device machine learning: Federated distillation and augmentation under non-IID private data. arXiv preprint arXiv:1811.11479 (2018)
11. Kadhe, S., Rajaraman, N., Koyluoglu, O.O., Ramchandran, K.: FastSecAgg: scalable secure aggregation for privacy-preserving federated learning. arXiv preprint arXiv:2009.11248 (2020)
12. Kairouz, P., et al.: Advances and open problems in federated learning (2019). https://arxiv.org/abs/1912.04977
13. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. arXiv preprint arXiv:1812.06127 (2018)
14. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282. PMLR (2017)
15. Nasr, M., Shokri, R., Houmansadr, A.: Comprehensive privacy analysis of deep learning: passive and active white-box inference attacks against centralized and federated learning. In: 2019 IEEE Symposium on Security and Privacy (SP), pp. 739–753. IEEE (2019)
16. Pati, S., et al.: The federated tumor segmentation (FETS) challenge. arXiv preprint arXiv:2105.05874 (2021)
17. Reina, G.A., et al.: OpenFL: an open-source framework for federated learning. arXiv preprint arXiv:2105.06413 (2021)
18. Sadilek, A., et al.: Privacy-first health research with federated learning. medRxiv (2020)
19. Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R.: A hybrid approach to privacy-preserving federated learning (2018)
20. Voigt, P., Von dem Bussche, A.: The EU General Data Protection Regulation (GDPR). A Practical Guide, 1st edn. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-57959-7
21. Wei, K., et al.: Federated learning with differential privacy: algorithms and performance analysis. IEEE Trans. Inf. Forensics Secur. **15**, 3454–3469 (2020)
22. Xu, J., Glicksberg, B.S., Su, C., Walker, P., Bian, J., Wang, F.: Federated learning for healthcare informatics. J. Healthc. Inform. Res. **5**(1), 1–19 (2021)
23. Yeganeh, Y., Farshad, A., Navab, N., Albarqouni, S.: Inverse distance aggregation for federated learning with non-IID data. In: Albarqouni, S., et al. (eds.) DART/DCL -2020. LNCS, vol. 12444, pp. 150–159. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60548-3_15
24. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-IID data. arXiv preprint arXiv:1806.00582 (2018)