ICS FRAMEWORK: EMPIRICAL EVALUATION OF FEDERATED LEARNING STRATEGIES FOR BRAIN DISEASE DIAGNOSIS

Vișan Ionuț Poață Andrei-Cătălin Vulpe Ștefan

Faculty of Automatics and Computer Science,
National University of Science and Technology Politehnica Bucharest

ABSTRACT

Federated Learning (FL) enables collaborative model training across medical institutions while preserving data privacy, but its effectiveness in real-world clinical settings is often limited by data heterogeneity, demographic imbalance, and communication constraints.

We present an empirical evaluation of federated learning strategies for medical neuroimaging analysis under realistic non-IID conditions. Using a multi-architecture framework for MRI-based brain tumor segmentation, we analyze the impact of aggregation methods, training schedules, client selection, data balancing, and communication-efficient techniques. Our results show that combining appropriate federated configurations with inference-time ensemble modeling improves robustness, fairness, and performance while maintaining strong privacy guarantees.

***K*eywords** Federated Learning · Medical Imaging · Neuroimaging Analysis · Brain Tumor Segmentation · Data Heterogeneity · Privacy-Preserving Machine Learning

1 Introduction

Advances in deep learning have significantly expanded the capabilities of medical imaging systems, enabling increasingly accurate models for complex tasks such as disease diagnosis and image segmentation. At the same time, modern clinical applications demand models that are not only accurate, but also robust across diverse patient populations, imaging protocols, and acquisition settings. As model architectures grow more complex and data distributions become increasingly heterogeneous, reliance on a single model or a single training environment often proves insufficient.

To address these challenges, recent research has moved toward learning paradigms that emphasize scalability, robustness, and diversity, including the use of multiple complementary model architectures and collaborative training strategies. Such approaches aim to capture heterogeneous patterns present across datasets and institutions, while mitigating overfitting and improving generalization in real-world clinical scenarios.

Prior work demonstrates that federated learning is highly affected by data heterogeneity and client drift, particularly in medical imaging scenarios with non-IID data. While Federated Averaging (FedAvg) provides a scalable baseline [1], its convergence degrades under heterogeneous conditions. FedProx mitigates this effect by constraining local updates via proximal regularization [2], whereas FedNova improves stability by normalizing client updates across uneven training workloads [3]. In parallel, recent studies show that ensemble strategies within federated learning can further enhance robustness and generalization by leveraging complementary model representations [4]. These insights motivate a systematic evaluation of federated learning strategies and ensemble-based approaches in realistic clinical settings.

This work focuses on federated learning for medical image analysis, where heterogeneous client data and diverse model architectures jointly induce a complex and structured optimization space. Model performance depends on multiple interacting factors, including aggregation strategies, local training schedules, client participation, and architectural design choices. These dependencies are further amplified by stochastic training dynamics and non-IID data distributions across clinical sites, leading to highly non-convex and noisy optimization landscapes that limit the effectiveness of naive or single-configuration approaches.

Given these challenges, exploring diverse federated learning strategies and ensemble-based modeling naturally emerges as an effective solution. Federated optimization methods such as alternative aggregation schemes and training configurations aim to stabilize convergence under heterogeneity, while ensemble approaches leverage complementary representations across architectures and clients to improve robustness and generalization. Building on these principles, this work systematically evaluates a range of federated learning strategies and introduces an inference-time ensemble framework that exploits architectural diversity, enabling improved performance and stability in realistic clinical federated learning scenarios.

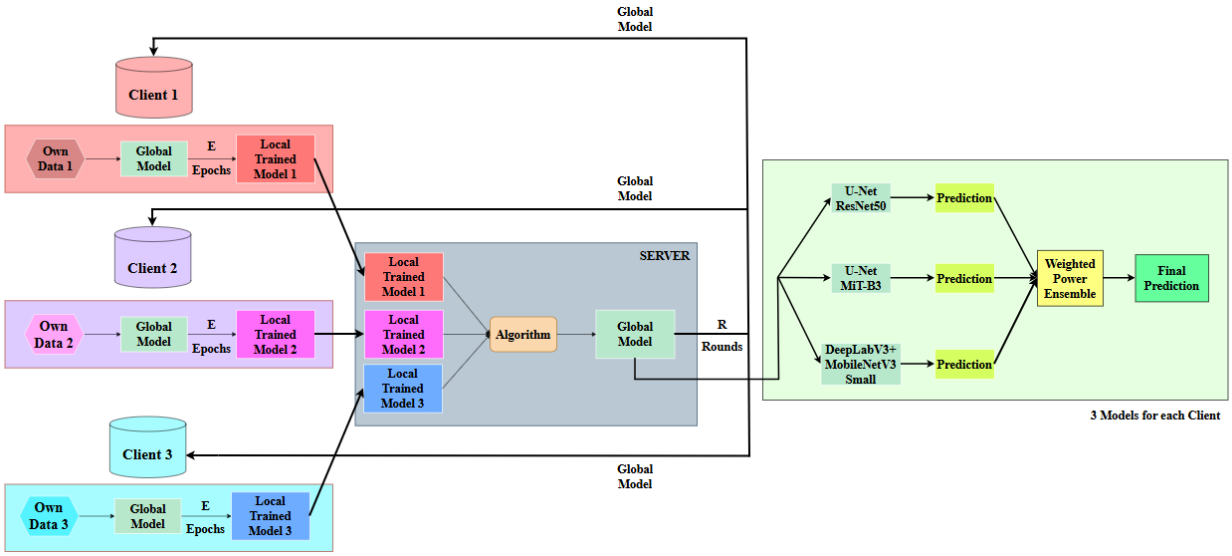

Figure 1: Overview of the proposed federated learning framework. Multiple clients locally train heterogeneous model architectures using private data and periodically transmit model updates to a central server for aggregation. The resulting global models are combined at inference time using a weighted ensemble strategy to produce the final prediction.

2 Method

This section presents the proposed federated learning framework, outlining the system architecture, the client–server training protocol, and the optimization objectives used in our experiments.

2.1 Overview

The proposed framework follows a standard federated learning paradigm in which multiple decentralized clients collaboratively train shared models under the coordination of a central server, without exchanging raw data. As illustrated in Figure 1, the system adopts a client–server architecture where each client $k \in \mathcal{C} = \{1, \ldots, K\}$ holds a private dataset $\mathcal{D}_k$. Training is performed over R communication rounds, during which the server broadcasts the global model parameters $\theta^{(r)}$ to all clients.

Upon receiving the global model, each client performs local optimization for E epochs using only its private data:

$$\theta_k^{(r)} = \arg\min_{\theta} \; \mathcal{L}(\theta; \mathcal{D}_k), \tag{1}$$

where $\mathcal{L}$ denotes the segmentation loss. The updated parameters are then transmitted back to the server and aggregated to form the next global model,

$$\theta^{(r+1)} = \mathcal{A}\big(\{\theta_k^{(r)}\}_{k \in \mathcal{C}}\big), \tag{2}$$

with $\mathcal{A}(\cdot)$ denoting a federated aggregation operator such as Federated Averaging. This iterative process enables the global model to progressively integrate knowledge from all clients while preserving data privacy.

To improve robustness under client-level data heterogeneity, each client independently trains an ensemble of M segmentation models with different architectural backbones, as shown in Figure 1 (right). Convolutional and transformer-based architectures are optimized separately under the same federated protocol, and aggregation is performed independently for each architecture, yielding a set of global models $\{\theta_m^{(R)}\}_{m=1}^{M}$ after the final round [5].

Model combination is applied exclusively at inference time. For an input sample x, each global model produces a logit prediction $f_m(x; \theta_m^{(R)})$, and the final output is obtained through a weighted ensemble,

$$\hat{y}(x) = \sum_{m=1}^{M} w_m \, f_m(x; \theta_m^{(R)}), \tag{3}$$

where the weights $\{w_m\}$ are non-negative, normalized, and derived from client-specific validation performance using a power-based scheme,

$$w_m = \frac{(\text{Dice}_m)^{\gamma}}{\sum_{j=1}^{M} (\text{Dice}_j)^{\gamma}}, \quad \gamma > 1. \tag{4}$$

This strategy emphasizes stronger-performing models while still leveraging architectural diversity.

In preliminary experiments, we also explored a CLIP-inspired multimodal formulation aligning visual MRI features with auxiliary clinical embeddings, following recent advances in multimodal medical representation learning [6]. However, this approach showed higher instability and inferior performance compared to the ensemble-based strategy. Consequently, the remainder of this work focuses on the ensemble-based federated learning framework.

2.2 Client-Side Training and Local Model Updates

At each client, local training is performed exclusively on private data, following the federated learning protocol described in the previous section. At the initial federated round ($r = 0$), the global model parameters are randomly initialized and broadcast to all clients. In subsequent rounds, each client receives the aggregated global parameters obtained from the previous round, ensuring a consistent initialization point for local optimization across clients and rounds:

$$\theta_{k,m}^{(r,0)} \leftarrow \theta_m^{(r)}, \quad \forall k \in \mathcal{C},\ m \in \{1, \dots, M\}, \tag{5}$$

where $\theta_{k,m}^{(r,0)}$ denotes the initial local parameters for client k and architecture m at round r.

For each federated round, every client initializes a local copy of the global model independently for each architecture considered in the framework. Local optimization is then carried out for a fixed number of local epochs E using only the client's private dataset $\mathcal{D}_k$, following standard practices in federated learning [7]. Training follows a standard mini-batch stochastic optimization procedure:

$$\theta_{k,m}^{(r,e+1)} = \theta_{k,m}^{(r,e)} - \eta \nabla \mathcal{L}\big(\theta_{k,m}^{(r,e)}; \mathcal{B}_k^{(e)}\big), \quad e = 0, \dots, E-1, \tag{6}$$

where η denotes the learning rate and $\mathcal{B}_k^{(e)} \subset \mathcal{D}_k$ represents a mini-batch sampled from the local dataset at epoch e.

Local training is performed independently for each model architecture. No parameter sharing, interaction, or knowledge transfer occurs between architectures during local optimization. This design ensures that all updates remain architecture-specific and can be aggregated consistently across clients at the server side:

$$\theta_{k,m}^{(r,E)} \perp \theta_{k,m'}^{(r,E)}, \quad \forall m \neq m'. \tag{7}$$

The selected architectures are designed to provide complementary inductive biases. Specifically, U-Net models with ResNet-50 and MiT-B3 encoders are employed to capture fine-grained spatial features and long-range contextual dependencies, respectively, while a DeepLabV3+ model with a MobileNetV3-Small encoder emphasizes multi-scale context aggregation with reduced computational overhead [8]. By training these architectures independently at each client, architectural diversity is preserved without increasing the complexity of the local optimization process.

During local training, each client may perform validation on a held-out subset of its private data. Validation is used solely for monitoring convergence and selecting the best-performing local checkpoint for each architecture:

$$\theta_{k,m}^{(r,\star)} = \arg\max_{\theta_{k,m}^{(r,e)}} \text{Dice}\big(\theta_{k,m}^{(r,e)}; \mathcal{D}_k^{\text{val}}\big), \tag{8}$$

where $\mathcal{D}_k^{\text{val}}$ denotes the local validation set. Validation metrics are not shared with the server and remain strictly local.

Upon completion of local training, each client transmits only the selected model parameters for each architecture to the central server:

$$\{\theta_{k,m}^{(r,\star)}\}_{m=1}^{M} \rightarrow \text{Server}, \tag{9}$$

while all raw data, intermediate activations, gradients, and validation statistics remain local. This communication protocol ensures privacy preservation throughout the federated training process.

2.3 Server-Side Aggregation and Federated Optimization

The central server coordinates the federated learning process by aggregating the locally updated model parameters received from all participating clients after each communication round. Following the client-side training procedure described in Section 2.2, the server collects one set of model parameters per client and per model architecture, while having no access to any client data and operating exclusively on transmitted parameters:

$$\{\theta_{k,m}^{(r,\star)}\}_{k \in \mathcal{C},\, m \in \{1,\dots,M\}} \rightarrow \text{Server}. \tag{10}$$

Aggregation is performed independently for each model architecture. For a given architecture m, the server constructs an updated global model by combining the locally optimized parameters received from all clients:

$$\theta_m^{(r+1)} = \mathcal{A}_m\big(\{\theta_{k,m}^{(r,\star)}\}_{k \in \mathcal{C}}\big), \tag{11}$$

where $\mathcal{A}_m(\cdot)$ denotes an architecture-specific federated aggregation operator. Treating each architecture separately preserves architectural diversity throughout federated training and avoids interference between heterogeneous model families.

To aggregate client updates, the server applies a federated optimization strategy that combines the locally updated parameters into a new global model. In the general case, the contribution of each client can be weighted according to client-specific characteristics, such as local dataset size:

$$\theta_m^{(r+1)} = \sum_{k \in \mathcal{C}} \alpha_k\, \theta_{k,m}^{(r,\star)}, \quad \text{with} \quad \alpha_k = \frac{|\mathcal{D}_k|}{\sum_{j \in \mathcal{C}} |\mathcal{D}_j|}. \tag{12}$$

This formulation encompasses standard Federated Averaging as well as alternative aggregation strategies, enabling a systematic comparison of server-side optimization methods under heterogeneous data distributions.

Once aggregation is completed, the server broadcasts the updated global model parameters for each architecture back to all clients:

$$\theta_m^{(r+1)} \rightarrow \{\text{Client } k\}_{k \in \mathcal{C}}, \tag{13}$$

which serve as the initialization for the subsequent round of local training. This iterative process is repeated for a total of R federated rounds, progressively refining the shared models across communication rounds.

Throughout the federated optimization process, the server acts solely as a coordination and aggregation entity. No raw data, intermediate feature representations, gradients, or client-side validation statistics are transmitted or stored at the server:

$$\text{Server} \cap \{\mathcal{D}_k, \nabla\mathcal{L}_k, \text{metrics}_k\} = \emptyset \tag{14}$$

ensuring that data privacy is preserved by design.

2.4 Inference-Time Ensemble Fusion and Prediction

After completion of the federated optimization process described in Section 2.3, each model architecture converges to a final global model obtained after R federated rounds. These models remain fixed during inference, and no further parameter updates or server–client communication are performed:

$$\theta_m^\star = \theta_m^{(R)}, \quad m = 1, \ldots, M. \tag{15}$$

In contrast to the training and aggregation stages, where each architecture is optimized independently, model combination is introduced exclusively at inference time. Given an input sample x, each global model produces an independent logit prediction:

$$z_m(x) = f_m(x; \theta_m^\star), \quad m = 1, \ldots, M. \tag{16}$$

This design preserves the modularity of federated training while enabling the exploitation of complementary representations learned by different architectures.

The final prediction is obtained by fusing the individual model outputs using a weighted ensemble strategy [9]:

$$z_{\text{ens}}(x) = \sum_{m=1}^{M} w_m \, z_m(x), \tag{17}$$

where $w_m \geq 0$ denotes the weight assigned to model m, and the weights are normalized such that $\sum_{m=1}^{M} w_m = 1$. The ensemble weights are determined based on post-training validation performance, allowing stronger-performing models to contribute more substantially to the final prediction while still benefiting from ensemble diversity.

The ensemble logit output is converted into a binary segmentation mask by applying a sigmoid activation followed by thresholding:

$$\hat{y}(x) = \mathbb{I}\big(\sigma(z_{\text{ens}}(x)) \geq \tau\big), \tag{18}$$

where $\sigma(\cdot)$ denotes the sigmoid function, $\mathbb{I}(\cdot)$ is the indicator function, and τ is a fixed decision threshold set to 0.5 in all experiments. This thresholding step controls the precision–recall trade-off and ensures a consistent binarization rule across clients and architectures.

By restricting ensemble fusion to inference time, the proposed framework avoids additional coupling during federated training and remains fully compatible with the aggregation strategies described in Section 2.3. The resulting ensemble prediction serves as the final output for evaluation and downstream analysis, leveraging architectural diversity to improve robustness and segmentation stability under client-level and data-level heterogeneity [10].

2.5 Training Objective and Loss Function

The training objective of the proposed framework is to learn segmentation models that accurately delineate tumor regions while remaining robust to client-level and data-level heterogeneity inherent in federated learning. Model optimization is performed locally at each client, while evaluation and model selection rely on segmentation metrics computed on held-out validation data. This separation ensures that local training objectives remain aligned with the global goal of reliable tumor segmentation.

Local training is formulated as a supervised segmentation task and optimized using a Dice-based loss function, which directly maximizes spatial overlap between predicted and ground-truth masks [11]. This choice is particularly suitable for medical image segmentation, where target regions often occupy a small fraction of the image and class imbalance is common. For a given client k and architecture m, the Dice loss is defined as:

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2\langle \hat{y}, y \rangle}{\|\hat{y}\|_1 + \|y\|_1 + \epsilon}, \tag{19}$$

where $\hat{y}$ and y denote the predicted and ground-truth masks, respectively, and ϵ is a small constant added for numerical stability. The loss is computed independently for each client and architecture, and no loss values or gradients are shared with the server.

Model performance is evaluated using standard segmentation metrics capturing complementary aspects of prediction quality. Specifically, we report the Dice coefficient, Intersection over Union (IoU), and pixel-wise accuracy, which together provide a comprehensive assessment of segmentation quality [12]:

$$\text{Dice} = \frac{2|\hat{Y} \cap Y|}{|\hat{Y}| + |Y|}, \quad \text{IoU} = \frac{|\hat{Y} \cap Y|}{|\hat{Y} \cup Y|}, \quad \text{Acc} = \frac{|\hat{Y} = Y|}{|Y|}. \tag{20}$$

During local training, each client maintains a validation split of its private dataset. Validation is performed at the end of each local epoch to monitor convergence and prevent overfitting, and a checkpoint is considered improved if the validation Dice score is non-decreasing while the validation loss is non-increasing. The best-performing checkpoint for each architecture is selected accordingly:

$$\text{Dice}_{k,m}^{(r,e)} > \text{Dice}_{k,m}^{(r,e-1)} \quad \wedge \quad \mathcal{L}_{k,m}^{(r,e)} < \mathcal{L}_{k,m}^{(r,e-1)}. \tag{21}$$

Only the parameters corresponding to the selected checkpoint are transmitted to the server at the end of each federated round, ensuring that aggregation operates on the most reliable local updates.

By combining a Dice-based training objective with robust evaluation metrics and validation-driven model selection, the proposed framework enables stable and effective optimization in a federated setting, while preserving privacy, scalability, and robustness across heterogeneous clients.

Figure 2: Example data sample from the MU-Glioma-Post dataset after preprocessing. From left to right: normalized post-contrast T1-weighted (T1c) MRI slice, corresponding binary tumor segmentation mask, and the affinely registered Harvard–Oxford brain atlas slice providing anatomical context.

3 Experimental Setup

3.1 Dataset

We conduct our experiments on the MU-Glioma-Post dataset, a publicly available TCIA collection comprising 203 post-operative glioma patients [13]. For each patient, the dataset provides post-contrast T1-weighted (T1c) MRI volumes, expert-annotated tumor segmentation masks, and rich clinical metadata, including demographic information, histological diagnosis, and several molecular markers. The cohort is dominated by adult patients with high-grade gliomas, particularly glioblastoma, and exhibits notable demographic and clinical imbalances, making it a realistic and challenging benchmark for robustness analysis in federated learning.

A standardized preprocessing pipeline is applied to all MRI volumes to ensure consistent model inputs. As illustrated in Figure 2, each 3D T1c scan is affinely registered to a combined Harvard–Oxford cortical and subcortical brain atlas using ANTs [14]. To reduce computational complexity while preserving salient pathology, we extract the single axial slice with the largest tumor area based on the ground-truth mask. The resulting MRI slice, corresponding tumor mask, and aligned atlas map are then min–max normalized to the range $[0, 1]$. In our experiments, models receive a two-channel input consisting of the T1c slice and the atlas regions, while the tumor mask serves as the segmentation target.

To simulate a realistic federated learning environment, the dataset is partitioned into N clients using a non-IID splitting strategy. Specifically, we employ a Dirichlet-based allocation with concentration parameter $\alpha = 0.3$, inducing significant heterogeneity across clients. Patient samples are first stratified by tumor size, defined as the tumor area in the selected axial slice, and then distributed across clients according to the Dirichlet proportions. Within each client, the assigned data is further split into training (80%) and validation (20%) sets using a fixed random seed to ensure reproducibility. This setup enables controlled evaluation of federated learning behavior under pronounced data imbalance and client-level heterogeneity.

In most experiments we use $N = 3$ clients; following the Dirichlet non-IID splitting strategy described above, the resulting client-wise distributions are summarized in Table 1.

Table 1: Client-wise data distribution for the $N = 3$ non-IID Dirichlet split ($\alpha = 0.3$).

Client	Split	n	Bin 0	Bin 1	Bin 2
0	Train	35	2	32	1
0	Val	9	1	8	0
1	Train	96	27	20	49
1	Val	25	4	4	17
2	Train	29	26	3	0
2	Val	8	8	0	0

3.2 Implementation Details

All experiments are conducted within a fixed and controlled implementation framework to ensure fair and reproducible comparisons. Model architectures, data preprocessing steps, local training procedures, optimization settings, and inference-time ensemble mechanisms are kept constant across all experiments.

Experimental evaluation varies key federated learning components, including server-side aggregation strategies, communication frequency, and the trade-off between federated rounds and local epochs. We further examine ensemble weighting parameters, client selection and participation schemes, as well as synchronous versus asynchronous training protocols, in order to assess their impact on convergence behavior, robustness, and segmentation performance under heterogeneous and non-IID data distributions.

4 Experiments and Analysis

This section evaluates the proposed framework by comparing centralized and federated training and analyzing the impact of key federated learning design choices, including aggregation strategies, communication efficiency, and deployment settings.

4.1 Centralized Training

As a reference baseline, we evaluate centralized training, where each model is optimized end-to-end using a single, unified training process without client partitioning, server coordination, or communication rounds. In this setting, all available training data are pooled together and training proceeds in a standard supervised manner, with checkpointing performed based on validation improvements.

Table 2: Centralized training performance across model architectures and client-specific validation splits.

Model	Metric	C0	C1	C2
U-Net (ResNet-50)	Train Loss	**0.1445**	**0.1181**	0.6088
	Val Loss	0.3632	0.2982	0.8064
	Val Dice	0.6547	0.7052	0.3153
	Val IoU	0.5051	0.5477	0.1872
	Val Acc	0.9737	0.9725	0.9689
	Epoch	29/50	24/50	16/50
U-Net (MiT-B3)	Train Loss	0.4272	0.1296	**0.4897**
	Val Loss	0.4724	**0.2633**	**0.7861**
	Val Dice	0.6433	**0.7412**	**0.3692**
	Val IoU	0.5019	**0.5905**	**0.2264**
	Val Acc	0.9701	**0.9769**	**0.9846**
	Epoch	16/50	21/50	20/50
DeepLabV3+ (MobileNetV3-S)	Train Loss	0.2387	0.2084	0.6533
	Val Loss	**0.3554**	0.3441	0.8813
	Val Dice	**0.6801**	0.6628	0.2655
	Val IoU	**0.5246**	0.5021	0.1531
	Val Acc	**0.9782**	0.9655	0.9747
	Epoch	12/50	8/50	9/50

The centralized results in Table 2 highlight noticeable variability across model architectures and client-specific validation splits, suggesting that performance is sensitive to distributional differences and imbalance across data subsets. While certain architectures achieve strong Dice/IoU on particular clients, no single model consistently dominates across all clients, indicating that architectural inductive biases interact differently with heterogeneous clinical data.

These findings, evidenced by the cross-client variation in Table 2, motivate the use of complementary architectures and ensemble-based inference to improve robustness and reduce sensitivity to client-level shifts. They also reinforce the need for federated learning, where knowledge from multiple clients can be integrated through collaborative optimization without direct data sharing, enabling more stable performance in realistic multi-institutional settings.

4.2 Baseline Federated Training

Building on the centralized baselines, we evaluate the framework described in Section 2 in a federated learning setting that reflects realistic multi-client collaboration scenarios. Training is conducted over $R = 5$ federated rounds, during which the server broadcasts the current global model parameters to all participating clients at each round, ensuring synchronized model initialization across sites.

Each client performs $E = 5$ local training epochs using only its private data and then returns the updated model parameters to the server. Server-side aggregation is performed using the Federated Averaging (FedAvg) algorithm, with aggregation applied independently for each model architecture. All federated experiments use a fixed learning rate of 10^{-3} and involve three clients, enabling direct comparison with centralized training while preserving data privacy and respecting communication constraints.

Table 3: Baseline federated training performance (FedAvg) across model architectures and client-specific validation splits.

Model	Metric	C0	C1	C2
U-Net (ResNet-50)	Train Loss	0.1817	0.1385	0.1761
	Val Loss	0.2735	0.2568	0.5232
	Val Dice	0.5696	0.6353	0.2183
	Val IoU	0.4362	0.4685	0.1225
	Val Acc	0.9756	0.9662	0.9320
	Epoch–Round	2–5	2–5	5–5
U-Net (MiT-B3)	Train Loss	0.2092	**0.1068**	0.1838
	Val Loss	0.2191	**0.1776**	**0.2854**
	Val Dice	**0.6634**	**0.7493**	**0.5281**
	Val IoU	0.5261	**0.6033**	**0.3588**
	Val Acc	**0.9808**	**0.9767**	**0.9882**
	Epoch–Round	2–5	5–5	3–5
DeepLabV3+ (MobileNetV3-S)	Train Loss	**0.1514**	0.1291	**0.1480**
	Val Loss	**0.2156**	0.2571	0.4041
	Val Dice	0.6614	0.6409	0.2828
	Val IoU	**0.5372**	0.4751	0.1647
	Val Acc	0.9804	0.9616	0.9843
	Epoch–Round	1–5	2–5	5–5

As shown in Table 3, federated training achieves validation performance largely comparable to centralized training across all evaluated architectures and clients. Although minor degradations are observed for certain metrics, several model–client combinations demonstrate stable or improved Dice and IoU scores, indicating that federated optimization can effectively integrate knowledge from distributed and heterogeneous clinical data sources.

In addition to segmentation performance, we monitor the execution time of the federated learning experiments in order to quantify their computational cost and training efficiency across different model architectures.

Table 4: Execution time of baseline federated training per model architecture.

Model / Encoder	Time (hh:mm:ss)
U-Net / ResNet-50	00:02:38
U-Net / MiT-B3	00:05:03
DeepLabV3+ / MobileNetV3-S	00:02:19
Total	**00:10:01**

These results highlight the additional computational cost associated with more expressive architectures, while maintaining a reasonable overall training time for the federated setup.

Based on client-specific performance of the individual architectures, we evaluate an inference-time ensemble constructed from the final federated models. A power-based weighting scheme with exponent 14 is employed to emphasize the strongest-performing model per client, resulting in a conservative ensemble that enhances robustness under heterogeneous data distributions.

Table 5: Inference-time ensemble weights derived from client-specific validation Dice scores.

Model	Metric	C0	C1	C2
U-Net (ResNet-50)	Best Dice	0.5696	0.6978	0.3456
	Weight	0.0260	0.2340	0.0090
U-Net (MiT-B3)	Best Dice	0.6634	**0.7493**	**0.5281**
	Weight	0.2220	0.6350	0.9300
DeepLabV3+ (MobileNetV3-S)	Best Dice	**0.7238**	0.6691	0.4310
	Weight	0.7520	0.1300	0.0610

Table 6: Segmentation performance of the inference-time ensemble across clients.

Client	Loss	Dice	IoU	Acc
Client 0	0.1871	0.7189	0.5891	0.9834
Client 1	0.1732	**0.7540**	0.6093	0.9778
Client 2	0.2829	**0.5341**	0.3644	0.9885

As shown in Tables 5 and 6, the inference-time ensemble matches or outperforms the best individual models across clients. The learned weights adapt to client-specific performance, emphasizing the architecture with the highest validation Dice and effectively exploiting complementary strengths under data heterogeneity.

Overall, the ensemble improves robustness and segmentation stability while remaining fully compatible with the privacy-preserving federated learning setup, motivating further analysis of its interaction with aggregation and communication strategies.

4.3 Data Balancing and Fairness

Motivated by the dataset analysis in Section 3.1, we investigate the effects of data imbalance and fairness in the federated learning setting [15]. The MU-Glioma-Post dataset exhibits substantial demographic skew, with underrepresented minority cohorts that may induce biased or unstable model behavior across clients.

Rather than applying demographic oversampling, which may introduce synthetic artifacts, we adopt a transparent and stratified evaluation strategy aligned with the intrinsic cohort composition. Given the severe pixel-level class imbalance inherent to tumor segmentation, samples are stratified by tumor size using quantile-based bins to ensure representative validation splits across clients and mitigate size-related bias.

During training, we employ a distribution-aware resampling scheme that reweights samples based on their frequency, aiming to mitigate tumor size imbalance across clients. This mechanism is designed to reduce bias toward dominant size categories while preserving sufficient sample diversity.

We conduct a sensitivity analysis over the resampling granularity (n_{bins}) and aggressiveness (β), evaluating multiple configurations to identify a stable and effective operating regime under non-IID data distributions. For all experiments in this analysis, the ensemble power coefficient is fixed to $\gamma = 8$ to isolate the effect of resampling strategies. We report only the final ensemble results, as performance differences are primarily driven by the resampling configuration itself, while intermediate ensemble dynamics across epochs and rounds exhibit minimal variation and do not affect the comparative conclusions.

Table 7: Impact of distribution-aware resampling settings on federated performance across clients.

Setting	Client	Loss	Dice	IoU	Acc
$n_{\text{bins}} = 3$, $\beta = 1.0$	C0	**0.1982**	**0.6887**	**0.5562**	**0.9816**
	C1	**0.1742**	**0.7532**	**0.6064**	**0.9748**
	C2	**0.2814**	**0.5091**	**0.3415**	**0.9884**
$n_{\text{bins}} = 5$, $\beta = 0.5$	C0	0.2093	0.6714	0.5398	0.9802
	C1	0.1834	0.7389	0.5882	0.9731
	C2	0.2945	0.4876	0.3219	0.9871
$n_{\text{bins}} = 10$, $\beta = 2.5$	C0	0.2287	0.6321	0.4893	0.9785
	C1	0.1992	0.6945	0.5412	0.9715
	C2	0.3105	0.4512	0.2987	0.9852
$n_{\text{bins}} = 2$, $\beta = 0.1$	C0	0.2056	0.6758	0.5421	0.9809
	C1	0.1798	0.7412	0.5923	0.9736
	C2	0.2889	0.4934	0.3298	0.9875
$n_{\text{bins}} = 20$, $\beta = 1.2$	C0	0.2153	0.6548	0.5187	0.9791
	C1	0.1912	0.7103	0.5634	0.9722
	C2	0.2998	0.4721	0.3099	0.9860
$n_{\text{bins}} = 4$, $\beta = 0.9$	C0	0.2015	0.6803	0.5489	0.9811
	C1	0.1768	0.7491	0.6012	0.9740
	C2	0.2856	0.4998	0.3345	0.9880

As summarized in Table 7, a moderate resampling configuration ($n_{\text{bins}} = 3$, $\beta = 1.0$) consistently achieves the best balance between performance and stability across clients. This setting improves Dice and IoU scores without introducing excessive variance during training.

In contrast, more aggressive resampling strategies lead to unstable optimization and overfitting to rare samples, resulting in noticeable performance degradation on Client 2. Under these conditions, ensemble weights shift toward DeepLabV3+, indicating greater robustness of this architecture under increased distributional noise.

These results support the use of the Dirichlet-based non-IID split described in Section 3.1, which induces realistic client-level heterogeneity. The observed trends indicate that the proposed framework remains robust and generalizes well despite pronounced data imbalance across clients.

4.4 Aggregation Algorithms

In federated learning, aggregation algorithms operate at the server level, where locally trained model updates from multiple clients are combined into a shared global model. This step is critical, as it governs how information learned from heterogeneous and distributed datasets is integrated across training rounds.

The aggregation strategy directly influences convergence speed, training stability, and final performance, particularly in medical imaging settings where clients may differ in data volume and optimization dynamics. The standard Federated Averaging (FedAvg) [1] algorithm aggregates client updates using a weighted mean based on local dataset size, but does not explicitly address client drift or heterogeneity.

In these experiments, client data distributions remain unchanged to preserve the naturally occurring non-IID setting described in Section 3.1. This allows us to isolate the effect of the aggregation mechanism and evaluate how different strategies handle heterogeneity without additional data balancing.

We compare several aggregation algorithms against the FedAvg baseline introduced in Section 4.2: FedProx [2], which constrains local updates through a proximal regularization term weighted by a coefficient μ that controls the strength of the penalty and limits client drift from the global model; FedOpt [16], which applies adaptive optimization methods (e.g., momentum or Adam) at the server level to improve convergence; and FedNova [3], which normalizes client updates to account for variability in local training effort and step counts.

In addition to individual model evaluation, we adopt an ensemble-based inference approach that combines global models trained with different aggregation strategies. Using a power-weighted scheme with exponent $\gamma = 8$, the ensemble emphasizes higher-performing models while preserving complementary information, providing a robust basis for comparing aggregation methods under heterogeneous federated conditions.

Table 8: Inference-time ensemble weights derived from client-specific validation Dice scores (FedNova).

Model	Metric	C0	C1	C2
U-Net (ResNet-50)	Best Dice	0.5419	**0.6975**	0.4087
	Weight	0.078	0.432	0.391
U-Net (MiT-B3)	Best Dice	0.5273	0.6600	0.1549
	Weight	0.063	0.278	0.001
DeepLabV3+ (MobileNetV3-S)	Best Dice	**0.7316**	0.6638	**0.4318**
	Weight	0.859	0.290	0.608

Table 9: Segmentation performance of the inference-time ensemble across clients (FedNova).

Client	Loss	Dice	IoU	Acc
Client 0	0.1714	**0.7354**	0.5944	0.9814
Client 1	0.1962	**0.7037**	0.5482	0.9750
Client 2	0.3316	0.4099	0.2578	0.9858

Table 10: Inference-time ensemble weights derived from client-specific validation Dice scores (FedOpt).

Model	Metric	C0	C1	C2
U-Net (ResNet-50)	Best Dice	0.3645	0.5754	0.1508
	Weight	0.007	0.295	0.027
U-Net (MiT-B3)	Best Dice	0.3777	0.3513	0.0629
	Weight	0.010	0.006	0.022
DeepLabV3+ (MobileNetV3-S)	Best Dice	**0.6722**	**0.6409**	**0.2847**
	Weight	0.983	0.699	0.951

Table 11: Segmentation performance of the inference-time ensemble across clients (FedOpt).

Client	Loss	Dice	IoU	Acc
Client 0	0.4297	0.6722	0.5352	0.9743
Client 1	0.2680	**0.6594**	0.4931	0.9628
Client 2	0.6159	**0.2830**	0.1682	0.9827

Table 12: Inference-time ensemble weights derived from client-specific validation Dice scores (FedProx, $\mu = 0.1$).

Model	Metric	C0	C1	C2
U-Net (ResNet-50)	Best Dice	0.5577	**0.6227**	0.2579
	Weight	0.144	0.410	0.187
U-Net (MiT-B3)	Best Dice	0.3639	0.5693	0.1869
	Weight	0.005	0.200	0.004
DeepLabV3+ (MobileNetV3-S)	Best Dice	**0.6966**	0.6189	**0.3944**
	Weight	0.852	0.390	0.962

Table 13: Segmentation performance of the inference-time ensemble across clients (FedProx, $\mu = 0.1$).

Client	Loss	Dice	IoU	Acc
Client 0	0.2043	0.6863	0.5531	0.9790
Client 1	0.2312	**0.6673**	0.5035	0.9702
Client 2	0.3557	**0.3951**	0.2462	0.9830

Table 14: Inference-time ensemble weights derived from client-specific validation Dice scores (FedProx, $\mu = 0.01$).

Model	Metric	C0	C1	C2
U-Net (ResNet-50)	Best Dice	0.5848	0.6783	0.4506
	Weight	0.105	0.300	0.207
U-Net (MiT-B3)	Best Dice	0.6599	**0.7223**	**0.5233**
	Weight	0.276	0.496	0.683
DeepLabV3+ (MobileNetV3-S)	Best Dice	**0.7299**	0.6467	0.4166
	Weight	0.619	0.205	0.110

Table 15: Segmentation performance of the inference-time ensemble across clients (FedProx, $\mu = 0.01$).

Client	Loss	Dice	IoU	Acc
Client 0	0.1668	**0.7451**	0.6175	0.9843
Client 1	0.1895	**0.7355**	0.5845	0.9767
Client 2	0.3035	0.5162	0.3479	0.9883

Tables 8–15 demonstrate that inference-time ensemble modeling yields consistently strong and stable performance across clients, even under pronounced data heterogeneity. The results suggest that combining complementary components, such as aggregation strategies and ensemble-based inference, constitutes a promising approach for improving robustness and overall segmentation performance in federated medical imaging scenarios.

4.5 Rounds vs. Epochs

In this set of experiments, we investigate the trade-off between the number of federated communication rounds and the number of local training epochs per round, while maintaining a comparable overall training budget. All experiments in this section employ the standard Federated Averaging (FedAvg) aggregation strategy to ensure a consistent optimization baseline.

To isolate the effect of the training schedule, client data distributions are kept fixed across all configurations, following the non-IID Dirichlet split described in Section 3.1. By varying the trade-off between communication frequency and local training depth, we examine how aggregation frequency influences convergence behavior, robustness to client drift, and overall segmentation performance under heterogeneous data conditions. In addition to segmentation metrics, we also report the total execution time (in seconds) to evaluate the computational and communication efficiency of different rounds–epochs configurations.

Table 16: Effect of the rounds–epochs trade-off.

Config	Client	Loss	Dice	IoU	Acc	Time
Rounds 2	C0	0.1687	0.7374	0.6101	0.9845	
Epochs 10	C1	0.1789	0.7474	0.5991	0.9770	419.34
LR 1e–3	C2	0.2896	0.4965	0.3303	0.9895	
Rounds 3	C0	0.1739	0.7365	0.6022	0.9828	
Epochs 7	C1	0.1783	0.7420	0.5929	0.9768	416.32
LR 1e–3	C2	0.2896	0.4965	0.3303	0.9895	
Rounds 4	C0	0.1863	0.7120	0.5831	0.9821	
Epochs 6	C1	0.1836	0.7368	0.5870	0.9768	534.79
LR 1e–3	C2	0.2905	0.5059	0.3386	0.9888	
Rounds 5	C0	0.1679	0.7352	0.6045	0.9841	
Epochs 10	C1	0.1691	**0.7599**	**0.6151**	0.9780	623.19
LR 1e–3	C2	0.2919	0.4950	0.3289	0.9894	
Rounds 7	C0	0.1614	0.7488	0.6216	0.9846	
Epochs 7	C1	0.1705	0.7558	0.6105	0.9777	614.70
LR 1e–3	C2	0.2747	0.5380	0.3679	0.9899	
Rounds 10	C0	0.1614	0.7488	0.6216	0.9846	
Epochs 5	C1	0.1705	0.7558	0.6105	0.9777	679.65
LR 1e–3	C2	0.2760	0.5246	0.3556	0.9896	
Rounds 10	C0	0.1616	0.7474	0.6186	0.9847	
Epochs 10	C1	0.1747	0.7496	0.6030	0.9776	962.58
LR 1e–3	C2	0.2676	0.5532	0.3824	0.9903	
Rounds 10	C0	**0.1554**	0.7537	**0.6273**	**0.9849**	
Epochs 10	C1	0.1675	0.7662	0.6253	0.9789	1655.51
LR 5e–4	C2	**0.2426**	**0.5829**	**0.4113**	**0.9909**	
Rounds 10	C0	0.1591	0.7547	0.6244	0.9843	
Epochs 15	C1	**0.1624**	0.7716	0.6326	**0.9793**	1210.46
LR 5e–4	C2	0.2492	0.5701	0.3987	0.9908	
Rounds 15	C0	0.1591	**0.7547**	0.6244	0.9843	
Epochs 10	C1	0.1632	**0.7727**	**0.6326**	0.9790	1559.88
LR 5e–4	C2	0.2492	0.5701	0.3987	0.9908	

As summarized in Table 16, the interaction between the number of communication rounds and local training epochs plays a critical role when the total training budget is fixed. Under non-IID data distributions, configurations that emphasize more frequent communication combined with fewer local epochs per round consistently yield more stable training dynamics and improved segmentation performance across clients.

In particular, Table 16 shows that increasing the aggregation frequency mitigates client drift and allows the global model to better integrate heterogeneous client updates, while excessive local optimization tends to amplify client-specific biases and overfitting. As a result, even for comparable total numbers of local epochs, distributing optimization across more communication rounds with moderate local updates proves to be a more effective and robust strategy for federated medical image segmentation.

The configuration with $R = 15$, $E = 10$, and learning rate 5×10^{-4} shows consistent behavior across clients with stable convergence. This motivates a more detailed analysis, and we report the complete results for this setting in the following tables.

Table 17: Inference-time ensemble weights derived from client-specific validation Dice scores (FedAvg, $LR = 5e-4$, $R = 15$, $E = 10$, $power = 8$).

Model	Metric	C0	C1	C2
U-Net (ResNet-50)	Best Dice	0.6870	0.7340	0.5351
	Weight	0.212	0.375	0.427
U-Net (MiT-B3)	Best Dice	**0.7512**	**0.7456**	**0.5405**
	Weight	0.433	0.425	0.462
DeepLabV3+ (MobileNetV3-S)	Best Dice	0.7331	0.6789	0.4524
	Weight	0.356	0.201	0.111

Table 18: Segmentation performance of the inference-time ensemble across clients (FedAvg, $LR = 5e-4$, $R = 15$, $E = 10$, $power = 8$).

Client	Loss	Dice	IoU	Acc
Client 0	0.1591	**0.7547**	0.6244	0.9843
Client 1	0.1632	**0.7727**	0.6326	0.9790
Client 2	0.2492	**0.5701**	0.3987	0.9908

As shown in Tables 17 and 18, extending federated training to a longer schedule ($R = 15$, $E = 10$) yields consistently strong and stable performance across all clients under non-IID data distributions. The increased number of communication rounds enables more effective integration of heterogeneous client updates, improving convergence and robustness.

Notably, the inference-time ensemble strategy leads to clear performance gains for all clients, highlighting the benefit of combining complementary architectures at prediction time. When coupled with appropriate training strategies, such as a balanced allocation between communication rounds and local epochs, this approach further enhances segmentation performance and stability in heterogeneous federated learning scenarios.

4.6 Communication Optimization

In practical federated learning deployments, communication efficiency is a key constraint, as client–server communication is often limited by uplink bandwidth and latency. Transmitting full-precision model updates at each communication round can significantly increase training time and limit scalability [17], particularly in resource-constrained clinical environments.

To address this, we evaluate lightweight communication optimization techniques that reduce the size of transmitted updates while maintaining competitive performance. All experiments follow the federated learning setup described in Section 2, using the original non-IID data split from Section 3.1, FedAvg aggregation, and a fixed inference-time ensemble with power $\gamma = 8$. We specifically analyze FP16 quantization, which lowers numerical precision, and Top-k sparsification, which transmits only the most significant parameter updates, isolating the impact of communication efficiency on convergence and segmentation quality.

Table 19: Communication optimization under different experiments

Trial	Client	Loss	Dice	IoU	Acc	Time
Base. (FP32)	C0	0.1591	0.7547	0.6244	0.9843	
	C1	0.1632	0.7727	0.6326	0.9790	1559.88
	C2	0.2492	0.5701	0.3987	0.9908	
Quant. (FP16)	C0	0.1650	0.7471	0.6155	0.9832	
	C1	0.1649	0.7728	0.6337	0.9783	1421.42
	C2	0.2695	0.5384	0.3681	0.9892	
Spars. (10%)	C0	0.2448	0.7264	0.5931	0.9804	
	C1	0.2095	0.7798	0.6426	0.9769	1357.89
	C2	0.4817	0.4777	0.3131	0.9837	

As summarized in Table 19, both communication optimization strategies achieve a noticeable reduction in execution time compared to the FP32 baseline, with only moderate effects on segmentation performance. FP16 quantization largely preserves accuracy across clients, while Top-k sparsification introduces a stronger trade-off between efficiency and robustness, particularly under severe client-level heterogeneity.

4.7 Synchronous vs. Asynchronous

In synchronous federated learning, the server aggregates updates only after receiving contributions from all selected clients, which can introduce delays due to slow participants. To mitigate this straggler effect, we also evaluate an asynchronous training protocol, where the server updates the global model immediately upon receiving each client update [18].

As asynchronous updates may introduce higher variance and model staleness, we increase the ensemble power coefficient to $\gamma = 14$ in this setting. This choice places stronger emphasis on the best-performing model at inference time, helping stabilize predictions and compensate for increased heterogeneity induced by asynchronous aggregation.

Table 20: Computational efficiency comparison between synchronous and asynchronous federated training.

Model	Metric	Value
U-Net / ResNet-50	Sync. Time (s)	419.93
	Async. Time (s)	363.24
	Reduction (%)	-13.5
U-Net / MiT-B3	Sync. Time (s)	1165.38
	Async. Time (s)	1025.95
	Reduction (%)	-12.0
DeepLabV3+ / MobileNet	Sync. Time (s)	373.32
	Async. Time (s)	323.31
	Reduction (%)	-13.4

As shown in Table 20, adopting an asynchronous training protocol consistently reduces the overall training time across all evaluated architectures. The transformer-based MiT-B3 model exhibits a reduction of approximately 12%, while the convolutional architectures (ResNet-50 and MobileNetV3) achieve time savings of around 13.5%. These results indicate that asynchronous aggregation effectively mitigates idle time induced by computational heterogeneity among models, leading to improved training efficiency.

Table 21: Performance comparison between synchronous and asynchronous federated learning.

Trial	Client	Loss	Dice	IoU	Acc
Sync.	C0	0.1809	0.7137	0.5878	0.9838
	C1	0.1792	0.7482	0.6000	0.9768
	C2	0.2485	0.5828	0.4113	0.9897
Async.	C0	**0.1368**	**0.7895**	**0.6628**	0.9857
	C1	0.1492	**0.7895**	0.6552	0.9808
	C2	0.2379	0.5923	0.4207	**0.9903**

As reported in Table 21, the asynchronous training protocol not only accelerates convergence but also yields consistently higher segmentation performance. Clients 0 and 1 exhibit a marked increase in Dice scores (up to 0.7895), while the most challenging client (Client 2) shows a modest yet stable improvement. This behavior suggests that frequent, non-blocking updates act as an implicit regularizer, reducing overfitting to client-specific batch statistics and enabling more effective knowledge integration, particularly for transformer-based models.

4.8 Cross-Silo vs. Cross-Device

The experiments presented so far follow a cross-silo federated learning setting, where a small, fixed number of clients correspond to stable medical institutions with relatively large local datasets. This regime benefits from predictable client participation, sufficient computational resources, and lower variance in local updates, enabling more stable convergence and reliable aggregation behavior in federated medical imaging tasks [19].

To evaluate scalability under increased decentralization, we extend this setup to five and seven clients, approximating a cross-device federated learning regime. As the number of clients grows, local datasets become smaller and more heterogeneous, amplifying non-IID effects, increasing update variance, and making the optimization process more sensitive to client imbalance and data fragmentation.

All experiments employ the non-IID Dirichlet splitting strategy described in Section 3.1 and keep the training protocol fixed to isolate the effect of client count. Federated training is conducted with $R = 7$ communication rounds and $E = 7$ local epochs, while inference relies on a power-weighted ensemble strategy with $\gamma = 8$. This controlled setup enables a direct comparison between cross-silo and cross-device regimes and highlights how increased decentralization impacts convergence, robustness, and ensemble effectiveness.

Table 22: Client-wise data distribution for the $N = 5$ non-IID Dirichlet split ($\alpha = 0.3$).

Client	Split	n	Bin 0	Bin 1	Bin 2
0	Train	32	1	31	0
0	Val	9	0	9	0
1	Train	49	9	17	23
1	Val	13	1	7	5
2	Train	4	2	2	0
2	Val	1	1	0	0
3	Train	43	42	0	1
3	Val	11	11	0	0
4	Train	32	1	1	30
4	Val	8	0	0	8

Table 23: Inference-time ensemble weights derived from client-specific validation Dice scores (5 clients).

Model/Encoder	Metric	C0	C1	C2	C3	C4
U-Net (ResNet-50)	Best Dice	0.6605	0.6334	0.7748	0.5156	0.7083
	Weight	0.197	0.202	0.221	0.097	0.306
U-Net (MiT-B3)	Best Dice	0.6929	**0.7036**	**0.8879**	**0.6405**	**0.7504**
	Weight	0.289	0.690	0.658	0.549	0.4806
DeepLabV3+ (MobileNetV3-S)	Best Dice	**0.7445**	0.6401	0.7188	0.6064	0.6745
	Weight	0.514	0.208	0.121	0.354	0.207

Table 24: Segmentation performance of the inference-time ensemble across clients (5 clients).

Client	Loss	Dice	IoU	Accuracy
Client 0	0.1642	0.7433	0.6066	0.9836
Client 1	0.1951	**0.7129**	0.5545	0.9742
Client 2	0.0941	0.8867	0.7964	0.9964
Client 3	0.2039	**0.6499**	0.4919	0.9891
Client 4	0.1860	**0.7552**	0.6067	0.9703

Table 25: Client-wise data distribution for the $N = 7$ non-IID Dirichlet split ($\alpha = 0.3$).

Client	Split	n	Bin 0	Bin 1	Bin 2
0	Train	76	38	28	10
0	Val	20	12	7	1
1	Train	56	2	17	37
1	Val	14	1	3	10
2	Train	8	5	2	1
2	Val	2	2	0	0
3	Train	4	0	3	1
3	Val	2	0	0	2
4	Train	1	0	0	1
4	Val	1	0	0	1
5	Train	4	3	0	1
5	Val	1	0	0	1
6	Train	10	5	4	1
6	Val	3	0	3	0

Table 26: Segmentation performance of the inference-time ensemble across clients (7 clients).

Client	Loss	Dice	IoU	Accuracy
Client 0	0.2694	**0.5568**	0.3907	0.9829
Client 1	0.1698	**0.7585**	0.6133	0.9773
Client 2	0.1487	**0.7537**	0.6047	0.9901
Client 3	0.1027	**0.8738**	0.7759	0.9820
Client 4	0.0916	**0.8775**	0.7818	0.9858
Client 5	0.3057	**0.5474**	0.3768	0.9599
Client 6	0.1186	**0.8244**	0.6958	0.9866

As shown in Tables 24 and 26, the proposed ensemble-based federated learning framework achieves consistently strong segmentation performance across both cross-silo and cross-device settings. Despite increased data fragmentation and heterogeneity with a larger number of clients (Tables 22 and 25), the inference-time ensemble effectively integrates complementary model representations, resulting in stable Dice and IoU scores across clients and demonstrating robustness to non-IID severity.

4.9 Client Selection

Standard federated learning typically relies on uniform random sampling for client selection, which may underutilize informative clients in highly heterogeneous scenarios, slowing convergence and limiting generalization.

To address this limitation, we investigate adaptive client selection strategies derived from FedAvg that replace the uniform probability $P_k = \frac{1}{N}$ with weighted distributions based on client-level evaluation feedback, enabling the server to prioritize more informative updates [20].

We consider two strategies: a *performance-aware* approach that favors clients with poorer validation metrics to focus learning on harder cases, and a *fairness-aware* approach that balances participation frequency to promote equitable client involvement across rounds.

Table 27: Impact of client selection strategies on segmentation performance.

Strategy	Client	Loss	Dice	IoU	Acc
Baseline	Client 0	0.1663	0.7330	0.6036	0.9841
	Client 1	0.1714	0.7520	0.6043	0.9772
	Client 2	0.2610	0.5489	0.3783	0.9880
Loss-Based	Client 0	0.1804	0.7403	0.6112	0.9822
	Client 1	0.1713	0.7560	0.6119	0.9785
	Client 2	0.2491	0.5736	0.4021	0.9896
IoU-Based	Client 0	0.1577	0.7607	0.6369	0.9838
	Client 1	0.1586	**0.7820**	**0.6444**	0.9796
	Client 2	0.2353	0.5966	0.4251	0.9896
Gradient-Based	Client 0	0.2017	0.6703	0.5458	0.9827
	Client 1	0.1653	0.7704	0.6286	0.9783
	Client 2	0.2621	0.5536	0.3828	**0.9897**
Fairness-Based	Client 0	0.1589	0.7640	0.6403	0.9857
	Client 1	0.1757	0.7554	0.6087	0.9772
	Client 2	0.2450	0.5762	0.4047	0.9876
Dice-Based	Client 0	0.1838	0.7146	0.5840	0.9823
	Client 1	**0.1576**	0.7803	0.6414	0.9790
	Client 2	0.2450	0.5762	0.4047	0.9876

As shown in Table 27, client selection has a substantial impact on federated learning performance. The IoU-based strategy consistently achieves the best overall results, improving segmentation quality across all clients and yielding the highest Dice score for the most challenging client (Client 2). This suggests that IoU is a reliable proxy for segmentation difficulty in this setting.

In contrast, gradient-based selection leads to unstable behavior, with noticeable performance degradation for certain clients, indicating sensitivity to noisy updates. The fairness-based strategy offers a stable alternative, achieving competitive performance while ensuring balanced client participation, albeit with slightly reduced gains for the hardest client.

5 Results

The experimental analysis in Section 4 demonstrates that no single design choice is sufficient to ensure reliable performance under heterogeneous and non-IID conditions. Instead, robustness and consistent client-level improvements emerge from the coordinated integration of multiple complementary strategies. In this section, we consolidate these findings and present the resulting framework that achieves stable and superior segmentation performance across clients compared to isolated local training or naive federated baselines.

5.1 Final Configuration

The final configuration represents the synthesis of the most effective strategies identified in Section 4, applied on top of the baseline framework described in Section 2. Rather than introducing new architectural components, this configuration refines the training protocol, aggregation behavior, communication strategy, and inference mechanism to maximize robustness and stability in realistic federated settings.

Concretely, we adopt a *synchronous FedAvg* setup, where the server applies a strict synchronization barrier and updates the global model only after the selected cohort of clients completes each communication round. This design reduces update staleness and yields more predictable convergence across rounds. Training is executed for $R = 15$ federated rounds with $E = 10$ local epochs per round.

To improve efficiency without compromising segmentation quality, we combine FP16 mixed-precision training with Top-10% gradient sparsification, substantially reducing compute and communication overhead. Client participation is determined via the same hybrid selection policy, using $\alpha = 0.7$ to maintain a representative yet performance-oriented subset of clients across rounds.

Finally, global predictions are produced using dynamic inference-time ensembling with a power-based weighting scheme ($\gamma = 14$). This acts as a strong filter that increases the influence of consistently high-performing models while attenuating noisy contributors, stabilizing the global optimization trajectory under heterogeneous and non-IID client distributions.

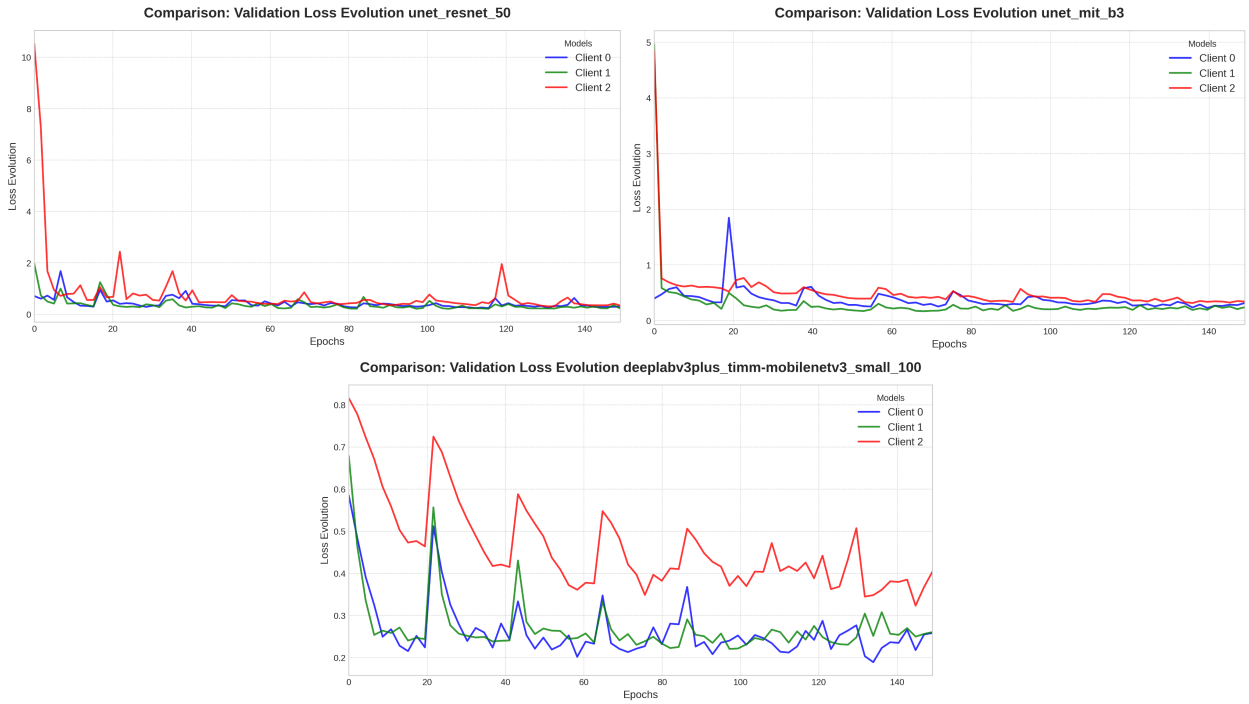

Figure 3: Validation loss evolution for the three evaluated architectures across federated clients, illustrating client-specific convergence behavior under non-IID data distributions.

Table 28: Synchronous federated training performance (FedAvg Sync, FP16 + Top-10% sparsification, Hybrid Client Selection) across model architectures and client-specific validation splits.

Model	Metric	C0	C1	C2
DeepLabV3+ (MobileNetV3-S)	Train Loss	**0.1468**	0.0947	**0.1913**
	Val Loss	0.2210	0.2224	**0.2902**
	Val Dice	0.7185	0.6897	0.5359
	Val IoU	0.5759	0.5308	0.3660
	Val Acc	0.9810	0.9705	0.9869
	Epoch–Round	6–7	9–5	2–14
U-Net (MiT-B3)	Train Loss	0.1552	**0.0924**	0.2084
	Val Loss	**0.2126**	**0.1676**	0.3450
	Val Dice	**0.7661**	**0.7764**	**0.5881**
	Val IoU	**0.6265**	**0.6398**	**0.4165**
	Val Acc	**0.9814**	**0.9777**	**0.9894**
	Epoch–Round	2–13	7–7	6–8
U-Net (ResNet-50)	Train Loss	0.0753	0.0969	0.2644
	Val Loss	0.2372	0.2120	0.3564
	Val Dice	0.6504	0.7221	0.5199
	Val IoU	0.4859	0.5732	0.3512
	Val Acc	0.9742	0.9737	0.9867
	Epoch–Round	10–11	4–10	5–8

Table 28 and Figure 3 indicate stable convergence under the synchronous FedAvg configuration, with validation loss curves showing reduced variance across clients despite non-IID data distributions. After the initial training phase, all architectures converge to a narrow loss range, suggesting effective mitigation of client drift.

The U-Net with MiT-B3 encoder achieves the strongest segmentation performance, attaining the lowest validation loss and highest Dice and IoU scores across almost all clients. This behavior is reflected in Figure 3, where MiT-B3 shows smoother loss trajectories and faster stabilization than CNN-based backbones, indicating improved robustness under heterogeneous client data.

Table 29: Execution time of synchronous federated training (FedAvg Sync, FP16 + Top-10% sparsification, Hybrid Client Selection) per model architecture.

Model / Encoder	Time (hh:mm:ss)
U-Net / ResNet-50	00:04:40
U-Net / MiT-B3	00:13:06
DeepLabV3+ / MobileNetV3-S	00:03:58
Total	**00:21:45**

As shown in Table 29, enabling FP16 mixed-precision training together with Top-10% gradient sparsification leads to a substantial reduction in execution time compared to the baseline synchronous FedAvg experiment without communication and precision optimizations. The total training time is reduced to **00:21:45**, while preserving stable convergence and segmentation performance, demonstrating that the proposed efficiency-oriented optimizations effectively lower computational and communication overhead without degrading model quality.

Table 30: Inference-time ensemble weights derived from client-specific validation Dice scores (FedAvg Sync, FP16 + Top-10% + Hybrid Selection).

Model	Metric	C0	C1	C2
U-Net (ResNet-50)	Best Dice	0.6505	0.7222	0.5199
	Weight	0.0670	0.2330	0.1230
U-Net (MiT-B3)	Best Dice	**0.7662**	**0.7765**	**0.5882**
	Weight	**0.6630**	**0.6440**	**0.6880**
DeepLabV3+ (MobileNetV3-S)	Best Dice	0.7185	0.6898	0.5359
	Weight	0.2700	0.1230	0.1880

Table 31: Segmentation performance of the inference-time ensemble across clients (FedAvg Sync, FP16 + Top-10% + Hybrid Selection).

Client	Loss	Dice	IoU	Acc
Client 0	0.1868	0.7844	**0.6524**	0.9838
Client 1	**0.1621**	**0.7847**	0.6502	0.9797
Client 2	0.3079	0.6040	0.4327	**0.9900**

The ensemble results in Table 31, together with the corresponding weights in Table 30, demonstrate that inference-time ensembling yields a substantial improvement in client-level segmentation performance compared to any individual model. Across all clients, the ensemble consistently surpasses the standalone architectures in Dice and IoU, indicating that complementary representations are effectively combined at inference.

The high power parameter ($\gamma = 14$) plays a critical role in this behavior by strongly emphasizing the most reliable backbone (U-Net with MiT-B3) while still preserving non-negligible contributions from the remaining models. This weighted aggregation allows the ensemble to import useful auxiliary information, improving robustness without being dominated by noisy or underperforming predictions, particularly for heterogeneous clients.

Importantly, the power-based weighting mechanism offers practical flexibility: the parameter γ can be tuned to reflect client- or deployment-specific preferences, such as favoring robustness over diversity or vice versa. In the final configuration, a larger γ proves advantageous, yielding stable gains across clients and reinforcing ensemble learning as a key component for performance consolidation in federated medical imaging systems.

5.2 Interpretability Analysis

High performance metrics alone are insufficient for clinical deployment; medical image analysis models must also ensure transparency and interpretability in their decision-making. While deep learning architectures such as U-Net and DeepLabV3+ achieve strong segmentation performance, their black-box nature limits insight into how predictions are formed.

This lack of interpretability raises concerns that models may rely on confounding factors, such as scanner-specific artifacts or demographic biases, rather than true pathological and anatomical features. Such hidden dependencies can negatively affect generalization and clinical reliability.

These challenges are amplified in a Federated Learning framework, where the global model is optimized without access to raw patient data. Explainable AI (XAI) techniques enable visualization of regions influencing predictions, helping verify anatomical relevance, improve robustness to inter-client heterogeneity, and foster trust for clinical adoption.

Grad-CAM

To visualize our segmentation models, we employ Gradient-weighted Class Activation Mapping (Grad-CAM) [21]. For this medical segmentation task, we define a scalar target score S that aggregates model confidence across the predicted mask, rather than a specific class logit. Given the output logits Y, we compute the mean of the sigmoid-activated probabilities before backpropagation:

$$S = \frac{1}{H \times W} \sum_{i,j} \sigma(Y_{ij}) \tag{22}$$

By backpropagating S to the final convolutional feature maps A^k, we compute the importance weights α_k via global average pooling of the gradients:

$$\alpha_k = \frac{1}{Z} \sum_i \sum_j \frac{\partial S}{\partial A^k_{ij}} \tag{23}$$

Finally, the localization heatmap is generated by a weighted combination of the feature maps, utilizing a ReLU activation to focus on features having a positive influence on the segmentation:

$$L_{Grad-CAM} = \mathrm{ReLU}\left(\sum_k \alpha_k A^k\right) \tag{24}$$

This process highlights the anatomical regions most effectively contributing to the tumor segmentation. The resulting heatmap is then upsampled to the original image resolution to overlay the salient regions.

Qualitative Analysis

The Grad-CAM visualizations reveal distinct behaviors across the evaluated architectures. The U-Net with MIT-B3 encoder demonstrates superior interpretability, consistently and comprehensively capturing the tumor region across different clients. Its attention maps align closely with the pathological area, suggesting that the model leverages relevant anatomical features for segmentation.

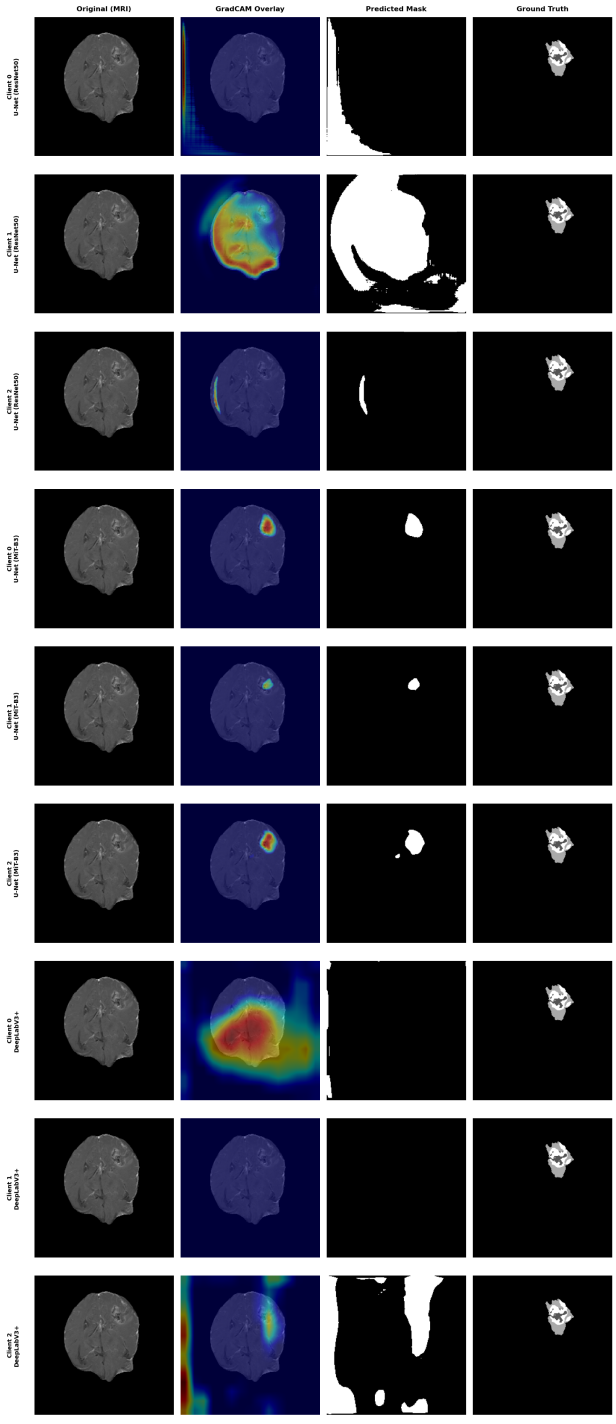

Figure 4: Grad-CAM visualization comparison across different clients (rows 1-3: U-Net + ResNet-50, rows 4-6: U-Net + MIT-B3, rows 7-9: DeepLabV3+). The image shows the original MRI, the Grad-CAM heatmap overlay, the predicted mask, and the ground truth mask.

In contrast, the CNN-based models (U-Net + ResNet-50 and DeepLabV3+) exhibit significant localization failures, indicating a reliance on spurious correlations rather than genuine features. For *U-Net + ResNet-50*, the heatmaps are erratic: Client 0 highlights an irrelevant region at the far left of the image; Client 1 shows diffuse activation across the entire brain; and Client 2 focuses on a thin region contralateral to the actual tumor. Similarly, *DeepLabV3+* struggles to capture meaningful information. In Client 0, it focuses on the image border outside the brain; Client 1 produces no activation (a completely black mask); and for Client 2, it highlights a disjointed region extending from the actual tumor site to the top of the image, alongside false positives on the left.

These qualitative findings correlate strongly with the quantitative ensemble weights derived during validation (e.g., Client 0 weights: $0.07, 0.66, 0.27$). The MIT-B3 model consistently receives the highest weight, implying that the Transformer-based encoder's self-attention mechanism enables it to capture global context more robustly than the localized receptive fields of CNNs. The CNN backbones appear more susceptible to overfitting scanner-specific artifacts or noise in this heterogeneous Federated Learning setting.

Shapley Value Analysis

In a Federated Learning environment [22], accurately quantifying the contribution of each client to the global model's performance is crucial for fairness and incentive mechanism design. We employ Shapley values [23], a solution concept from cooperative game theory, to rigorously measure the marginal contribution of each client to the ensemble. Unlike simple averaging or performance-based weighting on local data, Shapley values account for the interaction effects between clients when forming coalitions.

Mathematical Formulation

Let $N = \{1, \ldots, n\}$ be the set of all clients, and let $U(S)$ be a utility function (in our case, the Dice coefficient of the ensemble model) measured on a held-out global test set for any subset of clients $S \subseteq N$. The Shapley value ϕ_i for client i is defined as the average marginal contribution of client i across all possible coalitions:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} \left[U(S \cup \{i\}) - U(S) \right] \tag{25}$$

where $|S|!$ is the number of permutations of the subset S, $(|N| - |S| - 1)!$ is the number of permutations of the remaining players, and $|N|!$ is the total number of permutations of all players. This formula ensures that the total utility is distributed fairly among the contributors: $\sum_{i \in N} \phi_i = U(N)$.

Qualitative Analysis

We evaluated the Shapley values for our three clients using the ensemble's Dice coefficient as the utility metric. The results, summarized in Table 32, highlight the non-trivial nature of data contribution in medical imaging.

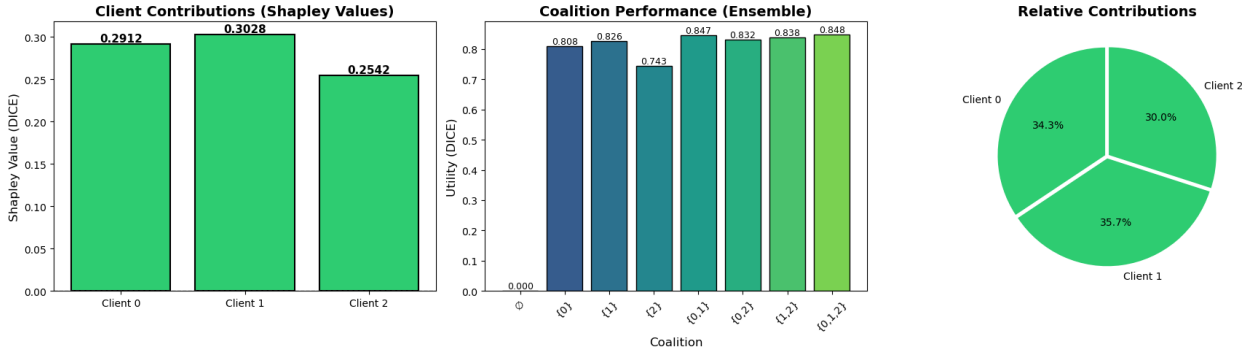

Figure 5: Shapley Value Analysis: Client contributions (left), Coalition utilities showing the improvement in Dice score (center), and Relative contributions as percentages (right).

Table 32: Shapley values and dataset sizes for each client. The total utility of the grand coalition was 0.8482.

Rank	Client	Shapley Value	Dataset Size (n)
1	Client 1	0.3028	96
2	Client 0	0.2912	35
3	Client 2	0.2542	29

As shown in Figure 5, Client 1, which has the largest dataset ($n = 96$), receives the highest Shapley value (0.3028). However, contribution is not strictly proportional to data volume. Client 0, despite having only slightly more samples than Client 2 (35 vs. 29), contributes substantially more to the coalition ($\phi_0 = 0.2912$ vs. $\phi_2 = 0.2542$), suggesting higher complementarity or a more informative data distribution. Moreover, the grand coalition achieves a utility of 0.8482, exceeding the best standalone client performance (Client 1, 0.8256), highlighting the benefit of collaborative learning.

These findings underscore a key insight in federated medical imaging: data quantity alone does not guarantee performance gains. Client 0's higher marginal contribution likely reflects greater data diversity or complexity, whereas Client 2's lower contribution suggests more redundant or easier-to-segment samples. From an operational perspective, Shapley values offer a principled mechanism for incentive allocation in collaborative settings. Although exact computation is costly ($O(2^N)$), the limited number of clients in typical hospital networks makes this approach practical for fairness-aware federated learning.

5.3 Robustness

Robustness is critical in federated medical imaging, where client data are heterogeneous and strongly non-IID. In our experiments, robustness is reflected by stable convergence and consistent segmentation performance under client-level distribution shifts.

We evaluated multiple robustness-oriented strategies in Section 4. The rounds–epochs trade-off (Table 16) shows that more frequent aggregation reduces client drift and stabilizes training. Alternative aggregation methods (Tables 9–15) further highlight differences in stability under uneven local optimization. We also stress-tested scalability by increasing the number of clients (Tables 24 and 26), where performance remains stable despite higher fragmentation.

These findings motivate the final configuration (Section 5.1): synchronous FedAvg with a longer schedule ($R = 15$, $E = 10$) to reduce staleness, a high-power ensemble weighting scheme ($\gamma = 14$) to emphasize the most reliable backbone while retaining complementary signals, and hybrid client selection to avoid sensitivity to outlier updates. Together, these choices consolidate the robustness gains observed in Section 4 into a single deployment-oriented framework.

6 Conclusion

This work presents an empirical study of federated learning strategies for brain tumor segmentation under strong client-level heterogeneity and non-IID data. Rather than focusing on a single optimization technique, we evaluate key federated design choices, including aggregation methods, training schedules, client participation strategies, and communication-efficient mechanisms, under realistic medical imaging conditions.

Our results show that federated performance is highly sensitive to these components, and that robustness cannot be achieved through isolated decisions. Instead, stable convergence and consistent client-level improvements arise from the integration of complementary strategies. Across experiments, inference-time ensemble modeling emerges as an effective mechanism for mitigating heterogeneity, consolidating diverse architectural representations without increasing training complexity or compromising privacy.

Overall, this study shows that careful empirical evaluation of federated learning strategies is essential for reliable deployment in medical imaging. The proposed framework, validated through extensive experimentation, provides a strong foundation for collaborative model training across institutions and highlights ensemble-based inference as a practical tool for improving generalization in heterogeneous federated environments.

References

[1] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1273–1282, 2017.

[2] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of the 3rd MLSys Conference*, 2020.

[3] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[4] Georgios Kaissis, Marcus Makowski, Daniel Rückert, and Rickmer Braren. Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence*, 3(6):473–484, 2021.

[5] Quande Liu, Qi Dou, Lequan Yu, and Pheng-Ann Heng. Federated learning with ensemble models for robust medical image segmentation. In *Proceedings of the 25th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 457–467, 2022.

[6] Yuhao Zhang, Ziyang Jiang, Yizhou Liu, and Xiaojian Wang. Clip-driven universal model for medical image analysis. *IEEE Transactions on Medical Imaging*, 41(10):2573–2585, 2022.

[7] Qinbin Li, Yisheng Diao, Quan Chen, and Bingsheng He. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 38(3):50–60, 2021.

[8] Ali Hatamizadeh, Yi Tang, Vishal Nath, Dong Yang, Andriy Myronenko, Can Xu, and Holger Roth. Unetr: Transformers for 3d medical image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 574–584, 2022.

[9] Zhi-Hua Zhou and Ji Feng. A review of ensemble learning in medical image analysis. *IEEE Transactions on Medical Imaging*, 40(11):2993–3012, 2021.

[10] Seungeun Oh, Hyeonwoo Kim, and Jaewoo Kim. Fedensemble: Improving federated learning robustness via ensemble models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10256–10264, 2022.

[11] Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M. Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 240–248, 2017.

[12] Abdel Aziz Taha and Allan Hanbury. Metrics for evaluating 3d medical image segmentation: Analysis, selection, and tool. *BMC Medical Imaging*, 15(1):29, 2015.

[13] Kenneth Clark, Barbara Vendt, Kirk Smith, Justin Freymann, Jeffrey Kirby, Paul Koppel, Stephen Moore, Shane Phillips, Daniel Maffitt, Mark Pringle, Laurel Tarbox, and Fred Prior. The cancer imaging archive (tcia): Maintaining and operating a public information repository. *Journal of Digital Imaging*, 26(6):1045–1057, 2013.

[14] Brian B. Avants, Nicholas J. Tustison, and Gang Song. A reproducible evaluation of ants similarity metric performance in brain image registration. *NeuroImage*, 54(3):2033–2044, 2011.

[15] Clara Petersen, Orianna DeMasi, Pranav Mishra, Shan Yu, and Michael C. Tschantz. Potential biases in machine learning algorithms using electronic health record data. *Journal of the American Medical Informatics Association*, 26(11):1164–1174, 2019.

[16] Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and Brendan McMahan. Adaptive federated optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[17] Alham Fikri Aji and Kenneth Heafield. Sparse communication for distributed gradient descent. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 440–445, 2017.

[18] Cong Xie, Sanmi Koyejo, and Indranil Gupta. Asynchronous federated optimization. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4173–4179, 2019.

[19] Peter Kairouz, Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2):1–210, 2021.

[20] Takayuki Nishio and Ryo Yonetani. Client selection for federated learning with heterogeneous resources in mobile edge. In *Proceedings of the IEEE International Conference on Communications (ICC)*, 2019.

[21] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[22] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[23] Lloyd S Shapley. A value for n-person games. In *Contributions to the Theory of Games*, volume 2, pages 307–317. Princeton University Press, 1953.