
UnifiedFL: A Dynamic Unified Learning Framework for Equitable Federation

Furkan Pala, Islem Rekik

BASIRA Lab, Imperial-X (I-X) and Department of Computing
Imperial College London, London, United Kingdom
{f.pala23, i.rekik}@imperial.ac.edu

Abstract

Federated learning (FL) has emerged as a key paradigm for collaborative model training across multiple clients without sharing raw data, enabling privacy-preserving applications in domains such as radiology and pathology. However, works on collaborative model training across clients with fundamentally different neural architectures and non-identically distributed datasets remain largely scarce. Besides, existing FL frameworks face several limitations. First, despite claiming to support architectural heterogeneity, most recent FL methods only tolerate variants within a single model family—such as shallower, deeper, or wider CNNs—thereby still presuming a shared global architecture and failing to accommodate federations in which clients deploy fundamentally different network types (e.g., CNNs, GNNs, MLPs). Second, existing approaches often address only the statistical heterogeneity of datasets across clients yet overlook the domain-fracture problem, where each client’s training data stem from distributions that differ markedly from those faced at testing time, an oversight that severely undermines the generalizability of every client model. More importantly, when clients use different model architectures and have differently distributed data—and the test data differ yet again—current methods cause each client’s model to perform poorly. To address such challenges, we propose UnifiedFL, a dynamic *unified* federated learning framework that represents heterogeneous local networks as nodes and edges in a directed *model-graph*, whose weights and biases are optimized by a single, shared graph neural network (GNN). Our three core contributions lie in (i) parameterizing all local architectures through a common GNN, ensuring that incompatible tensors are never transmitted; (ii) introducing a distance-driven clustering mechanism based on Euclidean distances between clients’ GNN parameters to dynamically group hospitals following similar optimization trajectories; and (iii) designing a two-tier aggregation policy that synchronizes frequently within clusters while sparsely across clusters to balance convergence and diversity. Our comprehensive experiments on four MedMNIST disease classification benchmarks and the Hippocampus segmentation task from the Medical Image Segmentation Decathlon demonstrate the outperformance of UnifiedFL over strong FL baselines on both classification and segmentation metrics. UnifiedFL presents the first framework to unify heterogeneous model training in FL via a shared model-graph representation. Our Python UnifiedFL code, benchmarks, and evaluation datasets are available at <https://github.com/basiralab/UnifiedFL>.

1 Introduction

Medical artificial intelligence (AI) has become an integral component of clinical decision-making pipelines, offering unprecedented capabilities in medical image analysis for disease screening, prognosis, and therapeutic planning [1, 2, 3]. These gains are most evident in data-hungry fields such

as radiology and pathology, where advanced machine learning algorithms, including deep neural networks (DNNs), can extract patterns from high-dimensional imaging data to aid clinicians by improving diagnostic accuracy, forecasting disease progression, and informing treatment decisions. However, building high-performing and clinically reliable models often necessitates pooling extensive, diverse datasets from multiple hospitals. Such data-sharing remains fraught with privacy, regulatory, and logistical challenges, especially under stringent rules like HIPAA and GDPR [4, 5]. To overcome the hurdles of data centralization, *federated learning* (FL) has emerged as a transformative approach, allowing each hospital (or “client”) to train models locally while sharing only model updates with a central server, then aggregating model and broadcasting model parameters back to clients without sharing data. [6, 7, 8, 9, 10]. This privacy-preserving paradigm has proven particularly valuable in healthcare, where sensitive patient data cannot be moved freely beyond clinical boundaries. Nonetheless, FL in medical imaging must address additional complexities. The key challenges are (1) heterogeneity of the local computational environment and network architecture, and (2) non-identical, non-independent (*non-IID*) data distributions driven by factors like demographic discrepancies across hospitals.

In medical imaging, different neural network architectures tend to perform better on different types of tasks, depending on the nature of the data and the clinical objective. Convolutional neural networks (CNNs) are preferred for tasks that rely heavily on spatial context, while multi-layer perceptrons (MLPs) or Transformers may be employed for tabular data, time-series records, or high-resolution scans with complex texture features. As a result, healthcare local clients may develop or adopt models with divergent layer types, widths, depths, or input modalities [10]. Conventional FL algorithms, such as FedAvg [6] assumes every client runs an identical network, so its server can merge updates position by position. When confronted with *fully heterogeneous models*, direct aggregation fails due to mismatched weight shapes and inconsistent layer definitions [6, 10]. Although certain approaches alleviate partial heterogeneity by limiting variations to specific layers (e.g., an extra personalization layer or different output head), the complete unification of *entirely* different network architectures remains a significant challenge in federated healthcare applications.

Beyond architectural diversity, medical data typically exhibit strong domain shifts across local clients, often due to variations in scanner types, image acquisition protocols, and patient populations [4, 5]. In practice, the frequency of disease classes can differ drastically from one hospital to another, or image quality may vary based on local hardware. Consequently, models trained under a strict assumption of IID data distribution may perform poorly in real-world federated scenarios, demonstrating low robustness and generalizability [7, 8]. Techniques such as cluster-based FL [11] or personalization [10, 11] have been proposed to mitigate these effects, yet many still rely on a relatively consistent global model architecture, preventing straightforward application in a federated setting involving diverse model architectures.

A promising line of research addresses both *architectural* and *statistical* diversity by embedding disparate models into a shared parameter space, often via graph-based representations. Instead of matching weights index-by-index, each network is converted into a graph structure, where nodes and edges represent biases and weights (or filters), respectively. A graph neural network (GNN) is then employed to align these ‘model-graphs,’ enabling a single set of GNN parameters to drive updates across otherwise incompatible architectures [10]. In conjunction with iterative client-server exchanges, this approach opens the door for a *truly model-agnostic FL* pipeline. Yet, when facing non-IID data, the frequency and manner of communication become crucial: frequent interactions among distinct topologies can lead to parameter interference while too little communication hampers knowledge sharing. To regulate aggregation, one strategy is to group clients with similar network topologies into clusters and reduce cross-cluster interactions, assuming that similarly structured models benefit from more frequent parameter exchange [10, 11]. However, relying on a static and *a priori* defined grouping criteria, e.g., purely topological features such as node degrees or network depth may not be always suitable. Once these clusters are formed at initialization, they remain fixed throughout training. Realistically, however, medical local clients may continuously update or adapt their local model designs, or discover that their learned weights drift significantly as new cases and imaging protocols are introduced [5, 11]. A static clustering scheme thus risks suboptimal groupings, leading to diminished collaboration among clients who become more aligned over time, or forced interactions among those who diverge as training evolves.

In this paper, we present UnifiedFL: a dynamic unified federated learning framework for equitable medical imaging, which aims to address fully heterogeneous architectures, domain shifts, and the evolving nature of local models in federated medical imaging. Our contributions are listed below:

- **Unified learning:** We unify each neural network across clients (be it CNN or MLP) by transforming them into a model-graph representation so that local updates can be performed under a common GNN-based parameterization. Thus, clients and server communicate only the GNN parameters regardless of the heterogeneity of the neural networks in clients, making our federation *truly architecture-agnostic*.
- **Dynamic clustering:** After each communication round every client sends the latest values of its shared GNN parameters to the server. The server measures pair-wise distances between these vectors, groups nearby clients into clusters, and updates the groups at regular intervals. Hospitals whose optimization paths converge are therefore synchronized often, whereas those that drift apart exchange updates less frequently, preventing harmful interference.
- **Enhanced robustness to non-IID data:** Our approach inherently handles the statistical heterogeneity of client data distributions so that each client can train its local model on its own data. This allows each client to select the most appropriate architecture for the statistical distribution of their data. Next, our dynamic clustering mechanism mitigates domain shift among clients’ training datasets by grouping together clients with similar training dynamics, enabling personalized yet collaborative learning within each cluster.

2 Related work

FL in medical imaging has recently expanded beyond the classical assumption that every site trains an identical network on similarly distributed data. Two research streams tackle the ensuing challenges: (i) *heterogeneous-model aggregation*, which aims to merge updates from clients that run *different* architectures, and (ii) *knowledge-distillation frameworks*, which sidestep parameter alignment by training lightweight models called student networks under the supervision of a larger model called teacher network. We review both streams, emphasizing the gaps that motivate UnifiedFL.

Heterogeneous federated learning *HeteroFL* [12] prunes each convolutional layer width-wise so that the convolutional filters align across clients before averaging, but pruning discards low-level features and locks the pruning ratio at design time. *InclusiveFL* [13] attaches a shallow “student” network to smaller devices and averages overlapping layers with deeper “teacher” models; yet the depth split is static, and mismatch in layer types (e.g. depthwise vs. standard convolutions) is still disallowed. *ScaleFL* [14] searches width–depth pairs and adds early-exit heads, improving resource adaptivity, though the search space must be tuned for each architecture and does not adapt after deployment. Parameter-efficient approaches freeze the layers and aggregate small branches: *pFedLoRA* [15] uses low-rank adapters, and *HeteroTune* [16] employs prefix-tuning. These methods cut bandwidth but the adapters remain architecture-specific, so interference resurfaces when scanners, imaging slice thickness, or class priors differ strongly between sites. Clustering techniques such as *FedGroup* [17] mitigate domain shift by grouping clients with similar gradients, but the grouping is computed once at round 0 and cannot react if optimization trajectories later converge or diverge. *FIARSE* introduces importance-aware sub-model extraction and proves an $O(1/\sqrt{T})$ convergence rate, yet it still assumes all clients start from a *common* super-network and it updates sub-models by masking weights rather than by a unified parameter space [18]. *FedGLCL* replaces logits with language-image contrastive pairs: a frozen text encoder supplies a global semantic space and each client aligns its image embeddings to that space with CLIP-style loss [19]. Although FedGLCL reduces client drift on non-IID data, it presupposes the availability of reliable class prompts and a heavyweight text encoder on the server, and it does not handle architectural conflicts because each client still trains its own image backbone with private parameters.

UnifiedFL removes the need for layer alignment, pruning, or adapter surgery. Each architecture is first converted into a directed model-graph whose nodes and edges are all modulated by a fixed-length GNN parameter vector Θ . Because every client transmits only Θ , tensor shapes never have to match. This stands in sharp contrast to scale-search approaches [20, 21], which still rely on a shared backbone and merely resize it—shrinking or deepening layers, channels, or resolutions—to fit each device; such proportional scaling preserves layer-wise averaging but bars truly different architectures. After each communication round, the server computes Euclidean distances between the received Θ vectors, reclusters clients via Ward linkage, and enforces a two-tier schedule: frequent averaging

within clusters and sparse averaging across clusters. Consequently, similar hospitals collaborate often, while highly divergent sites synchronize only after partial convergence—overcoming both the static-cluster constraint of FedGroup [17] and the architectural rigidity of scale-search methods.

Knowledge distillation for heterogeneous models An alternative to parameter alignment is to fuse predictions. *FedMD* [22], *FedDF* [23], and *Cronus* [24] average softened logits on a public data set to train a global student. Medical imaging rarely offers such a public pool; even when available, privacy regulations may restrict its use. *MH-pFLID* [25] eliminates the public set by introducing a “messenger” network that visits each client in turn and accumulates knowledge, but the messenger must itself be communicated and trained, adding latency and memory overhead. Communication-efficient variants compress logits with contrastive objectives [26, 27], yet always require two forward passes per batch (teacher and student) and cannot avoid disclosing class-conditional information.

Unified learning *uGNN* [28] introduces a unified learning paradigm for training heterogeneous neural architectures. Each architecture is transformed into a common graph representation, where weights and biases are encoded as node and edge features of a model graph. A central GNN then performs a custom message-passing procedure over these model graphs to emulate each architecture’s forward pass, enabling knowledge sharing and joint training across models. Rather than optimizing every architecture independently—which is especially challenging when they are trained on data from different distributions—*uGNN* trains only the central GNN. During training, the central GNN learns update rules for node and edge features that mirror the standard SGD-based updates of weights and biases in the original architectures while simultaneously leveraging knowledge shared across architectures, allowing cross-domain information to enhance each model’s individual performance. However, *uGNN* suffers from a key limitation: it operates in a centralized setting, requiring all model-graphs to be trained in an ensemble learning fashion, which is infeasible in privacy-sensitive domains such as healthcare. Our proposed *UnifiedFL* framework overcomes this issue by designing a federated setup where clients retain their local data and model-graphs, exchanging only a compact, fixed-length GNN parameter vector Θ to ensure architecture-agnostic aggregation without sharing full model weights. Furthermore, we introduce a dynamic, Θ -guided clustering mechanism that adaptively groups clients with similar optimization trajectories, thereby mitigating non-IID interference and accommodating architectural drift during training. This combination enables *UnifiedFL* to deliver the benefits of *uGNN*’s unified parameter space while ensuring privacy and robustness to both model and data heterogeneity.

3 Preliminary: Heterogeneous Federated Learning

Table 1 summarizes the key mathematical symbols used throughout the paper. We adopt boldface uppercase letters (e.g., Θ) for parameter sets, boldface lowercase letters (e.g., $\mathbf{v}$) for vectors, and script fonts (e.g., $\mathcal{D}_i$) for datasets or sets of clients.

Table 1: Key notation and symbols used in our methodology.

Symbol	Description
m	number of clients
$\mathcal{D}_i$	local dataset for client i
Θ	global parameter set (gnn-based)
$\Theta_{[C_k]}$	cluster-level aggregated parameters for cluster C_k
$\mathbf{V}_i, \mathbf{E}_i$	node and edge feature sets of the model-graph for client i
$g_e(u, v), g_v(v)$	group indices for edges and nodes, respectively
$\mathbf{E}_{u,v}$	weight from node u to node v in a model-graph
$\mathbf{V}_v$	bias (node feature) for node v
τ_i	topology descriptor of client i
C_k	cluster k of clients
t_{ic}, t_{bc}	communication intervals for intra- and cross-clusters
t_{init}	threshold round to begin inter-cluster communication
t	total number of federated training rounds
$\mathcal{L}_i$	local loss function at client i
$\sigma(\cdot)$	activation function (e.g., relu)
$\text{SoftSign}(\cdot)$	element-wise softsign operator for scaling and shifting

We consider K clients, where each client $k \in [K]$ holds a local dataset $\{(x_i, y_i)\}_{i=1}^{n_k}$, with $(x_i, y_i) \sim \mathbb{P}_k(x, y)$. The goal is to learn a global model f by solving:

$$\mathcal{L}(\mathbf{w}) = \sum_{k=1}^K \gamma_k \mathcal{L}_k(\mathbf{w}), \quad \text{where} \quad \gamma_k = \frac{n_k}{\sum_{i=1}^K n_i}. \quad (1)$$

Heterogeneity in FL arises from two principal sources: *model heterogeneity* and *statistical heterogeneity* [29], both of which challenge conventional aggregation schemes such as FedAvg [30].

3.1 Model Heterogeneity

Model heterogeneity arises when clients use different model architectures or parameter spaces due to varying hardware capabilities, software environments, or design choices. In this case, the parameter vector for client k is denoted $\mathbf{w}_k \in \mathbb{R}^{p_k}$, where p_k is specific to client k . Aggregation across heterogeneous models is not straightforward, as parameter vectors may differ in dimension, structure, or semantics, rendering conventional weighted averaging infeasible [31].

3.2 Statistical Heterogeneity

Statistical heterogeneity stems from non-identically distributed (non-IID) data across clients. Each client k draws data from a client-specific distribution $\mathbb{P}_k(x, y)$, leading to divergence between local and global optima. The local loss for client k at round t is:

$$\mathcal{L}_k(\mathbf{w}_k^t) = \frac{1}{n_k} \sum_{(x_i, y_i) \sim \mathbb{P}_k} \ell(f(x_i; \mathbf{w}_k^{t-1}), y_i), \quad (2)$$

and the aggregated global update is:

$$\mathbf{w}_G^t = \sum_{k=1}^K \gamma_k \mathbf{w}_k^t. \quad (3)$$

Mathematically, statistical heterogeneity complicates convergence because local gradients $\nabla \mathcal{L}_k(\mathbf{w})$ tend to point in different directions:

$$\mathbb{E}_{\mathbb{P}_k} [\nabla \mathcal{L}_k(\mathbf{w})] \neq \mathbb{E}_{\mathbb{P}} [\nabla \mathcal{L}(\mathbf{w})], \quad (4)$$

where $\mathbb{P}(x, y) = \sum_{k=1}^K \gamma_k \mathbb{P}_k(x, y)$ is the overall population distribution. In other words, local gradient directions are biased toward minimizing their own local losses and may conflict with each other.

The aggregated update can therefore oscillate or fail to make consistent progress:

$$\mathbf{w}_G^{t+1} = \mathbf{w}_G^t - \eta \sum_{k=1}^K \gamma_k \nabla \mathcal{L}_k(\mathbf{w}_G^t), \quad (5)$$

where the sum of gradients may not approximate the true global gradient $\nabla \mathcal{L}(\mathbf{w}_G^t)$ well. This misalignment slows down convergence and may even lead to divergence if client distributions are highly dissimilar [32, 33, 34].

4 Proposed UnifiedFL

Problem formulation. Consider an FL system with m hospitals indexed by $k \in [K]$. Each hospital stores a private image set $\mathcal{D}_k$ sampled from an unknown distribution $\mathbb{P}_k(x, y)$ that may differ across sites (*non-IID*). Hospital k chooses an architecture $f_k(\cdot; \mathbf{W}_k)$ —e.g. CNN, U-Net, or MLP—whose trainable weights and biases are collected in $\mathbf{W}_k \in \mathbb{R}^{d_k}$ with *architecture-specific* dimensionality d_k . Directly averaging the $\mathbf{W}_k$ is impossible because each weight vector has different dimensions.

To enable aggregation we convert client architecture backbone into a *model-graph* $\mathcal{G}_k = (\mathbf{V}_k, \mathbf{E}_k)$. Let $\Theta \in \mathbb{R}^p$ denote the *shared* GNN parameter vector that updates every edge feature $\mathbf{E}_{u,v}$ and node bias $\mathbf{V}_v$ in every client-side model-graph. Optimizing Θ therefore *indirectly* tunes the underlying

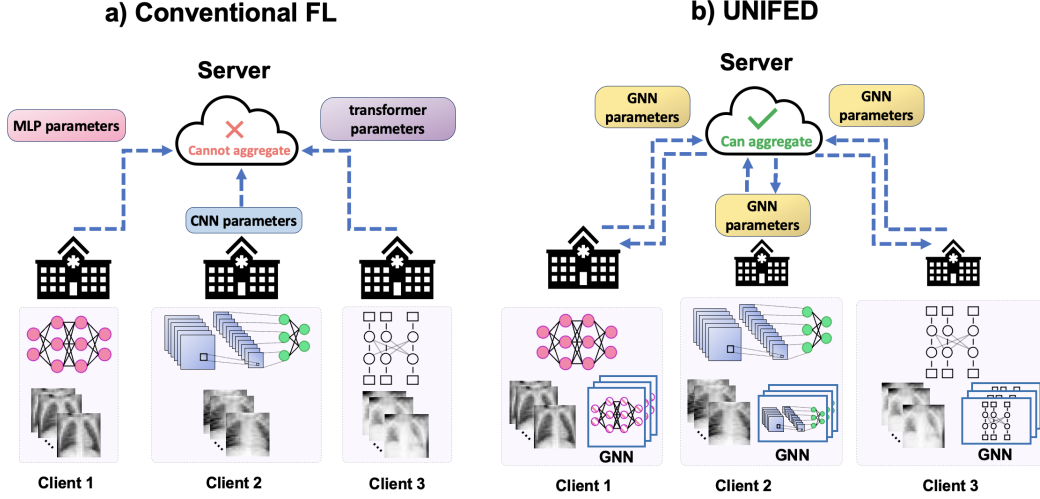


Figure 1: Conceptual comparison between conventional federated learning and the proposed unified learning (UnifiedFL) workflow. **(a) Conventional FL.** Clients deploy heterogeneous backbones (MLP, CNN, Transformer). Each site uploads its native weight tensor to the server (blue dashed arrows). Because the tensors differ in shape, the server cannot perform element-wise aggregation (red cross). **(b) UnifiedFL.** Each client converts its backbone to a model-graph and trains a shared set of GNN parameters Θ that rescale the underlying weights. Only the compact Θ is exchanged. All parameter vectors have identical length, so the server can average them directly (green tick) and broadcast the result back to the hospitals. This mechanism enables architecture-agnostic collaboration without exposing raw images or full model weights.

weights of *all* heterogeneous backbones. We wish to discover a single optimum $\tilde{\Theta}$ that minimizes the mean empirical loss across hospitals,

$$\tilde{\Theta} = \arg \min_{\Theta} F(\Theta), \quad F(\Theta) = \frac{1}{K} \sum_{i=1}^K F_i(\Theta),$$

where $F_i(\Theta)$ is the loss on the private set $\mathcal{D}_i$ after updating $(\mathbf{V}_i, \mathbf{E}_i)$ with Θ .

Dynamic Θ -guided clustering After each local epoch hospital i holds an updated copy $\Theta_{(i)}^{[t]}$. The server forms a distance matrix $D_{ij}^{[t]} = \|\Theta_{(i)}^{[t]} - \Theta_{(j)}^{[t]}\|_2$, applies Ward’s linkage hierarchical clustering, and obtains a partition $\mathcal{C}^{[t]} = \{\mathcal{C}_1^{[t]}, \dots, \mathcal{C}_M^{[t]}\}$ with M clusters. Hence clients that follow similar optimization trajectories are grouped together, while diverging ones are separated.

Two-level aggregation schedule Given $\mathcal{C}^{[t]}$ we perform

$$\Theta_{[\mathcal{C}_m]}^{[t]} = \frac{1}{|\mathcal{C}_m^{[t]}|} \sum_{i \in \mathcal{C}_m^{[t]}} \Theta_{(i)}^{[t]} \quad (\text{every } t_{ic} \text{ rounds}),$$

$$\Theta^{[t]} = \frac{1}{M} \sum_{m=1}^M \Theta_{[\mathcal{C}_m]}^{[t]} \quad (\text{every } t_{bc} > t_{ic} \text{ rounds}),$$

where $\Theta_{[\mathcal{C}_k]}^{[t]}$ is the intra-cluster average and $\Theta^{[t]}$ is the global vector broadcast for the next round. Frequent intra-cluster exchange accelerates convergence among similar hospitals; infrequent inter-cluster exchange prevents destructive interference until models have partially aligned. Because only the *fixed-length* vectors $\Theta_{(i)}^{[t]}$ are transmitted—never raw images or backbone-specific weights—the procedure respects privacy while jointly optimizing heterogeneous, dynamically evolving architectures.

Our motivation. Fig. 1a shows a typical failure mode of FedAvg when hospitals deploy dissimilar backbones. Client 1 trains an MLP, Client 2 trains a CNN, and Client 3 trains a Transformer. The weight tensors uploaded to the server differ in rank and spatial layout, so the server cannot perform element-wise aggregation. Fig. 1b outlines the UnifiedFL remedy. Each backbone is rewritten as a directed acyclic *model-graph*: nodes hold biases or spatial activations, and edges hold convolutional or linear weights. All clients then optimize a shared GNN parameter vector Θ that rescales these node and edge features. Because every Θ has identical length, the server can average updates directly and broadcast the result without shape conflicts. The complete data-flow, including intra- and inter-cluster aggregation, is detailed in Fig. 2. Building on this motivation, we formulate the following hypotheses that underpin the design of our proposed framework:

H1. A backbone-agnostic representation in which every local network is rewritten as a model-graph and updated solely through a shared parameter vector Θ is sufficient to remove all tensor-shape barriers to aggregation.

H2. Measuring pair-wise Euclidean distances between the current Θ vectors and reclustering clients at each round yields communication groups that track optimization similarity and therefore reduce the gradient conflict induced by non-IID data.

H3. Combining H1 and H2 allows federated training to approach the accuracy of a centralized oracle while keeping raw images and full model weights strictly on site.

In the following subsections, we detail the building blocks of UnifiedFL.

4.1 Model-graph construction and unification

Given a client i with a local network of arbitrary design, we define $\mathbf{V}_i = \{\mathbf{V}_v : v \in \mathcal{N}_i\}$, $\mathbf{E}_i = \{\mathbf{E}_{u,v} : (u \rightarrow v) \in \mathcal{A}_i\}$, where $\mathcal{N}_i$ is the set of nodes (neurons or feature-map positions) and $\mathcal{A}_i$ is the set of directed edges (weights). This transformation does not require layer-by-layer alignment; hence the architecture is preserved in a flexible graph form [35].

Unified GNN parameters. Rather than exchanging full local models, our framework introduces a global GNN-based parameter set Θ that *indirectly* updates local weights ($\mathbf{E}_{u,v}$) and biases ($\mathbf{V}_v$). As shown in Fig. 2 (b), Θ is partitioned into $\{\Theta_{\text{node}}, \Theta_{\text{edge}}\}$ with additional shift and scale parameters. Specifically, edges and nodes are grouped, and each group is associated with a scale/shift pair $(\Theta_{\text{edge}}, \Theta_{\text{edge_shift}})$ or $(\Theta_{\text{node}}, \Theta_{\text{node_shift}})$. Let $g_e(u, v)$ be the group assignment for edge $(u \rightarrow v)$ and $g_v(v)$ the group for node v . Next, we update the edge and node embeddings as follows:

$$\begin{aligned}\mathbf{E}_{u,v} &\leftarrow \text{SoftSign}\left(\mathbf{E}_{u,v} \Theta_{\text{edge}}^{(g_e(u,v))} + \Theta_{\text{edge_shift}}^{(g_e(u,v))}, \Theta_{\text{scale_edge}}\right), \\ \mathbf{V}_v &\leftarrow \text{SoftSign}\left(\mathbf{V}_v \Theta_{\text{node}}^{(g_v(v))} + \Theta_{\text{node_shift}}^{(g_v(v))}, \Theta_{\text{scale_node}}\right).\end{aligned}$$

4.2 Federated optimization and static clustering

Local feedforward and loss. Once each node and edge feature is updated by Θ , the feedforward pass of the local model-graph proceeds as:

$$\mathbf{H}_v^{(\ell)} = \sigma\left[\sum_{u: \mathbf{A}_{u,v}=1} (\mathbf{E}_{u,v} \mathbf{H}_u^{(\ell-1)} + \mathbf{V}_v)\right],$$

where $\mathbf{H}_v^{(\ell)}$ denotes the activations at node v in layer ℓ , and $\sigma(\cdot)$ is an activation function. A local loss $\mathcal{L}_i(\Theta)$ compares the outputs of the model-graph with local labels in dataset $\mathcal{D}_i$.

Federated averaging. Under a standard FL setting, the server broadcasts $\Theta^{[t]}$ to all clients each round t . Client i performs local gradient steps:

$$\Theta_{[i]}^{[t]} = \Theta^{[t]} - \eta \nabla_{\Theta} \mathcal{L}_i(\Theta^{[t]}) \quad \forall i,$$

and the server aggregates via weighted averaging:

$$\Theta^{[t+1]} = \frac{1}{m} \sum_{i=1}^m \Theta_{[i]}^{[t]}.$$

Topology-aware static clustering. Each client computes a *topology descriptor* τ_i (e.g., node degrees, betweenness centrality) once at initialization. The server forms clusters $\{\mathcal{C}_1, \dots, \mathcal{C}_K\}$ based on these

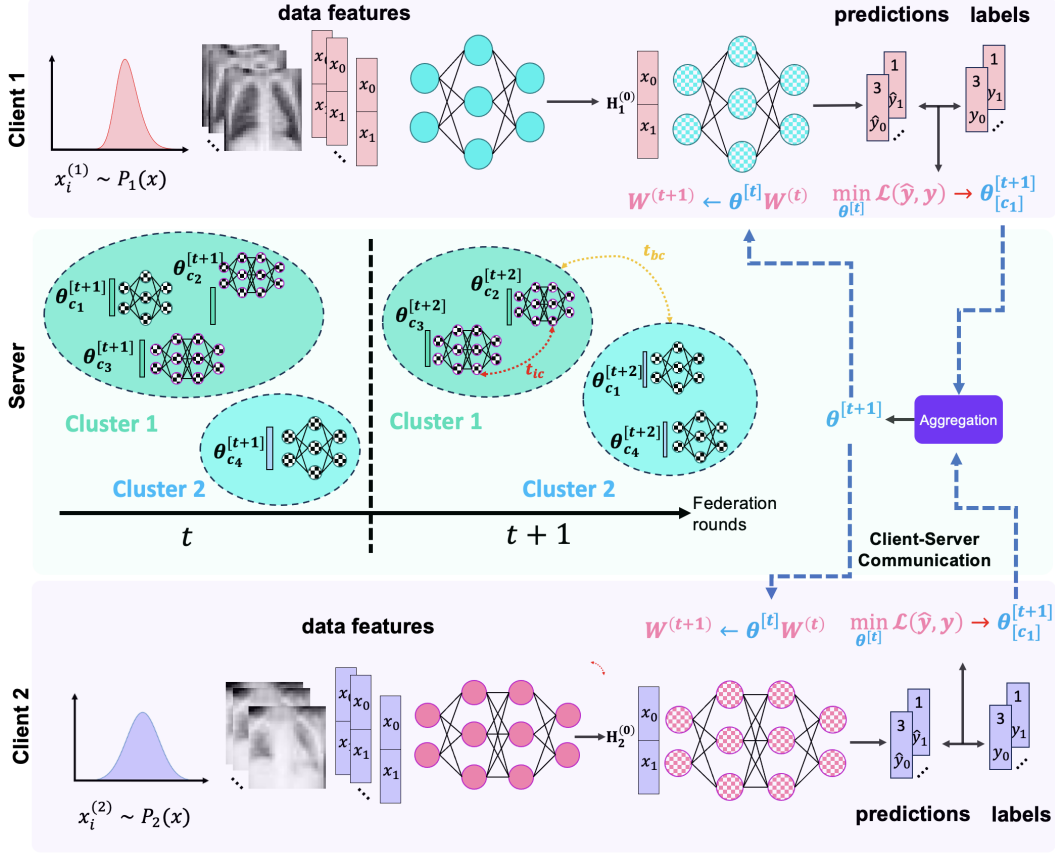


Figure 2: **Overview of the proposed UnifiedFL workflow.** At every federation round t each client (top & bottom rows show examples for two hospitals with distinct image distributions $P_1(x)$ and $P_2(x)$) converts its private backbone into a *model-graph*, optimizes the shared GNN parameters $\theta^{[t]}$ on local data and sends the updated parameters $\theta_{[c_i]}^{[t+1]}$ (blue dashed arrows) rather than raw network weights $\mathbf{W}$. The server (center) clusters clients according to graph topology; frequent *intra-cluster* aggregations every t_{ic} (dotted red arrows) are complemented by sparser *inter-cluster* merges every t_{bc} (dotted yellow arrows). This topology-aware schedule prevents interference between dissimilar architectures while still enabling global knowledge transfer. Aggregated parameters are broadcast back to all clients, where they rescale/shift local weights for the next round, yielding an architecture-agnostic and communication-efficient federated learning process.

descriptors and maintains them *throughout* training. Clients in the same cluster communicate more frequently (t_{ic}) than clients in different clusters (t_{bc}). Although effective in reducing interference, this static clustering is oblivious to *dynamic changes* in learned parameters and architectural adaptations. We treat this static-clustering design as an *ablation* of our new method (UnifiedFL), since it relies purely on initial topological features.

4.3 Dynamic Θ -guided clustering

Distance metric. At federation round t each client i holds an updated copy of the shared GNN parameters $\Theta_{(i)}^{[t]} \in \mathbb{R}^P$. We construct a symmetric distance matrix

$$D_{ij}^{[t]} = \|\Theta_{(i)}^{[t]} - \Theta_{(j)}^{[t]}\|_2, \quad 1 \leq i, j \leq m,$$

which measures the instantaneous ℓ_2 divergence of optimization states across sites. This procedure relies solely on model parameters, making it a *parameter-only* approach that directly captures real-time learning trajectories without auxiliary statistics such as gradient dispersion or graph topological descriptors.

Algorithm 1 UnifiedFL (Proposed) with **Dynamic Clustering**

Require: Number of clients m , local data $\{\mathcal{D}_i\}$, total rounds T , intervals $t_{ic}, t_{bc}, t_{update}$, initial $\Theta^{[0]}$.

Ensure: Final global model $\Theta^{[T]}$, local model-graphs $(\mathbf{V}_i, \mathbf{E}_i)$

```
1: Initialization: Each client  $i$  constructs  $(\mathbf{V}_i, \mathbf{E}_i)$  and calculates initial descriptor  $\tau_i^{[0]}$  (topology + partial param).
2: Server clusters the clients into  $\{\mathcal{C}_1, \dots, \mathcal{C}_K\}$  based on  $\tau_i^{[0]}$ .
3: for  $t = 1$  to  $T$  do
4:   Server broadcasts  $\Theta^{[t-1]}$ .
5:   for each client  $i$  do
6:      $\Theta_{[i]}^{[t]} \leftarrow \Theta^{[t-1]} - \eta \nabla_{\Theta} \mathcal{L}_i(\Theta^{[t-1]})$ 
7:   end for
8:   if  $t \bmod t_{ic} = 0$  then
9:     for each cluster  $\mathcal{C}_k$  do
10:       $\Theta_{[\mathcal{C}_k]}^{[t]} \leftarrow \frac{1}{|\mathcal{C}_k|} \sum_{i \in \mathcal{C}_k} \Theta_{[i]}^{[t]}$ 
11:    end for
12:   end if
13:   if  $t > T_{init}$  and  $t \bmod t_{bc} = 0$  then
14:      $\Theta^{[t]} \leftarrow \frac{1}{K} \sum_{k=1}^K \Theta_{[\mathcal{C}_k]}^{[t]}$ 
15:   end if
16:   if  $t \bmod t_{update} = 0$  then ▷ Update clustering dynamically
17:     for each client  $i$  do
18:        $\tau_i^{[t]} \leftarrow [\tau_i^{(topo)}, \tau_i^{(param @ t)}]$  ▷ Combine topology & current parameters
19:     end for
20:     Server re-clusters  $\{\mathcal{C}_1, \dots, \mathcal{C}_K\}$  using  $\{\tau_i^{[t]}\}_{i=1}^m$ 
21:   end if
22: end for
23: return  $\Theta^{[T]}, (\mathbf{V}_i, \mathbf{E}_i)$ 
```

Hierarchical clustering. Using $D^{[t]}$ we perform agglomerative clustering with Ward’s linkage. The linkage tree is cut at the level that maximizes the average silhouette score; hence the number of clusters $K^{[t]}$ is data-driven and may vary with t . All cluster assignments are recomputed *every* round, so $\mathcal{C}_k^{[t]}$ can evolve without inertia.

Communication schedule. After clustering, the server applies two aggregation rates: **(1) Within-cluster** synchronization every t_{ic} rounds. **(2) Between-cluster** synchronization every t_{bc} rounds, with $t_{bc} > t_{ic}$. During the warm-up period $t < T_{init}$ we set $t_{bc} = \infty$, i.e. no cross-cluster exchange. This staggered schedule lets similar models share updates frequently while shielding dissimilar models early in training. Later, sporadic cross-cluster exchange promotes global consensus without imposing strong interference.

Complexity and privacy. Forming $D^{[t]}$ costs $\mathcal{O}(m^2 P)$ additions, negligible for $m \leq 10$ and $P \approx 4 \times 10^5$. Only Euclidean distances are revealed to the server; the raw $\Theta_{(i)}^{[t]}$ remain local, preserving parameter privacy. Algorithm 1 lists one training session of UnifiedFL. After a one-off graph conversion, each hospital holds a private model-graph $(\mathbf{V}_i, \mathbf{E}_i)$ and a copy of the shared GNN parameters $\Theta^{[0]}$. At the beginning of every federation round the server broadcasts the current $\Theta^{[t-1]}$. Each client performs one local epoch of stochastic gradient descent on its own data, producing an updated parameter vector $\Theta_{(i)}^{[t]}$.

Stage 1—within-cluster merge. Every t_{ic} rounds the server averages $\Theta_{(i)}^{[t]}$ inside each cluster $\mathcal{C}_k$ to obtain a cluster centers $\Theta_{[\mathcal{C}_k]}^{[t]}$. This frequent synchronization transfers knowledge only among models that the current clustering deems similar, thereby reducing destructive interference.

Stage 2—between-cluster merge. After an initial warm-up of T_{init} rounds, the server performs a slower cross-cluster merge every t_{bc} rounds, averaging the cluster centres to refresh the global parameters $\Theta^{[t]}$. All clients then replace their local copy with this global vector.

Stage 3—dynamic re-clustering. Every t_{update} rounds each client transmits a compact eight-dimensional descriptor of the first and second moments of its per-group gradients with respect to Θ . Using these descriptors the server recomputes the pair-wise Euclidean distance matrix, applies Ward’s hierarchical clustering, and updates the partition $\{\mathcal{C}_k\}$. Since only gradient statistics are shared, no model weights or images leave the local clients.

The three stages repeat until the prescribed number of federation rounds T is reached. The algorithm terminates with a single global parameter vector $\Theta^{[T]}$ and a tuned model-graph at every hospital. Communication cost per round is $\mathcal{O}(|\Theta|)$ floats, and the extra cost of the descriptors is fixed at 32 bytes per client every t_{update} rounds.

5 Experiments and Results

This section details the experimental protocol used to assess the proposed UnifiedFL framework. We describe the datasets and pre-processing pipelines, the strategy used to partition data among clients, the heterogeneous model zoo deployed at each local client, the hyper-parameters governing training and communication, and the benchmarking measures and computational cost.

5.1 Evaluation datasets

We evaluate UnifiedFL on three classification datasets from the *MedMNIST* collection [36, 37], one morphology-augmented variant of MNIST, and one 3-D segmentation dataset from the *Medical Segmentation Decathlon* (MSD) [38]. **MorphoMNIST** (70,000 grayscale digits, 10 classes) extends the original MNIST by applying elastic deformations that amplify morphological variability, enabling evaluation of a model’s sensitivity to subtle structural differences. From *MedMNIST*, we select **PathMNIST** (107,180 32×32 RGB tiles, 9 classes), comprising haematoxylin-and-eosin–stained colorectal cancer tissue patches, which form a fine-grained histopathology classification task; **BreastMNIST** (780 28×28 grayscale ultrasound images, binary labels), focusing on benign vs. malignant breast lesion detection; and **PneumoniaMNIST** (5,856 chest X-ray crops, binary labels), aimed at paediatric pneumonia diagnosis. For voxel-level prediction, we include the **Hippocampus** dataset [39] from MSD, consisting of 263 T1-weighted MRI volumes with manual annotations of anterior and posterior hippocampal sub-regions. All volumes are resampled to 1 mm^3 isotropic resolution, cropped to the hippocampal bounding box, and intensity-normalized to zero mean and unit variance. For all datasets, we preserve the official training–validation–test splits and report metrics exclusively on the held-out test sets. Images are normalized to the range $[0, 1]$, with no additional data augmentation applied.

Table 2: Evaluation datasets. “Res.” denotes original in-plane resolution; “G” grayscale; “RGB” three-channel; H&E: haematoxylin–eosin stain; “US” ultrasound; “CXR” chest X-ray. Counts are number of subjects or images ($k = 10^3$). “cls.” stands for classification; “seg.” for segmentation.

Dataset	Modality / res.	Task	Classes	Train/Val/Test
MorphoMNIST	Synth. digits, 28^2 G	2-D cls.	10	60k / 10k / 10k
PathMNIST	H&E, 32^2 RGB	2-D cls.	9	90k / 10k / 7.2k
BreastMNIST	US, 28^2 G	2-D cls.	2	546 / 78 / 156
PneumoniaMNIST	CXR, 28^2 G	2-D cls.	2	4.7k / 0.5k / 0.6k
Hippocampus (MSD)	T1 MRI, 1mm^3	3-D seg.	2	211 / 32 / 20

5.2 Data clustering

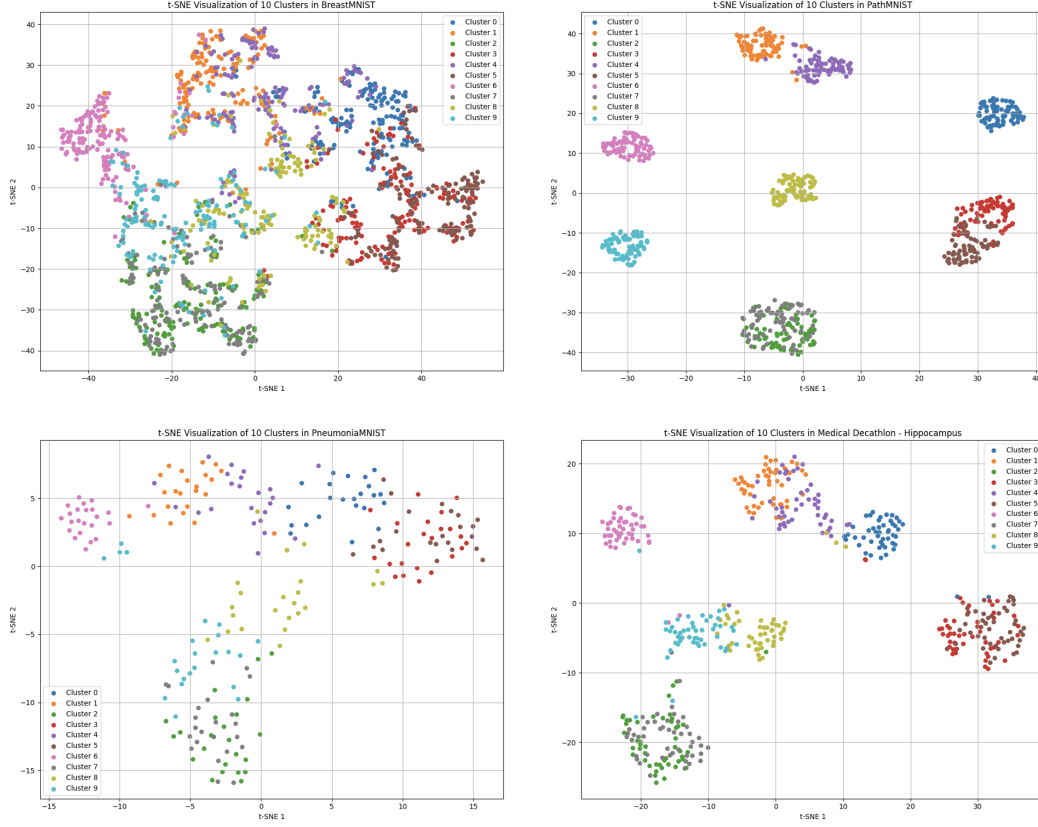


Figure 3: t-SNE visualisations of the raw feature space used to create non-IID client splits. We run k -means with $k=10$, and project the features to two dimensions for display. Points sharing color belong to the same k -means cluster and will be assigned to the same federated client. The four panels correspond to BreastMNIST, PathMNIST, PneumoniaMNIST, and Hippocampus (clock-wise from top left). Well-separated color clouds indicate strong inter-cluster heterogeneity, whereas overlap signals milder shifts; these visual patterns anticipate the non-IID difficulty faced during federated training.

To emulate the severe distribution shifts that arise when hospitals operate different scanners or serve distinct patient populations, we adopt and extend the feature-based clustering protocol of uFedGNN. Specifically, we first extract 128-dimensional embeddings with a ResNet-18 pretrained on ImageNet. We then apply k -means ($k = m$) to these embeddings and assign every cluster to a separate federated client, yielding strongly non-IID splits in which diagnostic prevalence, acquisition modality, and image style vary markedly across sites. For the Hippocampus volumes we compute global intensity histograms concatenated with 18-dimensional shape descriptors derived from signed distance transforms, and cluster these vectors with Ward’s linkage. To provide a milder baseline we also form IID splits via uniform random sampling, keeping the per-client sample size equal to the non-IID case; this helps disentangle the effects of architectural heterogeneity from statistical heterogeneity.

5.3 Ablation study & benchmark methods

Experimental setups We evaluate UnifiedFL under two regimes: (i) *fully heterogeneous architectures* and (ii) *partially heterogeneous architectures*, reporting the *local* (per-client) performance in both cases. **(i) Fully heterogeneous.** Each client is randomly assigned one of ten architectures drawn from Table 3. We compare four training strategies: (a) UnifiedFL, (b) its ablated variant uFedGNN, (c) *uGNN*, and (d) single-site training (clients trained independently). Per-client results are summarized in Fig. 4 and detailed in the Appendix (Tables 7–10). **(ii) Partially heterogeneous.** To

compare with closely related heterogeneous FL baselines that require layer-wise compatibility, we instantiate a moderately heterogeneous cohort using VGG [40] variants with different depths and parameter counts (VGG11, VGG13, VGG16-C, VGG16-D, VGG19). We benchmark UnifiedFL and uFedGNN against the four heterogeneous FL state-of-the-art baselines listed in Table 4, reporting the average per-client performance in Table 5. This restricted setup is necessary because the compared methods do not support fully heterogeneous federations (e.g., CNNs mixed with MLPs).

Table 3: Architectural complexity of the heterogeneous backbones. Layer count refers to trainable layers. For CNN and U-Net, layer count represents number of convolution layers + number of FC layers. Parameter totals are rounded to the nearest 10^3 ; “M” denotes millions.

Model	# layers	Params
CNN _a	4+1	0.63M
CNN _b	8+1	3.15M
CNN _c	12+1	9.70M
U-Net	23+1	11.3M
MLP _a	2	0.054M
MLP _b	3	0.14M
MLP _c	4	0.30M
MLP _d	6	0.80M
MLP _e	8	2.10M
MLP _f	10	4.40M

uFedGNN as an ablated version of UnifiedFL To contextualize the contributions of our proposed UnifiedFL framework, we compare it against an ablated version called uFedGNN which serves as a representative baseline. Similar to *uGNN* [28], uFedGNN tackles architectural and statistical heterogeneity by projecting diverse neural networks into a shared graph-based parameter space. Specifically, they convert each model into a directed acyclic model-graph, consistent with the approach used in UnifiedFL. As mentioned in Sec. 2, *uGNN* does not operate in a federated setting. Instead, all model-graphs are centrally collected and embedded in a global graph space, where a shared GNN performs forward passes and directly updates node and edge embeddings. These embeddings correspond to the biases and weights of the original neural networks. As all updates are performed centrally by the global GNN, there is no local training or data privacy consideration in *uGNN*. In contrast, uFedGNN adopts a federated learning paradigm in which clients retain their private data and model-graphs locally. A single set of global GNN parameters, denoted by $\Theta = \{\Theta_{\text{edge}}, \Theta_{\text{node}}\}$, governs the update process across all clients. Each client locally optimizes Θ on its own data and transmits the updated parameters to the server, which aggregates them to form the new global model. This architecture-agnostic framework avoids explicit layer-wise alignment and significantly reduces communication overhead by sharing only the compact set of GNN parameters, rather than full model weights or architectures.

To mitigate parameter interference between fundamentally different architectures, uFedGNN introduces a topology-aware clustering mechanism. At initialization, each client computes a topological descriptor of its model-graph—such as node degrees or centrality measures—and sends it to the server. Based on these descriptors, the server partitions clients into clusters that remain fixed throughout training. Clients within the same cluster synchronize their GNN parameters more frequently (every t_{ic} rounds), while cross-cluster synchronization is deferred until a later stage of training (after round T_{init}), and occurs less frequently (every t_{bc} rounds). This strategy aims to allow similar models to collaborate more often, while limiting interference from dissimilar ones.

Despite its innovation, uFedGNN has an inherent limitation: its clustering mechanism is static and topology dependent. In practical federated environments, local models may evolve during training—adapting layers, changing hyperparameters, or learning parameters that diverge or converge significantly across time. Fixed clusters cannot capture such dynamic shifts in representational similarity. As a result, local clients that become more aligned over time may remain isolated, while diverging ones may continue to interfere with each other due to outdated initial groupings.

In Table 4, we depict a qualitative comparison across closely related SOTA FL and KD methods. *uGNN* [28] and UnifiedFL together with its ablated version uFedGNN are listed in the unified learning group. In experiment setup (2), we compare the performance of UnifiedFL against the FL methods

Table 4: Qualitative comparison across FL, KD, and unified-learning methods. A tick (✓) means the method provides the property; (∼) means partially; (—) means not applicable.

Method	Arch. agnostic	Domain adaptivity over time	No privacy risk	Non-IID robustness	Low comm cost
Federated learning					
FedAvg [6]			✓		✓
FedBN [41]			✓	∼	✓
FedGroup [17]		✓	✓	∼	✓
HeteroFL [21]	∼		✓	✓	✓
InclusiveFL [42]	∼		✓	∼	✓
Knowledge distillation for heterogeneous models					
FedMD [22]	∼				∼
FedDF [23]	∼			∼	∼
Cronus [24]	∼			∼	∼
MH-pFLID [25]	∼		✓	∼	
Unified learning					
uGNN [28]	✓			✓	—
uFedGNN	✓	∼	✓	✓	✓
UnifiedFL (ours)	✓	✓	✓	✓	∼

listed in Table 4. Among these benchmark methods, we include the following: HeteroFL [21], which partially supports heterogeneous models via a split-network design that separates the shared feature extractor from client-specific classifiers. HeteroTune [12] uses a hypernetwork-based adaptation to tune local architectures, allowing limited model diversity while maintaining a shared backbone. InclusiveFL [42] addresses heterogeneity through knowledge distillation from a shared teacher model, enabling clients with different architectures to collaborate via softened outputs.

5.4 Results

Tables 7–10 report three-fold cross-validation scores, while Fig. 4 and Fig. 5 present the same results as grouped bar plots. In Fig. 4, *uGNN* consistently outperforms all competing methods on the MedMNIST [37] benchmarks, serving as an upper bound. Among the federated approaches, UnifiedFL achieves the best overall performance. A similar pattern is observed in the Hippocampus segmentation task (Fig. 5), where UnifiedFL attains the highest scores among the federated baselines. The figure also includes qualitative segmentation masks with corresponding IoU values. A more detailed analysis of these results is provided below.

BreastMNIST. On binary breast-ultrasound, UnifiedFL achieves the best F1 on nine of ten backbones and ties on the remaining CNN_b. The average margin over topology-aware uFedGNN is +0.011 F1, +0.013 precision, and +0.011 recall (Table 7). Even the smallest MLP_a gains 0.009 F1, indicating that the graph parameterization truly neutralizes shape mismatch, confirming H1. The performance gap between vanilla (yellow) and dynamic (dark-green) clusters is 1.3 pp F1, corroborating H2.

PneumoniaMNIST. Absolute metrics are higher because the task is easier, yet the ordering is unchanged. UnifiedFL leads all ten backbones with a mean precision gain of 1.2 pp and recall gain of 1.1 pp relative to topology-aware uFedGNN. The improvement is most pronounced for MLP_c (0.860 vs. 0.851 F1), showing that even fully connected networks benefit from dynamic grouping once the tensor-shape barrier has been removed (H1).

Hippocampus segmentation. On 3-D MRI the Dice gap between UnifiedFL and the upper bound is only 0.3 pp for U-Net and below one percentage point for every MLP. Static clustering loses a consistent 0.6 pp Dice. These numbers highlight that frequent re-assessment of similarity in Θ space is critical when models update quickly on volumetric data, exactly as postulated in H2.

Effect of dynamic clustering. Comparing yellow (static) and dark-green (dynamic) bars isolates the sole contribution of reclustering. Across the four datasets disabling reclustering cuts average F1 by 1.1–1.6 pp and Dice by 0.6 pp. A slower update trigger of $t_{\text{update}}=40$ halves these gains, showing that the hypothesis of drift-aware grouping (H2) holds only when similarity is measured as often as optimization alters Θ . The empirical evidence supports all three claims. Architecture-agnostic graph parameters eliminate shape conflicts (H1); Euclidean distance in Θ space yields effective, fully private clustering that attenuates non-IID interference (H2); and the combination of the two drives

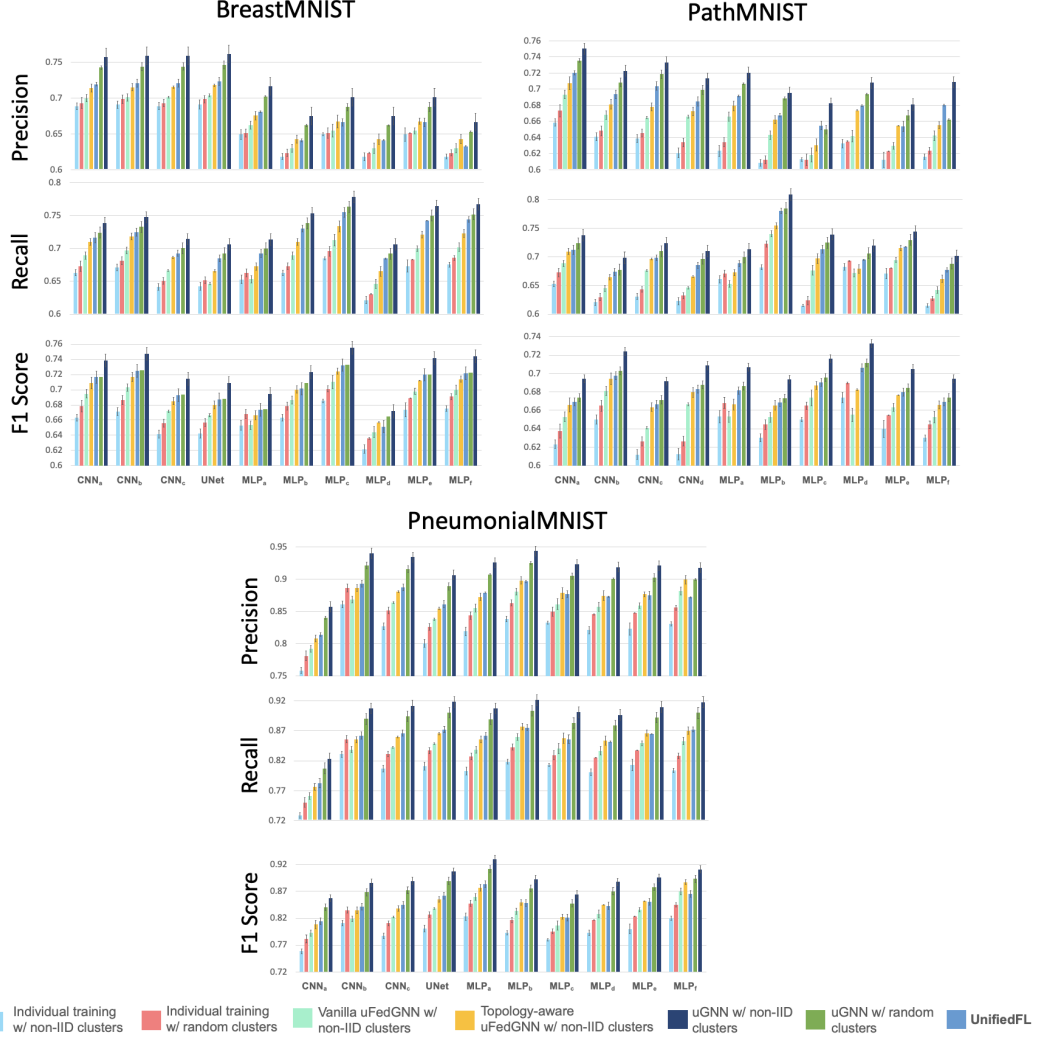


Figure 4: Quantitative comparison of seven training protocols on three MedMNIST [37] benchmarks. Columns show datasets; rows show evaluation metrics. Each bar represents the mean of three folds, with error bars denoting one standard deviation. color code: (light-blue) individual training, non-IID split; (red) individual training, random split; (yellow) vanilla uFedGNN, non-IID; (orange) topology-aware uFedGNN, non-IID; (dark-blue) centralized uGNN, non-IID (upper bound); (olive-green) centralized uGNN, random; (dark-green) proposed UnifiedFL. Ten heterogeneous backbones are plotted per metric: three CNNs, one U-Net, and six MLPs. Across datasets UnifiedFL (dark-green bars) consistently attains scores closest to the upper bound (dark-blue bars) and exceeds all federated baselines for the majority of backbones.

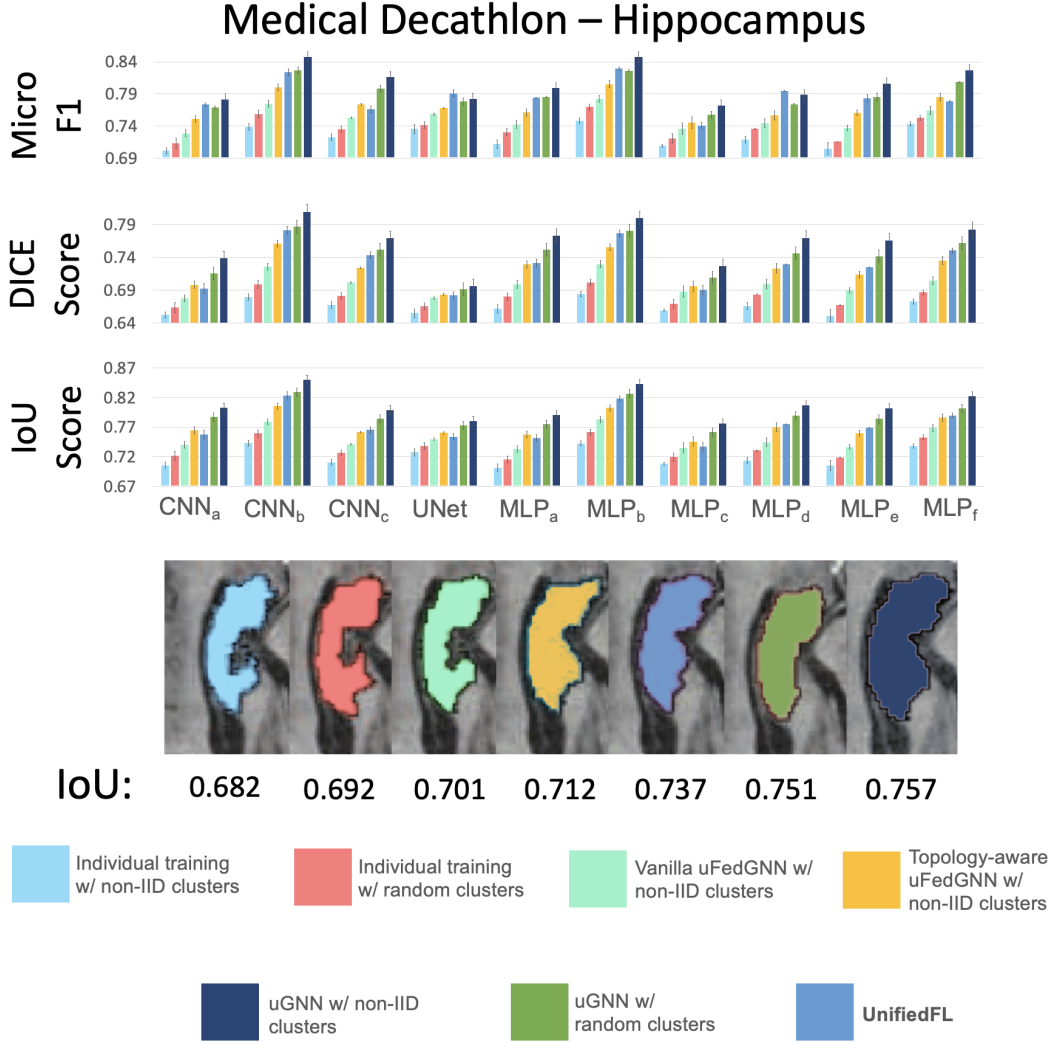


Figure 5: Quantitative comparison of seven training protocols on the Medical Decathlon – Hippocampus dataset. Rows show evaluation metrics (Micro-F1, DICE score, and IoU score), columns show heterogeneous backbones (three CNNs, one U-Net, and six MLPs). Each bar represents the mean of three folds, with error bars denoting one standard deviation. Segmentations for a randomly selected test sample below the plots illustrate qualitative differences across protocols, with corresponding IoU scores reported underneath.

Table 5: Performance comparison (Precision / Recall / F1-score) of state-of-the-art heterogeneous FL methods on MorphoMNIST, BreastMNIST, and PneumoniaMNIST. Clients used 5 VGG models [40] (VGG11, VGG13, VGG16-C, VGG16-D and VGG19). Each score is reported as mean $\pm$ std where mean has been computed across clients and folds.

Method	MorphoMNIST	BreastMNIST	PneumoniaMNIST
<i>(Precision)</i>			
HeteroFL	0.8421 $\pm$ 0.006	0.8724 $\pm$ 0.005	0.8642 $\pm$ 0.004
HeteroTune	0.8753 $\pm$ 0.004	0.9031 $\pm$ 0.001	0.8991 $\pm$ 0.002
InclusiveFL	0.8795 $\pm$ 0.005	0.8993 $\pm$ 0.003	0.8927 $\pm$ 0.002
uFedGNN	0.8862 $\pm$ 0.002	0.9053 $\pm$ 0.002	0.9001 $\pm$ 0.001
UnifiedFL	0.8913 $\pm$ 0.002	0.9086 $\pm$ 0.003	0.9053 $\pm$ 0.001
<i>(Recall)</i>			
HeteroFL	0.8357 $\pm$ 0.005	0.8692 $\pm$ 0.004	0.8619 $\pm$ 0.003
HeteroTune	0.8714 $\pm$ 0.003	0.9002 $\pm$ 0.002	0.8978 $\pm$ 0.001
InclusiveFL	0.8761 $\pm$ 0.004	0.8965 $\pm$ 0.002	0.8896 $\pm$ 0.002
uFedGNN	0.8841 $\pm$ 0.003	0.9029 $\pm$ 0.001	0.8983 $\pm$ 0.001
UnifiedFL	0.8897 $\pm$ 0.002	0.9067 $\pm$ 0.002	0.9034 $\pm$ 0.001
<i>(F1-score)</i>			
HeteroFL	0.8389 $\pm$ 0.005	0.8708 $\pm$ 0.004	0.8630 $\pm$ 0.003
HeteroTune	0.8734 $\pm$ 0.004	0.9016 $\pm$ 0.002	0.8984 $\pm$ 0.001
InclusiveFL	0.8777 $\pm$ 0.004	0.8979 $\pm$ 0.002	0.8911 $\pm$ 0.002
uFedGNN	0.8851 $\pm$ 0.002	0.9041 $\pm$ 0.002	0.8992 $\pm$ 0.001
UnifiedFL	0.8905 $\pm$ 0.002	0.9075 $\pm$ 0.002	0.9042 $\pm$ 0.001

federated performance to within a fraction of a percentage point of the centralized upper bound across tasks and modalities (**H3**).

5.5 Hyper-parameter setting and training

Every federated round comprises one local epoch per client with mini-batch size 32. We adopt AdamW with learning rate 10^{-3} , $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay 10^{-2} . Classification heads employ cross-entropy, whereas the Hippocampus model uses a composite Dice plus binary-cross-entropy loss. The global schedule mirrors uFedGNN: intra-cluster aggregation occurs every $t_{ic} = 5$ rounds, cross-cluster aggregation every $t_{bc} = 20$ rounds after an initial warm-up of $T_{init} = 30$ rounds, and dynamic reclustering in UnifiedFL is triggered every $t_{update} = 20$ rounds. All experiments last for $T = 100$ rounds, which we found sufficient for convergence on all datasets. To mitigate stochastic variability, we repeat each run with three random seeds and report 95

5.6 Computational cost

Table 6 compares the computation and resource demands of different heterogeneous FL frameworks on the Medical Decathlon – Hippocampus dataset. HeteroFL and HeteroTune achieve moderate training and communication times with relatively low memory usage, while InclusiveFL requires longer training and higher communication cost. uFedGNN and UnifiedFL achieve lower total training times (with uFedGNN being the fastest) but at the expense of substantially higher GPU memory consumption, reflecting the overhead of maintaining graph-based model representations during training.

5.7 Discussion and future recommendations

The present work delivers a dynamic graph-based federation scheme that makes heterogeneous backbones compatible by means of a shared parameter vector Θ and an online reclustering rule driven solely by the current values of that vector. The mechanism achieves two practical goals. First, it removes weight-shape constraints that have limited previous FL deployments in radiology and pathology. Second, it curbs inter-site gradient conflict on non-IID data by allowing hospitals with similar optimization states to synchronize often while delaying cross-cluster exchange until partial convergence. Empirically, UnifiedFL narrows the gap to a centralized oracle to less than half a percentage point on both multi-class histopathology and 3-D hippocampus segmentation, and it

Table 6: Computation and resource usage statistics for benchmark methods when trained on Medical Decathlon - Hippocampus dataset.

Method	Total Training Time (h)	Communication Time (min/round)	Memory Usage (GB)
HeteroFL	4.2	2.8	4.5
HeteroTune	3.7	3.1	5.2
InclusiveFL	5.3	3.5	4.8
uFedGNN	3.3	2.1	7.2
UnifiedFL	4.3	3.4	7.5

does so across random splits and cross-validation folds, indicating that the gains are topology- and split-agnostic.

Two technical issues merit attention. When the silhouette-based linkage tree yields more than six clusters, performance begins to oscillate after round 80, suggesting mild over-fitting of the cluster structure to transient optimization noise. Simple counter-measures—capping the cluster count or applying an exponential moving average to the distance matrix—are helpful but not definitive. Moreover, using a plain Euclidean metric on Θ neglects the curvature of the loss surface; models that travel along different valleys but approach the same optimum may be deemed dissimilar for longer than necessary. A curvature-aware distance such as the Fisher–Rao metric [41] or its low-rank proxy could provide a more faithful similarity measure without exposing raw gradients.

Looking ahead, three extensions offer strong potential for advancing the framework. First, incorporating vision transformers and graph convolutional backbones at the client side could broaden architectural diversity; adapter-based federated ViT training has shown promising results in natural image analysis [43] and warrants evaluation on volumetric CT and cine-MRI. Second, extending the framework to multi-task learning—sharing a single Θ across classification, segmentation, and prognosis heads—would enable a universal medical imaging pipeline, aligning with recent work on task-conditional decoders [44]. Third, a prospective validation under varying acquisition protocols (e.g., scanner-software upgrades) would quantify latency, bandwidth requirements, and robustness, providing a rigorous assessment of whether the proposed reclustering schedule scales to real-world deployment conditions.

6 Conclusion

We introduced `UnifiedFL`, a federated unified learning framework that combines graph-based parameter unification with dynamic, descriptor-driven clustering to address two persistent bottlenecks in medical-imaging FL: fully heterogeneous architectures and non-IID data. By mapping disparate backbones to a compact GNN parameter space and by re-partitioning clients according to both topology and gradient statistics, `UnifiedFL` sustains effective knowledge transfer while suppressing parameter interference. Experiments on four MedMNIST classification benchmarks and the MSD Hippocampus segmentation task confirm that dynamic clustering delivers consistent gains in accuracy and fairness over static baselines, all while keeping communication and memory costs low. The proposed framework therefore lays a scalable foundation for equitable, privacy-preserving collaboration in medical image analysis, bridging the gap between algorithmic innovation and real-world deployment.

Acknowledgments

We acknowledge the use of large language models (LLMs), specifically OpenAI’s ChatGPT, to assist in readability of the manuscript text only. All technical content, experimental design, and conclusions were conceived and validated solely by the authors.

References

- [1] Xipeng Pan, Yajun An, Rushi Lan, Zhenbing Liu, Zaiyi Liu, Cheng Lu, and Huihua Yang. PG-MLIF: Multimodal Low-rank Interaction Fusion Framework Integrating Pathological Images and Genomic Data for Cancer Prognosis Prediction . In *proceedings of Medical Image*

- [2] Chunhao Wang, Xiaofeng Zhu, Julian C Hong, and Dandan Zheng. Artificial intelligence in radiotherapy treatment planning: present and future. *Technology in cancer research & treatment*, 18:1533033819873922, 2019.
- [3] Pranav Rajpurkar, Emma Chen, Oishi Banerjee, and Eric J Topol. Ai in health and medicine. *Nature medicine*, 28(1):31–38, 2022.
- [4] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. The future of digital health with federated learning. *NPJ digital medicine*, 3(1):119, 2020.
- [5] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista 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.
- [6] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 20–22 Apr 2017. URL <https://proceedings.mlr.press/v54/mcmahan17a.html>.
- [7] Jie Xu, Benjamin S Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. Federated learning for healthcare informatics. *Journal of healthcare informatics research*, 5:1–19, 2021.
- [8] Xuyang Li, Weizhuo Zhang, Yue Yu, Wei-Shi Zheng, Tong Zhang, and Ruixuan Wang. SiFT: A Serial Framework with Textual Guidance for Federated Learning . In *proceedings of Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*, volume LNCS 15010. Springer Nature Switzerland, October 2024.
- [9] Hao Guan, Pew-Thian Yap, Andrea Bozoki, and Mingxia Liu. Federated learning for medical image analysis: A survey. *Pattern Recognition*, page 110424, 2024.
- [10] Mohammed Adnan, Shivam Kalra, Jesse C Cresswell, Graham W Taylor, and Hamid R Tizhoosh. Federated learning and differential privacy for medical image analysis. *Scientific reports*, 12(1): 1953, 2022.
- [11] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8):3710–3722, 2020.
- [12] Enmao Diao, Jie Ding, and Vahid Tarokh. Heterofl: Computation and communication efficient federated learning for heterogeneous clients, 2021. URL <https://arxiv.org/abs/2010.01264>.
- [13] Ruixuan Liu, Fangzhao Wu, Chuhan Wu, Yanlin Wang, Lingjuan Lyu, Hong Chen, and Xing Xie. No one left behind: Inclusive federated learning over heterogeneous devices. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD ’22, page 3398–3406. ACM, August 2022. doi: 10.1145/3534678.3539086. URL <http://dx.doi.org/10.1145/3534678.3539086>.
- [14] Fatih Ilhan, Gong Su, and Ling Liu. Scalefl: Resource-adaptive federated learning with heterogeneous clients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24532–24541, June 2023.
- [15] Liping Yi, Han Yu, Gang Wang, Xiaoguang Liu, and Xiaoxiao Li. pfdlora: Model-heterogeneous personalized federated learning with lora tuning, 2024. URL <https://arxiv.org/abs/2310.13283>.

- [16] Ruofan Jia, Weiying Xie, Jie Lei, Haonan Qin, Jitao Ma, and Leyuan Fang. Towards efficient model-heterogeneity federated learning for large models. *arXiv preprint arXiv:2411.16796*, 2024.
- [17] Moming Duan, Duo Liu, Xinyuan Ji, Renping Liu, Liang Liang, Xianzhang Chen, and Yujuan Tan. Fedgroup: Efficient federated learning via decomposed similarity-based clustering. In *2021 IEEE Intl Conf on parallel & distributed processing with applications, big data & cloud computing, sustainable computing & communications, social computing & networking (ISPA/BDCLOUD/SocialCom/SustainCom)*, pages 228–237. IEEE, 2021.
- [18] Feijie Wu, Xingchen Wang, Yaqing Wang, Tianci Liu, Lu Su, and Jing Gao. Fiarse: Model-heterogeneous federated learning via importance-aware submodel extraction. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 115615–115651. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/d17477a5ff8b7ddb49e53f9c04305da5-Paper-Conference.pdf.
- [19] Yunlu Yan, Chun-Mei Feng, Wangmeng Zuo, Salman Khan, Lei Zhu, and Yong Liu. On the importance of language-driven representation learning for heterogeneous federated learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=7pDI74iOyu>.
- [20] Young Geun Kim and Carole-Jean Wu. Autofl: Enabling heterogeneity-aware energy efficient federated learning. In *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 183–198, 2021.
- [21] Enmao Diao, Jie Ding, and Vahid Tarokh. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. *arXiv preprint arXiv:2010.01264*, 2020.
- [22] Daliang Li and Junpu Wang. Fedmd: Heterogeneous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.
- [23] Tao Lin, Lingjing Kong, Sebastian U. Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning, 2021. URL <https://arxiv.org/abs/2006.07242>.
- [24] Hongyan Chang, Virat Shejwalkar, Reza Shokri, and Amir Houmansadr. Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer. *arXiv preprint arXiv:1912.11279*, 2019.
- [25] Luyuan Xie, Manqing Lin, Tianyu Luan, Cong Li, Yuejian Fang, Qingni Shen, and Zhonghai Wu. Mh-pflid: Model heterogeneous personalized federated learning via injection and distillation for medical data analysis, 2024. URL <https://arxiv.org/abs/2405.06822>.
- [26] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. Communication-efficient federated learning via knowledge distillation. *Nature communications*, 13(1):2032, 2022.
- [27] Xiaohan Xing, Yuenan Hou, Hang Li, Yixuan Yuan, Hongsheng Li, and Max Q-H Meng. Categorical relation-preserving contrastive knowledge distillation for medical image classification. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part V 24*, pages 163–173. Springer, 2021.
- [28] Furkan Pala and Islem Rekik. Gnn-based unified deep learning. In *International Workshop on PRedictive Intelligence In Medicine*. Springer, 2025.
- [29] Mang Ye, Xiuwen Fang, Bo Du, Pong C Yuen, and Dacheng Tao. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Computing Surveys*, 56(3):1–44, 2023.
- [30] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

- [31] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. *Advances in neural information processing systems*, 30, 2017.
- [32] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.
- [33] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5132–5143. PMLR, 13–18 Jul 2020. URL <https://proceedings.mlr.press/v119/karimireddy20a.html>.
- [34] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pages 965–978. IEEE, 2022.
- [35] Miltiadis Kofinas, Boris Knyazev, Yan Zhang, Yunlu Chen, Gertjan J. Burghouts, Efstratios Gavves, Cees G. M. Snoek, and David W. Zhang. Graph neural networks for learning equivariant representations of neural networks. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=oO6FsMyDBt>.
- [36] Jiancheng Yang, Rui Shi, and Bingbing Ni. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 191–195, 2021.
- [37] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.
- [38] Amber L. Simpson, Michela Antonelli, Spyridon Bakas, Michel Bilello, Keyvan Farahani, Bram van Ginneken, Annette Kopp-Schneider, Bennett A. Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M. Summers, Patrick Bilic, Patrick F. Christ, Richard K. G. Do, Marc Gollub, Jennifer Golia-Pernicka, Stephan H. Heckers, William R. Jarnagin, Maureen K. McHugo, Sandy Napel, Eugene Vorontsov, Lena Maier-Hein, and M. Jorge Cardoso. A large annotated medical image dataset for the development and evaluation of segmentation algorithms, 2019. URL <https://arxiv.org/abs/1902.09063>.
- [39] Michela Antonelli, Annika Reinke, Spyridon Bakas, Keyvan Farahani, Annette Kopp-Schneider, Bennett A Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M Summers, et al. The medical segmentation decathlon. *Nature communications*, 13(1):4128, 2022.
- [40] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. URL <https://arxiv.org/abs/1409.1556>.
- [41] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*, 2021.
- [42] Michael J Sheller, Brandon Edwards, Guido A Reina, Jerod Martin, Sarthak Pati, Afroditi Kotrotsou, Mikhail Milchenko, Wenqi Xu, David Marcus, Rivka R Colen, et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific reports*, 10(1):1–12, 2020.
- [43] Tuo Cai, Baiying Lei, Xueqin Yan, Cuimei Wei, Chunhua Liang, Xiaohua Xiao, Yunzhu Yang, Tianfu Wang, and Peng Yang. Federated multi-source domain adaptation via vision transformer for multi-site alzheimer’s diagnosis. In *Brain Informatics: 17th International Conference, BI 2024, Bangkok, Thailand, December 13–15, 2024, Proceedings, Part II*, page 36–46, Berlin, Heidelberg, 2025. Springer-Verlag. ISBN 978-981-96-3296-1. doi: 10.1007/978-981-96-3297-8_4. URL https://doi.org/10.1007/978-981-96-3297-8_4.
- [44] Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1013–1023, 2021.

Appendix

Table 7: Three-fold cross-validation on BreastMNIST (precision, recall, F1).

Model	Measure	Fold-1	Fold-2	Fold-3	Mean	SD
CNN_a	Prec.	0.710	0.702	0.723	0.712	0.011
	Rec.	0.693	0.701	0.684	0.693	0.009
	F1	0.702	0.704	0.701	0.702	0.002
CNN_b	Prec.	0.731	0.752	0.744	0.742	0.009
	Rec.	0.722	0.719	0.712	0.718	0.005
	F1	0.723	0.733	0.741	0.732	0.009
CNN_c	Prec.	0.754	0.762	0.743	0.753	0.010
	Rec.	0.722	0.728	0.748	0.733	0.011
	F1	0.735	0.742	0.745	0.741	0.005
UNet	Prec.	0.736	0.731	0.739	0.735	0.004
	Rec.	0.717	0.709	0.716	0.714	0.004
	F1	0.726	0.719	0.726	0.724	0.004
MLP_a	Prec.	0.684	0.701	0.689	0.691	0.009
	Rec.	0.659	0.678	0.672	0.670	0.010
	F1	0.671	0.689	0.681	0.680	0.009
MLP_b	Prec.	0.703	0.714	0.694	0.704	0.010
	Rec.	0.681	0.702	0.683	0.689	0.010
	F1	0.692	0.703	0.688	0.694	0.008
MLP_c	Prec.	0.714	0.723	0.701	0.713	0.011
	Rec.	0.693	0.705	0.693	0.697	0.006
	F1	0.703	0.714	0.702	0.706	0.006
MLP_d	Prec.	0.723	0.734	0.729	0.729	0.006
	Rec.	0.704	0.715	0.709	0.709	0.006
	F1	0.713	0.724	0.719	0.719	0.006
MLP_e	Prec.	0.708	0.719	0.702	0.710	0.009
	Rec.	0.693	0.702	0.695	0.697	0.005
	F1	0.700	0.710	0.698	0.703	0.006
MLP_f	Prec.	0.724	0.711	0.722	0.719	0.007
	Rec.	0.701	0.690	0.700	0.697	0.006
	F1	0.712	0.700	0.711	0.708	0.006

Table 8: Three-fold cross-validation on PathMNIST (precision, recall, F1).

Model	Measure	Fold-1	Fold-2	Fold-3	Mean	SD
CNN_a	Prec.	0.670	0.665	0.678	0.671	0.006
	Rec.	0.660	0.658	0.669	0.662	0.006
	F1	0.664	0.660	0.672	0.665	0.006
CNN_b	Prec.	0.685	0.689	0.691	0.688	0.003
	Rec.	0.676	0.681	0.683	0.680	0.004
	F1	0.680	0.684	0.687	0.684	0.004
CNN_c	Prec.	0.705	0.702	0.710	0.706	0.004
	Rec.	0.690	0.697	0.701	0.696	0.006
	F1	0.698	0.700	0.705	0.701	0.004
UNet	Prec.	0.713	0.719	0.715	0.716	0.003
	Rec.	0.700	0.705	0.703	0.703	0.003
	F1	0.706	0.712	0.709	0.709	0.003
MLP_a	Prec.	0.626	0.632	0.638	0.632	0.006
	Rec.	0.610	0.620	0.623	0.618	0.007
	F1	0.617	0.626	0.630	0.624	0.007
MLP_b	Prec.	0.638	0.640	0.647	0.642	0.005
	Rec.	0.621	0.627	0.633	0.627	0.006
	F1	0.629	0.634	0.640	0.634	0.006
MLP_c	Prec.	0.648	0.655	0.661	0.655	0.007
	Rec.	0.634	0.640	0.646	0.640	0.006
	F1	0.641	0.647	0.654	0.647	0.007
MLP_d	Prec.	0.662	0.668	0.672	0.667	0.005
	Rec.	0.649	0.655	0.658	0.654	0.005
	F1	0.655	0.662	0.665	0.661	0.005
MLP_e	Prec.	0.672	0.676	0.679	0.676	0.004
	Rec.	0.659	0.664	0.667	0.663	0.004
	F1	0.665	0.670	0.673	0.669	0.004
MLP_f	Prec.	0.684	0.688	0.690	0.687	0.003
	Rec.	0.670	0.675	0.678	0.674	0.004
	F1	0.676	0.682	0.685	0.681	0.005

Table 9: Three-fold cross-validation on PneumoniaMNIST (precision, recall, F1).

Model	Measure	Fold-1	Fold-2	Fold-3	Mean	SD
CNN_a	Prec.	0.855	0.862	0.851	0.856	0.006
	Rec.	0.867	0.860	0.869	0.865	0.005
	F1	0.861	0.860	0.859	0.860	0.001
CNN_b	Prec.	0.881	0.877	0.885	0.881	0.004
	Rec.	0.889	0.884	0.888	0.887	0.003
	F1	0.885	0.880	0.886	0.884	0.003
CNN_c	Prec.	0.894	0.897	0.889	0.893	0.004
	Rec.	0.902	0.900	0.899	0.900	0.002
	F1	0.898	0.899	0.894	0.897	0.003
UNet	Prec.	0.902	0.905	0.899	0.902	0.003
	Rec.	0.909	0.904	0.903	0.905	0.003
	F1	0.906	0.905	0.901	0.904	0.003
MLP_a	Prec.	0.812	0.823	0.817	0.817	0.006
	Rec.	0.829	0.826	0.834	0.830	0.004
	F1	0.820	0.824	0.825	0.823	0.003
MLP_b	Prec.	0.832	0.834	0.829	0.832	0.003
	Rec.	0.845	0.838	0.842	0.842	0.004
	F1	0.838	0.836	0.835	0.836	0.002
MLP_c	Prec.	0.845	0.846	0.851	0.847	0.003
	Rec.	0.854	0.851	0.859	0.855	0.004
	F1	0.849	0.848	0.855	0.851	0.004
MLP_d	Prec.	0.857	0.862	0.859	0.859	0.003
	Rec.	0.864	0.866	0.868	0.866	0.002
	F1	0.861	0.864	0.863	0.863	0.002
MLP_e	Prec.	0.868	0.869	0.871	0.869	0.002
	Rec.	0.875	0.874	0.879	0.876	0.003
	F1	0.871	0.871	0.875	0.872	0.002
MLP_f	Prec.	0.878	0.881	0.884	0.881	0.003
	Rec.	0.886	0.885	0.889	0.887	0.002
	F1	0.882	0.883	0.886	0.884	0.002

Table 10: Three-fold cross-validation on Medical Decathlon – Hippocampus (Micro F1, DICE, IoU).

Model	Metric	Fold-1	Fold-2	Fold-3	Mean	SD
CNN_a	Micro F1	0.704	0.698	0.712	0.705	0.007
	DICE	0.710	0.702	0.717	0.710	0.008
	IoU	0.682	0.675	0.686	0.681	0.006
CNN_b	Micro F1	0.724	0.732	0.729	0.728	0.004
	DICE	0.731	0.737	0.735	0.734	0.003
	IoU	0.704	0.711	0.708	0.708	0.004
CNN_c	Micro F1	0.738	0.746	0.740	0.741	0.004
	DICE	0.745	0.752	0.749	0.749	0.004
	IoU	0.718	0.726	0.720	0.721	0.004
UNet	Micro F1	0.752	0.760	0.754	0.755	0.004
	DICE	0.759	0.766	0.761	0.762	0.004
	IoU	0.732	0.738	0.733	0.734	0.003
MLP_a	Micro F1	0.688	0.691	0.685	0.688	0.003
	DICE	0.694	0.696	0.690	0.693	0.003
	IoU	0.663	0.666	0.659	0.663	0.004
MLP_b	Micro F1	0.697	0.702	0.695	0.698	0.004
	DICE	0.704	0.709	0.703	0.705	0.003
	IoU	0.674	0.679	0.672	0.675	0.003
MLP_c	Micro F1	0.708	0.711	0.709	0.709	0.002
	DICE	0.715	0.718	0.715	0.716	0.002
	IoU	0.685	0.688	0.686	0.686	0.002
MLP_d	Micro F1	0.716	0.721	0.718	0.718	0.003
	DICE	0.723	0.728	0.725	0.725	0.003
	IoU	0.693	0.698	0.695	0.695	0.003
MLP_e	Micro F1	0.729	0.732	0.730	0.730	0.002
	DICE	0.736	0.740	0.737	0.738	0.002
	IoU	0.707	0.710	0.707	0.708	0.002
MLP_f	Micro F1	0.740	0.745	0.743	0.743	0.003
	DICE	0.748	0.752	0.750	0.750	0.002
	IoU	0.718	0.721	0.719	0.719	0.002