FEDSSP: SHARED-BRANCH STITCHING INTO PRIVATE MODELS FOR HETEROGENEOUS FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Heterogeneous Federated Learning (HtFL) has attracted increasing attention for its ability to support collaboration under both model and data heterogeneity. However, existing methods require visibility into private architectures, rely on public datasets, or introduce additional asymmetric knowledge transfer. To overcome these limitations, We propose FedSSP, which represents transferable global knowledge as a shared branch stitched into individual private models. Through a lightweight stitching layer, FedSSP jointly optimizes shared knowledge injection and local personalization in single-stage dual-path training, without a standalone proxy or separate distillation. FedSSP selects client-specific stitch positions by complementing intra-model KL-based block capacity with cross-entropy improvement to account for local task utility. For aggregation, shared-feature shifts and relative private-model updates serve as complementary contribution proxies, capturing representation changes and local adaptation. Extensive experiments on various datasets with eleven model architectures show that FedSSP consistently outperforms existing HtFL approaches, achieving an average accuracy improvement of 7.24%, without sharing any private model parameters or relying on public datasets.

1 INTRODUCTION

Federated Learning (FL) is a collaborative learning paradigm by allowing clients to train models without sharing raw data. By aggregating model updates on a central server, FL effectively mitigates the risk of privacy leakage (McMahan et al., 2017; Khan et al., 2021; Pei et al., 2024; Jia et al., 2025; Pramanik et al., 2026).

Most FL methods, however, assume that all clients hold the same model architecture and collaboratively train a shared global model (Karimireddy et al., 2020; Wang et al., 2020; Qu et al., 2025). Real-world deployments violate this assumption due to data, system, and model heterogeneity (Zhao et al., 2018; Ye et al., 2023; Fan et al., 2024). Specifically, clients' datasets are often highly Non-Independent and Identically Distributed (Non-IID), which leads to client drift and unstable convergence. Concurrently, differences in computation, memory, and communication budgets naturally result in diverse model scales and architectures. In cross-silo scenarios, beyond system constraints, intellectual property and commercial confidentiality concerns further prohibit participating institutions from disclosing or adapting their proprietary models (Liu et al., 2023; Yi et al., 2025b; Kang et al., 2026).

Heterogeneous federated learning (HtFL) (Wang et al., 2020; Diao et al., 2021) permits clients to retain distinct private models, but mismatched parameter spaces preclude direct averaging such as FedAvg (McMahan et al., 2017). Existing solutions includes aggregating common layers (Hu et al., 2026; Liu et al., 2024a; Wu et al., 2024a; Wang et al., 2023; Park & Joe-Wong, 2024), distilling knowledge through public or generated datasets (Li & Wang, 2019; Zhang et al., 2021; Huang et al., 2022; 2024; Luo et al., 2023; Zhu et al., 2021), exchanging class prototypes (Tan et al., 2022; Zhang et al., 2024; Wu et al., 2024b; Zhou et al., 2025; Jiang et al., 2025; Wang et al., 2026), and communicating through a shared proxy model or extractor (Wang et al., 2024; Qin et al., 2023; Xie et al., 2024; Yi et al., 2025a; 2023). The first three are respectively constrained by limited structural overlap, de-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

pendence on auxiliary transfer data, and coarse or unstable class summaries. Proxy-based methods mediate collaboration through a homogeneous auxiliary model or feature extractor, avoiding public dataset and architectural disclosure.

For proxy-based HtFL methods, the key limitation lies in the separation between the globally aggregated proxy and the private model that learns client-specific representations (Niu et al., 2026). Before aggregation, each client must distill or project its private knowledge into the proxy. Then, the resulting global knowledge must be transferred back to the private model after aggregation. As shown in Figure 1, this bidirectional, capacity-constrained translation creates an information bottleneck, especially when a compact proxy guides a stronger private model, and allows client-specific knowledge to influence global collaboration only indirectly.

This limitation motivates a different design principle: the shared component should remain homogeneous across clients for aggregation, yet be embedded in each private model's representation pathway and jointly optimized with the private model under local task supervision. Reconciling global homogeneity with local architectural diversity creates a representation-sitiching problem: the same shared component must connect to incompatible private feature spaces. Model stitching (Lenc & Vedaldi, 2015) provides a suitable mechanism by enabling a module to be connected to the intermediate representation flow of another model through a lightweight stitching layer. In this way, the shared component need not imitate the entire private predictor as a standalone proxy. Instead, it learns reusable semantics directly within the private model's representation context.

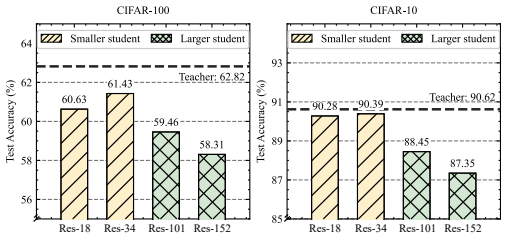


Figure 1: Asymmetric knowledge transfer from a pretrained ResNet-50 to heterogeneous models. Res-18/34/101/152 denote ResNet-18/34/101/152.

However, existing stitching methods (Bansal et al., 2021; Pan et al., 2023; Sanyal et al., 2026; Athanasiadis et al., 2026) mainly compose pretrained models with mature, stable representations and fine-tune the stitched model post hoc. In contrast, HtFL clients train from scratch under heterogeneous local data, leaving neither mature intermediate representations nor reliable source–target activation correspondences available a priori. For stitch selection, a large KL-based block-capacity score reflects substantial representation changes but not necessarily better local prediction, so relying on it alone may yield suboptimal results. Aggregation must also account for the different private models and stitch positions shaping shared-branch updates. These challenges raise three questions for adapting stitching to HtFL: **Q1: What should be stitched? Q2: Where should it be stitched? Q3: How should it be aggregated?**

We propose FedSSP, which encodes transferable global knowledge in a lightweight shared branch stitched into each private model. For **Q1**, single-stage dual-path training jointly optimizes the shared path for global knowledge injection and the private path for local personalization, without a standalone proxy or separate distillation. For **Q2**, FedSSP selects stitch positions using only the warmed-up private model, without requiring a pretrained shared branch. It complements intra-model KL-based block capacity with cross-entropy improvement, incorporating local task utility into personalized stitch selection. For **Q3**, shared-feature shifts capture changes in aggregated representations, while relative private-model updates measure the accompanying local adaptation. These complementary contribution proxies determine aggregation weights. Our main contributions are summarized as follows:

- We propose FedSSP, which embeds a lightweight shared branch directly into heterogeneous private prediction paths, enabling single-stage joint learning of globally shared and client-specific representations without a standalone proxy, separate distillation.
- We adapt stitch selection to HtFL by assessing representation transitions and local task utility using only warmed-up private models, and guide aggregation with complementary contribution proxies that capture shared-representation changes and private-model adaptation.
- Experiments across various datasets and models demonstrate an average accuracy gain of 7.24% over six HtFL baselines, with modest shared-branch overhead, competitive communication and memory costs, and reductions of up to 62.2% in training FLOPs and 66.9% in training time compared with multi-stage or mutual-learning proxy methods.

2 Related Work

Proxy model-based HtFL methods. These methods introduce a homogeneous proxy model or feature extractor as the shared aggregation carrier (e.g., FedType (Wang et al., 2024), FedAPEN (Qin et al., 2023), MH-pFLID (Xie et al., 2024), pFedES (Yi et al., 2025a) and FedGH (Yi et al., 2023)). Each client maintains a heterogeneous private model and a homogeneous proxy, while communicating only the latter. Knowledge exchange between them is typically realized through distillation, mutual learning, or alternating optimization. Although this avoids public data and structural overlap, it decouples global collaboration from private representation learning. Private knowledge must be transferred to the proxy before aggregation and global knowledge transferred back afterward, incurring extra local computation. This bidirectional transfer is also capacity-asymmetric. A compact proxy preserves communication and computation efficiency but constrains transfer fidelity, especially when guiding a stronger private model; enlarging it may improve transfer but increases both communication payload and local computation. Proxy-based HtFL therefore faces an intrinsic tradeoff between transfer fidelity and system efficiency.

Model stitching. Model stitching connects independently parameterized networks at intermediate representations through a lightweight mapping (Lenc & Vedaldi, 2015). Originating as a tool for studying representational equivalence, it was revisited to probe compatibility between frozen pretrained models (Bansal et al., 2021; Athanasiadis et al., 2026). Subsequent studies extend stitching toward efficient model composition. SN-Net stitches pretrained model families to realize different accuracy–efficiency tradeoffs (Pan et al., 2023), KLAS selects promising configurations using KL-based representation similarity (Sanyal et al., 2026), and FedStitch applies stitching to FL by selecting and aggregating pretrained blocks to assemble a global network (Zhan et al., 2024). These methods largely rely on pretrained, stable components to analyze or construct complete stitched predictors, leaving unaddressed how a shared component can adapt to heterogeneous private models with continuously evolving representations.

FedSSP instead stitches an aggregatable shared branch into heterogeneous private models trained from scratch and jointly optimizes it within their evolving representation pathways. This setting necessitates client-specific stitch selection and contribution-aware aggregation across distinct private-model contexts.

3 Preliminaries

3.1 Heterogeneous Federated Learning

A typical HtFL system (Li & Wang, 2019) consists of one central server and K clients. At communication round t, the server samples a subset of clients $\mathcal{I}^t$ for participation. Client k owns a local dataset $D_k = \{(x_i^k, y_i^k)\}_{i=1}^{n_k}$, and maintains a private model $f_k(\cdot; \phi_k)$.

The objective of client k is to minimize its empirical risk

$$\mathcal{L}_k(\phi_k; D_k) = \frac{1}{n_k} \sum_{(x,y)\in D_k} \ell\Big(f_k(x; \phi_k), y\Big), \tag{1}$$

where $\ell(\cdot, \cdot)$ denotes the task loss. Accordingly, the collaborative goal of HtFL can be written as

$$\min_{\phi_1,\ldots,\phi_K} \sum_{k=1}^{K} \frac{n_k}{n} \mathcal{L}_k(\phi_k; D_k), \qquad n = \sum_{k=1}^{K} n_k. \tag{2}$$

In HtFL, client models are structurally different, and thus $\phi_k \in \mathbb{R}^{d_k}$ with possibly different dimensions d_k. As a result, direct parameter aggregation is generally invalid, and effective collaboration must rely on a transferable knowledge carrier beyond homogeneous parameter averaging.

3.2 Model Stitching

Model stitching was originally introduced as a tool for analyzing representation compatibility among neural networks (Bansal et al., 2021; Lenc & Vedaldi, 2015), and has recently been extended to practical model construction across pretrained model families (Pan et al., 2023; Sanyal et al., 2026).

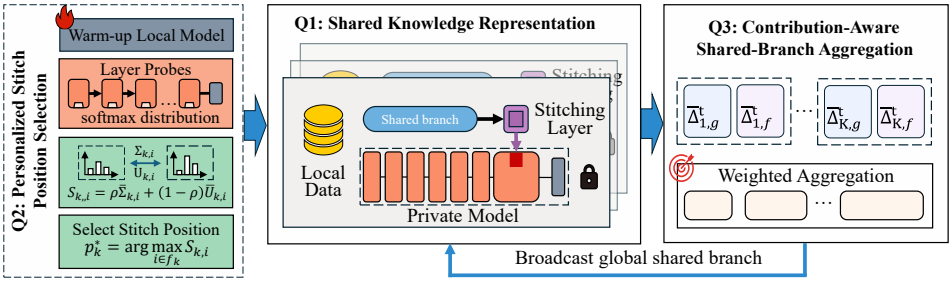


Figure 2: Overview of FedSSP. FedSSP first selects a personalized stitch position for each client's heterogeneous private model, then injects a lightweight shared branch via a stitching layer for local training, and finally updates the global shared branch via contribution-aware aggregation.

Let f be a feed-forward model with M layers, defined as

$$f := f^{(M)} \circ \cdots \circ f^{(1)}, \tag{3}$$

where $f^{(i)} : A_{i-1}^f \to A_i^f$ denotes the mapping from the activation space of layer $i-1$ to that of layer i, and $A_0^f = X$ denotes the input space. For convenience, we define the prefix and suffix of f at layer i as

$$f^{\leq i} := f^{(i)} \circ \cdots \circ f^{(1)}, \quad f^{>i} := f^{(M)} \circ \cdots \circ f^{(i+1)}. \tag{4}$$

Given two source and target networks g and f, model stitching connects the prefix of g up to layer j to the suffix of f after layer i. We refer to the pair $p^* := (j, i)$ as a stitch position.

To align the activation space A_j^g and A_i^f, a 1×1 convolutional layer $S : A_j^g \to A_i^f$ is responsible for the feature transformation. Throughout this paper, we refer to S as the stitching layer.

4 Proposed Method

FedSSP is a HtFL framework designed to address the challenges posed by mismatched parameter spaces across clients. As shown in Figure 2, FedSSP consists of three components: (1) representing shared knowledge as a homogeneous shared branch (Q1), (2) injecting shared knowledge into each private model at a personalized stitch position (Q2), and (3) aggregating the heterogeneous knowledge from clients via contribution-aware aggregation (Q3). We provide detailed pseudocode of FedSSP in the Appendix B.

4.1 Problem Formulation

We consider a HtFL system with K clients. Each client k maintains a private model $f_k(\cdot; \phi_k)$ with M_k blocks (e.g., layers), which is localized to preserve privacy. Further, a shared branch $g_k(\cdot; \psi_k)$ is introduced, which participates in the global aggregation process, replacing the private model for collaboration. The server maintains the global shared branch $g(\cdot; \psi)$. To stitch the shared branch with the private model, a stitching layer $s_k(\cdot; \theta_k)$ is employed, integrating the shared knowledge into the private model. We refer to $\{(f_k, s_k, g_k)\}_{k=1}^K$ as the personalized stitched models, where each client k utilizes its local dataset $D_k = \{(x_i^k, y_i^k)\}_{i=1}^{n_k}$ to locally train its personalized stitched model.

For notational simplicity, the stitching-layer parameters θ_k are absorbed into the client-specific parameter set ϕ_k. Thus, the objective of FedSSP in Eq. equation 2 can be reformulated as:

$$\min_{\psi, \phi_1, \ldots, \phi_K} \sum_{k=1}^{K} w_k^t \mathcal{L}_k \Big((f_k, s_k, g_k); D_k \Big), \tag{5}$$

where w_k^t is the aggregation weight of client k, $\mathcal{L}_k$ is the local loss function, and g_k denotes the shared branch, which is stitched into each private model f_k by stitching layer s_k at the personalized stitch position.

4.2 Shared Knowledge Representation

Typically, proxy-based HtFL methods keep globally shared knowledge in a full homogeneous auxiliary model or feature extractor outside the private model's internal representation pathway (Wang et al., 2024; Yi et al., 2025a). However, existing model-stitching studies (Bansal et al., 2021; Pan et al., 2023) suggest that intermediate representations are often more semantically transferable than shallow features, which are more architecture-dependent and closely tied to local low-level patterns.

Motivated by this observation, FedSSP embeds a homogeneous shared branch g_k into each private model's representation pathway, allowing globally aggregated knowledge to participate directly in task prediction. To align the mismatched dimension and channel, a client-specific stitching layer $s_k(\cdot;\theta_k)$ applies bilinear interpolation for spatial resizing, followed by a lightweight 1×1 convolution for channel projection before fusion. See Appendix A.5 for more details about the architecture and implementation of stitching layer.

Rather than replacing the private prefix, g_k forms a parallel path that carries globally transferable knowledge, while the private path retains client-specific representations. Both paths are jointly optimized under task supervision in a single stage, without an additional explicit and asymmetric knowledge-transfer stage between the shared and private components.

Specifically, for a mini-batch input x, the private-model prefix computes the local representation $z_k^{\mathrm{loc}} = f_k^{\leq p_k^*}(x)$, while the shared branch computes $z_k^{\mathrm{sh}} = g_k(x)$ in parallel. The stitching layer s_k then transforms the shared representation as $\tilde{z}_k^{\mathrm{sh}} = s_k(z_k^{\mathrm{sh}})$ to align it with the private activation space. The two paths are fused at the personalized stitch position p_k^* via

$$z_k = \beta z_k^{\mathrm{loc}} + (1-\beta)\tilde{z}_k^{\mathrm{sh}}, \tag{6}$$

where $\beta \in (0,1)$ is a feature fusion weight balancing local private knowledge and globally shared knowledge. The fused representation is subsequently forwarded through the private suffix to obtain the final prediction $\hat{y} = f_k^{>p_k^*}(z_k)$. The local stitched model is optimized under the unified task loss

$$\mathcal{L}_k = \frac{1}{n_k} \sum_{(x,y)\in D_k} \ell\big(\hat{y}, y\big). \tag{7}$$

Under this formulation, the private model, shared branch, and stitching layer are jointly updated by backpropagating $\mathcal{L}_k$ through the entire stitched prediction path. The detailed architectures of the shared branch are provided in the Appendix A.4.

4.3 Personalized Stitch-Position Selection

Existing model-stitching studies (Pan et al., 2023; Mai et al., 2026; Sanyal et al., 2026; Athanasiadis et al., 2026) have shown that different stitch configurations can lead to substantially different performance, indicating that the stitch position is a key factor affecting the quality of the stitched model. In FedSSP, this issue becomes more critical because different private models have diverse depths, stage structures, and representation hierarchies. Therefore, a globally fixed stitch position is unlikely to be optimal for all clients. Instead, the shared branch should be injected into each private model at a client-specific position.

Unlike conventional stitching between frozen pretrained networks, FedSSP trains private models from scratch, leaving their early intermediate representations semantically unstable for reliable stitch-position selection. We therefore warm up each private model once for several local epochs before FL. Jointly warming up the shared branch would require a predefined stitch position, creating a circular dependency. Unlike KLAS (Sanyal et al., 2026), which evaluates compatibility between pretrained anchors, FedSSP selects the injection region using only the warmed-up private model, since the shared branch cannot be pretrained independently before stitching.

Therefore, we attach lightweight linear probes to consecutive intermediate blocks of the private model and obtain their corresponding softmax distributions $P_{k,i}$. We retain KLAS's intra-model block capacity signal because it does not require a pretrained shared branch:

$$\begin{aligned} \Sigma_{k,i} &= \Theta(P_{k,i}, P_{k,i+1}) \\ &= \frac{1}{n_k} \sum_{x\in D_k} D_{\mathrm{KL}}(P_{k,i}(x) \,\|\, P_{k,i+1}(x)) \,, \end{aligned} \tag{8}$$

a larger $\Sigma_{k,i}$ indicates a stronger representation transition around position i.

However, using $\Sigma_{k,i}$ alone to select the stitch position may lead to suboptimal results. Although $\Sigma_{k,i}$ measures the strength of the distributional transition between two consecutive blocks, a large $\Sigma_{k,i}$ does not necessarily indicate that block $i{+}1$ improves task-relevant representations. Such a transition may reflect either beneficial semantic refinement or representation distortion. To distinguish between these cases, we further introduce the utility signal:

$$U_{k,i} = \max\left(\frac{\mathcal{L}_{\mathrm{CE}}(P_{k,i}) - \mathcal{L}_{\mathrm{CE}}(P_{k,i+1})}{\mathcal{L}_{\mathrm{CE}}(P_{k,i}) + \epsilon},\ 0\right), \tag{9}$$

which explicitly measures the relative classification loss improvement from block i to block $i+1$. Compared to $\Sigma_{k,i}$, $U_{k,i}$ more directly reflects whether the subsequent block effectively facilitates task learning. After normalizing $\Sigma_{k,i}$ and $U_{k,i}$ across all blocks, we combine them into a final stitch position score:

$$\mathcal{S}_{k,i} = \rho\,\bar{\Sigma}_{k,i} + (1-\rho)\,\bar{U}_{k,i}, \tag{10}$$

where $\rho \in (0,1)$ is a trade-off coefficient, and $\bar{\Sigma}_{k,i}$, $\bar{U}_{k,i}$ denote the normalized results.

Finally, the personalized stitch position of client k is determined by maximizing this joint score:$p_k^* = \arg\max_{1\le i<M_k} \mathcal{S}_{k,i}$. Once p_k^* is determined, client k instantiates its personalized stitched model accordingly and performs the local collaborative optimization.

Dynamically switching positions changes the stitching layer's alignment target and may require it to relearn its mapping. Since the shared branch and private model are jointly optimized, frequent switching can abruptly alter fused representations and gradient flow, potentially disrupting co-adaptation and destabilizing local optimization. Thus, FedSSP keeps p_k fixed throughout federated training.

4.4 Contribution-Aware Aggregation

In FedSSP, the aggregation object is a shared branch locally optimized under heterogeneous private-model contexts rather than a unified global model. Although architecturally homogeneous, it is jointly trained with different private models, data distributions, and stitch positions. Consequently, clients contribute differently to shared-branch aggregation, making uniform averaging potentially suboptimal (Liu et al., 2024b).

To characterize clients' contributions to shared-branch aggregation, FedSSP examines local updates from two complementary perspectives: the feature-space shift and the parameter-space shift. Motivated by prior work on adaptive aggregation (Yan et al., 2024; Zhang et al., 2025; Chen et al., 2020; An et al., 2023; Chen et al., 2024; Ribero & Vikalo, 2024), we treat these shifts as side information about local updates that help characterize clients' contributions to the shared model. In the feature space, since FedSSP is a representation-injection framework, a natural feature-space proxy for contribution is to examine how much the output behavior of the shared branch changes after local training. Let $\mathcal{B}_k \subset D_k$ denote a locally sampled mini-batch. We define the feature-space contribution signal as:

$$\Delta_{k,g}^t = \frac{1}{|\mathcal{B}_k|} \sum_{x\in\mathcal{B}_k} \left\| g_k(x;\psi_k^t) - g_k(x;\psi_k^{t-1}) \right\|. \tag{11}$$

A larger $\Delta_{k,g}^t$ indicates a greater change in the shared representation induced by local training, which may reflect greater incorporation of client-specific information into the shared branch. We use this feature-space shift as a proxy for the client's contribution to shared-branch learning.

Although the private model is not aggregated, its update magnitude can provide side information about the local optimization dynamics accompanying shared-branch training. Because the private model and shared branch are jointly optimized within the stitched computation graph, we use this signal to complement the shared-feature shift when constructing aggregation weights. To avoid bias toward larger private models, we measure the layer-wise relative update and average it across all blocks M_k of client k:

$$\Delta_{k,f}^t = \frac{1}{M_k} \sum_{m=1}^{M_k} \frac{\left\| \phi_k^{t,(m)} - \phi_k^{t-1,(m)} \right\|}{\left\| \phi_k^{t-1,(m)} \right\| + \epsilon}, \tag{12}$$

$\Delta_{k,g}^t$ and $\Delta_{k,f}^t$ are computed in different metric spaces, their raw magnitudes are not comparable. The server first normalizes both across all participating clients in round t:

$$\bar{\Delta}_{k,g}^t = \frac{\Delta_{k,g}^t}{\sum_{j\in\mathcal{I}^t}\Delta_{j,g}^t}, \qquad \bar{\Delta}_{k,f}^t = \frac{\Delta_{k,f}^t}{\sum_{j\in\mathcal{I}^t}\Delta_{j,f}^t}, \tag{13}$$

where $\mathcal{I}^t$ denotes the set of participating clients at round t. We then combine the two normalized signals into a final aggregation weight: $w_k^t = \lambda\bar{\Delta}_{k,g}^t + (1-\lambda)\bar{\Delta}_{k,f}^t$, where $\lambda \in (0,1)$ is a contribution trade-off coefficient that balances the feature-space and parameter-space contributions.

Finally, the server aggregates the uploaded shared branches as $\psi^t = \sum_{k\in\mathcal{I}^t} w_k^t \psi_k^t$. Note that only the shared branch and two signals needed to be uploaded to the server, while the private model and the stitching layer remain strictly local to preserve personalization.

5 Experiments

5.1 Experimental Setups

Datasets. We evaluate FedSSP and all baselines on Fashion-MNIST (Xiao et al., 2017), CIFAR-10, CIFAR-100 (Krizhevsky, 2009) and Tiny-ImageNet (Le & Yang, 2015). Client data are partitioned using the Dirichlet strategy (Hsu et al., 2019) with $\alpha \in \{0.1, 0.3, 0.5, 0.7\}$. $\alpha = 0.3$ is used by default.

Baseline methods. We compare six representative HtFL baselines from four categories: common-layer aggregation (FlexiFed (Wang et al., 2023)), knowledge distillation (FedMD (Li & Wang, 2019)), prototype-based (FedProto (Tan et al., 2022) and FedTGP (Zhang et al., 2024)), and proxy-based methods (FedType (Wang et al., 2024) and pFedES (Yi et al., 2025a)).

Model settings. We consider four backbone-family settings: Homo uses ResNet-50 (He et al., 2016) for all clients; VGG-Family uses VGG-11/13/16/19 (Simonyan & Zisserman, 2014); ResNet-Family uses ResNet-18/34/50/101/152; and Mixed-Family combines ResNet-34/50/101/152, VGG-11/13/16/19, MobileNet-V1 (Howard et al., 2017), and MobileNet-V2 (Sandler et al., 2018). ResNet-Family is used by default. In FedSSP, the shared branch is instantiated from a lightweight ResNet-style network by removing its classifier.

Training strategy. The number of clients ranges from 10 to 100, with the exact number specified for each experiment. We use full client participation ($C = 100\%$), 10 clients, 100 communication rounds, and five local epochs by default. Each client's data are split 80/20 for training/testing. To ensure reproducibility and fair comparison, we use the same random seed for all main comparison experiments. We report the best round-wise average test accuracy across clients in each trial. Baselines use the default hyperparameters reported in their original papers, while FedSSP uses $(\beta, \rho, \lambda) = (0.4, 0.6, 0.7)$. Hyperparameters in FedSSP were selected through sensitivity analyses conducted with three different random seeds. We provide a comprehensive hyperparameter sensitivity analysis in Appendix C.7 C.8 C.9. Further experimental details are provided in Appendix A.2.

5.2 Impact of Data and Model Heterogeneity

We compare FedSSP with six HtFL baselines under different data and model heterogeneity settings. Due to space constraints, we report the comparative results on Fashion-MNIST in Appendix C.1.

Data Heterogeneity. Table 1 shows that FedSSP achieves the best or second-best average accuracy across all completed Dirichlet configurations, outperforming other HtFL baselines by up to 7.73%. We attribute these gains to FedSSP's dual-path local design, which integrates global knowledge from the shared branch with client-specific representations learned by the private prefix. This design enables clients to benefit from global collaboration without sacrificing local personalization under Non-IID distributions.

Model Heterogeneity. Table 2 demonstrates the robustness of FedSSP to architectural diversity. FedSSP improves average accuracy over other HtFL method by up to 6.75%. This improvement can be attributed to FedSSP's personalized stitching design, which uses a homogeneous shared branch for global aggregation while adapting its injection position and feature mapping to each private architecture. Support for Transformer-based models is also provided in the Appendix C.2.

Table 1: Test accuracy (%) under Dirichlet Non-IID settings with ResNet-Family models. Avg. is the mean across the four Dirichlet settings for each dataset. Bold and underline mark the best and second-best results.

Datasets	CIFAR-10					CIFAR-100					Tiny-ImageNet				
α	0.7	0.5	0.3	0.1	Avg.	0.7	0.5	0.3	0.1	Avg.	0.7	0.5	0.3	0.1	Avg.
FlexiFed	85.32	90.08	89.79	84.36	87.39	**56.38**	61.21	65.75	72.56	63.98	**46.73**	48.15	51.85	57.66	51.10
FedMD	80.74	87.57	90.78	94.75	88.46	49.54	58.88	62.63	71.10	60.54	45.19	46.86	52.30	62.44	51.70
FedProto	80.22	84.09	86.92	87.62	84.71	40.27	49.12	48.42	64.77	50.65	23.10	30.31	34.09	47.43	33.73
FedTGP	80.52	87.59	91.56	92.91	88.15	40.60	48.72	54.69	67.48	52.87	32.46	34.71	41.83	44.06	38.27
pFedES	83.88	87.57	89.74	93.51	88.68	46.59	54.01	61.36	72.93	58.72	42.94	45.63	52.37	63.39	51.08
FedType	80.51	83.32	88.74	92.49	86.27	45.60	47.42	52.53	66.73	53.07	40.87	43.07	52.57	60.06	49.14
FedSSP	**85.70**	**90.16**	**93.32**	**94.96**	**91.04**	53.71	**62.97**	**68.29**	**78.11**	**65.77**	45.88	**49.04**	**52.98**	**65.35**	**53.31**

Table 2: Test accuracy (%) under model heterogeneity at $\alpha = 0.3$. Avg. is the mean across the four model groups for each dataset. Bold and underline mark the best and second-best results. "-Family" is omitted from model group names.

Datasets	CIFAR-10					CIFAR-100					Tiny-ImageNet				
Models	Homo	VGG	ResNet	Mixed	Avg.	Homo	VGG	ResNet	Mixed	Avg.	Homo	VGG	ResNet	Mixed	Avg.
FlexiFed	86.99	89.68	89.79	90.89	89.34	**71.45**	65.70	65.75	63.16	66.52	**56.84**	46.93	51.85	52.16	51.95
FedMD	89.90	89.52	90.78	89.53	89.93	52.57	62.94	62.63	58.42	59.14	51.17	45.53	52.30	45.15	48.54
FedProto	87.51	89.92	86.92	87.21	87.89	53.74	49.38	48.42	52.42	50.99	38.50	36.31	34.09	35.46	36.09
FedTGP	91.09	90.86	91.56	90.33	90.96	53.39	55.10	54.69	51.37	53.64	41.89	44.45	41.83	41.99	42.54
pFedES	90.71	90.75	89.74	89.87	90.27	61.63	61.52	61.36	58.67	60.80	53.14	48.64	52.37	50.83	51.25
FedType	87.19	89.87	88.74	88.56	88.59	56.69	52.42	52.53	52.58	53.56	50.61	46.18	52.57	47.51	49.22
FedSSP	**92.87**	**91.52**	**93.32**	**91.10**	**92.20**	68.68	**67.56**	**68.29**	**64.28**	**67.20**	55.57	**50.22**	**52.98**	**52.44**	**52.80**

5.3 Scalability Analysis

We evaluate scalability by varying the total client population K and participation rate C. As shown in Figure 3, FedSSP achieves the highest median accuracy in 11 out of 12 settings and is nearly tied with the best baseline in the remaining one. When fixed clients participate per round, increasing K shifts the client-level accuracy distributions of all methods downward and generally broadens them, whereas FedSSP maintains an overall higher distribution, particularly on CIFAR-100. Under full participation, where the number of participating clients grows with K, all methods still degrade as K increases, while FedSSP retains the better median throughout.

5.4 System Cost Analysis

Table 3 and Figure 4 compare the communication, runtime, and memory costs of all methods. Although prototype-based methods communicate less, they achieve lower accuracy in this setting. By exchanging only the shared branch and two scalar statistics while keeping private models and stitching layers local, FedSSP reduces per-round communication by 99.3% and 99.8% relative to FedType and FlexiFed, respectively. Its single-stage local optimization yields the second-lowest training time, reducing it by 66.1% and 66.9% relative to pFedES and FedType. FedSSP also uses less peak training memory than pFedES and a comparable amount to FedType, requiring 0.460 seconds and 336 MB for inference. We further report computational cost and warm-up/probe overhead of FedSSP in the Appendix C.3, C.4, C.5 and C.6. Overall, FedSSP improves accuracy with modest shared-branch overhead, keeping communication costs low and peak memory comparable to baselines while reducing computation and runtime relative to proxy methods with multi-stage training or mutual learning.

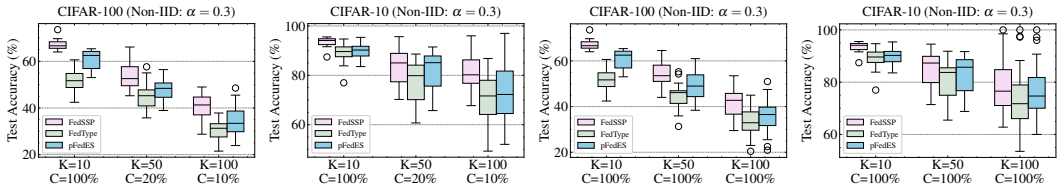


Figure 3: Scalability comparison of FedSSP, FedType, and pFedES with varying numbers of clients.

Table 3: Per-round communication on CIFAR-100 with ResNet-Family.

Method	Theory	Practice
FlexiFed	$\sum_i^{\lvert\mathcal{I}_t\rvert}(\lvert\phi_i\rvert + \lvert\phi^{com}\rvert)$	2510.93 MB
FedMD	$2 \times \lvert\mathcal{I}^t\rvert \times N_g \times C_{cls}$	400 MB
FedProto	$D \times \sum_{i\in\mathcal{I}_t}(C_i + C_{cls})$	4.10 MB
FedTGP	$D \times \sum_{i\in\mathcal{I}_t}(C_i + C_{cls})$	4.10 MB
pFedES	$2 \times \lvert\mathcal{I}_t\rvert \times \lvert P^t_{ext}\rvert$	0.05 MB
FedType	$2 \times \lvert\mathcal{I}_t\rvert \times \lvert P^t_{model}\rvert$	897.60 MB
FedSSP	$\lvert\mathcal{I}_t\rvert \times (2 \times \lvert\psi^t\rvert + 2)$	6.19 MB

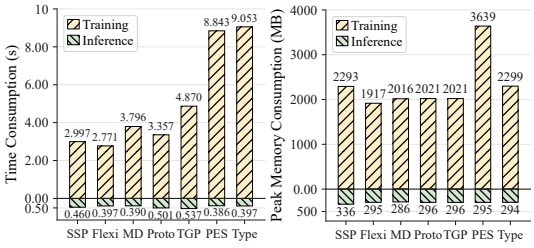
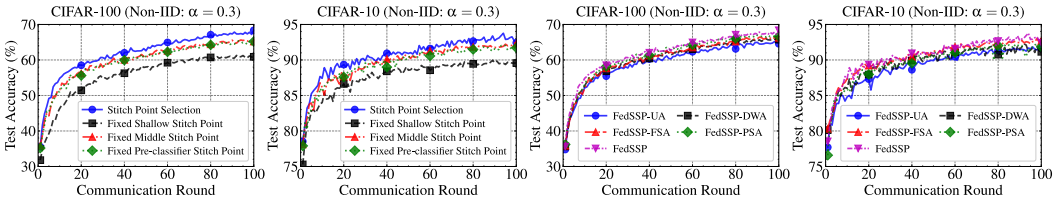


Figure 4: Client runtime per local epoch and peak GPU memory on CIFAR-100 ($K = 10$, $C = 100\%$). Method names omit "Fed".



(a) Personalized Stitch-Position Selection (b) Contribution-Aware Aggregation

Figure 5: (a) Sensitivity analysis of stitch positions and (b) effectiveness of contribution-aware aggregation for FedSSP.

5.5 Ablation Study

We conduct ablation studies to examine the stitched architecture, personalized stitch-position selection, and contribution-aware aggregation in FedSSP.

Effectiveness of the Stitched Model. Table 4 compares three standalone variants. The Private variant retains the full private model. The Shared variant replaces the private prefix with the shared branch while retaining the private suffix, forming a complete model. The stitched model achieves accuracy comparable to the private model, slightly exceeds the Shared variant. Stitching preserves local performance, while added parameters yield no consistent accuracy gain.

Table 4: Accuracy (%) and loss of private, shared, and stitched models.

Dataset	Accuracy (%)			Loss		
	Private	Shared	**Stitched**	Private	Shared	**Stitched**
CIFAR-100	65.45	65.10	**65.66**	1.64	1.59	**1.54**
CIFAR-10	**91.79**	91.68	91.72	0.39	0.36	**0.35**

Effectiveness of Personalized Stitch-Position Selection. Figure 5(a) compares the proposed selection strategy with three fixed stitch positions. The proposed strategy achieves the highest final accuracy on both datasets, whereas shallow stitching performs worst. The performance differences among fixed positions show that the injection location is consequential and support adapting it to each private backbone.

Effectiveness of Contribution-Aware Aggregation. Figure 5(b) also compares FedSSP with uniform averaging (UA), data-size weighting (DWA), and two single-signal variants (FSA and PSA). FedSSP achieves better accuracy on both datasets. Its advantage over UA and DWA supports contribution-aware weighting, while its advantage over FSA and PSA indicates that the two signals provide complementary information for shared-branch aggregation.

6 Conclusion

We introduced FedSSP, a HtFL framework to represent transferable global knowledge as a shared branch stitched into private models. It combines KL-based block capacity and task utility for personal stitch selection and uses contribution-aware weighting to adapt shared-branch aggregation to heterogeneous local contexts. This design avoids structural overlap, public data, and asymmetric proxy transfer while keeping private parameters local. Across four datasets and eleven architectures, FedSSP improves average accuracy over six HtFL baselines by up to 7.24%.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language polishing. Additionally, We also used these tools for table formatting and checking the consistency and numerical accuracy of reported results. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the assistance of generative AI.

REFERENCES

Xuming An, Li Shen, Han Hu, and Yong Luo. Federated learning with manifold regularization and normalized update reaggregation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 55097–55109. Curran Associates, Inc., 2023. doi: 10.52202/075280-2406. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/acf2b98eeb09b21968c2de6b1c6952e9-Paper-Conference.pdf.

Ioannis Athanasiadis, Anmar Karmush, and Michael Felsberg. Grounding functional similarity by invariance-aware model stitching. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=yUEFOa32CG.

András Balogh and Márk Jelasity. How not to stitch representations to measure similarity: Task loss matching versus direct matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 15472–15480, 2025.

Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. *Advances in neural information processing systems*, 34:225–236, 2021.

Wenlin Chen, Samuel Horvath, and Peter Richtarik. Optimal client sampling for federated learning. *arXiv preprint arXiv:2010.13723*, 2020.

Yuhang Chen, Wenke Huang, and Mang Ye. Fair federated learning under domain skew with local consistency and domain diversity. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12077–12086, 2024. doi: 10.1109/CVPR52733.2024.01148.

Enmao Diao, Jie Ding, and Vahid Tarokh. HeteroFL: Computation and communication efficient federated learning for heterogeneous clients. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=TNkPBBYFkXg.

Boyu Fan, Siyang Jiang, Xiang Su, Sasu Tarkoma, and Pan Hui. A survey on model-heterogeneous federated learning: Problems, methods, and prospects. In *2024 IEEE international conference on big data (BigData)*, pp. 7725–7734. IEEE, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications, 2017. URL https://arxiv.org/abs/1704.04861.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Qing Hu, Tianchi Liao, Shuyi Wu, Lei Yang, and Chuan Chen. FedLAGC: Towards high performance system-heterogeneous federated learning via layer-adaptive submodel extraction and gradient correction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 21858–21866, 2026.

Chun-Yin Huang, Kartik Srinivas, Xin Zhang, and Xiaoxiao Li. Overcoming data and model heterogeneities in decentralized federated learning via synthetic anchors. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=mNzkumTSVL.

Wenke Huang, Mang Ye, and Bo Du. Learn from others and be yourself in heterogeneous federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10143–10153, 2022.

Ninghui Jia, Zhihao Qu, Baoliu Ye, Yanyan Wang, Shihong Hu, and Song Guo. A comprehensive survey on communication-efficient federated learning in mobile edge environments. *IEEE Communications Surveys & Tutorials*, 27(6):3710–3741, 2025. doi: 10.1109/COMST.2025.3535957.

Lei Jiang, Xiaoding Wang, Xu Yang, Jiwu Shu, Hui Lin, and Xun Yi. Fedpa: Generator-based heterogeneous federated prototype adversarial learning. *IEEE Transactions on Dependable and Secure Computing*, 22(2):939–949, 2025. doi: 10.1109/TDSC.2024.3419211.

Honggu Kang, Seohyeon Cha, and Joonhyuk Kang. Gefl: Model-agnostic federated learning with generative models. *IEEE Transactions on Mobile Computing*, 25(4):4493–4505, 2026. doi: 10.1109/TMC.2025.3592483.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020.

Latif U Khan, Walid Saad, Zhu Han, Ekram Hossain, and Choong Seon Hong. Federated learning for internet of things: Recent advances, taxonomy, and open challenges. *IEEE Communications Surveys & Tutorials*, 23(3):1759–1799, 2021.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Ya Le and Xuan Yang. Tiny ImageNet visual recognition challenge. Stanford CS231N Course Report, 2015. URL https://cs231n.stanford.edu/reports/2015/pdfs/yle_project.pdf.

Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 991–999, 2015.

Daliang Li and Junpu Wang. FedMD: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

Jiahao Liu, Yipeng Zhou, Di Wu, Miao Hu, Mohsen Guizani, and Quan Z Sheng. FedLMT: Tackling system heterogeneity of federated learning via low-rank model training with theoretical guarantees. In *Forty-first International Conference on Machine Learning*, 2024a.

Jun Liu, Jianchun Liu, Hongli Xu, Yunming Liao, Zhiyuan Wang, and Qianpiao Ma. YOGA: Adaptive layer-wise model aggregation for decentralized federated learning. *IEEE/ACM Transactions on Networking*, 32(2):1768–1780, 2024b. doi: 10.1109/TNET.2023.3329005.

Qi Liu, Jinze Wu, Zhenya Huang, Hao Wang, Yuting Ning, Ming Chen, Enhong Chen, Jinfeng Yi, and Bowen Zhou. Federated user modeling from hierarchical information. *ACM Transactions on Information Systems*, 41(2):1–33, 2023.

Kangyang Luo, Shuai Wang, Yexuan Fu, Xiang Li, Yunshi Lan, and Ming Gao. DFRD: Data-free robustness distillation for heterogeneous federated learning. In *Advances in Neural Information Processing Systems*, 2023.

Zheda Mai, Ke Zhang, Fu-En Wang, Zixiao Ken Wang, Albert Y. C. Chen, Lu Xia, Min Sun, Wei-Lun Chao, and Cheng-Hao Kuo. Revisiting model stitching in the foundation model era. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 41342–41351, June 2026.

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*, pp. 1273–1282. PMLR, 2017.

Ziru Niu, Hai Dong, and AK Qin. Bridging generalization gap of heterogeneous federated clients using generative models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Zizheng Pan, Jianfei Cai, and Bohan Zhuang. Stitchable neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16102–16112, 2023.

Jong-Ik Park and Carlee Joe-Wong. Federated learning with flexible architectures. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 143–161. Springer, 2024.

Jiaming Pei, Wenxuan Liu, Jinhai Li, Lukun Wang, and Chao Liu. A review of federated learning methods in heterogeneous scenarios. *IEEE Transactions on Consumer Electronics*, 70(3):5983–5999, 2024. doi: 10.1109/TCE.2024.3385440.

Anik Pramanik, Murat Kantarcioglu, Vincent Oria, and Shantanu Sharma. FedDAG: Clustered federated learning via global data and gradient integration for heterogeneous environments. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=eIMXY9qkKz.

Zhen Qin, Shuiguang Deng, Mingyu Zhao, and Xueqiang Yan. FedAPEN: Personalized cross-silo federated learning with adaptability to statistical heterogeneity. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 1954–1964, 2023.

Zhihao Qu, Ninghui Jia, Baoliu Ye, Shihong Hu, and Song Guo. FedQClip: Accelerating federated learning via quantized clipped SGD. *IEEE Transactions on Computers*, 74(2):717–730, 2025. doi: 10.1109/TC.2024.3477972.

Mónica Ribero and Haris Vikalo. Reducing communication in federated learning via efficient client sampling. *Pattern Recognition*, 148:110122, 2024. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2023.110122. URL https://www.sciencedirect.com/science/article/pii/S0031320323008191.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Debopam Sanyal, Anantharaman S. Iyer, Alind Khare, Trisha Jain, Akshay Jajoo, Myungjin Lee, James Clayton Kerce, and Alexey Tumanov. KLAS: Using similarity to stitch neural networks for improved accuracy-efficiency tradeoffs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=iTaQmRWa7Y.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. FedProto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8432–8440, 2022.

Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BkluqlSFDS.

Jiaqi Wang, Chenxu Zhao, Lingjuan Lyu, Quanzeng You, Mengdi Huai, and Fenglong Ma. Bridging model heterogeneity in federated learning via uncertainty-based asymmetrical reciprocity learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 52290–52308. PMLR, 2024.

Kaibin Wang, Qiang He, Feifei Chen, Chunyang Chen, Faliang Huang, Hai Jin, and Yun Yang. FlexiFed: Personalized federated learning for edge clients with heterogeneous model architectures. In *Proceedings of the ACM Web Conference 2023*, pp. 2979–2990, 2023.

Xiao Wang, Zhengda Wu, and Jinghua Zhu. FedAPE: Heterogeneous federated learning with attention-guided aggregation and prototype enhancement. *Future Generation Computer Systems*, 180:108417, 2026.

Feijie Wu, Xingchen Wang, Yaqing Wang, Tianci Liu, Lu Su, and Jing Gao. FIARSE: Model-heterogeneous federated learning via importance-aware submodel extraction. *Advances in Neural Information Processing Systems*, 37:115615–115651, 2024a.

Shu Wu, Jindou Chen, Xueli Nie, Yong Wang, Xiancun Zhou, Linlin Lu, Wei Peng, Yao Nie, and Waseef Menhaj. Global prototype distillation for heterogeneous federated learning. *Scientific Reports*, 14:12057, 2024b. doi: 10.1038/s41598-024-62908-0.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

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. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54561–54575. PMLR, 2024.

Yunlu Yan, Lei Zhu, Yuexiang Li, Xinxing Xu, Rick Siow Mong Goh, Yong Liu, Salman Khan, and Chun-Mei Feng. A new perspective to boost performance fairness for medical federated learning. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*, pp. 13–23, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-72117-5.

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.

Liping Yi, Gang Wang, Xiaoguang Liu, Zhuan Shi, and Han Yu. FedGH: Heterogeneous federated learning with generalized global header. In *Proceedings of the 31st ACM International Conference on Multimedia*, 2023. doi: 10.1145/3581783.3611781.

Liping Yi, Han Yu, Chao Ren, Gang Wang, Xiaoguang Liu, and Xiaoxiao Li. pFedES: Generalized proxy feature extractor sharing for model heterogeneous personalized federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 22146–22154, 2025a. doi: 10.1609/aaai.v39i21.34368.

Liping Yi, Han Yu, Gang Wang, Xiaoguang Liu, and Xiaoxiao Li. Federated representation angle learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1314–1324, October 2025b.

Shichen Zhan, Yebo Wu, Chunlin Tian, Yan Zhao, and Li Li. Heterogeneity-aware coordination for federated learning via stitching pre-trained blocks. In *2024 IEEE/ACM 32nd International Symposium on Quality of Service (IWQoS)*, pp. 1–10, 2024. doi: 10.1109/IWQoS61813.2024.10682959.

Jianqing Zhang, Yang Liu, Yang Hua, and Jian Cao. FedTGP: Trainable global prototypes with adaptive-margin-enhanced contrastive learning for data and model heterogeneity in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16768–16776, 2024.

Jie Zhang, Song Guo, Xiaosong Ma, Haozhao Wang, Wencao Xu, and Feijie Wu. Parameterized knowledge transfer for personalized federated learning. In *Advances in Neural Information Processing Systems*, 2021.

Xiangtao Zhang, Sheng Li, Ao Li, Yipeng Liu, Fan Zhang, Ce Zhu, and Le Zhang. Subspace constraint and contribution estimation for heterogeneous federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20632–20642, 2025.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with Non-IID data. 2018. doi: 10.48550/ARXIV.1806.00582. URL https://arxiv.org/abs/1806.00582.

Yanbing Zhou, Xiangmou Qu, Chenlong You, Jiyang Zhou, Jingyue Tang, Xin Zheng, Chunmao Cai, and Yingbo Wu. FedSA: A unified representation learning via semantic anchors for prototype-based federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23009–23017, 2025. doi: 10.1609/aaai.v39i21.34464.

Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International Conference on Machine Learning*, pp. 12878–12889. PMLR, 2021.