H-SFP: Hierarchical Federated Learning with Decoupled Split-Model Prototyping

Dung T. Tran[1], Nguyen B. Ha[1], Minh-Duong Nguyen[2], Van-Dinh Nguyen[3], and Kok-Seng Wong[1]†

[1] Center for Environmental Intelligence, VinUniversity, Hanoi, Vietnam
[2] Center for AI Research, VinUniversity, Hanoi, Vietnam
[3] School of Computer Science and Statistics, Trinity College Dublin, Ireland

Abstract. Federated learning (FL) enables collaborative model training without sharing raw data, but practical deployments must balance statistical heterogeneity, resource-constrained clients, and communication efficiency. While conventional FL requires clients to train full models locally, split federated learning (SFL) reduces client-side computation by partitioning models across tiers, at the cost of transmitting high-dimensional activations and gradients during training. We propose Hierarchical Split-Federated Prototyping (H-SFP), a communication-efficient framework for hierarchical cloud–edge–client environments. Instead of exchanging sample-level activations or gradients, H-SFP communicates compact class-wise feature statistics and uses them to synthesize representative feature distributions at upper tiers. This statistical prototyping mechanism enables independent training of model segments while preserving the computational advantages of split-model execution and substantially reducing synchronization and communication overhead. A key finding of this work is that lightweight first- and second-order feature statistics are sufficient to support hierarchical split-model training, providing an effective communication interface without requiring sample-level feature exchange. We analyze the communication properties and stability of the proposed statistical interface under hierarchical aggregation. Extensive experiments on CIFAR-10/100, HAM10000, ImageNet-1K, and ISIC-2018 demonstrate that H-SFP remains robust under heterogeneous data distributions while reducing communication overhead by up to two orders of magnitude compared with federated and split-learning baselines.

Keywords: Federated Learning, Hierarchical Federated Learning, Split Federated Learning, Feature Prototyping, Communication Efficiency

1 Introduction

Federated learning (FL) [15] enables collaborative model training across distributed clients without exchanging raw data, making it a promising paradigm

† Corresponding author. Email: wong.ks@vinuni.edu.vn.

for privacy-sensitive visual learning. Despite this advantage, practical FL deployments must balance three competing factors: statistical heterogeneity across clients [18, 28, 30], limited computational resources on edge devices [9, 37], and communication efficiency in large-scale distributed systems [19,21]. Conventional FL methods typically require each client to train the full model locally, which can be prohibitively expensive for resource-constrained devices [5, 11, 22].

Split federated learning (SFL) addresses this limitation by partitioning model execution across clients and servers [29]. By assigning only shallow network layers to clients, SFL reduces client-side computation and memory requirements [8,33]. Hierarchical SFL further extends this idea to cloud–edge–client systems, where edge servers provide intermediate computation and aggregation. However, existing SFL and hierarchical SFL (HSFL) frameworks usually rely on transmitting intermediate activations during the forward pass and propagating gradients during backpropagation [16, 25, 35, 40]. This sample-level communication creates substantial bandwidth consumption, introduces synchronization dependencies across tiers, and couples the optimization of client, edge, and cloud components [13, 26]. These challenges become more pronounced as models, client populations, and data heterogeneity increase.

In this work, we revisit the communication interface of hierarchical split learning. Instead of exchanging sample-level activations and gradients, we ask whether compact class-level feature statistics can provide sufficient information for training upper model components. Our key observation is that first- and second-order statistics of client-side feature embeddings can preserve useful class-level semantic structure while requiring much less communication than activation-level exchange. This motivates a statistical feature-prototyping mechanism, in which upper tiers synthesize representative feature distributions from compact summaries rather than relying on direct cross-tier backpropagation.

Based on this principle, we propose **Hierarchical Split-Federated Prototyping (H-SFP)**, a communication-efficient framework for cloud–edge–client learning. In H-SFP, clients execute lightweight feature extractors and transmit only class-wise feature statistics to nearby edge servers. Edge servers aggregate these statistics, synthesize representative feature samples, and train intermediate model components. The cloud tier further aggregates distribution-level summaries to train task-specific layers. By replacing activation and gradient exchange with statistical feature prototyping, H-SFP preserves the computational advantage of split-model execution while decoupling optimization across hierarchical tiers and reducing communication overhead.

The main finding of this work is that compact statistical summaries can serve as an effective communication interface for HSFL. Although H-SFP does not transmit sample-level activations or gradients across tiers, it maintains strong performance under heterogeneous data distributions while substantially reducing communication cost. Our contributions are summarized as follows:

– We propose H-SFP, a communication-efficient hierarchical split-federated framework that replaces sample-level activation and gradient exchange with compact class-wise feature statistics.

- We introduce a statistical feature-prototyping mechanism that enables the edge and cloud tiers to synthesize representative feature distributions and to train upper-model components without cross-tier backpropagation.
- We provide theoretical analysis of prototype-induced distribution shift and the stability of statistical feature representations in hierarchical setups.
- We conduct experiments on image classification and medical image segmentation benchmarks, demonstrating robust performance under heterogeneous data distributions with substantially reduced communication overhead.

2 Background and Related Works

Our work lies at the intersection of research directions addressing the inherent trilemma of FL: statistical heterogeneity, system heterogeneity, and communication bottlenecks. The first category, standard FL, aims to collaboratively train global models without centralizing raw data. Fundamental algorithms like FedAvg [15] often suffer from severe "client drift" due to non-IID data distributions. To mitigate this, traditional solutions rely on regularization mechanisms, such as proximal terms in FedProx [5], or momentum-based optimization [3, 38]. To address variable local work across heterogeneous clients, FedNova [34] normalizes client updates before aggregation. However, a critical limitation of these frameworks is that they inherently assume clients possess the computational capacity to train and evaluate full models locally. This assumption exacerbates system heterogeneity and excludes resource-constrained edge devices from participating.

To alleviate these client-side computational constraints, SFL partitions the model and offloads heavy upper layers to the server. SplitFed [29] was introduced to parallelize client updates, resolving the sequential training bottlenecks of early split learning. This literature has expanded to address computational diversity and scale through HeteroSFL paradigms that build on sub-model deployments such as HeteroFL [7] and Hierarchical SFL (HSFL) [14, 27, 36], which introduce edge servers for intermediate synchronization. Despite these topological advancements, these frameworks suffer from limitations. They require the continuous transmission of high-dimensional activations during the forward pass and the subsequent return of gradients during the backward pass. This bidirectional dependency introduces synchronization overhead, exacerbates communication bottlenecks, and creates new computational bottlenecks on the server [13].

As an alternative to raw gradient and activation communication, the third category explores generative and prototype-based FL. Methods such as Federated Knowledge Distillation (FedDF [12]) and data-free Generative FL (FedGen [32]) address both communication overhead and non-IID challenges but typically require training computationally expensive generators, such as GANs, at the server. Conversely, prototype-based FL, such as FedProto [28] or FOUL [20], exchanges lightweight class prototypes instead of gradients or heavy activations. A major limitation of prototype methods, however, is that they primarily utilize these prototypes as client-side regularization terms to align local feature spaces. Because they are not used to synthesize actual training data, these methods can still suffer from prototype drift under extreme heterogeneity [1, 39].

H-SFP complements these resource-efficient and decoupled learning approaches. PriSM [24] reduces client computation by sampling principal kernels to construct lightweight sub-models, while Delta [23] studies asymmetric private/public computation flows with explicit privacy mechanisms. In contrast, H-SFP focuses on the HSFL setting, where clients, edge servers, and the cloud train different model segments. Its main design choice is a statistic-based cross-tier interface: upper tiers receive class-wise feature moments rather than sample-wise activations, gradients, full model updates, or learned generators. This makes H-SFP particularly suitable for hierarchical deployments.

Existing frameworks preserve a shared end-to-end optimization graph across tiers. Even when activations or prototypes are transmitted rather than full models, the upper and lower tiers remain coupled via cross-tier gradient propagation or parameter synchronization. This architectural dependency forces all components to participate in a single distributed optimization process, making communication and statistical drift intrinsically entangled. In contrast, H-SFP departs from this paradigm by severing the optimization graph across tiers. Rather than coordinating updates through sample-wise gradients or activations, H-SFP performs tier-wise optimization over synthesized feature distributions derived from aggregated statistical moments. This design provides a lightweight communication interface between client, edge, and cloud tiers while preserving the practical advantage of split-model execution on resource-constrained clients. A detailed comparison between approaches is shown in the Appendix A.

3 The Proposed H-SFP Framework

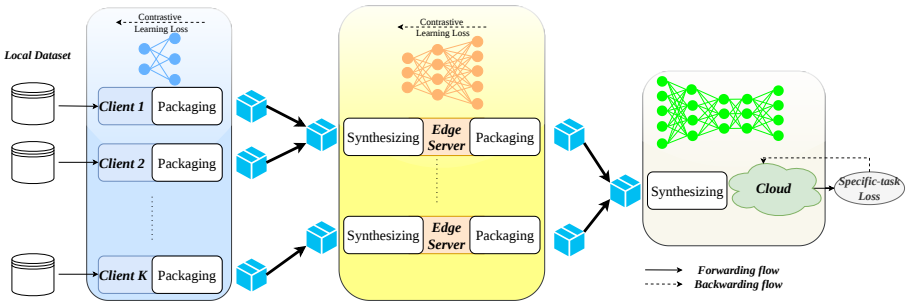
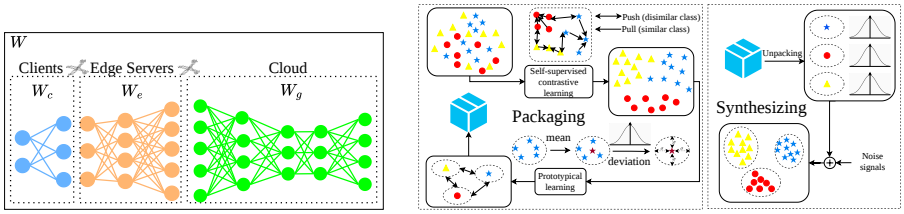


Fig. 1: The H-SFP architecture and data flow. Clients communicate class-wise feature statistics instead of sample-level activations and gradients. Edge and cloud tiers iteratively aggregate statistics, synthesize feature distributions, and train their assigned model components.

We propose H-SFP, which replaces cross-tier backpropagation with statistic-based feature synthesis. Each tier trains its assigned model part locally, packages



(a) Hierarchical Model Partitioning. **(b)** Generative Prototyping.

Fig. 2: The proposed H-SFP Framework. (a) Model partitioning (W_c, W_e, W_g) offloads computational burden from the clients. (b) The core generative prototyping mechanism, where self-supervised contrastive learning at the client level enables the compression of representations into lightweight statistical moments (μ, σ).

its learned feature distribution into lightweight class-wise statistics, and transmits only these statistics upward. The following subsections describe the system architecture, prototype packaging and synthesis, and the resulting communication behavior. Concrete algorithms for each tier are provided in the Appendix B.

3.1 System Architecture and Motivation

We consider a three-tier "cloud-edge-client" framework shown in Figure 1, comprising a global cloud server, M edge servers, and K clients partitioned into M disjoint clusters $\{\mathcal{C}_m\}_{m=1}^{M}$. This framework directly addresses the scalability bottleneck of centralized FL by allowing edge servers to perform intermediate aggregations. In real-world deployments, this partitioning would be determined by physical proximity or by network latency pinging. However, for our empirical evaluation, we deliberately adopt a random assignment strategy. This represents a strict, unoptimized baseline, demonstrating that H-SFP's generative aggregation can successfully stabilize distributions and mitigate client drift even without the benefit of a geographically optimized or clustered network topology.

To solve resource-constrained clients, H-SFP adopts a split-learning architecture. As illustrated in Figure 2a, the global model W is partitioned into three components: a lightweight client-side feature extractor W_c, an intermediate edge-side model W_e, and a cloud-side task model W_g. This reduces the computational and memory burden on clients by offloading deeper network layers to upper tiers. However, conventional split federated learning requires continuous transmission of intermediate activations and gradients to maintain end-to-end optimization, leading to substantial communication and synchronization overhead.

H-SFP resolves this by replacing backpropagation with a feed-forward, multi-tier generative prototyping mechanism (Figure 2b). By transmitting lightweight statistics rather than raw activations, we completely decouple the network segments, enabling communication-efficient and asynchronous training. Additionally, the precise cut points are treated as hyperparameters determined by the target device's memory constraints. For instance, clients may only compute the

first residual block of a ResNet architecture, offloading the deeper, computationally expensive blocks to the edge and cloud tiers.

3.2 Multi-tier Self-Learning

The first tier of learning occurs on resource-constrained clients. Each client k trains its local extractor $W_{c,k}$ using its private data $\mathcal{D}_k$. The objective $\mathcal{L}_k(W_{c,k})$ is to learn a robust representation space using the self-supervised NT-Xent contrastive loss [2]. For a batch B, let $z_i = W_{c,k}(x_i)$ where $x_i \in D_k$:

$$\mathcal{L}_k(W_{c,k}) = \mathcal{L}_{\text{con},k} = \sum_{i \in B} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(z_i \cdot z_p/\tau)}{\sum_{a \in A(i)} exp(z_i \cdot z_a/\tau)} \tag{1}$$

where $A(i)$ is the set of other samples in the batch, $P(i)$ contains positive samples sharing the same class label as i, and τ is a temperature scalar. After training, the client distills its feature space into class-conditional statistics. At the second tier, the edge server m operates without access to real data. It aggregates the statistics $\{\mu_k, \sigma_k\}$ from its client cluster $\mathcal{C}_m$ to form a stable distribution (μ_m, σ_m). The edge server m unpacks these statistics to synthesize a new feature dataset, D_m^{syn}, which serves as the input for the edge-side part $W_{e,m}$. $W_{e,m}$ is trained using the same contrastive objective on its own outputs, $\mathcal{L}_m(W_{e,m}) = \mathcal{L}_{\text{con},m}(W_e(D_{syn}^{\text{edge}}))$. Enforcing consistency across tiers ensures that W_e learns relative structural relationships rather than overfitting to synthetic noise.

Finally, the cloud tier aggregates the class-conditional feature statistics from all M edge servers to form a global synthetic dataset, $D_{\text{syn}}^{\text{global}}$. The aggregated means and standard deviations are used to sample representative feature embeddings for each class. The task-specific part W_g is then trained on this synthesized dataset using the standard supervised loss $\mathcal{L}_g(W_g) = \mathcal{L}_{\text{task}}\big(W_g(D_{syn}^{\text{global}})\big)$.

3.3 Dual-Timescale Transmission and Augmentation

Our communication mechanism operates on a dual-timescale to balance efficiency with long-term stability. At the fast timescale, clients transmit only their class-wise prototype statistics, including the local $\mu_k^{(j)}$ and $\sigma_k^{(j)}$ of class j:

$$\mu_k^{(j)} = \frac{1}{|\mathcal{D}_k^{(j)}|} \sum_{x_i \in \mathcal{D}_k^{(j)}} W_{c,k}(x_i), \quad \sigma_k^{(j)} = \sqrt{\frac{1}{|\mathcal{D}_k^{(j)}|} \sum_{x_i \in \mathcal{D}_k^{(j)}} (W_{c,k}(x_i) - \mu_k^{(j)})^2}. \tag{2}$$

This drastically reduces overhead to $O(J_k \times d_c)$ per client, where J_k is the class count and d_c the feature dimension. Infrequently (slow timescale interval I_c), a standard model averaging of $W_{c,k}$ occurs to correct long-term client drift.

Upon receiving (μ_k, σ_k), the edge server synthesizes features z_{syn} for class j by sampling from a multivariate Gaussian as in [17, 41] as follows:

$$z_{syn} \sim \mathcal{N}\big(\mu_m^{(j)}, \text{diag}(\sigma_m^{(j)} + \epsilon)^2\big) \tag{3}$$

where diag(·) creates a diagonal covariance matrix and ϵ ensures numerical stability. We intentionally restrict the synthesis to a diagonal covariance matrix to achieve an optimal trade-off. Transmitting a full covariance matrix would increase the communication cost to $O(d_c^2)$, recreating the SL bottleneck. However, our ablation studies in Section 4.5 confirm that transmitting per-dimension variance (σ) rather than just the mean (μ) is critical for capturing the elliptical boundaries of feature clusters. Unlike SFL, which transmits exact, invertible activation maps for every sample, H-SFP transmits aggregated, class-wise statistics that obfuscate individual features, serving as a natural defense against feature-space inversion attacks while remaining semantically rich.

3.4 Theoretical Justification: Information Preservation

A fundamental concern with generative prototyping is the potential loss of high-rank structural information when replacing raw activations with (μ, σ). We theoretically bound this distribution shift between the true client representations H_{true} and the edge-synthesized features $\hat{H}_{gen}$ using Maximum Mean Discrepancy (MMD). Let $\mathcal{X}$ be the feature space defined by W_c. Given a characteristic kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ mapping to a Reproducing Kernel Hilbert Space (RKHS), the empirical MMD between the true feature distribution $\mathbb{P}$ and the synthesized Gaussian distribution $\mathbb{Q}$ is bounded by:

$$\text{MMD}^2(\mathbb{P}, \mathbb{Q}) = \mathbb{E}_{x,x' \sim \mathbb{P}}[k(x, x')] + \mathbb{E}_{y,y' \sim \mathbb{Q}}[k(y, y')] - 2\mathbb{E}_{x \sim \mathbb{P}, y \sim \mathbb{Q}}[k(x, y)]. \quad (4)$$

To ensure $\text{MMD}^2(\mathbb{P}, \mathbb{Q}) \to 0$ and that the synthesized geometry preserves the semantic manifold, we introduce a statistical constraint on the feature extractor.
Assumption 1 (Feature-Space Lipschitz Continuity): There exists a constant $L_{stat} > 0$ such that for any two data samples x_1, x_2, the divergence of their generated prototype statistics is bounded by the input space distance:

$$||\mu(x_1) - \mu(x_2)||_2 + ||\sigma(x_1) - \sigma(x_2)||_2 \leq L_{stat}||x_1 - x_2||_2. \quad (5)$$

By enforcing Assumption 1 during local client updates via the contrastive loss, the divergence between the actual and synthesized manifolds remains strictly bounded across aggregation rounds, preventing information loss.

3.5 Asymptotic Convergence Analysis

Establishing end-to-end convergence guarantees for the H-SFP topology requires departing from standard federated optimization proofs. Because our framework decouples the computational tiers, optimization at the upper tiers is performed over a dynamically shifting probability measure space parameterized by the tier below. To bound this shifting measure, we adopt standard stochastic non-convex assumptions along with a novel measure-theoretic constraint.
Assumption 2 (Measure Theoretic Lipschitz Continuity). Let $\mathcal{P}(\mathcal{Z})$ denote the space of synthesized prototype measures. The expected loss functional $\mathcal{L}$ is L_m

Lipschitz continuous with respect to the Wasserstein 2 distance $\mathcal{W}_2$ between generated distributions $\mathbb{Q}_1, \mathbb{Q}_2 \in \mathcal{P}(\mathcal{Z})$:

$$|\mathbb{E}_{z\sim\mathbb{Q}_1}[\mathcal{L}(W;z)] - \mathbb{E}_{z\sim\mathbb{Q}_2}[\mathcal{L}(W;z)]| \leq L_m \cdot \mathcal{W}_2(\mathbb{Q}_1, \mathbb{Q}_2). \tag{6}$$

Assumption 3 (Bounded Variances and Second Moments). The variance and second moments of stochastic gradients for each layer l have upper bounds, denoted by σ_l^2 and G_l^2, respectively. Operating under these constraints, we formally establish the global convergence of the decoupled architecture. The rigorous measure-theoretic derivation is deferred to the Appendix C.

Theorem 1 (Asymptotic Convergence of H-SFP). Under standard non-convex assumptions and Assumptions 2 and 3, the decoupled parameters asymptotically converge to a bounded stationary neighborhood. Specifically, for the global classifier W optimized over R total communication rounds with learning rate η, local aggregation intervals I_m, and layer split boundaries L_m, the expected gradient norm satisfies

$$\frac{1}{R}\sum_{t=1}^{R}\mathbb{E}[||\nabla\mathcal{L}(\overline{W}^{t-1})||_2^2] \leq \frac{2\Delta}{\eta R} + \frac{L\eta\sum_{l=1}^{L_{total}}\sigma_l^2}{N} + 4L^2\eta^2\sum_{m=1}^{M-1}\left(\mathbb{I}_{\{I_m>1\}}I_m^2\sum_{l=L_{m-1}+1}^{L_m}G_l^2\right) \tag{7}$$

where $\Delta = \mathcal{L}(\overline{W}^0) - \mathcal{L}^*$, and $\mathbb{I}_{\{\cdot\}}$ is the indicator function.

Proof Sketch (Hierarchical Measure Stabilization). The proof proceeds inductively across the hierarchy. The induction initiates at the client tier, where the local extractors converge modulo standard SGD dynamics. The divergence between the aggregated sub-model and local sub-models is strictly bounded by the aggregation interval I_m and the second moments G_l^2. Given L_f Lipschitz embeddings, the sequence of client output measures stabilizes, driving the expected parametric shift to zero. As the edge model W_e optimizes over the synthesized measure $\mathbb{Q}_e^t$, its temporal update incurs a stochastic drift bounded by $\delta_m^t = L_m \cdot \mathcal{W}_2(\mathbb{Q}_e^{t+1}, \mathbb{Q}_e^t)$. Because the underlying client-tier measures have stabilized, this drift term rigorously decays $\delta_m^t \to 0$, guaranteeing that W_e converges to a stationary neighborhood despite the non-stationary nature of its input. Applying identical inductive logic to the cloud tier, the stabilization of the edge measures guarantees the convergence of the global synthetic distribution.

4 Experiments

H-SFP is evaluated on CIFAR-10/100 [10], HAM10000 [31], ImageNet-1K [6], and ISIC-2018 [4] against FedAvg, FedProx, FedProto, HierFL, SplitFed, and HSFL baselines. HSFL serves as our primary architectural baseline. FedAvg and FedProx are included to benchmark standard and regularized parameter-based

synchronization. We select FedProto and FedGen to compare our generative synthesis against state-of-the-art prototype-based and data-free distillation methods. To address system heterogeneity, we compare against SplitFed and HeteroSFL. HierFL and HSFL serve as our primary architectural baselines to isolate the specific benefits of our prototype-driven coordination within hierarchical topologies. Our setup comprises 200 clients, partitioned across 10 edge servers, for all hierarchical methods. All models use the Adam optimizer ($lr = 1e-4$), and we report classification accuracy, segmentation performance, communication overhead, and client-side RAM usage. Specific hyperparameters, the full experimental setup, and additional results are detailed in the Appendix D.

4.1 Main Results: Model Performance

Table 1: Accuracy (%) on CIFAR-10 and CIFAR-100 under IID and non-IID (Dirichlet) settings across 200 distributed devices. Results are averaged over 5 independent runs, showing mean ± std. (A), (B), and (C) represent client-edge (I_c) and edge-cloud (I_e) aggregation intervals of (5, 10), (10, 20), and (25, 50) local epochs, respectively. **Bold** indicates best federated performance; underline indicates second-best.

Method	CIFAR-10			CIFAR-100		
	IID	$dir(0.3)$	$dir(0.7)$	**IID**	$dir(0.3)$	$dir(0.7)$
Centralized	94.20 ± 0.35	N/A	N/A	75.60 ± 0.45	N/A	N/A
FedAvg	57.25 ± 1.24	12.15 ± 2.10	23.60 ± 1.85	48.85 ± 1.40	2.10 ± 0.85	5.40 ± 1.15
FedSGD	18.80 ± 1.10	5.80 ± 0.95	6.40 ± 0.80	10.75 ± 0.90	1.80 ± 0.40	4.35 ± 0.65
FedNova	23.50 ± 1.15	14.20 ± 1.50	18.35 ± 1.45	50.24 ± 1.35	3.15 ± 0.75	6.80 ± 0.95
FedProx	58.25 ± 1.20	16.50 ± 1.80	25.40 ± 1.60	52.75 ± 1.25	4.05 ± 0.85	7.20 ± 1.10
FedProto	55.10 ± 1.05	38.50 ± 1.45	45.20 ± 1.30	46.50 ± 1.15	6.50 ± 0.95	8.80 ± 1.05
HierFL	62.50 ± 1.30	13.10 ± 1.90	23.60 ± 1.75	30.15 ± 1.45	1.95 ± 0.60	4.78 ± 0.85
SplitFed	61.85 ± 1.25	31.20 ± 1.65	39.95 ± 1.40	35.00 ± 1.30	4.15 ± 0.80	7.05 ± 1.00
HeteroSFL	63.45 ± 1.10	33.80 ± 1.45	42.60 ± 1.30	41.50 ± 1.25	5.45 ± 0.70	8.20 ± 0.95
FedGen	59.80 ± 1.10	37.90 ± 1.50	46.10 ± 1.25	51.20 ± 1.25	6.70 ± 0.85	9.15 ± 1.10
FedDF	61.50 ± 1.15	32.00 ± 1.60	40.50 ± 1.35	53.10 ± 1.20	4.50 ± 0.70	7.80 ± 1.05
HSFL (A)	51.35 ± 1.10	17.80 ± 1.75	26.54 ± 1.50	35.47 ± 1.35	5.10 ± 0.80	8.25 ± 1.15
HSFL (B)	50.80 ± 1.05	17.15 ± 1.65	25.90 ± 1.45	33.58 ± 1.30	4.80 ± 0.75	7.65 ± 1.10
HSFL (C)	49.18 ± 1.15	16.90 ± 1.70	25.40 ± 1.55	32.65 ± 1.40	4.10 ± 0.85	6.95 ± 1.05
Ours (A)	**67.44 ± 0.85**	**40.15 ± 1.25**	**48.85 ± 1.10**	**55.10 ± 0.95**	**7.15 ± 0.65**	**10.48 ± 0.85**
Ours (B)	66.80 ± 0.80	39.80 ± 1.20	48.15 ± 1.05	54.25 ± 0.90	6.80 ± 0.60	9.95 ± 0.80
Ours (C)	65.44 ± 0.90	39.10 ± 1.30	47.50 ± 1.15	53.65 ± 1.00	6.15 ± 0.70	9.05 ± 0.90

Tables 1 and 2 summarize the classification accuracy across varying datasets, backbones, and statistical distributions. Across all configurations, our proposed H-SFP framework consistently yields the highest performance.

Mitigating severe client drift. H-SFP shows significant resilience in highly skewed non-IID scenarios. As shown in Table 1, standard aggregation methods such as FedAvg exhibit a sharp performance drop under extreme data het-

Table 2: Accuracy (%) on HAM10000 and ImageNet-1K under different backbones and data distributions. Results are averaged over 5 runs across 200 clients (mean ± std). (A), (B), and (C) denote different client-edge and edge-cloud aggregation intervals. **Bold** and underline indicate the best and second-best federated results, respectively.

Method	HAM10000			ImageNet-1K		
	ResNet-50 (IID)	**VGG-16 (IID)**	**ResNet-50 ($dir(0.7)$)**	**ResNet-50 (IID)**	**VGG-16 (IID)**	**ResNet-50 ($dir(0.7)$)**
Centralized	86.40 ± 0.60	84.20 ± 0.55	N/A	75.8 ± 0.2	72.4 ± 0.3	N/A
FedAvg	68.28 ± 1.45	65.15 ± 1.50	13.45 ± 1.65	23.1 ± 1.4	21.4 ± 1.5	11.2 ± 1.7
FedSGD	25.85 ± 1.20	22.40 ± 1.10	6.34 ± 1.05	12.5 ± 1.1	10.2 ± 1.2	5.4 ± 1.1
FedNova	69.60 ± 1.35	67.30 ± 1.45	14.35 ± 1.50	23.8 ± 1.3	22.1 ± 1.4	12.5 ± 1.6
FedProx	71.15 ± 1.30	68.85 ± 1.40	15.45 ± 1.55	24.1 ± 1.3	22.5 ± 1.4	13.8 ± 1.6
FedProto	67.20 ± 1.25	64.50 ± 1.35	35.00 ± 1.40	25.2 ± 1.2	23.8 ± 1.3	18.5 ± 1.4
HierFL	69.50 ± 1.25	67.10 ± 1.30	22.40 ± 1.55	24.5 ± 1.2	22.8 ± 1.3	13.8 ± 1.5
SplitFed	65.78 ± 1.35	62.90 ± 1.40	25.15 ± 1.45	25.5 ± 1.1	23.4 ± 1.2	14.1 ± 1.5
HeteroSFL	67.10 ± 1.20	64.80 ± 1.25	28.50 ± 1.35	26.3 ± 0.8	24.2 ± 0.9	16.2 ± 1.2
FedGen	72.50 ± 1.15	69.40 ± 1.20	35.80 ± 1.30	26.8 ± 1.0	24.5 ± 1.1	19.3 ± 1.3
FedDF	73.10 ± 1.20	70.20 ± 1.30	27.80 ± 1.45	25.9 ± 1.1	23.9 ± 1.2	15.6 ± 1.4
HSFL (A)	68.82 ± 1.25	66.15 ± 1.35	30.45 ± 1.40	24.6 ± 0.9	22.5 ± 1.0	14.8 ± 1.1
HSFL (B)	67.74 ± 1.20	65.20 ± 1.25	28.83 ± 1.35	24.1 ± 0.9	22.1 ± 1.0	13.9 ± 1.2
HSFL (C)	66.98 ± 1.30	64.50 ± 1.30	27.95 ± 1.45	23.5 ± 1.0	21.6 ± 1.1	13.1 ± 1.2
Ours (A)	**80.86 ± 0.75**	**78.45 ± 0.80**	**37.50 ± 1.05**	**27.9 ± 0.3**	**25.3 ± 0.4**	**22.4 ± 0.6**
Ours (B)	79.96 ± 0.70	77.50 ± 0.75	36.90 ± 1.00	27.4 ± 0.4	24.8 ± 0.5	21.8 ± 0.7
Ours (C)	79.15 ± 0.80	76.80 ± 0.85	36.15 ± 1.15	26.8 ± 0.5	24.2 ± 0.6	21.1 ± 0.8

erogeneity, falling to 12.15% on CIFAR-10 ($dir(0.3)$). H-SFP (A) effectively mitigates client drift, achieving 40.15% and outperforming specialized state-of-the-art methods, including FedProto (38.50%) and FedGen (37.90%). This confirms that synthesizing global prototypes at the edge and cloud tiers provides a stronger regularizing effect than flat architectural baselines. Furthermore, the varying aggregation intervals (A, B, C) indicate a direct correlation between edge-cloud synchronization frequency and final accuracy.

Scaling to large-scale vision tasks. To verify that our feature synthesis does not destroy high-rank semantic information, we extended our evaluation to HAM10000 and ImageNet-1K (Table 2). Under a strict 200-round budget across 200 devices, H-SFP (A) achieves 80.86% on HAM10000 (IID, ResNet-50), closely trailing the unconstrained Centralized Oracle (86.40%). On ImageNet-1K, H-SFP (A) reaches 27.9%, outperforming advanced baselines including HeteroSFL (26.3%) and SplitFed (25.5%). This demonstrates that transmitting lightweight Gaussian statistics is sufficient to preserve the complex feature manifolds required for standard ResNet and VGG backbones.

Application to dense medical segmentation. We further evaluate the framework's versatility on the ISIC-2018 skin lesion segmentation task depicted in Table 3. Dense prediction tasks expose the primary vulnerability of split learning: transmitting high-dimensional spatial feature maps creates severe commu-

Table 3: Segmentation performance on the ISIC-2018 dataset using a ResNet50-UNet backbone. All models were trained under a strict constraint of 200 communication rounds across 200 distributed edge devices. **Bold** indicates the best federated performance, and underline indicates the second-best.

Method	IoU (%) ↑	DICE (%) ↑	Training Time (h) ↓
Centralized	78.5 ± 0.4	86.2 ± 0.3	5.2 ± 0.1
FedAvg	48.7 ± 1.2	64.2 ± 1.5	28.5 ± 0.8
HierFL	51.2 ± 1.1	66.3 ± 1.3	24.3 ± 0.7
SplitFed	53.8 ± 0.9	68.1 ± 1.1	72.4 ± 1.6
HeteroSFL	56.4 ± 0.7	70.5 ± 0.8	45.8 ± 1.2
HSFL (A)	62.1 ± 0.8	74.8 ± 0.9	14.5 ± 0.5
Ours (A)	**68.3** ± 0.5	**79.5** ± 0.4	**6.8** ± 0.2

nication bottlenecks. Consequently, SplitFed requires 72.4 hours to reach 53.8% IoU. While weight-averaging methods like FedAvg complete training faster (28.5 hours), they stall at 48.7% IoU due to unmitigated client drift. H-SFP addresses this bottleneck directly by synthesizing the bottleneck features for the cloud-based decoder rather than transmitting raw activations. As a result, H-SFP achieves 68.3% IoU and 79.5% DICE in only 6.8 hours, delivering a 10× training speedup over SplitFed and a 19.6% absolute IoU improvement over FedAvg.

4.2 Analysis of Communication Overhead

Figure 4 provides a detailed breakdown of the communication overhead, quantifying the primary advantage of H-SFP. As shown, traditional FL methods like FedAvg are prohibitively expensive (∼3,865 GB) due to full model transfers, and HierFL incurs even higher costs (4,392.44 GB). While hybrid methods such as HSFL (113.12 GB) and prototype-based FedProto (17.25 GB) offer significant savings, our H-SFP framework yields the most dramatic reduction. Specifically, our most efficient configuration (Ours (C)) requires only 10.94 GB. This represents a 353× reduction compared to FedAvg, a 10.4× reduction over the HSFL baseline, and a 1.58× improvement over the strong FedProto baseline. These gains are directly attributable to our novel architecture: H-SFP eliminates the need for gradient backpropagation (which consumes 43.47 GB in HSFL) and replaces the continuous transfer of raw activations (54.93 GB in HSFL) with lightweight feature statistics (9.77 GB), thereby effectively resolving the communication bottleneck.

To better understand the communication savings of H-SFP, we compare its communication complexity with that of standard FSL. In FSL, clients transmit intermediate activations during the forward pass and receive corresponding gradients during the backward pass. As a result, the communication cost per synchronization round scales linearly with the local batch size B and the intermediate feature dimension d, FSL Communication $= \mathcal{O}(B \cdot d)$. In contrast, H-SFP

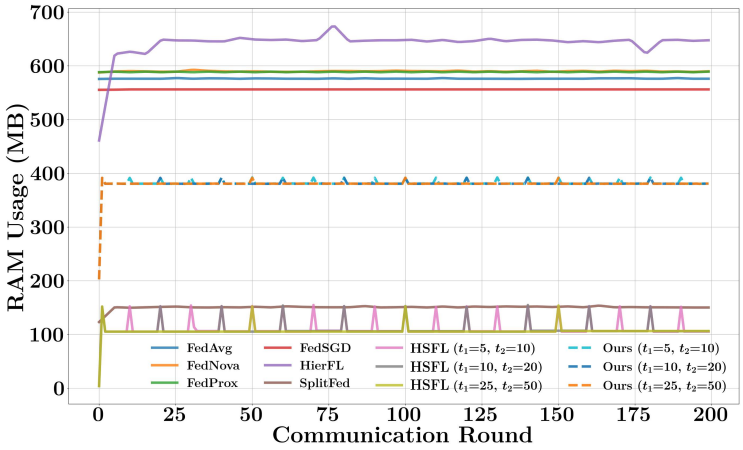


Fig. 3: Client-side RAM usage (MB) over 200 rounds. The chart contrasts the high, stable memory cost of full-model methods (∼600 MB) with the low, sub-100 MB cost of split-learning methods (SplitFed, HSFL, and Ours).

severs the cross-tier optimization graph, eliminating the need for sample-wise activation and gradient exchanges. Instead, clients in our framework compute and transmit only lightweight statistical moments (e.g., mean and variance) for each class present in their local data. Therefore, the communication overhead scales solely with the number of unique classes C and the feature dimension d, H-SFP Communication $= \mathcal{O}(C \cdot d)$. In typical distributed learning scenarios, the local batch size is significantly larger than the number of unique classes ($B \gg C$). Furthermore, because H-SFP utilizes these statistical moments for generative synthesis at the upper tiers, it completely circumvents the backward-pass communication phase required by FSL. This fundamental theoretical shift from sample-dependent $\mathcal{O}(B \cdot d)$ complexity to class-dependent $\mathcal{O}(C \cdot d)$ complexity provides the mathematical underpinning for the massive communication savings demonstrated in our empirical results.

4.3 Analysis of Client-Side Memory Cost

Figure 3 reveals a stark divide: traditional, full-model methods (e.g., FedAvg) require a stable ∼600 MB of RAM to load the *entire* model. In contrast, our H-SFP framework and other split-learning methods consume well under 100 MB. This dramatic reduction is due to H-SFP clients managing only the lightweight initial segment (W_c). This result confirms that ours is highly practical for resource-constrained clients for whom standard FL is infeasible.

4.4 Scalability Test

To evaluate robustness to data sparsity, we partition a fixed total dataset across 20 to 200 clients (Table 4). As expected, all methods degrade as client count in-

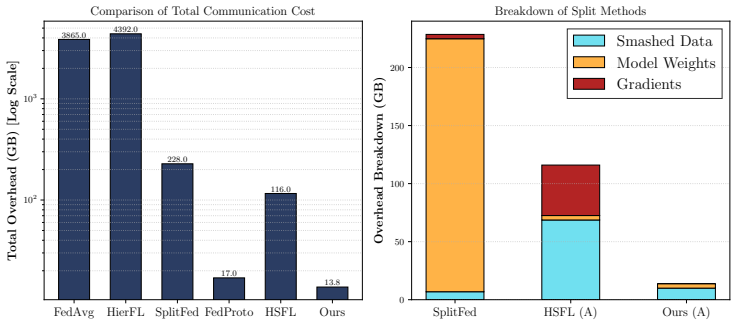


Fig. 4: Communication overhead analysis on CIFAR-100 over 200 rounds. **(Left)** Total communication cost across all methods plotted on a logarithmic scale. H-SFP (ours) reduces the total overhead by over 350× compared to standard FedAvg. **(Right)** A detailed breakdown of the communication components for the split-learning baselines. H-SFP's extreme efficiency stems from replacing high-dimensional smashed data with lightweight statistics and eliminating the need to transmit gradients.

Table 4: Accuracy (%) on CIFAR-100 and ImageNet-1K under the IID scenario with varying client cohort sizes. Results are averaged over 5 independent runs (mean ± std). (A), (B), and (C) represent client-edge and edge-cloud aggregation intervals. **Bold** indicates the best federated performance, and underline indicates the second-best.

Method	CIFAR-100				ImageNet-1K			
	200	100	50	20	200	100	50	20
FedAvg	48.85 ± 1.40	53.20 ± 1.25	57.10 ± 1.10	60.50 ± 0.90	23.1 ± 1.4	31.4 ± 1.2	42.5 ± 1.1	53.2 ± 0.9
FedSGD	10.75 ± 0.90	13.10 ± 0.85	15.80 ± 0.80	17.90 ± 0.75	12.5 ± 1.1	15.6 ± 1.0	21.4 ± 0.9	28.5 ± 0.8
FedNova	50.24 ± 1.35	54.80 ± 1.20	58.90 ± 1.05	61.70 ± 0.85	23.8 ± 1.3	32.5 ± 1.2	43.8 ± 1.0	55.1 ± 0.9
FedProx	52.75 ± 1.25	57.00 ± 1.15	60.90 ± 1.00	63.80 ± 0.85	24.1 ± 1.3	32.8 ± 1.1	44.2 ± 1.0	55.8 ± 0.8
FedProto	49.50 ± 1.15	53.80 ± 1.05	57.90 ± 0.95	61.20 ± 0.80	25.2 ± 1.2	34.5 ± 1.1	46.1 ± 0.9	57.5 ± 0.8
HierFL	50.15 ± 1.30	54.50 ± 1.15	58.40 ± 1.05	61.80 ± 0.90	24.5 ± 1.2	33.6 ± 1.1	45.4 ± 1.0	56.7 ± 0.8
SplitFed	51.00 ± 1.20	55.40 ± 1.10	59.30 ± 0.95	62.50 ± 0.85	25.5 ± 1.1	34.8 ± 1.0	46.5 ± 0.9	58.1 ± 0.7
HeteroSFL	52.10 ± 1.15	56.50 ± 1.05	60.50 ± 0.90	63.80 ± 0.80	26.3 ± 0.8	36.1 ± 0.8	48.2 ± 0.7	60.5 ± 0.6
FedGen	51.20 ± 1.25	55.45 ± 1.15	59.35 ± 1.00	62.25 ± 0.85	26.8 ± 1.0	36.5 ± 0.9	48.8 ± 0.8	61.2 ± 0.7
FedDF	53.10 ± 1.20	57.35 ± 1.10	61.25 ± 0.95	64.15 ± 0.85	25.9 ± 1.1	35.2 ± 1.0	47.1 ± 0.9	59.5 ± 0.7
HSFL (A)	52.80 ± 1.10	57.10 ± 1.00	61.00 ± 0.90	64.00 ± 0.75	24.6 ± 0.9	33.8 ± 0.8	45.7 ± 0.7	57.2 ± 0.6
HSFL (B)	52.10 ± 1.05	56.30 ± 0.95	60.20 ± 0.85	63.20 ± 0.75	24.1 ± 0.9	33.2 ± 0.8	44.9 ± 0.7	56.3 ± 0.6
HSFL (C)	51.40 ± 1.10	55.50 ± 1.00	59.30 ± 0.90	62.30 ± 0.80	23.5 ± 1.0	32.5 ± 0.9	44.1 ± 0.8	55.4 ± 0.7
Ours (A)	**55.10 ± 0.95**	**60.20 ± 0.85**	**64.50 ± 0.75**	**67.80 ± 0.60**	**27.9 ± 0.3**	**38.4 ± 0.3**	**51.5 ± 0.3**	**64.8 ± 0.2**
Ours (B)	54.25 ± 0.90	59.40 ± 0.80	63.80 ± 0.70	66.90 ± 0.55	27.4 ± 0.4	37.8 ± 0.4	50.7 ± 0.4	63.9 ± 0.3
Ours (C)	53.65 ± 1.00	58.70 ± 0.90	62.90 ± 0.80	65.90 ± 0.65	26.8 ± 0.5	37.1 ± 0.5	49.8 ± 0.4	62.8 ± 0.3

creases, since less data per client leads to significant "client drift". For instance, FedAvg on CIFAR-100 drops from 60.50% (20 clients) to 48.85% (200 clients). Specifically, H-SFP (ours) consistently outperforms all baselines, including the strong FedGen and FedDF methods, at all client counts. This superiority is most evident in the high-sparsity (200 clients) scenarios. Here, Ours (A) leads the next-best baseline by up to +2.00% on CIFAR-100 and +1.1% on ImageNet-1K. This validates our architecture's resilience. While baselines like FedAvg suffer

Table 5: Ablation study on the components of H-SFP, reporting Accuracy (%) at client-edge interval $I_c = 5$ and edge-cloud interval $I_e = 10$ across 200 distributed devices. The results show the incremental performance gains achieved by adding a contrastive loss and synthetic data. We validate our core design choices by comparing the **Full H-SFP** against two key ablations: a **Full (μ-only)** variant (lacking dimension-aware variance) and a **Full (Purely Generative)** variant without averaging.

Method	**CIFAR-100**		**ImageNet-1K (ResNet-50)**	
	IID	$dir(0.7)$	**IID**	$dir(0.7)$
Baseline (HSFL, no prototyping)	35.47 ± 1.35	8.25 ± 1.15	24.6 ± 0.9	14.8 ± 1.1
+ Contrastive Loss (CL)	41.20 ± 1.25	8.95 ± 1.05	25.4 ± 0.8	16.3 ± 1.0
+ Synthetic Data (SD)	47.60 ± 1.15	9.45 ± 0.95	26.1 ± 0.7	18.2 ± 0.9
Full (μ-only)	52.35 ± 1.05	9.85 ± 0.90	26.8 ± 0.5	20.5 ± 0.8
Full (Purely Generative, no model avg.)	48.20 ± 1.10	9.15 ± 1.00	25.8 ± 0.7	19.4 ± 0.9
Full H-SFP (μ + σ + model avg.)	**55.10 ± 0.95**	**10.48 ± 0.85**	**27.9 ± 0.3**	**22.4 ± 0.6**

Table 6: Ablation on the temperature hyperparameter τ of H-SFP. Results show the Accuracy (%) at client-edge interval $I_c = 5$ and edge-cloud interval $I_e = 10$. Results are averaged over 5 independent runs (mean ± std). The optimal performance is consistently achieved at $\tau = 0.5$.

τ	**CIFAR-10**		**CIFAR-100**		**HAM10000 (ResNet-50)**	
	IID	$dir(0.7)$	**IID**	$dir(0.7)$	**IID**	$dir(0.7)$
0.2	55.30 ± 1.15	36.20 ± 1.35	48.15 ± 1.20	6.20 ± 0.95	71.42 ± 1.10	30.55 ± 1.25
0.5	**67.44 ± 0.85**	**48.85 ± 1.10**	**55.10 ± 0.95**	**10.48 ± 0.85**	**80.86 ± 0.75**	**37.50 ± 1.05**
1.0	63.15 ± 0.95	43.10 ± 1.20	52.88 ± 1.05	8.50 ± 0.90	77.40 ± 0.90	34.70 ± 1.15
1.5	60.03 ± 1.05	41.50 ± 1.25	51.10 ± 1.10	7.20 ± 0.95	75.85 ± 0.95	33.10 ± 1.20
2.0	56.38 ± 1.10	38.40 ± 1.30	48.65 ± 1.15	6.05 ± 1.00	72.12 ± 1.05	30.45 ± 1.25

from aggregating poorly trained, high-variance weights, our generative process uses lightweight feature statistics. This allows edge servers to synthesize a stable cluster-level distribution, effectively filtering noise and mitigating the impact of client drift. Finally, (A) with more frequent aggregation intervals (5, 10) performs best, as it corrects for client drift. We note that this experiment tests robustness to data sparsity under IID conditions. An analysis of scalability combined with high heterogeneity remains a priority for future investigation.

4.5 Ablation Studies

Ablation on H-SFP components. Table 5 isolates the contribution of each algorithmic component. While adding Contrastive Loss (+CL) and Synthetic Data (+SD) provides steady incremental gains, the most critical insight lies in comparing the Full (μ-only) variant to the Full H-SFP model. The (μ-only) approach that resembles traditional prototype methods sacrifices up to 2.75% absolute accuracy on CIFAR-100 and 1.1% on ImageNet-1K. This empirical gap

directly validates Assumption 1 in Section 3.4: demonstrating that representing activations solely by their mean fails to preserve the geometry of structural features. Transmitting both μ and σ enables the edge and cloud servers to generate an elliptical boundary that accurately bounds the MMD divergence, driving the final performance of 55.10% and 27.9%, respectively.

Ablation on Temperature τ. We investigate the contrastive loss temperature τ in Table 6. The results show a clear "Goldilocks" effect. A low τ (e.g., 0.2) is too sharp and unstable, while a high τ (e.g., 2.0) is too soft and provides an insufficient learning signal. We find $\tau = 0.5$ consistently achieves the optimal accuracy across all datasets, balancing feature discrimination and training stability.

5 Limitations and Future Work

While our Gaussian feature synthesis effectively captures first- and second-order statistics, it may not fully represent multimodal feature distributions or cross-dimensional correlations. We therefore restrict synthesis to a diagonal covariance matrix to maintain communication efficiency and avoid the overhead associated with richer representations. Future work will investigate low-rank covariance approximations and more expressive distribution models that preserve the lightweight communication characteristics of H-SFP. Furthermore, our evaluation adopts a random client-to-edge assignment strategy. Exploring topology-aware clustering and broader forms of system heterogeneity remains an important direction for future study.

6 Conclusion

We presented H-SFP, a hierarchical federated learning framework that combines the computational advantages of split-model execution with a communication-efficient statistical interface. Instead of exchanging sample-level activations and gradients, H-SFP communicates compact class-wise feature statistics that enable upper tiers to synthesize representative feature distributions for training. This design reduces communication and synchronization overhead while preserving the practical benefits of hierarchical split learning. Through theoretical analysis and extensive experiments on image classification and medical image segmentation benchmarks, we showed that H-SFP remains effective under heterogeneous data distributions and achieves substantial communication savings compared with existing federated and split-learning approaches. These results suggest that lightweight first- and second-order feature statistics provide an effective communication mechanism for hierarchical federated learning.

Acknowledgements

This work was supported by the Green Serverless Computing for Resource-Efficient AI Training Project at VinUniversity under Grant VUNI.CEI.FS_0002.

References

1. Bensiah, O.A., Benaboud, R.: Feddpa: Dynamic prototypical alignment for federated learning with non-iid data. Electronics **14**(16), 3286 (2025)
2. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PmLR (2020)
3. Cheng, Z., Huang, X., Wu, P., Yuan, K.: Momentum benefits non-iid federated learning simply and provably. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=TdhkAcXkRi
4. Codella, N.C.F., Gutman, D., Celebi, M.E., Helba, B., Marchetti, M.A., Dusza, S.W., Kalloo, A., Liopyris, K., Mishra, N., Kittler, H., Halpern, A.: Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In: 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018). pp. 168–172 (2018). https://doi.org/10.1109/ISBI.2018.8363547
5. Dai, Y., Chen, Z., Li, J., Heinecke, S., Sun, L., Xu, R.: Tackling data heterogeneity in federated learning with class prototypes. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 7314–7322 (2023)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
7. Diao, E., Ding, J., Tarokh, V.: Heterofl: Computation and communication efficient federated learning for heterogeneous clients. In: 9th International Conference on Learning Representations, ICLR 2021 (2021)
8. Gupta, O., Raskar, R.: Distributed learning of deep neural network over multiple agents. Journal of Network and Computer Applications **116**, 1–8 (2018). https://doi.org/https://doi.org/10.1016/j.jnca.2018.05.003, https://www.sciencedirect.com/science/article/pii/S1084804518301590
9. Kairouz, P., McMahan, H.B., Avent, B., et al.: Advances and open problems in federated learning. Foundations and Trends® in Machine Learning **14**(1–2), 1–210 (2021). https://doi.org/10.1561/2200000083, http://dx.doi.org/10.1561/2200000083
10. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Tech. rep., University of Toronto (2009), https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf
11. Li, T., Sahu, A.K., Talwalkar, A., Smith, V.: Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine **37**(3), 50–60 (2020). https://doi.org/10.1109/MSP.2020.2975749
12. Lin, T., Kong, L., Stich, S.U., Jaggi, M.: Ensemble distillation for robust model fusion in federated learning. Advances in neural information processing systems **33**, 2351–2363 (2020)
13. Lin, Z., Wei, W., Chen, Z., Lam, C.T., Chen, X., Gao, Y., Luo, J.: Hierarchical split federated learning: Convergence analysis and system optimization. IEEE Transactions on Mobile Computing pp. 1–16 (2025). https://doi.org/10.1109/TMC.2025.3565509
14. Liu, L., Zhang, J., Song, S., Letaief, K.B.: Client-edge-cloud hierarchical federated learning. In: ICC 2020 - 2020 IEEE International Conference on Communications (ICC). pp. 1–6 (2020). https://doi.org/10.1109/ICC40277.2020.9148862

15. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.y.: Communication-Efficient Learning of Deep Networks from Decentralized Data. In: Singh, A., Zhu, J. (eds.) Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 54, pp. 1273–1282. PMLR (20–22 Apr 2017), https://proceedings.mlr.press/v54/mcmahan17a.html
16. Mu, Y., Shen, C.: Federated split learning with improved communication and storage efficiency. IEEE Transactions on Mobile Computing pp. 1–12 (2025). https://doi.org/10.1109/TMC.2025.3591744
17. Nguyen, M.D., Dao, T.T., Nguyen, L.T., Le, D.D., Wong, K.S.: Memory-efficient continual learning with prototypical exemplar condensation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings. pp. 7675–7685 (June 2026)
18. Nguyen, M.D., Do, H.K., Le, N.K., Tran, N.H., Yang, Z., Nguyen, V.D., Van Chien, T.: Towards layer-wise personalized federated learning: Adaptive layer disentanglement via conflicting gradients. IEEE Transactions on Networking (2026)
19. Nguyen, M.D., Lee, S.M., Pham, Q.V., Hoang, D.T., Nguyen, D.N., Hwang, W.J.: Hcfl: A high compression approach for communication-efficient federated learning in very large scale iot networks. IEEE transactions on mobile computing **22**(11), 6495–6507 (2022)
20. Nguyen, M.D., Wanasekara, S., Nguyen, L.T., Pham, Q.V., Yong, K.T., Tran, N.H., Le, D.D.: Computation and communication efficient federated unlearning via on-server gradient conflict mitigation and expression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3347–3357 (2026)
21. Nguyen, T.B., Nguyen, D.M., Park, J., Pham, V.Q., Hwang, W.J.: Federated domain generalization with data-free on-server matching gradient. In: The Thirteenth International Conference on Learning Representations (May 2025)
22. Nguyen, V.D., Chatzinotas, S., Ottersten, B., Duong, T.Q.: FedFog: Network-aware optimization of federated learning over wireless fog-cloud systems. IEEE Trans. Wire. Commun. **21**(10), 8581–8599 (2022). https://doi.org/10.1109/TWC.2022.3167263
23. Niu, Y., Ali, R.E., Prakash, S., Avestimehr, S.: All Rivers Run to the Sea: Private Learning with Asymmetric Flows . In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 12353–12362. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2024). https://doi.org/10.1109/CVPR52733.2024.01174, https://doi.ieeecomputersociety.org/10.1109/CVPR52733.2024.01174
24. Niu, Y., Prakash, S., Kundu, S., Lee, S., Avestimehr, S.: Overcoming resource constraints in federated learning: Large models can be trained with only weak clients. Transactions on Machine Learning Research (2023), https://openreview.net/forum?id=lx1WnkL9fk
25. Qiu, X., Leontiadis, I., Melis, L., Sablayrolles, A., Stock, P.: Evaluating privacy leakage in split learning. arXiv preprint arXiv:2305.12997 (2023)
26. Quan, M.K., Nguyen, D.C., Nguyen, V.D., Wijayasundara, M., Setunge, S., Pathirana, P.N.: Hiersfl: Local differential privacy-aided split federated learning in mobile edge computing. In: 2023 IEEE Virtual Conference on Communications (VCC). pp. 103–108. IEEE (2023)
27. Rana, O., Spyridopoulos, T., Hudson, N., Baughman, M., Chard, K., Foster, I., Khan, A.: Hierarchical and decentralised federated learning. In: 2022 Cloud Continuum. pp. 1–9. IEEE (2022)

28. Tan, Y., Long, G., Liu, L., Zhou, T., Lu, Q., Jiang, J., Zhang, C.: Fedproto: Federated prototype learning across heterogeneous clients. In: Proceedings of the AAAI conference on artificial intelligence. pp. 8432–8440 (2022)
29. Thapa, C., Arachchige, P.C.M., Camtepe, S., Sun, L.: Splitfed: When federated learning meets split learning. In: Proceedings of the AAAI conference on artificial intelligence. vol. 36, pp. 8485–8493 (2022)
30. Tran, T.K., Tran, H.P., Le, T.L., Tran, T.H.: Fedntproto: A prototype-based approach for personalized federated learning. In: 2024 International Conference on Multimedia Analysis and Pattern Recognition (MAPR). pp. 1–6 (2024). https://doi.org/10.1109/MAPR63514.2024.10660707
31. Tschandl, P., Rosendahl, C., Kittler, H.: The ham10000 dataset: A large collection of multi-source dermatoscopic images of common pigmented skin lesions. Scientific Data **5**, 180161 (2018). https://doi.org/10.1038/sdata.2018.161
32. Venkateswaran, P., Isahagian, V., Muthusamy, V., Venkatasubramanian, N.: Fedgen: Generalizable federated learning for sequential data. In: 2023 IEEE 16th International Conference on Cloud Computing (CLOUD). pp. 308–318. IEEE (2023)
33. Vepakomma, P., Gupta, O., Swedish, T., Raskar, R.: Split learning for health: Distributed deep learning without sharing raw patient data. In: AI for Social Good Workshop, International Conference on Learning Representations (ICLR) (2019)
34. Wang, J., Liu, Q., Liang, H., Joshi, G., Poor, H.V.: Tackling the objective inconsistency problem in heterogeneous federated optimization. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20, Curran Associates Inc., Red Hook, NY, USA (2020)
35. Wang, X., Wu, F., Miao, C., Li, T., Hu, H., Cao, Q., Gao, J., Su, L.: Towards privacy-preserving and heterogeneity-aware split federated learning via probabilistic masking. arXiv preprint arXiv:2509.14603 (2025)
36. Wang, Z., Xu, H., Liu, J., Huang, H., Qiao, C., Zhao, Y.: Resource-efficient federated learning with hierarchical aggregation in edge computing. In: IEEE INFOCOM 2021 - IEEE Conference on Computer Communications. pp. 1–10 (2021). https://doi.org/10.1109/INFOCOM42981.2021.9488756
37. Wong, K.S., Nguyen, D.M., Le, K., Ho, L., Do, C., Le-Phuoc, D.: An empirical study of federated learning on iot–edge devices: Resource allocation and heterogeneity. IEEE Transactions on Neural Networks and Learning Systems **37**(2), 753–765 (2026). https://doi.org/10.1109/TNNLS.2025.3611415
38. Zaccone, R., Karimireddy, S.P., Masone, C., Ciccone, M.: Communication-efficient heterogeneous federated learning with generalized heavy-ball momentum. Transactions on Machine Learning Research (2025), https://openreview.net/forum?id=LNoFjcLywb
39. Zhang, M., Sapra, K., Fidler, S., Yeung, S., Alvarez, J.M.: Personalized federated learning with first order model optimization. arXiv preprint arXiv:2012.08565 (2020)
40. Zhang, Z., Wong, L., Varghese, B.: Ampere: Communication-efficient and high-accuracy split federated learning (2025), https://arxiv.org/abs/2507.07130
41. Zhu, F., Zhang, X.Y., Wang, C., Yin, F., Liu, C.L.: Prototype augmentation and self-supervision for incremental learning. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5867–5876 (2021). https://doi.org/10.1109/CVPR46437.2021.00581