FEDSAM: HIERARCHICAL STRUCTURE ADAPTATION FOR MODEL-HETEROGENEOUS FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

In real-world Federated Learning (FL) settings, clients often differ in both computational resources and local data characteristics. Model-heterogeneous FL addresses this issue by assigning clients sub-models of different-sized network structures partitioned from a single large neural network. However, existing methods typically rely on heuristic partitioning criteria and do not allow clients to customize their sub-models to local data's unique statistical properties. In this paper, we propose FedSAM, a novel hierarchical Bayesian framework that enables probabilistic inference on client-specific network structures while sharing collaborative knowledge across the local models in the hierarchy. Specifically, in the framework we adapt a local model's neural network structure by modeling network depth and width as beta and conjugate Bernoulli processes. We further specify a global-level beta process as a hyper-prior to aggregate and share local models' structural information at central server. We develop a structured variational inference scheme for efficient computation. Extensive empirical studies demonstrate that by adapting local models' structures to non-IID data and sharing the structural information, our framework yields state-of-the-art performance.

1 INTRODUCTION

In real-world Federated Learning (FL) settings, client devices often have diverse computational resources and memory capacities (Diao et al., 2021; Li et al., 2020). Due to the limited computational capability of many client devices, it is often infeasible to train a homogeneous large model on all clients (Kairouz and McMahan, 2021). Model-heterogeneous FL addresses this issue by partitioning a global model into a set of sub-models with distinct network capacities and assigning them to clients for local training. These methods heuristically pre-specify partitioning criteria through width scaling, depth scaling, their combinations, or parameter importance-based selection, without considering the heterogeneous statistical properties of each client's local data (Wang et al., 2023; Wu et al., 2024; Zhang et al., 2025; Hu et al., 2026). Since the sub-models with fixed network structures created by this generic strategy often result in performance bottleneck (Ye et al., 2023; Ma et al., 2026), it highlights the need that clients can adapt their local model structures to local data characteristics while collaboratively sharing both network parameters and network structural information.

To address this challenge, we propose a novel hierarchical Bayesian Structure Adaptation Method (FedSAM) for model-heterogeneous FL that not only enables clients' local models to adapt their network structures to unique data characteristics but also share the structural statistics at the global level in the hierarchy. Specifically, we model a local neural network depth with a beta process (Hjort, 1990; Broderick et al., 2012) that allows layer number to go to infinity in theory, and prune network width using a conjugate Bernoulli process. A local model can thus tailor its neural network capacity by inferring the most plausible network structure warranted by its unique local data. At the global level, we define a base measure beta process as a hyper-prior (Thibaux and Jordan, 2007), as illustrated in Figure 1. This hyper-prior enables aggregating and sharing network structural information among local models at central server while allowing them to retain their unique structural components. We further develop an efficient structured variational inference strategy that perform joint Bayesian inference at both the local and global levels. Our theoretical and empirical analysis show that by adapting local models' network structures to non-IID data and sharing complementary

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.

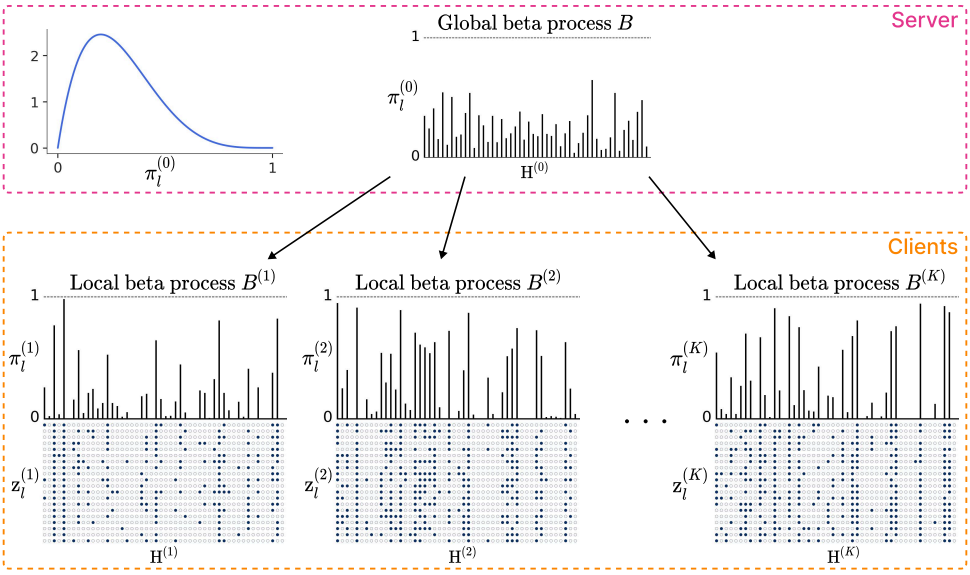

Figure 1: Demonstration of hierarchical beta processes. **Top row** shows the projection of a base measure B_0 in $[0, 1]$ (Left) and a derived hyper-prior $B = \sum_l \pi_l^{(0)} \delta_{\mathbf{h}_l^{(0)}}$ over hidden layer functional space $\mathbf{H^{(0)}} = \{\mathbf{h}_l^{(0)} | l \to \infty\}$ (Right), where $\pi_l^{(0)}$ denotes layer-wise activation probability, sharing complementary statistical information at the global level. **Bottom row** shows local beta process priors $B^{(k)}$ randomly drawn from B along with their conjugate Bernoulli process $\mathbf{Z}^{(k)}$, regularizing client k's local network structures to capture its distinctive characteristics, respectively. $\pi_l^{(k)}$ denotes local layer-wise activation probability. Solid dots denote $z_{ml}^{(k)} = 1$, and empty dots denote $z_{ml}^{(k)} = 0$.

structural information, FedSAM achieves state-of-the-art performance across various benchmark datasets with different types of backbone networks.

2 Related Works

Model-Heterogeneous Federated Learning: Model-heterogeneous FL mitigates resource-induced client exclusion by enabling resource-constrained clients to train smaller, capacity-matched sub-models, rather than requiring all clients to use the same homogeneous model. Existing approaches typically construct client-specific sub-models from a global model using predefined heuristics. Early approaches obtain sub-models by randomly dropping portions of the global model (Caldas et al., 2018), while width-scaling based methods construct local models by retaining only a subset of the global model's width (Horvath et al., 2021; Hong et al., 2022). HeteroFL (Diao et al., 2021) assigns clients varying-width sub-models according to their resource budgets. FedRolex (Alam et al., 2022) improves parameter coverage through a rolling sub-model extraction scheme. Subsequent methods consider additional model dimensions for constructing heterogeneous client models (Liu et al., 2022; Liao et al., 2023). DepthFL (Kim et al., 2022) partitions the global model along the depth dimension, while ScaleFL (Ilhan et al., 2023) jointly adapts width and depth according to client resources. FIARSE (Wu et al., 2024) uses parameter values to estimate importance and guide sub-model extraction. Built upon FIARSE, importance-aware approaches extract sub-models based on parameter-importance estimates (Zhang et al., 2025; Hu et al., 2026).

Bayesian (Personalized) Federated Learning: Bayesian inference has been used to construct global posterior distribution over model parameters from local estimates (Thorgeirsson and Gauterin, 2020; Chen and Chao, 2021; Al-Shedivat et al., 2021; Yurochkin et al., 2019; Talpini et al., 2025). Another line of work leverages Gaussian Processes (Yin et al., 2020; Achituve et al., 2021; Tang et al., 2022), mixed-effects models (Kotelevskii et al., 2022), and functional-space priors (Makhija et al., 2024) to improve performance. Bayesian personalized FLs model client-specific posterior distributions only over local parameters with homogeneous network structures across

clients (Zhang et al., 2022; Kotelevskii et al., 2022; Zhu et al., 2023; Bhatt et al., 2024; Wei et al., 2025; Mildner et al., 2025; Thapa and Li, 2025).

Stochastic Process Network Adaptation: Stochastic processes have been proposed to adapt single network structures (KC et al., 2021; Kessler et al., 2021; Thapa and Li, 2024) for model regularization and continual learning. However, these methods lack a mechanism for sharing complementary structural information across multiple models. Our ablation study shows independently adapting local model structures leads to inferior performance in FL.

3 Hierarchical Bayesian Inference for Model-heterogeneous FL

In FedSAM, while local beta processes adapt local network structures fitting network parameters to local data, the statistical information of these local structures are shared via a global beta process, as in Figure 1. To perform joint approximate inference, we present a structured variational inference scheme that splits the overall objective into a server-level objective for aggregation and a client-level objective for local adaptation.

3.1 Local Models with Adaptive Network Structures

For client k, let $\mathbf{h}_l^{(k)}$ denote the output of the l-th hidden layer in the infinite depth neural network:

$$\mathbf{h}_l^{(k)} = \sigma(\mathcal{W}_l^{(k)}\mathbf{h}_{l-1}^{(k)}) \odot \mathbf{z}_l^{(k)} + \mathbf{h}_{l-1}^{(k)},\ l \in \{1, 2, ..., \infty\} \tag{1}$$

where σ is a non-linear activation function, $\mathcal{W}_l^{(k)} \in R^{M\times M}$ is l-th layer weight matrix of the k-th client, and $\odot$ denotes the element-wise product between two vectors. We prune the output of the l-th layer by multiplying them element-wise with a binary vector $\mathbf{z}_l^{(k)}$, whose entries $z_{ml}^{(k)} \in \{0, 1\}$. Each random variable $z_{ml}^{(k)}$ take the value 1 with probability $\pi_l^{(k)}$, which represents the activation probability of the l-th layer, as in Figure 1.

Given a k-th client dataset $\mathcal{D}^{(k)} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{N_k}$, likelihood of the neural network is expressed as:

$$p(\mathcal{D}^{(k)}|\mathbf{Z}^{(k)}, \mathcal{W}^{(k)}) = \prod_{n=1}^{N_k} \mathcal{N}(\mathbf{y}_n|f(\mathbf{x}_n; \mathbf{Z}^{(k)}, \mathcal{W}^{(k)}), \sigma^2\boldsymbol{I}) \tag{2}$$

where $\mathbf{Z}^{(k)} = \{\ \mathbf{z}_l^{(k)}\}$ is a binary matrix, $\mathcal{W}^{(k)} = \{\mathcal{W}_l^{(k)}\}$ is set of weight matrices for k-th client and σ^2 is likelihood noise variance. For classification, we replace the normal distribution with a multinomial distribution and use softmax in the final layer.

3.2 Hierarchical Beta Processes as the Prior

We specify a global beta process $B \sim \text{BP}(c, B_0)$, where c is a concentration parameter and B_0 is a base measure. The global beta process B as a hyper-prior generates the local beta process priors $B_k \sim \text{BP}(c_k, B)$ for the local models, as in Figure 1.

At the server: the global beta process $B = \sum_l \pi_l^{(0)}\delta_{\mathbf{h}_l^{(0)}}$, where $\delta_{\mathbf{h}_l^{(0)}}$ denotes a unit point mass at $\mathbf{h}_l^{(0)}$, provides an infinite collection of pairs $(\mathbf{h}_l^{(0)}, \pi_l^{(0)})$, in which $\mathbf{h}_l^{(0)} \in \mathbf{H^{(0)}}$ corresponds to a global hidden-layer function and $\pi_l^{(0)}$ is its activation probability.

For each client k**:** local beta process $B_k = \sum_l \pi_l^{(k)}\delta_{\mathbf{h}_l^{(k)}}$ provides an infinite collection of pairs $(\mathbf{h}_l^{(k)}, \pi_l^{(k)})$ in which $\mathbf{h}_l^{(k)} \in \mathbf{H^{(k)}}$ corresponds to a local hidden-layer function and $\pi_l^{(k)}$ is its activation probability. The location $\delta_{\mathbf{h}_l^{(k)}}$ for all B_k is in same location as in $\delta_{\mathbf{h}_l^{(0)}}$ of the B and $E_k[\pi_l^{(k)}] = \pi_l^{(0)}$ (Thibaux and Jordan, 2007). We define the conjugate Bernoulli Process $\mathbf{Z}_{m\cdot}^{(k)} \sim \text{BeP}(B_k)$ as $\mathbf{Z}_{m\cdot}^{(k)} = \sum_l z_{ml}^{(k)}\delta_{\mathbf{h}_l^{(k)}}$ at the same location $\delta_{\mathbf{h}_l^{(k)}}$ as B_k where $z_{ml}^{(k)}$ are independent Bernoulli variables with a probability that $z_{ml}^{(k)} = 1$ equals to $\pi_l^{(k)}$.

Computationally, we employ stick-breaking construction of hierarchical beta-Bernoulli processes (Paisley et al., 2010; Teh et al., 2007) to model the global and local beta processes. At the server, the global beta process is defined as

$$\pi_l^{(0)} = \prod_{j=1}^{l} v_j^{(0)}, \;\; v_l^{(0)} \sim \text{Beta}(\alpha, \beta) \tag{3}$$

For client k, the local beta process and its conjugate Bernoulli process is

$$\pi_l^{(k)} \sim \text{Beta}(\gamma^{(k)}\pi_l^{(0)}, \gamma^{(k)}(1-\pi_l^{(0)})), \;\; z_{ml}^{(k)} \sim \text{Ber}(\pi_l^{(k)}) \tag{4}$$

where $\gamma^{(k)}$ denotes a local scaling parameter. We thus define the local prior over $\mathbf{Z}^{(k)}$ as

$$\begin{aligned} p(\mathbf{Z}^{(k)}, \boldsymbol{\pi}^{(k)} | \gamma^{(k)}, \boldsymbol{\pi}^{(0)}) &= p(\boldsymbol{\pi}^{(k)} | \gamma^{(k)} \boldsymbol{\pi}^{(0)}) p(\mathbf{Z}^{(k)} | \boldsymbol{\pi}^{(k)}) \\ &= \prod_{l=1}^{\infty} \text{Beta}(\pi_l^{(k)} | \gamma^{(k)}\pi_l^{(0)}, \gamma^{(k)}(1-\pi_l^{(0)})) \prod_{m=1}^{M} \text{Ber}(z_{ml}^{(k)} | \pi_l^{(k)}) \end{aligned} \tag{5}$$

3.3 Marginal Likelihood for Network Structure Inference

By combining the hierarchical beta process prior over hidden layers in Eqn. (3) and Eqn. (5) with the likelihood in Eqn. (2), we can infer the number of hidden layers for each local model by integrating over both the local and global structural variables. The overall marginal likelihood integrating over the global variables is

$$p\left(\mathcal{D} | \mathcal{W}, l, \alpha, \beta\right) = \int p(\mathcal{D} | \mathcal{W}, \boldsymbol{\pi}^{(0)}) p(\boldsymbol{\pi}^{(0)}) d\boldsymbol{\pi}^{(0)} \tag{6}$$

where the hyper-prior $p(\boldsymbol{\pi}^{(0)})$ is in Eqn. (3). Given all clients' datasets $\mathcal{D} = \{\mathcal{D}^{(k)}\}_{k=1}^{K}$, the global likelihood in Eqn. (6) can be re-written as:

$$p(\mathcal{D} | \mathcal{W}, \boldsymbol{\pi}^{(0)}) = \prod_{k=1}^{K} p(\mathcal{D}^{(k)} | \mathcal{W}^{(k)}, \boldsymbol{\pi}^{(0)}) \tag{7}$$

where $\mathcal{W} = \{\mathcal{W}^{(k)}\}_{k=1}^{K}$ denotes set of all clients' weights and $\boldsymbol{\pi}^{(0)} = \{\pi_l^{(0)}\}_{l=1}^{\infty}$.

In Eqn.(7), the k-th client's local model $p(\mathcal{D}^{(k)} | \mathcal{W}^{(k)}, \boldsymbol{\pi}^{(0)})$ integrates over local structural variables:

$$p(\mathcal{D}^{(k)} | \mathcal{W}^{(k)}, l^{(k)}, \gamma^{(k)}, \boldsymbol{\pi}^{(0)}) = \int p(\mathcal{D}^{(k)} | \mathcal{W}^{(k)}, \mathbf{Z}^{(k)}) p(\mathbf{Z}^{(k)}, \boldsymbol{\pi}^{(k)}) d\mathbf{Z}^{(k)} d\boldsymbol{\pi}^{(k)} \tag{8}$$

where local likelihood $p(\mathcal{D}^{(k)} | \mathcal{W}^{(k)}, \mathbf{Z}^{(k)})$ and local prior $p(\mathbf{Z}^{(k)}, \boldsymbol{\pi}^{(k)})$ are in Eqn.(2) and Eqn.(5).

3.4 Structured Variational Inference for the Marginal Likelihood

The exact inference of the marginal likelihood is intractable due to the nonlinearity of the neural network as well as the depth of the neural network goes to infinity. We employ structured stochastic variational inference (SSVI) as it captures the dependency between the structural variables $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(k)}$ and $\mathbf{Z}^{(k)}$ (Hoffman and Blei, 2015; Singh et al., 2017). We first specify a variational distribution with truncation level $T^{(0)}$ (i.e., maximum number of hidden layers) to approximate the posterior over the global network structure as:

$$q\left(\boldsymbol{v}^{(0)} | \boldsymbol{\alpha}^{(0)}, \boldsymbol{\beta}^{(0)}\right) = \prod_{t=1}^{T^{(0)}} \text{Beta}\left(v_t^{(0)} | \alpha_t^{(0)}, \beta_t^{(0)}\right) \tag{9}$$

where $(\boldsymbol{\alpha}^{(0)}, \boldsymbol{\beta}^{(0)}) = \{(\alpha_t^{(0)}, \beta_t^{(0)})\}_{t=1}^{T^{(0)}}$ are global variational parameters, and $\pi_t^{(0)} = \prod_{j=1}^{t} v_j^{(0)}$.

We further define a variational distribution for the local network structure with truncation level $T^{(k)}$:

$$q(\mathbf{Z}^{(k)}, \boldsymbol{\pi}^{(k)} | \boldsymbol{\alpha}^{(k)}, \boldsymbol{\beta}^{(k)}) = q(\boldsymbol{\pi}^{(k)}) q(\mathbf{Z}^{(k)} | \boldsymbol{\pi}^{(k)}) = \prod_{t=1}^{T^{(k)}} \text{Beta}(\pi_t^{(k)} | \alpha_t^{(k)}, \beta_t^{(k)}) \prod_{m=1}^{M} \text{ConBer}(z_{mt}^{(k)} | \pi_t^{(k)})$$

where $(\boldsymbol{\alpha}^{(k)}, \boldsymbol{\beta}^{(k)}) = \{(\alpha_t^{(k)}, \beta_t^{(k)})\}_{t=1}^{T^{(k)}}$ are local variational parameters. We also relax the discrete constraint for $\mathbf{Z}^{(k)}$ using concrete Bernoulli distribution (Jang et al., 2017; Maddison et al., 2017).

We can thus derive an evidence lower bound (ELBO) (Jaakkola and Jordan, 2000; Graves, 2011) for the overall log-marginal likelihood in Eqn. (6) as:

$$\mathcal{L} = \sum_{k=1}^{K} \mathbb{E}_{q(\boldsymbol{v}^{(0)})} \big[\mathbb{E}_{q(\mathbf{Z}^{(k)}, \boldsymbol{\pi}^{(k)})} (\log p(\mathcal{D}^{(k)} | \mathcal{W}^{(k)}, \mathbf{Z}^{(k)})) - \mathbb{KL}(q(\mathbf{Z}^{(k)} | \boldsymbol{\pi}^{(k)}) || p(\mathbf{Z}^{(k)} | \boldsymbol{\pi}^{(k)})) - \mathbb{KL}(q(\boldsymbol{\pi}^{(k)}) || p(\boldsymbol{\pi}^{(k)})) \big] - \mathbb{KL}(q(\boldsymbol{v}^{(0)}) || p(\boldsymbol{v}^{(0)})) \tag{10}$$

The first term fits the expected log-likelihood with given local network structure and parameters to the client-specific local data. The first two Kullback–Leibler (KL) divergence terms regularize the local structural variables. The last KL term acts as a regularization for the global network structure. Overall, the ELBO facilitates a trade-off between local model adaptation to client-specific non-IID data and global model generalization. We employ Monte Carlo estimation to approximate the expected log-likelihood and compute the KL divergence analytically.

We further split the overall objective into a client part to infer the most plausible local network structure and optimize local network parameters, and a server part to aggregate network parameters and network structure across clients.

Client Part of the ELBO: For each client k, we optimize the objective function:

$$\mathcal{L}^{(k)} = \mathbb{E}_{q(\boldsymbol{v}^{(0)})} \big[\mathbb{E}_{q(\mathbf{Z}^{(k)}, \boldsymbol{\pi}^{(k)})} (\log p(\mathcal{D}^{(k)} | \mathcal{W}^{(k)}, \mathbf{Z}^{(k)})) - \mathbb{KL}(q(\mathbf{Z}^{(k)} | \boldsymbol{\pi}^{(k)}) || p(\mathbf{Z}^{(k)} | \boldsymbol{\pi}^{(k)})) - \mathbb{KL}(q(\boldsymbol{\pi}^{(k)}) || p(\boldsymbol{\pi}^{(k)})) \big] \tag{11}$$

with respect to variational parameters $(\boldsymbol{\alpha}^{(k)}, \boldsymbol{\beta}^{(k)})$ to infer the local network structural variables and learn local network parameters $\mathcal{W}^{(k)}$ simultaneously. The global distribution thus only regularizes client k's truncation.

Server Part of the ELBO: At the central server, we optimize the objective function:

$$\mathcal{L} = \sum_{k=1}^{K} \mathbb{E}_{q(\boldsymbol{v}^{(0)})} \big[- \mathbb{KL}(q(\boldsymbol{\pi}^{(k)}) || p(\boldsymbol{\pi}^{(k)})) \big] - \mathbb{KL}(q(\boldsymbol{v}^{(0)}) || p(\boldsymbol{v}^{(0)})) \tag{12}$$

with respect to global variational parameters $(\boldsymbol{\alpha}^{(0)}, \boldsymbol{\beta}^{(0)})$ to infer an aggregated network structure across clients. As clients can have different truncation levels, each client contributes only through the structure components present under its assigned truncation.

Parameter Aggregation: We obtain the global model parameters by aggregating local model parameters through FedAvg (McMahan et al., 2017) in a block-wise manner. Let $\tilde{T} = \{T_1, T_2, \cdots, T_J\}$ denote the ordered set of distinct truncation levels across clients, where $T_1 < T_2 < \cdots < T_J$. Then, each global model network parameter block is computed as:

$$\mathcal{W}^{(0)}[T_{j-1} : T_j] = \sum_{k \in \mathcal{C}_j} \frac{N_k}{N_j} \mathcal{W}^{(k)}[T_{j-1} : T_j], \qquad T_0 = 0,\ j = 1, \ldots, J. \tag{13}$$

where $\mathcal{C}_j = \{k : T^{(k)} \geq T_j\}$ denotes the set of clients whose local truncation level contains the parameter block $[T_{j-1} : T_j]$. N_k denotes the number of data points at each client k, and $N_j = \sum_{k \in \mathcal{C}_j} N_k$. This aggregation ensures that each global parameter block is only averaged over clients whose truncated local models contain that parameter block.

4 Convergence Analysis

We first analyze the statistical convergence behavior of the local beta processes over layer activation for each client k: $p(\pi_l^{(k)} | \mathbf{z}_l)$ towards the global layer activation mean $\bar{\pi}_l^{(0)} = E[\pi_l^{(0)} | \mathbf{z}_l]$.

Table 1: Average test accuracy(%) on image datasets. Besides "Global Model" performance, "Local Model (T_1 toT_4)" reports performance for each truncation level, and "Local Model Average" reports the average across truncation levels. Best results are in bold.

Dataset	α	Method	Local Model					Global Model
			T_1	T_2	T_3	T_4	Average	
CIFAR10	0.3	HeteroFL	61.6 ± 2.8	69.5 ± 0.7	73.3 ± 1.5	67.8 ± 3.9	68.1 ± 1.1	64.7 ± 1.6
		FedRolex	49.9 ± 3.4	62.2 ± 1.4	69.7 ± 1.9	71.8 ± 1.6	63.4 ± 0.8	69.4 ± 0.6
		DepthFL	61.4 ± 1.7	70.1 ± 2.4	75.4 ± 2.9	73.9 ± 2.8	70.2 ± 2.3	71.1 ± 1.7
		ScaleFL	63.3 ± 1.3	70.2 ± 1.1	75.4 ± 2.4	71.2 ± 4.1	70.0 ± 1.5	68.1 ± 2.6
		FIARSE	66.2 ± 1.8	71.8 ± 0.8	73.8 ± 1.1	71.9 ± 1.1	70.9 ± 0.4	68.9 ± 1.0
		Ours	$\mathbf{74.1 \pm 1.6}$	$\mathbf{77.7 \pm 1.1}$	$\mathbf{79.5 \pm 0.6}$	$\mathbf{78.8 \pm 2.2}$	$\mathbf{77.5 \pm 0.5}$	$\mathbf{76.3 \pm 0.6}$
	10.0	HeteroFL	64.5 ± 1.3	68.1 ± 1.0	69.9 ± 0.9	70.1 ± 0.6	68.2 ± 0.5	69.1 ± 0.7
		FedRolex	60.6 ± 2.3	67.6 ± 1.1	71.1 ± 1.2	73.2 ± 0.6	68.1 ± 1.1	72.2 ± 0.8
		DepthFL	69.8 ± 0.6	73.7 ± 1.6	74.2 ± 1.2	74.3 ± 0.6	73.0 ± 1.0	73.3 ± 0.9
		ScaleFL	66.3 ± 1.0	70.2 ± 1.3	71.8 ± 0.8	71.5 ± 0.9	69.9 ± 0.7	70.4 ± 0.8
		FIARSE	69.7 ± 0.5	72.4 ± 1.1	71.9 ± 0.1	72.7 ± 0.7	71.7 ± 0.5	71.5 ± 0.6
		Ours	$\mathbf{76.7 \pm 0.2}$	$\mathbf{78.1 \pm 1.4}$	$\mathbf{78.5 \pm 0.7}$	$\mathbf{79.3 \pm 0.6}$	$\mathbf{78.1 \pm 0.7}$	$\mathbf{78.4 \pm 0.7}$
CIFAR100	0.3	HeteroFL	37.0 ± 0.5	40.4 ± 1.1	41.0 ± 0.5	38.8 ± 0.8	39.3 ± 0.1	37.4 ± 0.8
		FedRolex	34.3 ± 0.7	41.0 ± 0.2	43.5 ± 1.1	43.4 ± 0.8	40.6 ± 0.4	42.2 ± 0.7
		DepthFL	43.4 ± 1.3	45.9 ± 0.9	46.1 ± 0.2	44.0 ± 0.6	44.8 ± 0.4	42.1 ± 0.3
		ScaleFL	38.6 ± 1.4	42.4 ± 1.8	42.3 ± 0.5	41.1 ± 0.6	41.1 ± 0.7	39.1 ± 0.5
		FIARSE	41.6 ± 1.2	43.2 ± 0.6	43.6 ± 0.6	44.1 ± 0.5	43.1 ± 0.3	42.1 ± 0.8
		Ours	$\mathbf{48.5 \pm 1.5}$	$\mathbf{51.1 \pm 0.8}$	$\mathbf{52.7 \pm 0.5}$	$\mathbf{53.2 \pm 0.5}$	$\mathbf{51.4 \pm 0.5}$	$\mathbf{50.3 \pm 0.5}$
	10.0	HeteroFL	46.9 ± 0.3	50.9 ± 0.3	50.7 ± 0.1	49.1 ± 1.2	49.4 ± 0.3	49.5 ± 1.0
		FedRolex	45.9 ± 1.5	52.9 ± 0.2	54.0 ± 0.9	53.4 ± 0.2	51.6 ± 0.5	53.9 ± 0.4
		DepthFL	55.8 ± 0.1	57.3 ± 0.7	56.9 ± 0.6	53.1 ± 0.7	55.8 ± 0.2	53.8 ± 0.7
		ScaleFL	48.9 ± 0.6	52.4 ± 0.6	51.8 ± 0.8	49.6 ± 0.1	50.7 ± 0.4	49.6 ± 0.1
		FIARSE	54.3 ± 0.7	54.3 ± 0.5	54.7 ± 1.5	53.2 ± 0.7	54.1 ± 0.8	53.5 ± 0.9
		Ours	$\mathbf{61.3 \pm 0.5}$	$\mathbf{62.6 \pm 1.2}$	$\mathbf{61.5 \pm 1.1}$	$\mathbf{60.9 \pm 0.1}$	$\mathbf{61.6 \pm 0.2}$	$\mathbf{61.7 \pm 0.4}$
Tiny ImageNet	0.3	HeteroFL	23.7 ± 0.2	26.0 ± 0.4	27.4 ± 0.9	25.1 ± 0.6	25.6 ± 0.2	23.8 ± 0.5
		FedRolex	20.6 ± 0.8	27.9 ± 0.8	28.9 ± 0.7	28.3 ± 0.5	26.4 ± 0.4	27.9 ± 0.8
		DepthFL	29.6 ± 0.4	31.3 ± 0.9	30.7 ± 0.9	27.0 ± 0.9	29.7 ± 0.5	26.2 ± 0.5
		ScaleFL	24.9 ± 0.5	27.1 ± 0.8	27.6 ± 0.6	26.2 ± 1.2	26.4 ± 0.5	24.1 ± 1.1
		FIARSE	28.1 ± 0.6	28.7 ± 1.9	28.9 ± 0.4	27.9 ± 1.9	28.4 ± 1.1	27.2 ± 1.2
		Ours	$\mathbf{32.9 \pm 0.4}$	$\mathbf{34.8 \pm 0.7}$	$\mathbf{36.1 \pm 0.2}$	$\mathbf{36.0 \pm 0.9}$	$\mathbf{34.9 \pm 0.4}$	$\mathbf{33.8 \pm 0.3}$
	10.0	HeteroFL	26.5 ± 1.3	28.9 ± 0.2	28.9 ± 0.1	27.4 ± 1.3	27.9 ± 0.3	27.5 ± 0.5
		FedRolex	24.2 ± 0.3	31.1 ± 0.6	31.7 ± 0.3	32.0 ± 0.6	29.7 ± 0.2	32.2 ± 0.4
		DepthFL	33.5 ± 0.7	35.2 ± 0.3	32.8 ± 1.3	30.6 ± 0.4	33.0 ± 0.3	30.3 ± 0.2
		ScaleFL	28.6 ± 0.9	30.9 ± 0.6	29.3 ± 0.6	28.2 ± 0.5	29.3 ± 0.3	27.8 ± 0.3
		FIARSE	33.3 ± 0.7	34.5 ± 0.4	33.1 ± 1.0	32.0 ± 0.7	33.2 ± 0.4	31.9 ± 0.2
		Ours	$\mathbf{37.5 \pm 0.3}$	$\mathbf{37.7 \pm 0.4}$	$\mathbf{36.5 \pm 0.5}$	$\mathbf{35.1 \pm 0.3}$	$\mathbf{36.7 \pm 0.2}$	$\mathbf{37.5 \pm 0.1}$

Theorem 4.1 *Let the hyper-prior (α, β) fixed, and we approximate the global posterior with $p(\pi_l^{(0)}|\mathbf{z}_l) \approx \delta(\pi_l^{(0)} - \bar{\pi}_l^{(0)})$ for simplicity, the convergence of the mean of local layer-wise activation probability:*

$$\mathbb{E}[\pi_l^{(k)}] = \lambda_l^{(k)} \bar{\pi}_l^{(0)} + (1 - \lambda_l^{(k)}) \hat{\pi}_l^{(k)} \tag{14}$$

where $\lambda_l^{(k)} = \frac{\gamma^{(k)}}{\gamma^{(k)}+M}$. It indicates that the mean of local layer-wise activation lies in between the global activation mean $\bar{\pi}_l^{(0)}$ and the client-specific empirical estimate $\hat{\pi}_l^{(k)}$. The convergence of local layer-wise activation probability toward the global mean is thus governed by $\lambda_l^{(k)}$. In Section 5.1, our ablation study shows that without the regularization of global beta process local empirical estimates $\hat{\pi}_l^{(k)}$ alone are prone to overfitting.

We further analyze the algorithmic convergence of our method under partial client participation. The detailed assumptions and proof are provided in Appendix A.2.

Table 2: Average test accuracy (%) on text datasets. "Local Model (T_1 to T_4)" and "Local Model Average" report performance for each truncation level and average across them. Best and second-best results are bolded and underlined, respectively.

Dataset	α	Method	Local Model					Global Model
			T_1	T_2	T_3	T_4	Average	
AGNews	0.3	HeteroFL	86.9 ± 0.3	89.0 ± 1.0	87.3 ± 1.0	88.9 ± 1.3	88.0 ± 0.3	87.0 ± 1.2
		DepthFL	88.7 ± 0.6	90.2 ± 0.2	87.9 ± 1.1	89.5 ± 0.4	89.1 ± 0.5	88.3 ± 0.9
		ScaleFL	87.4 ± 0.2	89.8 ± 1.2	87.7 ± 0.8	88.4 ± 0.9	88.3 ± 0.5	87.1 ± 0.9
		Ours	$\mathbf{89.1 \pm 0.8}$	$\mathbf{91.3 \pm 0.5}$	$\mathbf{90.9 \pm 1.2}$	$\mathbf{91.4 \pm 0.6}$	$\mathbf{90.7 \pm 0.4}$	$\mathbf{88.6 \pm 0.2}$
	10.0	HeteroFL	88.4 ± 0.5	88.1 ± 0.5	88.3 ± 0.4	88.1 ± 0.8	88.2 ± 0.2	88.3 ± 0.2
		DepthFL	$\mathbf{89.5 \pm 0.2}$	$\mathbf{89.7 \pm 0.5}$	$\mathbf{89.4 \pm 0.5}$	$\mathbf{88.8 \pm 0.8}$	$\mathbf{89.4 \pm 0.3}$	$\mathbf{89.5 \pm 0.3}$
		ScaleFL	88.3 ± 0.4	88.9 ± 0.3	88.4 ± 0.5	88.1 ± 0.7	88.4 ± 0.1	88.6 ± 0.1
		Ours	89.4 ± 0.3	89.2 ± 0.7	89.2 ± 0.1	88.6 ± 0.6	89.1 ± 0.2	89.1 ± 0.3
IMDB	0.3	HeteroFL	81.6 ± 1.9	82.7 ± 0.5	81.8 ± 1.4	81.6 ± 0.9	81.9 ± 0.9	81.4 ± 0.5
		DepthFL	82.9 ± 0.3	83.7 ± 0.1	82.5 ± 0.3	82.8 ± 0.4	82.9 ± 0.2	$\mathbf{82.6 \pm 0.2}$
		ScaleFL	81.7 ± 1.3	83.9 ± 0.5	82.2 ± 1.3	81.7 ± 0.8	82.4 ± 0.7	81.2 ± 0.6
		Ours	$\mathbf{83.9 \pm 0.7}$	$\mathbf{84.0 \pm 0.6}$	$\mathbf{83.6 \pm 0.5}$	$\mathbf{83.5 \pm 0.4}$	$\mathbf{83.8 \pm 0.3}$	81.7 ± 1.1
	10.0	HeteroFL	$\mathbf{83.9 \pm 0.6}$	$\mathbf{83.5 \pm 0.9}$	82.9 ± 0.2	82.7 ± 0.9	$\mathbf{83.3 \pm 0.2}$	82.8 ± 0.5
		DepthFL	82.4 ± 0.5	82.7 ± 0.7	83.1 ± 0.2	82.9 ± 0.6	82.8 ± 0.2	82.9 ± 0.2
		ScaleFL	83.7 ± 0.1	83.0 ± 0.7	82.7 ± 0.7	82.8 ± 0.5	83.1 ± 0.4	82.9 ± 0.3
		Ours	82.7 ± 0.7	83.4 ± 0.2	$\mathbf{83.5 \pm 0.7}$	$\mathbf{83.2 \pm 0.2}$	83.2 ± 0.4	$\mathbf{83.2 \pm 0.4}$

Theorem 4.2 *Under the assumptions specified in Appendix A.2, if A out of K clients participate in each communication round c, our proposed method satisfies:*

$$\min_{c\in[C]} \mathbb{E}[\|\nabla F(\mathcal{W}_c)\|_2^2] \leq \mathcal{O}\left(\frac{K}{A}\frac{F(\mathcal{W}_0) - F^\star + \sigma^2}{\sqrt{C}}\right) + \mathcal{O}\left(\frac{K}{C}\sum_{c\in[C]} \mathbb{E}[\delta_c^2 \|\mathcal{W}_c\|_2^2]\right) \tag{15}$$

Our proposed algorithm achieves the standard $\mathcal{O}(\frac{1}{\sqrt{C}})$ dependence on the number of communication rounds when the mask-induced gradient discrepancy decays sufficiently fast. This rate is consistent with existing convergence results for the model-heterogeneous FL.

5 Experiments

Datasets and Models: We assess the performance via both image classification tasks using CIFAR10, CIFAR100, and Tiny-ImageNet datasets (Krizhevsky et al., 2009; Yang et al.), and text classification tasks using AGNews and IMDB datasets (Zhang et al., 2015; Maas et al., 2011). We adopt pre-activated ResNet proposed in (He et al., 2016; Nazaret and Blei, 2022) for image classification tasks and Pre-LN Transformer (Xiong et al., 2020) for text classification tasks. We adopt static batch normalization and layer normalization (Diao et al., 2021).

Data Heterogeneity: Following a commonly adopted setup in FL benchmarks (Yurochkin et al., 2019; Hsu et al., 2019), we partition each dataset using a Dirichlet distribution with concentration parameter settings $\alpha = 0.3$ and $\alpha = 10.0$, corresponding to high and low degrees of data heterogeneity, to simulate non-IID data distributions across clients.

Model Heterogeneity: For all the methods we adopt four local model capacity ratios: $\{1/5, 1/2, 3/4, 1\}$ and evenly split clients into the four groups for fair comparison. For our framework, these ratios roughly correspond to truncation levels $\{T_1 = 10, T_2 = 20, T_3 = 30, T_4 = 40\}$ for image classification and $\{T_1 = 2, T_2 = 4, T_3 = 6, T_4 = 8\}$ for text classification. The server maintains a full-capacity global model of truncation level T_4. For uneven split or resource allocation, robustness of our framework is in Appendix E.2.

Algorithm Setting: We adopt a common experimental configuration for all methods in Appendix C.2. The tuning of the additional hyperparameters introduced by our framework and their sensitivity

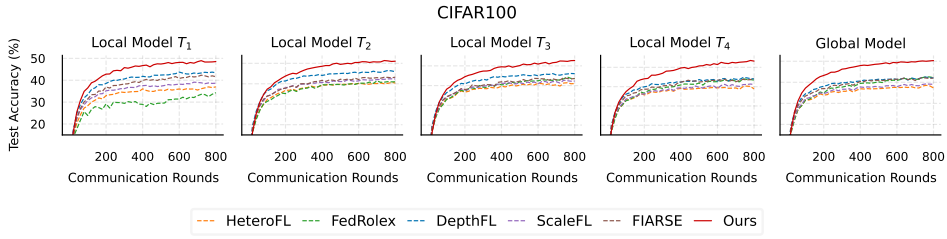

Figure 2: Test accuracy vs. communication rounds under high data heterogeneity level ($\alpha = 0.3$). Our framework shows rapid convergence. Additional results are in Appendix D.

Table 3: Time (hours) required to reach target global model test accuracy on CIFAR-100 under high data heterogeneity level ($\alpha = 0.3$)

Target Accuracy (%)	HeteroFL	FedRolex	DepthFL	ScaleFL	FIARSE	Ours
38.0	16.42	7.66	3.44	10.81	5.71	4.06
40.0	-	11.97	5.86	-	9.51	4.73
42.0	-	16.98	11.71	-	15.22	6.08
44.0	-	-	-	-	-	8.11
50.0	-	-	-	-	-	23.93

analysis are provided in Appendix C.3. Algorithm description is in Appendix B. We run all experiments for 800 communication rounds. To emulate partial client participation, 10% of the clients are randomly selected from a pool of 100 clients in each round.

5.1 Performance Analysis

Table 1 and 2 present comprehensive performance evaluation on the image and text datasets under high and low data heterogeneity scenarios. We report the average test accuracy (%) and standard error over multiple random seeds. Local models are evaluated on their corresponding local test sets, while the global model is evaluated on the superset comprising all clients' test sets.

Global Model: Our global model consistently outperforms baselines across image datasets and all heterogeneity settings, improving by 5.2% and 5.1% on CIFAR10, 8.1% and 7.8% on CIFAR100, and 5.9% and 5.3% on Tiny-ImageNet under high and low heterogeneity, respectively. It also achieves better or comparable performance on the text datasets. The results indicate that by tailoring local models to non-IID data and sharing the complementary network structural information the predictive performance of our global model are significantly better than the baselines.

Local Models: We evaluate each client's local model, which comprises global network parameters but maintain a local network structure, on its local test set. On average, our local models substantially outperform baselines, with gains of 6.6% and 5.1% on CIFAR10, 6.6% and 5.8% on CIFAR100, and 5.2% and 3.5% on Tiny-ImageNet under high and low heterogeneity, respectively. Our local model also demonstrates superior performance compared to competing methods on text datasets. These results highlight the benefits of adapting local network structures to each client's unique data while contributing shared knowledge through the global model.

Convergence and Efficiency Evaluation: Figure 5 shows our framework consistently converges faster than the baselines under high data heterogeneity, with particularly notable gains on CIFAR100, where it achieves competitive performance in fewer than half the communication rounds. The finding is consistent under low data heterogeneity. As in Table 3 our method reaches target accuracies in substantially less wall-clock time than the baselines, particularly at higher global model accuracy targets due to significant less communication rounds combined with comparable per-communication-round training time. The number of additional structural variables introduced by FedSAM is negligible overhead relative to the total communication cost. Further details are in Appendix D.

Local Network Structure Adaptation: Figure 3 shows the global model exhibits a gradual decrease in activation probability across deeper layers, indicating that earlier layers are more consis-

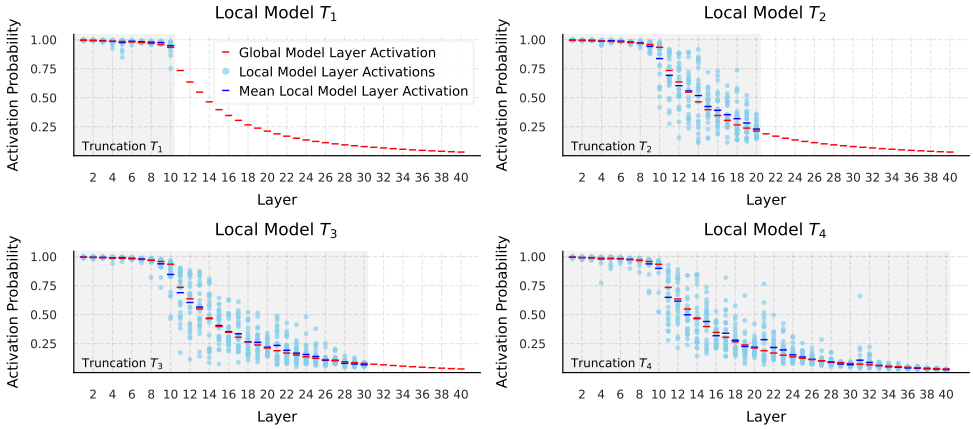

Figure 3: Inferred layer-wise activation probabilities of the global and the local models for different truncation levels on CIFAR10 under high data heterogeneity ($\alpha = 0.3$).

Table 4: "Local Model (T_1 to T_4)" and "Average" report test accuracy (%) for each truncation level and averaged performance with and without a global beta process (gBP). Best results are in bold.

Dataset	α	Method	Local Model				
			T_1	T_2	T_3	T_4	Average
CIFAR10	0.3	w/o gBP	69.5 ± 1.3	74.4 ± 2.9	75.9 ± 1.8	75.3 ± 2.9	73.8 ± 1.9
		w gBP	**74.1 ± 1.6**	**77.7 ± 1.1**	**79.5 ± 0.6**	**78.8 ± 2.2**	**77.5 ± 0.5**
	10.0	w/o gBP	75.1 ± 0.8	78.0 ± 1.2	77.7 ± 1.1	78.6 ± 0.8	77.3 ± 0.8
		w gBP	**76.7 ± 0.2**	**78.1 ± 1.4**	**78.5 ± 0.7**	**79.3 ± 0.6**	**78.1 ± 0.7**
CIFAR100	0.3	w/o gBP	48.3 ± 1.2	50.1 ± 0.7	50.4 ± 0.8	49.8 ± 0.5	49.6 ± 0.4
		w gBP	**48.5 ± 1.5**	**51.1 ± 0.8**	**52.7 ± 0.5**	**53.2 ± 0.5**	**51.4 ± 0.5**
	10.0	w/o gBP	59.7 ± 0.4	61.5 ± 0.7	61.4 ± 0.9	60.2 ± 0.4	60.7 ± 0.3
		w gBP	**61.3 ± 0.5**	**62.6 ± 1.2**	**61.5 ± 1.1**	**60.9 ± 0.1**	**61.6 ± 0.2**
Tiny-ImageNet	0.3	w/o gBP	31.9 ± 0.8	33.9 ± 1.4	34.3 ± 0.2	32.3 ± 1.0	33.1 ± 0.8
		w gBP	**32.9 ± 0.4**	**34.8 ± 0.7**	**36.1 ± 0.2**	**36.0 ± 0.9**	**34.9 ± 0.4**
	10.0	w/o gBP	36.4 ± 0.1	**37.8 ± 0.1**	36.3 ± 0.5	**35.9 ± 0.1**	36.7 ± 0.1
		w gBP	**37.5 ± 0.3**	37.7 ± 0.4	**36.5 ± 0.5**	35.1 ± 0.3	**36.6 ± 0.2**

tently shared across clients, whereas deeper layer are activated more selectively. Local activation patterns closely follow the global trend on average, while still exhibiting substantial client-level variability. This suggests FedSAM shares common representation components through the global model, while allowing the local model structures to be adapted to the complexity of their local data.

Ablation Study: We conduct an ablation study to investigate the effectiveness of the global beta process (gBP) on client-specific local model performance. For image datasets under both high and low data heterogeneity settings, Table 4 shows that incorporating the global beta process consistently improves the local models' performance, with more pronounced gains under high data heterogeneity ($\alpha = 0.3$). These results indicate that in comparison with adapting local network structures with independent beta processes, our framework benefits from a global beta process aggregating and sharing structural information and regularizing local models' learning process.

6 Conclusion

Our hierarchical Bayesian inference framework FedSAM enables local models to adapt network structures to client-specific data while aggregating and sharing the structural information via a hyper-prior at the global level. We develop a SSVI strategy for approximate inference. Empirical results show superior performance across various model-heterogeneous FL settings with high efficiency.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing polishing, literature search, and understanding related literature. We take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide the experimental details and hyperparameter configurations in Section 5 and in Appendix C. Additionally, we provide the full algorithm description in Appendix B.

REFERENCES

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

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

Peter Kairouz and H Brendan McMahan. Advances and open problems in federated learning. *Foundations and trends in machine learning*, 14(1-2):1–210, 2021.

Irene Wang, Prashant Nair, and Divya Mahajan. Fluid: Mitigating stragglers in federated learning using invariant dropout. *Advances in Neural Information Processing Systems*, 36:73258–73273, 2023.

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, 2024.

Letian Zhang, Bo Chen, Jieming Bian, Lei Wang, and Jie Xu. FedEL: Federated elastic learning for heterogeneous devices. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=DfeHiyTz2W.

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, pages 21858–21866, 2026.

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.

Chengjie Ma, Seungeun Oh, Jihong Park, and Seong-Lyun Kim. Breaking the capacity bottleneck in model-heterogeneous federated learning via gradual model restoration. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=iziZOSvtGK.

Nils Lid Hjort. Nonparametric bayes estimators based on beta processes in models for life history data. *the Annals of Statistics*, pages 1259–1294, 1990.

Tamara Broderick, Michael I Jordan, and Jim Pitman. Beta processes, stick-breaking and power laws. 2012.

Romain Thibaux and Michael I Jordan. Hierarchical beta processes and the indian buffet process. In *Artificial intelligence and statistics*, pages 564–571. PMLR, 2007.

Sebastian Caldas, Jakub Konečny, H Brendan McMahan, and Ameet Talwalkar. Expanding the reach of federated learning by reducing client resource requirements. *arXiv preprint arXiv:1812.07210*, 2018.

Samuel Horvath, Stefanos Laskaridis, Mario Almeida, Ilias Leontiadis, Stylianos Venieris, and Nicholas Lane. Fjord: Fair and accurate federated learning under heterogeneous targets with ordered dropout. *Advances in Neural Information Processing Systems*, 34:12876–12889, 2021.

Junyuan Hong, Haotao Wang, Zhangyang Wang, and Jiayu Zhou. Efficient split-mix federated learning for on-demand and in-situ customization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=_QLmakITKg.

Samiul Alam, Luyang Liu, Ming Yan, and Mi Zhang. Fedrolex: Model-heterogeneous federated learning with rolling sub-model extraction. *Advances in neural information processing systems*, 35:29677–29690, 2022.

Ruixuan Liu, Fangzhao Wu, Chuhan Wu, Yanlin Wang, Lingjuan Lyu, Hong Chen, and Xing Xie. No one left behind: Inclusive federated learning over heterogeneous devices. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3398–3406, 2022.

Dongping Liao, Xitong Gao, Yiren Zhao, and Cheng-Zhong Xu. Adaptive channel sparsity for federated learning under system heterogeneity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20432–20441, 2023.

Minjae Kim, Sangyoon Yu, Suhyun Kim, and Soo-Mook Moon. Depthfl: Depthwise federated learning for heterogeneous clients. In *The Eleventh International Conference on Learning Representations*, 2022.

Fatih Ilhan, Gong Su, and Ling Liu. Scalefl: Resource-adaptive federated learning with heterogeneous clients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24532–24541, 2023.

Adam Thor Thorgeirsson and Frank Gauterin. Probabilistic predictions with federated learning. *Entropy*, 23(1):41, 2020.

Hong-You Chen and Wei-Lun Chao. Fed{be}: Making bayesian model ensemble applicable to federated learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=dgtpE6gKjHn.

Maruan Al-Shedivat, Jennifer Gillenwater, Eric Xing, and Afshin Rostamizadeh. Federated learning via posterior averaging: A new perspective and practical algorithms. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=GFsU8a0sGB.

Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *International conference on machine learning*, pages 7252–7261. PMLR, 2019.

Jacopo Talpini, Marco Savi, and Giovanni Neglia. FedBEns: One-shot federated learning based on bayesian ensemble. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=oTCiv1bkjG.

Feng Yin, Zhidi Lin, Qinglei Kong, Yue Xu, Deshi Li, Sergios Theodoridis, and Shuguang Robert Cui. Fedloc: Federated learning framework for data-driven cooperative localization and location data processing. *IEEE Open Journal of Signal Processing*, 1:187–215, 2020.

Idan Achituve, Aviv Shamsian, Aviv Navon, Gal Chechik, and Ethan Fetaya. Personalized federated learning with gaussian processes. *Advances in Neural Information Processing Systems*, 34:8392–8406, 2021.

Minxue Tang, Xuefei Ning, Yitu Wang, Jingwei Sun, Yu Wang, Hai Li, and Yiran Chen. Fedcor: Correlation-based active client selection strategy for heterogeneous federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10102–10111, 2022.

Nikita Kotelevskii, Maxime Vono, Alain Durmus, and Eric Moulines. Fedpop: A bayesian approach for personalised federated learning. *Advances in Neural Information Processing Systems*, 35: 8687–8701, 2022.

Disha Makhija, Joydeep Ghosh, and Nhat Ho. A bayesian approach for personalized federated learning in heterogeneous settings, 2024. URL https://openreview.net/forum?id=OhTzuWzO6Q.

Xu Zhang, Yinchuan Li, Wenpeng Li, Kaiyang Guo, and Yunfeng Shao. Personalized federated learning via variational bayesian inference. In *International Conference on Machine Learning*, pages 26293–26310. PMLR, 2022.

Junyi Zhu, Xingchen Ma, and Matthew B Blaschko. Confidence-aware personalized federated learning via variational expectation maximization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24542–24551, 2023.

Shrey Bhatt, Aishwarya Gupta, and Piyush Rai. Federated learning with uncertainty via distilled predictive distributions. In *Asian Conference on Machine Learning*, pages 153–168. PMLR, 2024.

Ting Wei, Biao Mei, Junliang Lyu, Renquan Zhang, Feng Zhou, and Yifan Sun. Personalized bayesian federated learning with wasserstein barycenter aggregation. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 75398–75427. Curran Associates, Inc., 2025. doi: 10.52202/085713-2532. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/6d0c18c6edc02be434c883bdd9fbe3b5-Paper-Conference.pdf.

Terje Mildner, Oliver Hamelijnck, Paris Giampouras, and Theodoros Damoulas. Federated generalised variational inference: A robust probabilistic federated learning framework. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 44134–44174. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/mildner25a.html.

Mahendra Singh Thapa and Rui Li. Harnessing heterogeneous statistical strength for personalized federated learning via hierarchical bayesian inference. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Zn6hmmBnAa.

Kishan KC, Rui Li, and MohammadMahdi Gilany. Joint inference for neural network depth and dropout regularization. *Advances in neural information processing systems*, 34:26622–26634, 2021.

Samuel Kessler, Vu Nguyen, Stefan Zohren, and Stephen J Roberts. Hierarchical indian buffet neural networks for bayesian continual learning. In *Uncertainty in artificial intelligence*, pages 749–759. PMLR, 2021.

Jeevan Thapa and Rui Li. Bayesian adaptation of network depth and width for continual learning. In *International Conference on Machine Learning*, volume 235, pages 48038–48061. PMLR, 2024.

John W Paisley, Aimee K Zaas, Christopher W Woods, Geoffrey S Ginsburg, and Lawrence Carin. A stick-breaking construction of the beta process. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 847–854, 2010.

Yee Whye Teh, Dilan Grür, and Zoubin Ghahramani. Stick-breaking construction for the indian buffet process. In *Artificial intelligence and statistics*, pages 556–563. PMLR, 2007.

Matthew Hoffman and David Blei. Stochastic structured variational inference. In *Artificial Intelligence and Statistics*, pages 361–369. PMLR, 2015.

Rachit Singh, Jeffrey Ling, and Finale Doshi-Velez. Structured variational autoencoders for the beta-bernoulli process. In *NIPS 2017 Workshop on Advances in Approximate Bayesian Inference*, 2017.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rkE3y85ee.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=S1jE5L5gl.

Tommi S Jaakkola and Michael I Jordan. Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10(1):25–37, 2000.

Alex Graves. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011.

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

Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

Xuan Yang et al. Tiny imagenet visual recognition challenge.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150, 2011.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

Achille Nazaret and David Blei. Variational inference for infinitely deep neural networks. In *International Conference on Machine Learning*, pages 16447–16461. PMLR, 2022.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International conference on machine learning*, pages 10524–10533. PMLR, 2020.

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.

A Theoretical Analysis

A.1 Statistical Convergence behavior of the local beta processes

After estimating the approximate global posterior via SSVI, as described in Section 3.4, we analyze the convergence of the local beta processes over layer activation for each client k.

Specifically, we analyze the convergence behavior of the local layer-wise activation probability $\pi_l^{(k)}$ towards the global layer activation mean

$$\bar{\pi}_l^{(0)} = E[\pi_l^{(0)}|\mathbf{z}_l] \tag{16}$$

Let the hyper-prior (α, β) fixed. We adopt the point-mass approximation to the global posterior for simplicity:

$$p(\pi_l^{(0)}|\mathbf{z}_l) \approx \delta(\pi_l^{(0)} - \bar{\pi}_l^{(0)}) \tag{17}$$

Then the marginal distribution of the local layer-wise activation probability satisfies

$$p(\pi_l^{(k)}|\mathbf{z}_l) = \int p(\pi_l^{(k)}|\mathbf{z}_l, \pi_l^{(0)}) p(\pi_l^{(0)}|\mathbf{z}_l) d\pi_l^{(0)} \approx p(\pi_l^{(k)}|\mathbf{z}_l, \bar{\pi}_l^{(0)}) \tag{18}$$

Given the local beta-Bernoulli process,

$$\pi_l^{(k)}|\bar{\pi}_l^{(0)} \sim \text{Beta}(\gamma^{(k)}\bar{\pi}_l^{(0)}, \gamma^{(k)}(1-\bar{\pi}_l^{(0)})), \quad z_{ml}^{(k)}|\pi_l^{(k)} \sim \text{Ber}(\pi_l^{(k)}) \tag{19}$$

Let

$$P_l^{(k)} = \sum_{m=1}^{M} z_{ml}^{(k)} \text{ and } Q_l^{(k)} = M - P_l^{(k)} \tag{20}$$

denote the numbers of active and inactive neurons, respectively, in layer l for client k. By conjugacy, the posterior distribution of the local layer-wise activation probability is

$$\pi_l^{(k)}|\mathbf{z}_l, \bar{\pi}_l^{(0)} \sim \text{Beta}(\gamma^{(k)}\bar{\pi}_l^{(0)} + P_l^{(k)}, \gamma^{(k)}(1-\bar{\pi}_l^{(0)}) + Q_l^{(k)}) \tag{21}$$

The corresponding posterior mean is

$$E[\pi_l^{(k)}] = \frac{\gamma^{(k)}\bar{\pi}_l^{(0)} + P_l^{(k)}}{\gamma^{(k)} + P_l^{(k)} + Q_l^{(k)}} \tag{22}$$

Let,

$$\lambda_l^{(k)} = \frac{\gamma^{(k)}}{\gamma^{(k)} + P_l^{(k)} + Q_l^{(k)}} \text{ and } \hat{\pi}_l^{(k)} = \frac{P_l^{(k)}}{P_l^{(k)} + Q_l^{(k)}}, \tag{23}$$

Then, the mean of the local layer-wise activation probability becomes

$$E[\pi_l^{(k)}] = \lambda_l^{(k)}\bar{\pi}_l^{(0)} + (1-\lambda_l^{(k)})\hat{\pi}_l^{(k)} \tag{24}$$

This expression shows that the local layer-wise activation probability converges to a convex combination of the global activation mean and the client-specific empirical estimate. The coefficient $\lambda_l^{(k)}$ governs the degree of shrinkage towards the global mean. As $\lambda_l^{(k)} \to 0$, the mean of local layer-wise activation probability becomes dominated by the local empirical estimate $\hat{\pi}_l^{(k)}$. Conversely, when $\lambda_l^{(k)}$ remains large, the local layer-wise activation estimate stays close to the global activation mean.

A.2 Convergence Analysis of the Algorithm

We analyze the algorithmic convergence of our proposed framework under partial client participation, where the local beta-Bernoulli processes induce stochastic masks for each client that determine the active parameter blocks at each communication round.

Algorithm 1 Our Method

Input:
K: Total number of clients indexed by k
C: Total number of communication rounds
P: Client sampling rate
E: Total number of local epochs

Initialize:
Global model with truncation level $T^{(0)}$, characterized by global network parameters $\mathcal{W}^{(0)}$ and global network structural variables $(\boldsymbol{\alpha}^{(0)}, \boldsymbol{\beta}^{(0)})$

Local model for each client $k \in K$ with truncation level $T^{(k)}$, characterized by local network parameters $\mathcal{W}^{(k)}$ and local network structural variables $(\boldsymbol{\alpha}^{(k)}, \boldsymbol{\beta}^{(k)})$

Server executes:
for $c = 0$ **to** $C - 1$ **do**

$\mathcal{P}^c \leftarrow$ Select a random subset of clients

for $k \in \mathcal{P}^c$ **in parallel do**

$\mathcal{W}_{c+1}^{(k)}, \boldsymbol{\alpha}_{c+1}^{(k)}, \boldsymbol{\beta}_{c+1}^{(k)} \leftarrow$ **ClientUpdate**$(k, \mathcal{W}_c^{(0)}, \boldsymbol{\alpha}_c^{(0)}, \boldsymbol{\beta}_c^{(0)})$

$\mathcal{W}_{c+1}^{(0)}, \boldsymbol{\alpha}_{c+1}^{(0)}, \boldsymbol{\beta}_{c+1}^{(0)} \leftarrow$ **ServerUpdate**$(\{\mathcal{W}_{c+1}^{(k)}, \boldsymbol{\alpha}_{c+1}^{(k)}, \boldsymbol{\beta}_{c+1}^{(k)}\}_{k \in \mathcal{P}^c})$

ServerUpdate$(\{\mathcal{W}_{c+1}^{(k)}, \boldsymbol{\alpha}_{c+1}^{(k)}, \boldsymbol{\beta}_{c+1}^{(k)}\}_{k \in \mathcal{P}^c})$**:**

$\mathcal{W}_{c+1}^{(0)} \leftarrow$ Truncation-aware block-wise aggregation of network parameters using Eqn. (13)
$\boldsymbol{\alpha}_{c+1}^{(0)}, \boldsymbol{\beta}_{c+1}^{(0)} \leftarrow$ Update global model network structure via Eqn. (12), where each client contributes only up to its truncation level

return $\mathcal{W}_{c+1}^{(0)}, \boldsymbol{\alpha}_{c+1}^{(0)}, \boldsymbol{\beta}_{c+1}^{(0)}$

ClientUpdate$(k, \mathcal{W}_c^{(0)}, \boldsymbol{\alpha}_c^{(0)}, \boldsymbol{\beta}_c^{(0)})$**:**

▷ Initialize local model network parameters
$\mathcal{W}_c^{(k)} \leftarrow \mathcal{W}_c^{(0)}[: T^{(k)}]$

▷ Extract global structural variables up to the client's truncation level
$\tilde{\boldsymbol{\alpha}}_c^{(0)}, \tilde{\boldsymbol{\beta}}_c^{(0)} \leftarrow \boldsymbol{\alpha}_c^{(0)}[: T^{(k)}], \boldsymbol{\beta}_c^{(0)}[: T^{(k)}]$

▷ Optimize local model using Eqn. (11)
for $i = 1$ **to** E **do**

$$\mathcal{W}_c^{(k)}, \boldsymbol{\alpha}_c^{(k)}, \boldsymbol{\beta}_c^{(k)} = \underset{\mathcal{W}^{(k)}, \boldsymbol{\alpha}^{(k)}, \boldsymbol{\beta}^{(k)}}{\operatorname{argmin}} \mathcal{L}^{(k)}(\mathcal{W}_c^{(k)}, \boldsymbol{\alpha}_c^{(k)}, \boldsymbol{\beta}_c^{(k)}, \tilde{\boldsymbol{\alpha}}_c^{(0)}, \tilde{\boldsymbol{\beta}}_c^{(0)})$$

$\mathcal{W}_{c+1}^{(k)}, \boldsymbol{\alpha}_{c+1}^{(k)}, \boldsymbol{\beta}_{c+1}^{(k)} \leftarrow \mathcal{W}_c^{(k)}, \boldsymbol{\alpha}_c^{(k)}, \boldsymbol{\beta}_c^{(k)}$

return $\mathcal{W}_{c+1}^{(k)}, \boldsymbol{\alpha}_{c+1}^{(k)}, \boldsymbol{\beta}_{c+1}^{(k)}$

At the server, we consider a global model with the maximum truncation level $T^{(0)}$:

$$\mathcal{W} = \big(\mathcal{W}[1], \mathcal{W}[2], \ldots, \mathcal{W}[T^{(0)}]\big) \tag{25}$$

where $\mathcal{W}[j]$ denotes the parameters associated with block j.

Each client $k \in [K]$ is assigned a truncation level

$$T^{(k)} \leq T^{(0)} \tag{26}$$

At the beginning of each communication round c, client k samples a binary mask from local beta-Bernoulli Process $M_c^{(k)} = (M_c^{(k)}[1], \ldots, M_c^{(k)}[T^{(0)}])$, where $M_c^{(k)}[j] = 0, j > T^{(k)}$.

For the convergence analysis, we assume that the sampled mask remains fixed throughout the E local updates within round c. Therefore, the effective model of client k is:

$$\mathcal{W} \odot M_c^{(k)} = \big(\mathcal{W}[1] \odot M_c^{(k)}[1], \mathcal{W}[2] \odot M_c^{(k)}[2], \ldots, \mathcal{W}[T^{(k)}] \odot M_c^{(k)}[T^{(k)}], 0, \ldots, 0\big) \tag{27}$$

Let $F^{(k)}$ denote the local objective of client k. The corresponding masked objective at round c is

$$\Phi_c^{(k)}\big(\mathcal{W}; M_c^{(k)}\big) = F^{(k)}\big(\mathcal{W} \odot M_c^{(k)}\big) \tag{28}$$

Assumption 1 *(Masked-L-smoothness): For every $k \in [K]$, round c, sampled mask $M_c^{(k)}$, and $\mathcal{U}, \mathcal{V} \in R^d$,*

$$\|\nabla_{\mathcal{U}} \Phi_c^{(k)}(\mathcal{U}; M_c^{(k)}) - \nabla_{\mathcal{V}} \Phi_c^{(k)}(\mathcal{V}; M_c^{(k)})\|_2 \leq L\|\mathcal{U} - \mathcal{V}\|_2 \tag{29}$$

Assumption 2 *(Bounded block-wise client heterogeneity): For every parameter block $[T_{j-1} : T_j]$, there exist $\sigma_j \geq 0$ such that*

$$\sum_{k \in \mathcal{C}_j} \frac{N_k}{N_j} \|\nabla F^{(k)}(\mathcal{W}[T_{j-1} : T_j]) - \nabla F(\mathcal{W}[T_{j-1} : T_j])\|_2^2 \leq \sigma_j^2 \tag{30}$$

where $\mathcal{C}_j = \{k \in [K] : T^{(k)} \geq T_j\}, \sigma^2 = \sum_{j=1}^{J} \sigma_j^2$ and $N_j = \sum_{k \in \mathcal{C}_j} N_k$

Assumption 3 *(Bounded mask-induced gradient discrepancy): For every $k \in [K]$, round c, there exists $\delta_c^2 \in [0, 1)$ such that:*

$$\|\nabla F^{(k)}(\mathcal{W}_c) \odot M_c^{(k)} - \nabla F^{(k)}(\mathcal{W}_c \odot M_c^{(k)})\|_2^2 \leq \delta_c^2 \|\mathcal{W}_c\|_2^2 \tag{31}$$

At each communication round, the server samples A out of K clients uniformly without replacement, and each selected client performs E local updates.

Under Assumptions 1, 2, 3 and following the technique from the supplementary of (Wu et al., 2024), our proposed method satisfies:

$$\min_{c \in [C]} \mathbb{E}[\|\nabla F(\mathcal{W}_c)\|_2^2] \leq \frac{a_1(F(\mathcal{W}_0) - F^\star)K}{\eta_s \eta_l EAC} + \frac{a_2 K}{A} \eta_s \eta_l EL\sigma^2 + \frac{a_3 K}{C} \sum_{c \in [C]} \mathbb{E}[\delta_c^2 \|\mathcal{W}_c\|_2^2] \tag{32}$$

where $a_1, a_2, a_3 > 0$ are constants independent of C, and C, η_s, and η_l denote the total number of communication rounds, server learning rate, and local learning rate, respectively.

Choosing $\eta_s = 1, \eta_l = \frac{1}{E\sqrt{C}}$ and for sufficiently large C leads to:

$$\min_{c \in [C]} \mathbb{E}[\|\nabla F(\mathcal{W}_c)\|_2^2] \leq \mathcal{O}\left(\frac{K}{A} \frac{F(\mathcal{W}_0) - F^\star + \sigma^2}{\sqrt{C}}\right) + \mathcal{O}\left(\frac{K}{C} \sum_{c \in [C]} \mathbb{E}[\delta_c^2 \|\mathcal{W}_c\|_2^2]\right) \tag{33}$$

B Algorithmic Description

Our algorithm follows the scheme in (Kim et al., 2022; Ilhan et al., 2023; Wu et al., 2024). At each communication round, server uniformly samples a subset of clients and broadcasts the global parameters $\mathcal{W}^{(0)}$ and structural variables $(\boldsymbol{\alpha}^{(0)}, \boldsymbol{\beta}^{(0)})$. Each selected client k initializes its local parameters $\mathcal{W}^{(k)}$ using the subset of global parameters corresponding to its truncation, while preserving its local network structure. It then updates both $\mathcal{W}^{(k)}$ and $(\boldsymbol{\alpha}^{(k)}, \boldsymbol{\beta}^{(k)})$ through Eqn. (11). After local training, each client communicates its updated local model back to the server. The server aggregates these per Eqn. (13) and Eqn. (12) to obtain a new global model. Details are provided in Algorithm 1.

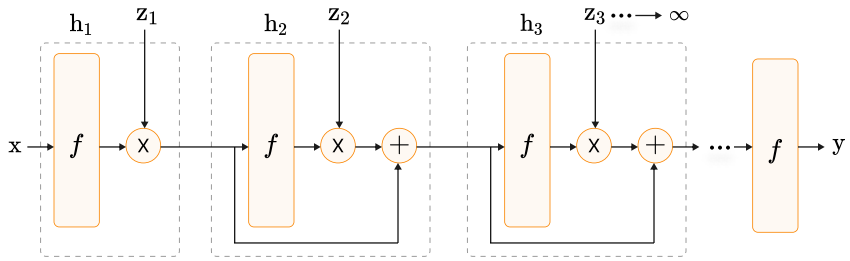

Figure 4: Block diagram of neural network. Each layer output is multiplied by a binary vector before residual addition. For CNNs, binary vector regularizes feature maps.

Table 5: Average test accuracy(%) on Cifar10 under high data heterogeneity ($\alpha = 0.3$). Besides "Global Model" performance, "Local Model (T_1 to T_4)" reports performance for each truncation level, and "Local Model Average" reports the average. Best results are bolded.

$(\boldsymbol{\alpha}^{(0)}, \boldsymbol{\beta}^{(0)})$	Local Model					Global Model
	T_1	T_2	T_3	T_4	Average	
(10.0, 0.5)	64.3 ± 1.8	73.1 ± 1.6	77.1 ± 0.3	76.3 ± 1.2	72.7 ± 0.6	72.6 ± 1.4
(10.0, 1.0)	74.1 ± 1.6	$\mathbf{77.7 \pm 1.1}$	$\mathbf{79.5 \pm 0.6}$	$\mathbf{78.8 \pm 2.2}$	$\mathbf{77.5 \pm 0.5}$	$\mathbf{76.3 \pm 0.6}$
(10.0, 2.0)	$\mathbf{74.7 \pm 0.9}$	74.6 ± 2.4	76.5 ± 1.1	74.7 ± 1.9	75.1 ± 1.3	73.3 ± 1.0
(10.0, 5.0)	67.5 ± 3.1	65.9 ± 3.2	67.6 ± 3.1	67.3 ± 3.2	67.1 ± 2.9	66.5 ± 2.7
(10.0, 10.0)	60.9 ± 0.7	59.9 ± 1.2	61.4 ± 1.1	60.8 ± 1.1	60.7 ± 0.2	60.9 ± 0.1
(5.0, 10.0)	56.3 ± 1.0	54.5 ± 1.9	56.6 ± 2.6	55.7 ± 1.8	55.8 ± 1.4	55.5 ± 1.5

C Experiments

C.1 Model with Adaptive Network Structure

For each client k, we define a neural network as follows:

$$\mathbf{h}_l^{(k)} = \sigma\left(\mathcal{W}_l^{(k)} \mathbf{h}_{l-1}^{(k)}\right) \odot \mathbf{z}_l^{(k)} + \mathbf{h}_{l-1}^{(k)},\ l \in \{1, 2, ..., \infty\} \tag{34}$$

where $\sigma(\cdot)$ consists of multiple operations such as convolution layer, non-linear activation, normalization layer, attention layer, linear layer, and so on (Figure 4). The global model also has a similar network configuration. For a fair comparison, we use the same network configuration for all baseline methods.

For image classification tasks on CIFAR10, CIFAR100, and Tiny-ImageNet datasets Krizhevsky et al. (2009); Yang et al., we adopt the pre-activated ResNet proposed in He et al. (2016); Nazaret and Blei (2022). Each pre-activated ResNet bottleneck block comprises three convolution layers (1×1, 3×3, 1×1), each preceded by static batch normalization Diao et al. (2021) and ReLU activation, with a residual shortcut connecting the block input directly to its output. For truncation level T, the network consists of three stages: the first two stages contain two pre-activated ResNet bottleneck blocks each, while the third contains $T - 4$ pre-activated ResNet bottleneck blocks.

For text classification tasks on AGNews and IMDB datasets Zhang et al. (2015); Maas et al. (2011), we use a Pre-LN Transformer architecture Xiong et al. (2020). Each Transformer block applies static layer normalization before self-attention, followed by a residual connection, and then applies static layer normalization before the feed-forward layer, followed by another residual connection. For truncation level T, the network contains T Transformer blocks.

Both of these techniques provide adaptable backbone networks. In both cases, the binary vector $\mathbf{z}_l^{(k)}$ is applied to block outputs before residual addition. A global average pooling layer is applied at the end of the network, followed by a classification layer. We emphasize that our goal is to demonstrate our framework's ability to infer network structure, rather than to search for an optimal architecture for the classification task.

Table 6: Average test accuracy(%) on Cifar10 under high data heterogeneity ($\alpha = 0.3$). Besides "Global Model" performance, "Local Model (T_1 to T_4)" reports performance for each truncation level, and "Local Model Average" reports the average. Best results are bolded.

γ	Local Model					Global Model
	T_1	T_2	T_3	T_4	Average	
1.0	71.9 ± 1.8	$\mathbf{78.9 \pm 0.5}$	$\mathbf{81.2 \pm 1.7}$	78.3 ± 1.2	$\mathbf{77.6 \pm 1.2}$	74.5 ± 1.2
5.0	$\mathbf{74.1 \pm 1.6}$	77.7 ± 1.1	79.5 ± 0.6	$\mathbf{78.8 \pm 2.2}$	77.5 ± 0.5	$\mathbf{76.3 \pm 0.6}$
10.0	73.7 ± 1.3	76.1 ± 0.9	77.7 ± 0.6	77.6 ± 1.8	76.3 ± 0.3	75.9 ± 0.2
20.0	72.4 ± 1.4	75.2 ± 0.7	77.1 ± 0.2	76.8 ± 1.9	75.3 ± 0.4	75.0 ± 0.4

Table 7: Average test accuracy(%) on Cifar10 under high data heterogeneity ($\alpha = 0.3$). Besides "Global Model" performance, "Local Model (T_1 to T_4)" reports performance for each truncation level, and "Local Model Average" reports the average. Best results are bolded.

KL coefficient	Local Model					Global Model
	T_1	T_2	T_3	T_4	Average	
10^{-2}	70.5 ± 1.2	74.7 ± 0.5	76.8 ± 0.5	76.5 ± 1.9	74.6 ± 0.2	75.3 ± 0.1
10^{-3}	73.2 ± 1.4	76.8 ± 0.6	78.0 ± 1.3	77.2 ± 1.6	76.3 ± 0.5	$\mathbf{76.6 \pm 0.4}$
10^{-4}	$\mathbf{74.1 \pm 1.6}$	$\mathbf{77.7 \pm 1.1}$	79.5 ± 0.6	$\mathbf{78.8 \pm 2.2}$	$\mathbf{77.5 \pm 0.5}$	76.3 ± 0.6
10^{-5}	72.1 ± 1.9	76.8 ± 2.2	$\mathbf{79.9 \pm 1.2}$	78.2 ± 0.8	76.8 ± 1.4	75.1 ± 1.8

C.2 Experimental configuration

Consistent with prior work (Wu et al., 2024), we adopt the same training configuration, setting the batch size to 20, number of local epochs to 5, local learning rate for network parameters to 0.1, local optimizer to SGD, and total number of communication rounds to 800, respectively.

C.3 Hyperparameters

For our proposed method, we generate 2 structural samples per iteration during training and 10 structural samples during inference. This setting balances the trade-off between performance and efficiency. We fine-tune the hyperparameters via grid search on CIFAR10 under high data heterogeneity ($\alpha = 0.3$), and use the selected configuration across all experimental settings. The resulting hyperparameter configuration uses a local learning rate of 1.0 for network structure variables, a global learning rate of 0.1, 5 global epochs, and Adam as the global structure optimizer.

We further perform hyperparameter sensitivity analyses for the global beta process prior parameters α, β, local beta-processes prior parameters γ, and the local $\mathbb{KL}$ coefficient, as shown in Tables 5, 6, and 7. The results show that our model is relatively insensitive to γ and the local $\mathbb{KL}$ coefficient. Based on this analysis, we select the best settings for $\alpha = 10.0, \beta = 1.0$, and set $\gamma = 5.0$ and the $\mathbb{KL}$ coefficient to 10^{-4}.

C.4 Hardware Specification

We conducted all experiments using NVIDIA A100 GPUs.

D Convergence and Efficiency Evaluation

D.1 Convergence Behavior

Figure 5 and 6 show the convergence behavior of various algorithms on CIFAR10 and CIFAR100. Our framework consistently converges faster than the baselines, with particularly notable gains on CIFAR100, where it achieves competitive performance in fewer than half the communication rounds. This reduction in communication rounds enhances our method's practical efficiency in real-world FL deployments.

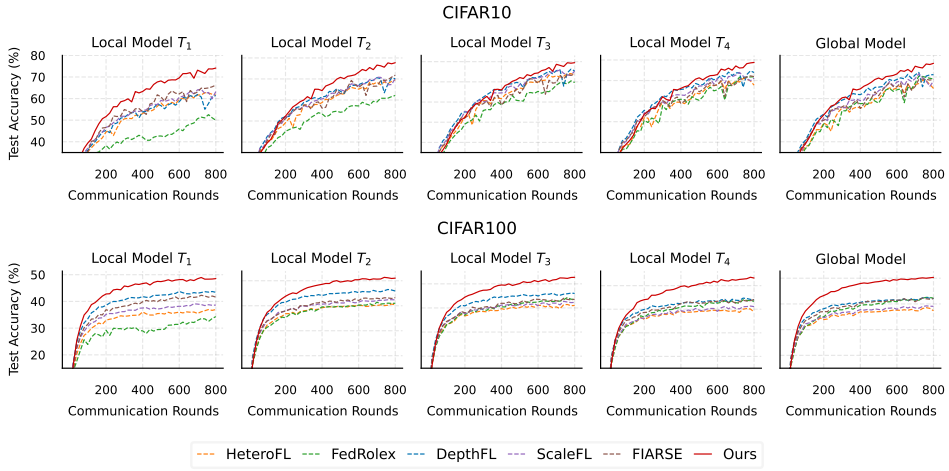

Figure 5: Test accuracy vs. communication rounds under high data heterogeneity level ($\alpha = 0.3$). Our framework shows rapid convergence.

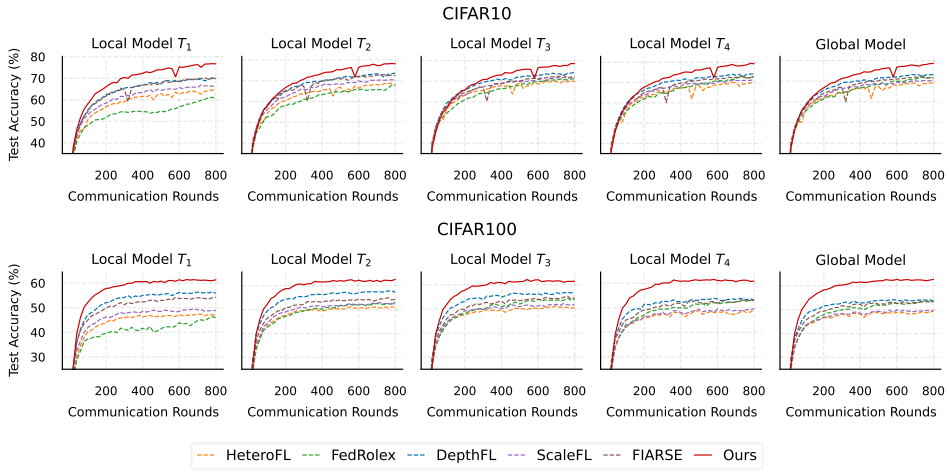

Figure 6: Test accuracy vs. communication rounds under low data heterogeneity level ($\alpha = 10.0$). Our framework shows rapid convergence.

D.2 End-to-end wall clock time

We measure the wall-clock time (hours) required to reach the predefined global model test accuracy targets on CIFAR100 under high ($\alpha = 0.3$) and low ($\alpha = 10.0$) data heterogeneity levels, with 10% of clients sampled from a total of 100 clients in each communication round. As shown in Table 8, our method is competitive at lower accuracy targets and becomes substantially more time-efficient as the target accuracy increases. In particular, under $\alpha = 0.3$, our method reaches 42.0% accuracy in 6.08 hours, compared with 11.71 hours for the next fastest method, and is the only method to reach 44.0% and 50.0% accuracy. Similarly, under $\alpha = 10.0$, our method reaches 54.0% accuracy in 4.06 hours, compared with 9.64 hours for the next fastest method, and is the only method to reach 60.0% accuracy. These results highlight that our method reaches target accuracies with substantially lower computational time than competing approaches, particularly when targeting higher model accuracy.

Table 8: Time (hours) required to reach target global model test accuracy on CIFAR-100 under different data heterogeneity levels.

α	Target Accuracy (%)	HeteroFL	FedRolex	DepthFL	ScaleFL	FIARSE	Ours
0.3	38.0	16.42	7.66	3.44	10.81	5.71	4.06
	40.0	-	11.97	5.86	-	9.51	4.73
	42.0	-	16.98	11.71	-	15.22	6.08
	44.0	-	-	-	-	-	8.11
	50.0	-	-	-	-	-	23.93
10.0	48.0	7.02	5.27	2.07	5.86	3.80	2.70
	50.0	-	7.66	2.76	17.01	5.71	2.70
	52.0	-	11.49	4.13	-	8.56	3.38
	54.0	-	18.96	9.64	-	-	4.06
	60.0	-	-	-	-	-	9.47

Table 9: Average training time (in seconds) per communication round on CIFAR100.

Methods	Average training time (seconds)
HeteroFL	84.2 ± 5.2
FedRolex	86.2 ± 5.6
DepthFL	62.0 ± 3.7
ScaleFL	81.1 ± 5.5
FIARSE	85.6 ± 3.4
Ours	121.7 ± 10.3

D.3 Computational Cost

We measure the average training time per communication round for several baseline methods and our proposed method. Specifically, we conduct experiments on CIFAR100 under a high data heterogeneity level ($\alpha = 0.3$), with 10% of clients sampled from a total of 100 clients in each communication round. As shown in Table 9, our method incurs a higher per-round training time than the compared baselines. However, it converges in fewer communication rounds, as shown in Figure 5 and Figure 6. This highlights a computation-communication trade-off, suggesting that our framework provides flexibility across different FL scenarios to balance per-round computational cost against the number of communication rounds required for convergence.

D.4 Communication Cost

We measure the total communication cost per round for the pre-activated ResNet models on CIFAR100, defined as the total number of communicated network parameters and structural variables. We compare the communication cost of our method with baseline methods. As shown in Table 10, our framework introduces only a negligible number of additional structure variables relative to the total communicated parameters. Moreover, the number of structural variables grows linearly with the local model truncation level, ranging from 20 for T_1 to 80 for T_4. Despite this minimal communication overhead, these structural variables enable client-specific customization of local network structures, leading to improved performance.

E Further Ablation Study

E.1 Local Network Structure Adaptation

Figure 7 presents the inferred layer-wise activation probabilities of the global model and the local models under varying truncation levels on CIFAR10 with low data heterogeneity. Compared with the high data heterogeneity scenario as shown in Figure 3, the client-level variability in the layer-wise activation probabilities of the local models is reduced. This reduced variability suggests that the inferred network structures adapt to varying degrees of data heterogeneity across clients.

Table 10: Comparison of the total number of communicated network parameters and structural variables across local models. 'M' denotes millions.

Local Model	Communicated parameters ($\mathcal{W}$)		Communicated structural variables ($\boldsymbol{\alpha}, \boldsymbol{\beta}$)	
	Baseline	Ours	Baseline	Ours
T_1	8.0M	8.0M	–	20
T_2	19.2M	19.2M	–	40
T_3	30.3M	30.3M	–	60
T_4	41.5M	41.5M	–	80

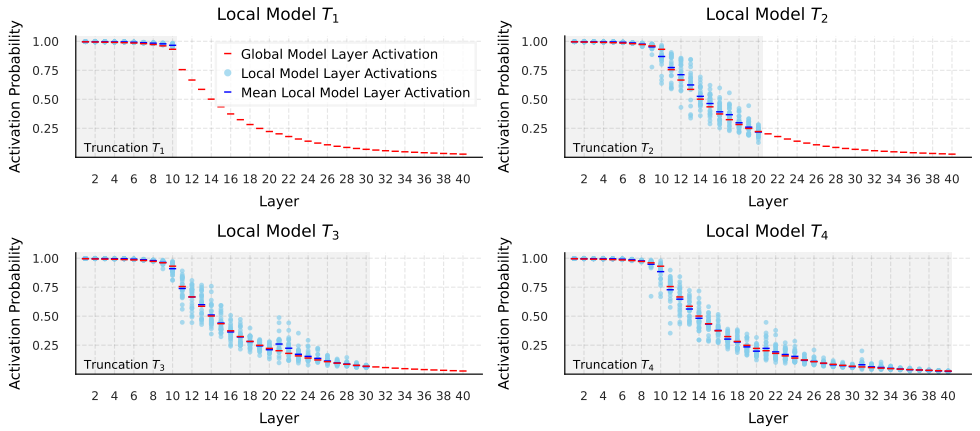

Figure 7: Inferred layer-wise activation probabilities of the global and local models across different truncation levels on CIFAR10 under low data heterogeneity($\alpha = 10.0$).

Table 11: Average test accuracy(%) on Cifar10 dataset under high data heterogeneity ($\alpha = 0.3$). Besides "Global Model" performance, "Local Model (T_1 to T_4)" reports performance for each truncation level, and "Local Model Average" reports the average across truncation levels. Best results are bolded.

Method	Local Model					Global Model
	T_1	T_2	T_3	T_4	Average	
DepthFL	71.4 ± 0.3	77.5 ± 1.0	78.5 ± 1.7	75.6 ± 0.2	75.8 ± 0.8	71.1 ± 0.5
FIARSE	69.3 ± 0.8	69.2 ± 1.1	70.9 ± 2.5	66.9 ± 1.8	69.1 ± 0.3	64.0 ± 1.7
Ours	$\mathbf{74.2 \pm 1.8}$	$\mathbf{80.1 \pm 1.2}$	$\mathbf{81.8 \pm 0.2}$	$\mathbf{77.8 \pm 0.4}$	$\mathbf{78.5 \pm 0.7}$	$\mathbf{74.2 \pm 1.6}$

E.2 Uneven Resource Allocation

To evaluate performance under uneven resource availability across clients, we follow the experimental setup of (Alam et al., 2022), which maps a real-world household income distribution to client computational capacities. Consequently, 55% of clients have computational capacity equivalent to truncation level T_1, 18% to T_2, 11% to T_3, and 16% to T_4. We conduct experiments on CIFAR10 under a high data heterogeneity level ($\alpha = 0.3$). As shown in Table 11, our method consistently outperforms the competing baselines. These results demonstrate the robustness of our framework to uneven computational resources across clients.