Identifiable Gated Residual Personalization for Federated Parameter-Efficient Fine-Tuning

Huimin Huang*, Wenhan Hu*, and Gang Yan✉

College of Computer Science and Technology, Jilin University, Changchun, China
huanghm2122@mails.jlu.edu.cn, huwh25@mails.jlu.edu.cn, gyan8@jlu.edu.cn
https://github.com/ahuang0324/fedsdg

Abstract. Federated parameter-efficient fine-tuning (PEFT) enables scalable adaptation of large pre-trained backbones under communication constraints. In heterogeneous federated settings, personalization is commonly realized through gated residual mixing between shared and client-specific branches. However, this parameterization is inherently scale non-identifiable: only the product of residual magnitude and mixing weight determines the functional contribution. When private residuals are optimized locally, their scales may drift across clients and communication rounds, rendering mixing coefficients an unreliable measure of personalization strength. We address this limitation with **FedSDG**, a structure-decoupled gating framework for federated PEFT. FedSDG maintains separate shared and private LoRA branches and introduces projection-level scalar gates for depth-aware personalization. Crucially, we propose *Dynamic Alignment*, a backbone-anchored calibration mechanism that stabilizes private residual scale prior to gated mixing, thereby improving gate identifiability without additional communication. Extensive experiments demonstrate consistent personalization improvements and stable depth-wise adaptation across diverse non-IID regimes.

Keywords: Personalized Federated Learning, Parameter-Efficient Fine-Tuning, Dual-Branch Residual Adaptation, Scale Identifiability

1 Introduction

Federated learning (FL) enables collaborative training across distributed clients without sharing raw data [33]. In realistic deployments, however, client data are rarely homogeneous: distributions often vary substantially in labels, domains, acquisition conditions, and feature statistics [21, 29]. Under such non-IID settings, a single global model may induce client drift, biased aggregation, and negative transfer [22], leaving uniform solutions suboptimal across clients. Personalized federated learning (PFL) therefore seeks to retain transferable structure while permitting controlled client-specific adaptation.

This challenge intensifies in contemporary transfer-learning regimes built upon large pre-trained backbones such as Vision Transformers (ViTs) [9]. Full-model federated fine-tuning is typically prohibitive due to communication cost

* Indicates equal contribution. ✉ Corresponding author. Contact: gyan8@jlu.edu.cn

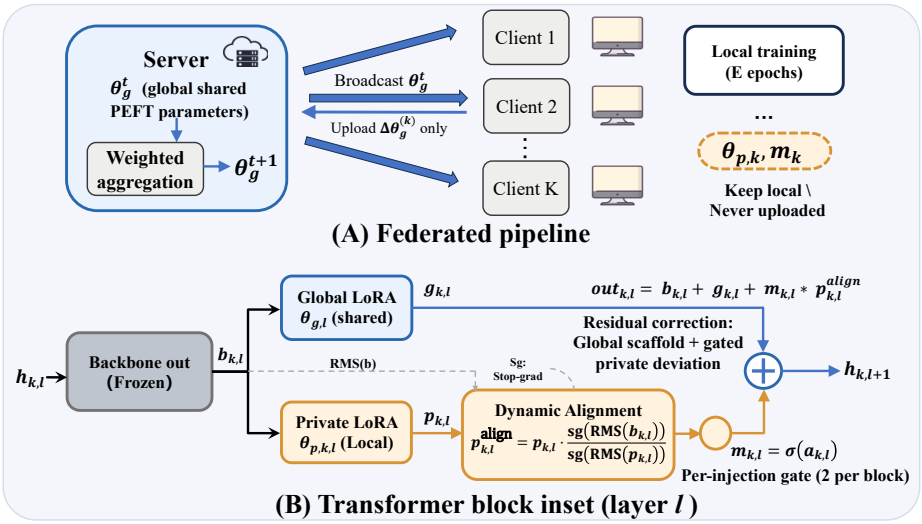


Fig. 1: Overview of FedSDG. Global adapters are aggregated across clients, while private adapters and gates remain local.

and optimization instability. Parameter-efficient fine-tuning (PEFT), particularly LoRA-based residual adaptation [20], alleviates these constraints by confining updates to lightweight residual modules. Yet this restriction reshapes the personalization problem. When adaptation is limited to a low-dimensional residual subspace, the question is no longer how to update the entire model, but where and to what extent client-specific corrections should be injected without disrupting shared representations. Under pronounced heterogeneity, misallocation of this limited residual capacity may either underfit client-specific structure or over-perturb transferable knowledge.

Existing federated PEFT approaches typically adopt shared residual modules or loosely coupled shared/private branches [35, 43, 45]. While effective in moderate regimes, these designs do not explicitly regulate how personalization is distributed across depth, nor do they ensure that personalization strength remains comparable across clients. In weighted residual mixing, the effective private contribution depends jointly on residual magnitude and mixing weight. Under heterogeneous local optimization and partial participation, residual scales may drift across clients and communication rounds [22, 29]. Because scaling the residual and inversely scaling its mixing coefficient preserves the functional output, personalization coefficients become scale non-identifiable. This ambiguity obscures interpretability, weakens sparsity-based control of personalization budgets, and can amplify optimization variability.

We address these issues with **FedSDG** (**Fed**erated **S**tructure-**D**ecoupled **G**ating), a structure-decoupled gating framework for federated PEFT. As shown in Figure 1, FedSDG separates adaptation into a shared LoRA branch aggregated across clients and a private LoRA branch maintained locally, and employs

projection-level scalar gates to regulate private residual injection across Transformer depth. This yields depth-aware and interpretable personalization while preserving communication efficiency. To eliminate scale non-identifiability, we further introduce *Dynamic Alignment (DA)*, a backbone-anchored calibration mechanism that stabilizes private residual scale prior to gated mixing. By reducing the scale degree of freedom between residual magnitude and mixing weight, DA improves comparability of personalization coefficients without altering the standard federated communication protocol.

This work formulates federated PEFT personalization as structured residual allocation and realizes it through a projection-level gated dual-path LoRA design for depth-aware adaptation. An intrinsic scale non-identifiability in weighted residual mixing under heterogeneous federated optimization is identified, clarifying its impact on interpretability and sparsity-based regulation. Dynamic Alignment eliminates this ambiguity by stabilizing private residual scale prior to gating without altering the communication protocol. Comprehensive experiments confirm consistent personalization improvements and stable depth-wise adaptation across non-IID regimes.

2 Related Work and Motivation

Personalized federated learning considers client-specific modeling in the presence of statistical heterogeneity, where data distributions differ across clients and a single global model may fail to generalize uniformly [5, 21, 28, 29, 40]. Heterogeneity affects not only predictive accuracy but also optimization dynamics, often leading to client drift under federated averaging [22, 29]. To accommodate such variability, prior work introduces personalization either by partitioning parameters into shared and client-specific components or by modifying the local objective. FedRep [5] and FedPer [2], for example, separate representation and head layers, retaining a common feature extractor while allowing client-dependent heads. Objective-based approaches instead regularize local updates toward a global reference; Ditto [28] employs a proximal term to balance stability and deviation, and related formulations include pFedMe [8] and Per-FedAvg [10]. Other directions cluster clients or maintain multiple global models to capture structured heterogeneity, at the cost of additional coordination.

As large pre-trained backbones such as Vision Transformers become standard [9], personalization increasingly unfolds in transfer-learning regimes. Full-model federated fine-tuning is often impractical in this setting, both in communication and in optimization. Parameter-efficient fine-tuning adapts frozen backbones through compact modules, including adapters, prefix or prompt tuning, and low-rank updates [18, 20, 27, 31]. LoRA in particular expresses weight updates ΔW via rank-r factors with a scaling coefficient, substantially reducing communication overhead [20]. In federated contexts, a straightforward strategy shares identical residual modules across clients and aggregates them via federated averaging. Such shared-only schemes can suffice when heterogeneity is limited, yet performance degrades under strong label skew or domain shift. Re-

cent work explores more structured variants, including partial sharing, sparsity constraints, heterogeneous rank allocation, and decomposed updates [35, 43, 45]. When adaptation is restricted to low-dimensional residual modules, the manner in which client-specific updates are allocated becomes increasingly consequential.

Separating shared and private residual branches provides one mechanism to increase flexibility. Dual-branch designs such as FedDPA [43] maintain parallel global and local PEFT modules, aggregating only the shared branch. In such formulations, personalization is implemented through weighted combinations of residual outputs. The resulting contribution of a private branch depends jointly on its magnitude and its mixing coefficient. Under heterogeneous local updates and aggregation dynamics [22, 29, 41, 42], residual scales can vary across clients and communication rounds. Since scaling the residual and inversely scaling its coefficient leaves the functional output unchanged, this parameterization introduces scale ambiguity. Consequently, mixing weights may not provide a consistent measure of personalization strength across layers or clients, and sparsity regularization on these coefficients may no longer correspond to a stable personalization budget. Addressing this ambiguity without increasing cross-client communication is the focus of our work.

3 FedSDG: Structure-Decoupled Gated Personalization

3.1 Problem Formulation

We consider federated learning with K clients [21, 33]. Client k holds a private dataset $\mathcal{D}_k$ sampled from a client-specific distribution $\mathcal{P}_k$, leading to statistical heterogeneity across clients [22, 29]. We adapt a pre-trained Vision Transformer (ViT) [9]. The backbone parameters $\boldsymbol{\theta}_{bb}$ remain frozen, and adaptation is performed using parameter-efficient fine-tuning modules [18, 20]. Freezing the backbone stabilizes optimization under non-IID updates while restricting communication to shared PEFT parameters.

Trainable parameters are partitioned into shared and private components. The shared parameters $\boldsymbol{\theta}_g$ are synchronized and aggregated across clients to capture cross-client structure [33]. The private parameters $\boldsymbol{\theta}_{p,k}$ remain local to client k and model client-dependent deviations [5, 28]. The shared branch encodes common structure, whereas the private branch provides a bounded local correction. The effective model on client k is

$$f_k(\cdot) = \mathcal{F}(\boldsymbol{\theta}_{bb}, \boldsymbol{\theta}_g, \boldsymbol{\theta}_{p,k}) \,, \tag{1}$$

where $\mathcal{F}(\cdot)$ composes the frozen backbone with shared and private adapters. In practice, $\mathcal{F}(\cdot)$ is realized via structure-decoupled dual-path LoRA with gated residual mixing [20, 43]. LoRA modules are inserted at selected linear projections of the Transformer. A global path parameterized by $\boldsymbol{\theta}_g$ participates in aggregation, while a private path parameterized by $\boldsymbol{\theta}_{p,k}$ remains on-device. A scalar gate is assigned to each injection site to modulate the private residual.

The following sections detail the dual-path LoRA construction and introduce Dynamic Alignment. Gate values and private residual magnitudes interact multiplicatively, inducing scale ambiguity that affects regularization and optimization under heterogeneous updates. DA anchors private residual energy to a backbone-derived reference, ensuring gate values represent a consistent injection ratio relative to the local activation scale without additional communication.

3.2 Gated Shared–Private Residual Decomposition

Personalization requires balancing cross-client generalization with controlled client-specific adaptation. We realize the composition $\mathcal{F}(\cdot)$ via a shared–private residual decomposition, where personalization is modeled as a residual correction to a shared scaffold. The shared branch captures transferable adaptation aggregated across clients, while the private branch introduces client-dependent deviations whose contribution at each injection site is modulated by a learnable gate.

Dual-Path LoRA Residual Parameterization. We implement the shared parameters $\boldsymbol{\theta}_g$ and private parameters $\boldsymbol{\theta}_{p,k}$ as dual-path LoRA adapters [20,43]. In each Transformer block, LoRA modules are inserted into the attention output projection W_o and the FFN output projection W_2. Each injected projection is indexed by $l \in \{1, \ldots, L_{\text{inj}}\}$, where $L_{\text{inj}} = 2L$ for a ViT with L blocks. Both shared and private adapters use the same rank r and scaling factor $s_{\text{lora}} = \alpha/r$, so behavioral differences arise solely from optimization and data heterogeneity rather than architectural asymmetry.

At each injected projection, client k maintains a scalar gate,

$$m_{k,l} = \sigma(a_{k,l}), \tag{2}$$

where $a_{k,l}$ is a learnable logit and $m_{k,l} \in (0, 1)$. The gate is broadcast across tokens and channels, modulating the residual energy at projection l without introducing fine-grained parameters. This design keeps the personalization mechanism low-dimensional and interpretable, so $m_{k,l}$ can be understood as a projection-level injection ratio rather than token-wise modulation.

Let $h_{k,l} \in \mathbb{R}^{B\times T\times d_l}$ denote the input feature to projection l on client k. Define the frozen backbone output $b_{k,l} = b_l(h_{k,l})$, the shared LoRA residual $g_{k,l} = g_l(h_{k,l}; \boldsymbol{\theta}_{g,l})$, and the private LoRA residual $p_{k,l} = p_l(h_{k,l}; \boldsymbol{\theta}_{p,k,l})$, all in the same output space [20]. The injected output takes the additive form:

$$\text{out}_{k,l} = b_{k,l} + g_{k,l} + m_{k,l} p_{k,l}. \tag{3}$$

In this decomposition, $g_{k,l}$ captures shared adaptation aggregated across clients, whereas the gated private residual $m_{k,l}p_{k,l}$ models deviations from the shared scaffold [5,28]. The additive structure preserves the backbone representation and treats personalization as a correction term. By default, the classifier head is included in $\boldsymbol{\theta}_g$ to maintain a common decision layer; when label spaces or decision boundaries diverge, the head may instead be privatized. This modification affects only parameter locality and leaves the residual mixing unchanged.

Regularizing Personalization Strength. While the residual formulation enables flexible personalization, it is equally important to control its strength. To this end, we bias the model toward shared adaptation by initializing gate logits with a negative constant,

$$a_{k,l} \leftarrow a_{\text{init}} < 0, \tag{4}$$

so that gates start from small values. This shared-scaffold-first strategy ensures that early training primarily stabilizes the shared branch before substantial private corrections are introduced, thereby mitigating instability under heterogeneous local updates [22, 29].

In addition, we impose an ℓ_1 penalty on the gate values,

$$\mathcal{R}_{gate}(\mathbf{m}_k) = \lambda_1 \sum_{l=1}^{L_{\text{inj}}} m_{k,l}, \tag{5}$$

which encourages sparse activation of private residuals across projection sites. When the private residual scale is stable, $\|\mathbf{m}_k\|_1$ admits a natural interpretation as the relative personalization budget allocated across layers. However, this interpretation presupposes that the magnitude of $p_{k,l}$ remains comparable across rounds and clients. If the private residual scale drifts, identical gate values may correspond to markedly different effective corrections.

Scale Non-Identifiability in Gated Mixing. A closer examination of Eq. 3 reveals that the effective private contribution depends solely on the product $m_{k,l}p_{k,l}$. Indeed, for any scalar $c > 0$ such that $cm_{k,l} \in (0, 1)$,

$$m'_{k,l} = cm_{k,l}, \quad p'_{k,l} = p_{k,l}/c \tag{6}$$

preserve the injected residual $m_{k,l}p_{k,l}$. Consequently, the parameterization is scale non-identifiable: infinitely many $(m_{k,l}, p_{k,l})$ pairs represent the same functional model.

Proposition 1. *Consider the local objective with gate regularization,*

$$\mathcal{L}_k = \mathcal{L}_{task} + \lambda_1 \sum_l m_{k,l} + \lambda_2 \|\boldsymbol{\theta}_{p,k}\|_2^2. \tag{7}$$

Under the reparameterization $m'_{k,l} = cm_{k,l}$ and $p'_{k,l} = p_{k,l}/c$, the task loss remains unchanged, while the regularization terms transform as $\sum_l m'_{k,l} = c\sum_l m_{k,l}$ and $\|\boldsymbol{\theta}'_{p,k}\|_2^2 = \|\boldsymbol{\theta}_{p,k}\|_2^2/c^2$. Hence, the relative influence of the two regularizers can be altered without affecting the functional output.

Proof. Because the task loss depends only on $m_{k,l}p_{k,l}$, it is invariant under the stated reparameterization. The ℓ_1 and ℓ_2 regularizers scale linearly and quadratically with c, respectively, which yields the stated transformation.

In federated settings, this coupling has concrete implications. Private adapters are optimized locally and never aggregated, so their output magnitudes may

drift across clients and communication rounds [28, 33, 43]. Under scale non-identifiability, the optimizer can trade off $\|p_{k,l}\|$ and $m_{k,l}$ without affecting the task loss but altering regularization. Consequently, (i) gate values lose a stable correspondence to injected residual energy, (ii) $\|\mathbf{m}_k\|_1$ no longer reflects a consistent personalization budget, and (iii) gate gradients, proportional to the residual magnitude, become sensitive to scale drift. This necessitates an explicit calibration step to fix the private residual scale.

3.3 Backbone-Anchored Energy Calibration

As discussed in Sec. 3.2, gated residual mixing is scale non-identifiable: multiple $(m_{k,l}, p_{k,l})$ pairs can represent the same functional model while inducing different regularization costs. To remove this ambiguity, we introduce *Dynamic Alignment* (DA), a backbone-anchored calibration mechanism that fixes the scale of the private residual prior to gating.

Backbone-Anchored RMS Alignment. To make this idea concrete, consider injection module l on client k, where $\text{out}_{k,l} = b_{k,l} + g_{k,l} + m_{k,l}p_{k,l}$. Rather than directly modulating $p_{k,l}$, we first calibrate it and rewrite the residual contribution as

$$r_{k,l} = g_{k,l} + m_{k,l} \cdot \text{Align}(p_{k,l}, b_{k,l}), \tag{8}$$

where $\text{Align}(p_{k,l}, b_{k,l}) \triangleq p_{k,l}^{\text{align}}$ and the backbone term $b_{k,l}$ is reintroduced in Eq. 15. This calibration step ensures that the private residual has a controlled and interpretable scale before being modulated by $m_{k,l}$.

More specifically, let $p_{k,l} \in \mathbb{R}^{B\times T\times d_l}$ denote the private residual after LoRA scaling in the projection output space. We first normalize it using the per-sample root-mean-square (RMS) over token and channel dimensions,

$$\hat{p}_{k,l} = \frac{p_{k,l}}{\text{RMS}(p_{k,l})}, \qquad \text{RMS}(z) = \sqrt{\text{mean}_{t,c}(z^2) + \epsilon}, \tag{9}$$

where $\epsilon = 10^{-6}$ and the statistic is computed independently for each sample in the batch and then broadcast to $[B, T, d_l]$. Gradients are stopped through $\text{RMS}(p_{k,l})$ so that the model cannot manipulate the denominator to implicitly rescale the residual.

At this point, one might consider anchoring to the shared residual $g_{k,l}$. However, because $g_{k,l}$ evolves across communication rounds through aggregation, its magnitude is not guaranteed to be round-stable. We therefore anchor instead to the frozen backbone output under the same activation,

$$b_{k,l} = b_l(h_{k,l}), \tag{10}$$

whose mapping remains fixed throughout training. We then define

$$p_{k,l}^{\text{align}} = \hat{p}_{k,l} \cdot \text{RMS}(b_{k,l}), \tag{11}$$

and similarly stop gradients through $\text{RMS}(b_{k,l})$. Equivalently,

$$p_{k,l}^{\text{align}} = p_{k,l} \cdot \frac{\text{sg}(\text{RMS}(b_{k,l}))}{\text{sg}(\text{RMS}(p_{k,l}))}, \tag{12}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient. Since LoRA residuals incorporate the scaling factor $s_{\mathrm{lora}} = \alpha/r$, alignment is applied directly to the injected residual.

Proposition 2. *Let* $p_{k,l}^{\mathrm{align}} = \frac{p_{k,l}}{\mathrm{RMS}(p_{k,l})}\,\mathrm{RMS}(b_{k,l})$. *Then, for each sample,*

$$\mathrm{RMS}(p_{k,l}^{\mathrm{align}}) = \mathrm{RMS}(b_{k,l}), \qquad \mathrm{RMS}(m_{k,l}p_{k,l}^{\mathrm{align}}) = m_{k,l}\,\mathrm{RMS}(b_{k,l}). \tag{13}$$

Proof. By construction, $\mathrm{RMS}(p_{k,l}^{\mathrm{align}}) = \mathrm{RMS}(b_{k,l})$. Since $m_{k,l}$ is scalar and broadcast over tokens and channels, $\mathrm{RMS}(m_{k,l}p_{k,l}^{\mathrm{align}}) = m_{k,l}\,\mathrm{RMS}(b_{k,l})$.

Corollary 1. *After alignment, the gate* $m_{k,l}$ *directly controls the private residual energy relative to the backbone activation scale. Consequently, within a client and across rounds,* $\|\mathbf{m}_k\|_1$ *corresponds to a consistent allocation of projection-level injection ratios.*

In other words, DA eliminates the scale degree of freedom that couples $m_{k,l}$ and $p_{k,l}$ in Eq. 3. Although $\mathrm{RMS}(b_{k,l})$ may differ across clients due to distributional heterogeneity, it remains stable for a fixed backbone and input. Thus, gate values encode a well-defined proportion of backbone-scale residual energy rather than compensating for private residual drift. Beyond interpretability, alignment also stabilizes optimization. Ignoring broadcasting notation,

$$\frac{\partial \mathcal{L}}{\partial a_{k,l}} = \frac{\partial \mathcal{L}}{\partial \mathrm{out}_{k,l}} \cdot \sigma(a_{k,l})(1-\sigma(a_{k,l})) \cdot p_{k,l}^{\mathrm{align}}. \tag{14}$$

$\mathrm{RMS}(p_{k,l}^{\mathrm{align}})$ is tied to the round-stable $\mathrm{RMS}(b_{k,l})$, so the magnitude of gate gradients no longer scales with variations of $\|p_{k,l}\|$, reducing optimization variability under heterogeneous local updates.

Scale-Calibrated Residual Decoupling. With alignment in place, the final mixing rule at injection module l on client k becomes

$$\mathrm{out}_{k,l} = b_{k,l} + g_{k,l} + m_{k,l} \cdot p_{k,l}^{\mathrm{align}}, \qquad p_{k,l}^{\mathrm{align}} = \frac{p_{k,l}}{\mathrm{RMS}(p_{k,l})} \cdot \mathrm{RMS}(b_{k,l}). \tag{15}$$

FedSDG decouples personalization along three complementary axes: structural decoupling of shared and private adapters, scale decoupling via RMS alignment, and gradient decoupling through detached statistics. Since $\mathrm{RMS}(b_{k,l})$ is computed locally from the frozen backbone, DA introduces no communication overhead beyond standard federated PEFT [20, 35]. At each communication round, client k optimizes $(\boldsymbol{\theta}_g^{(k)}, \boldsymbol{\theta}_{p,k}, \mathbf{a}_k)$ by minimizing:

$$\mathcal{L}_k = \mathcal{L}_{task}(f_k; \mathcal{D}_k) + \lambda_1 \|\mathbf{m}_k\|_1 + \lambda_2 \|\boldsymbol{\theta}_{p,k}\|_2^2, \tag{16}$$

where $\mathcal{L}_{task}$ is the cross-entropy computed using f_k (Eq. 1). Here, λ_1 controls projection-level sparsity of private injection, while λ_2 limits client-specific capacity and residual magnitude under purely local optimization. For consistency, the same (λ_1, λ_2) are used across all clients.

After local training, only the global update $\Delta\boldsymbol{\theta}_g^{(k)}$ is uploaded and aggregated [29, 33, 35]. The private adapters $\boldsymbol{\theta}_{p,k}$ and gate logits $\mathbf{a}_k$ remain strictly on-device. Likewise, backbone-derived statistics such as $\mathrm{RMS}(b_{k,l})$ are computed locally during forward propagation and are never transmitted.

3.4 Federated Optimization and Overhead Analysis

At communication round t, the server samples a subset of clients and broadcasts the shared adapter parameters $\boldsymbol{\theta}_g^t$. Each selected client initializes $\boldsymbol{\theta}_g^{(k)} \leftarrow \boldsymbol{\theta}_g^t$ and performs local optimization of $(\boldsymbol{\theta}_g^{(k)}, \boldsymbol{\theta}_{p,k}, \mathbf{a}_k)$ on $\mathcal{D}_k$ using the aligned residual mixing rule in Eq. 15. After E local epochs, only the shared update $\Delta\boldsymbol{\theta}_g^{(k)}$ is transmitted for aggregation. Private adapters $\boldsymbol{\theta}_{p,k}$, gate logits $\mathbf{a}_k$, and backbone-derived alignment statistics remain strictly on-device. Communication is therefore restricted to the shared branch.

We next analyze parameter and computational overhead. Let L denote the number of Transformer blocks, d the hidden dimension, and $r \ll d$ the LoRA rank. Each injected projection introduces two low-rank matrices of size $d \times r$ and $r \times d$, contributing $O(dr)$ parameters per branch. With two projections (W_o, W_2) per block and the dual-path design, each branch contains $O(Ldr)$ parameters per client, while gate logits add only $O(L)$ scalars. Only the shared branch contributes to communication cost.

In terms of computation, each LoRA branch incurs $O(BTdr)$ additional FLOPs per injection, where B is the batch size and T the number of tokens. The dual-path structure doubles this term but preserves linear dependence on r. Dynamic Alignment requires per-sample RMS statistics over tokens and channels at $O(BTd)$ cost. When $r \ll d$, the dominant complexity remains $O(BTdr)$, so alignment introduces only a minor constant overhead without changing asymptotic scaling. At each round, $\boldsymbol{\theta}_g$ of size $O(Ldr)$ is communicated, while all private parameters and alignment computations remain local.

4 Experiments

4.1 Experimental Setup

Datasets and Partitions. We evaluate on CIFAR-100, Tiny-ImageNet, Office-Home, and Mini-DomainNet [6, 25, 26, 37, 39] benchmark datasets. Images are resized to 224×224 and normalized using the ImageNet mean and standard deviation. We consider (i) global-α label skew on CIFAR-100/Tiny-ImageNet with $\alpha \in \{0.1, 0.3, 0.5, 1.0\}$ [19]; (ii) domain partition on Office-Home and Mini-DomainNet, where each client corresponds to a single domain, following standard domain-shift evaluation protocols [14, 16, 24, 38]; (iii) *dom-*α, where clients are first grouped by domain and then class-wise Dirichlet(α) partitioning is applied within each domain group. The original test splits are kept intact; for reporting average local accuracy, we evaluate each client on the subset of the original test split matching its domain/label support.

Baselines and Metric. We compare FedSDG with Local-only, FedAvg [33], FedProx+LoRA [29], FedRep [5], Ditto [28], FedAvg+LoRA, and FedDPA [43]. We report *average local top-1 test accuracy*, where clients are evaluated on their corresponding held-out local test subsets and results are averaged across participating clients. Results are reported as mean±std over three random seeds {42,123,456}, averaged over the last 10 communication rounds during training.

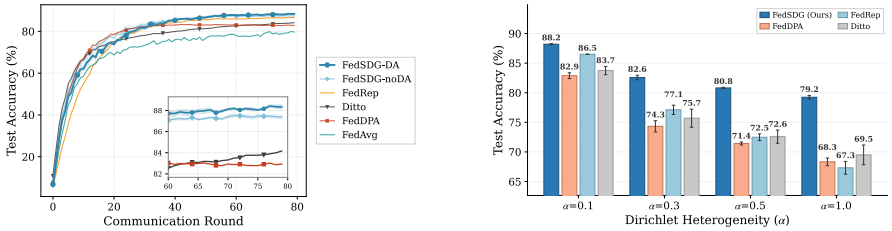
Table 1: Main results. Average local test accuracy (%) across four benchmarks. Results are reported as mean±std over three seeds and averaged over the last 10 communication rounds. Avg.: mean over all reported settings.

Setting	Standard			PFL		PEFT-FL		FedSDG
	Local	FedAvg	FP+Lo	FedRep	Ditto	FA+Lo	FedDPA	**(Ours)**
CIFAR-100 *(Label Skew)*								
$\alpha = 0.1$	65.16 (1.50)	79.22 (1.10)	71.88 (0.76)	86.50 (0.03)	83.75 (0.70)	71.67 (0.83)	82.88 (0.50)	**88.20 (0.09)**
$\alpha = 0.3$	47.46 (2.15)	82.20 (0.49)	77.06 (0.43)	77.11 (0.78)	75.70 (1.53)	76.34 (0.95)	74.33 (0.94)	**82.57 (0.38)**
$\alpha = 0.5$	40.17 (2.53)	**83.02 (0.35)**	75.58 (2.20)	72.48 (0.56)	72.58 (1.11)	78.16 (0.25)	71.42 (0.25)	80.81 (0.11)
$\alpha = 1.0$	32.82 (2.32)	**82.95 (0.83)**	79.06 (0.26)	67.31 (1.08)	69.50 (1.67)	79.16 (0.27)	68.32 (0.65)	79.24 (0.31)
Tiny-ImageNet *(Label Skew)*								
$\alpha = 0.1$	53.75 (0.83)	71.28 (0.08)	63.35 (0.47)	79.23 (0.55)	77.43 (0.57)	61.46 (0.46)	73.28 (0.24)	**80.63 (0.57)**
$\alpha = 0.3$	36.65 (0.24)	**74.35 (0.19)**	69.00 (0.38)	68.10 (0.27)	69.30 (0.63)	68.22 (0.32)	64.40 (0.12)	74.23 (0.05)
$\alpha = 0.5$	29.40 (1.08)	**74.96 (0.27)**	70.68 (0.18)	62.84 (0.80)	66.81 (0.92)	70.24 (0.21)	61.95 (0.23)	72.75 (0.25)
$\alpha = 1.0$	22.79 (1.01)	**75.15 (1.01)**	72.17 (0.44)	56.66 (0.42)	64.80 (0.57)	72.06 (0.39)	58.81 (0.47)	71.54 (0.38)
Office-Home *(Feature Shift)*								
dom-0.1	55.11 (0.99)	72.34 (0.08)	56.43 (0.64)	69.56 (0.64)	70.53 (0.89)	55.00 (0.38)	73.36 (0.25)	**75.40 (0.46)**
dom-0.3	39.93 (1.41)	**74.65 (0.18)**	63.02 (0.09)	54.67 (0.81)	60.22 (0.82)	61.66 (0.22)	64.84 (1.53)	70.40 (0.95)
dom-0.5	33.89 (0.78)	**75.44 (0.26)**	64.84 (0.45)	48.86 (0.67)	56.08 (1.08)	63.78 (0.23)	62.14 (0.41)	68.59 (0.54)
dom-1.0	26.65 (0.64)	**76.37 (0.19)**	67.05 (0.28)	40.44 (0.77)	53.06 (0.67)	65.93 (0.52)	59.81 (0.29)	67.65 (0.28)
global-0.1	50.26 (0.42)	72.81 (0.58)	56.73 (0.22)	66.94 (0.70)	67.48 (1.40)	55.72 (0.46)	69.11 (0.88)	**75.76 (1.11)**
Mini-DomainNet *(Feature Shift)*								
dom-0.1	48.48 (0.07)	38.36 (0.96)	52.40 (0.14)	58.26 (0.13)	62.02 (0.15)	23.14 (0.94)	56.74 (0.10)	**62.80 (0.11)**
global-0.1	49.39 (0.21)	41.99 (0.18)	59.07 (0.20)	59.41 (0.31)	62.95 (0.07)	26.72 (0.14)	57.02 (0.27)	**63.58 (0.25)**
Avg.	42.13	71.67	66.55	64.56	67.48	61.95	66.56	**74.27**

Model and Training. We use an ImageNet-21k pre-trained ViT-Tiny [9]. LoRA modules (r=8, α=16) are inserted into the attention output projection (W_o) and FFN output projection (W_2) of each transformer block (24 sites in total). We simulate K=100 clients for R=80 rounds with participation rate C=0.1 and E=5 local epochs, using batch size 128 and Adam [23]. We set η_{lora}=10^{-3} and $\eta_{\text{head}} \in \{10^{-3}, 10^{-2}\}$ (dataset-specific). FedSDG initializes gates at -3 with η_{gate}=0.05, λ_1=10^{-3}, and λ_2=5×10^{-4}. DA uses backbone-anchored RMS (DA_{base}) with stop-gradient. On Mini-DomainNet, we use private heads for dual-branch methods (FedSDG/FedDPA) for a structure-matched comparison. For all other datasets, we use a shared head unless stated otherwise. All experiments are conducted on NVIDIA RTX 5090 GPUs with 32GB memory.

4.2 Main Results

We evaluate FedSDG on four benchmarks covering label skew and domain shift (Sec. 4.1). Table 1 shows that FedSDG achieves the best or most consistent *average local test accuracy* across settings. On average across all tasks, FedSDG reaches 74.27%, outperforming the strongest standard baseline FedAvg (71.67%) by +2.60 points and the recent PEFT-FL method FedDPA (66.56%) by +7.71



(a) Convergence curve (α=0.1) **(b)** Accuracy under varying α

Fig. 2: CIFAR-100 results. FedSDG shows stable convergence and strong robustness across heterogeneity levels.

Table 2: Additional comparisons and variants. ViT-Tiny local accuracy (%).

Method	C100$_{0.1}$	C100$_{1.0}$	Tiny$_{0.1}$	Tiny$_{1.0}$	Avg.
Recent federated LoRA baselines					
FedSA-LoRA	78.82 (1.05)	61.05 (2.28)	71.64 (0.65)	57.44 (0.57)	67.24
FedALT	81.37 (1.25)	70.13 (0.94)	70.28 (1.69)	67.04 (0.82)	72.21
Controlled variants					
TS-LoRA	82.82 (0.87)	69.85 (0.33)	71.71 (0.38)	57.22 (0.42)	70.40
FedDPA+DA$_g$	86.41 (0.30)	77.03 (0.07)	78.76 (0.64)	68.88 (0.38)	77.77
Reference methods					
FedDPA	82.88 (0.50)	68.32 (0.65)	73.28 (0.24)	58.81 (0.47)	70.82
FedSDG	**88.20 (0.09)**	**79.24 (0.31)**	**80.63 (0.57)**	**71.54 (0.38)**	**79.90**

points. The advantage becomes more pronounced under strong heterogeneity. For example, on CIFAR-100 with α=0.1, FedSDG achieves 88.20%, improving over FedRep (86.50%) by +1.7 points and over FedDPA (82.88%) by +5.3 points. Figure 2 further indicates stable optimization at α=0.1 and strong robustness as the degree of heterogeneity varies.

On domain-shift benchmarks, FedSDG consistently outperforms the closest dual-branch baseline FedDPA. For instance, on Office-Home dom-0.1, FedSDG achieves 75.40% compared to 73.36% for FedDPA. These results support the effectiveness of structure-decoupled gated personalization under distribution shifts.

4.3 Additional Validation

Additional comparisons and controlled variants. Table 2 compares FedSDG with recent federated LoRA baselines and controlled variants on representative label-skew settings. FedSDG improves the average accuracy from 72.21% to 79.90% over the stronger recent baseline [3, 15]. TS-LoRA's weaker result shows that sequential shared-to-private adaptation cannot replace jointly optimized, depth-aware gated residual allocation; FedDPA+DA$_g$ improves FedDPA from 70.82% to 77.77%, while FedSDG further benefits from learnable structure-decoupled gates.

Table 3: Scale-up and OOD validation. Local accuracy (%).

Method	ViT-Base					ImageNet-R		
	$\mathbf{TN}_{0.1}$	$\mathbf{TN}_{1.0}$	$\mathbf{OH}_{d0.1}$	$\mathbf{OH}_{g0.1}$	**Avg.**	α=0.1	α=1.0	**Avg.**
FedRep	91.63 (0.11)	82.25 (0.19)	79.92 (0.15)	79.65 (0.32)	83.36	68.40 (0.16)	38.82 (0.21)	53.61
FedDPA	78.02 (0.71)	78.65 (0.22)	75.17 (0.94)	72.55 (1.04)	76.10	53.22 (0.35)	38.92 (0.62)	46.07
FedALT	87.52 (1.03)	89.74 (1.04)	**89.89 (0.70)**	88.78 (0.51)	88.98	78.62 (0.40)	66.92 (7.47)	72.77
FedSA-LoRA	76.52 (0.27)	67.66 (0.41)	68.41 (0.46)	64.38 (0.39)	69.24	39.41 (0.35)	23.98 (0.21)	31.70
FedSDG	**92.47 (0.16)**	**90.65 (0.15)**	85.94 (0.80)	**90.05 (0.20)**	**89.78**	**79.25 (0.20)**	**72.05 (0.14)**	**75.65**

Table 4: Gating ablation. Average local test accuracy (%).

Variant	**Tiny** (α=0.1)	**Office** (dom-α=0.1)
Fixed gate (m constant)		
m=0 (no pers.)	64.77 (0.24)	56.16 (0.57)
m=0.047	76.63 (0.64)	70.55 (0.39)
m=0.5	44.78 (2.21)	45.98 (0.92)
m=1.0 (full pers.)	31.05 (0.45)	36.88 (0.89)
Initialization (a_{init})		
a_{init}=−1	77.62 (0.41)	70.19 (0.58)
a_{init}=0	52.89 (3.30)	49.17 (1.03)
Granularity		
Coarse (shared blocks)	76.57 (0.64)	70.59 (0.21)
FedSDG (ours)	**80.63 (0.57)**	**75.40 (0.46)**

Scale-up and OOD generalization. Table 3 scales the backbone to ViT-Base on Tiny-ImageNet and Office-Home, where FedSDG achieves the best average accuracy (89.78%) and leads on three of four settings. On ImageNet-R OOD evaluation, FedSDG again gives the best average accuracy (75.65%) and remains stable across heterogeneity levels, supporting robust identifiable gated residual personalization under distribution shift.

4.4 Ablation Studies

Gating Design Ablation. We ablate the gating design by comparing fixed gates (constant m), learned projection-level gates with different logit initializations a_{init}, and coarse-grained gates shared across blocks. Results in Table 4 show that adaptive gating is important for reliable personalization in heterogeneous settings. Fixed gates are highly sensitive to m: on Tiny-ImageNet (α=0.1), accuracy ranges from 76.63% (m=0.047) to 31.05% (m=1.0), and on Office-Home (dom-α=0.1) from 70.55% to 36.88%. This indicates that a fixed trade-off is often insufficient, while learning gates improves robustness from 76.63% to 80.63% on Tiny-ImageNet (+4.0) and from 70.55% to 75.40% on Office-Home (+4.8). Coarse-grained gating performs reasonably (76.57% / 70.59%) but remains below the default projection-level design, suggesting that fine-grained control across different layers is beneficial.

Table 5: Dynamic Alignment effect. Local test accuracy (%) with vs. without DA.

Dataset	Variant	α=0.1	α=0.3	α=0.5	α=1.0
CIFAR-100	FedSDG w/o DA	87.44 (0.14)	81.42 (0.60)	79.13 (0.20)	77.22 (0.48)
	FedSDG+DA (**ours**)	**88.20** (0.09)	**82.57** (0.38)	**80.81** (0.11)	**79.24** (0.31)
Tiny-ImageNet	FedSDG w/o DA	79.32 (0.30)	72.98 (0.16)	70.21 (0.52)	68.24 (0.64)
	FedSDG+DA (**ours**)	**80.63** (0.57)	**74.23** (0.05)	**72.75** (0.25)	**71.54** (0.38)

Table 6: DA anchor choice. Local test accuracy (%) for different alignment targets.

Target	**Tiny** (α=0.1)	**Office-Home** (dom-α=0.1)
DA_{global} (moving)	80.09 (0.43)	64.64 (1.20)
DA_{base} (**ours**)	**80.63** (0.57)	**75.40** (0.46)

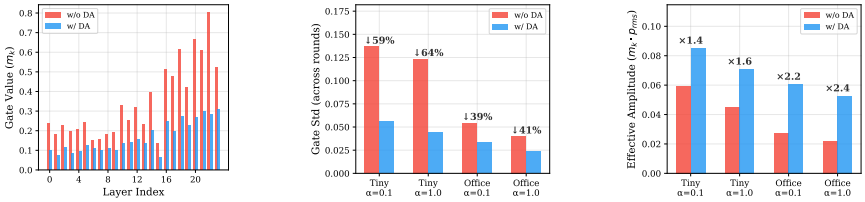
Dynamic Alignment under Heterogeneity. We evaluate Dynamic Alignment (DA) across $\alpha \in \{0.1, 0.3, 0.5, 1.0\}$ on CIFAR-100 and Tiny-ImageNet benchmarks. Table 5 shows consistent improvements from DA across all settings. On CIFAR-100, accuracy improves from 87.44% to 88.20% at α=0.1 (+0.76) and from 77.22% to 79.24% at α=1.0 (+2.02). On Tiny-ImageNet, gains range from +1.31 to +3.30 points. These gains show that DA consistently mitigates residual scale mismatch across label-skew levels.

Alignment Anchor Choice. We compare the default backbone-anchored target (DA_{base}) with a global-branch target (DA_{global}). Results in Table 6 show that DA_{base} performs better, especially on domain-shift benchmarks. For example, on Office-Home (dom-α=0.1), accuracy drops from 75.40% to 64.64% when using DA_{global}. A smaller gap is observed on Tiny-ImageNet (80.63% vs. 80.09%). This suggests that a stable backbone-referenced target is more reliable than a moving global target under heterogeneous aggregation.

4.5 Mechanistic Analysis

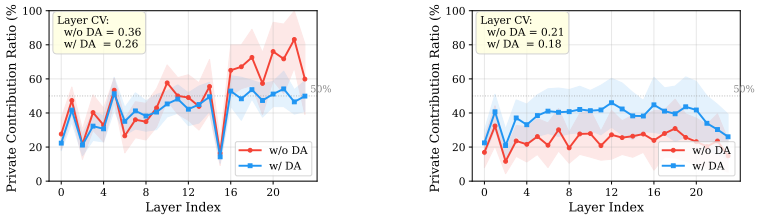
Scale-confounded Gating and Dynamic Alignment. Dynamic Alignment addresses *scale-confounded gating* in residual mixing, where gates may compensate for branch-scale mismatch while encoding private/shared preference. Figure 3 illustrates this effect through layer-wise gates, cross-round gate fluctuation, and effective private amplitude $m \cdot \text{RMS}(p^{align})$. Without DA, gates exhibit larger inter-layer variance and stronger temporal fluctuation; with DA, trajectories stabilize while effective private amplitude increases, indicating more consistent use of private residuals.

Layer-wise Private Contribution. We quantify the relative influence of the private branch via the *private contribution ratio* $\frac{m \cdot \text{RMS}(p^{align})}{(1-m) \cdot \text{RMS}(g) + m \cdot \text{RMS}(p^{align})}$. As shown in Fig. 4 under severe heterogeneity (α=0.1), DA produces a more balanced and layer-consistent private/shared mixture. This indicates improved gate identifiability and more stable personalized contributions across layers.



(a) Layer-wise gate values **(b)** Cross-round variance **(c)** Private residual scale

Fig. 3: Mechanistic analysis of DA.



(a) Tiny-ImageNet (α=0.1) **(b)** Office-Home (α=0.1)

Fig. 4: Layer-wise private contribution ratio (α=0.1). Private-branch contribution across layers under severe heterogeneity.

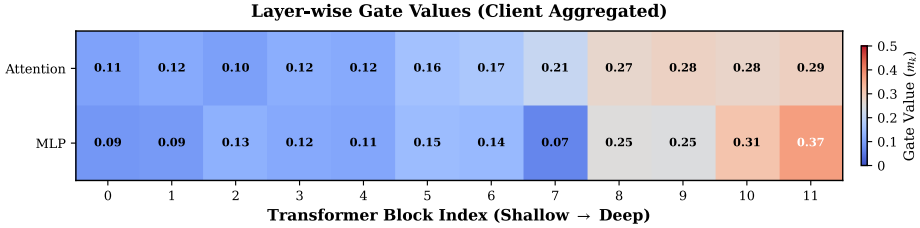
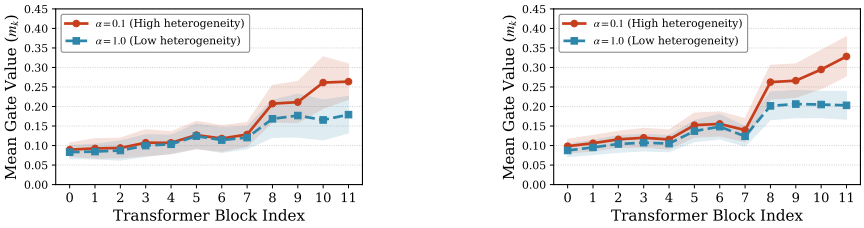


Fig. 5: Layer-wise gate patterns. Mean block gates for Attention (top) and MLP (bottom), aggregated across clients.

Layer-wise Personalization Patterns. Figure 5 visualizes the learned block-wise gates across transformer layers. Personalization concentrates in deeper blocks while early layers remain largely shared, suggesting that higher-level semantics benefit more from client-specific corrections. MLP gates also activate more strongly than attention gates in later blocks, indicating that feed-forward sublayers absorb most client-specific adaptation in our setting. This observation aligns with mechanistic analyses showing that Transformer FFNs store semantic patterns and play a key role in token prediction [12, 13].

Heterogeneity-aware Gating. Figure 6 compares layer-wise gates under different heterogeneity levels across clients. As label skew increases (smaller α), stronger yet bounded personalization emerges mainly in deeper blocks, consistent with controlled private corrections rather than replacement of the shared backbone representation.



(a) CIFAR-100 **(b)** Tiny-ImageNet

Fig. 6: Gating under varying heterogeneity. Smaller α leads to stronger personalization in deeper layers.

Table 7: Client-level quantiles. P10/P50/P90 local test accuracy (%).

	Tiny-ImageNet (α=0.1)			**Office-Home** (dom-0.1)		
Method	**P10**	**P50**	**P90**	**P10**	**P50**	**P90**
FedSDG (ours)	**74.65** (1.05)	**80.96** (0.56)	**86.15** (0.48)	**56.64** (2.55)	**77.24** (0.84)	**91.75** (1.33)
FedRep	72.97 (0.88)	79.17 (0.66)	85.56 (0.50)	48.41 (2.48)	71.04 (0.69)	88.00 (1.95)
FedDPA	65.86 (0.52)	73.44 (0.37)	80.74 (0.70)	55.66 (1.08)	75.34 (0.90)	89.27 (0.69)
Ditto	69.13 (1.13)	78.01 (0.15)	84.43 (0.39)	48.98 (0.90)	72.57 (0.92)	89.51 (1.72)
FedAvg	63.68 (0.29)	71.56 (0.25)	78.37 (0.06)	55.12 (0.17)	73.45 (0.33)	87.40 (0.43)

Client-level Robustness. Table 7 reports client-level quantiles following fairness-aware evaluation [30]. FedSDG improves lower-tail performance while maintaining strong median and upper-tail accuracy, indicating reliable personalization across heterogeneous clients.

5 Conclusions and Limitations

This work studies gated residual personalization in federated parameter-efficient fine-tuning and identifies scale non-identifiability in residual mixing. FedSDG addresses this issue with structure-decoupled dual-path LoRA, projection-level scalar gates, and backbone-anchored Dynamic Alignment, stabilizing private residual scale without additional communication. Limitations include the focus on Vision Transformers with LoRA adapters; future work will extend FedSDG to other PEFT methods, cross-modal and federated LLM fine-tuning [7, 32, 44], and privacy analysis under reconstruction and inference attacks [11, 17, 36, 46], potentially combined with secure aggregation and differential privacy [1, 4, 34].

Acknowledgements

This work was supported by the Excellent Young Scientists Fund (Overseas) of the National Natural Science Foundation of China, and in part by the Jilin Provincial Department of Science and Technology (Grant No. 20260102309JC).

References

1. Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS) (2016)
2. Arivazhagan, M.G., Aggarwal, V., Singh, A.K., Choudhary, S.: Federated learning with personalization layers. arXiv preprint arXiv:1912.00818 (2019)
3. Bian, J., Wang, L., Zhang, L., Xu, J.: Fedalt: Federated fine-tuning through adaptive local training with rest-of-world lora. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2026)
4. Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Ramage, D., Segal, A., Seth, K.: Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS) (2017)
5. Collins, L., Hassani, H., Mokhtari, A., Shakkottai, S.: Exploiting shared representations for personalized federated learning. In: Proceedings of the International Conference on Machine Learning (ICML). pp. 2089–2099 (2021)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 248–255 (2009)
7. Dettmers, T., Pagnoni, A., Holtzman, A., Zettlemoyer, L.: QLoRA: Efficient finetuning of quantized LLMs. In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) (2023)
8. Dinh, C.T., Tran, N.H., Nguyen, T.D.: Personalized federated learning with moreau envelopes. In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS). pp. 21394–21405 (2020)
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: Proceedings of the International Conference on Learning Representations (ICLR) (2021)
10. Fallah, A., Mokhtari, A., Ozdaglar, A.: Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS). pp. 3557–3568 (2020)
11. Geiping, J., Bauermeister, H., Dröge, H., Moeller, M.: Inverting gradients – how easy is it to break privacy in federated learning? In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) (2020)
12. Geva, M., Caciularu, A., Wang, K., Goldberg, Y.: Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP) (2022)
13. Geva, M., Schuster, R., Berant, J., Levy, O.: Transformer feed-forward layers are key-value memories. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP) (2021)
14. Gulrajani, I., Lopez-Paz, D.: In search of lost domain generalization. In: Proceedings of the International Conference on Learning Representations (ICLR) (2021), introduces DomainBed.
15. Guo, P., Zeng, S., Wang, Y., Fan, H., Wang, F., Qu, L.: Selective aggregation for low-rank adaptation in federated learning. In: Proceedings of the International Conference on Learning Representations (ICLR) (2025)

16. Hendrycks, D., Basart, S., Mu, N., et al.: The many faces of robustness: A critical analysis of out-of-distribution generalization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
17. Hitaj, B., Ateniese, G., Perez-Cruz, F.: Deep models under the GAN: Information leakage from collaborative deep learning. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS) (2017)
18. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for NLP. In: Proceedings of the International Conference on Machine Learning (ICML). pp. 2790–2799 (2019)
19. Hsu, T.M.H., Qi, H., Brown, M.: Measuring the effects of non-identical data distribution for federated visual classification. arXiv preprint arXiv:1909.06335 (2019)
20. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. In: Proceedings of the International Conference on Learning Representations (ICLR) (2022)
21. Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K.A., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R.G.L., Eichner, H., Rouayheb, S.E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P.B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konečný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Qi, H., Ramage, D., Raskar, R., Raykova, M., Song, D., Song, W., Stich, S.U., Sun, Z., Suresh, A.T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F.X., Yu, H., Zhao, S.: Advances and open problems in federated learning. Foundations and trends in machine learning **14**(1-2), 1–210 (2021)
22. Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A.T.: SCAFFOLD: Stochastic controlled averaging for federated learning. In: Proceedings of the International Conference on Machine Learning (ICML). pp. 5132–5143 (2020)
23. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of the International Conference on Learning Representations (ICLR) (2015)
24. Koh, P.W., Sagawa, S., Marklund, H., Xie, S.M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R.L., Gao, I., Lee, T., David, E., Stavness, I., Guo, W., Earnshaw, B., Haque, I., Beery, S.M., Leskovec, J., Kundaje, A., Pierson, E., Levine, S., Finn, C., Liang, P.: WILDS: A benchmark of in-the-wild distribution shifts. In: Proceedings of the International Conference on Machine Learning (ICML) (2021)
25. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Tech. rep., University of Toronto (2009)
26. Le, Y., Yang, X.: Tiny ImageNet visual recognition challenge. Stanford CS231N Course Project (2015), benchmark dataset commonly used as a 200-class subset derived from ImageNet
27. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 3045–3059 (2021)
28. Li, T., Hu, S., Beirami, A., Smith, V.: Ditto: Fair and robust federated learning through personalization. In: Proceedings of the International Conference on Machine Learning (ICML). pp. 6357–6368 (2021)
29. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. In: Proceedings of Machine Learning and Systems. pp. 429–450 (2020)

30. Li, T., Sanjabi, M., Beirami, A., Smith, V.: Fair resource allocation in federated learning. In: Proceedings of the International Conference on Learning Representations (ICLR) (2020)
31. Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL). pp. 4582–4597 (2021)
32. Lin, B.Y., He, C., Zeng, X., Wang, M.Z., Soltanolkotabi, M., Ren, X.: Fednlp: Benchmarking federated learning methods for natural language processing tasks. In: Findings of the North American Chapter of the Association for Computational Linguistics (Findings of NAACL) (2022)
33. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., Agüera y Arcas, B.: Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 54, pp. 1273–1282. PMLR (2017)
34. McMahan, H.B., Ramage, D., Talwar, K., Zhang, L.: Learning differentially private recurrent language models. In: Proceedings of the International Conference on Learning Representations (ICLR) (2018)
35. Nam, H.W., Moon, Y.B., Oh, T.H.: Fedpara: Low-rank hadamard product for communication-efficient federated learning. In: Proceedings of the International Conference on Learning Representations (ICLR) (2022)
36. Nasr, M., Shokri, R., Houmansadr, A.: Comprehensive privacy analysis of deep learning: Stand-alone and federated learning under passive and active white-box inference attacks. In: Proceedings of the IEEE Symposium on Security and Privacy (S&P) (2019)
37. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1406–1415 (2019)
38. Taori, R., Dave, A., Shankar, V., Carlini, N., Recht, B., Schmidt, L.: Measuring robustness to natural distribution shifts in image classification. In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) (2020)
39. Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5018–5027 (2017)
40. Yan, G., Li, J., Du, W.: Fedstep: Asynchronous and staleness-aware personalization for efficient federated learning. In: Proceedings of the 34th ACM International Conference on Information and Knowledge Management. pp. 3740–3750 (2025)
41. Yan, G., Wang, H., Yuan, X., Li, J.: Enhancing model poisoning attacks to byzantine-robust federated learning via critical learning periods. In: Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses. pp. 496–512 (2024)
42. Yan, G., Wang, H., Yuan, X., Li, J.: Fedrola: Robust federated learning against model poisoning via layer-based aggregation. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 3667–3678 (2024)
43. Yang, Y., Long, G., Shen, T., Jiang, J., Blumenstein, M.: Dual-personalizing adapter for federated foundation models. In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) (2024)

44. Zhang, Q., Chen, M., Bukharin, A., He, P., Cheng, Y., Chen, W., Zhao, T.: Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. In: Proceedings of the International Conference on Learning Representations (ICLR) (2023)
45. Zhang, Z., Yang, Y., Dai, Y., Wang, Q., Yu, Y., Qu, L., Xu, Z.: Fedpetuning: When federated learning meets the parameter-efficient tuning methods of pre-trained language models. In: Findings of the Annual Meeting of the Association for Computational Linguistics (ACL). pp. 9963–9977 (2023)
46. Zhu, L., Liu, Z., Han, S.: Deep leakage from gradients. In: Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) (2019)