FoRA: Forward-Only Residual Adaptation for Federated Fine-Tuning of Large Language Models

Anonymous authors
Paper under double-blind review

Abstract

Federated fine-tuning of large language models (LLMs) faces a tension between avoiding client-side backpropagation through the LLM backbone and retaining first-order optimization for effective adaptation. Existing methods either keep trainable modules inside the backbone, preserving first-order gradients at the cost of backbone backpropagation, or adopt zeroth-order optimization for forward-only execution at the cost of gradient-estimation error. We propose FoRA, a federated fine-tuning framework that keeps the LLM backbone forward-only while preserving first-order optimization for the trainable adaptation. FoRA detaches compact multi-level representations from a frozen backbone, learns a lightweight factorized residual with a direct *Split* branch and a globally weighted *Bridge* branch, and regularizes local training against a fixed global reference in function space. Our analysis verifies the execution-graph memory separation, characterizes the local Jacobian geometry of the auxiliary parameterization, characterizes the local dynamics induced by global-reference regularization under a fixed-tangent model, and establishes non-convex convergence for the federated protocol. Empirically, FoRA reduces peak allocated client GPU memory by 27.3%-59.2% over first-order baselines while achieving comparable quality, and outperforms FedMeZO by 6.99-18.37 points in Final-10 Accuracy on classification and 4.74 points in Final-10 ROUGE-L on generation, with 65.8%-74.7% less training time.

1 Introduction

Federated learning (FL) enables multiple clients to collaboratively adapt a shared model without centralizing raw data (McMahan et al., 2017; Konečnỳ et al., 2016), and is increasingly relevant to large language models (LLMs) whose downstream adaptation relies on private, distributed data (Ye et al., 2024; Xu et al., 2024). However, the scale of modern LLMs makes client-side training prohibitively expensive. Even with parameter-efficient fine-tuning (PEFT) such as LoRA (Hu et al., 2022), trainable modules remain inside the Transformer backbone, so backpropagation still requires intermediate activations and training states (Malladi et al., 2023). Zeroth-order (ZO) methods remove backbone backpropagation and enable forward-only training (Ghadimi & Lan, 2013; Malladi et al., 2023; Xu et al., 2024), but estimate updates from function evaluations, introducing smoothing bias, directional variance, and additional forward queries. Existing federated LLM fine-tuning therefore forces clients to choose between first-order optimization with backbone backpropagation and forward-only execution without first-order gradients.

This paper studies whether federated LLM fine-tuning can keep the LLM backbone forward-only while preserving first-order optimization for the trainable adaptation. Although side-network adaptation provides a natural architectural starting point, recent federated realizations often rely on server-side optimization, activation communication, or auxiliary proxy models. Therefore, we propose FoRA, a federated fine-tuning framework that places all trainable adaptation outside the frozen backbone. Each client extracts detached multi-level representations at supervised prediction sites and optimizes a compact Split-Bridge residual with first-order gradients. The broadcast auxiliary state initializes local training, defines a fixed function-space reference, and is aggregated by sample-weighted FedAvg. Thus, this design bridges the gap between PEFT, which pays the backbone backward cost, and ZO tuning, which removes that cost but loses first-order optimization.

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.

The key methodological contribution of FoRA is the co-design of a client-resident federated residual state, its factorized Split-Bridge optimization coordinates, and a function-space global reference that reuses the same broadcast state. *Backbone-detached representation extraction* collects and pools hidden states from a fixed set of Transformer layers, detaches them, and exposes a compact multi-level representation set, keeping the client-side backward graph entirely outside the backbone. A *multi-level functional residual bridge* maps these features into a compact correction space. The *Split* branch provides a direct multi-level correction, while the *Bridge* branch contains multiple independently parameterized residual banks combined by a shared softmax coefficient vector. Finally, *global-reference federated optimization* uses the same broadcast residual to initialize local training and define a fixed functional reference within the round. Clients therefore perform first-order optimization locally, while the server only aggregates auxiliary parameter states.

Our analysis verifies the execution-graph consequences of backbone detachment and characterizes the local geometry induced by the factorized Split-Bridge coordinates. We then study the global-reference mechanism, which induces a Jacobian-weighted local metric that controls functional displacement under a fixed-tangent approximation and changes the accumulated local-trajectory scaling from quadratic to linear in the number of local steps. Finally, we derive a non-convex stationarity bound for the effective objective induced by the heterogeneous federated protocol, with an explicit local-trajectory term. Across three settings, FoRA reduces peak allocated client GPU memory by 27.3%-59.2% relative to LoRA-based first-order baselines while matching or improving their downstream quality. Compared with FedMeZO, it improves Final-10 classification Accuracy by 6.99-18.37 points and generation ROUGE-L by 4.74 points while reducing training time by 65.8%-74.7%. Additional experiments evaluate client scalability, statistical heterogeneity, hyperparameter sensitivity, and component ablations in the evaluated scenarios.

Our contributions are as follows. First, we propose FoRA, a client-resident federated residual framework that couples detached multi-level representations, factorized Split-Bridge optimization, and global-reference regularization within a single parameter-only FedAvg state. Second, we analyze FoRA at three levels: exact execution-graph and auxiliary gradient properties, a local functional analysis of global-reference regularization, and non-convex federated convergence. Third, we evaluate FoRA against federated LLM fine-tuning baselines in downstream quality, peak allocated client GPU memory, and learning efficiency, together with scalability, sensitivity, and ablation studies.

2 Related Work

Our work lies at the intersection of FL, side-network adaptation, and federated LLM fine-tuning.

2.1 Federated Learning under Heterogeneous Clients

Federated learning trains a shared model from decentralized data through repeated local optimization and server aggregation (McMahan et al., 2017). A central challenge is statistical and system heterogeneity, which causes local models to follow different optimization trajectories. FedProx (Li et al., 2020) constrains local updates around the global model with a proximal term, SCAFFOLD (Karimireddy et al., 2020) corrects client drift with control variates, and FedNova (Wang et al., 2020) normalizes heterogeneous local updates to address objective inconsistency. These methods show that local trajectory behavior is critical for federated convergence under non-IID data and unequal workloads. FoRA follows the standard local-training and aggregation framework, but regularizes local adaptation in function space, while placing the local reference in residual-function space. Under local linearization, this design induces a Jacobian-weighted metric over the auxiliary parameters, matching the functional form of the adaptation.

2.2 Side-Network Adaptation and Federated Execution

Side-network methods adapt frozen backbones through lightweight auxiliary networks. Side-Tuning (Zhang et al., 2020) uses frozen backbone outputs, while Ladder Side-Tuning (LST) (Sung et al., 2022) extends this principle to intermediate representations from multiple depths. Recent LLM systems adopt related ideas with different training and communication structures: MobiLLM (Li et al., 2025a) and Fed MobiLLM (Yang et al., 2025) optimize side networks on the server from communicated representations, while FedProxy (Fan et al., 2026) federates a compressed proxy

model. FoRA therefore defines a distinct federated optimization topology: the trainable residual remains on clients, model-state communication replaces activation communication, and the broadcast residual jointly supports local initialization, functional drift control, and server aggregation.

2.3 Federated LLM Fine-Tuning and PEFT

Federated LLM fine-tuning has largely followed two directions for reducing client-side cost. First-order PEFT methods, from FedIT (Zhang et al., 2023) to recent LoRA variants (Singhal et al., 2025; Raje et al., 2025; Wang et al., 2025; Huang et al., 2026; Meng et al., 2026), reduce trainable and communicated parameters through improved low-rank parameterization, selective updates, and aggregation. These methods retain first-order optimization, but their trainable modules remain inside the LLM backbone, so backpropagation still requires intermediate activations and the associated memory. Forward-only methods take a different route: FwdLLM (Xu et al., 2024), FedMeZO (Ling et al., 2024), and subsequent federated ZO methods (Li et al., 2025b; Ran et al., 2025) avoid backbone backpropagation by estimating update directions, reducing memory at the cost of gradient-estimation error and a large number of forward queries. FoRA addresses this gap by moving trainable adaptation outside the backbone computation graph: the LLM remains forward-only, while a compact functional correction is optimized with first-order gradients and federated across clients, yielding a distinct trade-off from both embedded PEFT and zeroth-order tuning.

3 Background and Problem Setup

3.1 Federated Fine-Tuning

We consider a federated system with a central server and N clients $\{\mathcal{C}_i\}_{i=1}^N$. Client $\mathcal{C}_i$ holds a local dataset $\mathcal{D}_i$ with $n_i = |\mathcal{D}_i|$ examples. The local data distributions and participation patterns may differ across clients. Let $\boldsymbol{\phi}$ denote the parameters of a shared pre-trained LLM backbone, and let $\mathbf{w}$ denote the trainable adaptation parameters. Throughout federated fine-tuning, the backbone parameters $\boldsymbol{\phi}$ remain fixed. For a sample ζ drawn from client i, we denote the training loss by $f_i(\mathbf{w};\zeta)$. The local and global objectives are

$$F_i(\mathbf{w}) = \mathbb{E}_{\zeta\sim\mathcal{D}_i}\left[f_i(\mathbf{w};\zeta)\right], \qquad F(\mathbf{w}) = \sum_{i=1}^{N} p_i F_i(\mathbf{w}), \qquad p_i = \frac{n_i}{\sum_{j=1}^{N} n_j}. \tag{1}$$

At communication round t, the server samples a subset of clients $\mathcal{S}_t$ and broadcasts the current global parameters $\mathbf{w}^t$. Each selected client initializes from $\mathbf{w}^t$ and performs local optimization on its own data. We denote the resulting local model by $\mathbf{w}_i^{t,H_i}$, where H_i is the number of local update steps performed by client i. With E local epochs and mini-batch size B, the number of updates is approximately $H_i = E\left\lceil \frac{n_i}{B} \right\rceil$. After local training, the server performs sample-weighted aggregation following the standard FedAvg protocol (McMahan et al., 2017).

3.2 Problem Formulation

First-order federated PEFT reduces the number of trainable parameters, but trainable modules embedded in the LLM still require intermediate backbone activations for backpropagation, leaving substantial client-side memory overhead (Hu et al., 2022; Zhang et al., 2023). Zeroth-order methods remove this backward graph by estimating updates from forward evaluations, at the cost of gradient-estimation error and additional model queries (Malladi et al., 2023; Xu et al., 2024). We study the regime in which a client cannot efficiently sustain backbone backpropagation. The goal is to retain first-order local optimization while making the LLM backbone forward-only, and to support stable federated training under heterogeneous client data.

Formally, let $\mathcal{P}(x)$ denote the supervised prediction sites for input x. At each $p \in \mathcal{P}(x)$, the frozen backbone provides base logits $\mathbf{z}_{0,p}(x) \in \mathbb{R}^C$ and detached multi-level representations $\mathcal{R}_p(x)$. FoRA produces a correction $\mathbf{u}_{\mathbf{w}}(\mathcal{R}_p(x)) \in \mathbb{R}^{d_c}$, mapped to the output space by:

$$\mathbf{z}_{\mathbf{w},p}(x) = \mathbf{z}_{0,p}(x) + \boldsymbol{\delta}_{\mathbf{w},p}(x), \qquad \boldsymbol{\delta}_{\mathbf{w},p}(x) = \mathbf{A}\,\mathbf{u}_{\mathbf{w}}(\mathcal{R}_p(x)), \tag{2}$$

where $\mathbf{A} \in \mathbb{R}^{C\times d_c}$ is a fixed readout. For sequence classification, $\mathcal{P}(x)$ contains one site with $d_c = C$ and $\mathbf{A} = \mathbf{I}_C$; for autoregressive generation, it contains the supervised response positions with $d_c = d_h$ and $\mathbf{A}$ given by the frozen language-model head. Only $\mathbf{w}$ is trained and federated.

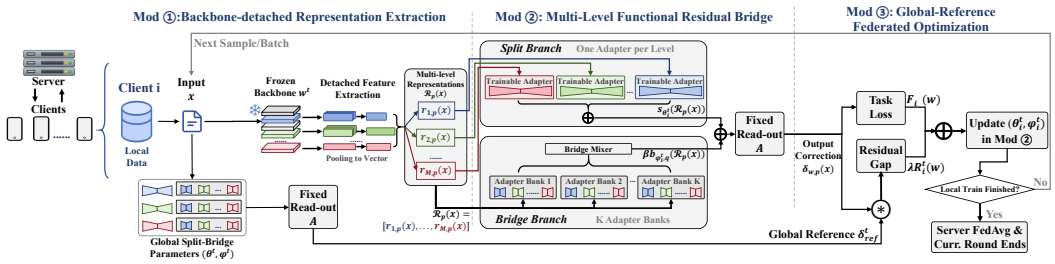

Figure 1: Overview of FoRA with level-wise color correspondence at global round t. Red, green, and blue trace distinct representation levels through *Split* and *Bridge* branches in Client $\mathcal{C}_i$.

This formulation motivates three design questions: (i) how to extract compact and informative representations from a forward-only backbone, (ii) how to parameterize a lightweight residual for local optimization and communication, and (iii) how to stabilize this state under heterogeneous data. These questions motivate the three components of FoRA: backbone-detached representation extraction, a multi-level functional residual bridge, and global-reference federated optimization.

4 Design of FoRA

4.1 Overview

FoRA solves the three design questions in Section 3.2 by making the auxiliary residual itself the federated model state and co-designing its local parameterization and function-space drift control under a forward-only LLM backbone. Specifically, detached multi-level representations define the trainable boundary outside the backbone; the Split-Bridge structure parameterizes w for local first-order optimization; and the same broadcast state supplies the functional reference used to control within-round drift before FedAvg aggregation. Sequence classification and autoregressive generation share this client-server protocol and differ only in their prediction sites and fixed readout. Fig. 1 illustrates the overall pipeline, and Algorithm 1 in Appendix B.2 summarizes the full procedure.

4.2 Backbone-Detached Representation Extraction

The first component of FoRA exploits pretrained LLM representations without backbone backpropagation on resource-constrained clients. Side-network methods have shown that frozen backbone representations can support task adaptation (Zhang et al., 2020; Sung et al., 2022). Recent federated variants typically introduce additional communication, server-side optimization, or auxiliary-data dependencies (Yang et al., 2025; Fan et al., 2026). FoRA instead places the trainable boundary on each client by extracting compact multi-level representations from a frozen, forward-only LLM and detaching them before adaptation. The backbone therefore remains outside the backward graph, while the auxiliary model is optimized with first-order gradients.

Specifically, let $f_0(\cdot;\phi)$ denote the frozen LLM with L Transformer layers. With a fixed layer interval b, we retain $\mathcal{L} = \{\ell_j\}_{j=1}^{M}$, where $M = \lceil L/b \rceil$ and $\ell_j = \min(jb, L)$; the same layer set is used across clients and communication rounds. For input x, the backbone produces hidden states $\mathbf{H}_{\ell_j}(x) \in \mathbb{R}^{S\times d_h}$ and frozen logits $\mathbf{z}_{0,p}(x)$ at each supervised prediction site $p \in \mathcal{P}(x)$. Let $\rho_p : \mathbb{R}^{S\times d_h} \to \mathbb{R}^{d_h}$ extract the representation associated with site p. We define

$$\mathbf{r}_{j,p}(x) = \mathrm{stopgrad}\big(\rho_p(\mathbf{H}_{\ell_j}(x))\big), \qquad \mathcal{R}_p(x) = \{\mathbf{r}_{j,p}(x)\}_{j=1}^{M}. \tag{3}$$

Here, $\mathrm{stopgrad}(\mathbf{u}) = \mathbf{u}$ in the forward pass and has zero derivative with respect to $\mathbf{u}$ in the backward pass. For sequence classification, $\mathcal{P}(x)$ contains one site; for autoregressive generation, it contains the supervised response positions. The backbone is evaluated with gradient tracking disabled, so $\mathbf{z}_{0,p}(x)$ and $\mathcal{R}_p(x)$ are fixed inputs during local optimization.

The retained interface remains compact: for classification, storing M representations for a mini-batch of size B requires $\mathcal{O}(BMd_h)$ additional memory; for generation, the interface scales additionally with the number of supervised response positions. In both cases, the retained tensors carry

no backbone gradient history. The shared layer set $\mathcal{L}$ provides a common multi-level representation interface for the subsequent Split-Bridge residual.

4.3 Multi-Level Functional Residual Bridge

With the backbone detached, task adaptation is carried by the auxiliary residual, which must remain compact enough for repeated local training and communication. FoRA parameterizes this residual over the detached representation set $\mathcal{R}_p(x)$ using one direct *Split* branch and a *Bridge* branch with K independently parameterized residual banks. An input-independent coefficient vector mixes the *Bridge* banks, increasing auxiliary capacity without changing backbone execution.

Both branches use the same bottleneck adapter in the correction space. For $\mathbf{r} \in \mathbb{R}^{d_h}$,

$$\mathbf{a}_{\boldsymbol{\omega}}(\mathbf{r}) = \mathbf{W}_2\,\mathrm{GELU}(\mathbf{W}_1\mathrm{LN}(\mathbf{r}) + \mathbf{b}_1) + \mathbf{b}_2, \tag{4}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_b \times d_h}$ and $\mathbf{W}_2 \in \mathbb{R}^{d_c \times d_b}$ define a bottleneck of width d_b. At prediction site p, the *Split* branch uses one adapter per representation level, while each *Bridge* bank contains its own set of M adapters:

$$\mathbf{s}_{\boldsymbol{\theta}}(\mathcal{R}_p(x)) = \sum_{j=1}^{M} \mathbf{a}_{\boldsymbol{\theta}_j}(\mathbf{r}_{j,p}(x)), \quad \mathbf{b}_k(\mathcal{R}_p(x)) = \sum_{j=1}^{M} \mathbf{a}_{\boldsymbol{\psi}_{k,j}}(\mathbf{r}_{j,p}(x)), \quad k = 1, \ldots, K. \tag{5}$$

Here, $\boldsymbol{\theta} = \{\boldsymbol{\theta}_j\}_{j=1}^{M}$ and $\boldsymbol{\psi} = \{\boldsymbol{\psi}_{k,j}\}_{k=1,j=1}^{K,M}$. A trainable vector $\mathbf{q} \in \mathbb{R}^K$, shared across inputs, defines $\boldsymbol{\pi}(\mathbf{q}) = \mathrm{softmax}(\mathbf{q})$ and $\mathbf{b}_{\boldsymbol{\psi},\mathbf{q}}(\mathcal{R}_p(x)) = \sum_{k=1}^{K} \pi_k(\mathbf{q})\mathbf{b}_k(\mathcal{R}_p(x))$. The number of *Bridge* banks K controls auxiliary capacity and computation while leaving backbone execution unchanged. The resulting correction-space residual is

$$\mathbf{u}_{\mathbf{w}}(\mathcal{R}_p(x)) = \mathbf{s}_{\boldsymbol{\theta}}(\mathcal{R}_p(x)) + \beta\,\mathbf{b}_{\boldsymbol{\psi},\mathbf{q}}(\mathcal{R}_p(x)), \tag{6}$$

where $\mathbf{w} = (\boldsymbol{\theta}, \boldsymbol{\psi}, \mathbf{q})$ and $\beta \geq 0$ scales the *Bridge* contribution. The fixed readout in Eq. 2 maps this correction to the task-output residual. We initialize $\mathbf{u}_{\mathbf{w}}$ at zero, so the adapted prediction initially matches the frozen backbone. Since $\mathcal{R}_p(x)$ and the readout $\mathbf{A}$ are fixed, gradients are computed only with respect to $\mathbf{w}$.

At any fixed parameter state, the same correction-space residual admits a folded realization as one wider adapter per representation level: the hidden units of the *Split* adapter and the *Bridge*-bank adapters can be concatenated, with each *Bridge* output block scaled by $\beta\pi_k(\mathbf{q})$. FoRA retains the factorized coordinates $(\boldsymbol{\theta}, \boldsymbol{\psi}, \mathbf{q})$ during local optimization and federated aggregation. Since the folding map contains products between the mixture weights and *Bridge* output projections, it is nonlinear in these trainable coordinates; local gradient updates and parameter averaging therefore generally follow different trajectories from direct optimization and aggregation of the folded adapter. Appendix C.4 gives the folded representation and the corresponding block-structured Jacobian.

4.4 Global-Reference Federated Optimization

Multi-step local optimization on heterogeneous data can drive clients toward different local solutions (Li et al., 2020; Karimireddy et al., 2020). Since FoRA adapts the frozen backbone through a functional residual, we regularize local updates directly in the residual space using the broadcast global model as a fixed reference.

At round t, the server broadcasts $\mathbf{w}^t = (\boldsymbol{\theta}^t, \boldsymbol{\psi}^t, \mathbf{q}^t)$, and each selected client $\mathcal{C}_i \in \mathcal{S}_t$ initializes $\mathbf{w}_i^{t,0} = \mathbf{w}^t$. For each supervised prediction site $p \in \mathcal{P}(x)$, define $\boldsymbol{\delta}_{\mathrm{ref},p}^t(x) = \mathrm{stopgrad}(\boldsymbol{\delta}_{\mathbf{w}^t,p}(x))$. Client i uses the functional reference penalty

$$R_i^t(\mathbf{w}) = \frac{1}{2}\mathbb{E}_{x\sim\mathcal{D}_i}\left[\frac{1}{|\mathcal{P}(x)|}\sum_{p\in\mathcal{P}(x)} \left\|\boldsymbol{\delta}_{\mathbf{w},p}^t(x) - \boldsymbol{\delta}_{\mathrm{ref},p}(x)\right\|_2^2\right]. \tag{7}$$

The expectation in Eq. 7 is over training examples, with the residual distance averaged over the supervised prediction sites of each example. The local objective is

$$\widetilde{F}_i^t(\mathbf{w}) = F_i(\mathbf{w}) + \lambda R_i^t(\mathbf{w}), \tag{8}$$

where $\lambda \geq 0$ controls the regularization strength. Because $\mathbf{w}_i^{t,0} = \mathbf{w}^t$, both $R_i^t(\mathbf{w}^t) = 0$ and $\nabla_{\mathbf{w}} R_i^t(\mathbf{w}^t) = \mathbf{0}$. Thus, the first local step follows the task gradient, while the reference term acts as the local residual moves away from the broadcast model. The same auxiliary state is used for initialization and reference construction.

Local training uses a standard first-order optimizer. Writing one stochastic gradient step in SGD form,

$$\mathbf{w}_i^{t,h+1} = \mathbf{w}_i^{t,h} - \eta_t^{\mathrm{loc}} \widehat{\nabla} \widetilde{F}_i^t \left(\mathbf{w}_i^{t,h}; \xi_{i,h}^t \right), \qquad h = 0, \ldots, H_i - 1. \tag{9}$$

Only the auxiliary parameters are differentiated. After H_i local steps, clients upload their auxiliary states, and the server performs sample-weighted FedAvg:

$$\mathbf{w}^{t+1} = \sum_{i \in \mathcal{S}_t} \omega_i^t \mathbf{w}_i^{t,H_i}, \qquad \omega_i^t = \frac{n_i}{\sum_{j \in \mathcal{S}_t} n_j}. \tag{10}$$

The aggregated state $\mathbf{w}^{t+1}$ defines the reference for the next round.

5 Theoretical Analysis

Our analysis proceeds in three parts: exact structural properties of the execution graph and Split-Bridge parameterization, a fixed-tangent analysis of global-reference regularization, and a non-convex stationarity guarantee for the federated protocol. This organization separates structural identities, local mechanism analysis, and protocol-level convergence, while linking reference-induced drift control to the trajectory-error term in the global bound. We provide the formal assumptions and proofs in Appendix C.

Execution-graph property. For a mini-batch ξ, let $P_\xi = \sum_{x \in \xi} |\mathcal{P}(x)|$ denote the number of supervised prediction sites. Let $M_{\mathrm{fwd}}(B, S)$ be the peak memory of a forward-only backbone pass, $M_{\mathrm{aux}}^{\mathrm{train}}(\xi)$ the memory required to differentiate the auxiliary residual path with respect to w, and $M_{\mathrm{ref}}(\xi)$ the memory associated with the fixed round reference. Then

$$M_{\mathrm{FoRA}}(\xi) \leq M_{\mathrm{fwd}}(B, S) + \mathcal{O}(P_\xi M d_h) + M_{\mathrm{aux}}^{\mathrm{train}}(\xi) + M_{\mathrm{ref}}(\xi). \tag{11}$$

For sequence classification, $P_\xi = |\xi| \leq B$, so the retained-feature term is $\mathcal{O}(|\xi| M d_h) \subseteq \mathcal{O}(B M d_h)$; for generation, P_ξ is the number of supervised response positions in the mini-batch.

Remark. Eq. 11 shows that backbone detachment removes the backbone backward-tape term because all representations passed to the auxiliary path are detached. Client training memory is determined by the forward-only backbone pass, detached interface, auxiliary state, and reference state.

Factorized-coordinate geometry. We compare the full FoRA parameterization with its Split-only restriction. On a finite analysis set, let $\mathbf{J}_{\mathrm{s}}$ and $\mathbf{J}_{\mathrm{b}}$ denote the Jacobians of the *Split* and *Bridge* outputs. The complete residual then satisfies

$$\mathbf{J}_{\mathrm{FoRA}} = [\mathbf{J}_{\mathrm{s}}, \beta \mathbf{J}_{\mathrm{b}}], \qquad \mathbf{K}_{\mathrm{FoRA}} = \mathbf{K}_{\mathrm{s}} + \beta^2 \mathbf{K}_{\mathrm{b}} \succeq \mathbf{K}_{\mathrm{s}}, \tag{12}$$

where $\mathbf{K} = \mathbf{J}\mathbf{J}^\top$ denotes the corresponding functional kernel.

Remark. Eq. 12 shows that the *Bridge* branch adds a positive-semidefinite kernel contribution to the *Split* branch; the local tangent space expands when the added *Bridge* directions are nonredundant.

Global-reference local dynamics. Because each client starts from the broadcast model, $R_i^t(\mathbf{w}^t) = 0$ and $\nabla_{\mathbf{w}} R_i^t(\mathbf{w}^t) = \mathbf{0}$, so the round-start gradient is the task gradient. Under the fixed-tangent surrogate $\boldsymbol{\delta}_{\mathbf{w}} - \boldsymbol{\delta}_{\mathbf{w}^t} = \mathbf{J}_i(\mathbf{w} - \mathbf{w}^t)$, the reference induces the Jacobian-weighted quadratic form $R_{i,\tan}^t(\mathbf{w}) = \frac{1}{2} \|\mathbf{J}_i(\mathbf{w} - \mathbf{w}^t)\|_2^2$. Writing $\mathbf{K}_i = \mathbf{J}_i \mathbf{J}_i^\top$ and $\mathbf{v}_{i,h} = \boldsymbol{\delta}_{\mathbf{w}_i^{t,h}} - \boldsymbol{\delta}_{\mathbf{w}^t}$, the functional dynamics are

$$\mathbf{v}_{i,h+1} = (\mathbf{I} - \eta\lambda \mathbf{K}_i)\mathbf{v}_{i,h} - \eta \mathbf{K}_i \mathbf{g}_{i,h}, \qquad \mathbf{v}_{i,0} = \mathbf{0}, \tag{13}$$

where $\|\mathbf{g}_{i,h}\|_2 \leq G_i$. Thus, the global reference introduces a Jacobian-dependent restoring term in residual space. For $\mathbf{K}_i \neq \mathbf{0}$, let $\kappa_i^+ = \lambda_{\max}(\mathbf{K}_i)$ and $\kappa_i^- = \lambda_{\min}^+(\mathbf{K}_i)$. If $0 < \eta\lambda\kappa_i^+ \leq 1$ for $\lambda > 0$, then

$$\|\mathbf{v}_{i,h}\|_2 \leq \begin{cases} \dfrac{\kappa_i^+ G_i}{\lambda \kappa_i^-} \left[1 - (1 - \eta\lambda\kappa_i^-)^h \right], & \lambda > 0, \\ \eta h \kappa_i^+ G_i, & \lambda = 0. \end{cases} \tag{14}$$

For fixed $\lambda > 0$, the reference-induced contraction keeps $\|\mathbf{v}_{i,h}\|_2$ uniformly bounded with respect to h, whereas the unregularized bound grows linearly.

Remark. Under the fixed-tangent model, the global reference bounds local functional drift; with locally smooth task gradients, the accumulated local-gradient residual scales as $O(H_i)$ for $\lambda > 0$, compared with $O(H_i^2)$ for $\lambda = 0$.

Federated protocol guarantee. Partial participation and client-specific local update counts induce effective client weights in the expected server update. Let $\widetilde{F}$ denote the corresponding effective federated objective, $\mathcal{B}^2_{\mathrm{traj}}(\lambda)$ bound the aggregated local residual, ζ^2 measure client heterogeneity, V_{rr} bound local sampling noise, $H_{\max} = \max_i H_i$, and $C_H = \mathbb{E}_{\mathcal{S}}[A_{\mathcal{S}}]$ with $A_{\mathcal{S}} = \sum_{i\in\mathcal{S}} \omega_i(\mathcal{S}) H_i$. For $0 < \eta \le C_H/[6L_{\widetilde{F}}(C_H^2 + 4\nu_H^2)]$,

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left\|\nabla \widetilde{F}(\mathbf{w}^t)\right\|_2^2 \le \mathcal{O}\left(\frac{\Delta_{\widetilde{F}}}{\eta T} + \mathcal{B}^2_{\mathrm{traj}}(\lambda) + \eta H_{\max}\zeta^2 + \eta V_{\mathrm{rr}}\right), \tag{15}$$

where $\Delta_{\widetilde{F}} = \widetilde{F}(\mathbf{w}^0) - \widetilde{F}^\star$, $C_H = \mathbb{E}_{\mathcal{S}}[A_{\mathcal{S}}]$, and $\nu_H^2 = \mathbb{E}_{\mathcal{S}}[(A_{\mathcal{S}} - C_H)^2]$.

Remark. The bound separates optimization progress from local trajectory error, client heterogeneity, and stochastic sampling, with $\mathcal{B}^2_{\mathrm{traj}}(\lambda)$ carrying the effect of global-reference regularization.

6 Evaluation

We evaluate FoRA across performance, scalability, hyperparameter sensitivity, and ablation studies. We implement FoRA using PyTorch on a workstation with an Intel Gold 6278C CPU and two Tesla V100-PCIE-32GB GPUs. Implementation details and additional results are in Appendix D.

6.1 Experimental Setup

Models and metrics. We evaluate OPT-350M (Zhang et al., 2022) on MRPC (Wang et al., 2019), OPT-2.7B (Zhang et al., 2022) on SST-2 (Wang et al., 2019), and Llama-3.1-8B (Team, 2024) on Dolly-15K (Conover et al., 2023). Classification reports Best and Final-10 Accuracy/F1; generation reports Best and Final-10 perplexity (PPL)/ROUGE-L. We additionally report federated training time and peak allocated client GPU memory.

Federated protocol. Unless otherwise stated, we use $N = 10$ clients, 50% participation, group-wise Dirichlet partitioning with $\alpha_{\mathrm{Dir}} = 0.7$, mini-batch size $B = 4$, and $E = 2$ local epochs. MRPC uses 200 global rounds, while SST-2 and Dolly-15K use 50; the Llama setting applies a common $0.1\times$ learning-rate scaling. All methods within a model-task setting share the same data split, client partition, participation schedule, and hardware allocation.

Methods. We compare FoRA with FedIT (Zhang et al., 2023), FedMeZO (Ling et al., 2024), GMFL (Huang et al., 2026), RAVAN (Raje et al., 2025), and FedProxy (Fan et al., 2026). Fed MobiLLM follows a server-assisted asynchronous optimization and activation-communication protocol, and is therefore evaluated separately under its native protocol (Appendix D.5.1). LoRA-based methods use a common base configuration with rank $r = 16$, scaling factor $\alpha = 32$, and matched target modules where applicable.

6.2 Comparison with Federated Fine-Tuning Baselines

Model quality and stability. We first examine whether FoRA can retain the task quality of first-order baselines while keeping the LLM backbone forward-only. Table 1 reports the results across all three model-task configurations. Across all three model-task configurations, FoRA matches or improves the terminal performance of FedIT and GMFL, and substantially outperforms FedMeZO, with 6.99-18.37 points higher Final-10 Accuracy on classification and 4.74 points higher ROUGE-L on Llama-3.1-8B. RAVAN requires less training time but consistently attains lower task quality, whereas FedProxy uses a compressed-proxy design yet exhibits larger terminal gaps, ranging from 2.44 to 20.11 points in Final-10 Accuracy on classification and 3.32 ROUGE-L points on generation. These results indicate that FoRA preserves strong terminal task quality across both classification and generation settings.

Table 1: Performance and client-resource results across classification and generation tasks under the common round-based federated evaluation protocol. Reported results are the mean ± standard deviation. Final-10 denotes the mean metric over the last ten rounds. For generation, R-L denotes ROUGE-L. Peak Memory reports the maximum allocated client GPU memory. Best results in each metric are in bold, and worst results are underlined.

Scenario: OPT-350M@MRPC

Method	Backbone Backpropagation	Best Acc. (%) ↑	Best F1 (%) ↑	Final-10 Acc. (%) ↑	Final-10 F1 (%) ↑	Training Time (h) ↓	Peak Memory (GiB) ↓
FedIT (LoRA)	✓	74.01 ± 0.30	83.33 ± 0.15	73.71 ± 0.23	83.10 ± 0.18	12.14 ± 0.58	3.180 ± 0.0
FedMeZO	×	69.39 ± 0.14	80.85 ± 0.13	68.69 ± 0.60	80.16 ± 0.60	32.09 ± 0.32	1.480 ± 0.0
RAVAN	✓	71.20 ± 0.14	82.30 ± 0.12	70.87 ± 0.26	82.15 ± 0.12	$\mathbf{6.32 \pm 0.49}$	3.200 ± 0.0
GMFL	✓	72.52 ± 0.12	82.55 ± 0.02	72.28 ± 0.23	82.48 ± 0.06	11.17 ± 0.59	3.180 ± 0.0
FedProxy	×	55.78 ± 0.14	69.05 ± 0.13	55.57 ± 0.18	68.86 ± 0.16	10.30 ± 0.53	2.300 ± 0.0
FoRA	×	$\mathbf{76.11 \pm 0.21}$	$\mathbf{83.99 \pm 0.11}$	$\mathbf{75.68 \pm 0.35}$	$\mathbf{83.80 \pm 0.15}$	8.59 ± 0.46	$\mathbf{1.306 \pm 0.0}$

Scenario: OPT-2.7B@SST-2

Method	Backbone Backpropagation	Best Acc. (%) ↑	Best F1 (%) ↑	Final-10 Acc. (%) ↑	Final-10 F1 (%) ↑	Training Time (h) ↓	Peak Memory (GiB) ↓
FedIT (LoRA)	✓	96.10 ± 0.13	96.20 ± 0.11	95.89 ± 0.18	96.00 ± 0.16	361.09 ± 12.90	15.920 ± 0.0
FedMeZO	×	79.94 ± 1.56	79.04 ± 1.82	77.63 ± 2.18	75.94 ± 2.93	501.84 ± 12.76	10.770 ± 0.0
RAVAN	✓	94.64 ± 0.07	94.75 ± 0.08	94.50 ± 0.10	94.62 ± 0.10	$\mathbf{76.64 \pm 9.61}$	16.110 ± 0.0
GMFL	✓	95.87 ± 0.07	95.97 ± 0.04	95.72 ± 0.12	95.83 ± 0.11	349.96 ± 17.39	15.920 ± 0.0
FedProxy	×	94.06 ± 0.07	94.27 ± 0.09	93.56 ± 0.39	93.68 ± 0.51	297.81 ± 10.91	11.850 ± 0.0
FoRA	×	$\mathbf{96.22 \pm 0.07}$	$\mathbf{96.30 \pm 0.07}$	$\mathbf{96.00 \pm 0.18}$	$\mathbf{96.10 \pm 0.16}$	126.87 ± 14.36	$\mathbf{10.157 \pm 0.0}$

Scenario: Llama-3.1-8B@Dolly-15K

Method	Backbone Backpropagation	Best PPL ↓	Best R-L (%) ↑	Final-10 PPL ↓	Final-10 R-L (%) ↑	Training Time (h) ↓	Peak Memory (GiB) ↓
FedIT (LoRA)	✓	3.659 ± 0.002	28.19 ± 0.17	3.660 ± 0.005	28.04 ± 0.13	236.26 ± 2.63	57.960 ± 0.0
FedMeZO	×	4.896 ± 0.007	23.93 ± 0.32	4.898 ± 0.002	23.67 ± 0.23	463.66 ± 9.47	44.428 ± 0.0
RAVAN	✓	3.703 ± 0.014	26.40 ± 0.09	3.758 ± 0.004	26.25 ± 0.12	$\mathbf{115.46 \pm 1.97}$	60.826 ± 0.0
GMFL	✓	3.654 ± 0.001	28.28 ± 0.02	3.657 ± 0.005	28.17 ± 0.12	246.80 ± 1.34	57.960 ± 0.0
FedProxy	×	4.277 ± 0.011	25.20 ± 0.04	4.290 ± 0.012	25.09 ± 0.08	183.66 ± 2.49	43.910 ± 0.0
FoRA	×	$\mathbf{3.652 \pm 0.003}$	$\mathbf{28.54 \pm 0.08}$	$\mathbf{3.656 \pm 0.001}$	$\mathbf{28.41 \pm 0.11}$	158.41 ± 1.79	$\mathbf{42.148 \pm 0.0}$

Client GPU memory and training-time trade-off. We then examine whether the quality gains of FoRA are accompanied by a favorable client-side resource trade-off. FoRA reduces peak allocated client GPU memory by 27.3-58.9% relative to FedIT and GMFL, and by 5.1-11.8% relative to FedMeZO, while reducing FedMeZO training time by 65.8-74.7%. Relative to RAVAN, FoRA lowers peak allocated client GPU memory by 30.7-59.2% at higher downstream quality, and it also improves both resource efficiency and task performance over FedProxy. Appendix Table 11 reports peak allocated client GPU memory on OPT-2.7B@SST-2: FoRA uses 10.11-10.19 GiB across maximum input sequence lengths $S \in \{64, 128, 256, 512\}$, compared with 13.45-17.23 GiB for FedIT. Overall, Table 1 shows that FoRA retains strong task quality across the evaluated model scales, while its client GPU-memory advantage persists across the tested input sequence lengths.

6.3 Scalability and Sensitivity

Hyperparameter sensitivity. We first examine how sensitive FoRA is to its key hyperparameters, including the local learning rate η^{loc}, the *Bridge* weight β, the block size b, the global-reference coefficient λ, the number of *Bridge* banks K, and the bottleneck dimension d_b. Fig. 2 reports controlled sweeps of these hyperparameters on OPT-350M@MRPC. Peak accuracy is highest at $\eta^{\text{loc}} = 2 \times 10^{-4}$, $\beta = 0.5$, and $\lambda = 7 \times 10^{-5}$. Increasing b from 1 to 2 reduces training time from 10.24 to 8.59 hours with only a 0.04-point accuracy change, whereas larger b reduces accuracy. The number of *Bridge* banks exhibits the strongest capacity-cost trade-off: $K = 1$ reaches 75.65% in 7.32 hours, while $K = 32$ reaches 72.67% in 20.20 hours; we adopt $K = 4$ as a compact configuration. Increasing d_b beyond 64 yields less than 0.15 points of additional accuracy while increasing training time. These results support the default configuration $(\eta^{\text{loc}}, \beta, b, \lambda, K, d_b) = (2 \times 10^{-4}, 0.5, 2, 7 \times 10^{-5}, 4, 64)$ as a compact quality-cost operating point.

Scalability and heterogeneity. We then examine whether FoRA remains effective as the federation grows and as client data become more heterogeneous. Table 2 reports client-count and statistical-heterogeneity scaling on OPT-350M@MRPC. From $N = 10$ to $N = 50$, FoRA maintains Final-10 Accuracy near 76%, while FedIT decreases from 73.71% to 70.42%; FoRA also retains its 1.306 GiB peak allocated client GPU memory. Moving from IID to $\alpha_{\text{Dir}} = 0.1$ reduces FoRA's Final-10 Accuracy/F1 by $3.49/2.16$ points, compared with $4.70/4.71$ points for FedIT. The resulting Final-10 F1 degradation is 54.1% smaller than that of FedIT, while FoRA's F1 advantage increases from 1.52 points under IID data to 4.07 points under strong heterogeneity. These results

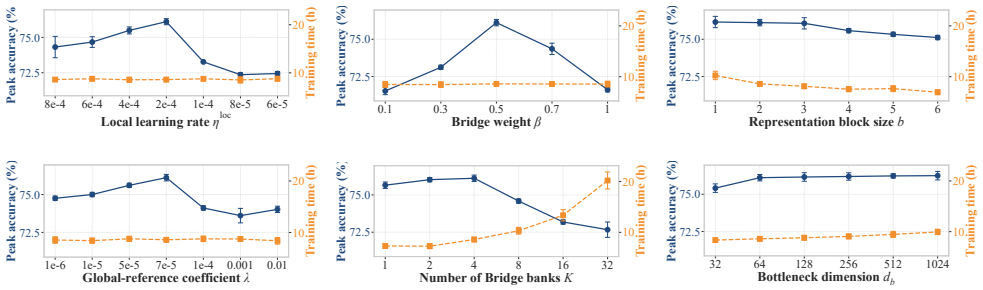

Figure 2: Sensitivity of FoRA to its key hyperparameters on OPT-350M@MRPC.

Table 2: Scalability results on OPT-350M@MRPC w.r.t. federation size and data heterogeneity.

Setting	Method	Peak Acc. (%) ↑	Peak F1 (%) ↑	Final-10 Acc. (%) ↑	Final-10 F1 (%) ↑	Training Time (h) ↓	Peak Memory (GiB) ↓
(a) Client Scalability ($\alpha_{\text{Dir}} = 0.7$)							
$N = 10$	FedIT (LoRA)	74.01 ± 0.30	83.33 ± 0.15	73.71 ± 0.23	83.10 ± 0.18	12.14 ± 0.58	3.180 ± 0.0
	FoRA	$\mathbf{76.11 \pm 0.21}$	$\mathbf{83.99 \pm 0.11}$	$\mathbf{75.68 \pm 0.35}$	$\mathbf{83.80 \pm 0.15}$	$\mathbf{8.59 \pm 0.46}$	$\mathbf{1.306 \pm 0.0}$
$N = 20$	FedIT (LoRA)	72.03 ± 0.43	82.77 ± 0.07	71.71 ± 0.31	82.63 ± 0.11	13.86 ± 0.33	3.180 ± 0.0
	FoRA	$\mathbf{76.73 \pm 0.50}$	$\mathbf{84.32 \pm 0.07}$	$\mathbf{76.24 \pm 0.45}$	$\mathbf{84.14 \pm 0.13}$	$\mathbf{9.80 \pm 0.36}$	$\mathbf{1.306 \pm 0.0}$
$N = 50$	FedIT (LoRA)	71.78 ± 0.14	82.67 ± 0.00	70.42 ± 0.12	81.97 ± 0.07	15.88 ± 0.25	3.180 ± 0.0
	FoRA	$\mathbf{76.49 \pm 0.00}$	$\mathbf{84.70 \pm 0.10}$	$\mathbf{76.26 \pm 0.18}$	$\mathbf{84.50 \pm 0.16}$	$\mathbf{11.64 \pm 0.25}$	$\mathbf{1.306 \pm 0.0}$
(b) Data Heterogeneity ($N = 10$)							
IID	FedIT (LoRA)	75.41 ± 0.14	83.64 ± 0.03	75.12 ± 0.24	83.21 ± 0.17	12.67 ± 0.29	3.180 ± 0.0
	FoRA	$\mathbf{77.15 \pm 0.29}$	$\mathbf{85.12 \pm 0.17}$	$\mathbf{76.81 \pm 0.29}$	$\mathbf{84.73 \pm 0.31}$	$\mathbf{8.85 \pm 0.48}$	$\mathbf{1.306 \pm 0.0}$
$\alpha_{\text{Dir}} = 0.7$	FedIT (LoRA)	74.01 ± 0.30	83.33 ± 0.15	73.71 ± 0.23	83.10 ± 0.18	12.14 ± 0.58	3.180 ± 0.0
	FoRA	$\mathbf{76.11 \pm 0.21}$	$\mathbf{83.99 \pm 0.11}$	$\mathbf{75.68 \pm 0.35}$	$\mathbf{83.80 \pm 0.15}$	$\mathbf{8.59 \pm 0.46}$	$\mathbf{1.306 \pm 0.0}$
$\alpha_{\text{Dir}} = 0.1$	FedIT (LoRA)	73.60 ± 0.14	82.68 ± 0.06	70.42 ± 3.12	78.50 ± 4.00	12.35 ± 0.36	3.180 ± 0.0
	FoRA	$\mathbf{75.99 \pm 0.40}$	$\mathbf{84.56 \pm 0.36}$	$\mathbf{73.32 \pm 0.23}$	$\mathbf{82.57 \pm 0.09}$	$\mathbf{9.56 \pm 0.31}$	$\mathbf{1.306 \pm 0.0}$

* Smaller α_{Dir} indicates stronger statistical heterogeneity.

demonstrate that FoRA is less sensitive to increasing federation size and statistical heterogeneity while maintaining stable per-client GPU memory.

6.4 Ablation Studies

Table 3: Ablation results on OPT-350M@MRPC.

Setting	Peak Acc. (%) ↑	Peak F1 (%) ↑	Training Time (h) ↓	Peak Memory (GiB) ↓
FoRA	76.11 ± 0.21	83.99 ± 0.11	8.59 ± 0.46	1.306 ± 0.0
Last-layer only	69.06 ± 1.23	74.36 ± 2.33	6.75 ± 0.40	1.305 ± 0.0
w/o *Split*	68.54 ± 0.53	76.87 ± 0.84	7.98 ± 0.37	1.306 ± 0.0
w/o *Bridge*	71.16 ± 0.56	79.47 ± 0.78	7.60 ± 0.23	1.306 ± 0.0
w/o Global Ref.	74.16 ± 1.01	83.27 ± 0.28	8.52 ± 0.30	1.306 ± 0.0

In this section, we examine how each component of FoRA contributes to task quality and client-side resource use. Table 3 isolates the main components on OPT-350M@MRPC. Using only the last-layer representation reduces Peak Accuracy/F1 by 7.05/9.63 points, indicating that retaining information from multiple Transformer depths is important in this setting. Removing the *Split* or *Bridge* branch reduces Peak Accuracy by 7.57 and 4.95 points, respectively, which supports their complementary roles in the functional residual. Removing global-reference regularization produces a 1.95-point Peak Accuracy reduction with nearly unchanged runtime, showing the contribution from the reference term in this setting. Peak client GPU memory remains nearly unchanged at approximately 1.31 GiB across all variants. Overall, the ablation shows that multi-level representations and both residual branches provide the largest quality gains, while global-reference regularization yields an additional improvement at little overhead.

7 Conclusion

We presented FoRA, a federated fine-tuning framework that keeps the LLM backbone forward-only while preserving first-order optimization for the trainable adaptation. FoRA detaches multi-level representations from a frozen backbone, learns a lightweight Split-Bridge functional residual, and regularizes local residual updates against the broadcast model in function space. We established memory, functional-geometry, drift, and non-convex convergence results. Across evaluated settings, FoRA achieves the best mean task performance across the reported metrics among the compared methods while reducing peak allocated client GPU memory by 27.3%-59.2% over LoRA-based methods and training 2.93×-3.96× faster than FedMeZO at comparable memory.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language polishing, manuscript revision, and internal review. We also used them to support the reproduction of related baselines by summarizing the workflow of FedProxy and by assisting our reading of the public implementations of GMFL and RAVAN when adapting them to our experimental settings. The authors inspected, executed, and validated all reproduced implementations. We also used AI tools to help organize experimental results and prepare plots. The underlying experimental outputs were produced by our implementations and were not generated, modified, or selectively adjusted by AI tools. We additionally used generative AI to help collect and organize potentially relevant literature. All cited papers were subsequently checked and read by the authors, who independently verified the claims and references used in the manuscript. After the theoretical analysis was developed, generative AI tools were used to check derivations, identify missing intermediate arguments, and flag potential issues for further inspection. No other use of generative AI was made in this work. We take full responsibility for the final content of the paper, including all text, claims, code, experiments, and theoretical results.

ETHICS STATEMENT

This work studies resource-efficient federated fine-tuning of large language models using publicly available pretrained models and benchmark datasets, including MRPC, SST-2, and Dolly-15K. We do not collect new human-subject data or release new datasets containing personal information. The federated experiments are conducted through simulated client partitions of these public datasets.

Although FoRA keeps raw client data local under the federated training protocol, it does not by itself provide formal privacy guarantees against information leakage through model updates, intermediate states, or malicious participants. Deployments involving sensitive data may therefore require complementary mechanisms such as secure aggregation, differential privacy, access control, or trusted execution environments. In addition, FoRA inherits the limitations and potential biases of the underlying pretrained models and training data, and resource-efficient adaptation may also make downstream customization of such models easier. Appropriate application-specific evaluation and safeguards remain necessary before deployment in sensitive or high-impact settings.

We report the experimental protocol, evaluation metrics, and implementation details needed to support reproducibility, and we take responsibility for the accuracy and integrity of the reported results.

REPRODUCIBILITY STATEMENT

We provide the information required to reproduce the proposed method, theoretical results, and reported experimental protocol throughout the paper and supplementary materials. The complete formulation of FoRA is given in Section 4, with the end-to-end training procedure and pseudocode provided in Appendix B. Appendix C states the assumptions and provides the proofs supporting the theoretical results. Appendix D documents the datasets and preprocessing, federated training protocol, model and optimization configurations, baseline settings, evaluation metrics, and resource-measurement procedure. The supplementary materials include the core implementation of FoRA corresponding to the method description and pseudocode, together with the implementations used for the reproduced baseline experiments.

REFERENCES

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm. 2023.

Tao Fan, Guoqiang Ma, Yuanfeng Song, Lixin Fan, Kai Chen, and Qiang Yang. Fedproxy: Federated fine-tuning of llms via proxy slms and heterogeneity-aware fusion. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 17482–17497. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.794. URL https://doi.org/10.18653/v1/2026.acl-long.794.

Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM journal on optimization*, 23(4):2341–2368, 2013.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

Xin Huang, Yan Hu, Yue-Jiao Gong, and Xinglin Zhang. GMFL: efficient global masking for federated LLM fine-tuning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 25293–25312. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.1160. URL https://doi.org/10.18653/v1/2026.acl-long.1160.

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

Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

Liang Li, Xingke Yang, Wen Wu, Hao Wang, Tomoaki Ohtsuki, Xin Fu, Miao Pan, and Xuemin Shen. Mobillm: Enabling on-device fine-tuning of billion-sized llms via server-assisted side-tuning. *IEEE J. Sel. Top. Signal Process.*, 19(7):1251–1265, 2025a. doi: 10.1109/JSTSP.2025.3633550. URL https://doi.org/10.1109/JSTSP.2025.3633550.

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.

Zhe Li, Bicheng Ying, Zidong Liu, Chaosheng Dong, and Haibo Yang. Achieving dimension-free communication in federated learning via zeroth-order optimization. In *International Conference on Learning Representations*, volume 2025, pp. 23769–23803, 2025b.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Zhenqing Ling, Daoyuan Chen, Liuyi Yao, Yaliang Li, and Ying Shen. On the convergence of zeroth-order federated tuning for large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1827–1838, 2024.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. *Advances in Neural Information Processing Systems*, 36:53038–53075, 2023.

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

Chuiyang Meng, Ming Tang, and Vincent W. S. Wong. Florg: Federated fine-tuning with low-rank gram matrices and procrustes alignment. *CoRR*, abs/2602.17095, 2026. doi: 10.48550/ARXIV.2602.17095. URL https://doi.org/10.48550/arXiv.2602.17095.

Arian Raje, Baris Askin, Divyansh Jhunjhunwala, and Gauri Joshi. Ravan: Multi-head low-rank adaptation for federated fine-tuning. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/456a6eed524d1d647c3366f1ef96d0ea-Abstract-Conference.html.

Yide Ran, Wentao Guo, Jingwei Sun, Yanzhou Pan, Xiaodong Yu, Hao Wang, Jianwen Xie, Yiran Chen, Denghui Zhang, and Zhaozhuo Xu. Mitigating non-iid drift in zeroth-order federated LLM fine-tuning with transferable sparsity. *CoRR*, abs/2506.03337, 2025. doi: 10.48550/ARXIV.2506.03337. URL https://doi.org/10.48550/arXiv.2506.03337.

Raghav Singhal, Kaustubh Ponkshe, and Praneeth Vepakomma. Fedex-lora: Exact aggregation for federated and efficient fine-tuning of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1316–1336, 2025.

Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. LST: ladder side-tuning for parameter and memory efficient transfer learning. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/54801e196796134a2b0ae5e8adef502f-Abstract-Conference.html.

Llama Team. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=rJ4km2R5t7.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/564127c03caab942e503ee6f810f54fd-Abstract.html.

Lei Wang, Jieming Bian, Letian Zhang, and Jie Xu. Adaptive lora experts allocation and selection for federated fine-tuning. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/6df1b2b45e64d402588746f79b68b82c-Abstract-Conference.html.

Mengwei Xu, Dongqi Cai, Yaozong Wu, Xiang Li, and Shangguang Wang. {FwdLLM}: Efficient federated finetuning of large language models with perturbed inferences. In *2024 USENIX Annual Technical Conference (USENIX ATC 24)*, pp. 579–596, 2024.

Xingke Yang, Liang Li, Sicong Li, Liwei Guan, Hao Wang, Xiaoqi Qi, Jiang Liu, Xin Fu, and Miao Pan. Fed mobillm: Efficient federated LLM fine-tuning over heterogeneous mobile devices via server assisted side-tuning. *CoRR*, abs/2508.06765, 2025. doi: 10.48550/ARXIV.2508.06765. URL https://doi.org/10.48550/arXiv.2508.06765.

Rui Ye, Wenhao Wang, Jingyi Chai, Dihan Li, Zexi Li, Yinda Xu, Yaxin Du, Yanfeng Wang, and Siheng Chen. Openfedllm: Training large language models on decentralized private data via federated learning. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 6137–6147, 2024.

Dun Zeng, Siqi Liang, Xiangjing Hu, Hui Wang, and Zenglin Xu. Fedlab: A flexible federated learning framework. *Journal of Machine Learning Research*, 24(100):1–7, 2023. URL http://jmlr.org/papers/v24/22-0440.html.

Jeffrey O. Zhang, Alexander Sax, Amir Zamir, Leonidas J. Guibas, and Jitendra Malik. Side-tuning: A baseline for network adaptation via additive side networks. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part III*, volume 12348 of *Lecture Notes in Computer Science*, pp. 698–714. Springer, 2020. doi: 10.1007/978-3-030-58580-8_41. URL https://doi.org/10.1007/978-3-030-58580-8_41.

Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Guoyin Wang, and Yiran Chen. Towards building the federated GPT: federated instruction tuning. *CoRR*, abs/2305.05644, 2023. doi: 10.48550/ARXIV.2305.05644. URL https://doi.org/10.48550/arXiv.2305.05644.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068, 2022. doi: 10.48550/ARXIV.2205.01068. URL https://doi.org/10.48550/arXiv.2205.01068.

A Comparison with State-of-the-Art Methods

Table 4 summarizes the key distinctions between FoRA and representative prior methods.

Table 4: Comparison of representative federated LLM fine-tuning methods. ● denotes explicit support as a native design property, ◐ denotes partial or indirect support, and ○ denotes that the capability is not supported by the original method.

Method	First-order Optimization	No Target LLM Backbone BP	Client-side Adaptation	No Activation Upload	No Proxy / Public Data	Multi-level Features	Non-IID Mitigation
FedIT (Zhang et al., 2023) / FedEx-LoRA (Singhal et al., 2025)	●	○	●	●	●	○	○
RAVAN (Raje et al., 2025) / GMFL (Huang et al., 2026)	●	○	●	●	●	○	●
FLoRG (Meng et al., 2026)	●	○	●	●	●	○	◐
FwdLLM (Xu et al., 2024)	○	●	●	●	●	○	○
FedMeZO (Ling et al., 2024)	○	●	●	●	●	○	●
Fed MobiLLM (Yang et al., 2025)	●	●	○	○	●	●	●
FedProxy (Fan et al., 2026)	●	●	●	●	○	○	●
FoRA (Ours)	●	●	●	●	●	●	●

"No Target LLM Backbone BP" indicates that local training does not require backpropagation through the target LLM backbone.
"Multi-level Features" denotes explicit use of representations extracted from multiple backbone depths for adaptation.

B Notation, Algorithm, and Complexity

In this section, we consolidate the notation used by FoRA, give the complete client-server procedure, and analyze its parameter, computation, memory, and communication costs.

B.1 Notation

We use the unsubscripted K for the number of *Bridge* banks; subscripted K-symbols in the theoretical analysis denote functional kernels. The number of clients selected in round t is denoted by $m_t = |\mathcal{S}_t|$. Unless stated otherwise, i, t, h, j, and k index clients, communication rounds, local updates, retained representation levels, and *Bridge* banks, respectively.

B.2 End-to-End Training Procedure

For a mini-batch ξ of input-label pairs, define the empirical regularized local objective

$$\widehat{\Phi}_i^t(w;\xi) = \frac{1}{|\xi|}\sum_{(x,y)\in\xi}\left[\ell\big(\{z_{0,p}(x)+\delta_{w,p}(x)\}_{p\in\mathcal{P}(x)},y\big) + \frac{\lambda}{2|\mathcal{P}(x)|}\sum_{p\in\mathcal{P}(x)}\big\|\delta_{w,p}(x)-\delta_{\mathrm{ref},p}^t(x)\big\|_2^2\right]. \tag{16}$$

where $\boldsymbol{\delta}_{\mathrm{ref},p}^t(x) = \mathrm{stopgrad}(\boldsymbol{\delta}_{\mathbf{w}^t,p}(x))$. We write $\widehat{F}_i^t$ and $\widehat{R}_i^t$ for the corresponding task and reference terms, so that $\widehat{\Phi}_i^t = \widehat{F}_i^t + \lambda\widehat{R}_i^t$. The minibatch objective averages examples uniformly. Algorithm 1 summarizes the complete synchronous procedure. The generic operator $\mathrm{LocalOptStep}$ denotes one first-order optimizer step.

The same broadcast parameters $\mathbf{w}^t$ initialize the local model and define the fixed global reference; no second model payload is required. The server aggregates the complete auxiliary state, including $\boldsymbol{\theta}$, $\boldsymbol{\psi}$, and the mixture logits $\mathbf{q}$. Backbone parameters, hidden representations, and reference outputs remain local to each client.

The pseudocode shows the generic on-the-fly execution schedule. In the reported OPT classification experiments, the frozen logits $z_0(x)$ and retained representations $R(x)$ are computed once per participating client and communication round and reused across local epochs. The fixed global-reference residual $\delta_{\mathrm{ref}}^t(x)$ is evaluated once per client and round before local updates. In the Llama-3.1-8B@Dolly-15K experiments, the two-GPU sharded implementation instead extracts the frozen-backbone representations online for each local minibatch. These execution schedules leave the corresponding auxiliary objectives unchanged.

Table 5: Notation for the federated setting and FoRA method.

Notation	Meaning
$N, \mathcal{C}_i, \mathcal{D}_i, n_i, p_i$	Number of clients, client i, its local dataset, local sample count $n_i = \lvert\mathcal{D}_i\rvert$, and full-population weight $p_i = n_i / \sum_j n_j$.
$t, T, \mathcal{S}_t, m_t, \omega_i^t$	Communication-round index, number of rounds, participating-client set, number of participating clients $m_t = \lvert\mathcal{S}_t\rvert$, and selected-set weight $\omega_i^t = n_i / \sum_{j\in\mathcal{S}_t} n_j$.
$E, B, H_i, \eta_t^{\mathrm{loc}}$	Number of local epochs, mini-batch size, number of local updates $H_i = E\lceil n_i/B\rceil$, and local learning rate.
$f_0, \boldsymbol{\phi}, L, S, d_h, C$	Frozen backbone, its parameters, number of Transformer layers, input sequence length, hidden dimension, and task-output dimension.
$p, \mathcal{P}(x)$	Supervised prediction site and the set of supervised prediction sites for input x.
$b, M, \mathcal{L}$	Transformer block size, number of retained representation levels $M = \lceil L/b\rceil$, and selected layer indices $\mathcal{L} = \{\ell_j\}_{j=1}^M$ with $\ell_j = \min(jb, L)$.
$\rho_p, \mathbf{H}_{\ell_j}(x), \mathbf{r}_{j,p}(x), \mathcal{R}_p(x)$	Prediction-site extractor, hidden states at layer ℓ_j, detached representation from level j at site p, and multi-level representation set $\mathcal{R}_p(x) = \{\mathbf{r}_{j,p}(x)\}_{j=1}^M$.
$\mathbf{z}_{0,p}(x), \mathbf{z}_{\mathbf{w},p}(x)$	Frozen and adapted task logits at prediction site p.
$d_c, \mathbf{u}_{\mathbf{w}}$	Dimension of the auxiliary correction space and the Split–Bridge correction before the fixed task readout.
$\mathbf{A}$	Fixed readout from the correction space to the task-output space; $\mathbf{A} = \mathbf{I}_C$ for classification and the frozen language-model head for generation.
$d_b, \mathbf{a}_{\boldsymbol{\omega}}$	Bottleneck width and one block adapter from $\mathbb{R}^{d_h}$ to $\mathbb{R}^{d_c}$.
$K, \boldsymbol{\theta}, \boldsymbol{\psi}, \mathbf{q}$	Number of Bridge banks, *Split*-branch parameters, *Bridge*-bank parameters, and trainable *Bridge* mixture logits.
$\boldsymbol{\pi}(\mathbf{q}), \beta$	Softmax-normalized *Bridge* coefficients and the fixed *Bridge* scaling coefficient.
$\mathbf{s}_{\boldsymbol{\theta}}, \mathbf{b}_k, \mathbf{b}_{\boldsymbol{\psi},\mathbf{q}}$	*Split* correction, *Bridge* bank k, and the learned *Bridge* mixture in correction space.
$\boldsymbol{\delta}_{\mathbf{w},p}(x), \mathbf{w}$	Task-output residual at prediction site p and all trainable auxiliary parameters $\mathbf{w} = (\boldsymbol{\theta}, \boldsymbol{\psi}, \mathbf{q})$.
$\boldsymbol{\delta}_{\mathrm{ref},p}^t, R_i^t, \lambda$	Fixed global residual at prediction site p in round t, client functional-reference regularizer, and its coefficient.
$F_i, F, \widetilde{F}_i^t$	Client task objective, sample-weighted global task objective, and regularized local objective $\widetilde{F}_i^t = F_i + \lambda R_i^t$.
$\widehat{F}_i^t, \widehat{R}_i^t, \widehat{\Phi}_i^t$	Minibatch task loss, reference penalty, and regularized objective $\widehat{\Phi}_i^t = \widehat{F}_i^t + \lambda \widehat{R}_i^t$; the subscript i, r identifies the realized minibatch at local step r.
$\mathbf{w}_i^{t,h}$	Client-i auxiliary parameters after h local updates in round t.

B.3 Complexity Analysis

The closed-form expressions in this subsection describe the pooled logit-space parameterization used in the classification experiments. The generation variant in Appendix D.3 uses hidden-space corrections at supervised response positions; its auxiliary costs also depend on the number of these positions and the frozen language-model head.

Auxiliary parameter count. The parameter count follows the general correction-space formulation. One adapter contains an affine LayerNorm, two projections, and two bias vectors:

$$P_{\mathrm{a}} = d_b d_h + d_b + d_c d_b + d_c + 2d_h = d_b(d_h + d_c + 1) + d_c + 2d_h. \tag{17}$$

The *Split* branch contains M adapters, while the *Bridge* branch contains KM adapters and K mixture logits. Hence,

$$P_{\mathrm{aux}} = M(K+1)P_{\mathrm{a}} + K. \tag{18}$$

Table 6: Notation reused in the theoretical and complexity analyses.

Notation	Meaning
$P_{\mathrm{a}}, P_{\mathrm{aux}}$	Parameter count of one block adapter and of the complete trainable auxiliary model.
$\mathbb{C}_{\mathrm{bb}}^{\mathrm{fwd}}, \mathbb{C}_{\rho}$	Cost of one frozen-backbone forward pass and of extracting the retained prediction-site representations.
$\mathbb{C}_{\mathrm{aux}}^{\mathrm{fwd}}, \mathbb{C}_{\mathrm{aux}}^{\mathrm{bwd}}$	Forward and backward costs of the complete Split–Bridge auxiliary model.
M_{FoRA}	Peak allocated client GPU memory under the FoRA execution schedule analyzed in the complexity section.
$M_{\mathrm{aux}}^{\mathrm{train}}$	Peak memory of differentiating the auxiliary residual path with respect to w, including auxiliary training states and fixed-readout workspace.
M_{ref}	Memory associated with the fixed round-reference evaluation.
$\mathbf{J}_{\mathrm{s}}, \mathbf{J}_{\mathrm{b}}, \mathbf{J}_{\mathrm{FoRA}}$	Jacobians of the Split, Bridge, and complete FoRA correction on the same normalized analysis set.
$\mathbf{K}_{\mathrm{s}}, \mathbf{K}_{\mathrm{b}}, \mathbf{K}_{\mathrm{FoRA}}$	Corresponding functional kernels $\mathbf{K} = \mathbf{J}\mathbf{J}^{\top}$.
d_a	Total dimension of the trainable auxiliary parameters, equal to P_{aux}.
$A_{\mathcal{S}}, C_H, \nu_H^2$	Weighted local-update count for a participating set, its mean, and its variance.
$\widetilde{p}_i, \widetilde{F}$	Effective client weights and objective $\widetilde{F} = \sum_i \widetilde{p}_i F_i$, distinct from the regularized local objective $\widetilde{F}_i^t$.
$\mathcal{B}_{\mathrm{traj}}^2(\lambda), \zeta^2, V_{\mathrm{rr}}$	Bounds on the aggregated squared local-trajectory residual, client gradient heterogeneity, and local sampling noise.
s_{w}	Number of bytes used to communicate one auxiliary scalar.

For classification, $d_c = C$; for generation, $d_c = d_h$, and the frozen language-model head contributes no trainable parameters.

Client computation. For a minibatch ξ, define the number of supervised prediction sites as

$$P_\xi = \sum_{x \in \xi} |\mathcal{P}(x)|. \tag{19}$$

Thus, $P_\xi = |\xi| \le B$ for sequence classification, whereas for autoregressive generation it equals the number of supervised response positions in the minibatch. Let $\mathbb{C}_{\mathrm{bb}}^{\mathrm{fwd}}(B, S)$ denote one frozen-backbone forward pass. For the prediction-site extractors used by FoRA,

$$\mathbb{C}_{\rho}(\xi) = \Theta(P_\xi M d_h). \tag{20}$$

A forward pass through the Split–Bridge correction model costs

$$\mathbb{C}_{\mathrm{aux}}^{\mathrm{fwd}}(\xi) = \Theta\big(P_\xi M(K+1)\big(d_h d_b + d_b d_c + d_h + d_b + d_c\big) + P_\xi K d_c\big)\,, \tag{21}$$

where the first term accounts for the $M(K+1)$ adapters and the second for mixing the K Bridge outputs. Reverse-mode differentiation through the auxiliary model has the same asymptotic order:

$$\mathbb{C}_{\mathrm{aux}}^{\mathrm{bwd}}(\xi) = \Theta\big(\mathbb{C}_{\mathrm{aux}}^{\mathrm{fwd}}(\xi)\big)\,. \tag{22}$$

For generation, the fixed language-model readout adds computation at the same P_ξ supervised positions; this cost is separate from auxiliary-model forward and backward costs ($\mathbb{C}_{\mathrm{aux}}^{\mathrm{fwd}}$, $\mathbb{C}_{\mathrm{aux}}^{\mathrm{bwd}}$) because the readout is not part of the trainable auxiliary model.

For a minibatch ξ, one local update consists of one backbone forward pass and representation extraction, one reference auxiliary forward pass, and one trainable auxiliary forward–backward pass. Including the optimizer update,

$$\mathbb{C}_{\mathrm{step}}(\xi) = \mathbb{C}_{\mathrm{bb}}^{\mathrm{fwd}} + \mathbb{C}_{\rho}(\xi) + 2\mathbb{C}_{\mathrm{aux}}^{\mathrm{fwd}}(\xi) + \mathbb{C}_{\mathrm{aux}}^{\mathrm{bwd}}(\xi) + \Theta(P_{\mathrm{aux}}), \tag{23}$$

with the fixed-readout cost added for generation. Under the generic online schedule, Client i therefore performs

$$\mathbb{C}_{i,t}^{\mathrm{client}} = \sum_{h=0}^{H_i - 1} \mathbb{C}_{\mathrm{step}}(\xi_{i,h}^t) \tag{24}$$

Algorithm 1: FoRA: backbone-detached first-order federated fine-tuning

Input: Frozen backbone $f_0(\cdot;\boldsymbol{\phi})$; initial auxiliary parameters $\mathbf{w}^0$; client datasets $\{\mathcal{D}_i\}_{i=1}^N$; rounds T; participation counts $\{m_t\}_{t=0}^{T-1}$; local epochs E; mini-batch size B; regularization coefficient λ; local learning rates $\{\eta_t^{\mathrm{loc}}\}_{t=0}^{T-1}$

Output: Global auxiliary parameters $\mathbf{w}^T$

for $t = 0, \dots, T-1$ **do**

  `// Server: select clients and distribute the current auxiliary model`

  Sample $\mathcal{S}_t$ uniformly without replacement with $|\mathcal{S}_t| = m_t$;

  Broadcast $\mathbf{w}^t$ to every client $i \in \mathcal{S}_t$;

  `// Clients: backbone-detached local first-order optimization`

  foreach $i \in \mathcal{S}_t$ ***in parallel*** **do**

    $\mathbf{w}_i^{t,0} \leftarrow \mathbf{w}^t$; keep an immutable reference to $\mathbf{w}^t$;

    $h \leftarrow 0$;

    for $e = 1, \dots, E$ **do**

      Randomly order $\mathcal{D}_i$ and form mini-batches of size at most B;

      foreach *mini-batch* $\xi_{i,h}^t$ **do**

        Run f_0 with gradient tracking disabled to obtain $\mathbf{z}_0(x)$ and $\mathcal{R}(x)$ for all $x \in \xi_{i,h}^t$;

        $\boldsymbol{\delta}_{\mathrm{ref},p}^t(x) \leftarrow \mathrm{stopgrad}(\boldsymbol{\delta}_{\mathbf{w}^t,p}(x))$ for all $x \in \xi_{i,h}^t$;

        Evaluate $\widehat{\Phi}_i^t(\mathbf{w}_i^{t,h}; \xi_{i,h}^t)$ using Eq. 16;

        $\mathbf{w}_i^{t,h+1} \leftarrow \mathrm{LocalOptStep}\Big(\mathbf{w}_i^{t,h}, \nabla_{\mathbf{w}} \widehat{\Phi}_i^t(\mathbf{w}_i^{t,h}; \xi_{i,h}^t), \eta_t^{\mathrm{loc}}\Big)$;

        $h \leftarrow h + 1$;

    $H_i \leftarrow h$ and upload $(\mathbf{w}_i^{t,H_i}, n_i)$ to the server;

  `// Server: sample-weighted aggregation`

  foreach $i \in \mathcal{S}_t$ **do**

    $\omega_i^t \leftarrow n_i / \sum_{j \in \mathcal{S}_t} n_j$;

  $\mathbf{w}^{t+1} \leftarrow \sum_{i \in \mathcal{S}_t} \omega_i^t \mathbf{w}_i^{t,H_i}$;

work in round t. When all classification minibatches are full-sized, $P_\xi = B$ and this reduces to $H_i C_{\mathrm{step}}$; the retained final minibatch may have $P_\xi < B$.

The fixed backbone outputs and detached representations may be precomputed once for deterministic reuse. If Ξ_i denotes the minibatches in one local pass, the one-time preprocessing cost is

$$\mathbb{C}_i^{\mathrm{base}} = \sum_{\xi \in \Xi_i} \left(\mathbb{C}_{\mathrm{bb}}^{\mathrm{fwd}} + \mathbb{C}_\rho(\xi) \right). \tag{25}$$

The reference outputs may likewise be evaluated once per round and reused during local optimization. The reported OPT classification experiments use this reuse schedule: the backbone-forward and representation extraction terms are incurred once per participating client and round instead of at every local update, while the local auxiliary updates reuse the frozen outputs and representations. The Llama-3.1-8B@Dolly-15K experiments use the online schedule in Eq. 23.

Server computation and parallel latency. The server averages m_t auxiliary states, requiring

$$\mathbb{C}_t^{\mathrm{server}} = \Theta(m_t P_{\mathrm{aux}}) \tag{26}$$

scalar operations. For any chosen client execution schedule, the aggregate work in round t is

$$\mathbb{C}_t^{\mathrm{system}} = \sum_{i \in \mathcal{S}_t} \mathbb{C}_{i,t}^{\mathrm{client}} + \Theta(m_t P_{\mathrm{aux}}). \tag{27}$$

With parallel clients, computation latency is determined by the slowest selected client followed by server aggregation; communication and straggler effects are system-dependent.

Memory. For a mini-batch ξ, the detached execution graph gives

$$M_{\mathrm{FoRA}}(\xi) \le M_{\mathrm{fwd}}(B, S) + \mathcal{O}(P_\xi M d_h) + M_{\mathrm{aux}}^{\mathrm{train}}(\xi) + M_{\mathrm{ref}}(\xi), \tag{28}$$

which is Eq. 11. The second term stores the detached representations retained at the P_ξ supervised prediction sites across M backbone levels. Transient backbone tensors used before extraction are included in $M_{\mathrm{fwd}}(B, S)$.

Here $M_{\mathrm{aux}}^{\mathrm{train}}$ includes the auxiliary parameters, activations, gradients, optimizer states, and any fixed-readout workspace required to propagate gradients to w. The reference term M_{ref} depends on the execution schedule; an immutable reference copy requires P_{aux} parameter scalars.

For sequence classification, $P_\xi = |\xi| \le B$, and the retained-feature term is bounded by $\mathcal{O}(BMd_h)$. For generation, P_ξ counts supervised response positions. The Gram-matrix implementation avoids retaining vocabulary-dimensional reference residuals at all such positions. In every execution schedule, no detached backbone representation retains gradient history.

Communication. Assume that the frozen backbone is already resident at each client and that one transmitted auxiliary scalar occupies s_{w} bytes. Each selected client uploads one complete auxiliary state and the sample count used for aggregation:

$$\mathrm{Comm}_{\mathrm{up}}(i,t) = s_{\mathrm{w}} P_{\mathrm{aux}} + \mathcal{O}(1) \text{ bytes.} \tag{29}$$

The server broadcasts the current auxiliary state once per round,

$$\mathrm{Comm}_{\mathrm{down}}(i,t) = s_{\mathrm{w}} P_{\mathrm{aux}} + \mathcal{O}(1) \text{ bytes per selected client.} \tag{30}$$

Under client-wise unicast, the total uplink and downlink traffic are both $\Theta(m_t P_{\mathrm{aux}})$ scalar words; a multicast implementation can transmit the common downlink payload once over shared links. The global-reference model introduces no additional network payload because the same $\mathbf{w}^t$ is used for client initialization and as the immutable round reference. FoRA also communicates no intermediate backbone activations.

Overall, the backward and communication costs scale with P_{aux}, while the LLM contributes only forward computation. Increasing K, decreasing the block size b, or increasing the bottleneck width d_b enlarges the auxiliary capacity and raises both local auxiliary cost and model communication according to Eq. 18. None of these choices restores a backward path through the backbone.

C Proofs and Additional Theoretical Details

This section gives the assumptions, exact constants, auxiliary results, and proofs supporting Section 5. Appendix C.1 collects the conditions and their scopes. The subsequent subsections develop graph separation, an architecture-controlled FO–ZO comparison, functional geometry, local drift, and federated convergence, followed by the minibatch specialization.

C.1 Notation, Scope, and Assumptions

Notation and scope. The frozen backbone has parameters $\boldsymbol{\phi}$, produces logits $\mathbf{z}_0(x)$, and supplies detached representations $\mathcal{R}(x)$. The trainable parameters are $\mathbf{w} = (\boldsymbol{\theta}, \boldsymbol{\psi}, \mathbf{q}) \in \mathbb{R}^{d_a}$, where d_a is their total dimension. At round t, the reference $\boldsymbol{\delta}_{\mathrm{ref}}^t(x) = \mathrm{stopgrad}(\boldsymbol{\delta}_{\mathbf{w}^t}(x))$ is held fixed during local differentiation. The residual maps and task losses are differentiable at the points considered. The graph-separation and kernel-decomposition results follow from this execution graph and the disjoint Split-Bridge parameter blocks. The functional-geometry comparison uses the Split-only restriction of the same parameterization as its reference model. The fixed-tangent results describe an affine local surrogate; the federated convergence results analyze constant-step local SGD.

The results below have three scopes. Appendix C.2 gives exact consequences of the detached execution graph. Appendices C.3-C.4 characterize oracle behavior and the local geometry of the auxiliary parameterization. Appendix C.5 isolates the global-reference mechanism through a fixed-tangent surrogate. Finally, Appendices C.6-C.9 establish convergence for the nonlinear federated protocol, keeping the local-trajectory residual explicit.

For a matrix $\mathbf{A}$, $\mathrm{Range}(\mathbf{A})$ and $\mathrm{rank}(\mathbf{A})$ denote its column space and rank. The norm $\|\cdot\|_2$ is the Euclidean norm for vectors and the induced operator norm for matrices. For a symmetric matrix, $\lambda_{\max}$ and $\lambda_{\min}$ denote the extreme eigenvalues, and $\lambda_{\min}^{+}$ denotes the smallest strictly positive eigenvalue when one exists. The relation $\mathbf{A} \succeq \mathbf{B}$ means that $\mathbf{A} - \mathbf{B}$ is positive semidefinite. Identity and zero matrices have the dimensions determined by their operands, and $\mathbf{1}\{\cdot\}$ denotes an indicator.

For each supervised prediction site, write $\delta_{w,p} = A\, u_w(R_p)$. All functional Jacobians in the analysis refer to the final task-output residual δ. If $\widetilde{J}_{u,p}$ denotes the Jacobian of the correction-space output, then $J_{\delta,p} = A\,\widetilde{J}_{u,p}$. Stacking multiple prediction sites gives the corresponding block-diagonal application of the fixed readout.

Conditional expectations. We condition on the client datasets and initial model. Let $\mathcal{H}_t$ be the history immediately before client selection in round t, so $\mathbf{w}^t$ is $\mathcal{H}_t$-measurable. For an integrable random quantity X, write

$$\mathbb{E}_t[X] = \mathbb{E}[X \mid \mathcal{H}_t], \qquad \mathbb{E}_{t,\mathcal{S}}[X] = \mathbb{E}[X \mid \mathcal{H}_t, \mathcal{S}_t]. \tag{31}$$

Thus, $\mathbb{E}_t$ averages over both client selection and subsequent local randomness, while $\mathbb{E}_{t,\mathcal{S}}$ keeps the selected set fixed. Unsubscripted expectations also average over earlier rounds. In the separate oracle comparison, $\mathcal{F}_k^{\mathrm{or}}$ denotes the history before oracle call k, and $\mathbb{E}_k[X] = \mathbb{E}[X \mid \mathcal{F}_k^{\mathrm{or}}]$.

Federated protocol and objective. For the realized minibatch $\xi_{i,r}^t$ at local step r, write $\widehat{F}_{i,r}^t(\mathbf{w}) = \widehat{F}_i^t(\mathbf{w}; \xi_{i,r}^t)$ for its task loss and $\widehat{R}_{i,r}^t(\mathbf{w}) = \widehat{R}_i^t(\mathbf{w}; \xi_{i,r}^t)$ for its reference penalty, including the factor $1/2$. The regularized minibatch objective is $\widehat{\Phi}_{i,r}^t(\mathbf{w}) = \widehat{\Phi}_i^t(\mathbf{w}; \xi_{i,r}^t)$ from Eq. 16, namely

$$\widehat{\Phi}_{i,r}^t(\mathbf{w}) = \widehat{F}_{i,r}^t(\mathbf{w}) + \lambda \widehat{R}_{i,r}^t(\mathbf{w}). \tag{32}$$

The reference and current residual coincide at $\mathbf{w}^t$, so $\nabla \widehat{R}_{i,r}^t(\mathbf{w}^t) = \mathbf{0}$. This equality is a property of the reference construction.

Assumption 1 (Federated sampling and local-SGD protocol)**.** *The client datasets are fixed with $n_i > 0$, and each client performs a fixed number $H_i \geq 1$ of local steps. The selected set is nonempty and has a time-independent conditional distribution $\mathbb{P}$:*

$$\Pr(\mathcal{S}_t = \mathcal{S} \mid \mathcal{H}_t) = \mathbb{P}(\mathcal{S}). \tag{33}$$

Selected clients use a common constant step size $\eta > 0$ and the updates

$$\mathbf{w}_i^{t,r+1} = \mathbf{w}_i^{t,r} - \eta \nabla \widehat{\Phi}_{i,r}^t(\mathbf{w}_i^{t,r}), \qquad \mathbf{w}_i^{t,0} = \mathbf{w}^t, \qquad 0 \leq r < H_i. \tag{34}$$

The server uses sample-weighted averaging of the resulting client parameters, as in Algorithm 1.

Justification. Uniformly sampling a fixed number of clients in each round satisfies the stated selection rule. Fixed local epochs and fixed client datasets determine the client-specific H_i. Together, these conditions give time-independent effective weights and permit analysis of a single federated objective. The update equation specifies the local-SGD specialization used by the convergence proof.

For a generic set $\mathcal{S} \sim \mathbb{P}$, define the selected-set weights by

$$\omega_i(\mathcal{S}) = \begin{cases} n_i / \sum_{j \in \mathcal{S}} n_j, & i \in \mathcal{S}, \\ 0, & i \notin \mathcal{S}. \end{cases} \tag{35}$$

The expected local-step coefficients, effective objective, and round-step statistics are

$$c_i = \mathbb{E}_{\mathcal{S}}[\mathbf{1}\{i \in \mathcal{S}\}\omega_i(\mathcal{S})H_i], \qquad C_H = \sum_{i=1}^N c_i > 0, \tag{36}$$

$$\widetilde{p}_i = c_i / C_H, \qquad \widetilde{F}(\mathbf{w}) = \sum_{i=1}^N \widetilde{p}_i F_i(\mathbf{w}), \tag{37}$$

$$A_{\mathcal{S}} = \sum_{i \in \mathcal{S}} \omega_i(\mathcal{S}) H_i, \qquad \nu_H^2 = \mathbb{E}_{\mathcal{S}}[(A_{\mathcal{S}} - C_H)^2], \qquad H_{\max} = \max_i H_i. \tag{38}$$

Here $C_H = \mathbb{E}_{\mathcal{S}}[A_{\mathcal{S}}]$. These quantities are definitions induced by the protocol. Appendix C.6 proves their relation to the expected round-start direction.

Assumption 2 (Smoothness and lower boundedness of the effective objective)**.** *The objective $\widetilde{F}$ is differentiable on an open neighborhood of a convex region $\mathcal{W}_{\mathrm{fed}}$ containing the global iterates and the segments between consecutive iterates, almost surely. For some finite $L_{\widetilde{F}} > 0$,*

$$\|\nabla \widetilde{F}(\mathbf{u}) - \nabla \widetilde{F}(\mathbf{v})\|_2 \le L_{\widetilde{F}} \|\mathbf{u} - \mathbf{v}\|_2, \qquad \mathbf{u}, \mathbf{v} \in \mathcal{W}_{\mathrm{fed}}. \tag{39}$$

The lower bound $\inf_{\mathbf{w}} \widetilde{F}(\mathbf{w}) > -\infty$ and the initial value $\widetilde{F}(\mathbf{w}^0)$ are finite.

Justification. A fixed nonnegative weighted average of smooth client objectives is smooth on their common region. Nonnegative task losses also provide a finite lower bound. Smoothness controls the change in the objective after server aggregation, and lower boundedness permits summing the descent inequalities over rounds. The region includes the segments used by the descent argument, so the condition applies between successive iterates as well as at the iterates themselves.

Define the initial objective gap by

$$\widetilde{F}^\star = \inf_{\mathbf{w}} \widetilde{F}(\mathbf{w}), \qquad \Delta_{\widetilde{F}} = \widetilde{F}(\mathbf{w}^0) - \widetilde{F}^\star. \tag{40}$$

Assumption 3 (Client gradient heterogeneity)**.** (a) *For the effective-objective analysis, the client gradients satisfy*

$$\sum_{i=1}^{N} \widetilde{p}_i \|\nabla F_i(\mathbf{w}) - \nabla \widetilde{F}(\mathbf{w})\|_2^2 \le \zeta^2 \tag{41}$$

for all global iterates, with finite $\zeta^2 \ge 0$. (b) *For the comparison with $F(\mathbf{w}) = \sum_i p_i F_i(\mathbf{w})$, where $p_i = n_i / \sum_j n_j$, assume on the points being compared that*

$$\sum_{i=1}^{N} p_i \|\nabla F_i(\mathbf{w}) - \nabla F(\mathbf{w})\|_2^2 \le \zeta_F^2 \tag{42}$$

for finite $\zeta_F^2 \ge 0$.

Justification. These bounds quantify gradient disagreement caused by heterogeneous client data. For finitely many clients with continuous gradients on a common compact region, both weighted disagreements have finite upper bounds. Part (a) controls the sampled round-start direction, and part (b) is used only to transfer the guarantee to the sample-weighted objective. Keeping the two weightings explicit separates statistical heterogeneity from the effect of unequal participation and local-step counts.

Local sampling and trajectory errors. Define the fixed-point sampling error and the accumulated change along a local trajectory as

$$\boldsymbol{\xi}_i^t = \sum_{r=0}^{H_i - 1} [\nabla \widehat{F}_{i,r}^t(\mathbf{w}^t) - \nabla F_i(\mathbf{w}^t)], \tag{43}$$

$$\mathbf{r}_i^t = \sum_{r=0}^{H_i - 1} [\nabla \widehat{\Phi}_{i,r}^t(\mathbf{w}_i^{t,r}) - \nabla \widehat{\Phi}_{i,r}^t(\mathbf{w}^t)]. \tag{44}$$

The first quantity evaluates every task minibatch at the same round model. The second includes changes in both task and reference gradients during local training. Their endpoint decomposition is proved in Lemma 33.

Assumption 4 (Controlled local sampling noise)**.** *For every round t, the aggregated fixed-point error satisfies*

$$\mathbb{E}_t \left[\sum_{i \in \mathcal{S}_t} \omega_i(\mathcal{S}_t) \boldsymbol{\xi}_i^t \right] = \mathbf{0}, \tag{45}$$

$$\mathbb{E}_t \left\| \sum_{i \in \mathcal{S}_t} \omega_i(\mathcal{S}_t) \boldsymbol{\xi}_i^t \right\|_2^2 \le V_{\mathrm{rr}}, \tag{46}$$

where $V_{\mathrm{rr}} \ge 0$ is finite and uniform over rounds. The expectation is conditioned on the pre-selection history as in Eq. 31.

Justification. Uniform reshuffling gives each minibatch average the full empirical gradient in expectation when evaluated at the fixed round model. This property holds for an incomplete final minibatch and permits dependence between minibatches within an epoch. Appendix C.9 derives the aggregate variance under independent client and epoch permutations. Averaging that conditional calculation over the selected set yields the V_{rr} used here. The assumption concerns fixed-point errors; the change caused by moving local parameters is included in $\mathbf{r}_i^t$.

Assumption 5 (Minibatch smoothness and bounded local gradients)**.** *For each client i, the regularized minibatch objectives have finite constants $L_{i,\mathrm{mb}}$ and $G_{i,\mathrm{mb}}$, uniform over rounds, local steps, and possible minibatches, such that*

$$\|\nabla\widehat{\Phi}_{i,r}^t(\mathbf{u}) - \nabla\widehat{\Phi}_{i,r}^t(\mathbf{v})\|_2 \le L_{i,\mathrm{mb}}\|\mathbf{u}-\mathbf{v}\|_2 \tag{47}$$

for $\mathbf{u}, \mathbf{v}$ in a common neighborhood of the round model and local trajectory, and

$$\|\nabla\widehat{\Phi}_{i,r}^t(\mathbf{w}_i^{t,r})\|_2 \le G_{i,\mathrm{mb}}. \tag{48}$$

Justification. For finite data, continuous gradients and locally bounded Hessians provide these constants on a compact parameter region containing the analyzed trajectories and reference parameters. The constants include the contribution of the reference penalty and may depend on λ. This condition gives a direct nonlinear residual bound and allows the auxiliary Jacobian to vary during training. Containment in such a region is part of the regularity condition being assumed.

Assumption 6 (Bounded local-trajectory residual)**.** *For every selected client and round, almost surely,*

$$\|\mathbf{r}_i^t\|_2 \le \Gamma_i(\lambda, H_i), \tag{49}$$

where $\Gamma_i(\lambda, H_i)$ is a finite deterministic bound uniform over rounds and sampling outcomes. Its notation keeps λ and H_i explicit; dependence on the fixed step size and regularity constants is suppressed.

Justification. Smooth minibatch gradients and a bounded local gradient norm give the explicit bound $\eta L_{i,\mathrm{mb}} G_{i,\mathrm{mb}} H_i(H_i-1)/2$ in Lemma 33. Corollary 28 gives a separate expression for the fixed-tangent surrogate. These provide sufficient conditions for the residual bound in their respective settings. A uniform bound over rounds requires uniform regularity constants; a bound obtained at a single local linearization applies to that linearization.

Define the aggregate residual measure by

$$\mathcal{B}_{\mathrm{traj}}^2(\lambda) = \mathbb{E}_{\mathcal{S}}\left[\sum_{i\in\mathcal{S}} \omega_i(\mathcal{S})\Gamma_i(\lambda, H_i)^2\right]. \tag{50}$$

This definition and the nonnegative aggregation weights imply the bound used in Eq. 156.

Minibatch sampling specializations. For a fixed parameter $\mathbf{w}$, let $\mathbf{g}_{i,j}(\mathbf{w})$ be the gradient of example j, with $\nabla F_i(\mathbf{w}) = n_i^{-1}\sum_{j=1}^{n_i}\mathbf{g}_{i,j}(\mathbf{w})$. Define

$$\sigma_i^2(\mathbf{w}) = \frac{1}{n_i}\sum_{j=1}^{n_i}\|\mathbf{g}_{i,j}(\mathbf{w}) - \nabla F_i(\mathbf{w})\|_2^2. \tag{51}$$

The next two assumptions specify alternative sampling schemes.

Assumption 7 (Uniform random reshuffling)**.** *Client i uses E_i epochs, each consisting of H_i^{ep} minibatches with fixed positive sizes $B_{i,1}, \ldots, B_{i,H_i^{\mathrm{ep}}}$, where $\sum_h B_{i,h} = n_i$ and $H_i = E_i H_i^{\mathrm{ep}}$. Conditional on $(\mathcal{H}_t, \mathcal{S}_t)$, each epoch uses a uniform permutation of the local examples, independently across selected clients and epochs. At fixed $\mathbf{w}$, each minibatch gradient is the arithmetic mean of its example gradients, and these example gradients are deterministic. For some finite $\overline{\sigma}_i^2$,*

$$\sigma_i^2(\mathbf{w}^t) \le \overline{\sigma}_i^2 \tag{52}$$

holds at every round model almost surely.

Justification. A full pass through a uniformly permuted dataset gives uniform marginal inclusion at every position. This yields the fixed-point mean identity and the finite-population covariance calculation in Appendix C.9. Independent client and epoch permutations permit summing their variances. For autoregressive generation, token-level task and reference terms are first averaged within each example and the resulting example losses are then averaged over the minibatch. Hence each minibatch gradient remains an arithmetic mean of example gradients.

Assumption 8 (Independent minibatch sampling)**.** *As an alternative to Assumption 7, client i draws minibatches of fixed size B_i independently with replacement across local steps and clients, conditional on $(\mathcal{H}_t, \mathcal{S}_t)$. At the round model,*

$$\mathbb{E}_{t,\mathcal{S}}[\nabla \widehat{F}^t_{i,r}(\mathbf{w}^t)] = \nabla F_i(\mathbf{w}^t), \qquad \mathbb{E}_{t,\mathcal{S}}\|\nabla \widehat{F}^t_{i,r}(\mathbf{w}^t) - \nabla F_i(\mathbf{w}^t)\|_2^2 \le \overline{\sigma}_i^2/B_i, \tag{53}$$

with finite constants $\overline{\sigma}_i^2$ uniform over rounds.

Justification. Uniform independent sampling of examples gives the empirical gradient in expectation, and averaging B_i independent gradients reduces their variance by B_i. Independence across local steps and clients yields the aggregate expression in Eq. 192. This is an alternative sufficient construction for Assumption 4.

Fixed-tangent functional analysis. For each example x, define the normalized residual stack $\widetilde{\delta}_w(x) = \frac{1}{\sqrt{|\mathcal{P}(x)|}} \operatorname{col}\big(\delta_{w,p}(x) : p \in \mathcal{P}(x)\big)$. For a finite ordered analysis set $\mathcal{X}_i$, define $\delta_w = \frac{1}{\sqrt{|\mathcal{X}_i|}} \operatorname{col}\big(\widetilde{\delta}_w(x) : x \in \mathcal{X}_i\big)$. With this normalization, the squared stacked displacement equals the example-average prediction-site mean squared displacement in Eq. 7. The same stacking convention is used for the Split and Bridge outputs on any common analysis set. When x is omitted below, $\boldsymbol{\delta}_{\mathbf{w}}$ denotes this normalized stack.

Assumption 9 (Fixed-tangent local surrogate and bounded task gradient)**.** *For the local functional-drift analysis, the residual is replaced by its affine expansion around $\mathbf{w}^t$:*

$$\boldsymbol{\delta}_{\mathbf{w}} - \boldsymbol{\delta}_{\mathbf{w}^t} = \mathbf{J}_i(\mathbf{w} - \mathbf{w}^t), \qquad \mathbf{K}_i = \mathbf{J}_i\mathbf{J}_i^\top. \tag{54}$$

The matrix $\mathbf{J}_i$ is the Jacobian of the normalized output stack evaluated at $\mathbf{w}^t$ and is fixed within the analyzed round. The surrogate starts at $\mathbf{w}_i^{t,0} = \mathbf{w}^t$ and uses a constant step size $\eta > 0$. With $\mathbf{v}_{i,h} = \boldsymbol{\delta}_{\mathbf{w}_i^{t,h}} - \boldsymbol{\delta}_{\mathbf{w}^t}$, gradient descent is applied to the surrogate task loss plus $\lambda\|\mathbf{v}_{i,h}\|_2^2/2$. The reference penalty uses the same complete analysis set at every local step. The task gradient $\mathbf{g}_{i,h}$ with respect to the stacked displacement satisfies

$$\|\mathbf{g}_{i,h}\|_2 \le G_i, \qquad 0 \le h < H_i, \tag{55}$$

for a finite constant G_i.

Justification. The affine surrogate isolates the restoring effect of the output-space reference penalty through a fixed matrix $\mathbf{K}_i$. The normalization above preserves the mean-squared penalty and its coefficient λ. Bounded task gradients then bound the forcing term in the functional recurrence. The result describes the local surrogate in Appendix C.5; changing minibatch reference sets or Jacobians requires the corresponding trajectory analysis. The nonlinear minibatch bound is developed separately in Appendix C.9.

Assumption 10 (Smooth task gradients in functional space)**.** *For the accumulated-residual result in the fixed-tangent model, let $\ell_i^\delta(\mathbf{v})$ denote the client task objective as a function of the normalized residual displacement. The local surrogate uses $\mathbf{g}_{i,h} = \nabla \ell_i^\delta(\mathbf{v}_{i,h})$. For finite $L_{i,\delta}$,*

$$\|\nabla \ell_i^\delta(\mathbf{u}) - \nabla \ell_i^\delta(\mathbf{v})\|_2 \le \ell_i^\delta \|\mathbf{u} - \mathbf{v}\|_2 \tag{56}$$

holds on a neighborhood containing $\mathbf{0}$ and the local functional trajectory.

Justification. This condition converts output displacement into a bound on the change in the task gradient. It holds for squared loss and for smooth logit losses on the analyzed output region, with constants determined by the loss and stacking normalization. Combined with Assumption 9, it gives the accumulated-residual bound in Corollary 28.

Oracle comparison. Let $\Phi : \mathbb{R}^{d_a} \to \mathbb{R}$ be a fixed-batch auxiliary objective with a fixed global reference. The subscripted sequence $\mathbf{w}_k$ below belongs to this oracle experiment; the superscripted sequence $\mathbf{w}^t$ belongs to federated training.

Assumption 11 (Regularity of the oracle objective)**.** (a) *The objective Φ is continuously differentiable on a neighborhood of each closed perturbation ball used in the analysis.* (b) *Its gradient has a finite Lipschitz constant $L_\Phi > 0$ on a convex region $\mathcal{W}_\Phi$ containing the oracle iterates, their connecting segments, and the perturbation balls:*

$$\|\nabla\Phi(\mathbf{u}) - \nabla\Phi(\mathbf{v})\|_2 \le L_\Phi\|\mathbf{u} - \mathbf{v}\|_2, \qquad \mathbf{u}, \mathbf{v} \in \mathcal{W}_\Phi. \tag{57}$$

(c) *For the oracle descent results, Φ has a finite lower bound $\Phi^\star$ and $\Delta_\Phi = \Phi(\mathbf{w}_0) - \Phi^\star < \infty$.*

Justification. Continuous differentiability on a compact perturbation neighborhood permits differentiation under the smoothing integral. The Lipschitz condition bounds finite-difference and smoothing errors and supports the descent lemma; lower boundedness is needed only when those descent inequalities are summed. Smooth auxiliary networks with smooth task losses satisfy the local regularity conditions on compact neighborhoods. The analysis conditions on a fixed objective throughout the oracle comparison.

Assumption 12 (Fresh spherical queries)**.** *Write $\mathbb{S}^{d_a-1} = \{\mathbf{u} \in \mathbb{R}^{d_a} : \|\mathbf{u}\|_2 = 1\}$ and $\mathbb{B}^{d_a} = \{\mathbf{v} \in \mathbb{R}^{d_a} : \|\mathbf{v}\|_2 \le 1\}$. Conditional on $\mathcal{F}_k^{\mathrm{or}}$, each query direction is uniform on $\mathbb{S}^{d_a-1}$ and independent of the earlier directions. When $Q \ge 1$ directions are averaged at one point, they are conditionally independent. Each two-point query evaluates the same deterministic objective Φ at $\mathbf{w} \pm \mu\mathbf{u}$, where $\mu > 0$.*

Justification. Spherical symmetry gives the smoothing identity and the moment $\mathbb{E}[\mathbf{u}\mathbf{u}^\top] = \mathbf{I}/d_a$. Independent directions yield the $1/Q$ reduction in directional variance. Holding the objective and reference fixed at both endpoints isolates the directional estimation error. The stochastic-oracle condition below covers a separate extension with oracle bias and noise.

Assumption 13 (Bounded stochastic-oracle moments)**.** *For the stochastic-oracle extension, let $\widehat{\mathbf{g}}_k$ be the update direction at $\mathbf{w}_k$. Its conditional mean and variance satisfy*

$$\mathbb{E}_k[\widehat{\mathbf{g}}_k] = \nabla\Phi(\mathbf{w}_k) + \mathbf{e}_k^{\mathrm{bias}}, \qquad \|\mathbf{e}_k^{\mathrm{bias}}\|_2^2 \le B_{\mathrm{bias}}^2, \tag{58}$$

$$\mathbb{E}_k\|\widehat{\mathbf{g}}_k - \mathbb{E}_k[\widehat{\mathbf{g}}_k]\|_2^2 \le V, \tag{59}$$

where $\mathbf{e}_k^{\mathrm{bias}}$ is $\mathcal{F}_k^{\mathrm{or}}$-measurable and B_{bias}^2, V are finite constants uniform over oracle calls.

Justification. The two moments separate systematic oracle error from centered fluctuations. Unbiased minibatch gradients give $B_{\mathrm{bias}} = 0$, while a smoothed zeroth-order oracle can have a nonzero bias. A finite variance bound is available when the oracle's second moments are uniformly bounded on the analyzed region. Lemma 23 uses only these conditional moments, so its noise may depend on the current iterate.

C.2 Training-Graph Separation and First-Order Gradient

Proof of Eq. 11. Consider a mini-batch ξ with $P_\xi = \sum_{x\in\xi} |\mathcal{P}(x)|$ supervised prediction sites. The backbone is evaluated with gradient tracking disabled. For each retained level and prediction site, the extractor produces one d_h-dimensional representation, which is detached before entering the auxiliary model. Retaining these representations therefore requires $\mathcal{O}(P_\xi M d_h)$ scalar storage.

The remaining resident weights and transient tensors required by the forward-only backbone execution are included in $M_{\mathrm{fwd}}(B, S)$. Since the retained representations carry no gradient history, none of these backbone tensors is kept for reverse-mode differentiation through the auxiliary objective.

The auxiliary residual path is evaluated with gradient tracking enabled, so its parameters, activations, gradients, optimizer states, and fixed-readout workspace contribute $M_{\mathrm{aux}}^{\mathrm{train}}(\xi)$. The fixed round reference contributes $M_{\mathrm{ref}}(\xi)$ according to the chosen reference-evaluation or caching schedule. Combining these terms gives Eq. 11.

For one example (x, y), define

$$\widetilde{f}_i^t(w; x, y) = \ell\big(\{z_{0,p}(x) + \delta_{w,p}(x)\}_{p\in\mathcal{P}(x)}, y\big) + \frac{\lambda}{2|\mathcal{P}(x)|}\sum_{p\in\mathcal{P}(x)} \big\|\delta_{w,p}(x) - \delta_{\mathrm{ref},p}^t(x)\big\|_2^2. \tag{60}$$

Let $J_{\delta,p}(x;w) = \frac{\partial \delta_{w,p}(x)}{\partial w}$. Because $z_{0,p}(x)$, $R_p(x)$, the fixed readout A, and $\delta^t_{\mathrm{ref},p}(x)$ are independent of w,

$$\nabla_w \widetilde{f}^t_i(w;x,y) = \sum_{p\in\mathcal{P}(x)} J_{\delta,p}(x;w)^\top \left[\nabla_{z_p}\ell + \frac{\lambda}{|\mathcal{P}(x)|}\big(\delta_{w,p}(x) - \delta^t_{\mathrm{ref},p}(x)\big)\right]. \tag{61}$$

For classification, $|\mathcal{P}(x)| = 1$. For autoregressive generation, both the task loss and the reference penalty are averaged over the supervised response positions of each example. Since $J_{\delta,p} = AJ_{u,p}$, reverse-mode differentiation traverses the auxiliary residual and fixed readout but not the LLM backbone.

Reverse-mode differentiation through the auxiliary model evaluates this expression. Averaging over a minibatch ξ gives $\nabla_{\mathbf{w}}\widehat{\Phi}^t_i(\mathbf{w};\xi)$; averaging over the empirical client dataset gives $\nabla_{\mathbf{w}}\widetilde{F}^t_i(\mathbf{w})$. □

C.3 Oracle-Level First-Order and Zeroth-Order Comparison

This subsection specializes standard first-order and spherical zeroth-order oracle analysis to the detached FoRA auxiliary objective. Both oracles use the same backbone representations, global-reference output, and auxiliary architecture. The comparison therefore isolates gradient-estimation and query costs after the backbone has been removed from the backward graph. We include the derivation for completeness; federated client sampling and local trajectories are treated separately in Appendix C.7.

For $\mu > 0$ and $\mathbf{u} \sim \mathrm{Unif}(\mathbb{S}^{d_a-1})$, define the standard two-point spherical estimator

$$\widehat{\mathbf{g}}_\mu(\mathbf{w};\mathbf{u}) = \frac{d_a}{2\mu}\left[\Phi(\mathbf{w}+\mu\mathbf{u}) - \Phi(\mathbf{w}-\mu\mathbf{u})\right]\mathbf{u}. \tag{62}$$

Define the ball-smoothed objective

$$\Phi_\mu(\mathbf{w}) = \mathbb{E}_{\mathbf{v}\sim\mathrm{Unif}(\mathbb{B}^{d_a})}\left[\Phi(\mathbf{w}+\mu\mathbf{v})\right]. \tag{63}$$

Lemma 14 (Spherical smoothing identity)**.** *Under Assumption 11*(a) *and the spherical query distribution in Assumption 12,*

$$\nabla\Phi_\mu(\mathbf{w}) = \frac{d_a}{\mu}\mathbb{E}_{\mathbf{u}\sim\mathrm{Unif}(\mathbb{S}^{d_a-1})}\left[\Phi(\mathbf{w}+\mu\mathbf{u})\mathbf{u}\right] = \mathbb{E}_{\mathbf{u}}\left[\widehat{\mathbf{g}}_\mu(\mathbf{w};\mathbf{u})\right]. \tag{64}$$

Proof. Let V_{d_a} be the volume of the unit ball and $A_{d_a} = d_a V_{d_a}$ the surface area of the unit sphere. Continuous differentiability on the compact perturbation neighborhood in Assumption 11(a) permits differentiation under the integral. For coordinate r,

$$\frac{\partial\Phi_\mu(\mathbf{w})}{\partial w_r} = \frac{1}{V_{d_a}}\int_{\mathbb{B}^{d_a}} \frac{\partial\Phi(\mathbf{w}+\mu\mathbf{v})}{\partial w_r}\,d\mathbf{v}. \tag{65}$$

Since $\partial\Phi(\mathbf{w}+\mu\mathbf{v})/\partial v_r = \mu\,\partial\Phi(\mathbf{w}+\mu\mathbf{v})/\partial w_r$, the divergence theorem gives

$$\frac{\partial\Phi_\mu(\mathbf{w})}{\partial w_r} = \frac{1}{\mu V_{d_a}}\int_{\mathbb{S}^{d_a-1}} \Phi(\mathbf{w}+\mu\mathbf{u})u_r\,dS(\mathbf{u}). \tag{66}$$

Using $A_{d_a} = d_a V_{d_a}$ and stacking the coordinates proves the first equality in Eq. 64. For the second equality, symmetry gives $\mathbf{u} \overset{d}{=} -\mathbf{u}$, and therefore

$$\begin{aligned}\mathbb{E}_{\mathbf{u}}\left[\widehat{\mathbf{g}}_\mu(\mathbf{w};\mathbf{u})\right] &= \frac{d_a}{2\mu}\mathbb{E}_{\mathbf{u}}\left[\Phi(\mathbf{w}+\mu\mathbf{u})\mathbf{u} - \Phi(\mathbf{w}-\mu\mathbf{u})\mathbf{u}\right] \\ &= \frac{d_a}{\mu}\mathbb{E}_{\mathbf{u}}\left[\Phi(\mathbf{w}+\mu\mathbf{u})\mathbf{u}\right] = \nabla\Phi_\mu(\mathbf{w}).\end{aligned} \tag{67}$$

□

Lemma 15 (Smoothing bias)**.** *Under Assumption 11*(a,b)*,*

$$\left\|\nabla\Phi_\mu(\mathbf{w}) - \nabla\Phi(\mathbf{w})\right\|_2 \le L_\Phi\mu. \tag{68}$$

Proof. Differentiating under the expectation in Eq. 63, followed by Jensen's inequality and Assumption 11(b), gives

$$\|\nabla\Phi_\mu(\mathbf{w}) - \nabla\Phi(\mathbf{w})\|_2 \le \mathbb{E}_{\mathbf{v}}\,\|\nabla\Phi(\mathbf{w}+\mu\mathbf{v}) - \nabla\Phi(\mathbf{w})\|_2 \le L_\Phi\mu\,\mathbb{E}_{\mathbf{v}}\|\mathbf{v}\|_2 \le L_\Phi\mu. \tag{69}$$

□

Lemma 16 (Vanishing-smoothing directional variance)**.** *Define*

$$\boldsymbol{\xi}_\mu(\mathbf{w};\mathbf{u}) = \widehat{\mathbf{g}}_\mu(\mathbf{w};\mathbf{u}) - \nabla\Phi_\mu(\mathbf{w}). \tag{70}$$

Under Assumptions 11(a,b) *and 12,*

$$\lim_{\mu\to 0}\mathbb{E}_{\mathbf{u}}\,\|\boldsymbol{\xi}_\mu(\mathbf{w};\mathbf{u})\|_2^2 = (d_a-1)\,\|\nabla\Phi(\mathbf{w})\|_2^2\,. \tag{71}$$

Proof. Let $\mathbf{g} = \nabla\Phi(\mathbf{w})$. Assumption 11(b) implies that the centered directional difference converges uniformly on the unit sphere:

$$\frac{\Phi(\mathbf{w}+\mu\mathbf{u}) - \Phi(\mathbf{w}-\mu\mathbf{u})}{2\mu} \longrightarrow \langle\mathbf{g},\mathbf{u}\rangle. \tag{72}$$

Thus $\widehat{\mathbf{g}}_\mu(\mathbf{w};\mathbf{u}) \to d_a\langle\mathbf{g},\mathbf{u}\rangle\mathbf{u}$ and, by Lemma 15, $\nabla\Phi_\mu(\mathbf{w}) \to \mathbf{g}$. Dominated convergence applies because the smoothness remainder is uniformly bounded on the compact sphere. Using $\mathbb{E}[\mathbf{u}\mathbf{u}^\top] = \mathbf{I}/d_a$ gives

$$\begin{aligned}\mathbb{E}_{\mathbf{u}}\,\|d_a\langle\mathbf{g},\mathbf{u}\rangle\mathbf{u} - \mathbf{g}\|_2^2 &= d_a^2\mathbb{E}_{\mathbf{u}}\langle\mathbf{g},\mathbf{u}\rangle^2 - 2d_a\,\mathbf{g}^\top\mathbb{E}_{\mathbf{u}}[\mathbf{u}\mathbf{u}^\top]\mathbf{g} + \|\mathbf{g}\|_2^2\\ &= d_a\|\mathbf{g}\|_2^2 - 2\|\mathbf{g}\|_2^2 + \|\mathbf{g}\|_2^2 = (d_a-1)\|\mathbf{g}\|_2^2.\end{aligned} \tag{73}$$

□

Proposition 17 (ZO oracle decomposition)**.** *Under Assumptions 11*(a,b) *and 12, use $\boldsymbol{\xi}_\mu$ from Eq. 70 and define*

$$\mathbf{e}_\mu^{\mathrm{sm}}(\mathbf{w}) = \nabla\Phi_\mu(\mathbf{w}) - \nabla\Phi(\mathbf{w}). \tag{74}$$

Then

$$\widehat{\mathbf{g}}_\mu(\mathbf{w};\mathbf{u}) = \nabla\Phi(\mathbf{w}) + \mathbf{e}_\mu^{\mathrm{sm}}(\mathbf{w}) + \boldsymbol{\xi}_\mu(\mathbf{w};\mathbf{u}), \qquad \mathbb{E}_{\mathbf{u}}[\boldsymbol{\xi}_\mu(\mathbf{w};\mathbf{u}) \mid \mathbf{w}] = \mathbf{0}. \tag{75}$$

Moreover,

$$\mathbb{E}_{\mathbf{u}}\,\|\widehat{\mathbf{g}}_\mu(\mathbf{w};\mathbf{u}) - \nabla\Phi(\mathbf{w})\|_2^2 = \|\mathbf{e}_\mu^{\mathrm{sm}}(\mathbf{w})\|_2^2 + \mathbb{E}_{\mathbf{u}}\|\boldsymbol{\xi}_\mu(\mathbf{w};\mathbf{u})\|_2^2. \tag{76}$$

The bias satisfies Eq. 68, and the directional term satisfies Eq. 71.

Proof. The algebraic decomposition follows from the definitions. Lemma 14 gives $\mathbb{E}_{\mathbf{u}}[\boldsymbol{\xi}_\mu \mid \mathbf{w}] = \mathbf{0}$. Conditional on $\mathbf{w}$, $\mathbf{e}_\mu^{\mathrm{sm}}$ is deterministic, so the cross term in $\mathbb{E}\|\mathbf{e}_\mu^{\mathrm{sm}} + \boldsymbol{\xi}_\mu\|_2^2$ vanishes. Lemmas 15 and 16 give the remaining statements. □

Corollary 18 (Direction–query trade-off)**.** *For Q directions sampled as in Assumption 12, under Assumption 11*(a,b)*, define $\widehat{\mathbf{g}}_\mu^{(Q)}(\mathbf{w}) = Q^{-1}\sum_{j=1}^{Q}\widehat{\mathbf{g}}_\mu(\mathbf{w};\mathbf{u}_j)$. Then*

$$\mathbb{E}\left\|\widehat{\mathbf{g}}_\mu^{(Q)}(\mathbf{w}) - \nabla\Phi(\mathbf{w})\right\|_2^2 = \|\mathbf{e}_\mu^{\mathrm{sm}}(\mathbf{w})\|_2^2 + \frac{1}{Q}\mathbb{E}_{\mathbf{u}}\|\boldsymbol{\xi}_\mu(\mathbf{w};\mathbf{u})\|_2^2. \tag{77}$$

In particular,

$$\lim_{\mu\to 0}\mathbb{E}\left\|\widehat{\mathbf{g}}_\mu^{(Q)}(\mathbf{w}) - \nabla\Phi(\mathbf{w})\right\|_2^2 = \frac{d_a-1}{Q}\,\|\nabla\Phi(\mathbf{w})\|_2^2\,. \tag{78}$$

For $\nabla\Phi(\mathbf{w}) \neq \mathbf{0}$, a limiting relative mean-squared error at most $\varepsilon_{\mathrm{rel}} > 0$ is ensured by $Q \ge (d_a-1)/\varepsilon_{\mathrm{rel}}$. A two-point estimate uses two objective evaluations per direction.

Proof. Define

$$\overline{\boldsymbol{\xi}}_\mu = \frac{1}{Q}\sum_{j=1}^{Q}\boldsymbol{\xi}_\mu(\mathbf{w};\mathbf{u}_j). \tag{79}$$

Conditional independence in Assumption 12 and the zero means in Proposition 17 remove the cross terms, so

$$\mathbb{E}\|\overline{\boldsymbol{\xi}}_\mu\|_2^2 = \frac{1}{Q}\mathbb{E}\|\boldsymbol{\xi}_\mu\|_2^2. \tag{80}$$

Proposition 17 proves Eq. 77. Lemma 16 then gives Eq. 78 as $\mu \to 0$. Dividing by the squared gradient norm gives the stated sufficient condition on Q. □

Lemma 19 (Second moment of the two-point estimator)**.** *Under Assumptions 11*(a,b) *and 12,*

$$\mathbb{E}_{\mathbf{u}}\,\|\widehat{\mathbf{g}}_\mu(\mathbf{w};\mathbf{u})\|_2^2 \le 2d_a\,\|\nabla\Phi(\mathbf{w})\|_2^2 + \frac{d_a^2 L_\Phi^2 \mu^2}{2}. \tag{81}$$

Proof. Let $\mathbf{g} = \nabla\Phi(\mathbf{w})$. Assumption 11(b) gives

$$\frac{\Phi(\mathbf{w}+\mu\mathbf{u}) - \Phi(\mathbf{w}-\mu\mathbf{u})}{2\mu} = \langle \mathbf{g}, \mathbf{u}\rangle + e(\mathbf{u}), \qquad |e(\mathbf{u})| \le \frac{L_\Phi \mu}{2}. \tag{82}$$

Since $\|\mathbf{u}\|_2 = 1$, the inequality $(a+b)^2 \le 2a^2 + 2b^2$ and $\mathbb{E}\langle \mathbf{g}, \mathbf{u}\rangle^2 = \|\mathbf{g}\|_2^2/d_a$ yield

$$\mathbb{E}_{\mathbf{u}}\|\widehat{\mathbf{g}}_\mu\|_2^2 = d_a^2\mathbb{E}_{\mathbf{u}}\left(\langle \mathbf{g}, \mathbf{u}\rangle + e(\mathbf{u})\right)^2 \le 2d_a^2\mathbb{E}_{\mathbf{u}}\langle \mathbf{g}, \mathbf{u}\rangle^2 + 2d_a^2\mathbb{E}_{\mathbf{u}}e(\mathbf{u})^2 \le 2d_a\|\mathbf{g}\|_2^2 + \frac{d_a^2 L_\Phi^2 \mu^2}{2}. \tag{83}$$

□

Theorem 20 (Oracle-level stationarity bounds for FO and spherical ZO)**.** *Under Assumption 11, use the initial gap Δ_Φ defined there. The spherical zeroth-order sequence also satisfies Assumption 12.*

For exact first-order updates $\mathbf{w}_{k+1} = \mathbf{w}_k - \eta\nabla\Phi(\mathbf{w}_k)$ with $0 < \eta \le 1/L_\Phi$,

$$\frac{1}{T}\sum_{k=0}^{T-1}\|\nabla\Phi(\mathbf{w}_k)\|_2^2 \le \frac{2\Delta_\Phi}{\eta T}. \tag{84}$$

For the update $\mathbf{w}_{k+1} = \mathbf{w}_k - \eta\widehat{\mathbf{g}}_\mu(\mathbf{w}_k;\mathbf{u}_k)$, choose a constant step size $0 < \eta \le 1/(4L_\Phi d_a)$. Then

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\,\|\nabla\Phi(\mathbf{w}_k)\|_2^2 \le \frac{4\Delta_\Phi}{\eta T} + 2L_\Phi^2\mu^2 + L_\Phi^3\eta d_a^2\mu^2. \tag{85}$$

In particular, $\eta = 1/(4L_\Phi d_a)$ gives

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\,\|\nabla\Phi(\mathbf{w}_k)\|_2^2 \le \frac{16L_\Phi d_a\Delta_\Phi}{T} + \left(2 + \frac{d_a}{4}\right)L_\Phi^2\mu^2. \tag{86}$$

Proof. For the first-order update, Assumption 11(b) gives

$$\Phi(\mathbf{w}_{k+1}) \le \Phi(\mathbf{w}_k) - \eta\left(1 - \frac{L_\Phi\eta}{2}\right)\|\nabla\Phi(\mathbf{w}_k)\|_2^2. \tag{87}$$

The step-size condition makes the coefficient at least $\eta/2$. Summing over k and using $\Phi(\mathbf{w}_T) \ge \Phi^\star$ proves Eq. 84.

For ZO, write $\mathbf{g}_k = \nabla\Phi(\mathbf{w}_k)$ and $\mathbf{e}_k^{\mathrm{sm}} = \nabla\Phi_\mu(\mathbf{w}_k) - \nabla\Phi(\mathbf{w}_k)$. Assumption 11(b), the fresh queries in Assumption 12, and Lemma 14 give

$$\mathbb{E}_k[\Phi(\mathbf{w}_{k+1})] \le \Phi(\mathbf{w}_k) - \eta\langle \mathbf{g}_k, \mathbf{g}_k + \mathbf{e}_k^{\mathrm{sm}}\rangle + \frac{L_\Phi\eta^2}{2}\mathbb{E}_k\|\widehat{\mathbf{g}}_\mu(\mathbf{w}_k;\mathbf{u}_k)\|_2^2. \tag{88}$$

By Young's inequality, $-\langle \mathbf{g}_k, \mathbf{e}_k^{\mathrm{sm}}\rangle \le \|\mathbf{g}_k\|_2^2/2 + \|\mathbf{e}_k^{\mathrm{sm}}\|_2^2/2$. Lemmas 15 and 19 therefore yield

$$\mathbb{E}_k[\Phi(\mathbf{w}_{k+1})] \le \Phi(\mathbf{w}_k) - \left(\frac{\eta}{2} - L_\Phi\eta^2 d_a\right)\|\mathbf{g}_k\|_2^2 + \frac{\eta L_\Phi^2\mu^2}{2} + \frac{L_\Phi^3\eta^2 d_a^2\mu^2}{4}. \tag{89}$$

The ZO step-size condition gives $\eta/2 - L_\Phi\eta^2 d_a \ge \eta/4$. Summing, taking total expectation, and dividing by $\eta T/4$ proves Eq. 85. Substitution of $\eta = 1/(4L_\Phi d_a)$ proves Eq. 86. □

Corollary 21 (Sufficient oracle complexities for stationarity)**.** *Consider the stationarity target*

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\|\nabla\Phi(\mathbf{w}_k)\|_2^2 \le \varepsilon. \tag{90}$$

The first-order bound in Theorem 20 with $\eta = 1/L_\Phi$ is satisfied by

$$T_{\mathrm{FO}} \ge \frac{2L_\Phi\Delta_\Phi}{\varepsilon}. \tag{91}$$

For the single-direction ZO bound, choose $\eta = 1/(4L_\Phi d_a)$ and

$$\mu^2 \le \frac{2\varepsilon}{L_\Phi^2(d_a+8)}. \tag{92}$$

It is then sufficient that

$$T_{\mathrm{ZO}} \ge \frac{32L_\Phi d_a\Delta_\Phi}{\varepsilon}. \tag{93}$$

These are worst-case sufficient upper bounds for the specified oracles.

Proof. Setting $\eta = 1/L_\Phi$ in Eq. 84 gives the first claim. For ZO, Eq. 92 makes the smoothing term in Eq. 86 at most $\varepsilon/2$, and Eq. 93 makes its optimization term at most $\varepsilon/2$. □

Remark 22 (Architecture-controlled time-to-target condition)**.** *For an architecture-controlled comparison using the same detached backbone features, global-reference output, and auxiliary model, let C_{back} be one backbone forward cost, C_{ref} one reference-model forward cost, and C^F_{aux} and C^B_{aux} the auxiliary forward and backward costs. One FO update costs*

$$C_{\mathrm{FO}} = C_{\mathrm{back}} + C_{\mathrm{ref}} + C^F_{\mathrm{aux}} + C^B_{\mathrm{aux}}. \tag{94}$$

If a ZO update averages Q two-point directions while reusing the same detached features and reference output, its cost is

$$C_{\mathrm{ZO}} = C_{\mathrm{back}} + C_{\mathrm{ref}} + 2Q\,C^F_{\mathrm{aux}}. \tag{95}$$

Writing $N_{\mathrm{FO}}(\varepsilon)$ and $N_{\mathrm{ZO}}(\varepsilon)$ for the updates needed to reach a target, FO has lower wall-clock time under the measurable condition

$$N_{\mathrm{FO}}(\varepsilon)C_{\mathrm{FO}} < N_{\mathrm{ZO}}(\varepsilon)C_{\mathrm{ZO}}. \tag{96}$$

Lemma 23 (Biased stochastic-oracle descent)**.** *Under Assumptions 11*(b,c) *and 13, consider the updates*

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \eta\widehat{\mathbf{g}}_k, \qquad 0 < \eta \le 1/(4L_\Phi). \tag{97}$$

Then

$$\frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\,\|\nabla\Phi(\mathbf{w}_k)\|_2^2 \le \frac{4\Delta_\Phi}{\eta T} + 3B^2_{\mathrm{bias}} + 2L_\Phi\eta V. \tag{98}$$

Proof. Let $\mathbf{g}_k = \nabla\Phi(\mathbf{w}_k)$. Conditional smoothness from Assumption 11(b) and the mean identity in Assumption 13 give

$$\mathbb{E}_k[\Phi(\mathbf{w}_{k+1})] \le \Phi(\mathbf{w}_k) - \eta\langle\mathbf{g}_k, \mathbf{g}_k + \mathbf{e}_k^{\mathrm{bias}}\rangle + \frac{L_\Phi\eta^2}{2}\mathbb{E}_k\|\widehat{\mathbf{g}}_k\|_2^2. \tag{99}$$

Assumption 13, variance decomposition, and $\|\mathbf{a}+\mathbf{b}\|_2^2 \le 2\|\mathbf{a}\|_2^2 + 2\|\mathbf{b}\|_2^2$ give $\mathbb{E}_k\|\widehat{\mathbf{g}}_k\|_2^2 \le 2\|\mathbf{g}_k\|_2^2 + 2B^2_{\mathrm{bias}} + V$. Young's inequality gives $-\langle\mathbf{g}_k, \mathbf{e}_k^{\mathrm{bias}}\rangle \le \|\mathbf{g}_k\|_2^2/2 + B^2_{\mathrm{bias}}/2$. Hence

$$\mathbb{E}_k[\Phi(\mathbf{w}_{k+1})] \le \Phi(\mathbf{w}_k) - \left(\frac{\eta}{2} - L_\Phi\eta^2\right)\|\mathbf{g}_k\|_2^2 + \left(\frac{\eta}{2} + L_\Phi\eta^2\right)B^2_{\mathrm{bias}} + \frac{L_\Phi\eta^2}{2}V. \tag{100}$$

Under the stated step size, the gradient coefficient is at least $\eta/4$ and the bias coefficient is at most $3\eta/4$. Summing and dividing by $\eta T/4$ proves Eq. 98. □

An unbiased minibatch FO oracle has $B_{\mathrm{bias}} = 0$ in Lemma 23; its remaining stochastic term is data-sampling variance. Standard spherical ZO adds the smoothing and directional terms described above. Appendix C.7 incorporates client sampling, non-IID data, and local trajectories.

C.4 Block-Structured Functional Geometry of the Split-Bridge Parameterization

Fix a client, communication round, and finite analysis set $\mathcal{X}$. Use the normalized stacking convention in Appendix C.1. All Jacobian identities below are evaluated at a common parameter point and use the differentiability and disjoint-parameter conditions stated there. The subscript FoRA denotes the complete FoRA residual in this subsection. Let $\boldsymbol{\gamma} = (\boldsymbol{\psi}, \mathbf{q})$ collect the *Bridge* parameters, and write

$$\mathbf{d}(\boldsymbol{\theta}, \boldsymbol{\gamma}) = \mathbf{d}_{\mathrm{s}}(\boldsymbol{\theta}) + \beta \mathbf{d}_{\mathrm{b}}(\boldsymbol{\gamma}). \tag{101}$$

At a fixed parameter point $(\overline{\boldsymbol{\theta}}, \overline{\boldsymbol{\gamma}})$, define

$$\mathbf{J}_{\mathrm{s}} = \frac{\partial \mathbf{d}_{\mathrm{s}}}{\partial \boldsymbol{\theta}}, \qquad \mathbf{J}_{\mathrm{b}} = \frac{\partial \mathbf{d}_{\mathrm{b}}}{\partial \boldsymbol{\gamma}}. \tag{102}$$

Both Jacobians in Eq. 102 are evaluated at $(\overline{\boldsymbol{\theta}}, \overline{\boldsymbol{\gamma}})$.

Folded realization. The Split-Bridge parameterization admits an equivalent folded representation at any fixed parameter state. The folding argument is applied to the correction-space function u_w. Because the task readout A is fixed, any folded realization of u_w induces the same task-output residual Au_w. For each adapter, absorb the affine LayerNorm parameters into its first projection and let $\mathbf{h}_{\boldsymbol{\omega}}(\mathbf{r}) \in \mathbb{R}^{d_b}$ denote the resulting hidden activation. At representation level j, define

$$\bar{\mathbf{h}}_j(\mathbf{r}) = \mathrm{col}\big(\mathbf{h}_{\boldsymbol{\theta}_j}(\mathbf{r}), \mathbf{h}_{\boldsymbol{\psi}_{1,j}}(\mathbf{r}), \ldots, \mathbf{h}_{\boldsymbol{\psi}_{K,j}}(\mathbf{r})\big)\,, \tag{103}$$

and

$$\bar{\mathbf{W}}_{2,j} = \big[\mathbf{W}_{2,\boldsymbol{\theta}_j}, \beta\pi_1(\mathbf{q})\mathbf{W}_{2,\boldsymbol{\psi}_{1,j}}, \ldots, \beta\pi_K(\mathbf{q})\mathbf{W}_{2,\boldsymbol{\psi}_{K,j}}\big]\,, \quad \bar{\mathbf{b}}_{2,j} = \mathbf{b}_{2,\boldsymbol{\theta}_j} + \beta \sum_{k=1}^{K} \pi_k(\mathbf{q}) \mathbf{b}_{2,\boldsymbol{\psi}_{k,j}}. \tag{104}$$

The level-j residual can then be written as

$$\mathbf{a}_{\boldsymbol{\theta}_j}(\mathbf{r}) + \beta \sum_{k=1}^{K} \pi_k(\mathbf{q}) \mathbf{a}_{\boldsymbol{\psi}_{k,j}}(\mathbf{r}) = \bar{\mathbf{W}}_{2,j} \bar{\mathbf{h}}_j(\mathbf{r}) + \bar{\mathbf{b}}_{2,j}, \tag{105}$$

which is a bottleneck adapter of hidden width $(K+1)d_b$. Thus, every fixed Split-Bridge state has a corresponding folded wide-adapter realization.

Let $\mathcal{T}$ denote the map from the factorized FoRA parameters to their folded representation. Because $\mathcal{T}$ contains products such as $\pi_k(\mathbf{q})\mathbf{W}_{2,\boldsymbol{\psi}_{k,j}}$, it is nonlinear in the trainable coordinates. Hence, for client states $\{\mathbf{w}_i\}$ and aggregation weights $\{\omega_i\}$,

$$\mathcal{T}\left(\sum_i \omega_i \mathbf{w}_i\right) \neq \sum_i \omega_i \mathcal{T}(\mathbf{w}_i) \tag{106}$$

in general. FoRA performs local optimization and FedAvg directly in the factorized coordinates $\mathbf{w} = (\boldsymbol{\theta}, \boldsymbol{\psi}, \mathbf{q})$. The Jacobian analysis below characterizes the local functional geometry induced by these coordinates.

Proof of Eq. 12. Differentiating the stacked residual over the disjoint *Split* and *Bridge* parameter blocks gives

$$\mathbf{J}_{\mathrm{FoRA}} = [\mathbf{J}_{\mathrm{s}},\ \beta \mathbf{J}_{\mathrm{b}}]. \tag{107}$$

Therefore

$$\begin{aligned} \mathbf{K}_{\mathrm{FoRA}} &= \mathbf{J}_{\mathrm{FoRA}} \mathbf{J}_{\mathrm{FoRA}}^{\top} \\ &= \mathbf{J}_{\mathrm{s}} \mathbf{J}_{\mathrm{s}}^{\top} + \beta^2 \mathbf{J}_{\mathrm{b}} \mathbf{J}_{\mathrm{b}}^{\top} = \mathbf{K}_{\mathrm{s}} + \beta^2 \mathbf{K}_{\mathrm{b}}. \end{aligned} \tag{108}$$

For every vector $\mathbf{a}$,

$$\mathbf{a}^{\top}(\mathbf{K}_{\mathrm{FoRA}} - \mathbf{K}_{\mathrm{s}})\mathbf{a} = \beta^2 \left\|\mathbf{J}_{\mathrm{b}}^{\top}\mathbf{a}\right\|_2^2 \geq 0. \tag{109}$$

Hence $\mathbf{K}_{\mathrm{FoRA}} \succeq \mathbf{K}_{\mathrm{s}}$. Eigenvalue monotonicity under the Loewner order, equivalently the Courant–Fischer characterization, gives

$$\lambda_r(\mathbf{K}_{\mathrm{FoRA}}) \geq \lambda_r(\mathbf{K}_{\mathrm{s}}) \tag{110}$$

for every consistently ordered eigenvalue. □

Lemma 24 (Tangent-space relation to the Split-only submodel)**.** *Define*

$$\mathcal{S}_{\mathrm{s}} = \mathrm{Range}(\mathbf{J}_{\mathrm{s}}), \qquad \mathcal{S}_{\mathrm{FoRA}} = \mathrm{Range}([\mathbf{J}_{\mathrm{s}}, \beta\mathbf{J}_{\mathrm{b}}])\,. \tag{111}$$

Then $\mathcal{S}_{\mathrm{s}} \subseteq \mathcal{S}_{\mathrm{FoRA}}$. *Let* $\mathbf{P}_{\mathrm{s}}$ *be the orthogonal projector onto* $\mathcal{S}_{\mathrm{s}}$. *If* $\beta \neq 0$*, then*

$$\mathrm{rank}([\mathbf{J}_{\mathrm{s}}, \beta\mathbf{J}_{\mathrm{b}}]) = \mathrm{rank}(\mathbf{J}_{\mathrm{s}}) + \mathrm{rank}((\mathbf{I} - \mathbf{P}_{\mathrm{s}})\mathbf{J}_{\mathrm{b}})\,. \tag{112}$$

For $\beta \neq 0$*, the inclusion is strict exactly when* $(\mathbf{I} - \mathbf{P}_{\mathrm{s}})\mathbf{J}_{\mathrm{b}} \neq \mathbf{0}$.

Proof. Every $\mathbf{J}_{\mathrm{s}}\mathbf{a} \in \mathcal{S}_{\mathrm{s}}$ can be written as $[\mathbf{J}_{\mathrm{s}}, \beta\mathbf{J}_{\mathrm{b}}][\mathbf{a}^\top, \mathbf{0}^\top]^\top$, proving the inclusion. For $\beta \neq 0$, decompose

$$\mathbf{J}_{\mathrm{b}} = \mathbf{P}_{\mathrm{s}}\mathbf{J}_{\mathrm{b}} + (\mathbf{I} - \mathbf{P}_{\mathrm{s}})\mathbf{J}_{\mathrm{b}}. \tag{113}$$

The first term lies in $\mathcal{S}_{\mathrm{s}}$ and the second lies in its orthogonal complement. Thus

$$\mathrm{Range}([\mathbf{J}_{\mathrm{s}}, \mathbf{J}_{\mathrm{b}}]) = \mathcal{S}_{\mathrm{s}} \oplus \mathrm{Range}((\mathbf{I} - \mathbf{P}_{\mathrm{s}})\mathbf{J}_{\mathrm{b}})\,. \tag{114}$$

Taking dimensions proves Eq. 112. Multiplication by nonzero β preserves the *Bridge* column space, and the direct-sum term is nonzero exactly under the stated strictness condition. □

Lemma 25 (Bridge mixture Jacobian and kernel decomposition)**.** *For the stacked Bridge output, write*

$$\mathbf{d}_{\mathrm{b}}(\boldsymbol{\psi}, \mathbf{q}) = \sum_{k=1}^{K} \pi_k(\mathbf{q})\mathbf{d}_k(\boldsymbol{\psi}_k), \qquad \boldsymbol{\pi}(\mathbf{q}) = \mathrm{softmax}(\mathbf{q}), \tag{115}$$

where $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_K$ *and* $\mathbf{q}$ *are disjoint parameter blocks. Let* $\mathbf{J}_k = \partial\mathbf{d}_k/\partial\boldsymbol{\psi}_k$ *and* $\mathbf{J}_q = \partial\mathbf{d}_{\mathrm{b}}/\partial\mathbf{q}$. *Then*

$$\mathbf{J}_{\mathrm{b}} = [\pi_1\mathbf{J}_1, \ldots, \pi_K\mathbf{J}_K, \mathbf{J}_q]\,, \tag{116}$$

$$\mathbf{K}_{\mathrm{b}} = \sum_{k=1}^{K} \pi_k^2 \mathbf{J}_k\mathbf{J}_k^\top + \mathbf{J}_q\mathbf{J}_q^\top. \tag{117}$$

For coordinate r *of* $\mathbf{q}$*,*

$$\frac{\partial\mathbf{d}_{\mathrm{b}}}{\partial q_r} = \pi_r\,(\mathbf{d}_r - \mathbf{d}_{\mathrm{b}})\,. \tag{118}$$

Proof. Only component k depends on $\boldsymbol{\psi}_k$, so $\partial\mathbf{d}_{\mathrm{b}}/\partial\boldsymbol{\psi}_k = \pi_k\mathbf{J}_k$. The softmax identity $\partial\pi_k/\partial q_r = \pi_k(\mathbf{1}\{k = r\} - \pi_r)$ gives

$$\frac{\partial\mathbf{d}_{\mathrm{b}}}{\partial q_r} = \sum_{k=1}^{K} \frac{\partial\pi_k}{\partial q_r}\mathbf{d}_k = \pi_r\mathbf{d}_r - \pi_r\sum_{k=1}^{K}\pi_k\mathbf{d}_k = \pi_r(\mathbf{d}_r - \mathbf{d}_{\mathrm{b}}). \tag{119}$$

Concatenating the parameter-block Jacobians proves Eq. 116. Since the blocks occupy disjoint columns, multiplying $\mathbf{J}_{\mathrm{b}}\mathbf{J}_{\mathrm{b}}^\top$ gives Eq. 117. □

Lemma 26 (Residual initialization and first-order gradients)**.** *Suppose the Split and Bridge output maps are initialized so that* $\mathbf{d}_{\mathrm{s}} = \mathbf{0}$ *and* $\mathbf{d}_k = \mathbf{0}$ *for all* k. *Let* $\mathcal{L}$ *be the task loss on the stacked residual and* $\mathbf{g}$ *its output gradient. Then*

$$\nabla_{\boldsymbol{\theta}}\mathcal{L} = \mathbf{J}_{\mathrm{s}}^\top\mathbf{g}, \qquad \nabla_{\boldsymbol{\psi}_k}\mathcal{L} = \beta\pi_k\mathbf{J}_k^\top\mathbf{g}. \tag{120}$$

At this initialization, $\mathbf{J}_q = \mathbf{0}$. *The mixture-logit Jacobian can become nonzero after the component outputs become distinct.*

Proof. The first two identities follow directly from the chain rule and Eq. 116. Equation 118 gives $\partial\mathbf{d}_{\mathrm{b}}/\partial q_r = \pi_r(\mathbf{0} - \mathbf{0}) = \mathbf{0}$ when every component output is zero. Once two component outputs differ, the same expression can be nonzero. The component-gradient identities hold at this initialization, while the mixture-logit gradient initially vanishes. Whether a component gradient is nonzero depends on its Jacobian and the task gradient. □

Proposition 27 (First-step gradient decomposition)**.** *At the beginning of a communication round, the local model equals the global parameters, so the gradient of the global-reference regularizer is zero. Let $\mathcal{L}$ denote the task loss and $\mathbf{g}$ its gradient with respect to the stacked residual. Then*

$$\left\|\nabla_{(\boldsymbol{\theta},\boldsymbol{\gamma})}\mathcal{L}\right\|_2^2 = \left\|\mathbf{J}_{\mathrm{s}}^\top \mathbf{g}\right\|_2^2 + \beta^2 \left\|\mathbf{J}_{\mathrm{b}}^\top \mathbf{g}\right\|_2^2. \tag{121}$$

The added term is positive exactly when $\beta \mathbf{J}_{\mathrm{b}}^\top \mathbf{g} \neq \mathbf{0}$.

Proof. At the round initialization, the current and reference residuals coincide, so the squared reference term has zero gradient. The task gradients are $\nabla_{\boldsymbol{\theta}}\mathcal{L} = \mathbf{J}_{\mathrm{s}}^\top \mathbf{g}$ and $\nabla_{\boldsymbol{\gamma}}\mathcal{L} = \beta \mathbf{J}_{\mathrm{b}}^\top \mathbf{g}$. The parameter blocks are disjoint Euclidean coordinates, so their squared norms add. □

Consequences of the *Split*-only inclusion. Since $\mathcal{S}_{\mathrm{s}} \subseteq \mathcal{S}_{\mathrm{FoRA}}$, optimization over the full FoRA tangent space cannot yield a larger optimum than over the *Split*-only subspace:

$$\inf_{\mathbf{v}\in\mathcal{S}_{\mathrm{FoRA}}} \mathcal{J}(\mathbf{v}) \leq \inf_{\mathbf{v}\in\mathcal{S}_{\mathrm{s}}} \mathcal{J}(\mathbf{v}). \tag{122}$$

For the quadratic residual surrogate, the gain is determined by the projection of the target residual onto the additional tangent directions in $\mathcal{S}_{\mathrm{FoRA}}$. Likewise, because the complete residual function class contains the *Split*-only class, its best achievable approximation error cannot be larger. Under a fixed positive-definite functional kernel, the standard gradient-descent contraction rate is governed by the extreme eigenvalues of that kernel.

These results characterize the local geometry of the factorized Split-Bridge coordinates and their Split-only restriction. The rank and surrogate-objective comparisons depend on *Bridge* Jacobian components outside the *Split* tangent space and on their alignment with the local task signal. A fixed Split-Bridge state also admits the folded wide-adapter realization described above; the analysis in this subsection concerns the parameter coordinates used by local optimization and federated aggregation.

C.5 Global-Reference Local Dynamics under a Fixed Tangent

This subsection isolates the local effect of the global-reference term under the fixed-tangent model in Assumption 9. The analysis keeps the residual Jacobian fixed within one communication round and studies how the output-space reference changes the within-round functional trajectory. The nonlinear federated analysis in Appendices C.7-C.9 does not require a fixed Jacobian.

For $\mathbf{K}_i \neq \mathbf{0}$, let $\kappa_i^+ = \lambda_{\max}(\mathbf{K}_i)$ and $\kappa_i^- = \lambda_{\min}^+(\mathbf{K}_i)$. The functional dynamics lie in $\mathrm{Range}(\mathbf{K}_i)$. When $\mathbf{K}_i = \mathbf{0}$, the auxiliary parameters produce no first-order functional movement on the analysis set, and the statements below are immediate.

Using the displacement $\mathbf{v}_{i,h}$ and task gradient $\mathbf{g}_{i,h}$ from Assumption 9, gradient descent on the surrogate gives

$$\mathbf{v}_{i,h+1} = (\mathbf{I} - \eta\lambda\mathbf{K}_i)\mathbf{v}_{i,h} - \eta\mathbf{K}_i\mathbf{g}_{i,h}, \qquad \mathbf{v}_{i,0} = \mathbf{0}, \tag{123}$$

where η denotes the local step size in this subsection.

Proof of Eq. 14. By Assumption 9, $\mathbf{J}_i$ is constant and $\mathbf{v}_{i,0} = \mathbf{0}$. For $\lambda > 0$ and $0 < \eta\lambda\kappa_i^+ \leq 1$, the eigenvalues of $\mathbf{I} - \eta\lambda\mathbf{K}_i$ restricted to $\mathrm{Range}(\mathbf{K}_i)$ lie in $[0, 1 - \eta\lambda\kappa_i^-]$. Hence

$$\|(\mathbf{I} - \eta\lambda\mathbf{K}_i)\mathbf{v}\|_2 \leq (1 - \eta\lambda\kappa_i^-)\|\mathbf{v}\|_2 \tag{124}$$

for every $\mathbf{v} \in \mathrm{Range}(\mathbf{K}_i)$. Combining Eq. 123 with the gradient bound in Assumption 9 and $\|\mathbf{K}_i\|_2 = \kappa_i^+$ gives

$$\|\mathbf{v}_{i,h+1}\|_2 \leq (1 - \eta\lambda\kappa_i^-)\|\mathbf{v}_{i,h}\|_2 + \eta\kappa_i^+ G_i. \tag{125}$$

At the round initialization,

$$R_i^t(w^t) = 0, \qquad \nabla R_i^t(w^t) = 0. \tag{126}$$

Under Assumption 9,

$$R_{i,\tan}^t(w) = \frac{1}{2}\|J_i(w - w^t)\|_2^2 = \frac{1}{2}(w - w^t)^\top J_i^\top J_i(w - w^t). \tag{127}$$

Let $\widetilde{J}_i$ denote the Jacobian of the stacked correction-space outputs before the fixed readout. Since the final residual is $\delta = \mathbf{A}u$, the corresponding task-output Jacobian satisfies

$$J_i = \mathbf{A}_{\mathcal{X}_i} \widetilde{J}_i, \tag{128}$$

where $\mathbf{A}_{\mathcal{X}_i}$ denotes the block-diagonal application of A over the stacked prediction sites. Hence

$$J_i^\top J_i = \widetilde{J}_i^\top \, \mathbf{A}_{\mathcal{X}_i}^\top \mathbf{A}_{\mathcal{X}_i} \widetilde{J}_i. \tag{129}$$

For classification, $\mathbf{A} = \mathbf{I}_C$. For generation, this metric contains the Gram matrix of the frozen language-model head.

Thus, the functional reference induces the local parameter-space metric $J_i^\top J_i$. Let $v_{i,h} = \delta_{w_i^{t,h}} - \delta_{w^t}$ and $K_i = J_i J_i^\top$. Gradient descent on the surrogate gives

$$\|\mathbf{v}_{i,h}\|_2 \le \eta \kappa_i^+ G_i \sum_{r=0}^{h-1} (1 - \eta\lambda\kappa_i^-)^r = \frac{\kappa_i^+ G_i}{\lambda \kappa_i^-} \left[1 - (1 - \eta\lambda\kappa_i^-)^h\right], \tag{130}$$

which proves Eq. 14. For $\lambda = 0$, Eq. 123 becomes $\mathbf{v}_{i,h+1} = \mathbf{v}_{i,h} - \eta \mathbf{K}_i \mathbf{g}_{i,h}$. The triangle inequality gives $\|\mathbf{v}_{i,h+1}\|_2 \le \|\mathbf{v}_{i,h}\|_2 + \eta\kappa_i^+ G_i$, and unrolling proves the $\lambda = 0$ branch of Eq. 14. □

The next result links the functional displacement to the accumulated local-update residual used in the federated convergence theorem.

Corollary 28 (Accumulated residual in the fixed-tangent model)**.** *Under Assumptions 9 and 10, define the accumulated parameter-gradient residual*

$$\mathbf{r}_{i,\tan}^t = \sum_{h=0}^{H_i - 1} \mathbf{J}_i^\top \left[\nabla \ell_i^\delta(\mathbf{v}_{i,h}) - \nabla \ell_i^\delta(\mathbf{0}) + \lambda \mathbf{v}_{i,h}\right]. \tag{131}$$

For $\lambda > 0$ and $0 < \eta\lambda\kappa_i^+ \le 1$,

$$\|\mathbf{r}_{i,\tan}^t\|_2 \le \Gamma_{i,\tan}(\lambda, H_i) := \sqrt{\kappa_i^+}(L_{i,\delta} + \lambda) \frac{\kappa_i^+ G_i}{\lambda \kappa_i^-} \times \left[H_i - \frac{1 - (1 - \eta\lambda\kappa_i^-)^{H_i}}{\eta\lambda\kappa_i^-}\right]. \tag{132}$$

For $\lambda = 0$,

$$\|\mathbf{r}_{i,\tan}^t\|_2 \le \Gamma_{i,\tan}(0, H_i) := \eta L_{i,\delta} (\kappa_i^+)^{3/2} G_i \frac{H_i(H_i - 1)}{2}. \tag{133}$$

With the step size and regularity constants fixed, the positive-λ bound grows at most linearly in H_i, while the zero-λ bound is quadratic in H_i. The finite-H_i values also depend on λ and the kernel spectrum.

Proof. The operator norm satisfies $\|\mathbf{J}_i^\top\|_2 = \sqrt{\kappa_i^+}$. Assumption 10 gives

$$\left\|\nabla \ell_i^\delta(\mathbf{v}_{i,h}) - \nabla \ell_i^\delta(\mathbf{0}) + \lambda \mathbf{v}_{i,h}\right\|_2 \le (L_{i,\delta} + \lambda)\|\mathbf{v}_{i,h}\|_2. \tag{134}$$

Combining Eqs. 131 and 134 gives

$$\|\mathbf{r}_{i,\tan}^t\|_2 \le \sqrt{\kappa_i^+}(L_{i,\delta} + \lambda) \sum_{h=0}^{H_i - 1} \|\mathbf{v}_{i,h}\|_2. \tag{135}$$

For $\lambda > 0$, substitute Eq. 14 and use

$$\sum_{h=0}^{H_i - 1} \left[1 - (1 - \eta\lambda\kappa_i^-)^h\right] = H_i - \frac{1 - (1 - \eta\lambda\kappa_i^-)^{H_i}}{\eta\lambda\kappa_i^-}. \tag{136}$$

This proves Eq. 132. For $\lambda = 0$, substitute $\|\mathbf{v}_{i,h}\|_2 \le \eta h \kappa_i^+ G_i$ into Eq. 135 and use $\sum_{h=0}^{H_i - 1} h = H_i(H_i - 1)/2$. □

Connection to the federated trajectory term. The fixed-tangent result gives an explicit instantiation of the local-trajectory quantity used by the federated convergence theorem. Define

$$B^2_{\tan}(\lambda) = \mathbb{E}_S\left[\sum_{i\in S}\omega_i(S)\Gamma^2_{i,\tan}(\lambda, H_i)\right]. \tag{137}$$

If the task-gradient bounds, functional smoothness constants, and relevant kernel factors are uniformly bounded across the analyzed rounds, then the fixed-tangent federated dynamics satisfy

$$B^2_{\mathrm{traj}}(\lambda) \le B^2_{\tan}(\lambda). \tag{138}$$

For fixed $\lambda > 0$ and fixed regularity constants,

$$B^2_{\tan}(\lambda) = O\left(\mathbb{E}_S\left[\sum_{i\in S}\omega_i(S)H_i^2\right]\right), \tag{139}$$

whereas

$$B^2_{\tan}(0) = O\left(\mathbb{E}_S\left[\sum_{i\in S}\omega_i(S)H_i^4\right]\right). \tag{140}$$

Thus, within the fixed-tangent regime, the change from linear to quadratic growth in the client-level trajectory bound propagates directly to the trajectory-error term in the federated stationarity guarantee.

For federated training that uses this same affine local surrogate, a round-uniform choice of Γ_i additionally requires uniform bounds on the task gradients, functional smoothness, and relevant kernel factors across the analyzed rounds. Appendix C.9 supplies a separate nonlinear minibatch bound. The two results describe different scopes of the local-trajectory analysis.

C.6 Effective Federated Objective and Sampling Lemmas

Use the distribution and weights in Assumption 1 and Eqs. 36–38. The following identities explain why $\widetilde{F}$ describes the expected round-start direction. Client heterogeneity enters its variance through Assumption 3(a).

Lemma 29 (Expected round-start direction)**.** *Under Assumption 1, for a fixed* $\mathbf{w}$*, define*

$$\mathbf{G}_{\mathcal{S}}(\mathbf{w}) = \sum_{i\in\mathcal{S}}\omega_i(\mathcal{S})H_i\nabla F_i(\mathbf{w}), \tag{141}$$

we have

$$\mathbb{E}_{\mathcal{S}}[\mathbf{G}_{\mathcal{S}}(\mathbf{w})] = C_H\nabla\widetilde{F}(\mathbf{w}). \tag{142}$$

Proof. The fixed sampling law in Assumption 1 and linearity of expectation give

$$\begin{aligned}\mathbb{E}_{\mathcal{S}}[\mathbf{G}_{\mathcal{S}}(\mathbf{w})] &= \sum_{i=1}^N \mathbb{E}_{\mathcal{S}}[\mathbf{1}\{i\in\mathcal{S}\}\omega_i(\mathcal{S})H_i]\,\nabla F_i(\mathbf{w}) \\ &= \sum_{i=1}^N c_i\nabla F_i(\mathbf{w}) = C_H\sum_{i=1}^N \widetilde{p}_i\nabla F_i(\mathbf{w}) = C_H\nabla\widetilde{F}(\mathbf{w}).\end{aligned} \tag{143}$$

□

Lemma 30 (Variance of the sampled round-start direction)**.** *Under Assumptions 1 and 3*(a)*, at every point covered by the heterogeneity bound,*

$$\mathbb{E}_{\mathcal{S}}\left\|\mathbf{G}_{\mathcal{S}}(\mathbf{w}) - C_H\nabla\widetilde{F}(\mathbf{w})\right\|_2^2 \le 2\nu_H^2\left\|\nabla\widetilde{F}(\mathbf{w})\right\|_2^2 + 2H_{\max}C_H\zeta^2. \tag{144}$$

Proof. Let $\mathbf{g} = \nabla\widetilde{F}(\mathbf{w})$, $\mathbf{h}_i = \nabla F_i(\mathbf{w}) - \mathbf{g}$, and $a_i(\mathcal{S}) = \mathbf{1}\{i\in\mathcal{S}\}\omega_i(\mathcal{S})H_i$. Then

$$\mathbf{G}_{\mathcal{S}}(\mathbf{w}) - C_H\mathbf{g} = (A_{\mathcal{S}} - C_H)\mathbf{g} + \sum_{i=1}^N a_i(\mathcal{S})\mathbf{h}_i. \tag{145}$$

Using $\|\mathbf{x}+\mathbf{y}\|_2^2 \le 2\|\mathbf{x}\|_2^2 + 2\|\mathbf{y}\|_2^2$ gives

$$\mathbb{E}_{\mathcal{S}}\|\mathbf{G}_{\mathcal{S}} - C_H\mathbf{g}\|_2^2 \le 2\nu_H^2\|\mathbf{g}\|_2^2 + 2\mathbb{E}_{\mathcal{S}}\left\|\sum_i a_i(\mathcal{S})\mathbf{h}_i\right\|_2^2. \tag{146}$$

All $a_i(\mathcal{S})$ are nonnegative and $\sum_i a_i(\mathcal{S}) = A_{\mathcal{S}} \le H_{\max}$. Weighted Cauchy–Schwarz yields

$$\left\|\sum_i a_i(\mathcal{S})\mathbf{h}_i\right\|_2^2 \le A_{\mathcal{S}}\sum_i a_i(\mathcal{S})\|\mathbf{h}_i\|_2^2 \le H_{\max}\sum_i a_i(\mathcal{S})\|\mathbf{h}_i\|_2^2. \tag{147}$$

Taking expectation and using Assumption 3(a) gives

$$\mathbb{E}_{\mathcal{S}}\left\|\sum_i a_i(\mathcal{S})\mathbf{h}_i\right\|_2^2 \le H_{\max}\sum_i c_i\|\mathbf{h}_i\|_2^2 = H_{\max}C_H\sum_i \widetilde{p}_i\|\mathbf{h}_i\|_2^2 \le H_{\max}C_H\zeta^2. \tag{148}$$

Substitution proves Eq. 144. □

C.7 Federated Convergence under Non-IID Data

Appendix C.5 isolates the reference-induced local dynamics under a fixed tangent. We now analyze the full federated protocol while keeping the accumulated local-trajectory residual abstract. This yields a stationarity bound that applies whenever the local residual satisfies Assumption 6; the fixed-tangent and nonlinear minibatch analyses provide two concrete instantiations.

Under Assumption 1, summing the local-SGD steps and using the errors defined in Eqs. 43-44 gives

$$\mathbf{w}_i^{t,H_i} = \mathbf{w}^t - \eta\left[H_i\nabla F_i(\mathbf{w}^t) + \boldsymbol{\xi}_i^t + \mathbf{r}_i^t\right]. \tag{149}$$

Lemma 33 gives the full algebraic derivation. The initial gap and aggregate residual measure are defined in Eqs. 40 and 50. The argument below uses $\mathbb{E}_t$ before client selection; $\mathcal{S}_t$ remains random inside this expectation.

Theorem 31 (Stationarity of the heterogeneous federated protocol)**.** *Under Assumptions 1, 2, 3*(a)*, 4, and 6, let the common constant local step size satisfy*

$$0 < \eta \le \frac{C_H}{6L_{\widetilde{F}}(C_H^2 + 4\nu_H^2)}, \tag{150}$$

then

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla\widetilde{F}(\mathbf{w}^t)\right\|_2^2 \le \frac{2\Delta_{\widetilde{F}}}{\eta C_H T} + \left(\frac{2}{C_H^2} + \frac{3L_{\widetilde{F}}\eta}{C_H}\right)\mathcal{B}_{\mathrm{traj}}^2(\lambda) + 12L_{\widetilde{F}}\eta H_{\max}\zeta^2 + \frac{6L_{\widetilde{F}}\eta}{C_H}V_{\mathrm{rr}}. \tag{151}$$

Proof. Aggregating Eq. 149 gives

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta\left[\mathbf{G}_{\mathcal{S}_t}(\mathbf{w}^t) + \boldsymbol{\xi}_{\mathrm{rr}}^t + \mathbf{r}_t\right], \tag{152}$$

where

$$\boldsymbol{\xi}_{\mathrm{rr}}^t = \sum_{i\in\mathcal{S}_t}\omega_i(\mathcal{S}_t)\boldsymbol{\xi}_i^t, \qquad \mathbf{r}_t = \sum_{i\in\mathcal{S}_t}\omega_i(\mathcal{S}_t)\mathbf{r}_i^t. \tag{153}$$

Let $\mathbf{g}_t = \nabla\widetilde{F}(\mathbf{w}^t)$ and define

$$\boldsymbol{\xi}_t = \mathbf{G}_{\mathcal{S}_t}(\mathbf{w}^t) - C_H\mathbf{g}_t + \boldsymbol{\xi}_{\mathrm{rr}}^t. \tag{154}$$

By Assumption 1, Lemma 29 applies conditional on $\mathcal{H}_t$. Together with Assumption 4, it gives $\mathbb{E}_t[\boldsymbol{\xi}_t] = \mathbf{0}$. Lemma 30, the noise second moment in Assumption 4, and $\|\mathbf{a}+\mathbf{b}\|_2^2 \le 2\|\mathbf{a}\|_2^2 + 2\|\mathbf{b}\|_2^2$ give

$$\mathbb{E}_t\|\boldsymbol{\xi}_t\|_2^2 \le 4\nu_H^2\|\mathbf{g}_t\|_2^2 + 4H_{\max}C_H\zeta^2 + 2V_{\mathrm{rr}}. \tag{155}$$

Assumption 6, convexity of the squared norm, and averaging over $\mathcal{S}_t$ give

$$\mathbb{E}_t\|\mathbf{r}_t\|_2^2 \le \mathcal{B}_{\mathrm{traj}}^2(\lambda). \tag{156}$$

Assumption 2 applies on the segment from $\mathbf{w}^t$ to $\mathbf{w}^{t+1}$ and gives

$$\mathbb{E}_t[\widetilde{F}(\mathbf{w}^{t+1})] \le \widetilde{F}(\mathbf{w}^t) - \eta C_H \|\mathbf{g}_t\|_2^2 - \eta\langle \mathbf{g}_t, \mathbb{E}_t[\mathbf{r}_t]\rangle + \frac{L_{\widetilde{F}}\eta^2}{2}\mathbb{E}_t\|C_H\mathbf{g}_t + \boldsymbol{\xi}_t + \mathbf{r}_t\|_2^2. \tag{157}$$

Young's inequality gives

$$-\langle \mathbf{g}_t, \mathbb{E}_t[\mathbf{r}_t]\rangle \le \frac{C_H}{4}\|\mathbf{g}_t\|_2^2 + \frac{1}{C_H}\mathbb{E}_t\|\mathbf{r}_t\|_2^2. \tag{158}$$

Using $\|\mathbf{a}+\mathbf{b}+\mathbf{c}\|_2^2 \le 3(\|\mathbf{a}\|_2^2 + \|\mathbf{b}\|_2^2 + \|\mathbf{c}\|_2^2)$ yields

$$\begin{aligned}\mathbb{E}_t[\widetilde{F}(\mathbf{w}^{t+1})] \le \widetilde{F}(\mathbf{w}^t) &- \left[\frac{3\eta C_H}{4} - \frac{3L_{\widetilde{F}}\eta^2}{2}(C_H^2 + 4\nu_H^2)\right]\|\mathbf{g}_t\|_2^2 \\ &+ \left(\frac{\eta}{C_H} + \frac{3L_{\widetilde{F}}\eta^2}{2}\right)\mathcal{B}_{\mathrm{traj}}^2(\lambda) + 6L_{\widetilde{F}}\eta^2 H_{\max} C_H \zeta^2 + 3L_{\widetilde{F}}\eta^2 V_{\mathrm{rr}}.\end{aligned} \tag{159}$$

The step-size condition ensures

$$\frac{3L_{\widetilde{F}}\eta^2}{2}(C_H^2 + 4\nu_H^2) \le \frac{\eta C_H}{4}. \tag{160}$$

The descent coefficient is therefore at least $\eta C_H/2$. Rearranging, taking total expectation, and using the lower bound in Assumption 2 after summing over t gives

$$\begin{aligned}\frac{\eta C_H}{2}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla\widetilde{F}(\mathbf{w}^t)\|_2^2 \le \Delta_{\widetilde{F}} &+ T\left(\frac{\eta}{C_H} + \frac{3L_{\widetilde{F}}\eta^2}{2}\right)\mathcal{B}_{\mathrm{traj}}^2(\lambda) \\ &+ 6L_{\widetilde{F}}\eta^2 T H_{\max} C_H\zeta^2 + 3L_{\widetilde{F}}\eta^2 T V_{\mathrm{rr}}.\end{aligned} \tag{161}$$

Dividing by $\eta C_H T/2$ proves Eq. 151. □

Derivation of Eq. 15. Apply Theorem 31. For the fixed federated protocol, define

$$\begin{aligned}\eta_{\max} &= \frac{C_H}{6L_{\widetilde{F}}(C_H^2 + 4\nu_H^2)}, \\ C_{\mathrm{th}} &= \max\left\{\frac{2}{C_H}, \frac{2}{C_H^2} + \frac{3L_{\widetilde{F}}\eta_{\max}}{C_H}, 12L_{\widetilde{F}}, \frac{6L_{\widetilde{F}}}{C_H}\right\}.\end{aligned} \tag{162}$$

For every $0 < \eta \le \eta_{\max}$, Eq. 151 is bounded term by term by C_{th} times the quantity inside $\mathcal{O}(\cdot)$ in Eq. 15. This establishes Eq. 15. □

Theorem 31 separates the effect of the local trajectory from client sampling, statistical heterogeneity, and minibatch noise. Substituting the fixed-tangent bound from Appendix C.5 makes the dependence on global-reference regularization explicit. Appendix C.9 provides a second specialization for the full nonlinear minibatch trajectory.

C.8 Gap to the Intended Sample-Weighted Objective

Recall $F(\mathbf{w}) = \sum_i p_i F_i(\mathbf{w})$ with $p_i = n_i/\sum_j n_j$. Define

$$\chi_p^2 = \sum_{i=1}^N \frac{(\widetilde{p}_i - p_i)^2}{p_i}. \tag{163}$$

The required sample-weighted heterogeneity condition is Assumption 3(b).

Corollary 32 (Gap to the sample-weighted objective)**.** *For the effective weights in Eq. 37, under Assumption 3*(b)*,*

$$\left\|\nabla\widetilde{F}(\mathbf{w}) - \nabla F(\mathbf{w})\right\|_2^2 \le \chi_p^2\zeta_F^2. \tag{164}$$

Under full participation, $\widetilde{p}_i = p_i H_i/\overline{H}$ with $\overline{H} = \sum_i p_i H_i$, and

$$\chi_p^2 = \sum_{i=1}^N p_i\left(\frac{H_i}{\overline{H}} - 1\right)^2. \tag{165}$$

Proof. Since $\sum_i \widetilde{p}_i = \sum_i p_i = 1$,

$$\nabla\widetilde{F}(\mathbf{w}) - \nabla F(\mathbf{w}) = \sum_i (\widetilde{p}_i - p_i)\nabla F_i(\mathbf{w}) = \sum_i (\widetilde{p}_i - p_i)\left[\nabla F_i(\mathbf{w}) - \nabla F(\mathbf{w})\right]. \tag{166}$$

Weighted Cauchy–Schwarz and Assumption 3(b) give

$$\begin{aligned}\left\|\nabla\widetilde{F}(\mathbf{w}) - \nabla F(\mathbf{w})\right\|_2^2 &\le \left[\sum_i \frac{(\widetilde{p}_i - p_i)^2}{p_i}\right]\left[\sum_i p_i\|\nabla F_i(\mathbf{w}) - \nabla F(\mathbf{w})\|_2^2\right] \\ &\le \chi_p^2\zeta_F^2.\end{aligned} \tag{167}$$

Under full participation, $\omega_i = p_i$, so $c_i = p_i H_i$, $C_H = \overline{H}$, and $\widetilde{p}_i = p_i H_i/\overline{H}$. Substitution into Eq. 163 proves Eq. 165. □

The stationarity transfer follows from

$$\|\nabla F(\mathbf{w})\|_2^2 \le 2\|\nabla\widetilde{F}(\mathbf{w})\|_2^2 + 2\chi_p^2\zeta_F^2. \tag{168}$$

Combining this inequality with Eq. 15 gives a stationarity neighborhood for F. The gap is zero when the effective and intended weights coincide, and also when all client gradients agree.

C.9 Direct Minibatch Random-Reshuffling Specialization

This subsection provides a complementary specialization of the federated trajectory term. Unlike the fixed-tangent analysis in Appendix C.5, the auxiliary Jacobian and regularized minibatch gradients may vary along the local trajectory. The resulting bound is therefore more conservative, but applies directly to the nonlinear local-SGD updates used by FoRA.

Assumption 5 controls the nonlinear trajectory, while Assumption 7 specifies random reshuffling and the finite-population sampling variance. The fixed-tangent analysis isolates the mechanism induced by the reference term; the present analysis supplies a protocol-level bound without freezing the local Jacobian. The step index r runs from 0 to $H_i - 1$ across the E_i local epochs.

Use $\widehat{\Phi}_{i,r}^t$ and the local update in Eqs. 32 and 34. In this specialization, write $\boldsymbol{\xi}_{i,\mathrm{mb}}^t = \boldsymbol{\xi}_i^t$ and $\mathbf{r}_{i,\mathrm{mb}}^t = \mathbf{r}_i^t$ for the errors defined in Eqs. 43–44.

Because the current and global-reference residuals coincide at $\mathbf{w}^t$,

$$\nabla\widehat{R}_{i,r}^t(\mathbf{w}^t) = \mathbf{0}. \tag{169}$$

Lemma 33 (Exact minibatch endpoint decomposition)**.** *Under the local-SGD protocol in Assumption 1, the errors defined above satisfy*

$$\mathbf{w}_i^{t,H_i} = \mathbf{w}^t - \eta\left[H_i\nabla F_i(\mathbf{w}^t) + \boldsymbol{\xi}_{i,\mathrm{mb}}^t + \mathbf{r}_{i,\mathrm{mb}}^t\right]. \tag{170}$$

Under Assumption 5, they also satisfy

$$\|\mathbf{r}_{i,\mathrm{mb}}^t\|_2 \le \Gamma_{i,\mathrm{mb}} := \eta L_{i,\mathrm{mb}} G_{i,\mathrm{mb}} \frac{H_i(H_i-1)}{2}. \tag{171}$$

Proof. Summing Eq. 34 gives

$$\mathbf{w}_i^{t,H_i} = \mathbf{w}^t - \eta\sum_{r=0}^{H_i-1}\nabla\widehat{\Phi}_{i,r}^t(\mathbf{w}_i^{t,r}). \tag{172}$$

Add and subtract $\nabla\widehat{\Phi}_{i,r}^t(\mathbf{w}^t)$ in every summand. Equation 169 gives

$$\nabla\widehat{\Phi}_{i,r}^t(\mathbf{w}^t) = \nabla\widehat{F}_{i,r}^t(\mathbf{w}^t). \tag{173}$$

Adding and subtracting $\nabla F_i(\mathbf{w}^t)$ proves Eq. 170.

For the bound, Eq. 34 and the gradient-norm condition in Assumption 5 give

$$\|\mathbf{w}_i^{t,r} - \mathbf{w}^t\|_2 \le \eta r G_{i,\mathrm{mb}}. \tag{174}$$

Using the smoothness condition in Assumption 5 in Eq. 44,

$$\begin{aligned}\|\mathbf{r}_{i,\mathrm{mb}}^{t}\|_2 &\le \sum_{r=0}^{H_i-1} L_{i,\mathrm{mb}}\|\mathbf{w}_i^{t,r}-\mathbf{w}^t\|_2 \\ &\le \eta L_{i,\mathrm{mb}}G_{i,\mathrm{mb}}\sum_{r=0}^{H_i-1} r = \eta L_{i,\mathrm{mb}}G_{i,\mathrm{mb}}\frac{H_i(H_i-1)}{2}.\end{aligned} \tag{175}$$

□

We next use the example gradients defined before Assumption 7. For a fixed $\mathbf{w}$, let ϖ be one uniform epoch permutation and let $\mathcal{B}_{i,h}(\varpi)$ be its h-th minibatch. Define

$$\widehat{\mathbf{g}}_{i,h}(\mathbf{w};\varpi) = \frac{1}{B_{i,h}}\sum_{j\in\mathcal{B}_{i,h}(\varpi)} \mathbf{g}_{i,j}(\mathbf{w}), \tag{176}$$

$$\boldsymbol{\xi}_{i,\mathrm{ep}}(\mathbf{w};\varpi) = \sum_{h=1}^{H_i^{\mathrm{ep}}} \widehat{\mathbf{g}}_{i,h}(\mathbf{w};\varpi) - H_i^{\mathrm{ep}}\nabla F_i(\mathbf{w}). \tag{177}$$

Lemma 34 (Mean of the random-reshuffling error)**.** *Under the uniform-permutation and arithmetic-mean conditions in Assumption 7, for every fixed* $\mathbf{w}$*, including an incomplete final minibatch,*

$$\mathbb{E}_{\varpi}\left[\boldsymbol{\xi}_{i,\mathrm{ep}}(\mathbf{w};\varpi)\right] = \mathbf{0}. \tag{178}$$

Applying this identity at $\mathbf{w}^t$ *yields* $\mathbb{E}_{t,\mathcal{S}}[\boldsymbol{\xi}_{i,\mathrm{mb}}^t] = \mathbf{0}$ *for every selected client, and hence the aggregate centering condition in Assumption 4.*

Proof. Under Assumption 7, every permutation position has the uniform marginal distribution over the n_i examples. Hence each minibatch average satisfies $\mathbb{E}_{\varpi}[\widehat{\mathbf{g}}_{i,h}(\mathbf{w};\varpi)] = \nabla F_i(\mathbf{w})$ for any positive batch size $B_{i,h}$. Summing over h proves Eq. 178. The epoch errors are all evaluated at the same fixed round model. Summing their conditional means and applying the tower property gives the round-wise claim. □

The finite-population variance is $\sigma_i^2(\mathbf{w})$ from Eq. 51.

Lemma 35 (Exact one-epoch reshuffling variance)**.** *Under the uniform-permutation and arithmetic-mean conditions in Assumption 7, for* $n_i > 1$*,*

$$\mathbb{E}_{\varpi}\left\|\boldsymbol{\xi}_{i,\mathrm{ep}}(\mathbf{w};\varpi)\right\|_2^2 = \frac{n_i}{n_i-1}\left(\sum_{h=1}^{H_i^{\mathrm{ep}}}\frac{1}{B_{i,h}} - \frac{(H_i^{\mathrm{ep}})^2}{n_i}\right)\sigma_i^2(\mathbf{w}). \tag{179}$$

For $n_i = 1$*, the error is zero. The variance is also zero when all minibatches in the epoch have the same size.*

Proof. Let $\mathbf{z}_j = \mathbf{g}_{i,j}(\mathbf{w}) - \nabla F_i(\mathbf{w})$, so $\sum_{j=1}^{n_i}\mathbf{z}_j = \mathbf{0}$. Index permutation positions by $s = 1,\dots,n_i$. If position s belongs to minibatch h, set $a_s = 1/B_{i,h}$. Then

$$\boldsymbol{\xi}_{i,\mathrm{ep}}(\mathbf{w};\varpi) = \sum_{s=1}^{n_i} a_s\mathbf{z}_{\varpi(s)}. \tag{180}$$

Since $\sum_s a_s = H_i^{\mathrm{ep}}$ and $\sum_j \mathbf{z}_j = \mathbf{0}$, define $c_s = a_s - H_i^{\mathrm{ep}}/n_i$ without changing the sum. Then $\sum_s c_s = 0$.

The uniform permutation in Assumption 7 gives

$$\mathbb{E}[\mathbf{z}_{\varpi(s)}\mathbf{z}_{\varpi(s)}^{\top}] = \frac{1}{n_i}\sum_{j=1}^{n_i}\mathbf{z}_j\mathbf{z}_j^{\top}, \tag{181}$$

$$\mathbb{E}[\mathbf{z}_{\varpi(s)}\mathbf{z}_{\varpi(r)}^\top] = -\frac{1}{n_i(n_i-1)}\sum_{j=1}^{n_i}\mathbf{z}_j\mathbf{z}_j^\top, \qquad s \neq r. \tag{182}$$

Expanding the squared norm and using $\sum_s c_s = 0$ gives

$$\mathbb{E}\left\|\sum_{s=1}^{n_i} c_s \mathbf{z}_{\varpi(s)}\right\|_2^2 = \frac{1}{n_i-1}\left(\sum_{s=1}^{n_i} c_s^2\right)\sum_{j=1}^{n_i}\|\mathbf{z}_j\|_2^2. \tag{183}$$

A direct calculation yields

$$\sum_{s=1}^{n_i} c_s^2 = \sum_{h=1}^{H_i^{\mathrm{ep}}}\frac{1}{B_{i,h}} - \frac{(H_i^{\mathrm{ep}})^2}{n_i}. \tag{184}$$

Finally, $\sum_j \|\mathbf{z}_j\|_2^2 = n_i\sigma_i^2(\mathbf{w})$, which proves Eq. 179. Equal batch sizes give $B_{i,h} = n_i/H_i^{\mathrm{ep}}$ and make the parenthesized factor zero. □

Independence across epochs in Assumption 7 and its uniform variance bound give the following conditional upper bound for client i. For $n_i > 1$, define

$$V_{i,\mathrm{rr}} = E_i\frac{n_i}{n_i-1}\left(\sum_{h=1}^{H_i^{\mathrm{ep}}}\frac{1}{B_{i,h}} - \frac{(H_i^{\mathrm{ep}})^2}{n_i}\right)\overline{\sigma}_i^2, \tag{185}$$

with $V_{i,\mathrm{rr}} = 0$ for $n_i = 1$. Define

$$\mathcal{B}_{\mathrm{mb}}^2 = \mathbb{E}_{\mathcal{S}}\left[\sum_{i\in\mathcal{S}}\omega_i(\mathcal{S})\Gamma_{i,\mathrm{mb}}^2\right], \tag{186}$$

$$V_{\mathrm{mb}} = \mathbb{E}_{\mathcal{S}}\left[\sum_{i\in\mathcal{S}}\omega_i(\mathcal{S})^2 V_{i,\mathrm{rr}}\right]. \tag{187}$$

For this specialization, take

$$\Gamma_i(\lambda, H_i) = \Gamma_{i,\mathrm{mb}}, \qquad B_{\mathrm{traj}}^2(\lambda) = B_{\mathrm{mb}}^2. \tag{188}$$

Corollary 36 (Stationarity of minibatch FoRA under random reshuffling)**.** *Under Assumptions 1, 2, 3*(a)*, 5, and 7, with the step-size condition in Eq. 150,*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla\widetilde{F}(\mathbf{w}^t)\right\|_2^2 \le \frac{2\Delta_{\widetilde{F}}}{\eta C_H T} + \left(\frac{2}{C_H^2} + \frac{3L_{\widetilde{F}}\eta}{C_H}\right)\mathcal{B}_{\mathrm{mb}}^2 + 12L_{\widetilde{F}}\eta H_{\max}\zeta^2 + \frac{6L_{\widetilde{F}}\eta}{C_H}V_{\mathrm{mb}}. \tag{189}$$

Proof. Lemma 33 verifies Assumption 6 with $\Gamma_i = \Gamma_{i,\mathrm{mb}}$. Conditional convexity of the squared norm and Assumption 1 give

$$\mathbb{E}_t\left\|\sum_{i\in\mathcal{S}_t}\omega_i(\mathcal{S}_t)\mathbf{r}_{i,\mathrm{mb}}^t\right\|_2^2 \le \mathcal{B}_{\mathrm{mb}}^2. \tag{190}$$

Lemma 34 supplies zero conditional means given $(\mathcal{H}_t, \mathcal{S}_t)$. Independence across clients from Assumption 7 removes cross-client terms. The tower property and Lemma 35 then give

$$\begin{aligned}\mathbb{E}_t\left\|\sum_{i\in\mathcal{S}_t}\omega_i(\mathcal{S}_t)\boldsymbol{\xi}_{i,\mathrm{mb}}^t\right\|_2^2 &= \mathbb{E}_t\left[\sum_{i\in\mathcal{S}_t}\omega_i(\mathcal{S}_t)^2\mathbb{E}_{t,\mathcal{S}}\|\boldsymbol{\xi}_{i,\mathrm{mb}}^t\|_2^2\right] \\ &\le \mathbb{E}_{\mathcal{S}}\left[\sum_{i\in\mathcal{S}}\omega_i(\mathcal{S})^2 V_{i,\mathrm{rr}}\right] = V_{\mathrm{mb}}.\end{aligned} \tag{191}$$

These calculations verify Assumption 4 with $V_{\mathrm{rr}} = V_{\mathrm{mb}}$ and Eq. 50 with $\mathcal{B}_{\mathrm{traj}}^2(\lambda) = \mathcal{B}_{\mathrm{mb}}^2$. Theorem 31 therefore gives the claimed bound. □

Remark 37 (Independent minibatch sampling)**.** *Replacing Assumption 7 by Assumption 8, while retaining the other conditions of Corollary 36, gives the same stationarity bound with*

$$V_{\mathrm{mb}} = \mathbb{E}_{\mathcal{S}}\left[\sum_{i\in\mathcal{S}} \omega_i(\mathcal{S})^2 \frac{H_i \overline{\sigma}_i^2}{B_i}\right]. \tag{192}$$

Indeed, the fixed-point errors are conditionally centered, and independence gives the sum of H_i per-minibatch variance bounds on each client. A second variance sum across clients, followed by averaging over $\mathcal{S}_t$, verifies Assumption 4. The nonlinear residual bound remains Lemma 33.

D Evaluation Details

This section provides additional details on the datasets, federated partitions, model configurations, baseline implementations, and evaluation protocol. Within each model–task configuration, all methods use the same backbone checkpoint, tokenizer, client partition, client-selection schedule, local batch size, and evaluation data.

D.1 Datasets and Preprocessing

MRPC. The Microsoft Research Paraphrase Corpus (MRPC) is a binary sentence-pair classification task from GLUE (Wang et al., 2019). Each example contains two sentences and a label indicating whether they convey the same meaning. We use the released split with 3,668 training examples and 408 validation examples, tokenize the two sentences jointly, and treat the paraphrase label as the positive class. We report both Accuracy and positive-class F1.

SST-2. SST-2 is a binary sentiment-classification task from GLUE (Wang et al., 2019). It contains 67,349 training examples and 872 validation examples, with each sentence labeled as positive or negative. We tokenize each sentence independently and use positive sentiment as the positive class. Its larger training set complements MRPC and is used to study FoRA at a larger model scale.

Dolly-15K. Databricks Dolly-15K contains 15,011 human-written instruction-following examples across eight task categories (Conover et al., 2023). Each record contains an instruction, an optional context, a reference response, and a category. We construct the model input from the instruction and nonempty context using a fixed prompt template, and use the response as the causal-language-modeling target. Prompt and padding positions are masked with label -100. During training, token losses are averaged over the supervised response positions within each example and then averaged across minibatch examples. Since the released dataset contains a single split, we construct a fixed held-out evaluation set of $n_{\mathrm{eval}} = 1{,}487$ examples and reuse the same example indices across methods. Perplexity is computed over all supervised response tokens from the complete held-out set, while autoregressive ROUGE-L uses the same fixed subset of $n_{\mathrm{gen}} = 500$ examples. Generation uses a maximum prompt length of 256 tokens and at most 256 newly generated tokens.

Common preprocessing. Within each task, all methods use the same tokenizer, truncation rule, label mapping, prompt template, and training/evaluation split. The partition indices and client-selection schedules are saved before training and reused across paired runs. Evaluation examples remain centralized only for model assessment and do not participate in client optimization. For MRPC and SST-2, the released validation split serves as the reporting split; no separate labeled test split is used in our experiments.

D.2 Federated Data Partition and Training Protocol

Statistical heterogeneity. We construct non-IID client datasets with a group-wise Dirichlet partition (Zeng et al., 2023). A group corresponds to a class label for MRPC and SST-2 and to the released category field for Dolly-15K. For each group g, we sample

$$\mathbf{a}_g^{\mathrm{part}} \sim \mathrm{Dirichlet}(\alpha_{\mathrm{Dir}} \mathbf{1}_N), \tag{193}$$

where $\mathbf{a}_g^{\mathrm{part}} \in \mathbb{R}^N$ specifies the proportions assigned to the N clients. Examples within each group are shuffled and distributed according to these proportions. The resulting client subsets are disjoint and jointly cover the complete training split. Unless stated otherwise, we use $\alpha_{\mathrm{Dir}} = 0.7$; smaller values are evaluated in the heterogeneity study.

Table 7: Task-specific federated training budgets. The round budget, batch size, local-epoch count, and learning-rate scaling are shared by all methods within each model-task configuration. The scale is applied to each method's base learning rate.

Configuration	Global rounds	B	E	LR scale
OPT-350M@MRPC	200	4	2	$1.0\times$
OPT-2.7B@SST-2	50	4	2	$1.0\times$
Llama-3.1-8B@Dolly-15K	50	4	2	$0.1\times$

Client participation and local training. The main experiments use $N = 10$ clients. At communication round t, the server samples $m_t = 5$ clients uniformly without replacement, corresponding to a participation rate of 50%. Every selected client initializes from the current global model and performs $E = 2$ local epochs with mini-batch size $B = 4$. Client i therefore performs

$$H_i = E \left\lceil \frac{n_i}{B} \right\rceil \tag{194}$$

local optimizer steps when the final incomplete mini-batch is retained. Differences in client dataset size can therefore produce different values of H_i even under the same local-epoch budget.

Task-specific training budgets. We use a common training budget for all methods within each model-task configuration. OPT-350M@MRPC uses 200 global rounds. OPT-2.7B@SST-2 is limited to 50 global rounds because SST-2 contains substantially more training examples and therefore incurs a larger local-computation cost per round. Llama-3.1-8B@Dolly-15K is also limited to 50 global rounds because of the larger backbone and autoregressive training cost. These limits are fixed at the configuration level and are shared by all compared methods.

Aggregation. After local training, the server aggregates the complete trainable state using sample-weighted FedAvg (McMahan et al., 2017). For client $i \in \mathcal{S}_t$, the aggregation weight is

$$\omega_i^t = \frac{n_i}{\sum_{j \in \mathcal{S}_t} n_j}. \tag{195}$$

For FoRA, the communicated state contains the Split parameters $\boldsymbol{\theta}$, Bridge parameters $\boldsymbol{\psi}$, and mixture logits $\mathbf{q}$. The LLM backbone remains fixed and is not included in the federated payload.

Controlled comparisons. Within a given model-task configuration, all methods use the same backbone checkpoint, tokenizer, training and evaluation splits, client partition, sampled-client sequence, mini-batch size, local-epoch count, global-round limit, and hardware allocation. Method-specific optimizers and parameterizations follow their respective algorithms.

Repetitions. Main classification experiments are repeated with three random seeds, and results are reported as mean $\pm$ standard deviation. Each seed determines parameter initialization, client partitioning, local data order, and client participation. The same seed-specific partition and client schedule are reused across methods. Results from generation experiments follow the same protocol.

D.3 Models and Optimization Configurations

OPT classification models. We use OPT-350M for MRPC and OPT-2.7B for SST-2 (Zhang et al., 2022). Both are decoder-only Transformer models equipped with a binary sequence-classification output. The checkpoint, tokenizer, label ordering, and classification architecture are shared by all methods within each task.

Llama generation model. We use Llama-3.1-8B (Team, 2024) for Dolly-15K (Conover et al., 2023). The model is trained with the causal language-modeling objective on supervised response tokens. Owing to its scale, the backbone is partitioned across two GPUs in a single process. All trainable FoRA auxiliary parameters receive standard first-order gradients, and the Llama backbone remains frozen.

FoRA configuration. The local learning rate is $\eta^{\mathrm{loc}} = 2 \times 10^{-4}$ for the OPT configurations and 2×10^{-5} for Llama-3.1-8B, following the scaling in Table 7. Local client optimization uses minibatch SGD, with one optimizer step per local minibatch. Each block adapter has bottleneck dimension $d_b = 64$. We use $K = 4$ *Bridge* banks, *Bridge* scale $\beta = 0.5$, and global-reference coefficient $\lambda = 7 \times 10^{-5}$.

Representation extraction. For a backbone containing L Transformer layers, FoRA retains the output of each consecutive block of b layers, giving $M = \lceil L/b \rceil$ representation levels. Classification experiments use $b = 2$ and last-token pooling, where the selected token is the last non-padding position according to the attention mask. The Llama-3.1-8B configuration uses $b = 4$ to reduce the auxiliary cost at the larger backbone scale. The extracted representations are detached before entering the trainable *Split-Bridge* residual.

Autoregressive realization of the residual interface. For Dolly-15K, the correction space uses $d_c = d_h$, with the frozen language-model head $A = W_{\mathrm{LM}}$ as the fixed readout. We define $\mathcal{P}(x)$ as the supervised causal prediction sites whose shifted target labels differ from the ignore index -100. At each $p \in \mathcal{P}(x)$, the Split-Bridge model produces the hidden-space correction $u_w(R_p(x))$, which is mapped to vocabulary-logit space by $\delta_{w,p}(x) = W_{\mathrm{LM}} u_w(R_p(x))$.

For $\Delta u_{x,p} = u_w(R_p(x)) - u_{w^t}(R_p(x)), \quad G_{\mathrm{LM}} = W_{\mathrm{LM}}^\top W_{\mathrm{LM}}$, the corresponding logit-space displacement satisfies $\|W_{\mathrm{LM}} \Delta u_{x,p}\|_2^2 = \Delta u_{x,p}^\top G_{\mathrm{LM}} \Delta u_{x,p}$. For a minibatch ξ, the reference penalty is $\widehat{R}_i^t(w;\xi) = \frac{1}{2|\xi|} \sum_{x \in \xi} \frac{1}{|\mathcal{P}(x)|} \sum_{p \in \mathcal{P}(x)} \Delta u_{x,p}^\top G_{\mathrm{LM}} \Delta u_{x,p}$. This is the empirical form of Eq. 7, evaluated through the Gram matrix without materializing vocabulary-dimensional residuals.

Execution environment. Experiments are implemented in PyTorch and executed on a workstation with an Intel Gold 6278C CPU and two NVIDIA Tesla V100-PCIE GPUs with 32 GiB of memory each. All reported runs use FP32 arithmetic. OPT experiments use the same single-GPU allocation for all compared methods, and Llama-3.1-8B experiments use the same two-GPU model-sharding policy. Client resource profiles are homogeneous within the main experiments. The current evaluation focuses on statistical heterogeneity under homogeneous client hardware; heterogeneous devices and network conditions are outside the scope of the reported system measurements.

D.4 Baseline Methods

Baseline tuning protocol.

For each baseline, we tune only the base learning rate while fixing the remaining method-specific hyperparameters to the settings recommended in the original paper or the defaults in the official implementation. Candidate learning-rate grids and selected values are listed below; selection follows the task's primary Final-10 metric. The selected base learning rate is fixed before the three-seed main runs, and the Llama-3.1-8B setting applies the common $0.1\times$ scaling.

FedIT (Zhang et al., 2023). FedIT is the standard first-order federated LoRA reference. Each selected client updates its LoRA parameters on local data and uploads the resulting adapter state for sample-weighted aggregation. For classification, the task-specific output parameters follow the same trainable scope across paired LoRA experiments. We use LoRA rank $r = 16$, scaling factor $\alpha = 32$, and dropout 0.1. We tune the base learning rate over $\{1 \times 10^{-6}, 5 \times 10^{-6}, 1 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}\}$ and select 1×10^{-5}.

FedMeZO (Ling et al., 2024). FedMeZO applies MeZO-style finite-difference optimization on each selected client and aggregates the resulting local parameter states. Local parameters are updated after every zeroth-order estimate, so each client follows its own sequence of probe points during the local epoch. We use the trainable parameter scope of the reproduced implementation, with the random-sampling step count set to 128, $n_{\mathrm{perturb}} = 2$, and a scalar-clipping threshold of 20.0. We tune the base learning rate of zeroth-order perturbation over $\{1 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}, 5 \times 10^{-3}, 1 \times 10^{-2}\}$ and select 1×10^{-3}.

RAVAN (Raje et al., 2025). RAVAN represents each adaptation update with multiple low-rank heads and trains lightweight head transformations and scaling parameters. We preserve its multi-head parameterization, trainable-budget rule, and federated aggregation procedure. The underlying LoRA target modules are matched to the common LoRA configuration where applicable. We use four heads and initialize the scaling factors to 1.0, following the multi-head parameterization and trainable-budget rule of RAVAN. We tune the base learning rate over $\{1 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}\}$ and select 5×10^{-4}.

GMFL (Huang et al., 2026). GMFL monitors the global update magnitude of LoRA layers and periodically constructs masks that freeze layers with small updates. We follow its global

masking schedule and retain the same base LoRA parameterization used by the other LoRA-based methods. We apply the common base LoRA parameterization and update the masking state every 20 communication rounds. The progressive freezing schedule starts from an initial freezing ratio of 0.30 and increases the ratio by 0.05 at each mask update. We use a minimum learning rate of 1×10^{-7} and weight decay 0.005. We tune the base learning rate over $\{1 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}\}$ and select 5×10^{-4}.

FedProxy (Fan et al., 2026). FedProxy constructs a compressed proxy language model from the target LLM, performs federated optimization on the proxy, and integrates the learned update into the target model through its fusion procedure. We use the proxy construction, aggregation, and fusion settings of the reproduced implementation. In our reproduced implementation, we use a compression ratio of 0.30 and Stanford Alpaca as the public dataset for proxy construction, with at most $5{,}000$ public samples. The PCR regularization coefficient is 1×10^{-5}. For H-TIES, we use a base retention rate $r_0 = 1.0$, heterogeneity penalty $\delta = 0.2$, consensus reward $\rho = 1.1$, and numerical stability constant $\epsilon = 10^{-8}$. We tune the base learning rate over $\{1 \times 10^{-6}, 5 \times 10^{-6}, 1 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}\}$ and select 5×10^{-6}.

Fairness controls. All methods within a model-task configuration use the same backbone checkpoint, tokenizer, data partition, sampled-client sequence, local epoch count, batch size, evaluation checkpoints, and hardware allocation. LoRA-based methods use a common base LoRA configuration where their parameterizations permit direct matching. Algorithm-specific components, including zeroth-order perturbations, multi-head adaptation, global masks, and proxy fusion, follow the corresponding method definitions. *Backbone BP* indicates whether gradients traverse the target LLM backbone.

D.5 Relation to Side-Tuning and Fed MobiLLM

FoRA shares the frozen-representation principle of side-network adaptation, while its federated design specifies a different location for residual optimization and a different communication interface. We compare FoRA with Ladder Side-Tuning (LST) and Fed MobiLLM along these dimensions, covering the centralized, server-assisted, and client-resident settings.

Ladder Side-Tuning (LST) provides the architectural precedent for adapting a frozen backbone through intermediate representations. LST combines multi-level backbone features with a trainable side network while keeping the backbone outside the adaptation path. This establishes the representation-side separation that FoRA also adopts.

Fed MobiLLM extends the side-network paradigm to federated LLM adaptation through server-assisted training. Clients execute the backbone locally and transmit intermediate activations and output residuals, while the shared side network is maintained and optimized on the server under an asynchronous protocol. Its design further accommodates heterogeneous client backbones through activation sampling and feature alignment.

FoRA places the trainable residual on participating clients and treats its auxiliary parameters as the federated state. Clients optimize the Split-Bridge residual with first-order gradients over detached backbone representations, then upload only the auxiliary parameters for sample-weighted FedAvg. Backbone activations remain local, and the server performs aggregation without gradient-based training. The broadcast auxiliary state also provides the functional reference for local updates, coupling the communication state with within-round drift control.

Thus, the three methods share the principle of separating adaptation from the backbone, but differ in optimization location, communication interface, and federated state. Table 8 summarizes these protocol-level differences. A controlled empirical comparison between Fed MobiLLM and FoRA is provided in Section D.5.1, with each method retaining its native training protocol.

D.5.1 Controlled Empirical Comparison

We evaluate Fed MobiLLM and FoRA under matched model-task settings and report task performance, client resource use, and communication measurements. Each method retains its native optimization and communication procedure. The comparison therefore characterizes their end-to-end protocols under the same model-task setting instead of a matched communication or server-update budget.

Table 8: Protocol-level relation among Ladder Side-Tuning (LST), Fed MobiLLM, and FoRA.

Aspect	LST	Fed MobiLLM	FoRA
Optimization setting	Centralized	Federated, asynchronous	Federated, round-based
Backbone execution	Frozen	Forward-only	Forward-only
Adaptation input	Intermediate features	Intermediate activations	Detached multi-level representations
Trainable model location	Side network	Server	Participating clients
Gradient computation	Side network	Server-side	Client-side auxiliary model
Client upload	N/A	Activations + output residual	Auxiliary parameters
Server gradient update	N/A	Yes	No
Server aggregation	N/A	Gradient-based update	Sample-weighted FedAvg
Activation transmission	N/A	Yes	No

Table 9: Controlled comparison between Fed MobiLLM and FoRA on OPT-350M@MRPC and OPT-2.7B@SST-2. Both methods use forward-only client backbones while retaining their native optimization and communication protocols. Uplink reports total client-to-server traffic. N/A denotes a pipeline component that is not used by the corresponding method.

(a) Task quality and client GPU memory

Scenario	Method	Peak Acc. (%) ↑	Peak F1 (%) ↑	Final-10 Acc. (%) ↑	Final-10 F1 (%) ↑	Peak Memory (GiB) ↓
OPT-350M@MRPC	Fed MobiLLM	68.98 ± 0.52	79.00 ± 0.21	65.82 ± 1.15	73.92 ± 1.37	1.485
	FoRA	$\mathbf{76.11 \pm 0.21}$	$\mathbf{83.99 \pm 0.11}$	$\mathbf{75.68 \pm 0.35}$	$\mathbf{83.80 \pm 0.15}$	**1.306**
OPT-2.7B@SST-2	Fed MobiLLM	91.90 ± 0.46	92.00 ± 0.28	89.93 ± 0.63	90.26 ± 0.76	10.762
	FoRA	$\mathbf{96.22 \pm 0.07}$	$\mathbf{96.30 \pm 0.07}$	$\mathbf{96.00 \pm 0.18}$	$\mathbf{96.10 \pm 0.16}$	**10.157**

(b) Optimization placement and communication

Scenario	Method	Trainable Location	Trainable Params. (M)	Total Uplink (GiB) ↓	Upload Events	Server Training Compute (h)	Client Reload Overhead (h)
OPT-350M@MRPC	Fed MobiLLM	Server	0.296	**0.333**	913 mini-batches	0.63	0.17
	FoRA	Client	4.067	15.150	5×200 client-rounds	N/A	N/A
OPT-2.7B@SST-2	Fed MobiLLM	Server	1.312	20.350	16,674 mini-batches	3.51	17.21
	FoRA	Client	13.532	**12.603**	5×50 client-rounds	N/A	N/A

Discussion. Table 9 highlights the different scaling behavior of the two protocols. FoRA improves Final-10 Accuracy/F1 by $9.86/9.88$ points on MRPC and $6.07/5.84$ points on SST-2, while reducing peak allocated client GPU memory by 12.1% and 5.6%, respectively. Communication follows different scaling across the two settings: FoRA uses 15.150 GiB versus 0.333 GiB uplink on MRPC, while on SST-2 it uses 12.603 GiB versus 20.350 GiB, corresponding to a 38.1% reduction. Fed MobiLLM uses a substantially smaller server-side trainable model, while FoRA keeps optimization on the client and requires no server-side gradient training or client-model reload stage.

D.6 Folded-Wide Control for the Split-Bridge Parameterization

Appendix C.4 shows that, at any fixed parameter state, the factorized *Split-Bridge* residual admits an equivalent direct wide-adapter realization. We use this construction as a controlled comparison to separate the effect of the factorized optimization coordinates from the static representational capacity of the residual. On OPT-350M@MRPC, the Folded-Wide model uses the same 12 detached representation levels as FoRA and replaces the *Split* branch and the $K = 4$ *Bridge* banks, each with bottleneck dimension $d_b = 64$, by one direct adapter of width $(K + 1)d_b = 320$ at each level. The model is initialized through the exact folding map in Appendix C.4; subsequent local optimization and FedAvg are performed directly in the folded coordinates. The backbone, representation selector, global-reference coefficient, optimizer, local training protocol, client partitions, and participation schedule are otherwise unchanged. Because the folded parameterization changes the optimization coordinates, we tune its local learning rate independently using the same selection protocol as FoRA. The selected learning rate is 1×10^{-4} from the candidate set $\{1 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}\}$. The Folded-Wide results in Table 10 use this selected value.

Results. As shown in Table 10, Folded-Wide reaches a similar peak F1 to FoRA (83.72% versus 83.99%), while its best Accuracy is 1.11 points lower and its Final-10 Accuracy is 5.38 points lower. Folded-Wide reaches a Final-10 Accuracy of 70.30 ± 3.33, showing larger across-seed variability than FoRA. Peak allocated client GPU memory is essentially unchanged at 1.306 GiB. Exact folding removes $122{,}980$ trainable coordinates, corresponding to a 3.02% reduction from 4.067M to 3.944M parameters and the same relative reduction in logical model-state upload. This reduction

Table 10: Folded-Wide control on OPT-350M@MRPC. The FoRA and Folded-Wide rows report the three-seed mean ± standard deviation under the same experimental protocol. Logical upload counts the transmitted model tensors and excludes serialization overhead.

(a) Task performance and client GPU memory

Method	Best Acc. (%) ↑	Best F1 (%) ↑	Final-10 Acc. (%) ↑	Peak Memory (GiB) ↓
FoRA	76.11 ± 0.21	83.99 ± 0.11	75.68 ± 0.35	1.306
Folded-Wide	75.00 ± 0.13	83.72 ± 0.06	70.30 ± 3.33	1.306

(b) Parameter and communication control

Method	Representation Levels	Residual Bottleneck	Trainable Params. (M)	Logical Upload (MiB/client-round)	Total Uplink (GiB)
FoRA	12	5×64	4.067	15.513	15.150
Folded-Wide	12	320	3.944	15.044	14.691

follows from eliminating redundant affine normalization, branch-wise output bias, and mixture-logit coordinates in the direct realization.

Interpretation. In this OPT-350M@MRPC control, the performance difference is larger in Final-10 Accuracy than in best Accuracy: the respective gaps are 5.38 and 1.11 percentage points. Folded-Wide also exhibits a larger gap between best and Final-10 Accuracy (4.70 versus 0.43 points). These differences are observed with the same multi-level representation interface, an exactly folded initial residual function, and comparable memory and communication costs.

The folding analysis establishes equivalence of the represented residual at a fixed parameter state, while leaving the training procedures distinct. *Split-Bridge* retains branch parameters and mixture logits as separate optimization coordinates, whereas Folded-Wide directly optimizes the projections into which these factors have been absorbed. Ordinary gradient updates are generally not invariant to this reparameterization, and the nonlinear folding map does not generally commute with parameter averaging (Appendix C.4). These properties explain why the two realizations can follow different training trajectories. The lower Final-10 Accuracy of Folded-Wide, despite comparable peak performance, client GPU memory, and communication scale, is consistent with the factorized *Split-Bridge* coordinates affecting optimization and federated aggregation beyond static representational capacity.

D.7 Evaluation Metrics

Classification metrics. For an evaluation set of n examples, Accuracy is

$$\mathrm{Acc} = \frac{1}{n} \sum_{u=1}^{n} \mathbf{1}[\widehat{y}_u = y_u]\,. \tag{196}$$

For the positive class, precision, recall, and F1 are

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \qquad R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \qquad \mathrm{F1} = \frac{2PR}{P + R}. \tag{197}$$

Class 1 is treated as the positive class for both MRPC and SST-2. Evaluation is performed after every completed global round.

Best and Final-10 performance. Let $V_{m,r}^{\mathrm{Acc}}$ and $V_{m,r}^{\mathrm{F1}}$ be the classification scores of method m after round r, and let R_m be its final completed round. We report

$$V_{m,\mathrm{best}}^{\mathrm{Acc}} = \max_{1 \le r \le R_m} V_{m,r}^{\mathrm{Acc}}\,, \qquad V_{m,\mathrm{best}}^{\mathrm{F1}} = \max_{1 \le r \le R_m} V_{m,r}^{\mathrm{F1}}. \tag{198}$$

The two maxima are computed independently. Let $J_m = \min(10, R_m)$. The Final-10 score for metric V is

$$V_{m,\mathrm{final10}} = \frac{1}{J_m} \sum_{r=R_m-J_m+1}^{R_m} V_{m,r}. \tag{199}$$

Final-10 reduces sensitivity to a single late-round fluctuation and summarizes the terminal region of the training trajectory.

Perplexity. For Dolly-15K, teacher-forced loss is averaged over the supervised response tokens. If $\mathcal{T}$ denotes this token set, perplexity is

$$\mathrm{PPL} = \exp\left(-\frac{1}{|\mathcal{T}|}\sum_{\tau\in\mathcal{T}}\log p_{\mathbf{w}}(y_\tau \mid x, y_{<\tau})\right). \tag{200}$$

Lower perplexity indicates a higher conditional likelihood of the reference responses.

ROUGE-L. We use ROUGE-L F1 to measure the longest-common-subsequence overlap between a generated response and its reference (Lin, 2004). The generation configuration and normalization procedure are fixed across methods.

Best and final generation performance. For generation checkpoint r, let $\mathrm{PPL}_{m,r}$ and $\mathrm{RL}_{m,r}$ denote perplexity and ROUGE-L. The reported best values are

$$\mathrm{PPL}_m^{\mathrm{best}} = \min_r \mathrm{PPL}_{m,r}, \qquad \mathrm{RL}_m^{\mathrm{best}} = \max_r \mathrm{RL}_{m,r}\,. \tag{201}$$

For generation, Final-10 PPL and ROUGE-L are computed as the arithmetic mean over the last $J_m = \min(10, R_m)$ completed evaluation checkpoints, following Eq. 199. Each best metric may occur at a different checkpoint.

D.8 Resource Measurement and Statistical Reporting

Training time. The reported training time is cumulative federated-training wall-clock time. For each round, timing starts before selected-client preparation and ends after server aggregation. It includes client preparation, local optimization, client checkpoint writing, and server aggregation. Model evaluation, teacher-forced validation, autoregressive generation, generation-metric computation, and evaluation cleanup are recorded separately and excluded from the training-time column. For parallel or sharded GPU execution, the metric is elapsed wall-clock time and is not summed across devices.

Peak allocated client GPU memory. For each client execution, we sum the peak allocated memory recorded on the devices assigned to that client. Before measurement, CUDA is initialized with a warm-up allocation on each device, the devices are synchronized, unused cached blocks are released, and the peak-memory counters are reset. For the device set $\mathcal{G}$ used by a run, the reported memory is

$$M_{\mathrm{peak}} = \max_t \max_{i\in\mathcal{S}_t} \sum_{g\in\mathcal{G}} M_{i,t,g}^{\mathrm{alloc,peak}}, \tag{202}$$

where $M_{i,t,g}^{\mathrm{alloc,peak}}$ is the peak allocated memory recorded on device g during client i's execution in round t. This reduces to the single-device peak for the OPT experiments and to the sum of the two per-device peaks for the sharded Llama-3.1-8B experiments. The device peaks need not occur at the same instant, so their sum is distinct from the peak of simultaneous total allocation. Server evaluation memory and memory used by unrelated processes are excluded. With fixed model shapes, batch size, and allocation policy, peak allocated memory was identical across the three seeds at the reported precision.

Memory scaling with sequence length. We additionally measure peak allocated client GPU memory on OPT-2.7B@SST-2 while varying the maximum input sequence length $S \in \{64, 128, 256, 512\}$, with the remaining training configuration fixed. Table 11 reports the results. FedIT increases from 13.45 GiB at $S = 64$ to 17.23 GiB at $S = 512$, whereas FoRA changes from 10.11 GiB to 10.19 GiB over the same range. The corresponding memory reduction of FoRA relative to FedIT increases from 24.83% to 40.86%. These measurements complement Eq. 11 by characterizing the sequence-length dependence of the reported memory advantage under the classification setting.

Statistical aggregation. Best, Final-10, training-time, and peak-memory quantities are computed independently for each seed and then aggregated across seeds. Tables report the sample mean and standard deviation.

Table 11: Peak allocated client GPU memory on OPT-2.7B@SST-2 under different maximum input sequence lengths. All other training settings are fixed.

Max sequence length S	FedIT (GiB)	FoRA (GiB)	Reduction (%)
64	13.45	10.11	24.83
128	14.46	10.15	29.81
256	15.92	10.17	36.12
512	17.23	10.19	40.86