HiLoRA: Hierarchical Low-Rank Adaptation for Personalized Federated Learning

Supplementary Material

Supplementary Material Organization

This supplementary material complements the main paper as follows. We review related work and discuss its connections to existing studies in Section A. The theoretical analysis is provided in Section B, where we present the generalization analysis of HiLoRA, along with the requisite lemmas and complete proofs. Implementation details of the proposed algorithm, including notation, pseudocode, and procedural descriptions, are given in Section C. Section D further elaborates on the experimental setup, and Section E provides additional experimental results and analyzes that complement the main paper.

A. Related Works

A.1. LoRA for Vision Foundation Models

Parameter-Efficient Fine-Tuning (PEFT) adapts large, pre-trained vision models to new tasks while updating only a small subset of parameters [22, 69]. Representative techniques include Prefix-Tuning [31], Adapter-Tuning [8], Prompt-Tuning [27], and LoRA [24]. Among them, LoRA stands out thanks to its plug-and-play design and introduces no inference overhead, achieving performance comparable to full fine-tuning on diverse vision tasks—e.g., visual tracking [5, 32, 82], image generation [47, 49, 80], and semantic segmentation [52, 70]. Meanwhile, LoRA updates only a few parameters and consumes little memory, making it ideal for communication-constrained federated vision tasks. Currently, several studies have integrated LoRA-based PEFT into FL [1, 9, 35, 64–66], yet its effectiveness for personalization and generalization under non-IID data remains largely unexplored.

A.2. LoRA in Personalized FL

There are two major strategies in federated learning: generalized federated learning (GFL) [26, 74] and personalized federated learning (PFL) [10, 61]. GFL trains a single global model expected to perform well on all clients, yet severe statistical heterogeneity limits global generalization, and numerous remedies have been proposed [6, 28, 53, 59, 60]. PFL instead tailors a model to each client, or to a small client group, in order to match local data [25, 46, 73, 76, 77]. Although personalization usually increases per-client accuracy, it can overfit local patterns and reduce transferability to unseen users or domains [54, 68]. Most existing GFL and PFL approaches pay little attention to the recently popular parameter-efficient regimes.

Recent studies have moved into the PEFT setting, particularly in LoRA [2, 37, 50]. FedIT averages the LoRA factors $\mathbf{B}$ and $\mathbf{A}$, while FlexLoRA [1] and FLoRA [64] aggregate the entire update space. In the pFL context, FedSA-LoRA [18] aggregates the shared basis $\mathbf{A}$ while keeping $\mathbf{B}$ local, with $\mathbf{A}$ capturing global knowledge and $\mathbf{B}$ preserving personalization. FDLoRA [50] and FedDPA [37] adopt dual-side LoRA, alternating training between global and local updates to capture generalization and personalization separately. FedALT [2] employs a Rest-of-World (RoW) LoRA for global knowledge and a local LoRA for client-specific adaptation. Existing approaches fuse shared and personalized LoRA, but their inherent design limitations fail to strike a fine-grained balance between global generalization and client adaptation. Notably, (1) *Treat clients in isolation*, preventing the exploitation of shared structures among related clients to enhance global generalization, especially when some share similar patterns [17, 36, 38]. (2) *Use a single adapter for local patterns*, which may lead to overfitting on local patterns while neglecting shared knowledge, ultimately impairing personalization [54, 68]. Despite recent work on tree-structured LoRA for continual learning, which builds a task-driven hierarchy to mitigate forgetting under sequential tasks [51], no prior study has explored hierarchical LoRA within personalized FL. This paper introduces HiLoRA, a hierarchical framework that organizes LoRA modules into root, cluster, and leaf layers, allowing for a more refined capture of both multi-level global knowledge and client-specific personalization.

B. HiLoRA Generalization Proofs

B.1. Lemmas

Lemma 1 (Uniform Rademacher bound)**.** *Let $\mathcal{F} \subset [0,1]$ and $S \sim Q^m$. With a probability of at least $1-\delta$,*

$$\sup_{f\in\mathcal{F}} |L_Q(f) - \hat{L}_S(f)| \;\le\; 2\,\widehat{\mathfrak{R}}_S(\mathcal{F}) \;+\; 3\sqrt{\frac{\log(2/\delta)}{2m}}.$$

Lemma 2 (Empirical replacement)**.** *Let $\mathcal{H}$ be a hypothesis class and $\mathcal{F}$ its loss-induced function class. For any distribution Q, any sample S, and any $h \in \mathcal{H}$,*

$$L_Q(h) - \inf_{h'\in\mathcal{H}} L_Q(h') \;\le\; \Big(\hat{L}_S(h) - \inf_{h'\in\mathcal{H}} \hat{L}_S(h')\Big) + 2\sup_{f\in\mathcal{F}} \big|L_Q(f) - \hat{L}_S(f)\big|.$$

Proof. For any $h \in \mathcal{H}$, $L_Q(h) \le \hat{L}_S(h) + \sup_{f\in\mathcal{F}} \big|L_Q(f) - \hat{L}_S(f)\big|$. For any $h' \in \mathcal{H}$, $L_Q(h') \ge$

$\hat{L}_S(h') - \sup_{f\in\mathcal{F}} \left|L_Q(f) - \hat{L}_S(f)\right|$. Taking the infimum over $h' \in \mathcal{H}$ in the second inequality and subtracting it from the first yields the claim. □

B.2. Detailed Proofs

Proof. On client i, we consider the excess risk $L_{D_i}(h^{(r,c,\ell)}) - \inf_{h\in\mathcal{H}} L_{D_i}(h)$. By inserting and subtracting the intermediate risks $L_{D_i}(h^{(r)})$ and $L_{D_i}(h^{(r,c)})$, this can be decomposed into three increments corresponding to the root, cluster, and leaf contributions:

$$L_{D_i}\big(h^{(r,c,\ell)}\big) - \inf_{h\in\mathcal{H}} L_{D_i}(h) = \Delta_r + \Delta_c + \Delta_\ell, \tag{15}$$

where

$$\begin{aligned}
\Delta_r &:= L_{D_i}\big(h^{(r)}\big) - \inf_{h\in\mathcal{H}} L_{D_i}(h),\\
\Delta_c &:= L_{D_i}\big(h^{(r,c)}\big) - L_{D_i}\big(h^{(r)}\big),\\
\Delta_\ell &:= L_{D_i}\big(h^{(r,c,\ell)}\big) - L_{D_i}\big(h^{(r,c)}\big).
\end{aligned} \tag{16}$$

We first bound the root contribution Δ_r.

$$\begin{aligned}
\Delta_r &= L_{D_i}\big(h^{(r)}\big) - \inf_{h\in\mathcal{H}} L_{D_i}(h)\\
&\overset{(a_1)}{\le} \Big[L_D\big(h^{(r)}\big) + \mathrm{disc}_{\mathcal{F}_r}(D_i, D)\Big] - \inf_{h\in\mathcal{H}} L_{D_i}(h)\\
&\overset{(a_2)}{\le} \Big[L_D\big(h^{(r)}\big) + \mathrm{disc}_{\mathcal{F}_r}(D_i, D)\Big]\\
&\quad - \Big[\inf_{h\in\mathcal{H}} L_D(h) - \mathrm{disc}_{\mathcal{F}}(D_i, D)\Big]\\
&\overset{(a)}{\le} \big[L_D(h^{(r)}) - \inf_{h\in\mathcal{H}} L_D(h)\big] + 2\ \mathrm{disc}_{\mathcal{F}}(D_i, D)\\
&\overset{(b_1)}{\le} \big(\hat{L}_S(h^{(r)}) - \inf_{h\in\mathcal{H}} \hat{L}_S(h)\big) + 2\sup_{f\in\mathcal{F}} \left|L_D(f) - \hat{L}_S(f)\right|\\
&\quad + 2\,\mathrm{disc}_{\mathcal{F}}(D_i, D)\\
&\overset{(b_2)}{\le} \big(\hat{L}_S(h^{(r)}) - \inf_{h\in\mathcal{H}} \hat{L}_S(h)\big) + 4\,\widehat{\mathfrak{R}}_S(\mathcal{F}) + 6\sqrt{\tfrac{\log(2/\delta)}{2m}}\\
&\quad + 2\ \mathrm{disc}_{\mathcal{F}}(D_i, D),
\end{aligned} \tag{17}$$

where (a_1) uses the discrepancy on $\mathcal{F}_r$ for the fixed $h^{(r)}$; (a_2) uses the discrepancy on $\mathcal{F}$ to obtain a bound uniform over $h \in \mathcal{H}$. (b1) applies Lemma 2 to $\mathcal{H}$, replacing population risks under D by empirical risks on S. (b2) further uses Lemma 1 for $\mathcal{F}$ with i.i.d. $S \sim D^m$.

We rewrite the cluster increment Δ_c as an identity that separates the in-cluster improvement from client-cluster deviations. Let $j = j(i)$. Adding and subtracting the risks under C_j yields:

$$\begin{aligned}
\Delta_c &= \big[L_{C_{j(i)}}(h^{(r,c)}) - L_{C_{j(i)}}(h^{(r)})\big]\\
&\quad + \big(L_{D_i} - L_{C_{j(i)}}\big)(h^{(r,c)}) + \big(L_{C_{j(i)}} - L_{D_i}\big)(h^{(r)}).
\end{aligned} \tag{18}$$

We next bound the cluster contribution Δ_c. Using the identity that inserts and subtracts risks under $C_{j(i)}$,

$$\begin{aligned}
\Delta_c &= \big[L_{C_{j(i)}}(h^{(r,c)}) - L_{C_{j(i)}}(h^{(r)})\big]\\
&\quad + \big(L_{D_i} - L_{C_{j(i)}}\big)(h^{(r,c)}) + \big(L_{C_{j(i)}} - L_{D_i}\big)(h^{(r)})\\
&\overset{(a)}{\le} \big[L_{C_{j(i)}}(h^{(r,c)}) - L_{C_{j(i)}}(h^{(r)})\big] + 2\ \mathrm{disc}_{\mathcal{F}_c^{\perp(j(i))}}(D_i, C_{j(i)})\\
&\overset{(b)}{\le} \big(\hat{L}_{S_{j(i)}}(h^{(r,c)}) - \hat{L}_{S_{j(i)}}(h^{(r)})\big) + 2\ \mathrm{disc}_{\mathcal{F}_c^{\perp(j(i))}}(D_i, C_{j(i)})\\
&\quad + 2\sup_{f\in\mathcal{F}_c^{\perp(j(i))}} \left|L_{C_{j(i)}}(f) - \hat{L}_{S_{j(i)}}(f)\right|\\
&\overset{(c)}{\le} \big(\hat{L}_{S_{j(i)}}(h^{(r,c)}) - \hat{L}_{S_{j(i)}}(h^{(r)})\big) + 4\,\widehat{\mathfrak{R}}_{S_{j(i)}}\big(\mathcal{F}_c^{\perp(j(i))}\big)\\
&\quad + 6\sqrt{\tfrac{\log(2/\delta)}{2m_{j(i)}}} + 2\ \mathrm{disc}_{\mathcal{F}_c^{\perp(j(i))}}(D_i, C_{j(i)}),
\end{aligned} \tag{19}$$

where (a) uses the discrepancy on the restricted class $\mathcal{F}_c^{\perp(j(i))}$, justified by $\Delta W_c^{(j(i))} \in U_c^{(j(i))}$ from Assumption 1. (b) applies Lemma 2 with the restricted hypothesis class $\mathcal{H}_c^{\perp(j(i))} := \{x \mapsto h^{(r)}(x; W_0 + \Delta W_c) : \Delta W_c \in U_c^{(j(i))}\}$ whose loss-induced class is $\mathcal{F}_c^{\perp(j(i))}$, replacing population risks under $C_{j(i)}$ with empirical risks on $S_{j(i)}$. (c) further uses Lemma 1 for $\mathcal{F}_c^{\perp(j(i))}$ with i.i.d. $S_{j(i)} \sim C_{j(i)}^{m_{j(i)}}$.

Finally, we bound the leaf contribution Δ_ℓ. Since training and evaluation are both on D_i and $\Delta W_\ell^{(i)} \in U_\ell^{(i)}$ by Assumption 1, we work with the parameter-induced restricted class $\mathcal{F}_\ell^{\perp(i)}$:

$$\begin{aligned}
\Delta_\ell &= L_{D_i}\big(h^{(r,c,\ell)}\big) - L_{D_i}\big(h^{(r,c)}\big)\\
&\overset{(a)}{\le} \big(\hat{L}_{\mathcal{D}_i}(h^{(r,c,\ell)}) - \hat{L}_{\mathcal{D}_i}(h^{(r,c)})\big)\\
&\qquad + 2\sup_{f\in\mathcal{F}_\ell^{\perp(i)}} \left|L_{D_i}(f) - \hat{L}_{\mathcal{D}_i}(f)\right|\\
&\overset{(b)}{\le} \big(\hat{L}_{\mathcal{D}_i}(h^{(r,c,\ell)}) - \hat{L}_{\mathcal{D}_i}(h^{(r,c)})\big) + 4\,\widehat{\mathfrak{R}}_{\mathcal{D}_i}\big(\mathcal{F}_\ell^{\perp(i)}\big)\\
&\qquad + 6\sqrt{\tfrac{\log(2/\delta)}{2n_i}},
\end{aligned} \tag{20}$$

where (a) applies Lemma 2 to the restricted class $\mathcal{H}_\ell^{\perp(i)}$, replacing population risks under D_i by empirical risks on $\mathcal{D}_i$. (b) further applies Lemma 1 to $\mathcal{F}_\ell^{\perp(i)}$ with i.i.d. $\mathcal{D}_i \sim D_i^{n_i}$. By setting the per–event failure probability to δ and applying the union bound, we obtain

$$\Pr\left(\mathcal{E}_r \cap \mathcal{E}_c \cap \mathcal{E}_\ell\right) \ge 1 - 3\delta. \tag{21}$$

Here $\mathcal{E}_r$, $\mathcal{E}_c$, and $\mathcal{E}_\ell$ denote the high–probability events guaranteed by Lemma 1 for $\mathcal{F}$, $\mathcal{F}_c^{\perp(j(i))}$, and $\mathcal{F}_\ell^{\perp(i)}$, used in (17), (19), and (20). On the event $\mathcal{E}_r \cap \mathcal{E}_c \cap \mathcal{E}_\ell$, summing the bounds for Δ_r, Δ_c, and Δ_ℓ and using the telescoping decomposition yields the claimed result in Theorem 1.

□

Table 5. Summary of main notation.

Symbol	Description
$\mathbf{W}_0 \in \mathbb{R}^{p\times q}$	Frozen backbone weight matrix.
$\Delta\mathbf{W} = \mathbf{BA}$	LoRA update with basis $\mathbf{B} \in \mathbb{R}^{p\times r}$ and coefficient $\mathbf{A} \in \mathbb{R}^{r\times q}$.
r	LoRA rank ($r \ll \min(p, q)$).
H	Number of hierarchy tiers.
$\mathcal{M}$	Set of LoRA modules.
$(\mathbf{B}_r, \mathbf{A}_r)$	Root adapter, shared by all N clients.
$(\mathbf{B}_{c,j}, \mathbf{A}_{c,j})$	Cluster adapter for cluster $\mathcal{C}_j$, $j \in [K]$.
$(\mathbf{B}_{\ell,i}, \mathbf{A}_{\ell,i})$	Leaf adapter for client $i \in [N]$.
$\mathcal{C}_j$	Index set of clients in cluster j; $\bigsqcup_{j=1}^{K} \mathcal{C}_j = \{1, \dots, N\}$.
$j(i)$	Cluster assignment map of client i.
$\mathcal{D}_i$	Local dataset of client i, $\{(\mathbf{x}_i^{(k)}, \mathbf{y}_i^{(k)})\}_{k=1}^{n_i}$.
n_i	Number of local samples, $n_i = \lvert\mathcal{D}_i\rvert$.
π_i^{root}	Root-level aggregation weight, $\frac{n_i}{\sum_{u=1}^{N} n_u}$.
$\pi_{i,j}^{\text{cluster}}$	Intra-cluster aggregation weight for $i \in \mathcal{C}_j$, $\frac{n_i}{\sum_{u\in\mathcal{C}_j} n_u}$.
$\ell_i(\mathbf{W}_0 + \Delta\mathbf{W})$	Empirical loss of client i on $\mathcal{D}_i$.
$\mathcal{R}(\mathbf{B})$	Column space of $\mathbf{B}$.
$\lVert\cdot\rVert_F$, $\langle\cdot,\cdot\rangle_F$	Frobenius norm and Frobenius inner product.
$\mathbf{B}_{i,m}^{(t)}$, $\bar{\mathbf{B}}_{i,m}^{(t)}$	Module-wise LoRA basis for client i at round t and its EMA-normalized version.
λ	EMA decay for $\bar{\mathbf{B}}$, with $\lambda \in (0, 1)$.
d_{ij}, $\mathbf{D}_{\text{dist}} \in \mathbb{R}^{N\times N}$	Cosine distance between clients i, j and the resulting distance matrix.
$K^\star$	Selected number of clusters.
$\rho_{\text{rel}}^{(t)}$, τ_{rel}, ε	Relative-update statistic at round t, its threshold, and a small stabilizer.
γ_c, γ_ℓ	Regularization strengths encouraging cross-tier orthogonality.
$\widehat{\mathfrak{R}}_S(\mathcal{F})$	Empirical Rademacher complexity of function class $\mathcal{F}$ on sample set S.
$h^{(r)}$, $h^{(r,c)}$, $h^{(r,c,\ell)}$	Predictors using root, root–cluster, and root–cluster–leaf adapters, respectively.
$\text{disc}_{\mathcal{F}}(Q_1, Q_2)$	Distribution discrepancy under loss class $\mathcal{F}$: $\text{disc}_{\mathcal{F}}(Q_1, Q_2) := \sup_{f\in\mathcal{F}} \lvert L_{Q_1}(f) - L_{Q_2}(f)\rvert$.

C. Algorithm Details

C.1. Notation

We summarize the main symbols in Table 5.

Algorithm 1: HiLoRA: Hierarchical LoRA Training for Personalized Federated Learning.

Input: $\mathbf{W}_0$; $\{\mathcal{D}_i\}_{i=1}^N$; weights $\{\pi_i^{\text{root}}\}_{i=1}^N$, $\{\pi_{i,j}^{\text{cluster}}\}_{i\in\mathcal{C}_j}$; $\tau_{\text{rel}}, \varepsilon$; γ_c, γ_ℓ.
Output: $(\mathbf{B}_r^\star, \mathbf{A}_r^\star)$, $\{(\mathbf{B}_{c,j}^\star, \mathbf{A}_{c,j}^\star)\}$, $\{(\mathbf{B}_{\ell,i}^\star, \mathbf{A}_{\ell,i}^\star)\}$.

$(\mathbf{B}_r^\star, \mathbf{A}_r^\star) \leftarrow$ TrainTier$\big(I = [N], \{\pi_i^{\text{root}}\}_{i=1}^N, \text{Init} = (\mathbf{B}_r, \mathbf{A}_r),\ \mathcal{A}_i(\Phi) = \varnothing,\ C = \varnothing\big)$; ▷ Root Stage

$\{\mathcal{C}_j\}_{j=1}^{K^\star} \leftarrow$ SubspaceClustering (Alg. 2);

for (***in parallel***) $j = 1$ **to** $K^\star$ **do**

  $(\mathbf{B}_{c,j}^\star, \mathbf{A}_{c,j}^\star) \leftarrow$ TrainTier$\big(I = \mathcal{C}_j,\ \{\pi_{i,j}^{\text{cluster}}\}_{i\in\mathcal{C}_j},\ \text{Init} = (\mathbf{B}_{c,j}, \mathbf{A}_{c,j}),\ \mathcal{A}_i(\Phi) = \{(\mathbf{B}_r^\star, \mathbf{A}_r^\star)\},\ C = \{(\mathbf{B}_r^\star, \gamma_c)\}\big)$;

end ▷ Cluster Stage

for (***in parallel***) $i \in [N]$ **do**

  $(\mathbf{B}_{\ell,i}^\star, \mathbf{A}_{\ell,i}^\star) \leftarrow$ TrainTier$\big(I = \{i\}, \{\pi_i{=}1\},\ \text{Init} = (\mathbf{B}_{\ell,i}, \mathbf{A}_{\ell,i}), \mathcal{A}_i(\Phi) = \{(\mathbf{B}_r^\star, \mathbf{A}_r^\star), (\mathbf{B}_{c,\,j(i)}^\star, \mathbf{A}_{c,\,j(i)}^\star)\},\ C = \{(\mathbf{B}_r^\star, \gamma_c), (\mathbf{B}_{c,\,j(i)}^\star, \gamma_\ell)\}\big)$;

end ▷ Leaf Stage

Function TrainTier$(I, \{\pi_i\}_{i\in I},\ \mathit{Init} = (\mathbf{B}^{(0)}, \mathbf{A}^{(0)}),\ \mathcal{A}_i(\Phi),\ C)$**:**

  if $\mathcal{A}_i(\Phi) \neq \varnothing$ **then**

    $\widetilde{\mathbf{W}}_0 \leftarrow \mathbf{W}_0 + \sum_{(\tilde{\mathbf{B}}, \tilde{\mathbf{A}}) \in \mathcal{A}_i(\Phi)} \tilde{\mathbf{B}}\tilde{\mathbf{A}}$;

  for $t = 0, 1, 2, \dots$ **do**

    foreach $i \in I$ ***in parallel*** **do**

      $\mathcal{R}_i(\mathbf{B}) \leftarrow \sum_{(\hat{\mathbf{B}}, \gamma) \in C} \gamma \lVert\hat{\mathbf{B}}^\top \mathbf{B}\rVert_F^2$;

      $\mathcal{L}_i^{(t)} \leftarrow \ell_i\big(\widetilde{\mathbf{W}}_0 + \mathbf{B}^{(t)}\mathbf{A}^{(t)}\big) + \mathcal{R}_i\big(\mathbf{B}^{(t)}\big)$;

      $(\mathbf{B}_i^{(t+1)}, \mathbf{A}_i^{(t+1)}) \leftarrow$ LocalOptimize$\big(\mathcal{L}_i^{(t)}, \mathbf{B}^{(t)}, \mathbf{A}^{(t)}\big)$;

    end

    $\Delta\mathbf{W}^{(t)} \leftarrow \sum_{i\in I} \pi_i \mathbf{B}_i^{(t+1)} \mathbf{A}_i^{(t+1)}$;

    if $t > 0$ *and* $\dfrac{\lVert\Delta\mathbf{W}^{(t)} - \Delta\mathbf{W}^{(t-1)}\rVert_F}{\lVert\Delta\mathbf{W}^{(t-1)}\rVert_F + \varepsilon} \leq \tau_{rel}$ **then** $(\mathbf{B}^\star, \mathbf{A}^\star) \leftarrow (\mathbf{B}^{(t)}, \mathbf{A}^{(t)})$; **break**;

    $(\mathbf{B}^{(t+1)}, \mathbf{A}^{(t+1)}) \leftarrow$ SVDUpdate$(\Delta\mathbf{W}^{(t)})$;

  end

  return $(\mathbf{B}^\star, \mathbf{A}^\star)$;

C.2. Training Procedure

We follow the cascade in Algorithm 1: *(i)* train the root tier and freeze it; *(ii)* run the LoRA-Subspace Adaptive Clustering in Algorithm 2 to obtain $\{\mathcal{C}_j\}_{j=1}^{K^\star}$; *(iii)* train and freeze each cluster tier; *(iv)* train the leaf tier.

Algorithm 2: LoRA-Subspace Adaptive Clustering.

Input: Streaming LoRA bases $\{\mathbf{B}_{i,m}^{(t)}\}$ up to root round t for clients $i \in [N]$ and modules $m \in [\mathcal{M}]$; EMA decay $\lambda \in (0,1)$; search range $K \in [K_{\min}, K_{\max}]$.
Output: Partition result $\{\mathcal{C}_j\}_{j=1}^{K^\star}$ and $K^\star$.

// EMA of update direction $\mathbf{B}$ at root stage round t
foreach *client* $i \in [N]$, *module* $m \in [\mathcal{M}]$ **do**
  $\hat{\mathbf{B}}_{i,m}^{(t)} \leftarrow \mathbf{B}_{i,m}^{(t)} / \|\mathbf{B}_{i,m}^{(t)}\|_F$;
  $\bar{\mathbf{B}}_{i,m}^{(t)} \leftarrow \lambda\, \bar{\mathbf{B}}_{i,m}^{(t-1)} + (1-\lambda)\, \hat{\mathbf{B}}_{i,m}^{(t)}$;
  $\bar{\mathbf{B}}_{i,m}^{(t)} \leftarrow \bar{\mathbf{B}}_{i,m}^{(t)} / \|\bar{\mathbf{B}}_{i,m}^{(t)}\|_F$;
  $\bar{\mathbf{B}}_{i,m}^{(t)} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top,\ \mathbf{U}_{i,m} \leftarrow \mathbf{U}_{[:,1:r]}$;
end
// Pairwise distance
foreach $i, j \in [N]$ **do**
  $d_{ij} = 1 - \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \frac{1}{r} \left\| \mathbf{U}_{i,m}^{(t)\top} \mathbf{U}_{j,m}^{(t)} \right\|_F^2$;
  $\mathbf{D}_{ij} \leftarrow d_{ij}$;
end
// Spectral clustering
$\sigma \leftarrow \mathrm{median}\{\mathbf{D}_{ij} : i \neq j\}$;
$\mathbf{S}_{ij} \leftarrow \exp\big(-\mathbf{D}_{ij}^2/(2\sigma^2)\big),\ \mathbf{S}_{ii} \leftarrow 1$;
$\mathbf{1} \leftarrow$ all-ones vector in $\mathbb{R}^N$; $\mathbf{D}_{\text{deg}} \leftarrow \mathrm{diag}(\mathbf{S}\,\mathbf{1})$;
$\lambda_1 \le \cdots \le \lambda_N \leftarrow \mathrm{eig}(\mathbf{I} - \mathbf{D}_{\text{deg}}^{-1/2}\,\mathbf{S}\,\mathbf{D}_{\text{deg}}^{-1/2})$;
$K^\star \leftarrow \arg\max_{K \in [K_{\min}, K_{\max}]} \big(\lambda_K - \lambda_{K-1}\big)$;
$\{\mathcal{C}_j\}_{j=1}^{K^\star} \leftarrow \mathrm{SC}(\mathbf{S}, K^\star)$;
Return $\{\mathcal{C}_j\}_{j=1}^{K^\star}$;

Pseudocode. We abstract the training loop for each tier in TRAINTIER in Algorithm 1. Tier-specific inputs differ only in the active client set S, the LoRA initialization parameters, the set of frozen adapters $\mathcal{A}_i(\Phi)$ included in the composed backbone, and the orthogonality regularizers C. After the cascade completes, each client i is associated with a personalized path of frozen adapters

$$\mathcal{A}_i(\Phi) = \{(\mathbf{B}_r^\star, \mathbf{A}_r^\star), (\mathbf{B}_{c,\,j(i)}^\star, \mathbf{A}_{c,\,j(i)}^\star), (\mathbf{B}_{\ell,\,i}^\star, \mathbf{A}_{\ell,\,i}^\star)\}, \tag{22}$$

which defines its effective model.

Specifically, Algorithm 2 details the LoRA-Subspace Adaptive Clustering procedure. In this stage, each client maintains LoRA adapters inserted into multiple modules of the backbone. Let $\mathcal{M}$ denote the set of such LoRA modules and let $m \in \mathcal{M}$ index one of them. During the root stage, we collect the streaming LoRA bases $\{\mathbf{B}_{i,m}^{(t)}\}$ for each client i and module m. For each update, we normalize and apply EMA:

$$\hat{\mathbf{B}}_{i,m}^{(t)} = \frac{\mathbf{B}_{i,m}^{(t)}}{\|\mathbf{B}_{i,m}^{(t)}\|_F}, \quad \bar{\mathbf{B}}_{i,m}^{(t)} = \lambda\, \bar{\mathbf{B}}_{i,m}^{(t-1)} + (1-\lambda)\hat{\mathbf{B}}_{i,m}^{(t)}, \tag{23}$$

followed by re-normalization and SVD to obtain the top-r left singular vectors $\mathbf{U}_{i,m}^{(t)}$. We compute pairwise distances by averaging squared principal-angle cosines across modules:

$$d_{ij} = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} d_{ij}^{(m)} = 1 - \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \frac{1}{r} \left\| \mathbf{U}_{i,m}^{(t)\top} \mathbf{U}_{j,m}^{(t)} \right\|_F^2. \tag{24}$$

Collect all d_{ij} into $\mathbf{D}_{\text{dist}}$, convert it to an affinity matrix with a Gaussian kernel, and run normalized spectral clustering. Choose K by sweeping $K \in [K_{\min}, K_{\max}]$ and selecting the maximizer of the eigengap of the normalized Laplacian spectrum [62], yielding $\{\mathcal{C}_j\}_{j=1}^{K^\star}$.

D. Experimental Details

D.1. Federated Partitioning

We conduct experiments on CIFAR-100 under multiple levels of *label heterogeneity*. (1) Global Dirichlet (GL–Dir) samples each client's label prior from a Dirichlet distribution over the entire label space, simulating global label imbalance. (2) Superclass-conditioned Dirichlet (SC–Dir) further structures the distribution by sampling from CIFAR-100's 20 superclasses. Each client is assigned one superclass, and within that superclass, the local label proportions follow a Dirichlet distribution over its member classes. (3) Pathological (Patho) partition assigns each client data from only a limited number of classes, simulating an extreme non-IID setting.

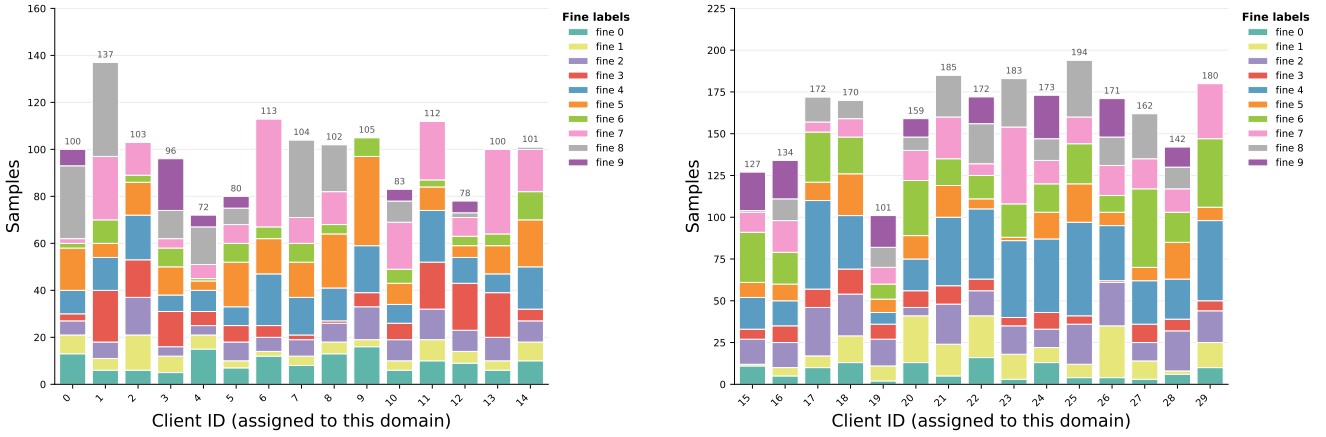
On DomainNet, we introduce *feature heterogeneity*. Each client receives data exclusively from a single domain (e.g., clipart, real, sketch). Within each domain, clients further follow a Dirichlet label distribution, creating a more challenging and realistic federated setting. Figure 6 visualizes part of the resulting client data distributions for DomainNet.

D.2. Baselines

In our experiments, we compare the following categories of baseline methods:

Single-layer LoRA methods:

- **Local-LoRA:** Each client fine-tunes its LoRA module solely on local data without participating in any communication or global aggregation.
- **FedIT [78]:** A straightforward federated aggregation strategy that directly averages the two LoRA factor matrices $\mathbf{B}$ and $\mathbf{A}$ across clients.



(a) Clipart domain (b) Infograph domain

Figure 6. **An example of per-domain client label distributions on DomainNet.** Each subplot shows the fine-grained class composition for clients in a domain, with stacked bars indicating class proportions per client—representing a more challenging non-IID scenario.

- **FlexLoRA [1]:** Each client first computes its LoRA update $\Delta\mathbf{W} = \mathbf{BA}$ and performs aggregation in the product space.

Dual-LoRA or hybrid methods:

- **FedSA-LoRA [18]:** Aggregates only the shared $\mathbf{A}$ matrices while keeping $\mathbf{B}$ local, aiming to balance global generalization with client-specific adaptation.
- **FDLoRA [50]:** Each client maintains both a global LoRA and a local LoRA. The global LoRA is aggregated across clients, while the local one remains private, and the two are adaptively fused during inference.
- **FedDPA-F / FedDPA-T [37]:** This method introduces two training paradigms. The FedDPA-F variant sequentially optimizes the global LoRA followed by the local LoRA, while the FedDPA-T variant alternates between them within each training round to improve personalization.
- **PF2LoRA [23]:** Employs a two-level LoRA design with shared and client-specific adapters, trained via a bi-level optimization scheme that dynamically selects the rank to balance efficiency and accuracy.
- **FedALT [2]:** Maintains an Individual LoRA and a global Rest-of-the-World (RoW) LoRA, using an adaptive gating network to fuse them, enabling personalized yet globally consistent adaptation.

Overall, these methods employ *flat* LoRA optimization and aggregation strategies that do not explicitly capture the hierarchical or clustered structure among clients. In contrast, our proposed **HiLoRA** is, to our knowledge, the first framework that explicitly integrates client-side hierarchical relationships into LoRA adaptation and aggregation.

D.3. Evaluation and Hyperparameters

Stopping and budget fairness. We employ a per-tier relative-improvement stopping rule with a threshold of $\tau_{\text{rel}} = 0.03$ [45]:

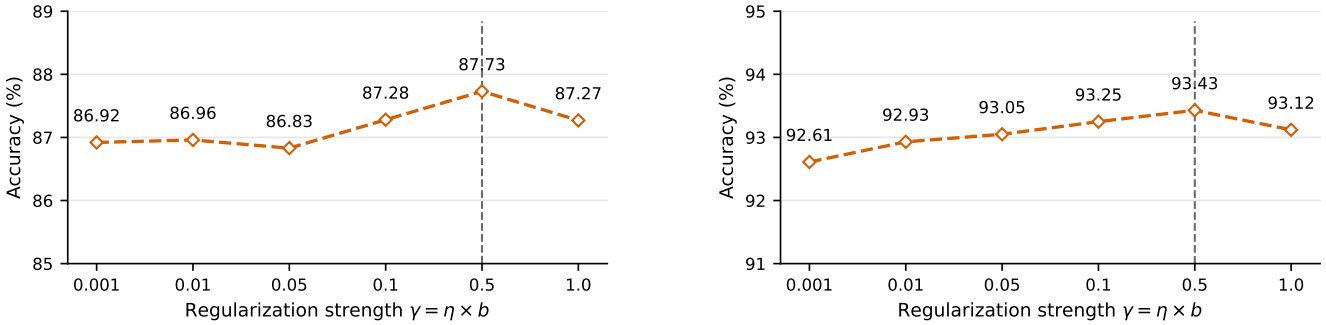
$$\frac{\|\Delta\mathbf{W}^{(t)} - \Delta\mathbf{W}^{(t-1)}\|_F}{\|\Delta\mathbf{W}^{(t-1)}\|_F + \varepsilon} \leq \tau_{\text{rel}}.$$

To ensure a fair comparison across methods, we cap the maximum number of rounds per tier at $T_{\text{root}} = 25$, $T_{\text{cluster}} = 15$, and $T_{\text{leaf}} = 10$. Each tier terminates once either the relative criterion is met or the cap is reached. This yields an overall budget of at most 50 rounds per client, matching the single-branch setting and slightly lower than typical dual-branch configurations, while keeping local epochs and batch sizes identical across methods.

Hyperparameter settings. The LoRA rank is set to $r = 16$. We sweep the learning rate $\eta \in \{1\text{e}{-3}, 5\text{e}{-4}, 3\text{e}{-4}, 1\text{e}{-4}, 5\text{e}{-5}, 1\text{e}{-5}\}$. The best setting is selected from the grid, yielding $\eta = 3\text{e}{-4}$ for DomainNet and $\eta = 1\text{e}{-4}$ for CIFAR-100. For Local-LoRA, we report the best local fine-tuning results within the first 20 local epochs [39]. For the LoRA basis $\mathbf{B}$, the EMA used in subspace clustering employs a decay coefficient of $\lambda = 0.9$, which smooths cross-round noise without excessive lag [4]. For spectral clustering, we scan $K \in [5, 30]$ and select $K^\star$ via the eigengap of the normalized Laplacian. We set the update steps to 20 to assign each new client to its nearest cluster, and load the corresponding root and cluster adapters. All experiments use the AdamW optimizer, a batch size of 64, and a total of 50 global rounds to ensure a comparable training budget. Random seeds are fixed across all libraries to ensure reproducibility. All experiments are run on NVIDIA A100 80GB GPUs.

E. Additional Experimental Analyzes

Regularization strength analysis. We analyze the sensitivity of the orthogonality regularization by scaling the regularization strength γ with respect to the



(a) DomainNet (b) CIFAR-100

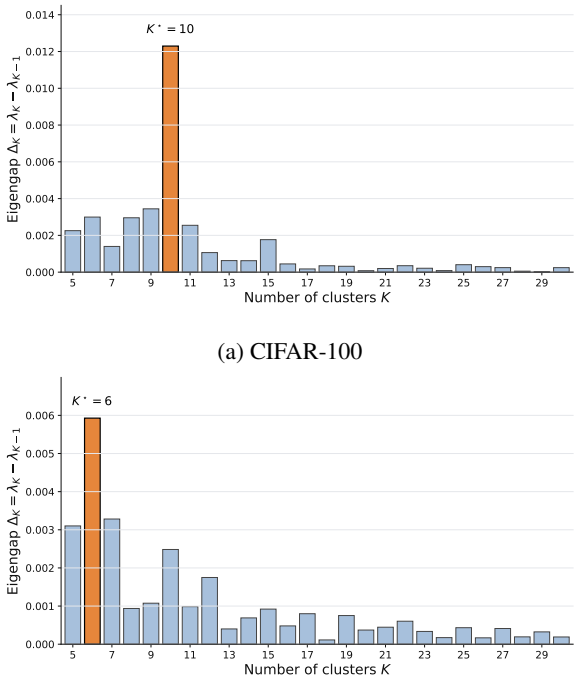
Figure 7. Regularization strength analysis on DomainNet and CIFAR-100. We observe an optimum near $b \approx 0.5$ and recommend using $\gamma = 0.5 \times \eta$ by default.

Table 6. Effect of similarity metric on clustering HiLoRA. We compare cosine, L2, and Pearson metrics on DomainNet and CIFAR-100, and report both Personalized (Per.) and Generalized (Gen.) performance.

Similarity metric	**DomainNet**		**CIFAR-100**	
	Per. ↑	**Gen.** ↑	**Per.** ↑	**Gen.** ↑
L2	0.861	0.849	0.834	0.827
Pearson	0.868	0.852	0.839	0.831
Cosine	**0.877**	**0.861**	**0.846**	**0.841**

local training learning rate η using a factor $b \in \{0.001, 0.01, 0.05, 0.1, 0.5, 1.0\}$, where $\gamma = \eta \times b$. We set the same regularization strength for the cluster and leaf tiers, i.e., $\gamma = \gamma_c = \gamma_\ell$. As shown in Fig. 7, results on DomainNet and CIFAR-100 exhibit an optimum around $b \approx 0.5$. A small b imposes insufficient constraint, allowing redundant cross-tier directions and leading to interference across tiers. A large b over-regularizes the adapters, limiting their subspace flexibility and hindering effective optimization. We select $b = 0.5$ as the default value used throughout this study.

Selecting the number of clusters. As shown in Fig. 8, we visualize the spectrum of the symmetric normalized Laplacian and its eigengap histogram $\Delta_K = \lambda_K - \lambda_{K-1}$ on CIFAR-100 and DomainNet. Within the search range $K \in [5, 30]$, we set $K^\star = \arg\max_K \Delta_K$, where the largest eigenvalue gap indicates the optimal cluster number according to the eigengap principle [62]. On DomainNet, a pronounced gap at $K = 6$ yields $K^\star = 6$, with adjacent K producing strictly smaller gaps. For CIFAR-100, under a deliberately pathological non-IID partition, the spectrum exhibits its largest jump at $K = 10$, leading to $K^\star = 10$. We then perform normalized spectral clustering with $K^\star$ to obtain the partition $\{\mathcal{C}_j\}_{j=1}^{K^\star}$.



(a) CIFAR-100

(b) DomainNet

Figure 8. Cluster number selection on CIFAR-100 and DomainNet. We visualize the eigenvalue spectrum of the normalized Laplacian and its eigengap $\Delta_K = \lambda_K - \lambda_{K-1}$. A pronounced gap at $K = 10$ on CIFAR-100 and at $K = 6$ on DomainNet suggests the optimal cluster numbers $K^\star = 10$ and $K^\star = 6$, respectively.

Effects of similarity metric. During our clustering phase, cosine similarity is used by default as the metric for client subspace similarity. Compared to other metrics, cosine similarity exhibits better scale normalization and numerical stability in high-dimensional spaces, and its effectiveness and robustness have been demonstrated in multiple FL studies

Table 7. Efficiency comparison on DomainNet. All methods are trained for 50 communication rounds with 2 local epochs.

Method	FedIT	FlexLoRA	FedSA-LoRA	FDLoRA	FedDPA-F	FedDPA-T	PF2LoRA	FedALT	HiLoRA
GPU-hours	5h 39m	5h 42m	5h 40m	11h 18m	6h12m	11h52m	7h 10m	7h 8m	5h 43m
Relative FLOPs	$1.0\times$	$1.0\times$	$1.0\times$	$\approx 2.0\times$	$\approx 1.1\times$	$\approx 2.0\times$	≈ 1.2–$1.5\times$	≈ 1.1–$1.3\times$	$\approx 1.0\times$

Table 8. Trainable parameters and complexities of LoRA variants.

Method	**Trainable Params (M)**	**Upload Complexity**	**Storage Complexity**
Single-branch	≈ 0.67	$\mathcal{O}(NT)$	$\mathcal{O}(N)$
Dual-branch	$\approx 0.67 \times 2$	$\mathcal{O}(NT)$	$\mathcal{O}(N{+}1)$ or $\mathcal{O}(2N)$
HiLoRA (ours)	≈ 0.67	$\mathcal{O}\big(N\,(T_{\text{root}}{+}T_{\text{cluster}})\big)$	$\mathcal{O}(N{+}K^{\star}{+}1)$

[3]. We also compared similarity metrics based on the L2 norm and the Pearson correlation coefficient to analyze the impact of different distance definitions on clustering quality and subsequent optimization. As shown in Table 6, cosine similarity outperforms other metrics in both cluster consistency and model performance, demonstrating that metrics that capture directional consistency are more discriminative in high-dimensional LoRA subspaces.

Efficiency analysis. We evaluate efficiency on DomainNet using 50 communication rounds and 2 local epochs on a single NVIDIA A100 80 GB GPU. Table 7 reports wall-clock time in GPU-hours and Relative FLOPs normalized to FedIT. HiLoRA achieves nearly the same wall-clock cost as the fastest baselines, about 1.01 × that of FedIT. The communication and storage complexities of different LoRA–FL variants are summarized in Table 8. We group the variants by the number of branches. *Single-branch* uploads one LoRA branch per communication round, and with N clients and T global communication rounds, the total uplink scales as $\mathcal{O}(NT)$. *Dual-branch* maintains global and local adapters but uploads only the global branch, so the uplink remains $\mathcal{O}(NT)$, while storage is typically $\mathcal{O}(2N)$ or $\mathcal{O}(N+1)$. ***HiLoRA*** follows a three-tier hierarchy with root, cluster, and leaf levels, and communicates only root and cluster. With T_{root} and T_{cluster} ($T_{\text{root}} + T_{\text{cluster}} \leq T$) denoting their respective round counts, the uplink scales as $\mathcal{O}\big(NT_{\text{root}} + NT_{\text{cluster}}\big)$ and storage as $\mathcal{O}(N + K^{\star} + 1)$ for $K^{\star}$ clusters, while leaf adapters remain local.