ALTERNATING LORA FOR DECENTRALIZED FINE-TUNING: A MISALIGNMENT–STALENESS TRADE-OFF

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-rank adaptation (LoRA) offers a communication-efficient approach to decentralized federated fine-tuning, but mixing independently trained LoRA factors can create discrepancies in their effective updates. Alternating factor updates mitigate a related aggregation problem in centralized federated learning, where server synchronization maintains a shared frozen factor; neighbor-only mixing does not provide this guarantee. We propose TAD-LoRA, which alternates local updates of the two factors in fixed-length phases while jointly mixing both factors at each communication round. Joint mixing reduces frozen-factor disagreement with the same per-round communication payload as standard decentralized LoRA. Under an idealized one-step stochastic-gradient model, we derive a finite-horizon bound for the loss of an averaged effective adapter. The bound separates a phase-averaged factor-product discrepancy that decreases with the switching interval from a delayed block-coverage term that increases with it; the minimizer of a loose upper-bound surrogate depends on network mixing. On four GLUE tasks, a matched-$T = 1$ ablation shows that joint mixing generally improves over active-factor-only mixing, while inter-client similarity measurements support improved factor alignment. Fixed-interval sweeps reveal that favorable switching regimes depend on network connectivity, with the largest best-observed gains over decentralized baselines under sparse communication. Additional experiments examine the method across multiple data-heterogeneity levels, communication topologies, and client counts.

1 INTRODUCTION

Fine-tuning pretrained language models on data distributed across organizations or devices motivates federated learning (FL), where clients collaborate without transferring their raw data. The size of these models makes parameter-efficient fine-tuning particularly attractive: low-rank adaptation (LoRA) Hu et al. trains two small factor matrices, A and B, instead of updating the full model. Its reduced trainable state also lowers the amount of information exchanged between clients.

Applying LoRA to FL nevertheless raises an aggregation problem. The effective weight update is the product BA, whereas clients typically exchange and aggregate the two factors separately. In general, the product of the aggregated factors differs from the average of the clients' effective updates. When local data are heterogeneous, the independently trained factors can disagree substantially across clients, making this discrepancy relevant to optimization. Existing federated LoRA methods address it through alternative aggregation schemes or by fixing one factor, but fixing a factor also restricts how the adapter can evolve.

Alternating optimization offers a way to update both factors without changing them simultaneously. In centralized FL (CFL), RoLoRA Chen et al. (2025) alternates which factor is trained and uses the server to synchronize the aggregated factor after each round. Consequently, clients begin the next update with a shared frozen factor. This property does not carry over directly to decentralized FL (DFL). Clients exchange parameters only with neighbors, and information takes multiple rounds to propagate through the network. Even with a common phase schedule, their frozen factors need not agree. If only the active factor is mixed, disagreement in the frozen factor persists and changes the context in which received updates are applied.

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.

We investigate this interaction between factor alternation and decentralized mixing. We propose TAD-LoRA, which locally updates one factor during each phase but mixes *both* factors at every communication round. The frozen factor therefore continues to move toward agreement across clients without receiving local gradient updates. This uses the same per-round factor payload as standard decentralized LoRA. Joint mixing alone, however, does not determine how often to switch factors: short phases leave limited time for network mixing, whereas long phases delay the co-adaptation of A and B. Network connectivity consequently affects the trade-off associated with the switching interval T.

Our contributions are as follows:

- We characterize why alternating LoRA with active-factor-only neighbor mixing can suffer from frozen-factor disagreement in DFL, even when all clients follow the same switching schedule.
- We introduce TAD-LoRA, combining alternating local factor updates with joint mixing of both factors. A comparison at the matched interval $T = 1$ examines the contribution of joint mixing independently of switching frequency.
- We prove that joint mixing contracts frozen-factor disagreement and, for an idealized one-step stochastic-gradient model, derive a finite-horizon bound for the loss of an averaged effective adapter. The bound exposes competing effects of factor-product discrepancy and delayed block coverage; a loose upper-bound surrogate suggests connectivity-dependent switching intervals.
- We evaluate fixed switching intervals on four GLUE tasks across data partitions and communication probabilities. The interval sweeps reveal connectivity-dependent performance regimes, while the largest best-observed gains over decentralized baselines arise under sparse communication.

2 Related Work

2.1 Federated and Decentralized Learning

FL enables training across clients with distributed data McMahan et al. (2017). Data heterogeneity, partial participation, and communication cost can all affect optimization Zhao et al. (2018); Li et al. (2020); Kairouz et al. (2021). Prior work addresses these challenges through compression Sattler et al. (2019), sparse updates Aji & Heafield (2017), and personalization Fallah et al. (2020). Decentralized FL removes the central aggregation server and instead relies on communication over a graph Lian et al. (2017); Lalitha et al. (2019); Wang et al. (2025). As a result, synchronized rounds do not imply that clients hold identical parameters. This incomplete consensus is central to our analysis. It differs from asynchronous FL, which also studies clients updating at different times Xie et al. (2019); our clients follow a shared round and phase schedule.

2.2 LoRA in Federated Learning

Federated LoRA methods address several distinct challenges. SLoRA Babakniya et al., FedSA-LoRA Guo et al. (2025), and pFedLoRA Yi et al. (2024) study heterogeneous or personalized adaptation. To address aggregation of low-rank factors, FLoRA Wang et al. stacks client adapters rather than averaging their factors, while FlexLoRA Bai et al. (2024) reconstructs updates and uses SVD to redistribute them. FFA-LoRA Sun et al. (2024) fixes one factor and trains the other, avoiding the factor-aggregation discrepancy but limiting which factor can adapt. These approaches offer different ways to handle aggregation or client heterogeneity; our focus is the interaction between alternating updates and incomplete neighbor consensus.

RoLoRA Chen et al. (2025) is the closest alternating method. It updates and aggregates one factor per round in CFL, where the server provides a common frozen factor to all clients. Our setting replaces server synchronization with peer-to-peer mixing. We examine the resulting frozen-factor disagreement, mix both factors during each communication step, and study how the phase length interacts with network connectivity. AltLoRA Yu et al. (2025) also uses alternation, but its single-machine alternating projections target gradient approximation rather than multi-client aggregation.

3 MOTIVATION AND ANALYSIS

LoRA models a weight update using a low-rank factorization $\Delta W = BA$. In federated learning, each client updates local *factors* (A_i, B_i), and aggregation is performed on each factor rather than their product:

$$A^{\text{agg}} = \sum_i w_i A_i, \qquad B^{\text{agg}} = \sum_i w_i B_i,$$

and reconstructing the update gives

$$B^{\text{agg}} A^{\text{agg}} = \underbrace{\sum_i w_i^2 B_i A_i}_{\text{desired average update}} + \underbrace{\sum_{i \neq j} w_i w_j \, B_i A_j}_{\text{cross-client interference}} .$$

The first term corresponds to the desired average update, while the second mixes B_i from one client with A_j from another—an update direction that no client ever computed. These cross terms are a primary source of instability when LoRA is trained on heterogeneous data.

Why alternating works in centralized FL. In centralized federated learning (CFL), alternating updates eliminate these cross terms Chen et al. (2025). During a B-phase, all clients share the same A, so $B_i A$ averages cleanly; during an A-phase, they share the same B, so BA_i averages cleanly. Server-enforced synchronization ensures that the frozen block is identical across clients, restoring the stability of two-block coordinate descent.

Challenges of Alternating LoRA in Decentralized Settings. In decentralized federated learning (DFL), this synchronization assumption no longer holds. Clients exchange parameters only with neighbors, so even within the same logical phase, their local models may differ due to incomplete information propagation. As a result, the frozen block is no longer shared across clients, and alternating updates can no longer fully suppress cross terms. This leads to two DFL-specific effects:

- **Phase-state mismatch:** clients enter the same phase with different versions of the frozen block.
- **Block-wise drift:** discrepancies in the frozen block accumulate over time, reintroducing cross-client interactions during mixing.

The issue can be illustrated by the following example:

$$\begin{aligned} &\text{client } i \text{ holds } A_i \text{ and receives } A_j \neq A_i \text{ from neighbors} \\ \Longrightarrow\ &\text{the B-phase can no longer eliminate } B_i A_j \text{ cross terms.} \end{aligned}$$

These observations motivate mixing the frozen factor and studying switching frequency as a separate design choice. Joint mixing contracts frozen-factor disagreement within each phase, whereas longer phases delay updates to the other factor. Their combined effect on final accuracy must be assessed empirically.

4 METHODOLOGY

The above analysis shows that decentralized communication disrupts the synchronization required for alternating LoRA to remain stable, mainly due to phase-state mismatch and block-wise drift resulting from time-varying peer-to-peer model mixing. To study and mitigate these effects, we first formalize the decentralized FL communication model, and then introduce our joint-mixing alternating LoRA procedure. An overview of the `TAD-LoRA` framework is at Fig 1

4.1 DECENTRALIZED FL COMMUNICATION MODEL

We adopt a standard decentralized FL (DFL) setting where clients maintain local parameters and exchange them with neighbors through a time-varying mixing matrix W_t:

$$x_i^{t+1} = \sum_{j=1}^{N} (W_t)_{ij} \, x_j^t.$$

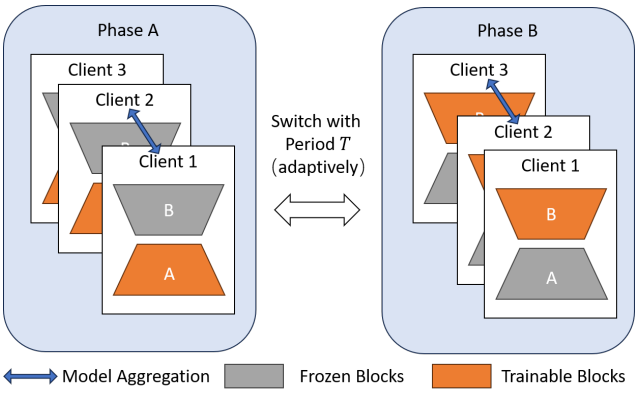

Figure 1: Overall illustration of TAD-LoRA under decentralized federated learning (DFL). Clients communicate directly with a dynamically selected subset of peers in a peer-to-peer manner, as determined by the underlying communication topology. At each alternating phase (lasting T rounds), the A or B block is actively updated and coordinated across clients, while the other remains frozen with respect to local optimization; both blocks are jointly mixed across clients after every round.

The matrices $\{W_t\}$ are *doubly-stochastic* and satisfy standard connectivity and spectral-gap assumptions used in decentralized optimization. All clients perform local updates every round. Communication is serverless: each round consists of (i) local computation and (ii) optional peer-to-peer mixing when communication is triggered. This contrasts with centralized FL (CFL), where a parameter server enforces identical global states at each round.

4.2 Alternating LoRA in FL

LoRA represents a low-rank update as $\Delta W = BA$, where A and B denote the down- and up-projection matrices. Alternating LoRA updates only one block per phase:

$$A_i^{t+1} = A_i^t - \eta \nabla_A \mathcal{L}_i(A_i^t, B_i^t), \qquad B_i^{t+1} = B_i^t,$$

during the A-phase, and symmetrically for the B-phase. This block splitting removes LoRA cross terms and stabilizes optimization.

CFL: Centralized Alternating LoRA (RoLoRA). In centralized FL, all clients follow identical A/B phases and the server conducts model aggregation through FedAvg:

$$A^{t+1} = \frac{1}{N}\sum_{i=1}^{m} A_i^{t+1}, \qquad B^{t+1} = \frac{1}{N}\sum_{i=1}^{m} B_i^{t+1}.$$

Since only the active block is aggregated in each phase, the centralized procedure behaves as standard two-block coordinate descent.

DFL: Decentralized Alternating LoRA Baseline. In DFL, aggregation is replaced by peer-to-peer mixing, but only the *active* block is synchronized. During an A-phase:

$$A_i^{t+1} = \sum_j (W_t)_{ij} A_j^t, \qquad B_i^{t+1} = B_i^t,$$

and during a B-phase:

$$B_i^{t+1} = \sum_j (W_t)_{ij} B_j^t, \qquad A_i^{t+1} = A_i^t.$$

Because the frozen block is not synchronized across clients during its inactive phase, clients gradually accumulate inconsistent versions of it, leading to *phase-state mismatch* and block-wise drift, the main failure mode of alternating LoRA under DFL.

Algorithm 1 TAD-LoRA: Joint-Mixing Alternating LoRA in Decentralized FL

Require: Initial blocks $\{(A_i^0, B_i^0)\}_{i=1}^m$; total rounds R; switching interval T; mixing matrices $\{W_t\}_{t=0}^{R-1}$.

for $t = 0, \dots, R-1$ **do**
  if $\lfloor t/T \rfloor$ is even **then**
    $\widetilde{A}_i^{t+1} \leftarrow A_i^t$
    $\widetilde{B}_i^{t+1} \leftarrow \mathrm{LocalUpdate}_B(A_i^t, B_i^t)$
  else
    $\widetilde{A}_i^{t+1} \leftarrow \mathrm{LocalUpdate}_A(A_i^t, B_i^t)$
    $\widetilde{B}_i^{t+1} \leftarrow B_i^t$
  end if
  $A_i^{t+1} \leftarrow \sum_{j=1}^m (W_t)_{ij} \widetilde{A}_j^{t+1}$
  $B_i^{t+1} \leftarrow \sum_{j=1}^m (W_t)_{ij} \widetilde{B}_j^{t+1}$
end for

4.3 TAD-LoRA: Joint-Mixing Alternating LoRA

To mitigate drift in decentralized alternating LoRA, we introduce `TAD-LoRA`, which synchronizes both LoRA blocks in every round, regardless of which block is being updated. After local updates, each client performs joint mixing:

$$A_i^{t+1} = \sum_j (W_t)_{ij} \widetilde{A}_j^{t+1}, \qquad B_i^{t+1} = \sum_j (W_t)_{ij} \widetilde{B}_j^{t+1}.$$

Joint mixing reduces the frozen block misalignment across the network, mitigating phase-state mismatch and stabilizing alternating LoRA under decentralized communication.

We study a decentralized learning process over R communication rounds, where each round $t \in \{0, \dots, R-1\}$ consists of a local update followed by a communication-induced mixing step. Algorithm 1 summarizes the full procedure.

Throughout the paper, we use the terms *iteration* and *round* interchangeably to denote one local update followed by a communication-induced mixing step.

5 Finite-Horizon Analysis of Switching Intervals

Appendix A analyzes an idealized version of `TAD-LoRA` with one unbiased stochastic-gradient step followed by joint factor mixing in each round. The analysis distinguishes the discrepancy between averaged effective LoRA updates and the product of averaged factors from the delay in updating both factors.

Theorem 5.1 (Finite-horizon trade-off (informal))**.** *Suppose the mixing matrices satisfy the conditional mean-square contraction assumption with coefficient $\rho \in (0,1)$, and the smoothness, gradient, and PL conditions in Appendix A hold. Let $\bar{W}_t = m^{-1} \sum_i B_i^t A_i^t$, let $\hat{t}$ be sampled uniformly from the R training rounds, and let $T \mid R$ with $R/T \geq 2$. Writing $\mathcal{D} = F(\bar{x}_0) - F_\star$, we have*

$$\mathbb{E}\big[\mathcal{L}(W_{\mathrm{base}} + \bar{W}_{\hat{t}}) - F_\star\big] \leq \underbrace{\frac{6\mathcal{D}}{\mu\eta R} + \frac{3L\eta G^2}{\mu} + \frac{3L^2\rho^2\eta^2 G^2}{\mu(1-\rho)^2}}_{\textit{optimization and persistent disagreement}} + \underbrace{\frac{H\rho^2\eta^2 G^2(1-\rho^T)}{T(1-\rho)^3}}_{\textit{factor-product discrepancy}} + \underbrace{\frac{2L^2\eta^2 G^2}{\mu} T^2}_{\textit{delayed coverage of both factors}}.$$

Here ρ describes expected contraction of inter-client disagreement. The bound concerns the population loss of the averaged effective adapter at a sampled training round.

Two effects of the switching interval. During a phase, joint mixing contracts disagreement in the frozen factor even though that factor receives no local gradient updates. This yields a phase-averaged upper bound on factor-product discrepancy that decreases with T. Longer phases, however, delay updates to the other factor. The resulting finite-horizon bound on full-gradient stationarity contains a term proportional to T^2. Neither bound asserts that test accuracy changes monotonically with T.

Network connectivity and interval choice. Replacing $1 - \rho^T$ by its upper bound of one gives a looser T-dependent surrogate of the form

$$U(T; \rho) = aT^2 + \frac{b\rho^2}{T(1-\rho)^3}, \qquad a, b > 0.$$

Its continuous minimizer scales as $T_{\mathrm{sur}}(\rho) \propto \rho^{2/3}/(1-\rho)$ when the other constants are fixed. Thus weaker expected mixing favors longer intervals in this surrogate. This minimizer does not identify the accuracy-optimal interval or provide a test-free selection rule. Our experiments instead examine a fixed candidate set of intervals under different communication conditions.

5.1 Experimental Setup

Model and data. We evaluate `TAD-LoRA` on four GLUE tasks: SST-2, QNLI, QQP, and MNLI. Following Sun et al. (2024) where applicable, we fine-tune RoBERTa-Large (335M parameters) with LoRA on the Q/V projections ($r = 8$, $\alpha = 16$, dropout 0.1), while keeping the classification head frozen. Data are partitioned across 10 clients. For each binary task, the client class proportions are $3 \times [0.9, 0.1]$, $3 \times [0.1, 0.9]$, and $4 \times [0.5, 0.5]$; for MNLI, they are $4 \times [0.9, 0.05, 0.05]$, $3 \times [0.05, 0.9, 0.05]$, and $3 \times [0.05, 0.05, 0.9]$.

Communication. At each round, we sample an undirected Erdős–Rényi communication graph Erdős & Rényi (1959), with each client pair connected independently with probability $p \in \{0.5, 0.2, 0.1, 0.05, 0.02, 0.01\}$. Clients mix parameters with their current neighbors. Results for structured topologies, including ring networks, are reported in the appendix.

Switching intervals. Training lasts $R = 150$ rounds. We evaluate $T \in \{1, 2, 3, 5, 10, 15\}$, all of which divide R and therefore end training at a phase boundary. For each task d and communication probability p, we report the best-observed interval $\hat{T}(p, d)$, selected retrospectively by the highest mean test accuracy over three seeds. These results characterize the evaluated fixed-interval performance landscape; they do not constitute a test-free interval-selection rule.

5.1.1 Baselines.

We compare `TAD-LoRA` with the following methods:

- `LoRA`: Standard LoRA fine-tuning under decentralized federated learning, where both LoRA matrices are locally trained and aggregated via FedAvg.
- `FFA-LoRA` Sun et al. (2024): A LoRA variant that freezes A and only updates B, effectively restricting adaptation to a fixed low-dimensional subspace.
- `RoLoRA` Chen et al. (2025): A centralized alternating LoRA method extended to DFL by aggregating only the active trainable parameters. Following the original paper, we use a per-round alternation, i.e., $T = 1$.
- `TAD-LoRA`: our topology-aware joint-mixing decentralized alternating LoRA method.

5.1.2 Training and Evaluation Details.

Each FL round uses **20 local steps**, for **150 rounds** total. We use AdamW (HuggingFace defaults), sequence length 128, batch size 32, and search learning rate over $\{2 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}, 2 \times 10^{-3}, 5 \times 10^{-3}\}$. All experiments run on NVIDIA L40S clusters. In DFL, no global model exists. For each seed, we evaluate all 10 client models and compute the **mean accuracy across clients**. We then report the **mean and standard deviation across random seeds** as the final performance.

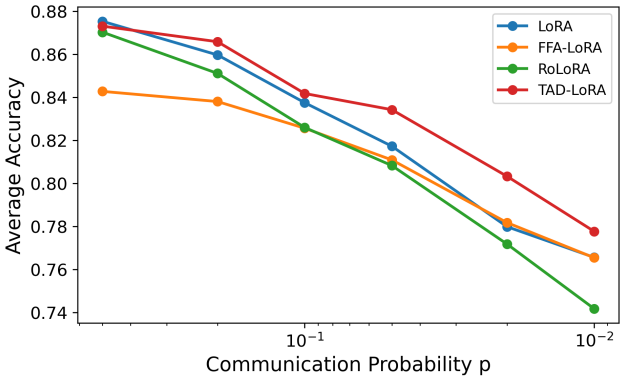

Figure 2: Mean test accuracy across four GLUE tasks under different communication probabilities p. For TAD-LoRA, the fixed interval is selected retrospectively for each task and p from the evaluated candidate set.

Table 1: Best-observed fixed-interval test accuracy under representative communication regimes (mean $\pm$ standard deviation over three seeds). For TAD-LoRA, T is selected retrospectively from $\{1, 2, 3, 5, 10, 15\}$ for each task and p.

p	Method	SST-2	QQP	QNLI	MNLI	Avg.
0.5	LoRA	**0.9468** $\pm$ 0.0019	**0.8347** $\pm$ 0.0085	**0.9067** $\pm$ 0.0043	0.8132 $\pm$ 0.0194	**0.8754**
	FFA-LoRA	0.9436 $\pm$ 0.0032	0.8051 $\pm$ 0.0061	0.8911 $\pm$ 0.0036	0.7313 $\pm$ 0.0333	0.8428
	RoLoRA	0.9462 $\pm$ 0.0023	0.8216 $\pm$ 0.0108	0.9021 $\pm$ 0.0031	0.8115 $\pm$ 0.0088	0.8703
	TAD-LoRA	0.9448 $\pm$ 0.0007	0.8328 $\pm$ 0.0072	0.9003 $\pm$ 0.0048	**0.8145** $\pm$ 0.0066	0.8731
0.1	LoRA	0.9370 $\pm$ 0.0058	**0.8098** $\pm$ 0.0076	0.8779 $\pm$ 0.0159	0.7253 $\pm$ 0.0440	0.8375
	FFA-LoRA	0.9313 $\pm$ 0.0046	0.7915 $\pm$ 0.0086	0.8710 $\pm$ 0.0048	0.7086 $\pm$ 0.0242	0.8256
	RoLoRA	0.9325 $\pm$ 0.0048	0.7890 $\pm$ 0.0027	0.8711 $\pm$ 0.0045	0.7113 $\pm$ 0.0133	0.8260
	TAD-LoRA	**0.9401** $\pm$ 0.0019	0.8050 $\pm$ 0.0073	**0.8815** $\pm$ 0.0145	**0.7405** $\pm$ 0.0376	**0.8418**
0.02	LoRA	0.8668 $\pm$ 0.0811	0.7702 $\pm$ 0.0066	0.8416 $\pm$ 0.0031	0.6407 $\pm$ 0.0118	0.7798
	FFA-LoRA	0.9186 $\pm$ 0.0038	0.7627 $\pm$ 0.0121	0.8264 $\pm$ 0.0178	0.6191 $\pm$ 0.0358	0.7817
	RoLoRA	0.9147 $\pm$ 0.0022	0.7582 $\pm$ 0.0116	0.8160 $\pm$ 0.0045	0.5980 $\pm$ 0.0291	0.7717
	TAD-LoRA	**0.9263** $\pm$ 0.0019	**0.7783** $\pm$ 0.0036	**0.8480** $\pm$ 0.0063	**0.6604** $\pm$ 0.0408	**0.8032**

5.2 Main Results

Performance across communication regimes. Figure 2 summarizes accuracy across the four tasks. Under dense communication ($p = 0.5$), TAD-LoRA is competitive but does not exceed standard decentralized LoRA on average. Its advantage grows as communication becomes sparser: at $p = 0.1$, its average accuracy is 0.8418, compared with 0.8375 for LoRA; at $p = 0.02$, it reaches 0.8032, compared with 0.7817 for the strongest baseline. The contrasting trends of RoLoRA and FFA-LoRA are qualitatively consistent with the switching–mixing trade-off: per-round alternation becomes less effective as communication weakens, while the method without factor switching becomes relatively more competitive. Because these baselines also differ in how they update and mix factors, this comparison alone does not isolate the effect of switching frequency.

Per-task results. Table 1 shows that the gains under sparse communication are not driven by a single task. At $p = 0.02$, TAD-LoRA has the highest mean accuracy on all four tasks. At $p = 0.1$, it leads on SST-2, QNLI, and MNLI, while LoRA remains stronger on QQP. At $p = 0.5$, LoRA has the highest overall average. Thus, the evidence supports the largest benefits under weak communication, rather than uniform superiority across topologies.

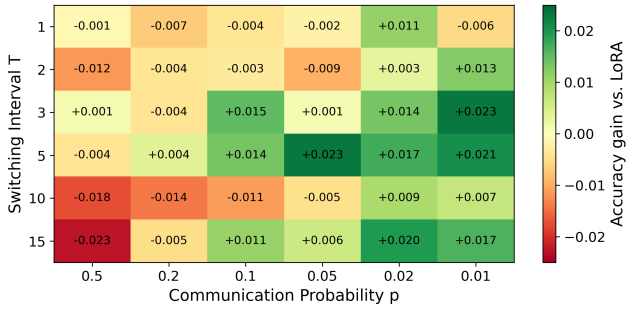

Figure 3: MNLI test-accuracy difference between TAD−LoRA and the decentralized LoRA baseline across communication probabilities p and fixed switching intervals T. Positive values favor TAD−LoRA.

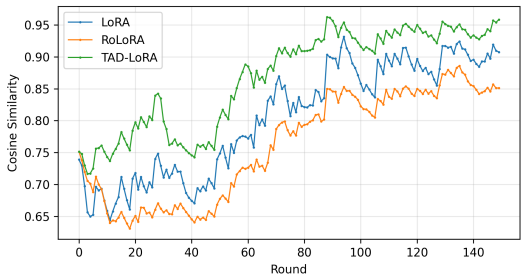

Figure 4: Mean inter-client cosine similarity of the LoRA factors during MNLI training at $p = 0.02$. TAD−LoRA uses a fixed switching interval $T = 15$; results are averaged over three seeds.

5.3 Ablation Study

Effect of the switching interval. Figure 3 shows that the effect of T depends strongly on communication probability. With dense communication ($p = 0.5$), long intervals reduce MNLI accuracy relative to the decentralized LoRA baseline. At $p = 0.1$, intermediate intervals such as $T = 3$ and $T = 5$ yield clear gains. Under sparser communication ($p = 0.02$), all evaluated intervals outperform the baseline, with the largest gain at $T = 15$. At $p = 0.01$, however, $T = 3$ performs better than $T = 15$, while switching every round ($T = 1$) falls below the baseline.

These results indicate that sparse communication can make less frequent switching beneficial and can broaden the range of effective intervals. They do not imply that the best interval increases monotonically for every task and every p. The dataset-wise best-observed intervals in Appendix C show an increasing cross-task median from $p = 0.5$ to $p = 0.02$, followed by a reversal at $p = 0.01$. Taken together, the response surface and the cross-task summary support a connectivity-dependent switching trade-off while also showing the limits of selecting an interval by a noisy, finite-grid argmax.

Factor alignment. To examine the misalignment mechanism, we track the mean inter-client cosine similarity of the two LoRA factors during training. Figure 4 compares TAD−LoRA with decentralized baselines on MNLI at $p = 0.02$. With the fixed interval $T = 15$, TAD−LoRA generally maintains stronger factor alignment. This is consistent with joint mixing: although only one factor receives local gradient updates in a phase, both factors continue to exchange information with neighboring clients, reducing disagreement in the frozen factor before the next switch.

The additional trajectories in Appendix D show faster and more persistent alignment with denser communication. The appendix also reports an association between similarity in the first 50 rounds and final accuracy across configurations. Because both p and T vary in that comparison, we treat the association as supporting diagnostic evidence rather than a causal estimate.

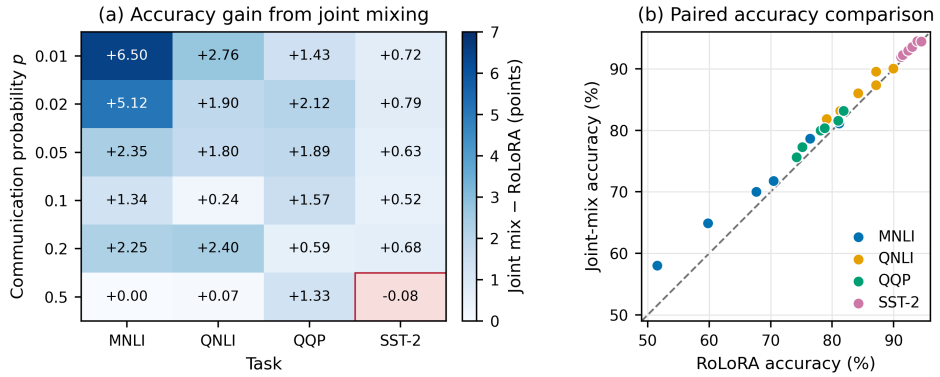

Figure 5: Effect of joint factor mixing at a fixed switching interval $T = 1$. (a) Difference in mean test accuracy (percentage points) between joint A/B mixing and RoLoRA-style active-factor-only mixing; positive values favor joint mixing. (b) Paired accuracies for the same task and communication probability; the dashed line indicates equal accuracy. Results use 10 clients, 150 rounds, and three seeds. Each method's learning rate is selected independently from the same candidate grid. The negative cell is highlighted in red.

Effect of joint mixing. At the matched switching interval $T = 1$, joint mixing improves mean accuracy over active-factor-only mixing in nearly all task–topology settings (Figure 5). The gains are largest for MNLI under sparse communication, reaching 6.50 percentage points at $p = 0.01$, while the methods are nearly tied in some dense settings. This comparison supports mixing the frozen factor as a useful component of decentralized alternation. It does not imply that per-round alternation is preferable to standard decentralized LoRA; the interval sweep in Figure 3 examines the separate effect of changing T.

5.4 Switching Interval Selection in Practice

In our experiments, the best switching intervals are selected in hindsight to characterize the performance landscape and isolate the effect of T. Importantly, our results show that under weak communication, a broad range of switching intervals yields consistent improvements, indicating robustness to imperfect T selection. A simple practical heuristic is to use smaller T under strong communication and gradually increase T as communication becomes weaker.

6 Conclusion

We introduced `TAD-LoRA` for decentralized federated fine-tuning. It alternates local updates of the two LoRA factors while mixing both factors after each communication round. Joint mixing reduces frozen-factor disagreement without increasing per-round communication relative to standard decentralized LoRA.

For an idealized one-step setting, we prove that joint mixing contracts disagreement in the frozen factor. Our finite-horizon analysis bounds the loss of an averaged effective adapter through two competing effects: factor-product discrepancy decreases with the switching interval, while delayed updates to the other factor increase the bound. The resulting upper-bound surrogate favors longer intervals as network mixing weakens.

Experiments on four GLUE tasks support the role of joint mixing: at a matched interval of $T = 1$, it generally improves on active-factor-only mixing, and inter-client similarity measurements show stronger factor alignment. Fixed-interval sweeps reveal different favorable regimes across communication conditions. With retrospectively selected intervals, `TAD-LoRA` remains competitive under dense communication and achieves its largest gains under sparse communication. Selecting an interval without test-based retrospective evaluation remains open.

AI USE STATEMENT

We used generative AI to assist with drafting and editing parts of the manuscript, including its organization, title, abstract, and descriptions of the results. We also used generative AI to help develop and draft the theoretical analysis, including mathematical claims, derivations, and proofs in Appendix A, and to critique the interpretation of experimental results. The authors developed the method and conducted the experiments, reviewed the AI-assisted analysis and text, and take responsibility for the final content of this paper. The authors developed the method, conducted the experiments, and reviewed the mathematical claims, reported results, references, and AI-assisted text. We take responsibility for the final content of this paper.

ETHICS STATEMENT

We evaluate the method on publicly available NLP benchmarks partitioned to simulate decentralized clients. The study does not recruit human participants or collect new private client data. Although the proposed training procedure exchanges model factors rather than raw examples, it does not provide a formal privacy guarantee. Applications involving sensitive client data would require additional privacy protections and an assessment appropriate to the deployment setting.

REPRODUCIBILITY STATEMENT

The method section specifies the factor-update, switching, and joint-mixing procedures. The assumptions and complete proofs of the theoretical results are provided in the appendix. The experimental setup documents the client data partitions, construction of communication graphs, learning-rate search range, random seeds, and the retrospective test-based selection of switching intervals used in the reported results. The appendix also reports additional experiments across data heterogeneity levels, client counts, and communication topologies.

REFERENCES

Alham Fikri Aji and Kenneth Heafield. Sparse communication for distributed gradient descent. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 440–445, 2017.

Sara Babakniya, Ahmed Roushdy Elkordy, Yahya H Ezzeldin, Qingfeng Liu, Kee-Bong Song, MOSTAFA EL-Khamy, and Salman Avestimehr. Slora: Federated parameter efficient fine-tuning of language models. In *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*.

Jiamu Bai, Daoyuan Chen, Bingchen Qian, Liuyi Yao, and Yaliang Li. Federated fine-tuning of large language models under heterogeneous tasks and client resources. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=gkOzoHBXUw.

Shuangyi Chen, Yuanxin Guo, Yue Ju, Harik Dalal, Zhongwen Zhu, and Ashish Khisti. Robust federated finetuning of llms via alternating optimization of lora. *arXiv preprint arXiv:2502.01755*, 2025.

P. Erdős and A. Rényi. On random graphs i. *Publ. math. debrecen*, 6(290-297):18, 1959.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in neural information processing systems*, 33:3557–3568, 2020.

Pengxin Guo, Shuang Zeng, Yanran Wang, Huijie Fan, Feifei Wang, and Liangqiong Qu. Selective aggregation for low-rank adaptation in federated learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=iX3uESGdsO.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021.

Anusha Lalitha, Osman Cihan Kilinc, Tara Javidi, and Farinaz Koushanfar. Peer-to-peer federated learning on graphs. *arXiv preprint arXiv:1901.11173*, 2019.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020.

Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. *Advances in neural information processing systems*, 30, 2017.

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.

Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems*, 31(9):3400–3413, 2019.

Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. Improving lora in privacy-preserving federated learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Xiaoyu Wang, Guojun Xiong, Houwei Cao, Jian Li, and Yong Liu. Decentralized federated learning with model caching on mobile agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 21296–21303, 2025.

Ziyao Wang, Zheyu Shen, Yexiao He, Guoheng Sun, Hongyi Wang, Lingjuan Lyu, and Ang Li. Flora: Federated fine-tuning large language models with heterogeneous low-rank adaptations. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Cong Xie, Sanmi Koyejo, and Indranil Gupta. Asynchronous federated optimization. *arXiv preprint arXiv:1903.03934*, 2019.

Liping Yi, Han Yu, Gang Wang, Xiaoguang Liu, and Xiaoxiao Li. pfedlora: Model-heterogeneous personalized federated learning with lora tuning, 2024. URL https://arxiv.org/abs/2310.13283.

Xin Yu, Yujia Wang, Jinghui Chen, and Lingzhou Xue. Altlora: Towards better gradient approximation in low-rank adaptation with alternating projections. *arXiv preprint arXiv:2505.12455*, 2025.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

A Detailed Theoretical Analysis

This appendix analyzes three properties of alternating factor updates with joint decentralized mixing: contraction of a frozen factor, the difference between averaging effective LoRA updates and multiplying averaged factors, and the finite-horizon delay in optimizing both factor blocks. The analysis uses an idealized one-step stochastic-gradient update in each communication round. It does not assert a convergence guarantee for the multi-step AdamW implementation used in the experiments.

A.1 Algorithm, objectives, and assumptions

There are m clients. Client i has factors $x_i = (A_i, B_i)$ and objective $f_i(x) = \mathcal{L}_i(W_{\text{base}} + BA)$. Write

$$F(x) = \frac{1}{m}\sum_{i=1}^{m} f_i(x) = \mathcal{L}(W_{\text{base}} + BA), \qquad \mathcal{L}(W) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}_i(W).$$

For clarity, notation displays one adapted weight matrix; the same factor-disagreement calculation applies componentwise to multiple adapted projections when their norms are stacked. The matrix X_t stacks the vectorized factor pairs x_i^t by client, and $J = \mathbf{1}\mathbf{1}^\top/m$. Define

$$\bar{x}_t = \frac{1}{m}\sum_i x_i^t, \qquad D_t = (I-J)X_t, \qquad D_{A,t} = (I-J)A_t, \quad D_{B,t} = (I-J)B_t.$$

All norms of stacked factors are Frobenius norms; $\|\cdot\|_2$ denotes the spectral norm of a mixing matrix. Let P_t project onto the active factor. The same factor is active at every client for T consecutive rounds, and the active factor switches after each such phase. If U_t stacks the projected stochastic gradients $P_t g_{i,t}$, one round is

$$X_{t+1} = W_t(X_t - \eta U_t), \qquad \bar{x}_{t+1} = \bar{x}_t - \eta P_t \bar{g}_t, \quad \bar{g}_t = \frac{1}{m}\sum_i g_{i,t}. \tag{1}$$

In particular, the local update precedes mixing, and both factor blocks are mixed in W_t even though only one is locally updated.

The following conditions make the scope of the bounds explicit.

Assumption A.1 (Smoothness, gradients, and initialization)**.** Each f_i and F has L-Lipschitz gradients on a domain containing the iterates and the line segments used below. The global objective is bounded below by $F_\star$. Conditional on the current iterates, stochastic gradients satisfy

$$\mathbb{E}[g_{i,t} \mid X_t] = \nabla f_i(x_i^t), \qquad \mathbb{E}\|G_t\|_F^2 \le mG^2,$$

where G_t stacks the unprojected $g_{i,t}$. All clients initialize from identical LoRA factors, so $D_0 = 0$. The moment bound includes possible data heterogeneity; gradients at different clients need not have the same conditional mean.

Assumption A.2 (Conditional network contraction)**.** Each W_t is a nonnegative doubly-stochastic mixing matrix, sampled independently of the pre-mixing local update conditional on the past. For every centered, pre-mixing client matrix Z,

$$\mathbb{E}\big[\|(W_t - J)Z\|_F^2 \mid Z\big] \le \rho^2 \|Z\|_F^2, \qquad 0 < \rho < 1. \tag{2}$$

Individual communication graphs may be disconnected. The condition is an expected contraction of each disagreement vector; it is distinct from a bound on $\mathbb{E}\|W_t - J\|_2^2$.

The finite-horizon stationarity result below uses no PL condition. For its function-value counterpart, assume additionally that all attained network averages $\bar{x}_t$ obey

$$\|\nabla F(\bar{x}_t)\|^2 \ge 2\mu\big(F(\bar{x}_t) - F_\star\big), \qquad \mu > 0. \tag{3}$$

This is a substantive assumption about the relevant LoRA loss region, not a property established for pretrained language models. Finally, when bounding the loss of an averaged effective adapter, we assume $\mathcal{L}$ is H-Lipschitz on the relevant effective weight matrices.

A.2 Consensus of the two factor blocks

Lemma A.3 (Joint-mixing consensus)**.** *Under Assumptions A.1 and A.2, let*

$$d_{\max} := \frac{\rho\eta\sqrt{m}\,G}{1-\rho}.$$

Then $\big(\mathbb{E}\|D_t\|_F^2\big)^{1/2} \le d_{\max}$ *for every round. If factor* $H \in \{A, B\}$ *is frozen for* s *consecutive rounds beginning at round* t_0*, then*

$$\mathbb{E}\|D_{H,t_0+s}\|_F^2 \le \rho^{2s}\mathbb{E}\|D_{H,t_0}\|_F^2. \tag{4}$$

Proof. Double stochasticity gives $W_t J = J W_t = J$. Consequently,

$$D_{t+1} = (W_t - J)\big(D_t - \eta(I - J)U_t\big).$$

Let $d_t = (\mathbb{E}\|D_t\|_F^2)^{1/2}$. Apply the conditional contraction in equation 2, Minkowski's inequality in L^2, the fact that $I - J$ and P_t are projections, and the gradient moment bound:

$$d_{t+1} \le \rho d_t + \rho\eta(\mathbb{E}\|U_t\|_F^2)^{1/2} \le \rho d_t + \rho\eta\sqrt{m}\,G.$$

Starting from $d_0 = 0$ and summing the geometric series yields $d_t \le \rho\eta\sqrt{m}\,G(1-\rho^t)/(1-\rho) \le d_{\max}$. For a frozen block H, its rows receive no local gradient step; $D_{H,t+1} = (W_t - J)D_{H,t}$. Apply the conditional contraction successively for s rounds to obtain equation 4. □

The frozen block in an active-factor-only mixing baseline stays fixed throughout its phase, so its client disagreement does not contract through communication during that phase. This establishes the specific consensus benefit of mixing the frozen factor; it does not by itself establish an accuracy improvement.

A.3 The factor-product discrepancy

Let $\bar{A}_t = m^{-1}\sum_i A_i^t$ and $\bar{B}_t = m^{-1}\sum_i B_i^t$. The average effective LoRA update and its discrepancy from the product of average factors are

$$\bar{W}_t := \frac{1}{m}\sum_i B_i^t A_i^t = \bar{B}_t\bar{A}_t + C_t, \qquad C_t = \frac{1}{m}\sum_i (B_i^t - \bar{B}_t)(A_i^t - \bar{A}_t). \tag{5}$$

By Cauchy–Schwarz for matrix products and for expectations,

$$\mathbb{E}\|C_t\|_F \le \frac{1}{m}\sqrt{\mathbb{E}\|D_{A,t}\|_F^2}\sqrt{\mathbb{E}\|D_{B,t}\|_F^2}. \tag{6}$$

The factor $1/m$ follows from using unnormalized stacked-client Frobenius norms for $D_{A,t}$ and $D_{B,t}$.

Proposition A.4 (Phase-average discrepancy)**.** *For a complete phase of length T starting at t_0,*

$$\frac{1}{T}\sum_{s=0}^{T-1}\mathbb{E}\|C_{t_0+s}\|_F \le \frac{d_{\max}^2}{m}\frac{1-\rho^T}{T(1-\rho)} = \frac{\rho^2\eta^2G^2(1-\rho^T)}{T(1-\rho)^3}. \tag{7}$$

If T divides the total number of rounds R, the same bound holds for the time average over all R rounds.

Proof. In each phase one factor is frozen. At its sth round, Lemma A.3 bounds its root mean-square disagreement by $\rho^s d_{\max}$, while the other factor's root mean-square disagreement is at most $d_{\max}$. Insert these bounds in equation 6 and sum the geometric series $\sum_{s=0}^{T-1}\rho^s = (1-\rho^T)/(1-\rho)$. Every round belongs to exactly one complete phase when $T \mid R$. □

This is an upper bound on the time-averaged product discrepancy. It does not assert that discrepancy or final test accuracy changes monotonically with T in every run.

A.4 Finite-horizon optimization of the averaged factors

The exact averaged factor update in equation 1 allows a descent argument for $F(\bar{x}_t)$. It does not imply that the average effective weight $\bar{W}_t$ itself follows a gradient update. Define the expected gradient bias from client disagreement:

$$e_t := \frac{1}{m}\sum_i \nabla f_i(x_i^t) - \nabla F(\bar{x}_t).$$

Client smoothness and Jensen's inequality imply

$$\mathbb{E}\|e_t\|^2 \le \frac{L^2}{m}\mathbb{E}\|D_t\|_F^2 \le \frac{L^2 d_{\max}^2}{m}. \tag{8}$$

Lemma A.5 (Descent on the active block)**.** *For each round,*

$$\mathbb{E}F(\bar{x}_{t+1}) \le \mathbb{E}F(\bar{x}_t) - \frac{\eta}{2}\mathbb{E}\|P_t\nabla F(\bar{x}_t)\|^2 + \frac{\eta}{2}\mathbb{E}\|e_t\|^2 + \frac{L\eta^2G^2}{2}. \tag{9}$$

Proof. Apply L-smoothness to $\bar{x}_{t+1} = \bar{x}_t - \eta P_t\bar{g}_t$. Conditional unbiasedness gives $\mathbb{E}[\bar{g}_t \mid X_t] = \nabla F(\bar{x}_t) + e_t$. The inner product involving e_t is bounded by $|\langle P_t\nabla F, P_te_t\rangle| \le (\|P_t\nabla F\|^2 + \|P_te_t\|^2)/2$. Finally, Jensen's inequality and the gradient second-moment bound give $\mathbb{E}\|P_t\bar{g}_t\|^2 \le G^2$. □

An inactive gradient cannot be bounded by the same-round active gradient. Pairing an A-phase round with the corresponding round of the following B-phase gives a finite-horizon alternative.

Lemma A.6 (Coverage of both blocks)**.** *Suppose $T \mid R$ and $R/T \ge 2$. Across adjacent phases,*

$$\frac{1}{R}\sum_{t=0}^{R-1}\mathbb{E}\|\nabla F(\bar{x}_t)\|^2 \le \frac{6}{R}\sum_{t=0}^{R-1}\mathbb{E}\|P_t\nabla F(\bar{x}_t)\|^2 + 4L^2\eta^2G^2T^2. \tag{10}$$

Proof. For each adjacent pair of phases $j, j+1$, pair $t = jT + s$ with $t' = t + T$, $0 \le s < T$. One member updates A, the other updates B (or vice versa). Lipschitz gradients give

$$\|\nabla F(\bar{x}_t)\|^2 + \|\nabla F(\bar{x}_{t'})\|^2 \le 3\big(\|P_t\nabla F(\bar{x}_t)\|^2 + \|P_{t'}\nabla F(\bar{x}_{t'})\|^2\big) + 4L^2\|\bar{x}_{t'} - \bar{x}_t\|^2.$$

The exact mean update and Cauchy–Schwarz imply

$$\mathbb{E}\|\bar{x}_{t'} - \bar{x}_t\|^2 \le \eta^2T\sum_{r=t}^{t'-1}\mathbb{E}\|P_r\bar{g}_r\|^2 \le \eta^2T^2G^2.$$

Let S_j and A_j be the respective sums of full and active gradient squares over phase j. The preceding inequality gives $S_j + S_{j+1} \le 3(A_j + A_{j+1}) + 4L^2\eta^2G^2T^3$. Sum this inequality over all adjacent phase pairs. Each interior phase appears twice and each endpoint once. Since $S_j, A_j \ge 0$ and there are $R/T - 1$ such pairs, the total full sum is at most six times the total active sum plus $4L^2\eta^2G^2(R/T-1)T^3 \le 4L^2\eta^2G^2RT^2$. Dividing by R proves equation 10. □

Theorem A.7 (Finite-horizon factor stationarity)**.** *Under Assumptions A.1 and A.2, if $T \mid R$, $R/T \ge 2$, and $\mathcal{D} := F(\bar{x}_0) - F_\star$, then*

$$\frac{1}{R}\sum_{t=0}^{R-1}\mathbb{E}\|\nabla F(\bar{x}_t)\|^2 \le \frac{12\mathcal{D}}{\eta R} + \frac{6L^2\rho^2\eta^2G^2}{(1-\rho)^2} + 6L\eta G^2 + 4L^2\eta^2G^2T^2. \tag{11}$$

Proof. Sum equation 9 over all rounds and use $F(\bar{x}_R) \ge F_\star$. This gives

$$\frac{1}{R}\sum_t\mathbb{E}\|P_t\nabla F(\bar{x}_t)\|^2 \le \frac{2\mathcal{D}}{\eta R} + \frac{1}{R}\sum_t\mathbb{E}\|e_t\|^2 + L\eta G^2.$$

Use equation 8, substitute $d_{\max}^2/m = \rho^2\eta^2G^2/(1-\rho)^2$, and then apply Lemma A.6. □

The term proportional to T^2 is a finite-horizon penalty for waiting to update the other factor. It is not a claim that a fixed-T alternating algorithm has a different asymptotic optimum.

Theorem A.8 (Loss of the averaged effective adapter)**.** *Assume in addition equation 3 and the H-Lipschitz condition for $\mathcal{L}$. Let $\widehat{t}$ be uniform on $\{0, \dots, R-1\}$ and independent of the algorithm. For $T \mid R$ and $R/T \ge 2$,*

$$\begin{aligned}\mathbb{E}\left[\mathcal{L}\left(W_{\text{base}} + \frac{1}{m}\sum_i B_i^{\widehat{t}}A_i^{\widehat{t}}\right) - F_\star\right] \\ \le \frac{6\mathcal{D}}{\mu\eta R} + \frac{3L^2\rho^2\eta^2G^2}{\mu(1-\rho)^2} + \frac{3L\eta G^2}{\mu} + \underbrace{\frac{2L^2\eta^2G^2}{\mu}T^2}_{\textit{block coverage}} + \underbrace{\frac{H\rho^2\eta^2G^2(1-\rho^T)}{T(1-\rho)^3}}_{\textit{factor-product discrepancy}}.\end{aligned} \tag{12}$$

Proof. The PL condition and Theorem A.7 bound $R^{-1}\sum_t \mathbb{E}[F(\bar{x}_t) - F_\star]$ by the first four terms on the right of equation 12. By equation 5, $W_{\text{base}} + \bar{W}_t = W_{\text{base}} + \bar{B}_t\bar{A}_t + C_t$. Thus the Lipschitz condition implies

$$\mathcal{L}(W_{\text{base}} + \bar{W}_t) \le F(\bar{x}_t) + H\|C_t\|_F.$$

Average over t and apply Proposition A.4. □

The model in this theorem uses an average of client adapters' effective weights. The experiments evaluate each client and average test accuracy. Thus the theorem bounds a specified population loss; it does not bound the reported test accuracy or guarantee improvement over standard decentralized LoRA.

A.5 Interpreting the switching interval

Dropping the factor $1 - \rho^T \le 1$ in Theorem A.8 yields a looser upper bound with the following T-dependent part:

$$U(T;\rho) = \frac{2L^2\eta^2G^2}{\mu}T^2 + \frac{H\rho^2\eta^2G^2}{T(1-\rho)^3}. \tag{13}$$

When $L, H, G > 0$ and $0 < \rho < 1$, its continuous minimizer is

$$T_{\text{sur}}(\rho) = \left(\frac{H\mu\rho^2}{4L^2(1-\rho)^3}\right)^{1/3}. \tag{14}$$

This expression increases with ρ and is of order $(1-\rho)^{-1}$ as $\rho \uparrow 1$, assuming the other constants are fixed. For a finite training horizon, only admissible integer intervals with $T \mid R$ and $R/T \ge 2$ are covered by the adjacent-phase proof; they can be compared using equation 13.

The minimizer of this loose bound need not minimize the tighter bound equation 12, final model loss, or test accuracy.

A.6 A concrete edge-activation mixing example

The relation between an edge activation probability p and ρ depends on the weights assigned to the activated edges. The following lemma applies to one explicit, simultaneous lazy-Laplacian rule. It should be used to interpret an experimental implementation only if that implementation uses this rule.

Lemma A.9 (Independent edge activation with lazy Laplacian mixing)**.** *Let $G = (V, E)$ be a fixed, connected undirected graph with Laplacian L and maximum degree $d_{\max}^{\text{graph}}$. Each edge is active independently with probability $p \in (0, 1]$ in each round. Let L_t be the Laplacian of the active edges and set*

$$W_t = I - \alpha L_t, \qquad 0 < \alpha \le \frac{1}{2d_{\max}^{\text{graph}}}.$$

Then W_t is nonnegative and doubly stochastic. For every centered client matrix Z,

$$\mathbb{E}\|W_tZ\|_F^2 \le \big(1 - \alpha p\lambda_2(L)\big)\|Z\|_F^2. \tag{15}$$

Consequently, Assumption A.2 holds with $\rho^2 = 1 - \alpha p\lambda_2(L)$, and

$$1 - \rho \ge \frac{\alpha p\lambda_2(L)}{2}. \tag{16}$$

Proof. The degree bound ensures that the diagonal entries of W_t are nonnegative; its off-diagonal entries are nonnegative and $W_t\mathbf{1} = \mathbf{1}$. Symmetry gives double stochasticity. For every active subgraph, $\lambda_{\max}(L_t) \le 2d_{\max}^{\text{graph}}$. The eigenvalues of W_t therefore lie in $[0, 1]$, so $W_t^2 \preceq W_t$. Moreover, $\mathbb{E}L_t = pL$. For a centered Z, the Rayleigh bound gives

$$\mathbb{E}\|W_tZ\|_F^2 = \operatorname{tr}\big(Z^\top\mathbb{E}[W_t^2]Z\big) \le \operatorname{tr}\big(Z^\top(I - \alpha pL)Z\big) \le (1 - \alpha p\lambda_2(L))\|Z\|_F^2.$$

Finally, set $q = \alpha p\lambda_2(L) \in (0, 1]$ and use $1 - \sqrt{1-q} = q/(1 + \sqrt{1-q}) \ge q/2$. □

For the complete underlying graph, $\lambda_2(L) = m$. More generally, the example shows how greater activation probability can improve the expected vector-contraction factor for this mixing rule. It does not imply that an arbitrary degree-normalized neighbor average has the same $\rho(p)$ or that the optimal empirical switching interval is monotone in p.

A.7 Scope and reproducibility of the analysis

The contraction and finite-horizon results require the explicitly stated mixing weights, gradient conditions, and update order. Theorem A.8 concerns a uniform random training round and the loss of the averaged effective adapter. It does not cover the final-round client-averaged accuracy used in the experiments. Neither partial client participation nor multi-step AdamW local optimization is established by this proof. Reporting the actual mixing weights and local optimization protocol is therefore necessary to assess which theoretical assumptions apply to a reproduction.

Table 2: Full results under different communication probabilities. Test accuracy is reported as mean $\pm$ standard deviation over 3 random seeds. Best and second-best results for each dataset under the same communication probability p are highlighted in bold and underline, respectively.

p	Method	SST-2	QQP	QNLI	MNLI	Avg.
0.5	LoRA	**0.9468** $\pm$ 0.0019	**0.8347** $\pm$ 0.0085	**0.9067** $\pm$ 0.0043	0.8132 $\pm$ 0.0194	**0.8754**
	FFA-LoRA	0.9436 $\pm$ 0.0032	0.8051 $\pm$ 0.0061	0.8911 $\pm$ 0.0036	0.7313 $\pm$ 0.0333	0.8428
	RoLoRA	0.9462 $\pm$ 0.0023	0.8216 $\pm$ 0.0108	0.9021 $\pm$ 0.0031	0.8115 $\pm$ 0.0088	0.8703
	TAD-LoRA	0.9448 $\pm$ 0.0007	0.8328 $\pm$ 0.0072	0.9003 $\pm$ 0.0048	**0.8145** $\pm$ 0.0066	0.8731
0.2	LoRA	0.9451 $\pm$ 0.0059	0.8181 $\pm$ 0.0063	0.8921 $\pm$ 0.0162	0.7836 $\pm$ 0.0139	0.8597
	FFA-LoRA	0.9365 $\pm$ 0.0025	0.7996 $\pm$ 0.0047	0.8859 $\pm$ 0.0032	0.7300 $\pm$ 0.0285	0.8380
	RoLoRA	0.9407 $\pm$ 0.0017	0.8119 $\pm$ 0.0142	0.8826 $\pm$ 0.0065	0.7693 $\pm$ 0.0140	0.8511
	TAD-LoRA	**0.9453** $\pm$ 0.0038	**0.8254** $\pm$ 0.0019	**0.8953** $\pm$ 0.0030	**0.7974** $\pm$ 0.0107	**0.8659**
0.1	LoRA	0.9370 $\pm$ 0.0058	**0.8098** $\pm$ 0.0076	0.8779 $\pm$ 0.0159	0.7253 $\pm$ 0.0440	0.8375
	FFA-LoRA	0.9313 $\pm$ 0.0046	0.7915 $\pm$ 0.0086	0.8710 $\pm$ 0.0048	0.7086 $\pm$ 0.0242	0.8256
	RoLoRA	0.9325 $\pm$ 0.0048	0.7890 $\pm$ 0.0027	0.8711 $\pm$ 0.0045	0.7113 $\pm$ 0.0133	0.8260
	TAD-LoRA	**0.9401** $\pm$ 0.0019	0.8050 $\pm$ 0.0073	**0.8815** $\pm$ 0.0145	**0.7405** $\pm$ 0.0376	**0.8418**
0.05	LoRA	0.9309 $\pm$ 0.0047	0.7720 $\pm$ 0.0226	0.8598 $\pm$ 0.0157	0.7014 $\pm$ 0.0139	0.8173
	FFA-LoRA	0.9263 $\pm$ 0.0043	0.7814 $\pm$ 0.0145	0.8659 $\pm$ 0.0147	0.6699 $\pm$ 0.0079	0.8109
	RoLoRA	0.9238 $\pm$ 0.0036	0.7851 $\pm$ 0.0089	0.8424 $\pm$ 0.0129	0.6818 $\pm$ 0.0130	0.8083
	TAD-LoRA	**0.9328** $\pm$ 0.0033	**0.7965** $\pm$ 0.0118	**0.8743** $\pm$ 0.0130	**0.7333** $\pm$ 0.0126	**0.8342**
0.02	LoRA	0.8668 $\pm$ 0.0811	0.7702 $\pm$ 0.0066	0.8416 $\pm$ 0.0031	0.6407 $\pm$ 0.0118	0.7798
	FFA-LoRA	0.9186 $\pm$ 0.0038	0.7627 $\pm$ 0.0121	0.8264 $\pm$ 0.0178	0.6191 $\pm$ 0.0358	0.7817
	RoLoRA	0.9147 $\pm$ 0.0022	0.7582 $\pm$ 0.0116	0.8160 $\pm$ 0.0045	0.5980 $\pm$ 0.0291	0.7717
	TAD-LoRA	**0.9263** $\pm$ 0.0019	**0.7783** $\pm$ 0.0036	**0.8480** $\pm$ 0.0063	**0.6604** $\pm$ 0.0408	**0.8032**
0.01	LoRA	0.9182 $\pm$ 0.0103	0.7444 $\pm$ 0.0081	0.8074 $\pm$ 0.0196	0.5921 $\pm$ 0.0029	0.7655
	FFA-LoRA	0.9123 $\pm$ 0.0040	0.7555 $\pm$ 0.0132	0.8178 $\pm$ 0.0080	0.5759 $\pm$ 0.0242	0.7654
	RoLoRA	0.9120 $\pm$ 0.0014	0.7420 $\pm$ 0.0209	0.7974 $\pm$ 0.0143	0.5153 $\pm$ 0.0110	0.7417
	TAD-LoRA	**0.9190** $\pm$ 0.0024	**0.7605** $\pm$ 0.0139	**0.8218** $\pm$ 0.0087	**0.6092** $\pm$ 0.0178	**0.7776**

B Additional Experimental Results

Additional Communication Probabilities. Table 2 consolidates the results across all evaluated communication probabilities, including additional settings beyond those emphasized in the main paper. The same evaluation protocol as in the main paper is used, reporting test accuracy as mean $\pm$ standard deviation over 3 random seeds.

These additional results provide a more complete picture of how different methods behave across a wide range of communication regimes. As the communication probability decreases, performance degradation is observed for all methods, which is expected due to reduced model exchanges. Nevertheless, TAD-LoRA consistently achieves the strongest performance under low communication probabilities, not only in terms of the average across datasets, but also on individual tasks.

Best and second-best results for each dataset under the same communication probability are highlighted to emphasize per-dataset robustness. This demonstrates that the gains of TAD-LoRA under sparse communication are consistent across heterogeneous tasks, rather than being driven by a single dataset.

B.1 Weak-Regime Performance Summary

We define the weak communication regime as $p \leq 0.05$. Table 3 reports the average performance obtained by uniformly averaging the per-p average accuracy across datasets. This summary complements the main results by highlighting the robust advantage of TAD-LoRA under communication-constrained settings.

Table 3: Average test accuracy in the weak-communication regime ($p \leq 0.05$). The average is computed over all evaluated datasets and communication probabilities $p \in \{0.05, 0.02, 0.01\}$. The best result in each block is highlighted in bold.

Method	Weak-regime avg. accuracy
LoRA	0.7875
FFA-LoRA	0.7860
RoLoRA	0.7739
TAD-LoRA	**0.8050**

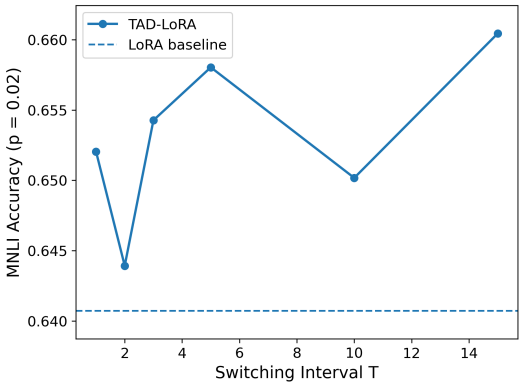

Figure 6: Illustrative example of the non-monotonic dependence on the switching interval T on MNLI under a weak communication regime ($p = 0.02$). The y-axis reports the accuracy gain of TAD-LoRA over the LoRA baseline. Due to discrete scheduling, finite training budgets, and optimization noise, the theoretical trade-off appears empirically as a noisy, non-monotonic dependence on T, with several competitive intervals rather than a single sharp optimum.

B.2 Illustrative non-monotonic behavior.

Figure 6 provides a representative example of the non-monotonic dependence on the switching interval predicted by our theory. In practice, the idealized U-shaped behavior is obscured by training noise, discrete alternation schedules, and finite training budgets, resulting in a range of effective switching intervals rather than a single sharp optimum. This observation is consistent with the aggregate trends and heatmap visualizations reported in the main paper.

B.3 Robustness across Data Heterogeneity Settings

We follow the data partition protocol in prior work Sun et al. (2024) and evaluate three representative heterogeneity levels: IID, Mild Heterogeneous, and Severe Heterogeneous. For MNLI, the Mild setting assigns label proportions following permutations of $[0.6, 0.2, 0.2]$ across clients, while the Severe setting uses permutations of $[0.9, 0.05, 0.05]$. The IID setting samples local data uniformly from the global training distribution.

Our main experiments focus on the Severe Heterogeneous setting, since it represents the most challenging regime for decentralized federated learning: client objectives are highly biased, and sparse peer-to-peer communication further amplifies topology-induced model mismatch. To verify that the improvement of TAD-LoRA is not specific to this difficult setting, we additionally evaluate MNLI under Mild Heterogeneous and IID partitions across different communication probabilities. As shown in Table 5, TAD-LoRA achieves the best or near-best performance across the full spectrum of data heterogeneity, with particularly clear gains under sparse communication and severe heterogeneity.

Table 4: Best-observed fixed switching interval $\hat{T}(p, d)$ for each dataset d and communication probability p, selected from $T \in \{1, 2, 3, 5, 10, 15\}$.

p	SST-2	QQP	QNLI	MNLI	Mean	Range
0.5	1	1	1	3	1.50	1–3
0.2	1	5	1	5	3.00	1–5
0.1	5	3	3	3	3.50	3–5
0.05	5	5	15	5	7.50	5–15
0.02	3	1	15	15	8.50	1–15
0.01	10	5	3	3	5.25	3–10

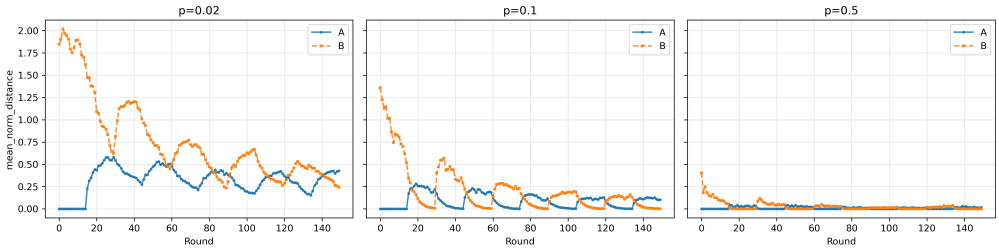

Figure 7: Evolution of A- and B-factor normalized distance under fixed switching interval $T = 15$ on MNLI. Each subplot corresponds to a different communication probability p. The figure shows the complementary dynamics between the two LoRA factors across communication regimes.

B.4 Scalability to More Clients

Client scalability. To examine whether the advantage of TAD−LoRA persists beyond the 10-client setting, we further evaluate MNLI under a larger 30-client decentralized network. We consider two scaling protocols. First, we keep the same edge probability p as in the 10-client experiments. In this case, the expected degree naturally increases with the number of clients, leading to a denser communication graph. Second, we reduce the edge probability in the 30-client setting to approximately maintain the same expected communication degree as the corresponding 10-client setting, resulting in a more stringent scalability test.

As shown in Table 6, TAD−LoRA remains consistently strong when scaling from 10 to 30 clients. Under the same edge probability, TAD-LoRA achieves the best accuracy across all communication regimes. More importantly, under the same-degree setting, where the 30-client graph becomes substantially sparser in terms of edge probability, TAD-LoRA still outperforms all baselines. The advantage is especially pronounced in the most communication-limited case, where scaling from $p = 0.02$ with 10 clients to $p = 0.007$ with 30 clients causes all methods to degrade, but TAD−LoRA retains a significantly higher accuracy than standard LoRA, FFA-LoRA, and RoLoRA.

B.5 Additional Network Topologies

B.5.1 Ring Topology

Beyond the representative random (Erdős–Rényi) communication topology considered in the main results, we further evaluate TAD−LoRA under a ring topology, which represents a highly structured network with limited connectivity and slow information mixing. This setting serves as an extreme stress test rather than a typical deployment scenario.

As shown in Table 7, under the ring topology, TAD−LoRA remains competitive with LoRA on average and clearly outperforms FFA-LoRA and RoLoRA, while achieving the best MNLI result. This suggests that the proposed alternating strategy does not rely exclusively on Erdős–Rényi communication graphs, although performance gains are less pronounced under this structured topology.

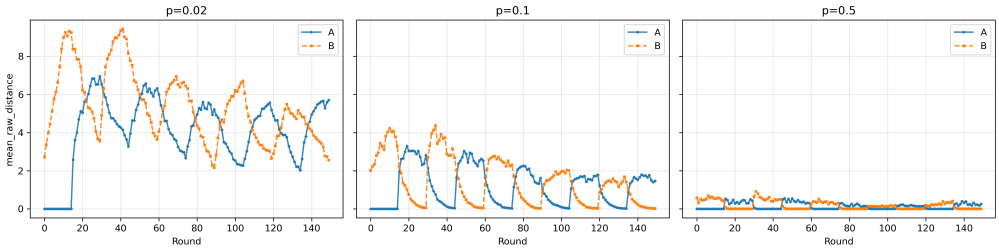

Figure 8: Evolution of A- and B-factor raw distance under fixed switching interval $T = 15$ on MNLI. Each subplot corresponds to a different communication probability p. Similar to normalized distance, the two factors exhibit alternating distance patterns during decentralized training.

Table 5: MNLI accuracy under different communication probabilities and data heterogeneity settings. Avg. denotes the average accuracy across the three heterogeneity settings. The best result in each block is highlighted in bold.

Comm. Prob.	Method	Severe Non-IID	Mild Non-IID	IID	Avg.
$p = 0.5$	LoRA	0.8132	0.8452	0.8452	0.8346
	FFA-LoRA	0.7313	0.8197	0.8237	0.7915
	RoLoRA	0.8115	0.8473	0.8486	0.8358
	TAD-LoRA	**0.8145**	**0.8479**	**0.8502**	**0.8376**
$p = 0.1$	LoRA	0.7253	0.8437	**0.8518**	0.8069
	FFA-LoRA	0.7086	0.8221	0.8267	0.7858
	RoLoRA	0.7113	0.8353	0.8392	0.7953
	TAD-LoRA	**0.7405**	**0.8454**	0.8492	**0.8117**
$p = 0.02$	LoRA	0.6407	0.8137	0.8345	0.7630
	FFA-LoRA	0.6191	0.8030	0.8136	0.7452
	RoLoRA	0.5980	0.7983	0.8196	0.7386
	TAD-LoRA	**0.6604**	**0.8257**	**0.8375**	**0.7745**

C Best-Observed Switching Intervals Across Tasks

Figure 9 shows a broad shift toward longer best-observed intervals as communication weakens from $p = 0.5$ to $p = 0.02$: the cross-task median increases from 1 to 9. The trend reverses at $p = 0.01$, where the median is 4, and individual tasks vary substantially. These finite-grid, post hoc selections therefore illustrate a qualitative tendency rather than a monotone law for each task or a direct estimate of the bound-minimizing interval.

D Additional Factor-Alignment Diagnostics

Figure 10 shows that denser communication generally produces faster and stronger factor alignment. Figure 11 shows an association between early alignment and final accuracy. Because communication probability and switching interval both vary across these configurations, this association should be interpreted as a diagnostic rather than a causal effect of similarity on accuracy.

E Additional Experimental Details

E.1 Compute Resources

All experiments were conducted on an internal GPU cluster using NVIDIA L40S GPUs with 48GB memory. A typical run required approximately 2–4 GPU-hours, depending on the dataset, communication probability, and number of clients. The full experimental campaign, including six communication probabilities, multiple choices of the switching interval T, three random seeds, four

Table 6: MNLI accuracy when scaling from 10 to 30 clients under the Severe Heterogeneous setting. We report two 30-client settings: using the same edge probability p as the 10-client setting, and reducing p to maintain a similar expected communication degree. The best result in each block is highlighted in bold.

10-Client p	**Method**	**10 Clients**	**30 Clients**	**30-Client p**	**30 Clients, Same Degree**
$p = 0.5$	LoRA	0.8132	0.8344	$p = 0.155$	0.8270
	FFA-LoRA	0.7313	0.7749		0.7814
	RoLoRA	0.8115	0.8393		0.8256
	TAD-LoRA	**0.8145**	**0.8431**		**0.8303**
$p = 0.1$	LoRA	0.7253	0.8021	$p = 0.031$	0.7556
	FFA-LoRA	0.7086	0.7469		0.7189
	RoLoRA	0.7113	0.8079		0.7157
	TAD-LoRA	**0.7405**	**0.8220**		**0.7650**
$p = 0.02$	LoRA	0.6407	0.7301	$p = 0.007$	0.5334
	FFA-LoRA	0.6191	0.6724		0.4776
	RoLoRA	0.5980	0.6669		0.4843
	TAD-LoRA	**0.6604**	**0.7401**		**0.6019**

Table 7: Results under a ring communication topology. Test accuracy is reported as mean $\pm$ standard deviation over 3 random seeds. The ring topology induces slow information mixing and serves as a structured stress test for decentralized LoRA fine-tuning.

Method	SST-2	QQP	QNLI	MNLI	Avg.
LoRA	0.9464 ± 0.0016	0.8308 ± 0.0057	0.9015 ± 0.0048	0.8185 ± 0.0018	0.8743
FFA-LoRA	0.9392 ± 0.0026	0.8044 ± 0.0096	0.8977 ± 0.0074	0.7728 ± 0.0150	0.8535
RoLoRA	0.9419 ± 0.0002	0.8246 ± 0.0024	0.8957 ± 0.0064	0.7997 ± 0.0112	0.8655
TAD-LoRA	0.9438 ± 0.0012	0.8290 ± 0.0046	0.9010 ± 0.0020	0.8208 ± 0.0046	0.8736

learning-rate candidates, baselines, preliminary exploration, and failed runs, required approximately 1200 GPU-hours. Once hyperparameters are selected, reproducing the final reported configurations requires substantially less compute.

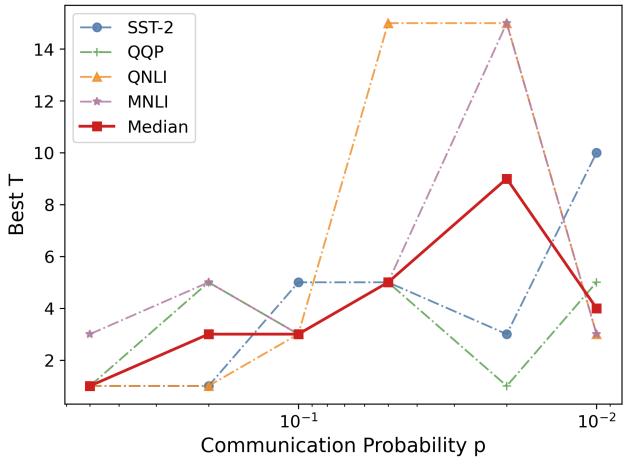

Figure 9: Best-observed switching interval for each task and communication probability p (thin lines), and the median across the four tasks (thick red line). Each interval is selected retrospectively from the evaluated candidate set using mean test accuracy over three seeds; it is not an online interval-selection rule.

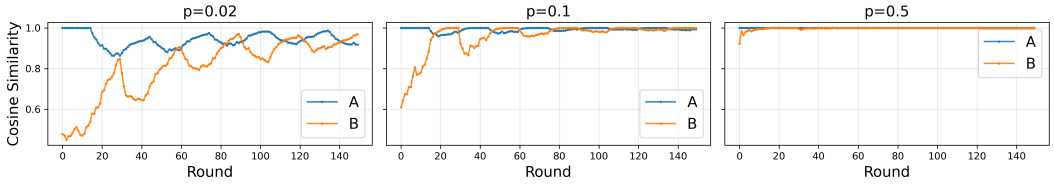

Figure 10: Inter-client cosine similarity of LoRA factors A and B over training rounds under different communication probabilities, using a fixed switching interval $T = 15$.

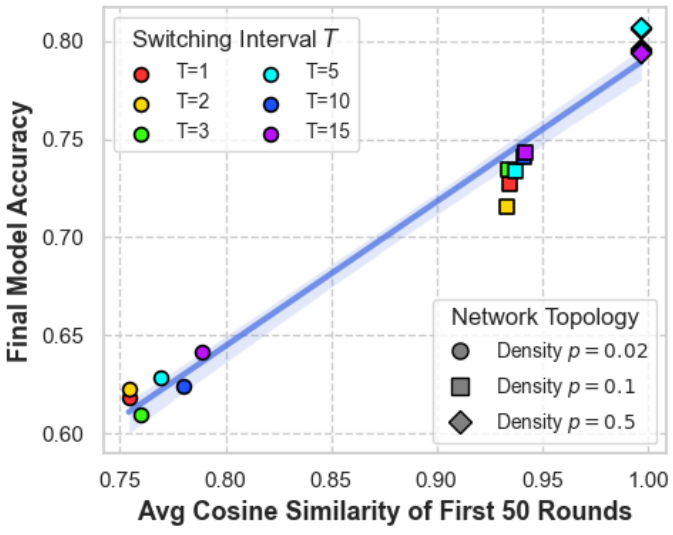

Figure 11: Association between mean inter-client factor similarity during the first 50 rounds and final model accuracy across the evaluated configurations.