# FedLSA: Cosine-Triggered Reparameterization Augmented by Output Boundary Calibration for Private Federated Low-Rank Adaptation

Zhencheng Fan[1], Tingyi Chen[1], Da Huang[1], Qihang Zhang[2], and Songzhu Mei[1(✉)]

[1] College of Computer Science and Technology, National University of Defense Technology, Changsha, China
[2] National Key Laboratory on Test and Evaluation for Electromagnetic Space Security, Luoyang, China
sz.mei@nudt.edu.cn

**Abstract.** Low-Rank Adaptation (LoRA) has become the standard paradigm for Parameter-Efficient Fine-Tuning (PEFT) within federated learning (FL) under privacy preservation. However, while reparameterization strategies in recent orthogonalization baselines like FedSVD enhance matrix expressiveness, they often overlook the actual dynamic requirements for subspace updates during training. To balance computational efficiency and model expressiveness under differential privacy (DP) constraints, we propose FedLSA, a Federated Low-rank Subspace Adaptive update framework. Specifically, FedLSA utilizes the cosine similarity of updates to the low rank coefficient matrix $B$ as an economical proxy for subspace drift, which minimizes overhead on the server through dynamic Singular Value Decomposition (SVD) allocation. To complement this mechanism triggered by events, we establish an Output Boundary Calibration (OBC) within the logits space to ensure the optimization trajectory remains robust even when matrix basis updates are suspended. Experimental results on five representative Natural Language Understanding (NLU) benchmarks demonstrate that FedLSA consistently outperforms all evaluated baselines in terms of accuracy. Notably, compared to the strongest state-of-the-art (SOTA) baseline, FedLSA achieves a better Pareto trade-off.

**Keywords:** Federated Low-Rank Adaptation, Cosine Triggered Reparameterization, Differential Privacy.

## 1 Introduction

Language models have demonstrated remarkable performance across various tasks [1, 2, 3]. While these models provide strong general capabilities, adapting them to specific domains or tasks typically require fine-tuning with domain-specific datasets [4]. In real-world deployments, however, training data is frequently fragmented across various organizations or user devices, and strict privacy regulations often prohibit direct data sharing [5]. Federated Learning (FL) [6] provides a viable solution by allowing

clients to fine-tune models locally on their private data, while a central server aggregates model updates without accessing raw training data, enabling privacy-preserving collaborative training.

However, traditional federated learning algorithms face severe computational bottlenecks when fine-tuning ultra large scale base models such as BERT [7] or GPT [8]. Therefore, the Parameter-Efficient Fine-Tuning (PEFT) [9] technique has emerged, in which Low-Rank Adaption (LoRA) [10] introduces a pair of low rank decomposition matrices $A$ and $B$ to reparameterize weight updates, significantly reducing the size of parameters that need to be trained and transmitted.

Against this backdrop, Federated Low-Rank Adaptation (FedLoRA) [11]) has emerged as a prominent research focus, where the massive model backbone remains completely frozen while requiring clients to train and upload only low rank incremental matrices. By significantly alleviating communication pressure while effectively preserving the expressive power of foundation models, this approach establishes a new trajectory for federated fine tuning under privacy preservation.

Although FL improves privacy by exchanging model updates instead of raw data, it does not provide formal guarantees against information leakage. Sophisticated attacks such as membership inference [12] or model inversion [13], can reconstruct sensitive information from shared updates, particularly given the capacity of language models to memorize training data [14, 15]. Therefore, integrating differential privacy (DP) [16] is essential to provide formal privacy guarantees and enhance the trustworthiness of collaborative model training. A common approach to enforcing DP in deep neural networks is DP-SGD [17, 18, 19], which clips the norm of each per-sample gradient to a predefined threshold and adds Gaussian noise to the average of the clipped gradients.

Recent work indicates that under the perturbation of DP-SGD, the dual matrix multiplication structure $W = BA$ inherent in LoRA triggers severe noise amplification effects. To mitigate this issue, prior research such as FFA-LoRA [20] chose to freeze the randomly initialized basis matrix $A$, which essentially trades model expressiveness for optimization stability under privacy preservation. To break the dilemma between efficiency and expressiveness, FedSVD [21] proposed a global reparameterization mechanism. This approach utilizes Singular Value Decomposition (SVD) on the server to periodically project aggregated weights back into the orthogonal basis space, thereby dynamically updating matrix $A$ without compromising DP guarantees. While FedSVD enhances matrix expressiveness while avoiding noise amplification under differential privacy, its strategy of full frequency SVD updates imposes substantial computational overhead on the server.

To address the problems discussed above, we propose FedLSA, a framework for adaptive updates of low rank subspaces triggered by cosine similarity. Specifically, this framework dynamically schedules SVD computations on the server by tracking the cosine similarity of the global coefficient matrix B across adjacent communication rounds. Furthermore, to calibrate effectively compensates for matrix basis obsolescence during periods of suspended updates，we propose an Output Boundary Calibration (OBC) mechanism to establish a structural safety boundary by imposing a norm constraint on the objective loss within the logits space. Specifically, the main contributions of our work are summarized as follows:

1. We propose an event triggered mechanism for adaptive subspace updates, utilizing the cosine similarity of updates to the coefficient matrix $B$ as an economical proxy to monitor low rank subspace drift and dynamically allocate SVD computations to minimize overhead on the server.

2. We propose the Output Boundary Calibration (OBC) mechanism to establish a structural safety boundary by imposing a norm constraint within the logits space, suppressing gradient inflation under differential privacy constraints.

3. We demonstrate across multiple NLU benchmarks that FedLSA consistently outperforms existing baselines in accuracy under both non-DP and DP constraints. Notably, FedLSA achieves Pareto improvements over SOTA by improving performance while reducing server-side SVD frequency.

## 2 Methodology

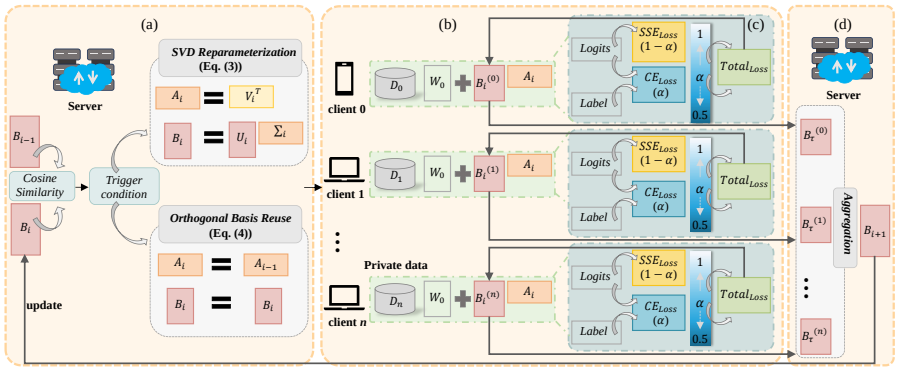


**Fig. 1.** The overview of our proposed FedLSA. (a) event triggered mechanism, (b) local update, (c) OBC, (d) global aggregation.

The overview of our proposed FedLSA framework is illustrated in Fig.1. The architecture initiates with an event triggered mechanism driven by cosine similarity as shown in Fig. 1(a) to dynamically schedule subspace updates. This is followed by local training across clients as shown in Fig. 1(b), during which the logits norm is constrained via the OBC module as shown in Fig. 1(c) to ensure optimization stability. Finally, the locally updated coefficient matrix $B$ from each participating client is uploaded to the server for global aggregation as shown in Fig. 1(d), thereby completing the federated training round.

### 2.1 Subspace Evolution Analysis

To reveal the computational redundancy within the FedSVD method, the evolution trajectory of model weights is quantitatively analyzed. On the one hand, during local training on private data within the FedSVD framework, the basis matrix $A$ remains frozen, ensuring that all incremental information is exclusively carried by the coefficient matrix $B$. On the other hand, inspired by dynamics research in large scale distributed

optimization, the directional consistency of update vectors has been proven as a critical metric for evaluating model convergence states and the stability of optimization trajectories [22, 23]. Motivated by these theoretical foundations, we characterize the degree of subspace drift by tracking the cosine similarity of update vectors for the global matrix $B$ between adjacent communication rounds.

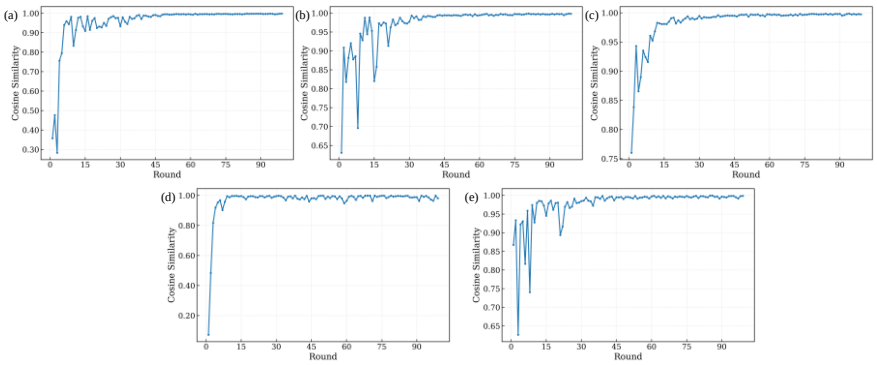


**Fig. 2.** Evolution trajectory of cosine similarity for updates to the coefficient matrix B across five NLU datasets. (a) SNLI [24], (b)MNLI [25], (c)SST-2 [26], (d)QQP [27], (e)QNLI [28].

As illustrated in Fig.2, the directional evolution of matrix $B$ within the FedSVD method is tracked across multiple NLU benchmark datasets. A highly consistent pattern emerges. In the early stages of fine tuning, the update direction of matrix $B$ fluctuates drastically, indicating that the subspace is undergoing rapid exploration and reconstruction. As training progresses, however, the cosine similarity of $B$ rises sharply and quickly stabilizes at an extremely high level. This dynamic observation regarding matrix $B$ reveals a substantial source of computational redundancy in performing full frequency SVD at every step. Once the update direction encoded by $B$ has effectively converged, repeatedly enforcing costly SVD on the weight matrix contributes little to additional model expressiveness while incurring significant computational overhead.

## 2.2 Cosine Triggered Low Rank Subspace Update

Building upon Section 2.1, we monitor the low rank subspace evolution via the cosine similarity between global coefficient matrices $B$ across adjacent rounds. For aggregated matrices $B_i$ and $B_{i-1}$ at rounds $i$ and $i-1$ , their similarity is defined as:

$$Cossim_i = \frac{\langle vec(B_i), vec(B_{i-1}) \rangle}{\|vec(B_i)\|_2 \|vec(B_{i-1})\|_2} \tag{1}$$

where $vec(\cdot)$ denotes the vectorization operator that flattens a matrix into a dimensional column vector, and $\|\cdot\|_2$ represents the standard vector $\ell_2$ norm. The computational complexity of this proxy metric is merely $O(N)$ (where $N$ is the total parameter count of the coefficient matrix). This mechanism allows the server to precisely quantify the directional consistency of the optimization trajectory of the coefficient matrix $B$ without introducing almost any additional computational burden. It provides a reliable

quantitative decision basis for triggering the adaptive reparameterization of the low rank subspace.

The server determines the update path of the low rank subspace as illustrated in Fig. 1(a) based on the relative relationship between the proxy metric $Cossim_i$ and the predefined threshold $\gamma$. This decision logic can be formally expressed as:

$$(A_i, B_i) = \begin{cases} SVD(B_i A_{i-1}) & Cossim_i \le \gamma \\ \{A_{i-1}, B_i\} & otherwise \end{cases} \tag{2}$$

where $\gamma$ serves as the triggering threshold, the specific value of which is empirically calibrated according to the optimization dynamic characteristics of diverse downstream tasks. The aforementioned adaptive decision logic balances routine pure coefficient matrix aggregation with computationally prohibitive basis matrix SVD reparameterization. When the system detects that the low rank subspace is undergoing severe directional drift, it triggers the SVD operation to reshape an optimization landscape. Conversely, once the subspace evolution enters a smooth steady convergence state, the system reuses the existing orthogonal basis to compress the computational overhead.

**SVD Reparameterization.** When $Cossim_i \le \gamma$ , the low rank subspace is in a rapid exploration phase. To reconstruct the basis, the server executes SVD on the aggregated global incremental weights. Specifically, the server computes the product of the current aggregated coefficient matrix $B_i$ and the previous basis matrix $A_{i-1}$, and factorizes it via SVD:

$$B_i := U_i[:, :r] \sum_i [:r, :r] \quad A_i := V^T[:r, :] \quad U_i \sum_i V_i^T = B_i A_{i-1} \tag{3}$$

where $r$ denotes the predefined rank size. The new basis matrix $A_i$ is a row orthogonal matrix that captures the most dominant data distribution directions within the current global update, while the magnitude and energy along these orthogonal directions are carried by the new coefficient matrix $B_i$. Benefiting from the low rank structural constraint, this truncated SVD achieves the equivalent mathematical exchange of $B_i A_i = B_i A_{i-1}$. It reconstructs an orthogonal well conditioned space to optimize the training trajectory while strictly ensuring the lossless transmission of global incremental information.

**Orthogonal Basis Reuse.** If the trigger condition is not satisfied, the subspace is deemed to have trended towards stability. During this steady phase, the basis matrix $A$ directly inherits the basis space coordinate system from the previous round, while the coefficient matrix $B$ is directly updated with the weighted aggregated value of the current round:

$$A_i = A_{i-1} \quad B_i = B_i \tag{4}$$

Implementing this orthogonal basis reuse strategy substantially duces SVD computational overhead while strictly preserving the orthogonal properties of the low rank subspace.

### 2.3 Output Boundary Calibration

Under noise perturbations, cross-entropy loss exhibits an implicit propensity to drive an unbounded growth of parameter norms [29]. This tendency is further exacerbated within the restricted subspaces where the basis matrix remains frozen for extended periods [30]. To counteract the optimization instability triggered by such norm inflation during the basis reuse phase, we establish an output boundary calibration mechanism during local training, as illustrated in Fig.1(b).

It imposes a Hinge style norm constraint on the model prediction output $z$, thereby mitigating gradient inflation:

$$L_{SSE} = \max(0, \|z\|_2 - \tau)^2 \tag{5}$$

where $\tau$ is a predefined threshold. Equation (5) applies a quadratic hinge penalty to the $\ell_2$-norm of logits $z$, which is activated only when $\| z \|_2 > \tau$. This regularization softly constrains excessive logit magnitudes and improves training stability during prolonged basis-freezing rounds.

The client optimizes the final local objective of the model by integrating task accuracy and structural stability:

$$L_{total} = \alpha L_{CE} + (1 - \alpha) \cdot L_{SSE} \tag{6}$$

where $\alpha \in [0.5, 1]$ controls the trade-off between task optimization and output boundary regularization. We use a larger α in early training to prioritize task learning and gradually decrease it in later rounds to enhance stability.

## 3 Experiment

### 3.1 Experimental Setups

**Datasets.** Following FedSVD, we use five datasets, including four from the GLUE benchmark [31]:
**SNLI** [24]: A sentence-pair classification task for textual entailment, categorized into three labels (entailment, neutral, contradiction).
**MNLI** [25]**:** A multi-genre natural language inference task. We evaluate it on both matched (in-domain) and mismatched (cross-domain) splits.
**SST-2** [26]**:** A single-sentence sentiment classification task predicting positive or negative sentiment.
**QQP** [27]**:** A paraphrase detection task identifying whether two questions are semantic duplicates.
**QNLI** [28]**:** A binary classification task predicting whether a context sentence contains the answer to a given question.

**Baselines.** We compare our method, FedLSA, against the following baselines:

**FedAvg** [6, 32]: The most fundamental federated learning algorithm. In this baseline, clients locally train both the $A$ and $B$ matrices of LoRA, while the server performs independent average aggregation on each matrix.
**FFA-LoRA** [20]: To mitigate the noise amplification effect within differential privacy environments, this method randomly initializes the $A$ matrix and keeps it strictly frozen, exclusively training and aggregating the $B$ matrix on the clients.
**FedSVD** [21]: The current SOTA. It utilizes SVD at the server side to periodically project the aggregated weights back into the orthogonal basis space, thereby dynamically updating the basis matrix to compensate for the inadequate expressive capacity of the FFA-LoRA method.

**Implementation details.** We partition non-I.I.D. data across 6 clients via a Dirichlet distribution (concentration parameter 0.5), randomly sampling 3 clients per round. Using RoBERTa-large [33], we apply LoRA (rank $r = 8$, scaling factor $\alpha = 8$, dropout 0.05) exclusively to the query and value projections. Federated training spans 100 rounds, with clients performing 10 local steps via vanilla SGD. For our FedLSA framework, the triggering threshold $\gamma$ is set to 0.975. For OBC, the boundary threshold $\tau$ is set to 10. The adjustment factor $\alpha$ is initialized at 1.0 for the first 15 rounds and linearly decayed to 0.5 over the remaining 85 rounds. For differential privacy, we use the Opacus library [34] to enforce different privacy budgets $\epsilon$ across experiments. All experiments are conducted on three NVIDIA RTX 4090 GPUs.

### 3.2 Main Results

**Effectiveness of FedLSA without privacy constraints.** We first assess the proposed FedLSA on five datasets, where MNLI is evaluated on matched and mismatched splits, in a non-private setting. As shown in Table 1, FFA-LoRA substantially outperforms standard FedAvg, which we attribute to the reduced aggregation error brought by its single-matrix aggregation strategy. FedSVD further improves performance by periodically updating the basis matrix through SVD reparameterization. Our proposed FedLSA achieves the best overall performance, obtaining the highest average accuracy of 87.63%, outperforming FedSVD by 0.70 percentage points and FFA-LoRA by 1.95 points. In terms of per-task results, FedLSA achieves the best performance on SNLI, both MNLI splits, SST-2, and QQP, while FedSVD performs slightly better on QNLI. These results suggest that combining adaptive subspace updates with OBC regularization provides a more effective optimization strategy than fixed update schedules in non-DP federated training.

**Effectiveness of FedLSA with DP-SGD.** Table 2 summarizes the performance under DP constraints ($\epsilon = 6$). While DP noise inevitably degrades accuracy across all methods, FedLSA maintains the leading average accuracy of 76.95%. Notably, the performance margin between FedLSA and FFA-LoRA expands significantly from 1.95% in non-DP settings to 8.07% under DP, while consistently surpassing the strongest baseline, FedSVD, by 0.51%. This superior robustness stems from the synergy between the

mathematical optimization of orthogonal subspaces and the norm-bounding benefits of OBC, which collectively ensure a well-conditioned geometric space. Consequently, FedLSA secures top performance across five datasets. Although FFA-LoRA exhibits an outlier high on SST-2 (90.94%), its drastic decline on other tasks (e.g., 59.12% on MNLI-m) highlights FedLSA's superior stability and practical viability for privacy-preserving deployments. Furthermore, Table 3 reports additional results on MNLI under different privacy budgets. FedLSA consistently surpasses FedSVD at both $\epsilon = 2$ and $\epsilon = 4$, confirming that the proposed framework remains effective even under stronger privacy noise.

**Table 1.** Test accuracy (%) in non-DP settings. MNLI-m and MNLI-mm denote the matched and mismatched subsets of MNLI, respectively.

| Method | SNLI | MNLI-m | MNLI-mm | SST-2 | QQP | QNLI | Avg |
|---|---|---|---|---|---|---|---|
| FedAvg | 85.09 | 75.99 | 76.75 | 85.64 | 63.18 | 75.08 | 76.96 |
| FFA-LoRA | 84.42 | 83.33 | 83.91 | 94.72 | 81.56 | 86.11 | 85.68 |
| FedSVD | 86.07 | 84.82 | 85.57 | 94.61 | 83.45 | **87.04** | 86.93 |
| **FedLSA** | **86.55** | **85.73** | **87.24** | **95.72** | **84.01** | 86.52 | **87.63** |

**Table 2.** Test accuracy (%) under differential privacy ($\epsilon = 6$).

| Method | SNLI | MNLI-m | MNLI-mm | SST-2 | QQP | QNLI | Avg |
|---|---|---|---|---|---|---|---|
| FedAvg | 54.28 | 65.00 | 66.22 | 80.92 | 63.18 | 59.46 | 64.84 |
| FFA-LoRA | 63.14 | 59.12 | 60.73 | **90.94** | 67.31 | 72.03 | 68.88 |
| FedSVD | 73.17 | 70.76 | 71.61 | 89.68 | 76.89 | 76.51 | 76.44 |
| **FedLSA** | **73.75** | **71.93** | **72.39** | 89.11 | **77.35** | **77.18** | **76.95** |

**Table 3.** Accuracy (%) on MNLI under different privacy budgets.

| DP Budget | Method | MNLI-m | MNLI-mm | Avg |
|---|---|---|---|---|
| $\epsilon = 2$ | FedSVD | 69.32 | 70.17 | 69.75 |
| | **FedLSA** | **69.56** | **70.97** | **70.27** |
| $\epsilon = 4$ | FedSVD | 70.32 | 71.39 | 70.86 |
| | **FedLSA** | **71.13** | **72.14** | **71.64** |

**Performance-Efficiency Trade-off.** Methods such as FedAvg and FFA-LoRA avoid SVD computation entirely, but their substantially lower accuracy in Tables 1 and 2 indicates a clear utility gap. We therefore focus on SVD-based methods and evaluate the trade-off between model performance and server-side reparameterization cost. Specifically, we measure utility by test accuracy and efficiency by the total number of SVD operations (#SVD), which serves as a practical proxy for server-side computational overhead. Tables 4 and 5 summarize the results across five datasets. Compared with

FedSVD, FedLSA dramatically reduces the number of SVD operations from 100 to only 10–25 over 100 training rounds, corresponding to a reduction of 75%–90% in decomposition frequency. Under our implementation, each server-side SVD operation takes approximately 5–6 seconds, indicating substantial wall-clock savings. Despite this large reduction in cost, FedLSA maintains comparable or higher accuracy on all datasets, improving performance on SNLI, both MNLI splits, SST-2, and QQP, while incurring only a minor decrease on QNLI. Overall, these results demonstrate that FedLSA achieves a substantially more favorable performance-efficiency trade-off than FedSVD.

**Table 4.** Trade-off between accuracy (%) and #SVD on SNLI and MNLI.

| Method | SNLI | | MNLI-m | | MNLI-mm | |
|---|---|---|---|---|---|---|
| | Acc | #SVD | Acc | #SVD | Acc | #SVD |
| FedSVD | 86.07 | 100 | 84.82 | 100 | 85.57 | 100 |
| **FedLSA** | 86.55 | 21 | 85.73 | 25 | 87.24 | 25 |

**Table 5.** Trade-off between accuracy (%) and #SVD on SST-2, QQP, and QNLI.

| Method | SST-2 | | QQP | | QNLI | |
|---|---|---|---|---|---|---|
| | Acc | #SVD | Acc | #SVD | Acc | #SVD |
| FedSVD | 94.61 | 100 | 83.45 | 100 | 87.04 | 100 |
| **FedLSA** | 95.72 | 10 | 84.01 | 14 | 86.52 | 20 |

### 3.3 Ablation Study

**Component Ablation.** To evaluate the individual contribution of each module, we conduct an ablation study on the GLUE benchmarks, as shown in Table 6. The baseline variant without CTM or OBC achieves an average accuracy of 86.93%. Using CTM only reduces the average score to 85.60%, indicating that adaptive skipping of SVD updates alone may hurt optimization quality when no stabilization mechanism is applied. Delayed basis refresh can cause training to proceed in an outdated low-rank subspace, reducing adaptability. Using OBC only improves the average accuracy to 87.54%, showing that output boundary calibration provides effective regularization and enhances training stability. When CTM and OBC are combined, FedLSA achieves the best overall result of 87.63%. This suggests that the two modules are complementary: CTM improves efficiency through selective triggering, while OBC stabilizes training when updates are less frequent. Overall, the results show that adaptive triggering alone is insufficient, while its combination with OBC is essential to achieving the full performance of FedLSA.

**Table 6.** Ablation study on the contributions of the Cosine Trigger Mechanism (CTM) and OBC. Results are reported in accuracy (%).

| CTM | OBC | SNLI | MNLI-m | MNLI-mm | SST-2 | QQP | QNLI | Avg |
|---|---|---|---|---|---|---|---|---|
| ✗ | ✗ | 86.07 | 84.82 | 85.57 | 94.61 | 83.45 | 87.04 | 86.93 |
| ✓ | ✗ | 85.42 | 83.11 | 84.26 | 92.67 | 81.98 | 86.13 | 85.60 |
| ✗ | ✓ | 85.60 | 85.13 | 86.64 | **96.53** | 83.85 | **87.50** | 87.54 |
| ✓ | ✓ | **86.55** | **85.73** | **87.24** | 95.72 | **84.01** | 86.52 | **87.63** |

**Fixed-frequency vs Adaptive Frequency.** Table 7 presents the performance of different SVD update strategies on five GLUE benchmark tasks. We compare the update rule adopted by the current state-of-the-art baseline FedSVD, which performs decomposition at every communication round, with less frequent fixed schedules that update every 5 rounds or every 10 rounds, as well as the proposed FedLSA framework. The results show that FedLSA achieves the best overall performance with an average score of 87.63, surpassing SVD-1 by 0.70 points. Comparing the fixed schedules further shows that more frequent decomposition is generally beneficial, as updating every round consistently outperforms updating every 5 or 10 rounds. This indicates that timely low-rank reparameterization is important for maintaining model adaptability during federated optimization. Although the sparse schedules SVD-5 and SVD-10 reduce the number of SVD operations, their average scores drop to 85.09 and 84.73, respectively, suggesting that naively delayed decomposition weakens adaptation quality. By contrast, FedLSA combines adaptive decomposition scheduling with the proposed OBC mechanism, which allows the model to preserve efficient updates while maintaining stronger optimization dynamics. These results demonstrate that the coordinated design of adaptive scheduling and OBC yields the best overall trade-off between efficiency and performance.

**Table 7.** Effect of Fixed vs. Adaptive SVD Update Strategies on GLUE Tasks (SVD-k denotes decomposition every k rounds). Results are reported in accuracy (%).

| Strategy | SNLI | MNLI-m | MNLI-mm | SST-2 | QQP | QNLI | Avg |
|---|---|---|---|---|---|---|---|
| SVD-1 | 86.07 | 84.82 | 85.57 | 94.61 | 83.45 | **87.04** | 86.93 |
| SVD-5 | 85.22 | 83.53 | 84.17 | 93.50 | 81.14 | 84.99 | 85.09 |
| SVD-10 | 85.19 | 82.89 | 83.59 | 93.07 | 79.22 | 84.41 | 84.73 |
| **FedLSA** | **86.55** | **85.73** | **87.24** | **95.72** | **84.01** | 86.52 | **87.63** |

### 3.4 Hyperparameter Sensitivity Analysis

We further conduct a hyperparameter sensitivity analysis on the trigger threshold $\gamma$, which controls when adaptive SVD decomposition is activated in FedLSA. Fig.3 and Fig.4 report the task performance and the corresponding number of SVD operations under different values of $\gamma$, ranging from 0.90 to 0.99. Overall, FedLSA shows stable

performance across a relatively wide range of threshold values, indicating that the proposed method is not overly sensitive to this hyperparameter. For most tasks, performance variations remain small, and FedLSA consistently outperforms or remains competitive with the FedSVD baseline. From the efficiency perspective, increasing $\gamma$ leads to a larger number of SVD operations, since decomposition is triggered more frequently under stricter thresholds. When $\gamma = 0.90$, only a small number of decompositions are required across tasks, whereas the number increases substantially when $\gamma = 0.99$. This demonstrates that $\gamma$ provides a simple and effective mechanism for controlling server-side computational overhead. Notably, strong performance is maintained across a broad interval, particularly between $\gamma = 0.925$ and $\gamma = 0.975$, where the model achieves competitive accuracy with moderate decomposition cost. Based on this robust efficiency trade-off, we adopt $\gamma = 0.975$ as the default setting in experiments.

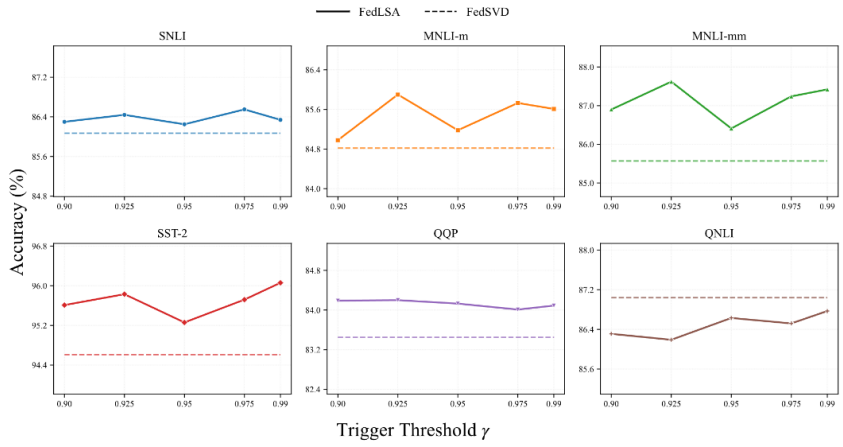


**Fig. 3.** Accuracy under Different Trigger Thresholds $\gamma$.

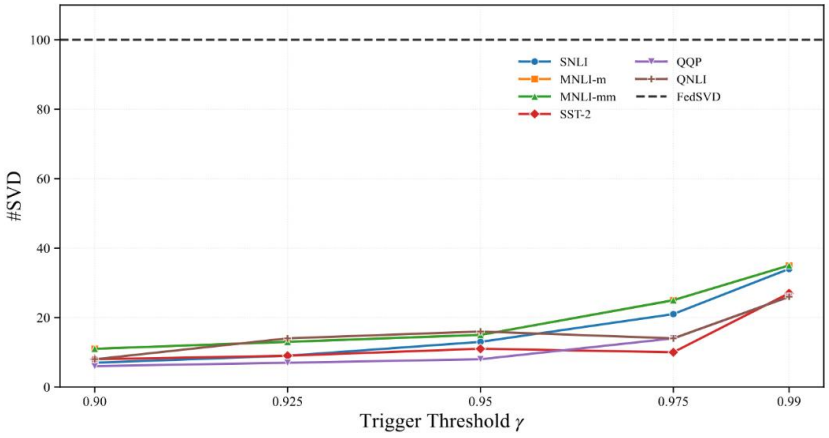


**Fig. 4.** SVD Frequency under Different Trigger Thresholds $\gamma$.

## 4 Conclusion

We propose FedLSA, an efficient and robust FedLoRA framework designed to alleviate the server-side computational overhead associated with frequent SVD reparameterization. FedLSA combines a dynamic cosine triggering mechanism with an Output Boundary Calibration (OBC) module in a complementary manner. Specifically, the adaptive trigger reduces the number of server-side SVD operations by approximately 80%, while OBC stabilizes local optimization under sparse update schedules through logits norm regularization. Extensive experiments demonstrate that FedLSA consistently outperforms strong baselines such as FedSVD and FFA-LoRA. Overall, FedLSA provides an effective solution for improving both model utility and computational efficiency in practical federated deployments.

## References

1. Touvron, H., Martin, L., Stone, K., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
2. Team, G., Mesnard, T., Hardin, C., et al.: Gemma: Open models based on gemini research and technology. arXiv preprint arXiv:2403.08295 (2024)
3. Devlin, J., Chang, M.W., Lee, K., et al.: Bert: Pre-training of deep bidirectional transformers for language understanding. Proc. NAACL-HLT. 1, 4171–4186 (2019)
4. Bommasani, R., Hudson, D.A., Adeli, E., et al.: On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258 (2021)
5. Regulation, P.: Regulation (EU) 2016/679 of the European Parliament and of the Council. Regulation (EU). 679, 10–3 (2016)
6. McMahan, B., Moore, E., Ramage, D., et al.: Communication-efficient learning of deep networks from decentralized data. Proc. AISTATS. PMLR, 1273–1282 (2017)
7. Devlin, J., Chang, M.W., Lee, K., et al.: Bert: Pre-training of deep bidirectional transformers for language understanding. Proc. NAACL-HLT. 1, 4171–4186 (2019)
8. Radford, A., Narasimhan, K., Salimans, T., et al.: Improving language understanding by generative pre-training. OpenAI Tech. Rep. (2018)
9. Houlsby, N., Giurgiu, A., Jastrzebski, S., et al.: Parameter-efficient transfer learning for NLP. Proc. ICML. PMLR, 2790–2799 (2019)
10. Hu, E.J., Shen, Y., Wallis, P., et al.: Lora: Low-rank adaptation of large language models. Proc. ICLR. 1, 3 (2022)
11. Yang, Y., Long, G., Lu, Q., et al.: Federated low-rank adaptation for foundation models: A survey. arXiv preprint arXiv:2505.13502 (2025)
12. Shokri, R., Stronati, M., Song, C., et al.: Membership inference attacks against machine learning models. Proc. IEEE S&P. 3–18 (2017)
13. Fredrikson, M., Jha, S., Ristenpart, T.: Model inversion attacks that exploit confidence information and basic countermeasures. Proc. ACM CCS. 1322–1333 (2015)
14. Carlini, N., Tramer, F., Wallace, E., et al.: Extracting training data from large language models. Proc. USENIX Security. 2633–2650 (2021)
15. Carlini, N., Ippolito, D., Jagielski, M., et al.: Quantifying memorization across neural language models. Proc. ICLR. (2022)
16. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. 9, 211–487 (2014)

17. Song, S., Chaudhuri, K., Sarwate, A.D.: Stochastic gradient descent with differentially private updates. Proc. GlobalSIP. 245–248 (2013)
18. Bassily, R., Smith, A., Thakurta, A.: Private empirical risk minimization: Efficient algorithms and tight error bounds. Proc. FOCS. 464–473 (2014)
19. Abadi, M., Chu, A., Goodfellow, I., et al.: Deep learning with differential privacy. Proc. ACM CCS. 308–318 (2016)
20. Sun, Y., Li, Z., Li, Y., Ding, B.: Improving LoRA in privacy-preserving federated learning. In: The Twelfth International Conference on Learning Representations (ICLR). (2024)
21. Lee, S., Park, S., Lee, D.B., et al.: FedSVD: Adaptive Orthogonalization for Private Federated Learning with LoRA. Proc. NeurIPS. (2025)
22. Dean, J., Corrado, G., Monga, R., et al.: Large scale distributed deep networks. Proc. NeurIPS. 25 (2012)
23. Li, H., Xu, Z., Taylor, G., et al.: Visualizing the loss landscape of neural nets. Proc. NeurIPS. 31 (2018)
24. Bowman, S., Angeli, G., Potts, C., et al.: A large annotated corpus for learning natural language inference. Proc. EMNLP. 632–642 (2015)
25. Williams, A., Nangia, N., Bowman, S.: A broad-coverage challenge corpus for sentence understanding through inference. Proc. NAACL-HLT. 1, 1112–1122 (2018)
26. Socher, R., Perelygin, A., Wu, J., et al.: Recursive deep models for semantic compositionality over a sentiment treebank. Proc. EMNLP. 1631–1642 (2013)
27. Sharma, L., Graesser, L., Nangia, N., et al.: Natural language understanding with the quora question pairs dataset. arXiv preprint arXiv:1907.01041 (2019)
28. Wang, A., Singh, A., Michael, J., et al.: GLUE: A multi-task benchmark and analysis platform for natural language understanding. Proc. EMNLP. 353–355 (2018)
29. Soudry, D., Hoffer, E., Nacson, I.T., et al.: The implicit bias of gradient descent on separable data. Journal of Machine Learning Research. 19(70), 1--57 (2018)
30. Garg, A., Saratchandran, H., Garg, R., et al.: Stable Forgetting: Bounded Parameter-Efficient Unlearning in LLMs. arXiv preprint arXiv:2509.24166 (2025)
31. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: GLUE: A multi-task benchmark and analysis platform for natural language understanding. Proc. ICLR. (2019)
32. Zhang, J., Vahidian, S., Kuo, M., Li, C., Zhang, R., Yu, T., Wang, G., Chen, Y.: Towards building the federatedgpt: Federated instruction tuning. Proc. ICASSP. (2024)
33. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
34. Yousefpour, A., Shilov, I., Sablayrolles, A., Testuggine, D., Prasad, K., Malek, M., Nguyen, J., Ghosh, S., Bharadwaj, A., Zhao, J., Cormode, G., Mironov, I.: Opacus: User-friendly differential privacy library in PyTorch. arXiv preprint arXiv:2109.12298 (2021)