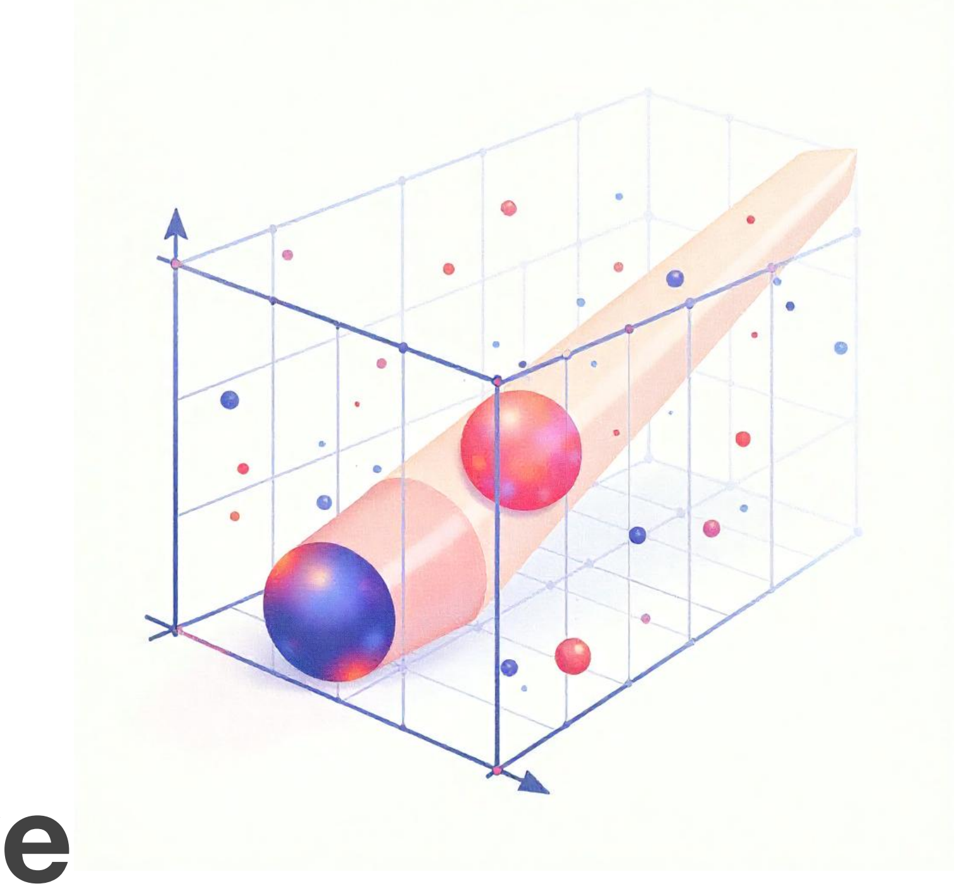
Federated Full-Parameter Tuning at Scale for LLMs

Yao Shu*[1], **Wenyang Hu***[2,3]
See-Kiong Ng[3], Bryan Kian Hsiang Low[3], Fei Yu[4]

HKUST (GZ)[1], SAP[2], NUS[3], Guangdong Lab of AI and Digital Economy (SZ)[4]

Federated full-parameter tuning a 7B LLM with just 6K communicated parameters!



Objective

Given **random bases** $\mathbf{V} = [\mathbf{v}_1 \mathbf{v}_2 \cdots \mathbf{v}_K] \in \mathbb{R}^{d \times K}$ generated by a seed s and a local update Δ, we want to project the update Δ into coordinates

$$\underset{\text{Coordinates}}{\boldsymbol{\gamma}} \triangleq \arg\min_{\mathbf{y}} \|\underset{\text{Random Bases}}{\mathbf{V}}\mathbf{y} - \Delta\|$$

$$\boldsymbol{\gamma} = \underbrace{(\mathbf{V}^\top \mathbf{V})^{-1}}_{\text{Calculating this inversion is very costly!}} \mathbf{V}^\top \Delta, \quad \widetilde{\Delta} = \mathbf{V}(\mathbf{V}^\top \mathbf{V})^{-1}\mathbf{V}^\top \Delta \,.$$

Calculating this inversion is very costly!

Theorem 1 (Unbiased Reconstruction). *Given the reconstruction in (7), we have*

$$\mathbb{E}\left[\widetilde{\Delta}\right] = \Delta \,.$$

Theorem 2 (Reconstruction Error). *Given the reconstruction in (7), we have*

$$\mathbb{E}\left[\left\|\widetilde{\Delta} - \Delta\right\|\right] \leq \max\left\{2\sqrt{\frac{2\ln(2d)}{\rho K}}, \frac{2\ln(2d)}{\rho K}\right\}\|\Delta\| \,.$$

Project a first-order gradient using random vectors and recover it using shared randomness

Our Solution Ferret

1. Reconstruction w/o Inversion

Approximate $\mathbf{V}^\top \mathbf{V}$ with $\mathbf{I}_K$

$$\boldsymbol{\gamma} \approx (\rho K)^{-1}\mathbf{V}^\top \Delta$$

$$\widetilde{\Delta} = (\rho K)^{-1}\mathbf{V}\mathbf{V}^\top \Delta$$

2. Blockwise Reconstruction

Divide the full dimension d into L blocks. Then for each block l, we have

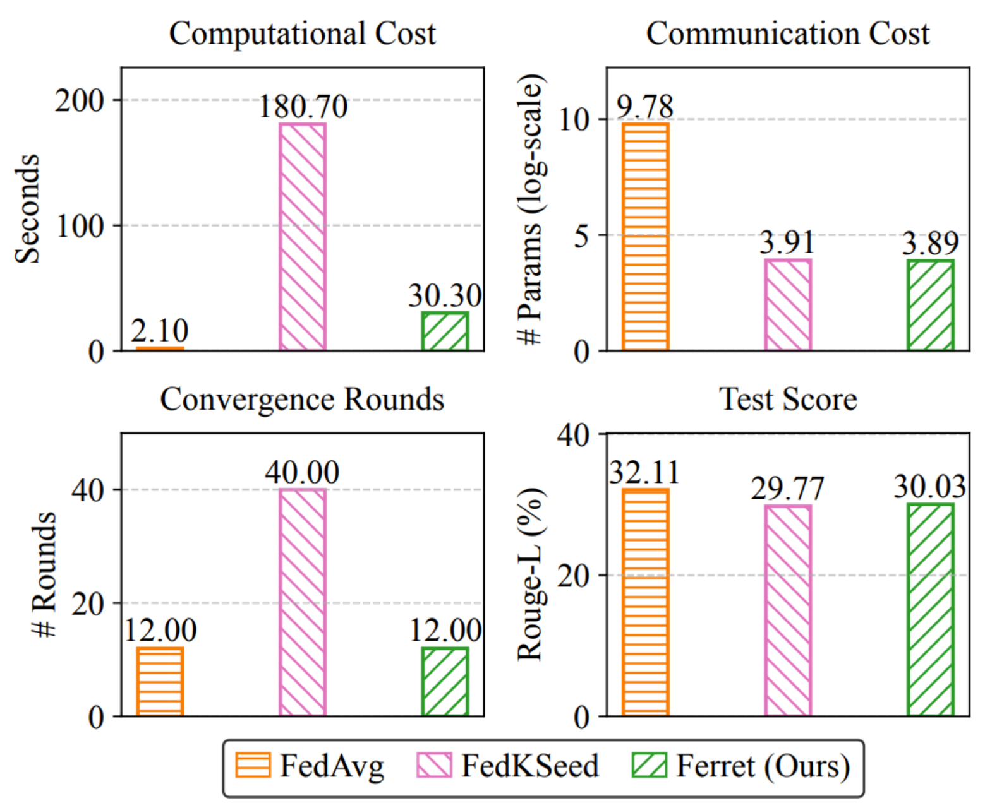
$$\gamma_l = (\rho_l K)^{-1}\mathbf{V}_l^\top \Delta_l$$

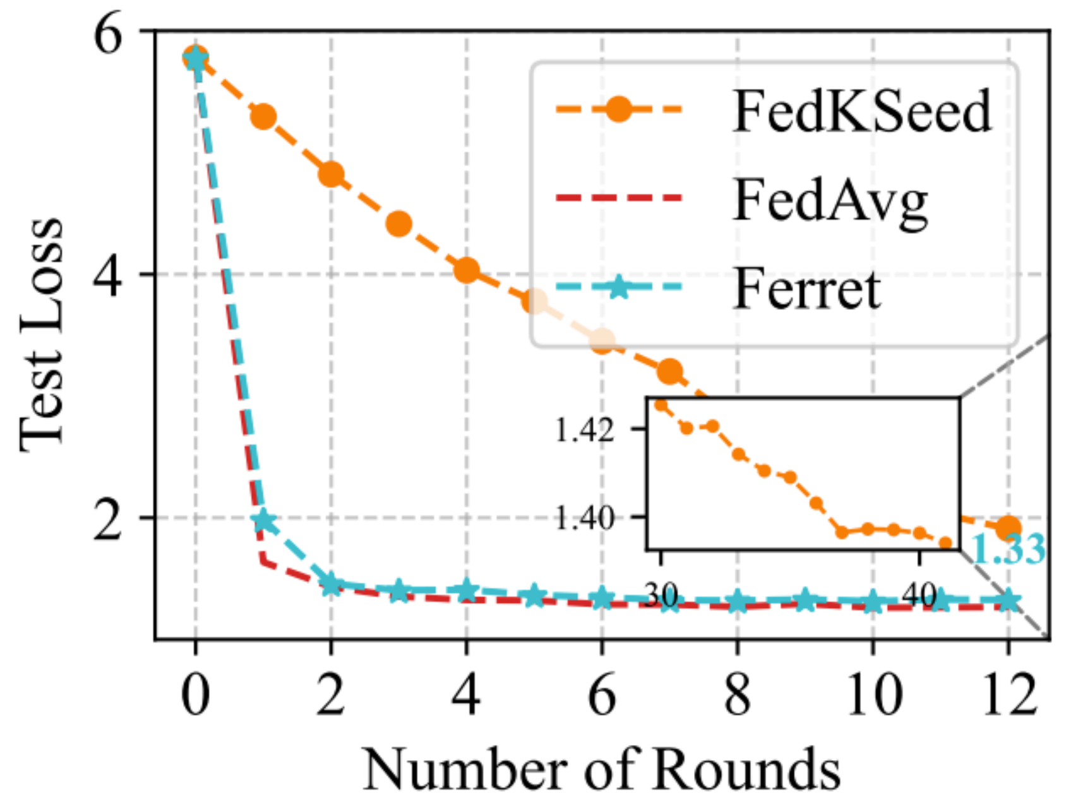
$$\widetilde{\Delta}_l = (\rho_l K_l)^{-1}\mathbf{V}_l\mathbf{V}_l^\top \Delta_l$$

Proposition 2 (Block-Wise Reconstruction Error). *For block-wise reconstruction (8) of size L, when $\sqrt{d_l} \geq K_l$ for any $l \in [L]$,*

$$\mathbb{E}\left[\left\|\widetilde{\Delta} - \Delta\right\|\right] < \widetilde{\mathcal{O}}\left(\sum_{l \in [L]} \frac{\|\Delta_l\|}{\rho_l K_l}\right) \,,$$

which is minimized by choosing $K_l \propto \sqrt{\|\Delta_l\| / \rho_l}$.





- *Reduced communication cost*
- *High computational efficiency*
- *Fast convergence*

Algorithm	**CodeAlpaca**		**GSM8K**	
	LLaMA2-7B	LLaMA2-13B	LLaMA2-7B	LLaMA2-13B
FedIT	4.66 ± 0.18	6.10 ± 0.18	30.31 ± 0.29	13.46 ± 0.34
FedZO	4.58 ± 0.26	6.19 ± 0.32	30.41 ± 0.31	13.63 ± 0.34
FedKSeed	8.33 ± 0.98	10.70 ± 0.47	28.26 ± 3.60	33.67 ± 1.15
FedAvg	**15.41** ± 0.43	**14.68** ± 0.26	**38.30** ± 0.40	**39.82** ± 0.17
Ferret (ours)	12.10 ± 0.47	11.84 ± 0.91	36.10 ± 1.18	34.50 ± 1.42

Algorithm 1 Ferret

```
Input: $\mathbf{w}_0, N, R, T, K, \eta$
for each round $r \in [R]$ do
    for each client $j \in [N]$ in parallel do
        if $r > 1$ then // Step ①: Global Aggregation
            Receive $\{s^{(i)}\}_{i=1}^N$ and $\{\gamma_k^{(i)}\}_{i=1,k=1}^{N,K}$
            Generate bases $\{\mathbf{v}_k^{(i)}\}_{i=1,k=1}^{N,K}$ using $\{s^{(i)}\}_{i=1}^N$
            $\mathbf{w}_{r-1} \leftarrow \mathbf{w}_{r-2} - \sum_{i \in [N]} \left(\sum_{k=1}^K \gamma_k^{(i)} \mathbf{v}_k^{(i)}\right) / N$
        $\mathbf{w}_{r,0} \leftarrow \mathbf{w}_r$
        for $t \in [T]$ do // Step ②: Local Updates
            $\mathbf{w}_{r,t}^{(j)} \leftarrow \mathbf{w}_{r,t-1}^{(j)} - \eta \nabla \ell(\mathbf{w}_{r,t-1}^{(j)}; \mathbf{x}_{r,t-1}^{(j)})$
        // Step ③: Projected Updates
        Randomly set $s^{(j)}$ and generate bases $\{\mathbf{v}_k^{(j)}\}_{k=1}^K$
        $\Delta_r^{(j)} \leftarrow \mathbf{w}_{r-1}^{(j)} - \mathbf{w}_r^{(j)}$, compute $\{\gamma_k^{(j)}\}_{k=1}^K$ with (6)
        Send $s^{(j)}$ and $\{\gamma_k^{(j)}\}_{k=1}^K$ to the central server
```