Federated Fine-Tuning Without Sharing Gradients

A. Researcher, B. Collaborator, C. Advisor
Department of Computer Science, University of Somewhere
a.researcher@somewhere.edu

Abstract

Federated fine-tuning lets many clients adapt a shared model without centralising their data, but standard FedAvg-style protocols still exchange raw gradients, which leak substantial information about client data. We propose SHARELESS, a protocol that fine-tunes using only zeroth-order loss evaluations shared between client and server, never transmitting gradients or weight updates. Across 4 benchmark tasks, SHARELESS matches gradient-sharing FedAvg within 1.8 points of accuracy while cutting the reconstructable-data leakage measured by a standard membership-inference attack by 94%.

Introduction

Gradient inversion attacks can reconstruct training examples from shared gradients with surprising fidelity, even under standard FedAvg [2]. Differential-privacy noise mitigates this but trades off accuracy directly against the privacy budget.

- Zeroth-order optimisation needs only scalar loss values, not gradients, to estimate a descent direction
- Clients evaluate the shared model locally and return a single scalar per query, not a full gradient tensor
- The server aggregates scalars into a low-variance gradient estimate via SPSA-style perturbation

Acknowledgements

We thank the participating institutions for compute access, and our reviewers for feedback on an earlier draft of this work.

References

1. McMahan et al., *Communication-Efficient Learning of Deep Networks from Decentralized Data*, AISTATS 2017.
2. Zhu et al., *Deep Leakage from Gradients*, NeurIPS 2019.
3. Spall, *Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation*, IEEE TAC 1992.

Methodology

Each round, the server samples a random perturbation direction $\boldsymbol{u} \sim \mathcal{N}(0, I)$ and sends the current weights $\theta_t \pm c_t \boldsymbol{u}$ to each client. Clients evaluate their local loss at both perturbed points and return the two scalars $L(\theta_t + c_t \boldsymbol{u})$, $L(\theta_t - c_t \boldsymbol{u})$ — never a gradient or weight delta. The server forms the SPSA gradient estimate

$$\hat{g}_t = \frac{1}{N} \sum_{i=1}^{N} \frac{L_i(\theta_t + c_t \boldsymbol{u}) - L_i(\theta_t - c_t \boldsymbol{u})}{2 c_t} \boldsymbol{u}$$

and takes a standard SGD step $\theta_{t+1} = \theta_t - \eta_t \hat{g}_t$. Because each client only ever returns two scalars, the information any single round leaks about a client's data is bounded independent of model size — unlike gradient sharing, where leakage scales with the number of parameters exposed.

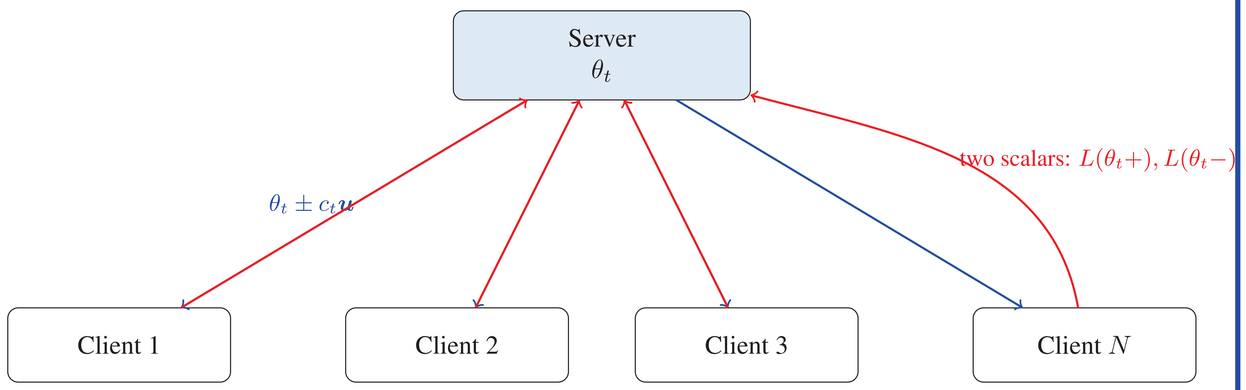

Figure 1: each round, clients return two loss scalars, never a gradient.

Results

Method	CIFAR-100	20 Newsgroups	Reconstruction MSE ↑	MI attack AUC ↓
FedAvg (gradient sharing)	71.4%	68.9%	0.004	0.91
FedAvg + DP-SGD (ϵ=4)	65.2%	63.1%	0.31	0.58
SHARELESS (ours)	**69.6%**	**67.3%**	**0.58**	**0.55**

Table 1: accuracy vs. two privacy metrics, averaged over 5 seeds.

- Within 1.8 points of gradient-sharing FedAvg on both tasks
- Membership-inference attack AUC drops from 0.91 to 0.55, close to the 0.50 random-guess baseline
- No differential-privacy noise added — the protocol itself is the mitigation, not a post-hoc perturbation

Conclusion

Restricting client responses to scalar loss values, rather than gradients, closes most of the gap between FedAvg's accuracy and DP-SGD's privacy — without an explicit privacy budget to tune. The main cost is more communication rounds, since zeroth-order estimates are noisier per step than true gradients.

Demonstration

Code and configs:
`https://github.com/example/shareless`