

Federated Learning Playground

Bryan Guanrong Shan¹, Alysa Ziyang Tan^{1,2,3}, Han Yu¹

¹Nanyang Technological University (NTU), Singapore

²Alibaba-NTU Singapore Joint Research Institute, NTU, Singapore

³Alibaba Group, Hangzhou, China

{bshan001,s190109,han.yu}@ntu.edu.sg

Abstract

We present *Federated Learning Playground*, an interactive browser-based platform inspired by and extends TensorFlow Playground that teaches core Federated Learning (FL) concepts. Users can experiment with heterogeneous client data distributions, model hyperparameters, and aggregation algorithms directly in the browser without coding or system setup, and observe their effects on client and global models through real-time visualizations, gaining intuition for challenges such as non-IID data, local overfitting, and scalability. The playground serves as an easy to use educational tool, lowering the entry barrier for newcomers to distributed AI while also offering a sandbox for rapidly prototyping and comparing FL methods. By democratizing exploration of FL, it promotes broader understanding and adoption of this important paradigm.

Introduction

Federated learning (FL) is a collaborative machine learning paradigm that enables multiple clients to train a shared model without exchanging raw data. This paradigm enables collaboration without exposing raw data, making it especially valuable in domains such as healthcare, finance, and mobile intelligence where privacy is critical (Kairouz et al. 2021; Hard et al. 2019).

Despite its growing importance, FL introduces unique challenges that distinguish it from traditional centralized learning. These include handling non-independent and identically distributed (non-IID) client data, partial client participation, and variability in systems and resources (Tan et al. 2023). Understanding the effects of these challenges on different algorithmic and system-level choices often requires coding expertise and access to specialized infrastructure, which discourage newcomers from experimenting with FL.

In the broader machine learning community, open-source tools such as TensorFlow Playground (Smilkov et al. 2017) have shown the power of interactive visualization for lowering entry barriers. By allowing users to tinker with small neural networks directly in the browser, it has helped learners and practitioners intuitively grasp the effects of hyperparameters and model choices without writing code. This com-

ination of simplicity and interactivity has made it a widely used educational and demonstration resource.

Motivated by this success, we present the *Federated Learning Playground*, an interactive, browser-based platform that brings the same spirit of accessibility to the domain of FL. Users can adjust data distributions, hyperparameters, and aggregation strategies, then observe the effects on client and global models through real-time visualizations. The playground serves as both an educational tool and a sandbox that lowers the entry barrier for newcomers exploring privacy-preserving and distributed AI.

System Design and Implementation

The system is built around an FL engine and orchestration layer that extends the standard centralized training loop in TensorFlow Playground. Implemented as `oneStepFL`, this step (1) samples a fraction of clients, (2) performs local training for a configurable number of epochs, and (3) performs server-side aggregation. When the “Run Federated” option is disabled, the system falls back to the standard centralized training mode, enabling learners to compare the two modes side by side. The entire implementation runs client-side, making it lightweight and easily deployable in a web browser without external dependencies. Combined with a clear and intuitive interface, this design ensures that the *Federated Learning Playground* remains accessible, interactive, and extensible.

FL Aggregation Algorithms. Users can select:

- **FedAvg** (McMahan et al. 2017). Weighted averaging of client deltas by data size.
- **FedProx** (Li et al. 2020). Adds a proximal term to stabilize local updates under data heterogeneity.
- **FedAdam** (Reddi et al. 2021). Applies Adam optimization at the server for adaptive aggregation.
- **SCAFFOLD** (Karimireddy et al. 2020). Uses control variates to correct client drift on non-IID data.

FL Hyperparameters. Users can tune the following:

- **Non-IID partitioning.** Data are split across N virtual clients using a Dirichlet-based sampler to control the degree of heterogeneity (see Figure 1).
- **Clustered FL.** Supports k -means clustering (cosine or ℓ_2) of client updates, illustrating the benefits of grouping when client distributions are multi-modal.

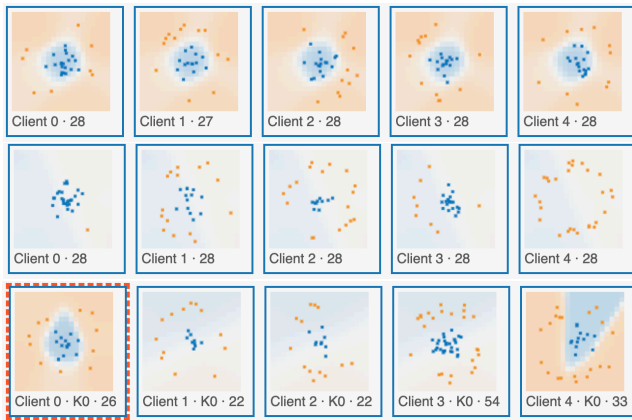


Figure 1: Client Data Distributions. **Top**: IID distribution across clients (post-training). **Middle**: Non-IID distribution (pre-training). **Bottom**: Uniform class distribution with skewed sample sizes, after local training of clients 0 and 4.

- **Differential privacy.** Each client update is clipped to a norm bound and perturbed with Gaussian noise before aggregation, following DP-SGD (Abadi et al. 2016). Visualizations illustrate the privacy–utility trade-off as the noise level varies.
- **Client behavior.** Users can adjust probability of dropout, client learning rate and epochs. Updates are real-time to illustrate impact on communication costs.
- **Local model training.** Training of individual clients can be run for specific clients to visualize the effect of local training and global model aggregation.

Integrating FL support has expanded the codebase to more than twice the size of the original TensorFlow Playground, with the increase coming from additional modules for client sampling, local training, server aggregation, FL algorithms, and privacy mechanisms. Key extensions include:

Weight representation. Weights and Bias are flattened into a single vector (see `nnFlattenWeights`), enabling efficient computation of deltas, proximal terms, control variates, and DP noise in a single pass.

Round orchestration. `oneStepFL` samples clients, runs local epochs with optional FedProx or SCAFFOLD, forms deltas, applies DP clipping/noise, and aggregates. FedAdam is supported via a lightweight server-side Adam optimizer.

Clustering. Clustered FL initializes K models and performs k -means clustering every n rounds after warmup, with client weights averaged within each cluster.

User Interface and Visualizations

The UI adds FL-specific controls, i.e. clients no. & fraction, local epochs, non-IID (α), clustering (k), differential privacy with clipping/noise, and algorithm selector (Figure 2).

We provide visualizations (Figure 3) for:

1. **Client participation** for sampling fairness & coverage.
2. **Communication cost** for efficiency comparisons.



Figure 2: Overview of interface controls. FL-specific controls in Rows 2–4, with original controls retained in Row 1.



Figure 3: Simulated increase in client dropout halfway through training, with visualizations of **Client participation**, **Comms cost** (scales with no. of clients), **Client loss distribution** (max, mean, min), and **Convergence rate**.

3. **Client loss distribution** for heterogeneity & drift.
4. **Convergence Rate** for training progress.
5. **Client Data distribution** for data spread.

Reproducibility and Artifact

The system is released as a static web application hosted on GitHub Pages at <https://oseltamivir.github.io/playground>. The code is open-sourced, with documentation.

Conclusions and Future Work

The *Federated Learning Playground* is a lightweight didactic tool using tiny MLPs and synthetic 2D data to illustrate core FL concepts accessibly. It bridges education and research in FL for students and researchers alike, lowering barriers and inspiring exploration of distributed-AI challenges. Future work will add trustworthy-FL features for explainability and fairness, and broaden the simulator to vertical FL and FL foundation models. (Ren et al. 2025).

Acknowledgments

This work is supported, in part, by the Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore Joint Research Institute (JRI) (Alibaba-NTU-AIR2019B1); and Nanyang Technological University, Singapore; the Ministry of Education, Singapore, under its Academic Research Fund Tier 1 (RG101/24), and conducted as part of the IMDA SG Digital Leadership Accelerator Mentorship Program for Young Talent.

References

- Abadi, M.; Chu, A.; Goodfellow, I.; et al. 2016. Deep Learning with Differential Privacy. In *CCS*.
- Hard, A.; Rao, K.; Mathews, R.; Ramaswamy, S.; Beaufays, F.; Augenstein, S.; Eichner, H.; Kiddon, C.; and Ramage, D. 2019. Federated Learning for Mobile Keyboard Prediction. arXiv:1811.03604.
- Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. 2021. Advances and open problems in federated learning. *Found. Trends Mach. Learn.*, 14(1–2): 1–210.
- Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; and Suresh, A. T. 2020. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. In *ICML*.
- Li, T.; Sahu, A. K.; Talwalkar, A.; and Smith, V. 2020. Federated Optimization in Heterogeneous Networks. In *PMLR*.
- McMahan, H. B.; Moore, E.; Ramage, D.; Hampson, S.; and Arcas, B. A. y. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *AISTATS*.
- Reddi, S.; Charles, Z.; Zaheer, M.; et al. 2021. Adaptive Federated Optimization. In *ICLR*.
- Ren, C.; Yu, H.; Peng, H.; Tang, X.; Li, A.; Gao, Y.; Tan, A. Z.; Zhao, B.; Li, X.; Li, Z.; et al. 2025. Advances and open challenges in federated learning with foundation models. *IEEE COMST*.
- Smilkov, D.; Carter, S.; Sculley, D.; Viégas, F. B.; and Wattenberg, M. 2017. Direct-Manipulation Visualization of Deep Networks. arXiv:1708.03788.
- Tan, A. Z.; Yu, H.; Cui, L.; and Yang, Q. 2023. Towards Personalized Federated Learning. *IEEE TNNLS*, 34(12): 9587–9603.