SIMD: Federated Learning for CFD Simulation

team@simd.space

Abstract

Problem: High-fidelity CFD simulations generate massive datasets (9.6+ GB) distributed across organizations unable to share raw data due to privacy, IP, and transfer costs. Centralized training is infeasible.

Solution: SIMD implements federated learning via Flower AI for collaborative GNN surrogate training without data centralization. We partition 9.6 GB CFD data into 40 shards (simulating organizations), train MeshGraphNet locally, and aggregate via FedAvg—achieving collaboration while preserving privacy.

Approach

Architecture

SIMD uses a **MeshGraphNet-inspired architecture** built with PyTorch Geometric for learning fluid dynamics on unstructured meshes:

- **Input Features:** Node features (N, 8) containing position [x, y, z] and physical properties [T, p, u_x, u_y, u_z]; edge features (E, 4) encoding geometric relationships [dx, dy, dz, r]
- **Model:** 3 NNConv graph convolutional layers with learned edge networks, hidden dimension 128, ReLU activations
- **Output:** Predicted deltas (N, 5) for [ΔT, Δp, Δu_x, Δu_y, Δu_z]

Federated Learning Strategy

We employ **Flower AI's federated learning framework** to orchestrate distributed training:

1. **Data Partitioning:** 40 shards representing independent data silos (organizations/labs)
2. **Local Training:** Each shard trains a GNN model on its private CFD data partition
3. **Weight Aggregation:** FedAvg combines model parameters from all shards into a unified global model
4. **Privacy Preservation:** Raw simulation data (9.6 GB) never leaves local silos; only model weights (~50 MB) are shared

Dataset

FEniCSx/DOLFINx simulations of **nitrogen flow in cylinders**: (1) Hollow (forced convection), (2) Sealed (natural convection). Solves Navier-Stokes + Boussinesq buoyancy + temperature diffusion. JSON timestep snapshots. HuggingFace dataset

Process

Two-Level Partitioning: (1) Job-level: 40 contiguous shard blocks (orgs), (2) Client-level: shards subdivided for Flower clients. Each job loads shard → trains GNN → shares weights → receives global model → saves checkpoint. W&B monitoring

Experiments

Scaling Strategy Evolution

We iteratively optimized shard size to fit GPU cluster constraints (max 4 concurrent jobs): **4 shards** (>60 min, timeout), **10 shards** (>45 min, timeout), **20 shards** (>30 min, timeout), **40 shards** (~7 min, success). The 40-shard configuration enables efficient parallel federated learning with 4 concurrent jobs processing all shards in waves.

Configuration

Adam optimizer (lr=1e-3), MSE loss, 50 epochs/shard, AMD GPUs, per-channel metrics.

Results

Training & Aggregation

All 40 shards trained successfully (~7 min/shard, 9.6 GB total). FedAvg aggregation across checkpoints achieved: **Privacy** (raw data never leaves silos), **Collaboration** (global model benefits from all data), **Efficiency** (~50 MB weights vs 9.6 GB raw data), **Scalability** (40 distributed jobs coordinated).

Inference Performance

Global model as fast surrogate: **Traditional solver** (hours) → **GNN** (seconds). Use cases: real-time simulation, design optimization. **Frontend:** github.com/simd-ai/f

Next Steps After Hackathon

Immediate: (1) Weighted FedAvg & FedProx for heterogeneous data, (2) Differential privacy (DP-SGD), (3) Production Flower server with versioning, (4) Scale to 100+ shards with FedAsync, (5) Model weight compression.

Long-term: (1) Integration with SIMD P2P GPU network, (2) Trustless verification & compute incentives, (3) Multi-physics (conjugate heat transfer, multiphase), (4) Topology optimization, (5) Secure enclave execution, (6) Enterprise SaaS platform.

Code Repository: https://github.com/simd-ai/fed-train
Dataset: https://huggingface.co/datasets/tihiera/cfd-metadata-json/
Monitoring Dashboard: https://wandb.ai/simd/simd-cfd/workspace