

# Communication-Efficient Federated Learning through Adaptive Weight Clustering and Server-Side Distillation

Vasileios Tsouvalas · Aaqib Saeed · Tanir Ozcelebi · Nirvana Meratnia

Eindhoven University of Technology  
IEEE ICASSP 2024

## 01 Motivation

### The problem

Repeated model exchange makes federated learning (FL) expensive on constrained devices. Compressing client weights alone offers little download savings because aggregation breaks their shared-centroid structure.

### Contributions

- **Bidirectional compression** reduces uploads and downloads without changing the aggregation rule.
- **Adaptive cluster capacity** tracks local representation diversity through effective rank.
- **Five-task evaluation** measures accuracy, traffic and model size across vision and audio.

### Aggregation mismatch

Different clients learn different centroids. Averaging their weights disrupts this structure, so client compression alone does not preserve a compact broadcast model.

### Available data

Clients hold labeled samples and a small unlabeled validation set. The server requires auxiliary out-of-distribution (OOD) data, which need not match client distributions.

### Scope

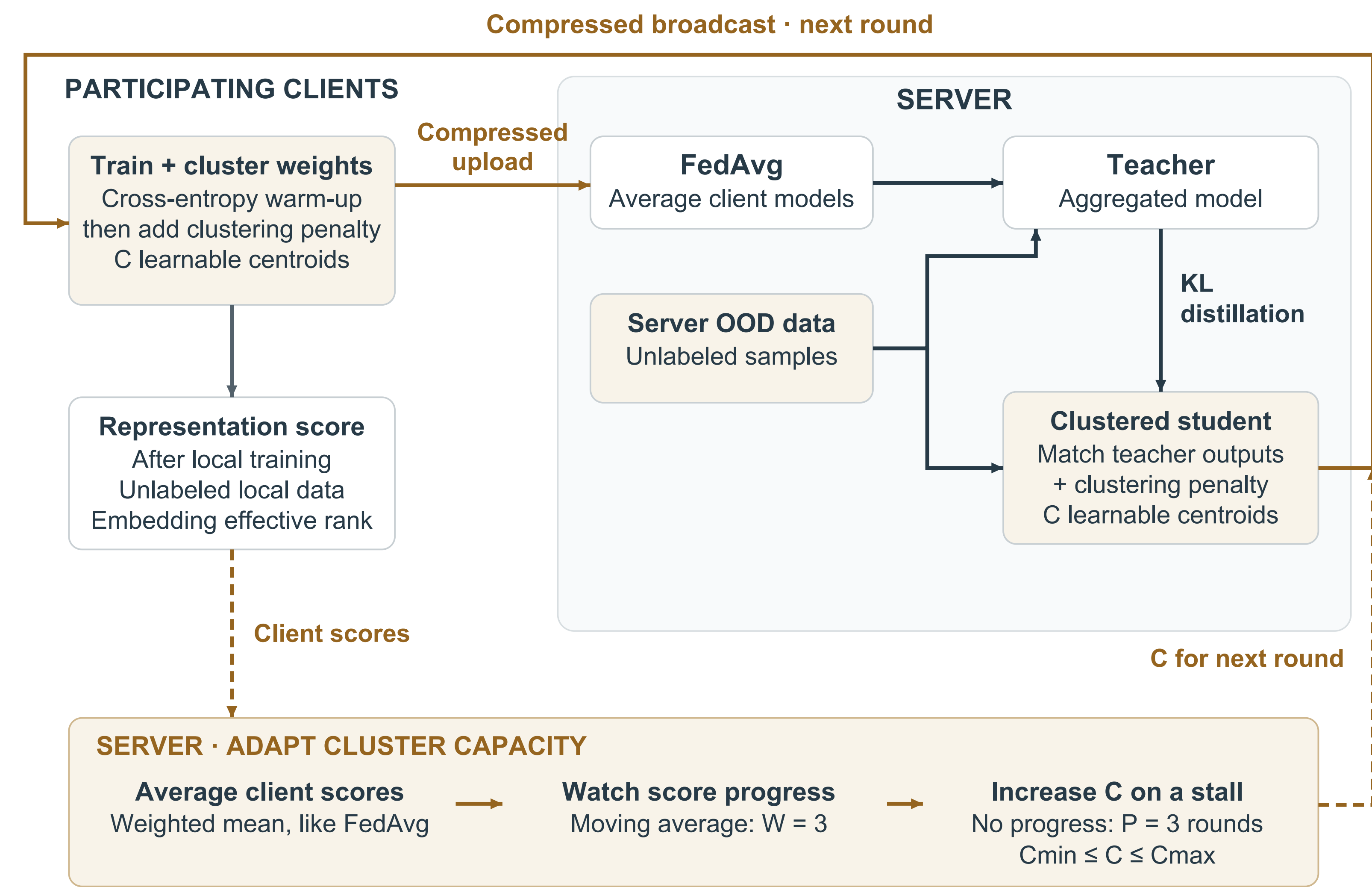
Server distillation adds optimization work. Evaluation uses simulated clients and two-run accuracy means without reported dispersion.

### Notation

Kullback-Leibler divergence (KL) · weight clustering (WC)

## 02 FedCompress

### Compress uploads and downloads with adaptive cluster capacity



### Technical core

$$\mathcal{L}_S = \mathcal{L}_{\text{KL}}(p_T \| p_\theta) + \beta \mathcal{L}_{\text{WC}}(\theta, \mu, C)$$

Teacher  $p_T$  guides student  $p_\theta$ ; the clustering penalty groups weights around C centroids  $\mu$ .  $\beta$  balances compression and imitation.

$$\mathcal{E}_k = \exp\left(-\sum_j r_j \log r_j\right)$$

$r_j$  are normalized singular values of client k's embeddings. Their effective rank tracks representation diversity and guides cluster capacity.

## 03 Evaluation

5.41×

communication reduction

VoxForge versus FedAvg

-0.31 pp

accuracy difference

Same VoxForge comparison

### Communication savings with small accuracy losses

| Dataset        | FedAvg (%) | Δ acc. (pp) | Comm. (×) | Model (×) |
|----------------|------------|-------------|-----------|-----------|
| CIFAR-10       | 86.26      | -1.83       | 4.53      | 5.18      |
| CIFAR-100      | 60.68      | -1.88       | 3.80      | 3.93      |
| PathMNIST      | 88.22      | -1.72       | 4.79      | 5.27      |
| SpeechCommands | 95.75      | -0.42       | 5.04      | 5.09      |
| VoxForge       | 81.05      | -0.31       | 5.41      | 5.64      |

20 clients, 20 rounds, 10 local/server epochs; two-run mean accuracy. Signed accuracy changes and reduction ratios are relative to FedAvg.

### Why compress again at the server?

| CIFAR-100 method           | Δ acc. (pp)  | Comm. (×)   | Model (×)   |
|----------------------------|--------------|-------------|-------------|
| FedZip                     | -2.57        | 1.94        | 2.11        |
| Without server compression | -2.67        | 1.02        | 1.62        |
| <b>FedCompress (Ours)</b>  | <b>-1.88</b> | <b>3.80</b> | <b>3.93</b> |

CIFAR-100, same protocol; accuracy changes and reduction ratios relative to FedAvg.

### Takeaway

**Communication falls across all five tasks**, with small accuracy losses. On CIFAR-100, **server compression** raises the communication reduction from 1.02× to 3.80×.



Paper



Code