Federated Learning with Differential Privacy for End-to-End ASR: Benchmarks, Adaptive Optimizers and Gradient Clipping

Martin Pelikan[1,*] Shams Azam[1,*] Vitaly Feldman[1] Jan Silovsky[1] Kunal Talwar[1] Chris Brinton[2] Tatiana Likhomanenko[1,*]

Neural Information Processing Systems (NeurIPS), 2025

| *Equal Contribution | [1]Apple | [2]Purdue University |

Outline

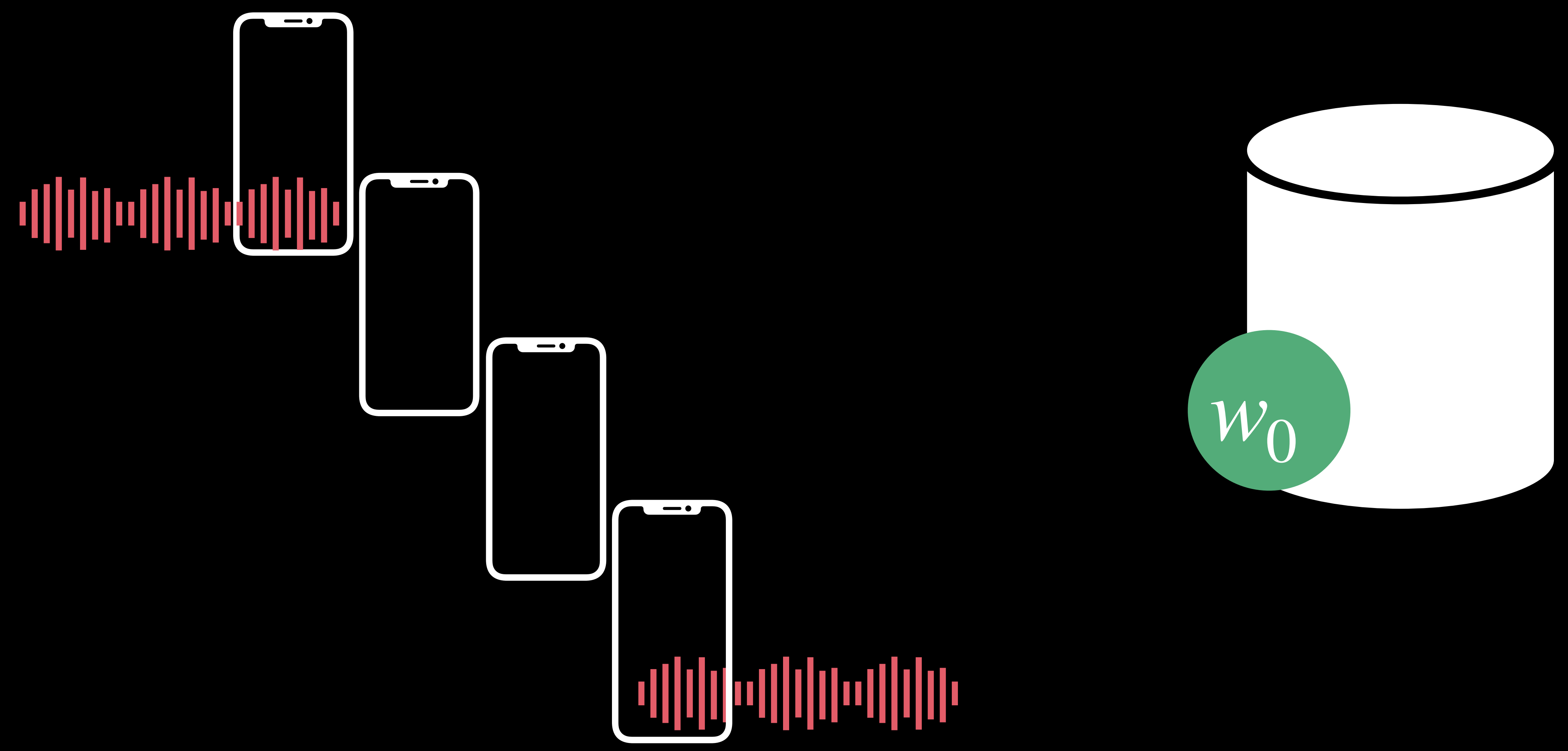
- Introduction
- Problem Statement
- Contributions
- Key Takeaways

Introduction

Terminology and Framework

Federated Learning

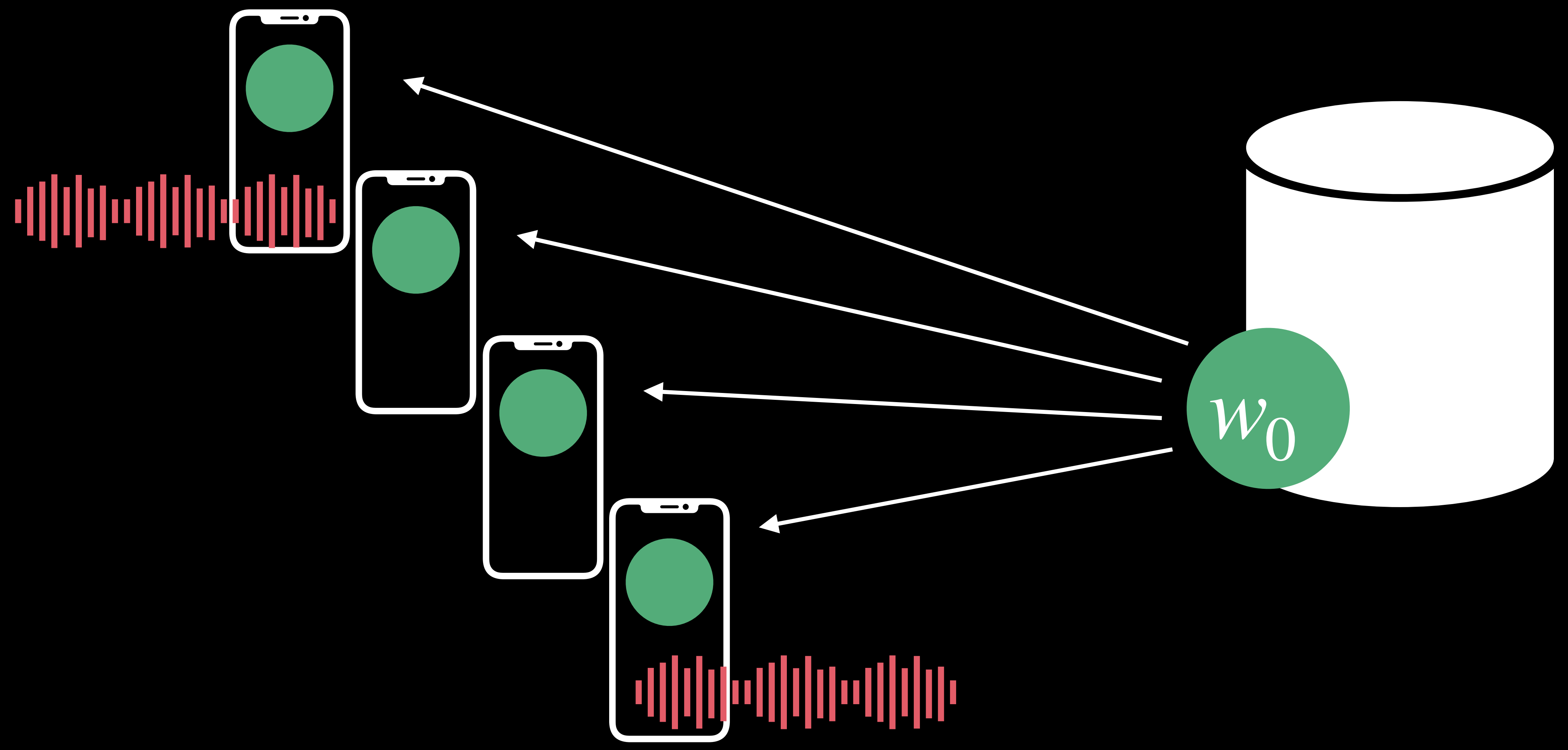
- Initialize server model



Federated Learning

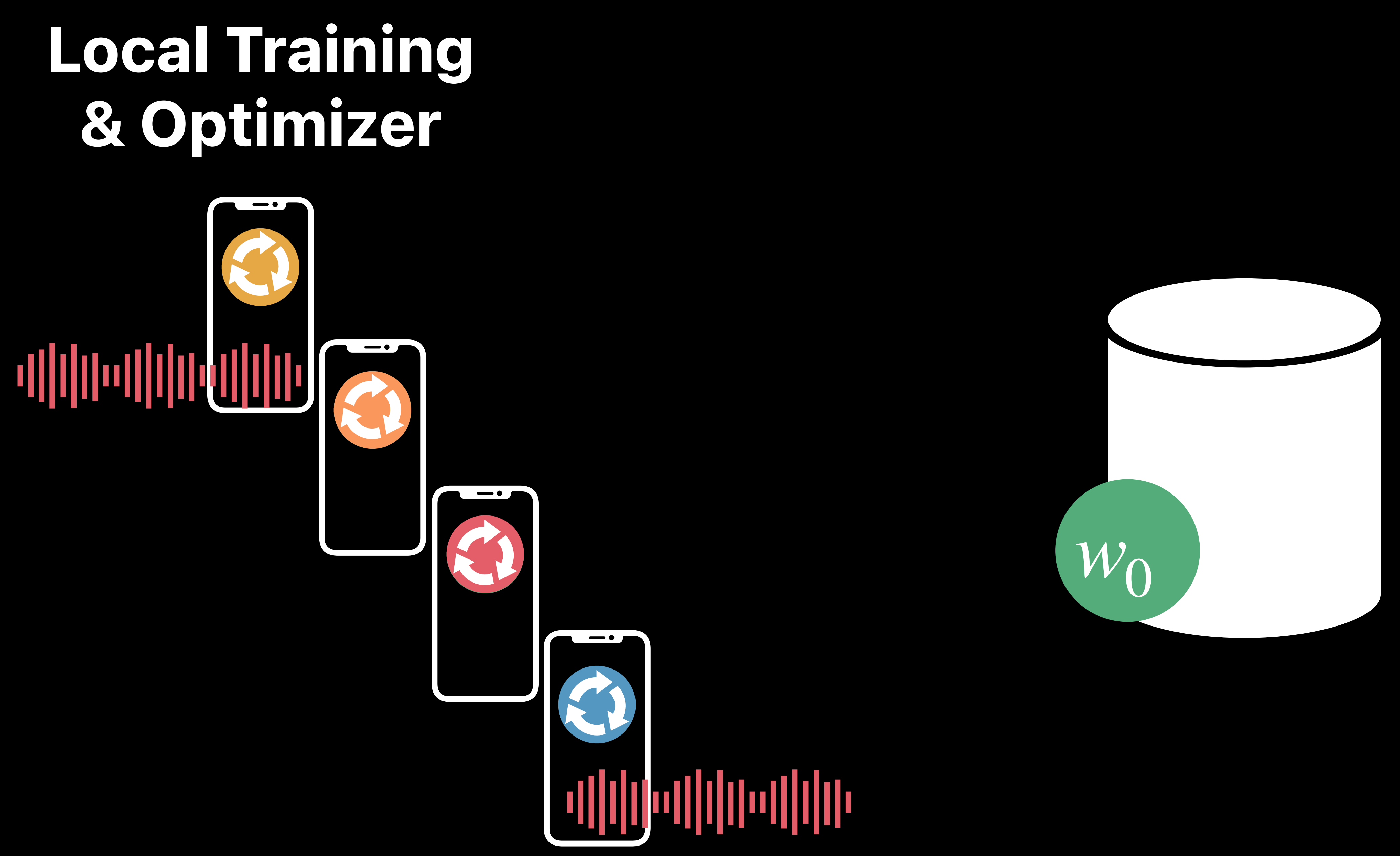
- Initialize server model

- Broadcast server model to a subset of devices



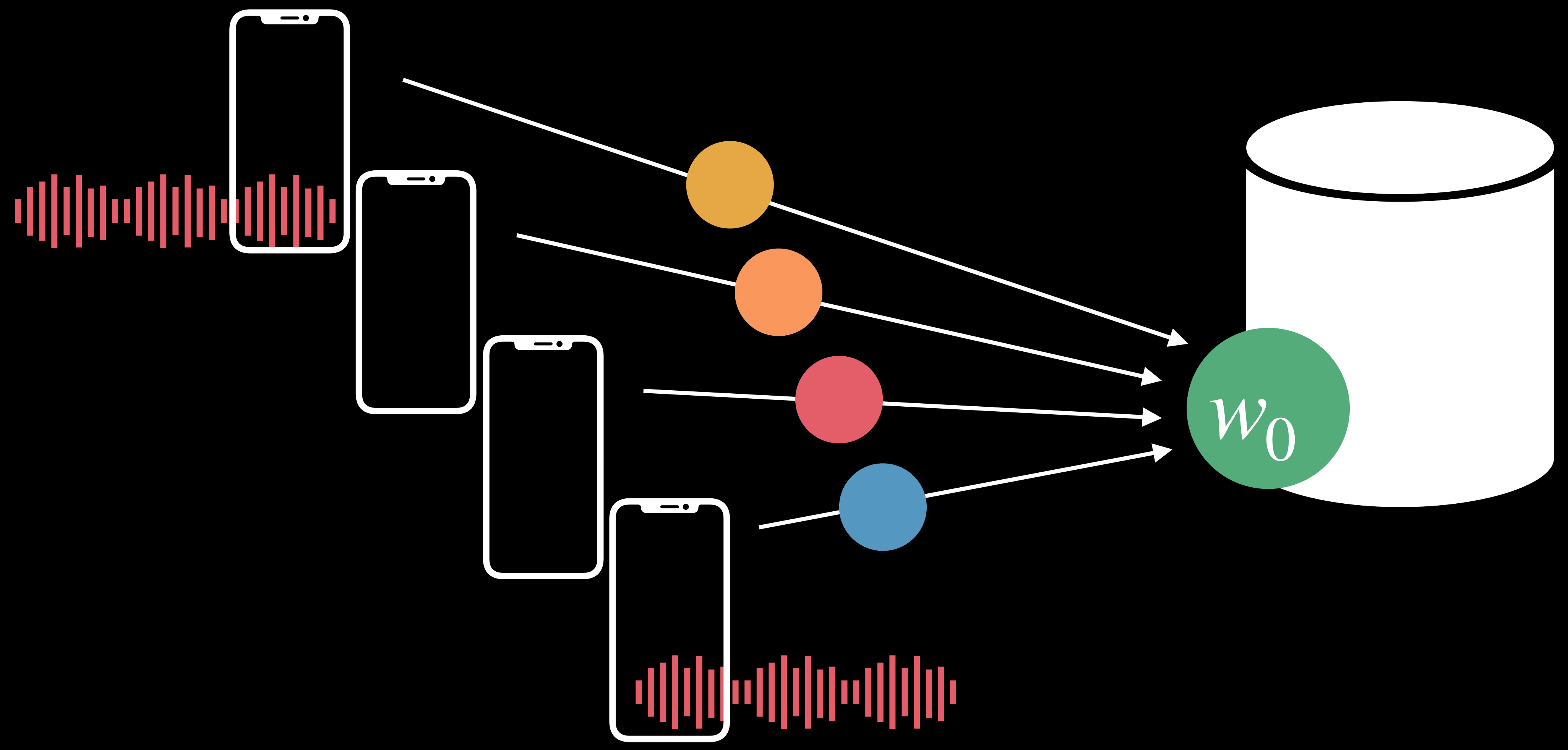
Federated Learning

- Initialize server model
- Broadcast server model to a subset of devices
- Train each local model on client data



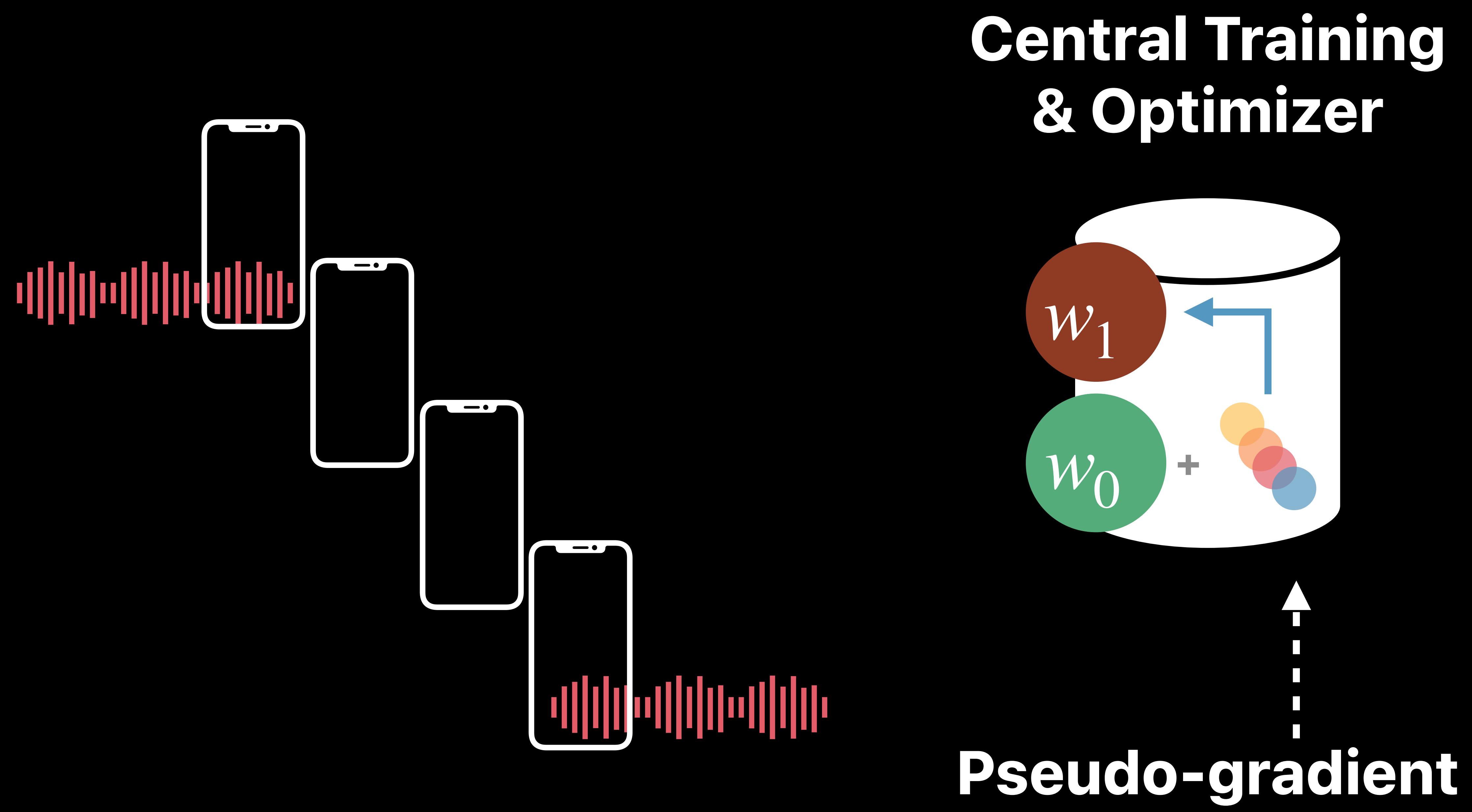
Federated Learning

- Initialize server model
- Broadcast server model to a subset of devices
- Train each local model on client data
- Clients share the model updates back to server



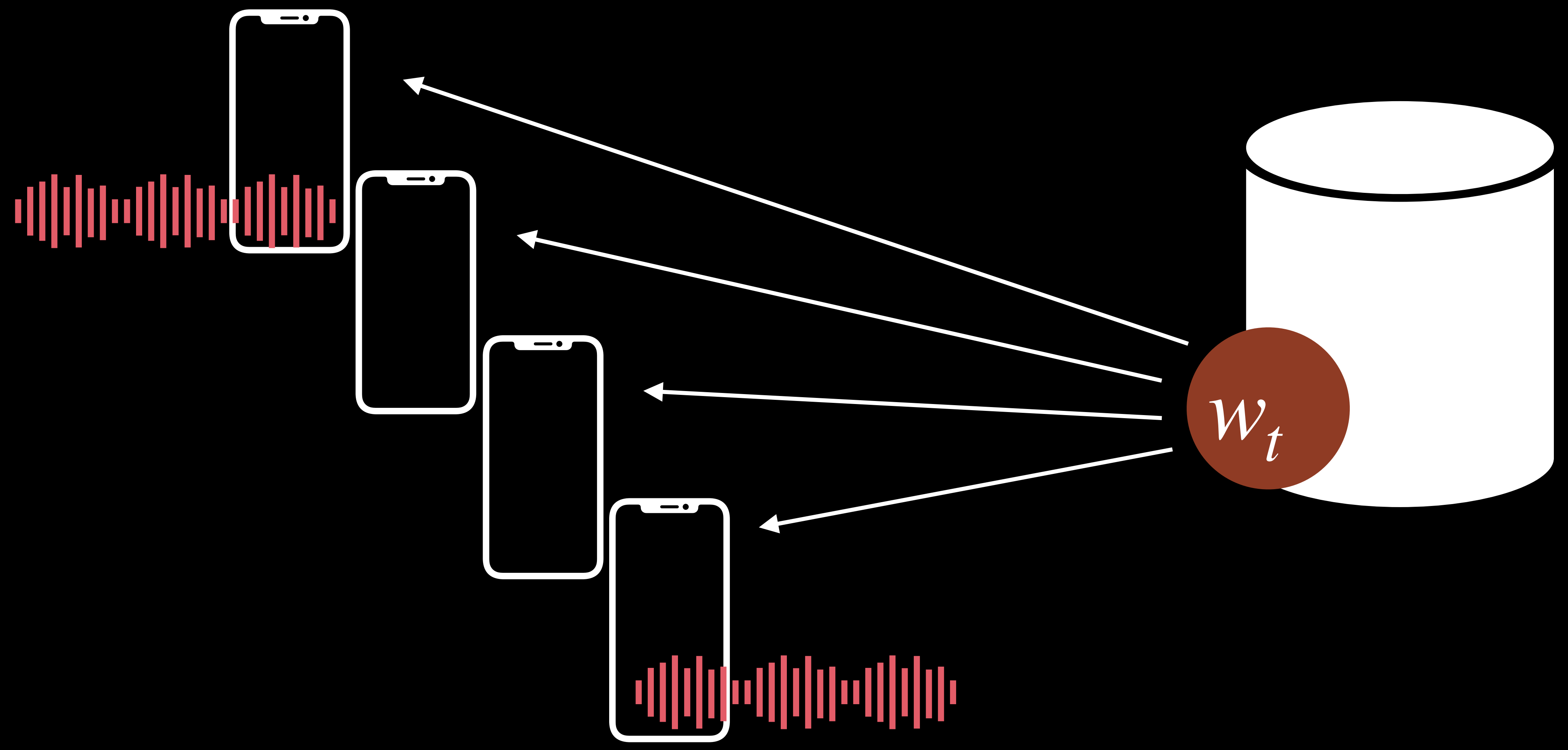
Federated Learning

- Initialize server model
- Broadcast server model to a subset of devices
- Train each local model on client data
- Clients share the model updates back to server
- Update server model by averaging clients updates



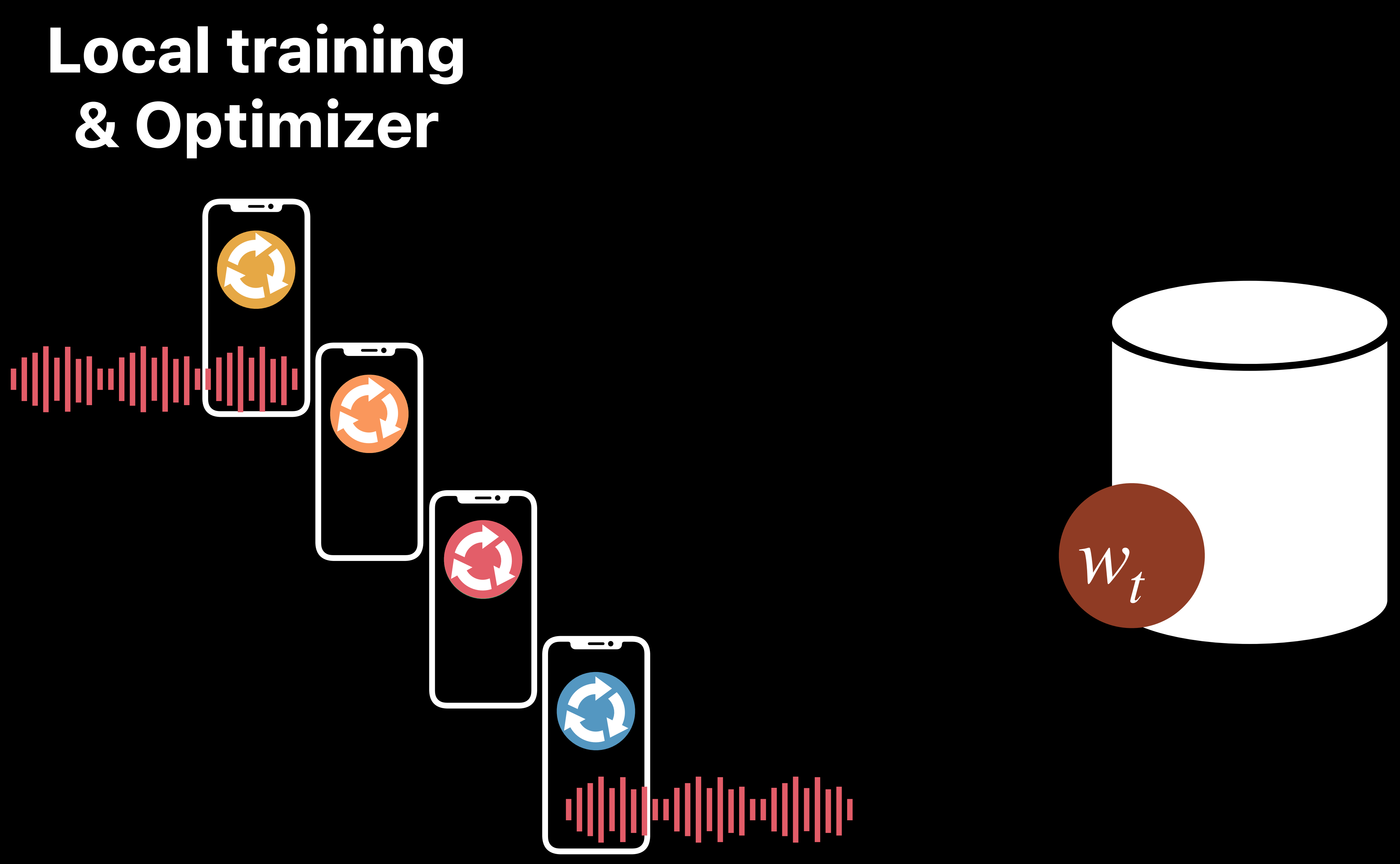
Federated Learning

- Initialize server model
- For every central training step
  - Broadcast server model to a subset of devices
  - Train each local model on client data
  - Clients share the model updates back to server
  - Update server model by averaging clients updates



Federated Learning

- Initialize server model
- For every central training step
  - Broadcast server model to a subset of devices
  - Train for multiple epochs on each client [1]
    - Train each local model on client data
  - Clients share the model updates back to server
  - Update server model by averaging clients updates



[1] Mishchenko, Konstantin, et al. "Proxskip: Yes! local gradient steps provably lead to communication acceleration! finally!" International Conference on Machine Learning. PMLR, 2022

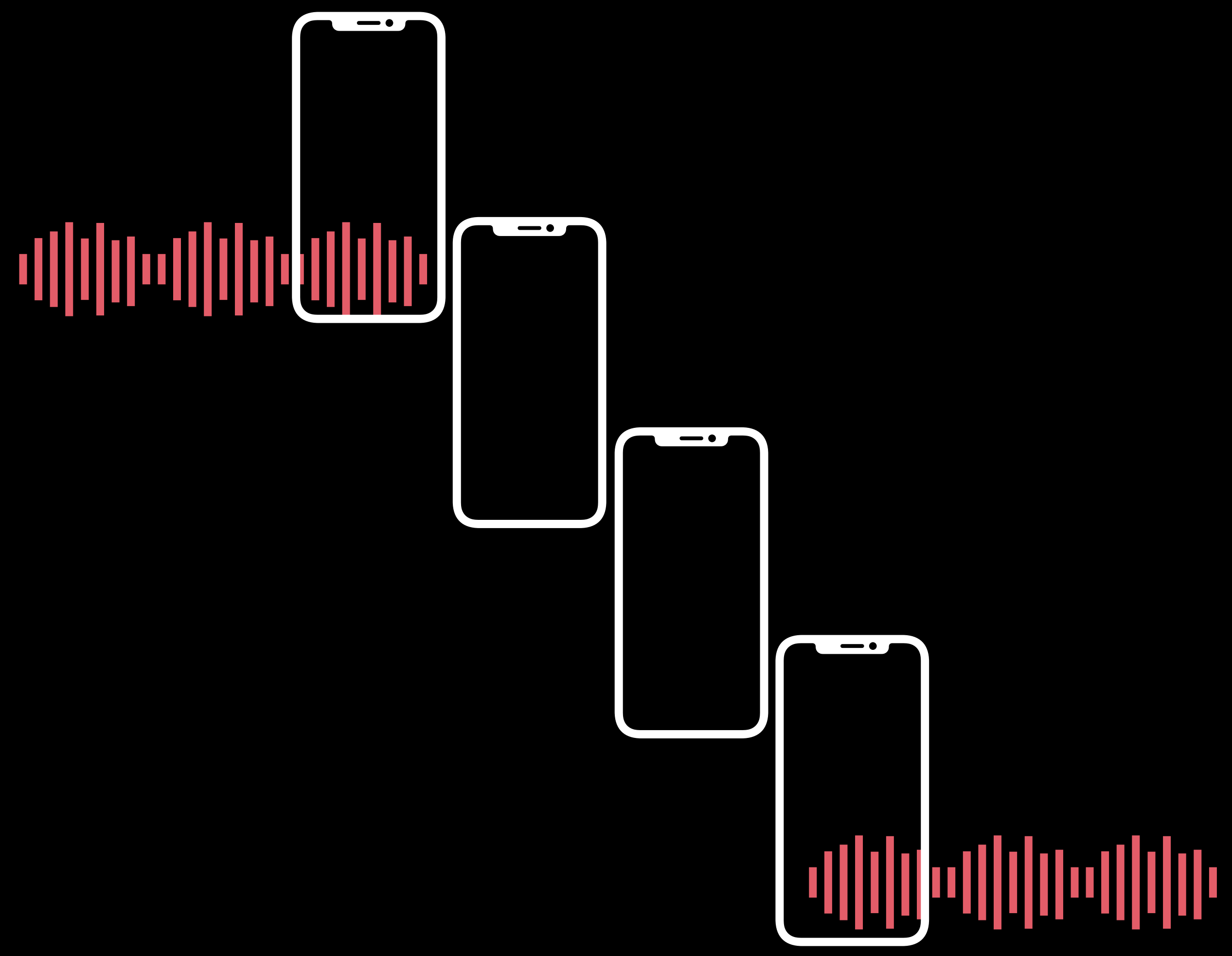
Problem Statement

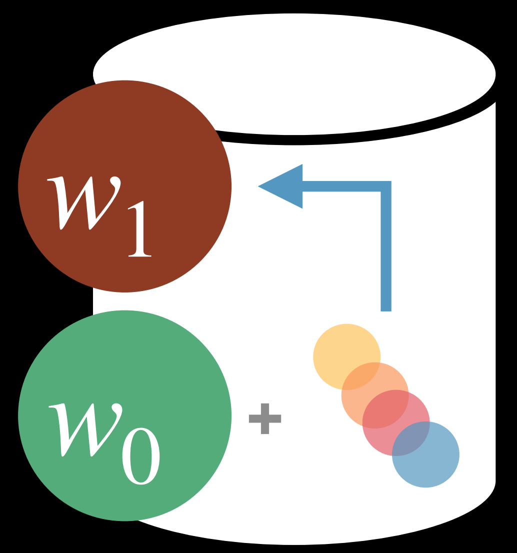
Prior Works, Differential Privacy, and Model Size

Federated Learning for ASR

Prior works

- Initialize server model
- For every central training step
  - Broadcast server model to a subset of devices
  - Train for multiple epochs on each client
    - Train each local model on client data
  - Clients share the model updates back to server
  - Update server model by averaging clients updates





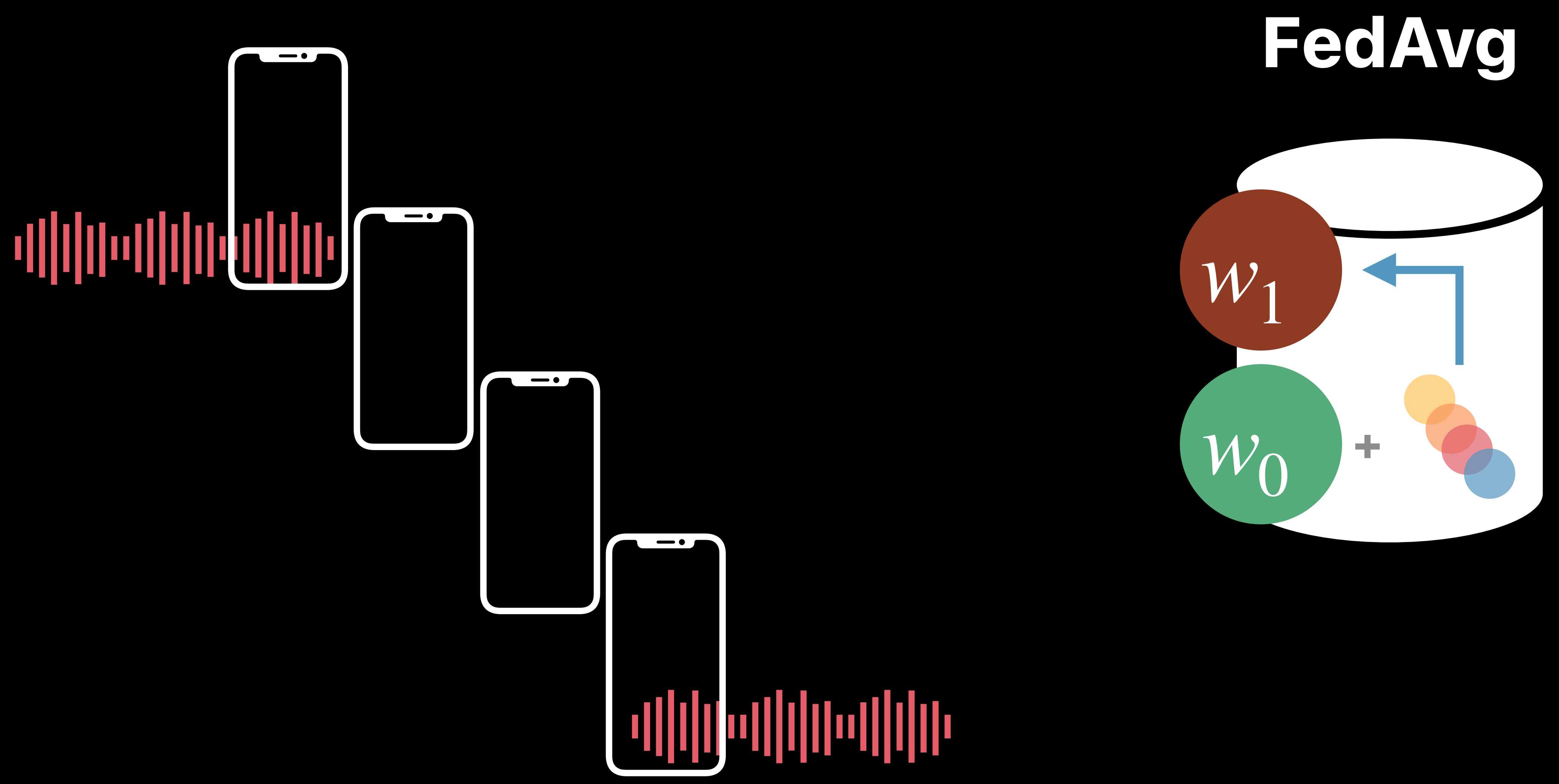
Does not converge!! [1]

[1] Yan Gao et al. "End-to-end Speech Recognition from Federated Acoustic Models," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022.

Federated Learning for ASR

Prior works

- Initialize server model **with pre-trained model** [1]
- For every central training step
  - Broadcast server model to a subset of devices
  - Train for multiple epochs on each client
    - Train each local model on client data
  - Clients share the model updates back to server
  - Update server model by averaging clients updates

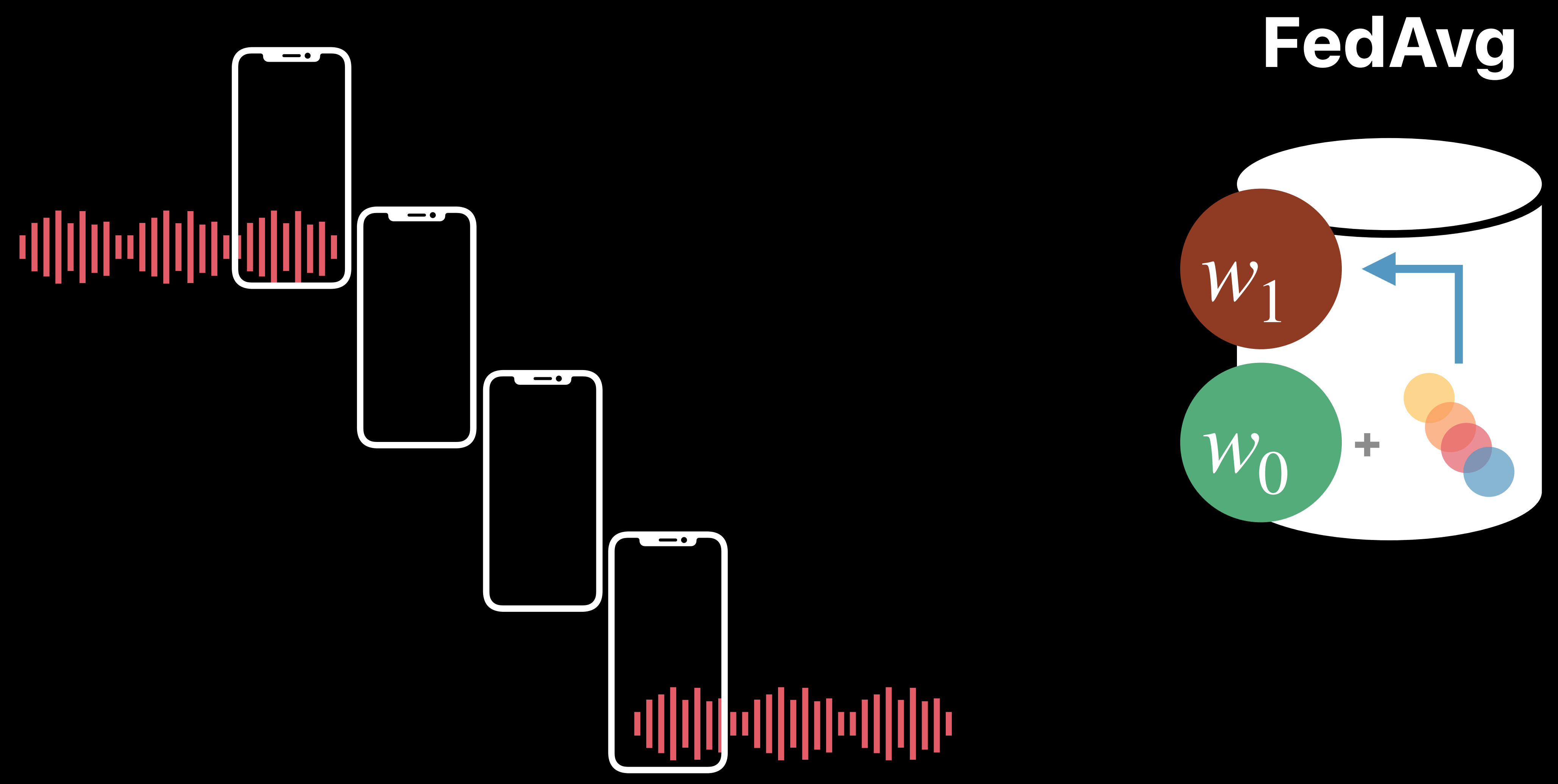


[1] Yan Gao et al. "End-to-end Speech Recognition from Federated Acoustic Models," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022.

Federated Learning for ASR

Prior works

- Initialize server model **with pre-trained model** [1]
- For every central training step **(T=200k)** [2]
  - Broadcast server model to a subset of devices
  - Train for multiple epochs on each client
    - Train each local model on client data
  - Clients share the model updates back to server
  - Update server model by averaging clients updates

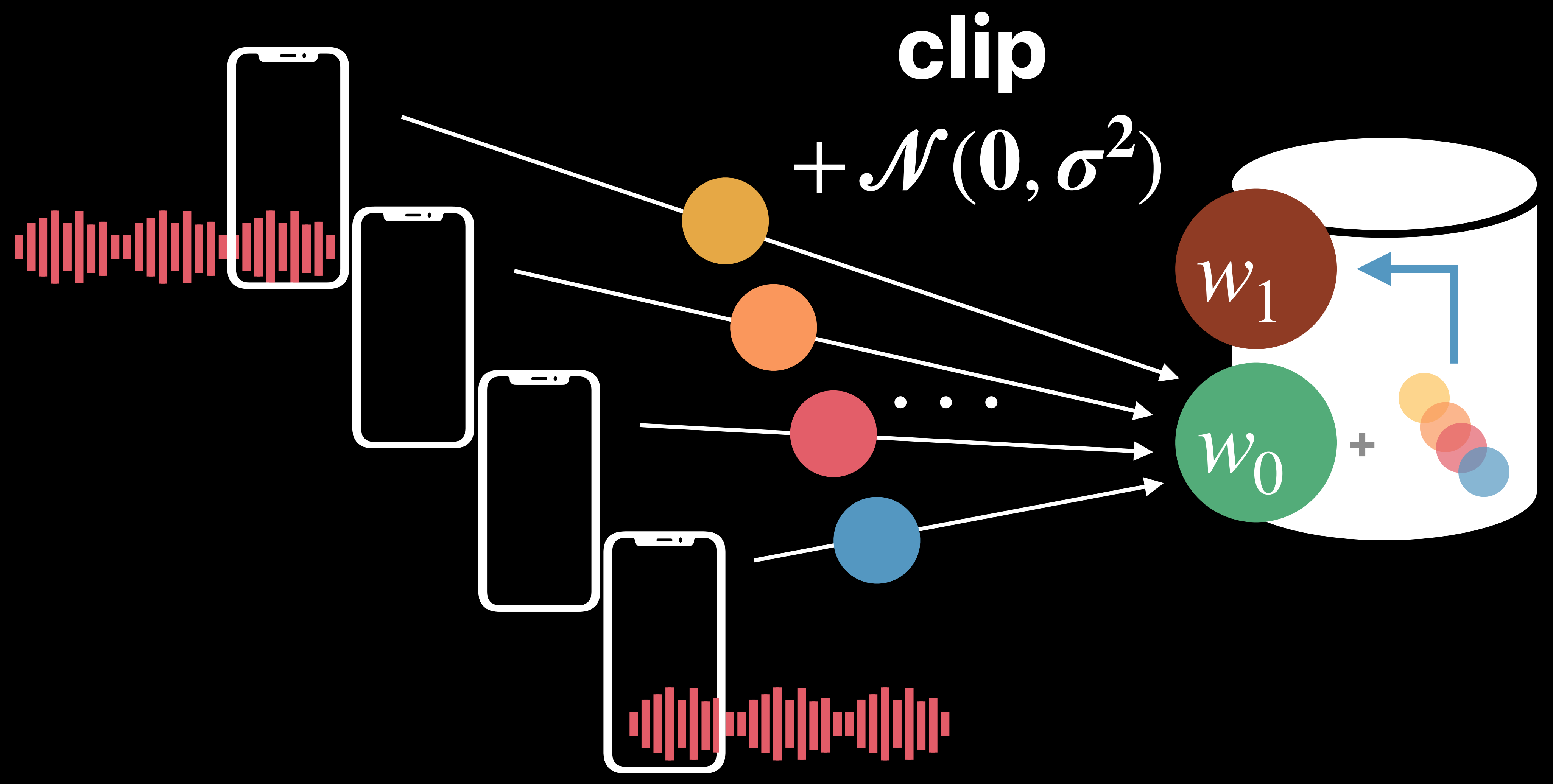


[1] Yan Gao et al. "End-to-end Speech Recognition from Federated Acoustic Models," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022.
[2] Dhruv Guliani et al. "Enabling on-device Training of Speech Recognition Models with Federated Dropout," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022.

Private Federated Learning for ASR

Our Setup

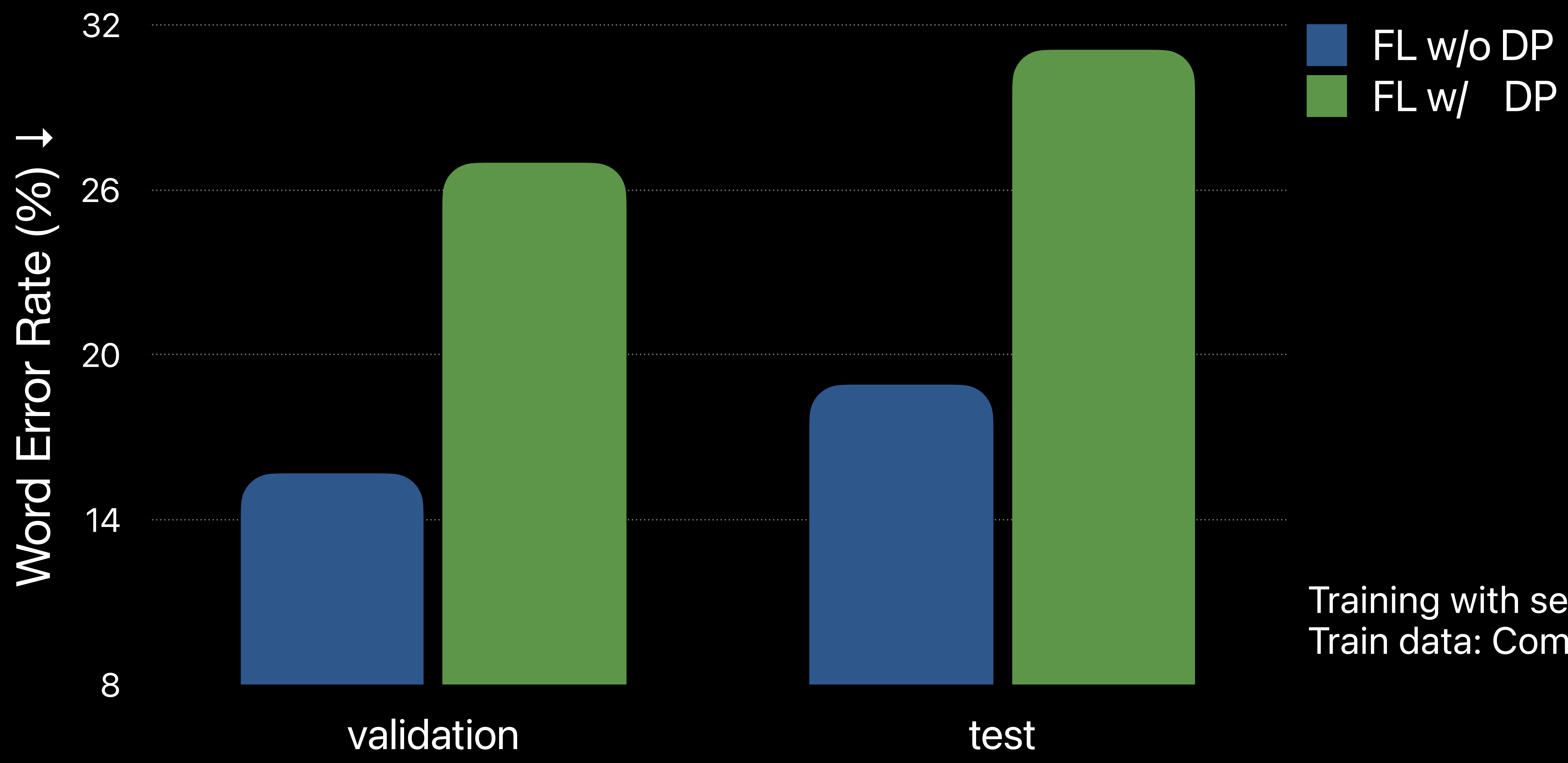
- Initialize server model **from scratch**
- For every central training step **(T=2k)**
  - Broadcast server model to a subset of devices
  - Train for multiple epochs on each client
    - Train each local model on client data
  - **Clip and add noise to model updates for DP**
  - Clients share the model updates back to server
  - Update server model using **pseudo-gradients & adaptive optimization [1]**



[1] Sashank J. Reddi et al. "Adaptive Federated Optimization," in International Conference on Learning Representations (ICLR), 2021.

FL Training with Differential Privacy

Adding noise degrades performance significantly as expected



Training with seed model
Train data: Common Voice (en)

Model Size Comparison

Model sizes in FL are several orders of magnitude smaller than SoTA ASR models

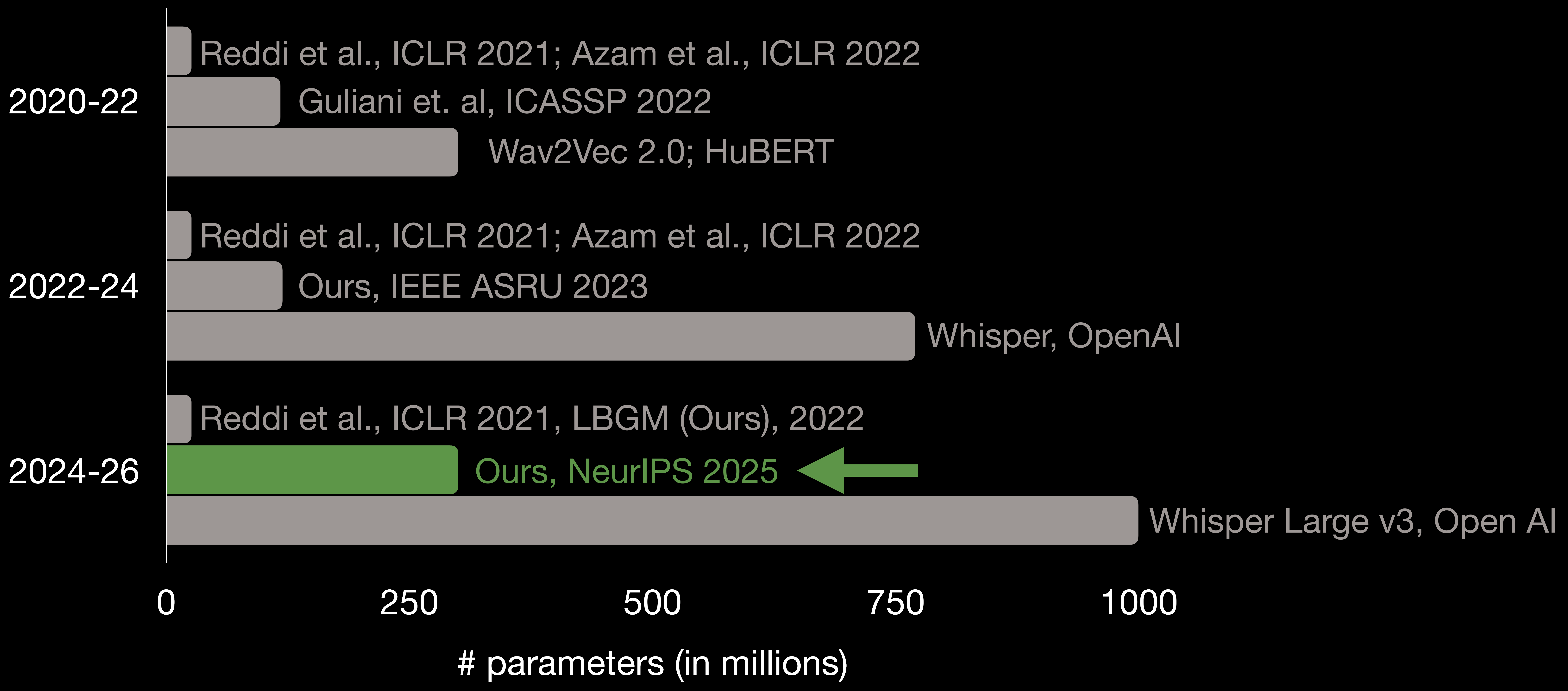
Why Does Model Size Matter?

Different heuristics for optimization of larger models

- **Adaptive optimization** is necessary; SGD underperforms for same compute
  - **Hessian heterogeneity** explains why coordinate-wise adaptive descent is needed
- Adaptive optimization needs warm-up schedule, pre-layer normalization, clipping, etc.
- In the context of FL:
  - Gradient **heterogeneity across clients** further aggravated across some layers
  - Warmup is more essential given client heterogeneity, especially at the start of training
  - **Larger models can easily overfit** on limited local data
  - Communication bottleneck and memory requirements when using Adam, LAMB, etc.
  - How we **clip and apply noise** in the context of Differential Privacy

Model Size Comparison

Model sizes in FL are several orders of magnitude smaller than SoTA ASR models

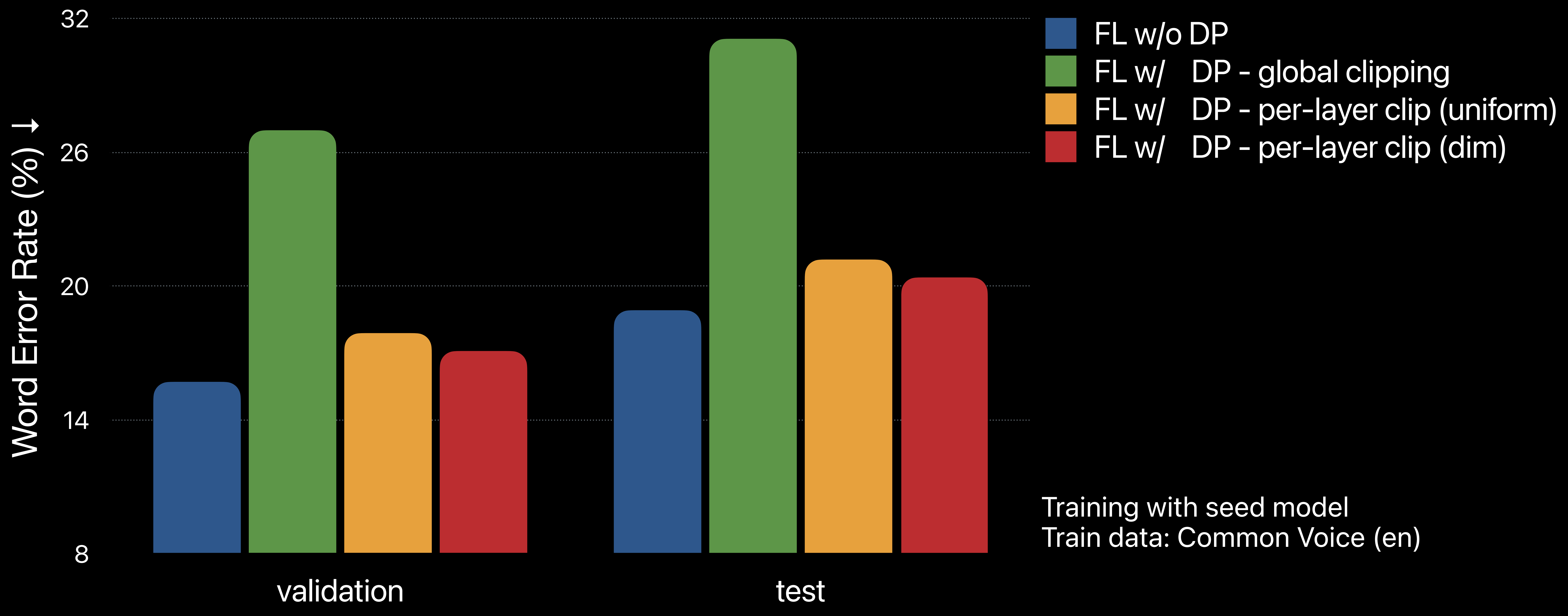


Contributions

Domain Shift, Per-Layer Clipping, and Theoretical Analysis

Enabling FL Training with Differential Privacy

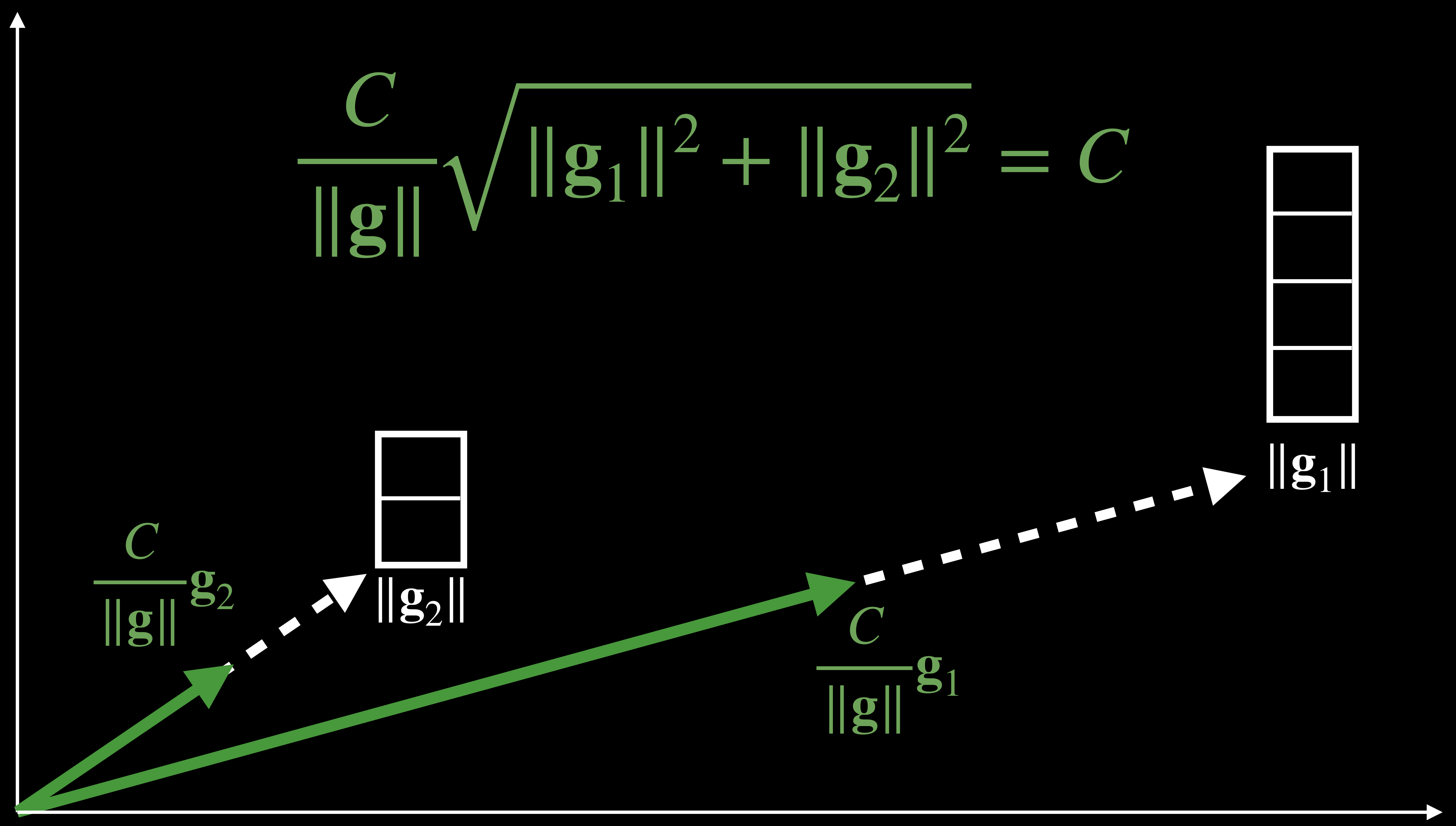
Per-layer clipping extracts better performance for same privacy budget.



Training with seed model
Train data: Common Voice (en)

Why is Per-Layer Clipping Important?

Formulation of global ("flat") clipping

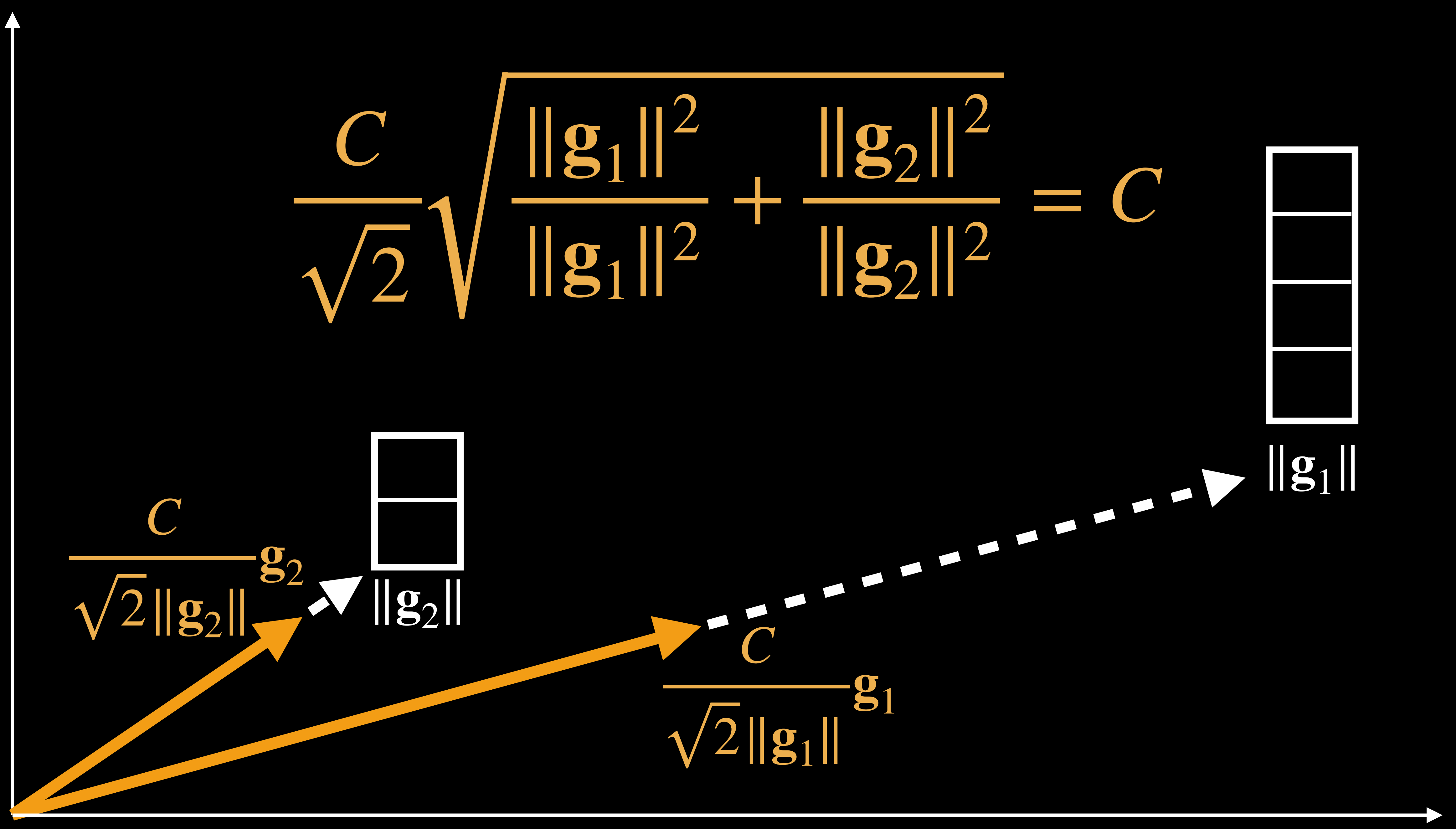


$$\mathrm{Clip}(\mathbf{g}, C) \leftarrow \frac{C}{\|\mathbf{g}\|}\mathbf{g}, \text{ if } \|\mathbf{g}\| > C$$

$$\|\mathbf{g}\| = \sqrt{\|\mathbf{g}_1\|^2 + \|\mathbf{g}_2\|^2}$$

Why is Per-Layer Clipping Important?

Formulation of uniform per-layer clipping

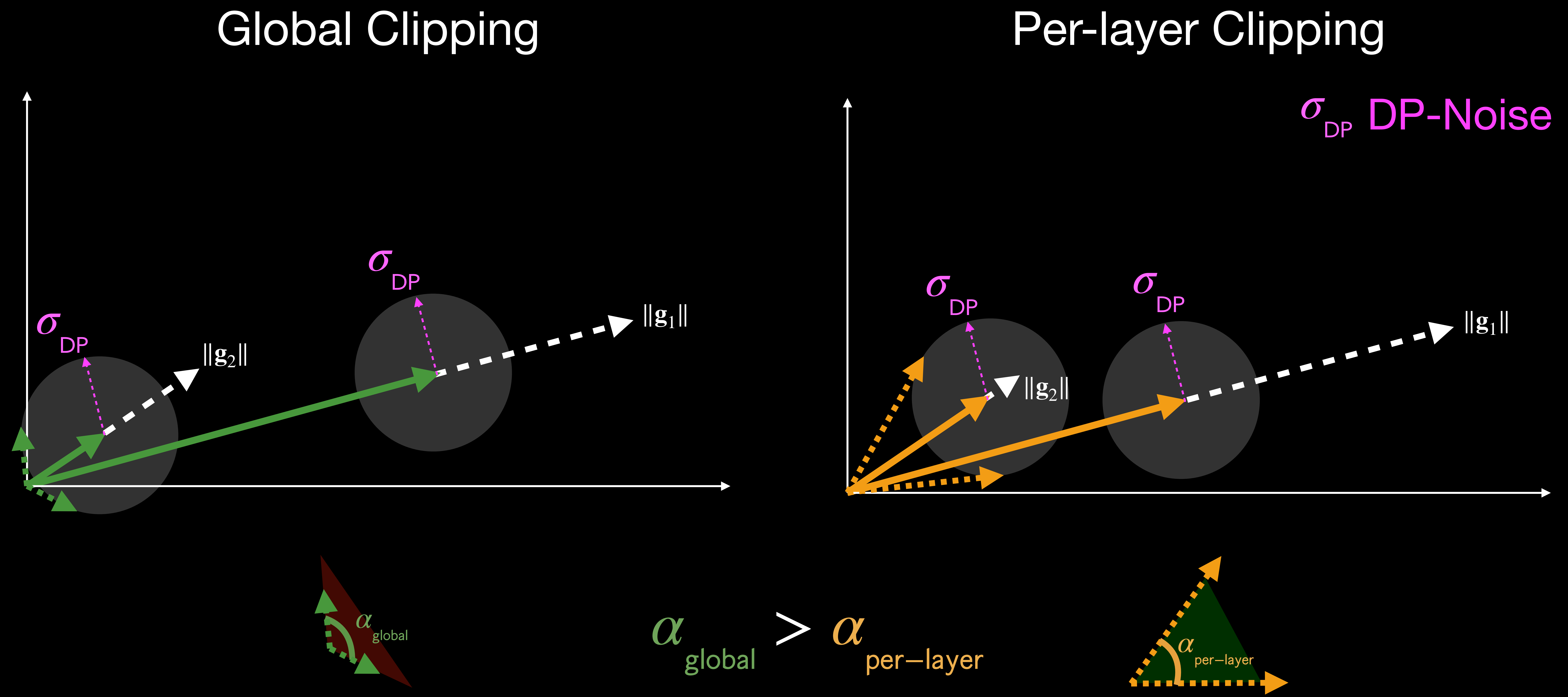


$$\text{Clip}_{\text{uniform}}(\mathbf{g}, C) \leftarrow \left\{ \frac{C\,\mathbf{g}_h}{\sqrt{H}\|\mathbf{g}_h\|} \right\}_{h=0}^{H}$$

$$\text{if } \|\mathbf{g}\| > C;\ \ \|\mathbf{g}\| = \sqrt{\|\mathbf{g}_1\|^2 + \|\mathbf{g}_2\|^2}$$

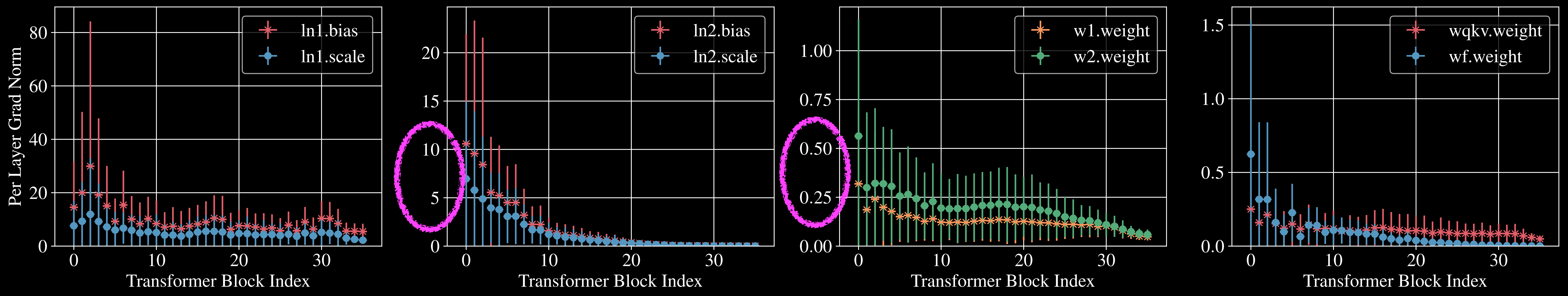
Why is Per-Layer Clipping Important?

Adding DP Noise maintains better signal-to-noise ration (SnR) in per-layer clipping

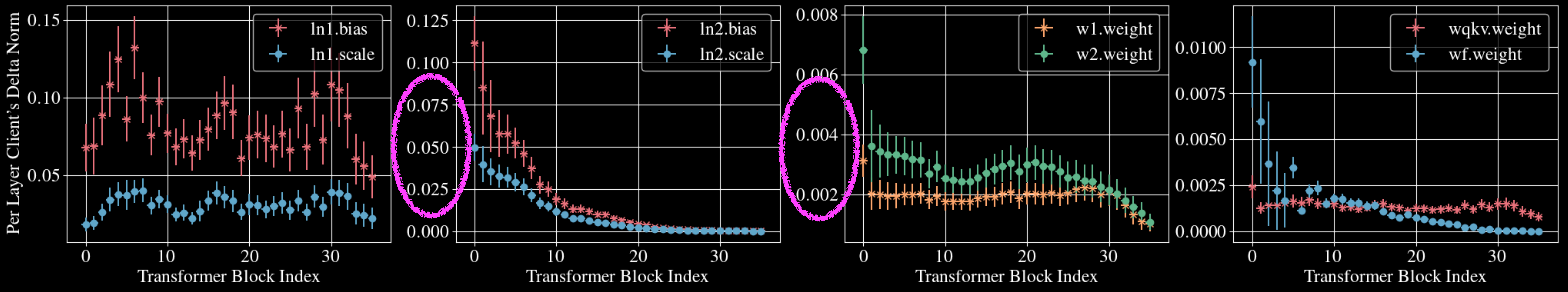


Observing Gradient Imbalance In Practice

Central training



FL (+DP) training



Convergence Analysis of Per-Layer Clipping with LAMB

$$
\begin{aligned}
\frac{\kappa}{T}\sum_{t=0}^{T-1} E_\tau\left[\|\nabla F^{(t)}\|^2\right] \leq\ & \underbrace{\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)}_{\text{optimization}} + \underbrace{\mathcal{O}\left(\frac{\tau\sigma_{glob}^2}{T}\right)}_{\text{global update noise}} + \underbrace{\mathcal{O}\left(\frac{\tau\sigma^2}{T}\right)}_{\text{local update noise}} \\
& \underbrace{+\mathcal{O}\left(C^2\sigma_{DP}^2\sum_{h=1}^{H}R_h^2 d_h\right)}_{\text{differential privacy noise}} + \underbrace{\mathcal{O}\left(\frac{\tau}{T}\sum_{h=1}^{H}\frac{M_h^2}{C_h^2}\right)}_{\text{clipping bias}} \\
& \underbrace{+\mathcal{O}\left(\frac{\tau}{T}\sum_{h=1}^{H}\frac{R_h^2 M_h^2}{C_h^2}\left[\Psi_h^{\text{intra}}+\Psi_h^{\text{inter}}\right]\right)}_{\text{intra– and inter– client update variance}}
\end{aligned}
$$

Convergence Analysis of Per-Layer Clipping with LAMB

Intra- and inter-client update variance

$$r_h^{(t)} \triangleq \frac{\|\boldsymbol{\theta}_h^{(t)}\|}{\|\mathbf{u}_h^{(t)}\|}; \quad \left[\mathbf{u}_h^{(t)}\right]_i = \frac{\left[\mathbf{m}_h^{(t)}\right]_i}{\left[(\mathbf{v}_h^{(t)})^{\frac{1}{2}} + \xi\right]_i}$$

LAMB trust ratio
for layer h

gradient norm
of layer h

$$\frac{\kappa}{T}\sum_{t=0}^{T-1} E_\tau\left[\|\nabla F^{(t)}\|^2\right] \leq \cdots + \mathcal{O}\left(\frac{\tau}{T}\sum_{h=1}^{H}\frac{R_h^2 M_h^2}{C_h^2}\left[\Psi_h^{\text{intra}} + \Psi_h^{\text{inter}}\right]\right)$$

clipping constant
for layer h

Convergence Analysis of Per-Layer Clipping with LAMB

Intra- and inter-client update variance

$$\frac{\kappa}{T}\sum_{t=0}^{T-1} E_\tau\left[\|\nabla F^{(t)}\|^2\right] \leq \cdots + \mathcal{O}\left(\frac{\tau}{T}\sum_{h=1}^{H}\frac{R_h^2 M_h^2}{C_h^2}\left[\Psi_h^{\text{intra}} + \Psi_h^{\text{inter}}\right]\right)$$

- shuffling data on clients,
- data augmentation,
- increasing batch size, etc.

- server-side adaptive optimization,
- anchored optimization such as FedProx,
- weighted averaging of client updates, etc.

Convergence Analysis of Per-Layer Clipping with LAMB

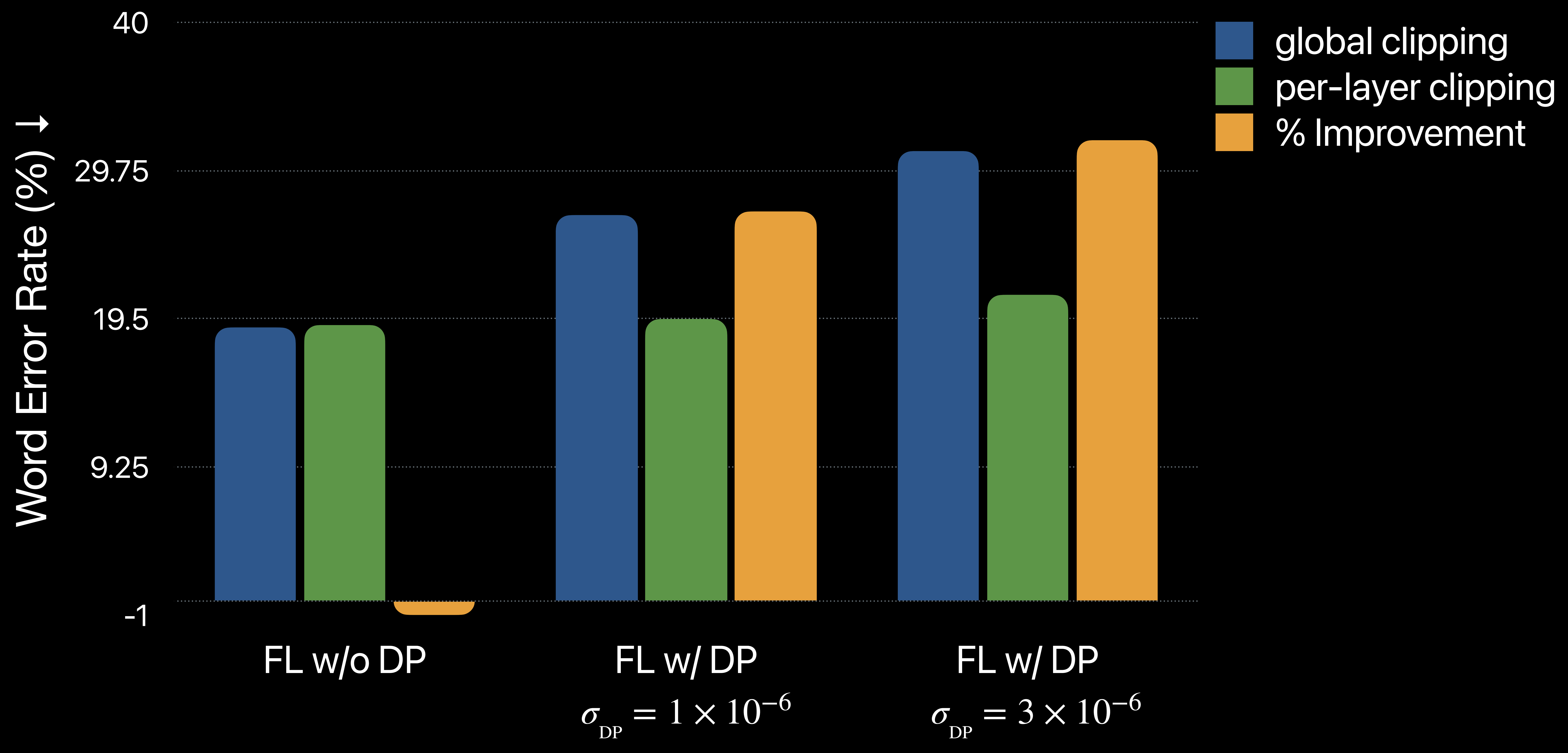
Per-layer intervention should yield better result with deeper models

$$\frac{\kappa}{T}\sum_{t=0}^{T-1} E_\tau\left[\|\nabla F^{(t)}\|^2\right] \leq \cdots + \mathcal{O}\left(\frac{\tau}{T}\sum_{h=1}^{H}\frac{R_h^2 M_h^2}{C_h^2}\left[\Psi_h^{\text{intra}} + \Psi_h^{\text{inter}}\right]\right)$$

decomposition over layers
yields a tighter bound for networks
with more heterogeneous layers

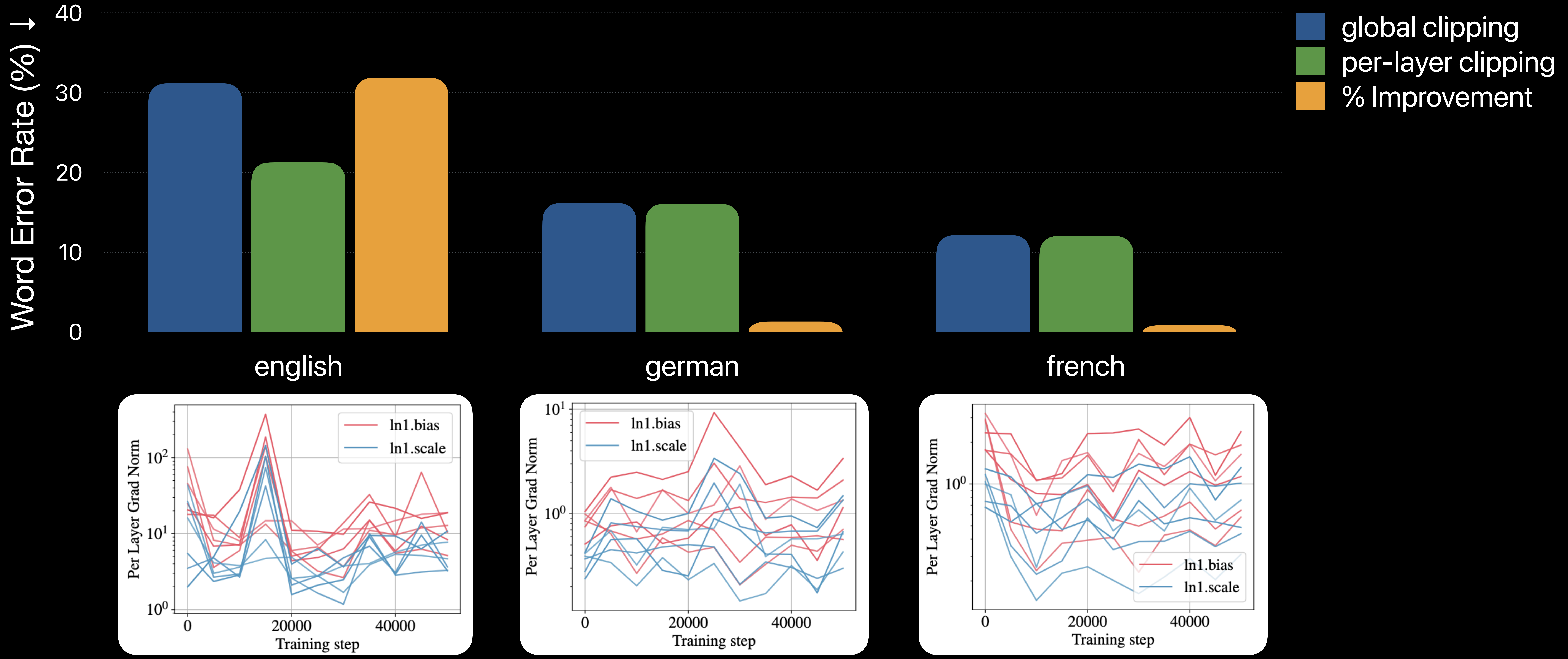
Empirical Support for Theoretical Analysis

As DP noise increases, so does the impact of LAMB + per-layer clipping



Empirical Support for Theoretical Analysis

Higher the intra-layer heterogeneity, higher the improvements



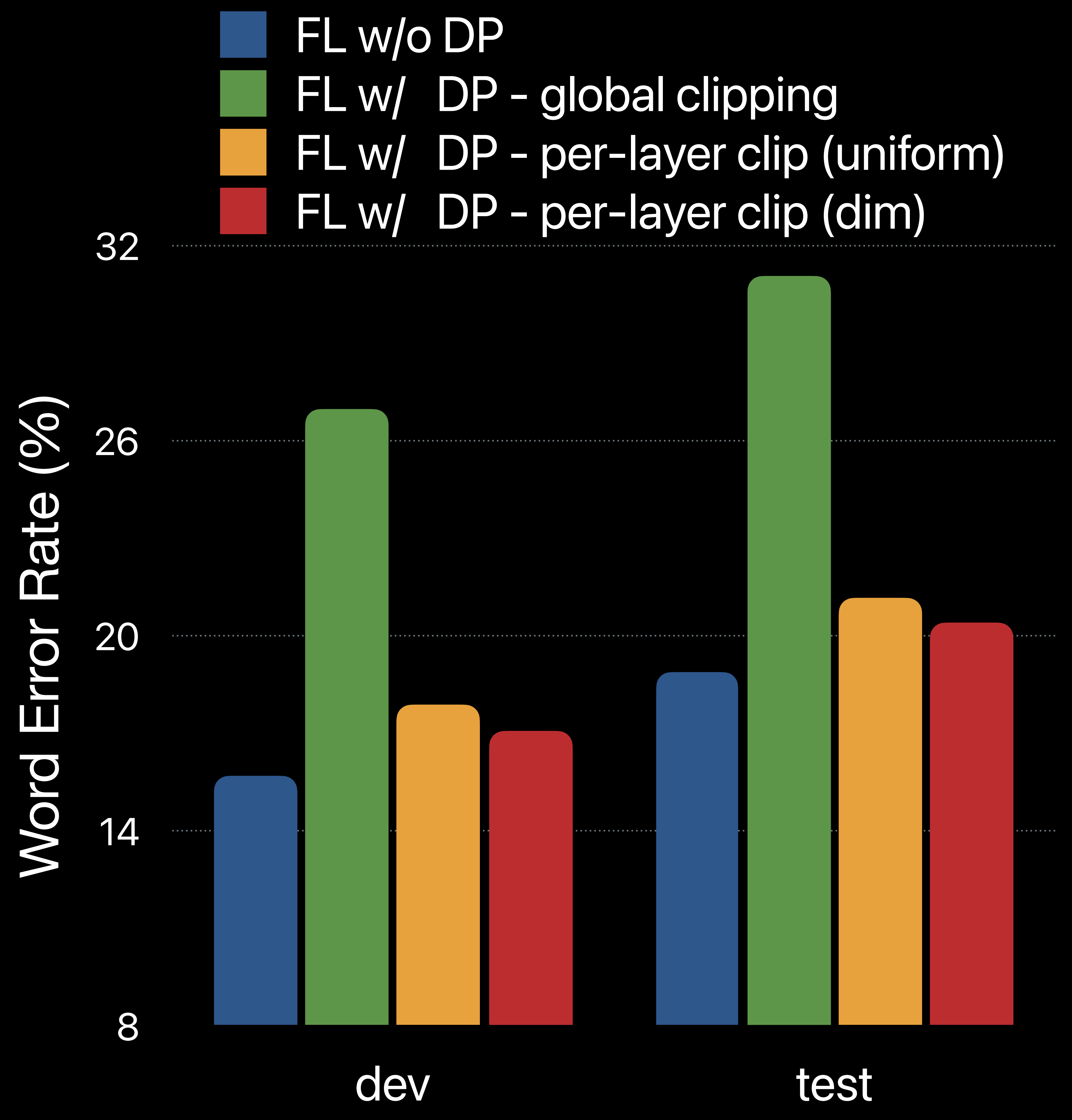
Key Takeaways

Recap of Per-layer Intervention

First benchmark for FL with DP in ASR

Detailed study that includes

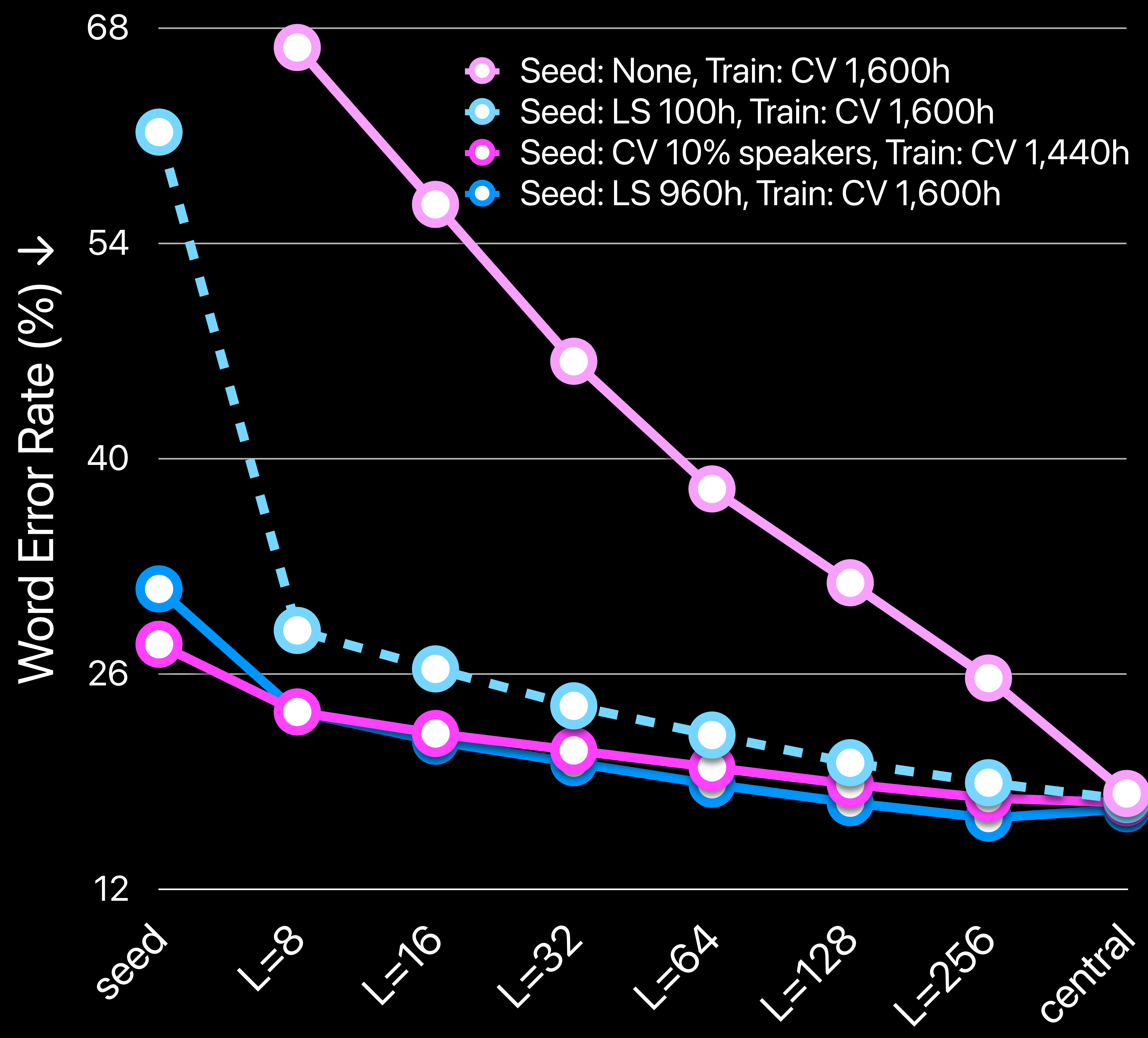
- DP training with **per-layer clipping**
- **layer-wise adaptive optimization**
- **impact of model size** on FL with DP
- theoretically-backed **convergence proof**
- **empirical evidence** of theoretical analysis
  - recovery of prior bounds as special case



Other Contributions

Comprehensive study of FL factors:

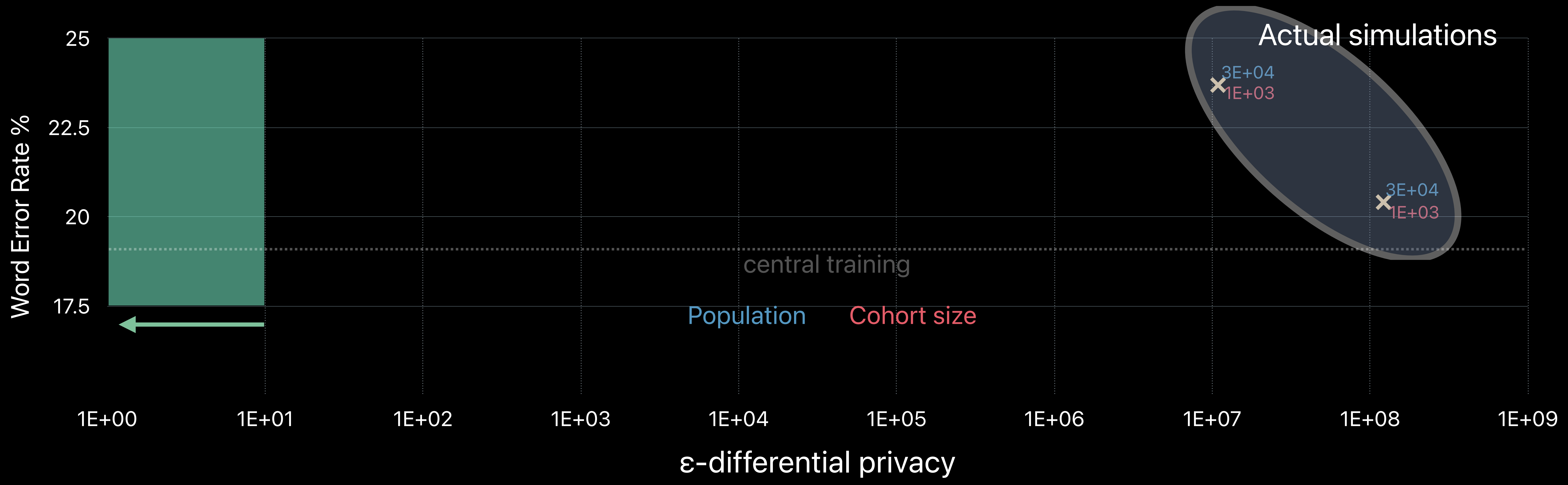
- **data heterogeneity**
  - among clients
  - among seed data and FL data
- **optimization hyperparameters**
  - optimizer: LAMB, LARS, Adam, etc.
  - cohort size, clipping, layer norm, etc.
  - prior works: SpecAugment, FedProx, etc.



L is the number of clients participating in each FL aggregation round

(ϵ, δ)-DP Guarantees

Central seed (train on LS 100h) is fine-tuned with FL & DP (train on CV 1,500h)



(ϵ, δ)-DP Guarantees

Central seed (train on LS 100h) is fine-tuned with FL & DP (train on CV 1,500h)

Get *practical* {quality, (ϵ, δ)-DP} with extrapolation to larger population and cohort