Frugal Decentralized Learning

Anne-Marie Kermarrec

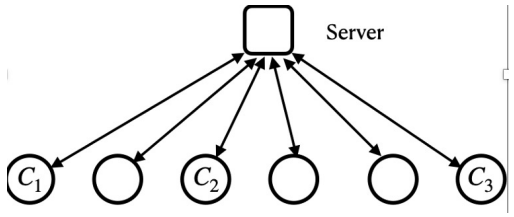
Collaboration with Akash Dhasade, Rafel Pires, Othmane Safsafi , Rishi Sharma

A shift towards distributed learning

Surge in data volumes

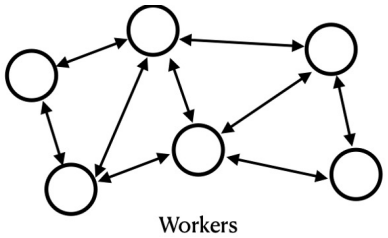
Computational complexity of training

Rising privacy concerns



Federated Learning

Decentralized Learning

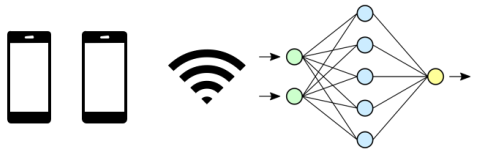


Basic Principle: Let the data stay where it is, learn by exchanging models

Issues of FL & DL

Expensive communication

Low end participating devices must upload/ download deep neural network models



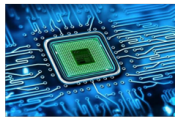
Data heterogeneity

Local data distributions of clients could be arbitrarily different from global distributions.

....

System heterogeneity

Clients differ in their processor, memory, network capabilities, etc.



Data Center

High bandwidth links connecting clusters in data centres

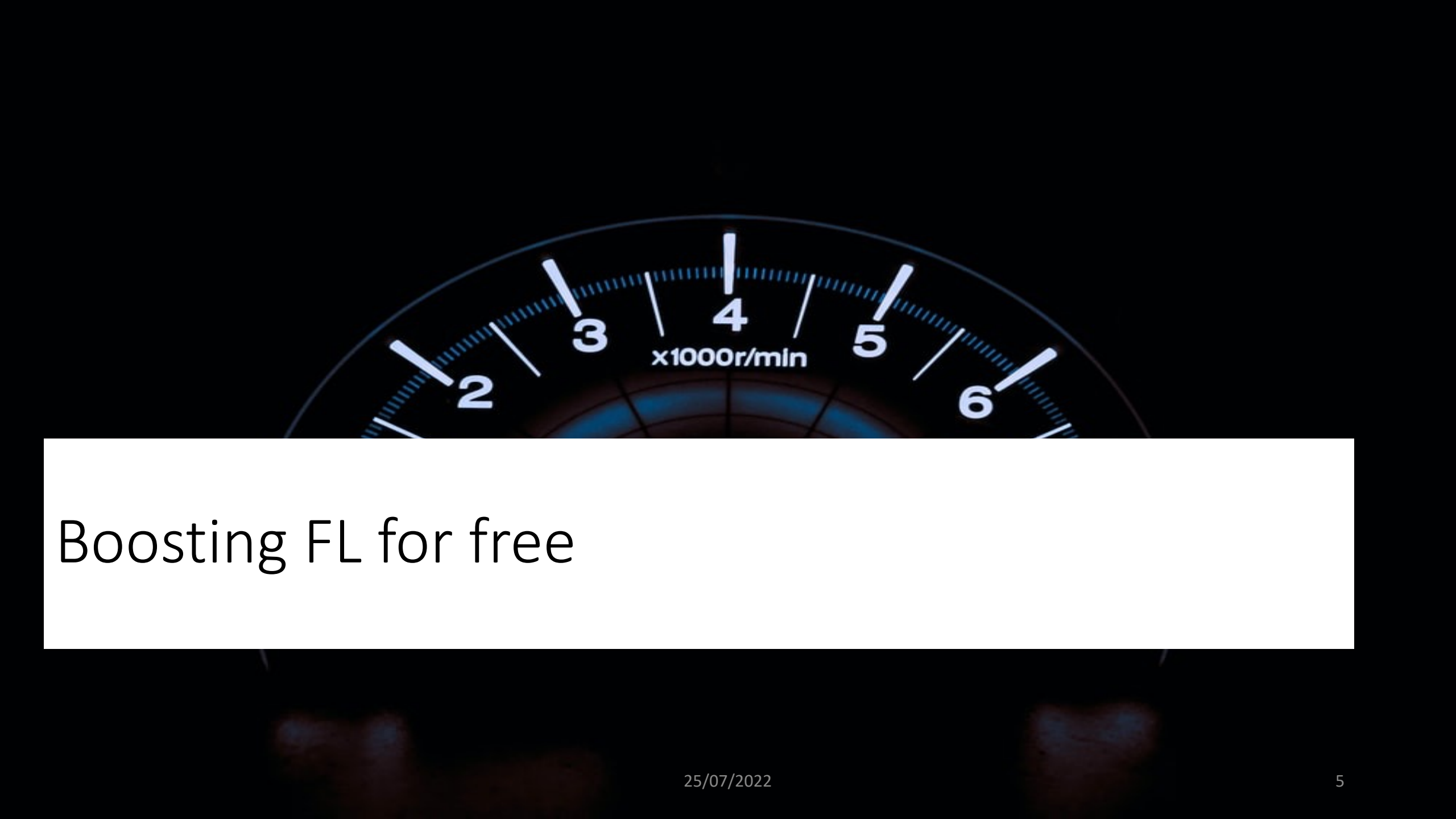
IID data fully available for training.

All nodes are similarly equipped.

Frugal Decentralized Learning: Objectives

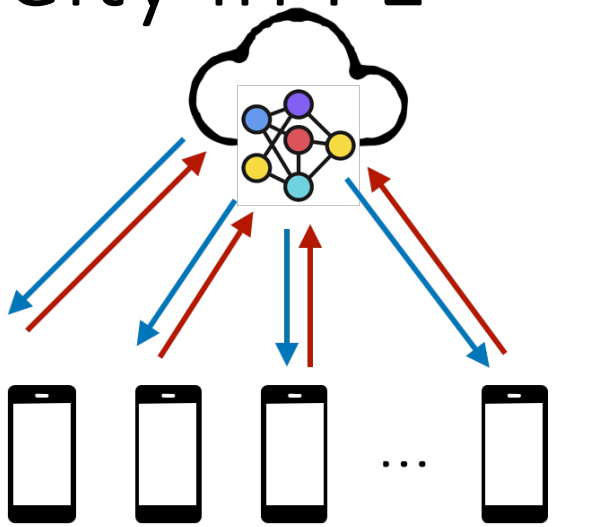
- Compensate for the heterogeneeoous computation capabilities
- Limit the communications for efficiency

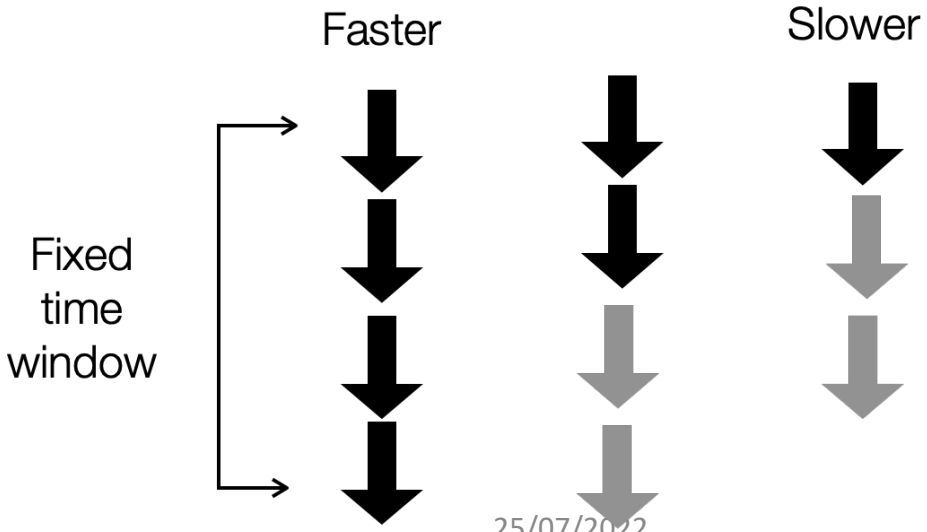
A system view: not everything is needed



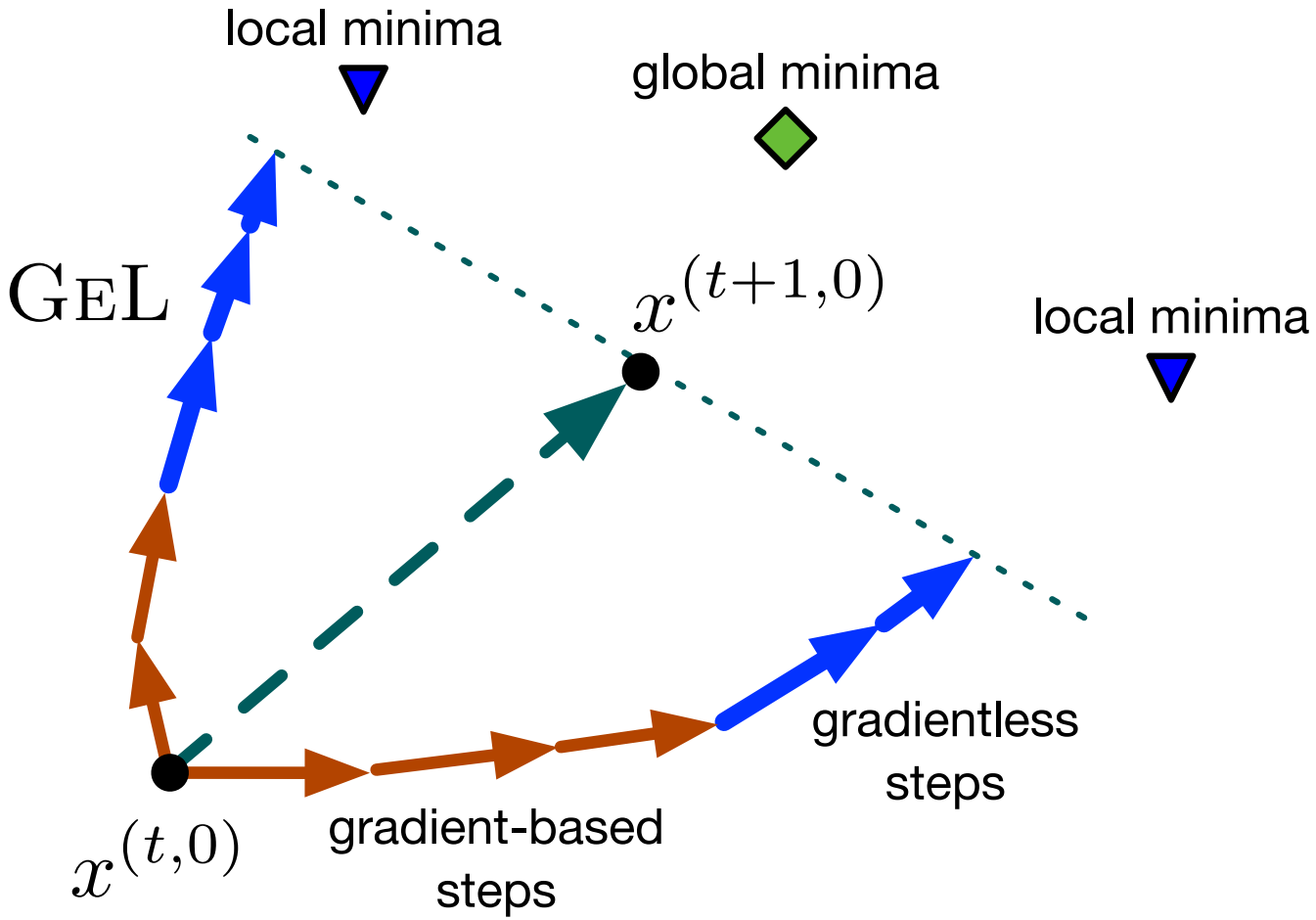
Boosting FL for free

System Heterogeneity in FL





The gist of GeL: exploiting the local momentum for gradientless learning steps

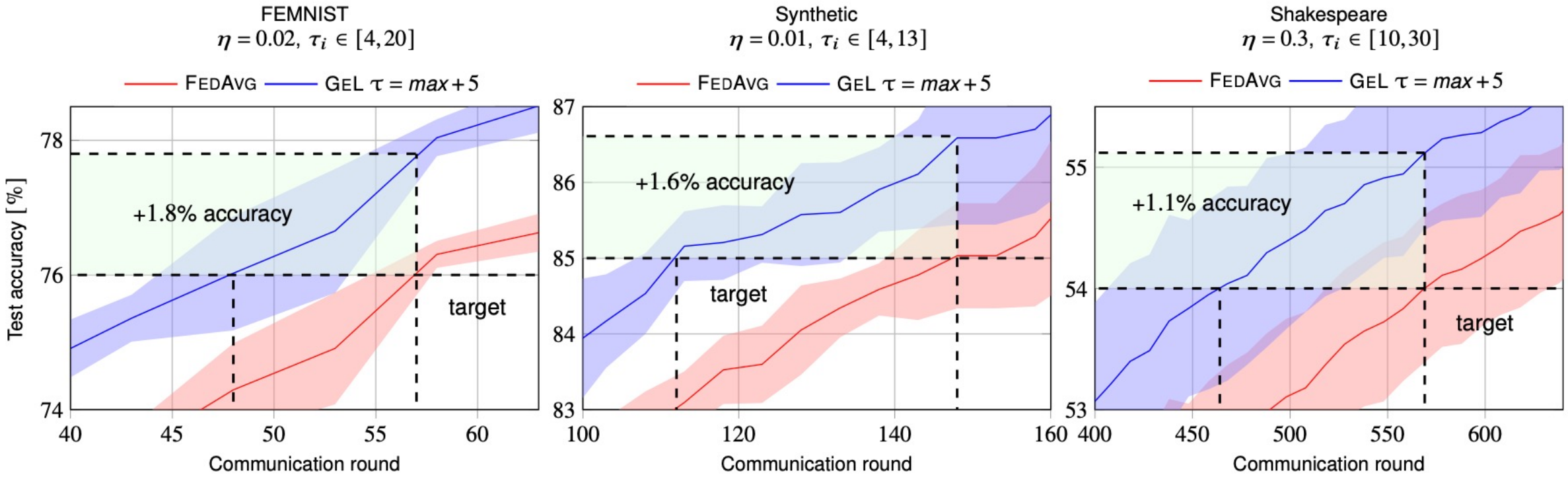


τ is the expected number of updates for each client

τ_i is the number of performed updates for client *i*

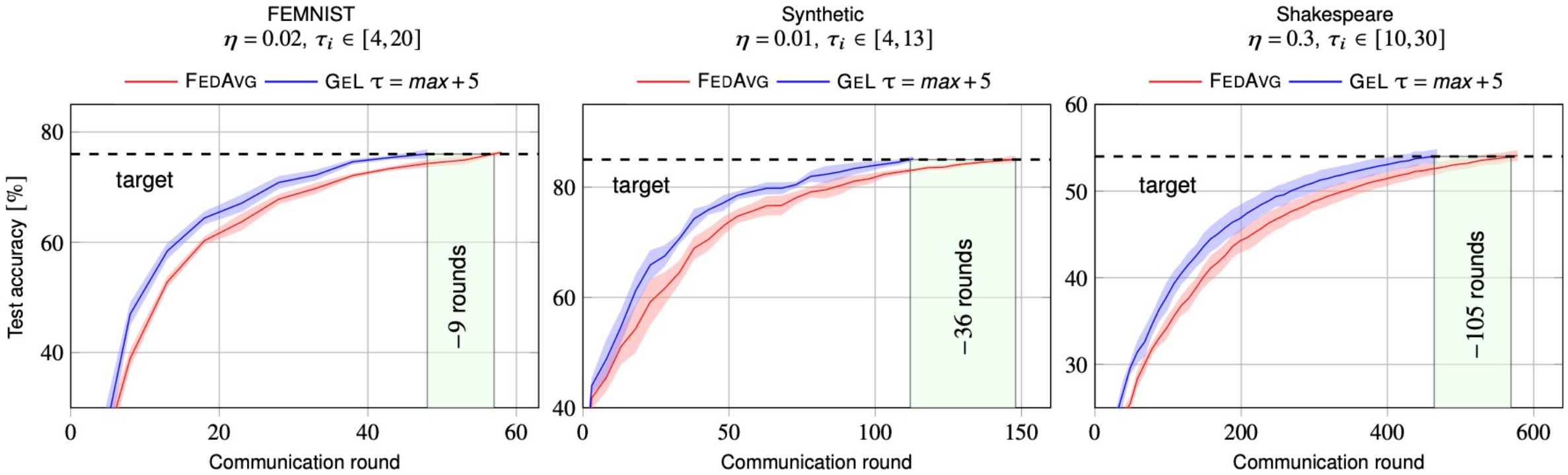
GEL performs τ'
$i = \tau - \tau_i$
guessed updates for every client

Gel provides a better accuracy

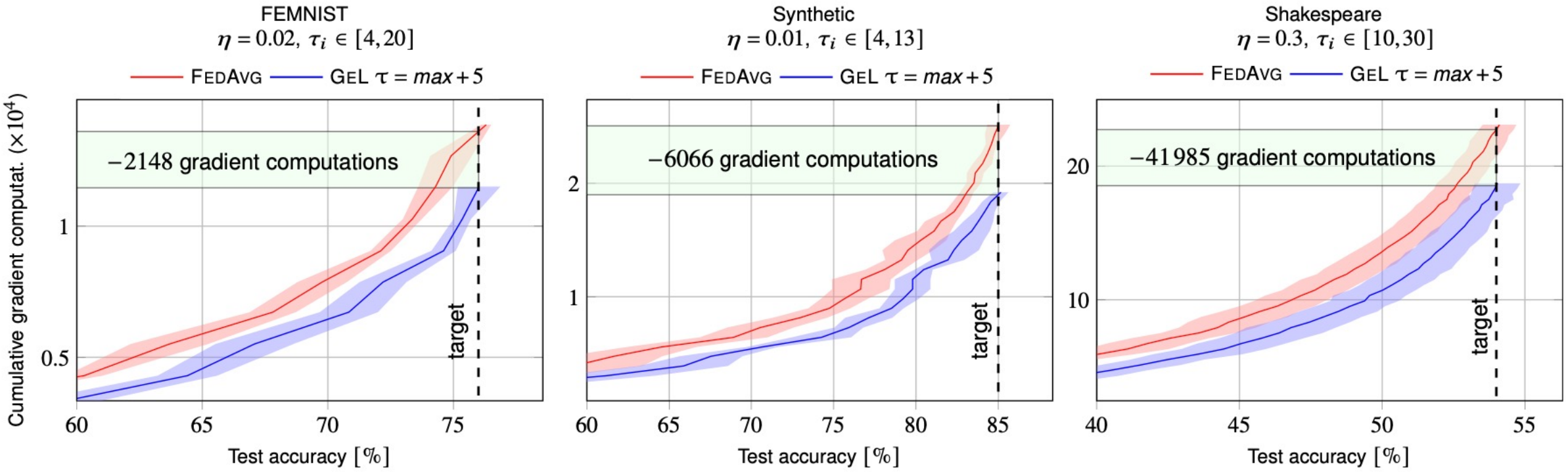


- 20 clients selected at each round(From 600 to 4000 clients total)

Quicker


FEMNIST
$\eta = 0.02,\ \tau_i \in [4,20]$
FedAvg
GeL $\tau = max + 5$
Test accuracy [%]
target
−9 rounds
Communication round
Synthetic
$\eta = 0.01,\ \tau_i \in [4,13]$
FedAvg
GeL $\tau = max + 5$
target
−36 rounds
Communication round
Shakespeare
$\eta = 0.3,\ \tau_i \in [10,30]$
FedAvg
GeL $\tau = max + 5$
target
−105 rounds
Communication round

While saving a lot of computation



Get More for Less in Decentralized Learning

JWINS: Just what is needed sharing

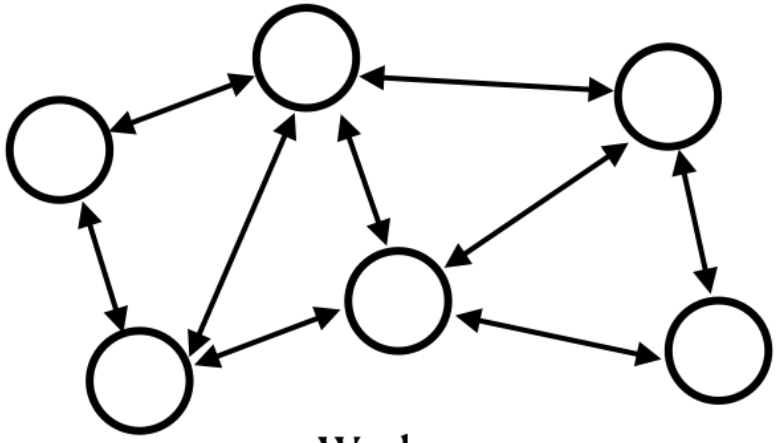
Decentralized Learning

$$(X_i)^{k+1} = \sum_{j\ in\ N} W_{ji}.x^{j,k}$$

N is the set of all Nodes

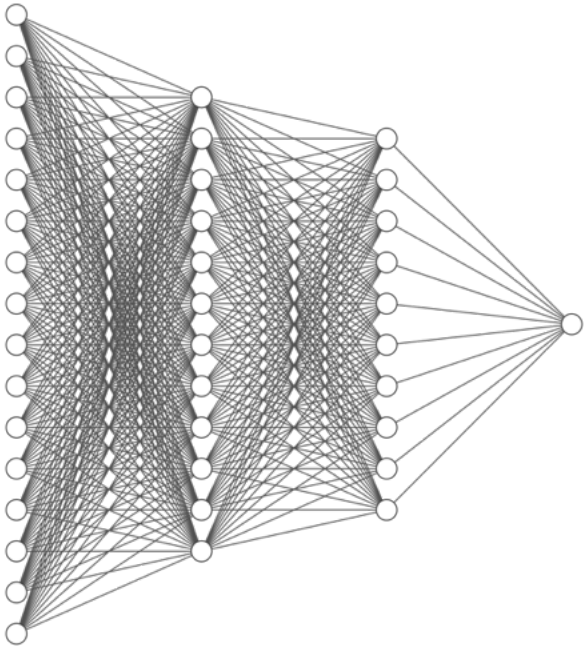
where W_{ji} is the weight of edge from j to i, $\sum_{j \in N} W_{ji} = 1$

x_j^k is the model parameters of worker j at step k

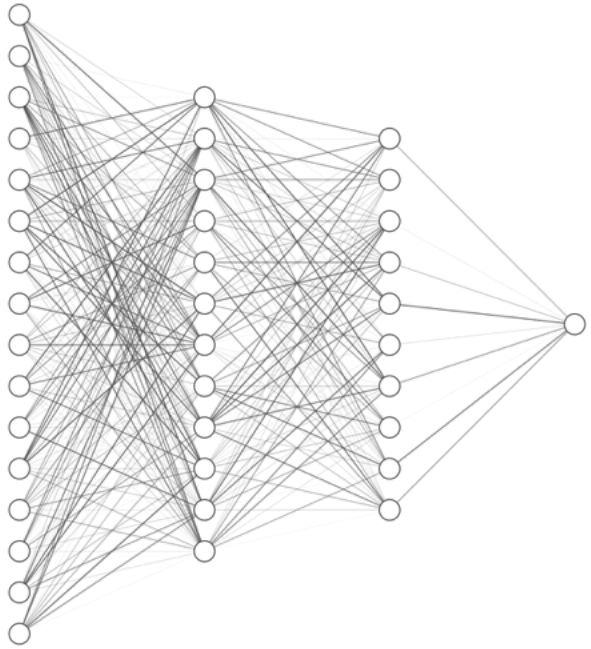


Workers

Not everything is needed at all times

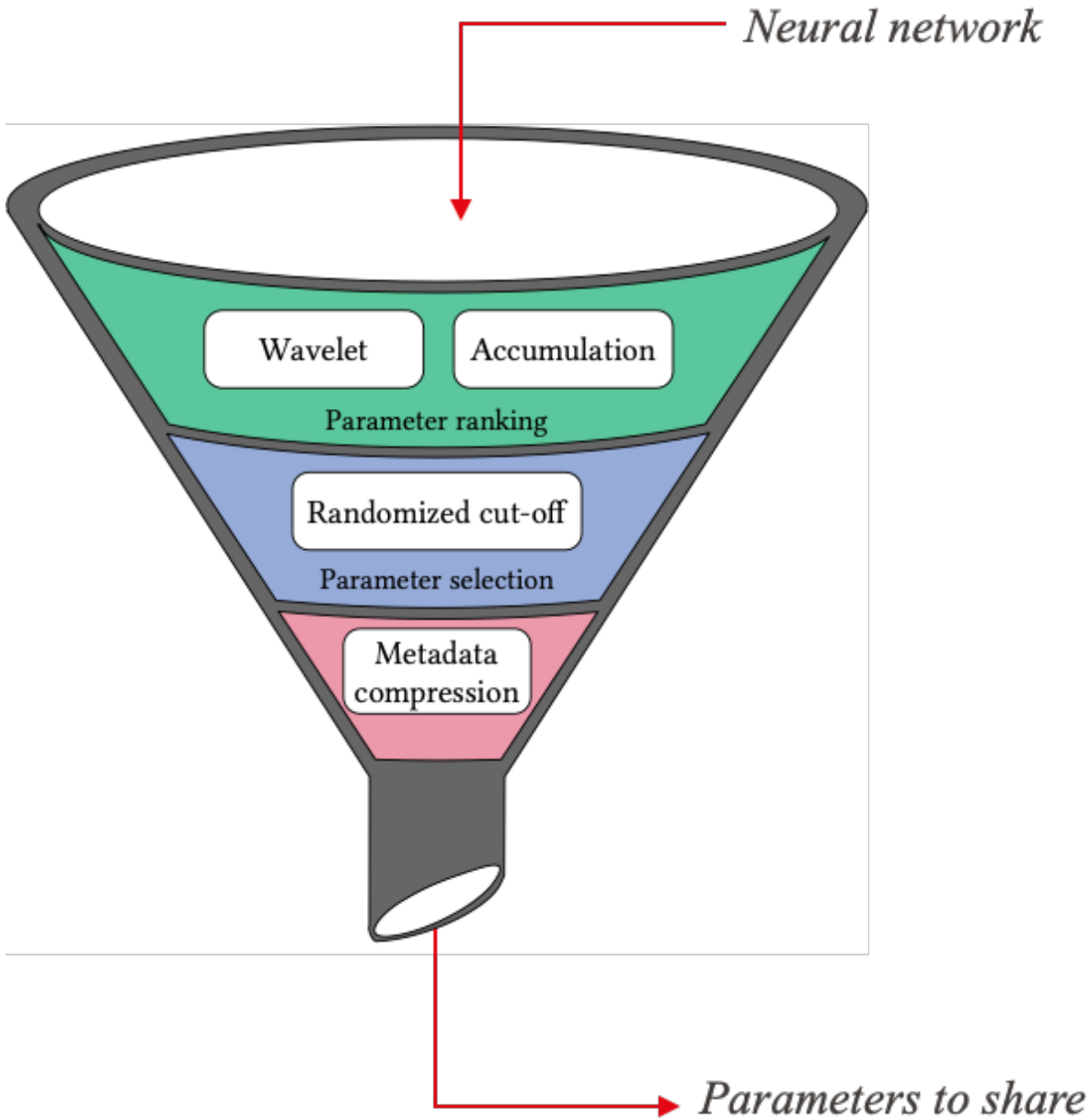


Model

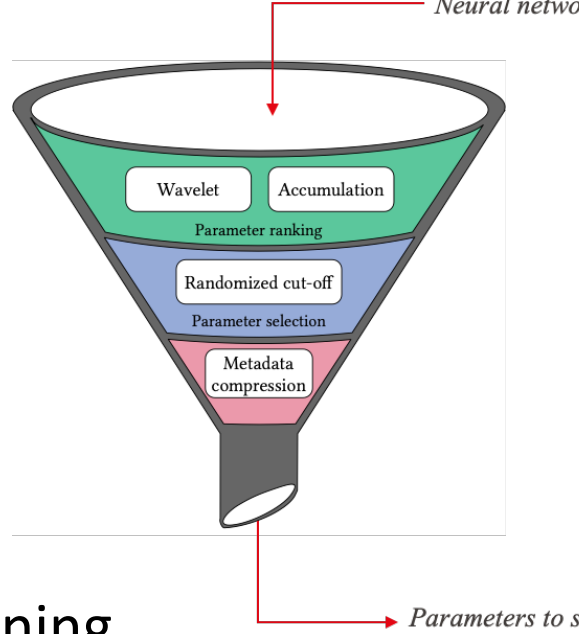


Share only the important parameters

Jwins

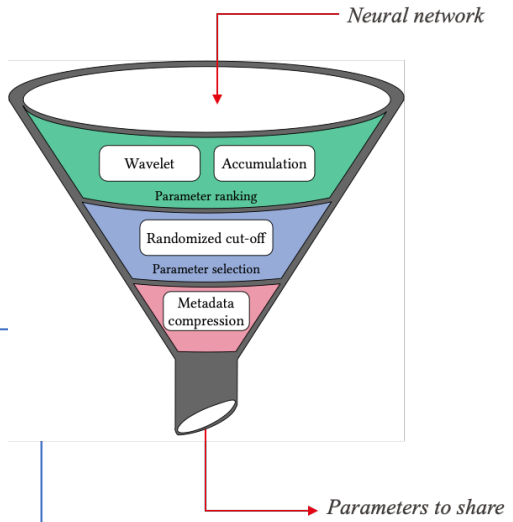

Neural network
Wavelet
Accumulation
Parameter ranking
Randomized cut-off
Parameter selection
Metadata compression
Parameters to share

Parameters Ranking



- Objective: fair and relevant ranking scheme
- Accumulation
  - Captures the importance of each parameter over the entire training
  - Reset to 0 after sharing
  - Unshared low ranked parameters retained over time
- Discrete Wavelet Transform
  - Projection of the parameters in a frequency domain
  - Importance of parameters captured by wavelet coefficients

Parameter selection

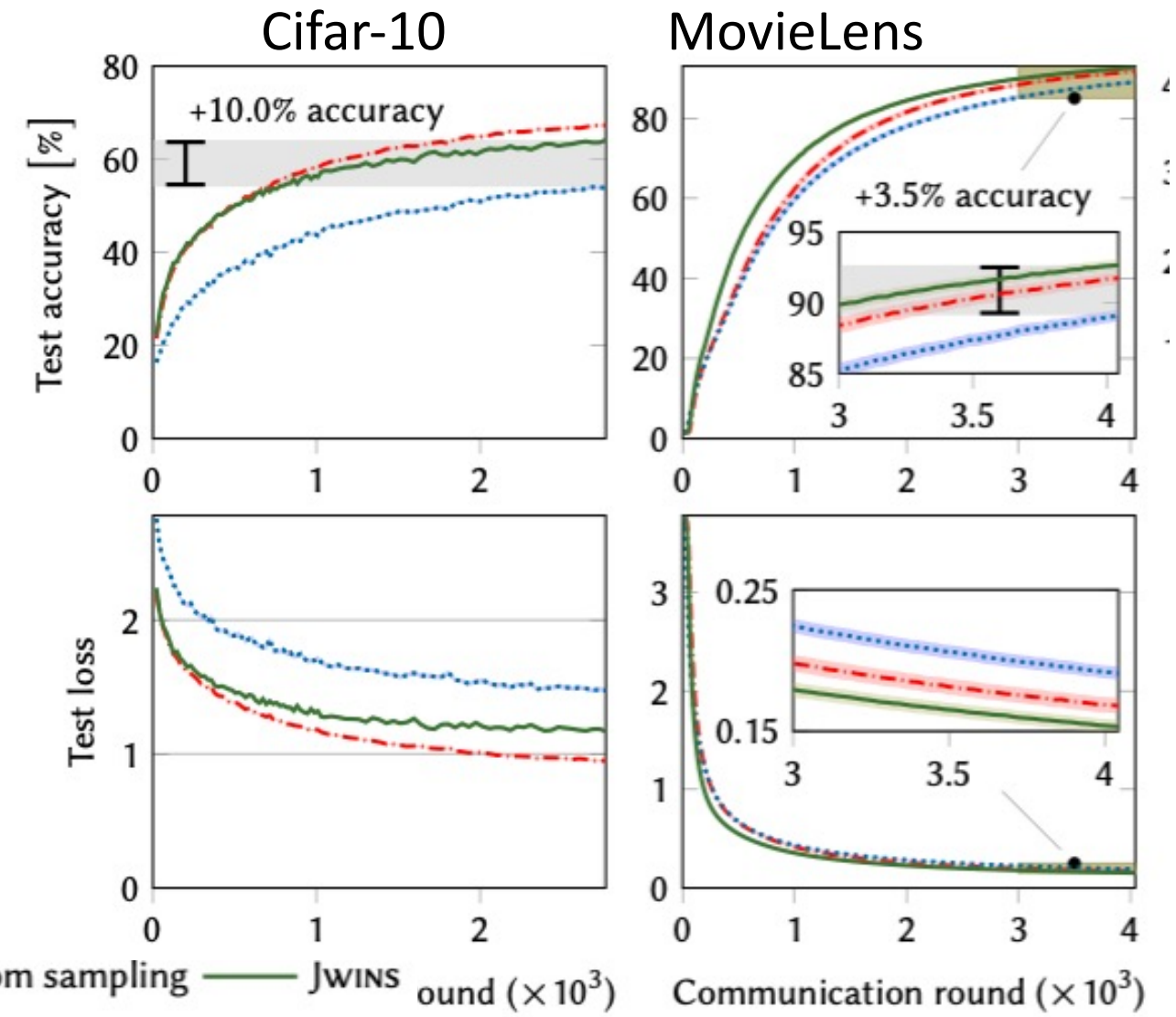


- Decides the percentage α of high-ranked parameters to share
- Avoids low-ranked parameters to never be shared by setting a low α
- Avoids congestion by setting a high α

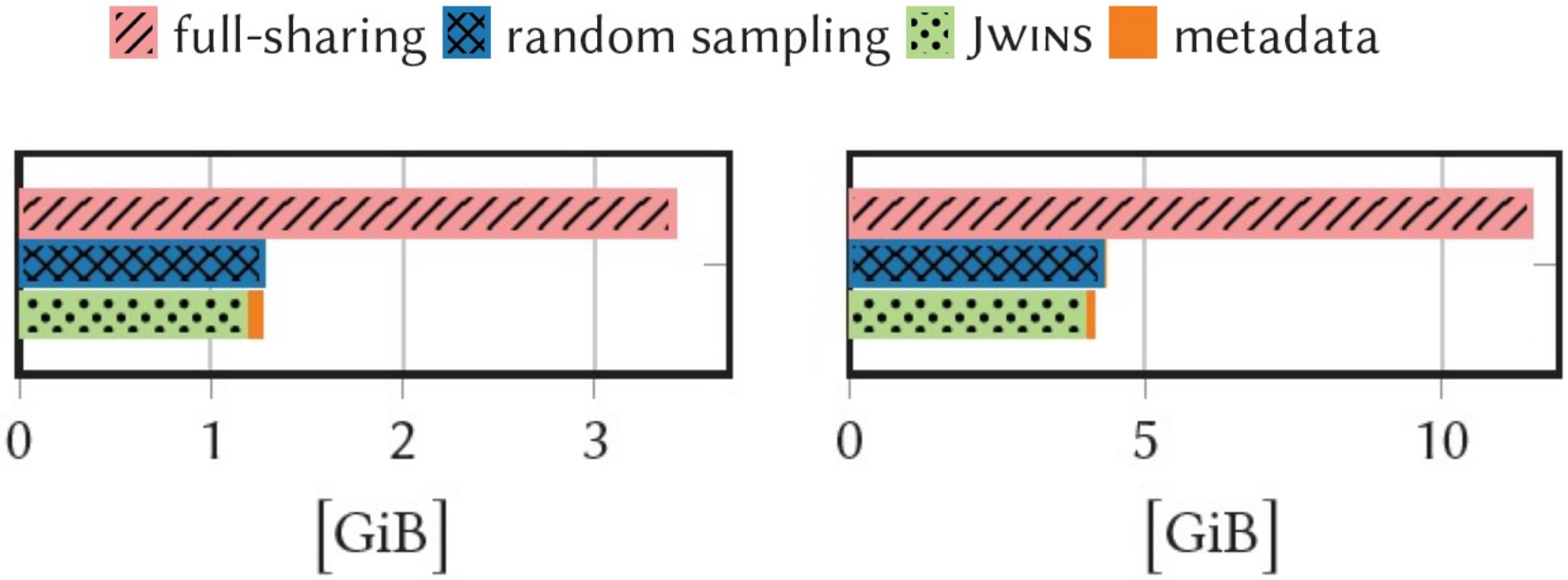
Jwins Randomized cut-off strategy
Each node picks a α uniformly at random among {10%, 15%, 20%, 25%, 30%, 40%, 100%}

Jwins achieves accuracy

96 nodes,
4-regular random graph



Saving bandwidth along the way



Cifar-10

MovieLens

Frugal decentralized learning

- Gel compensates for heterogeneous system capabilities in FL
- Jwins ensuring efficient ML by limiting communications in DL
- Not everything is needed in ML

Thank you