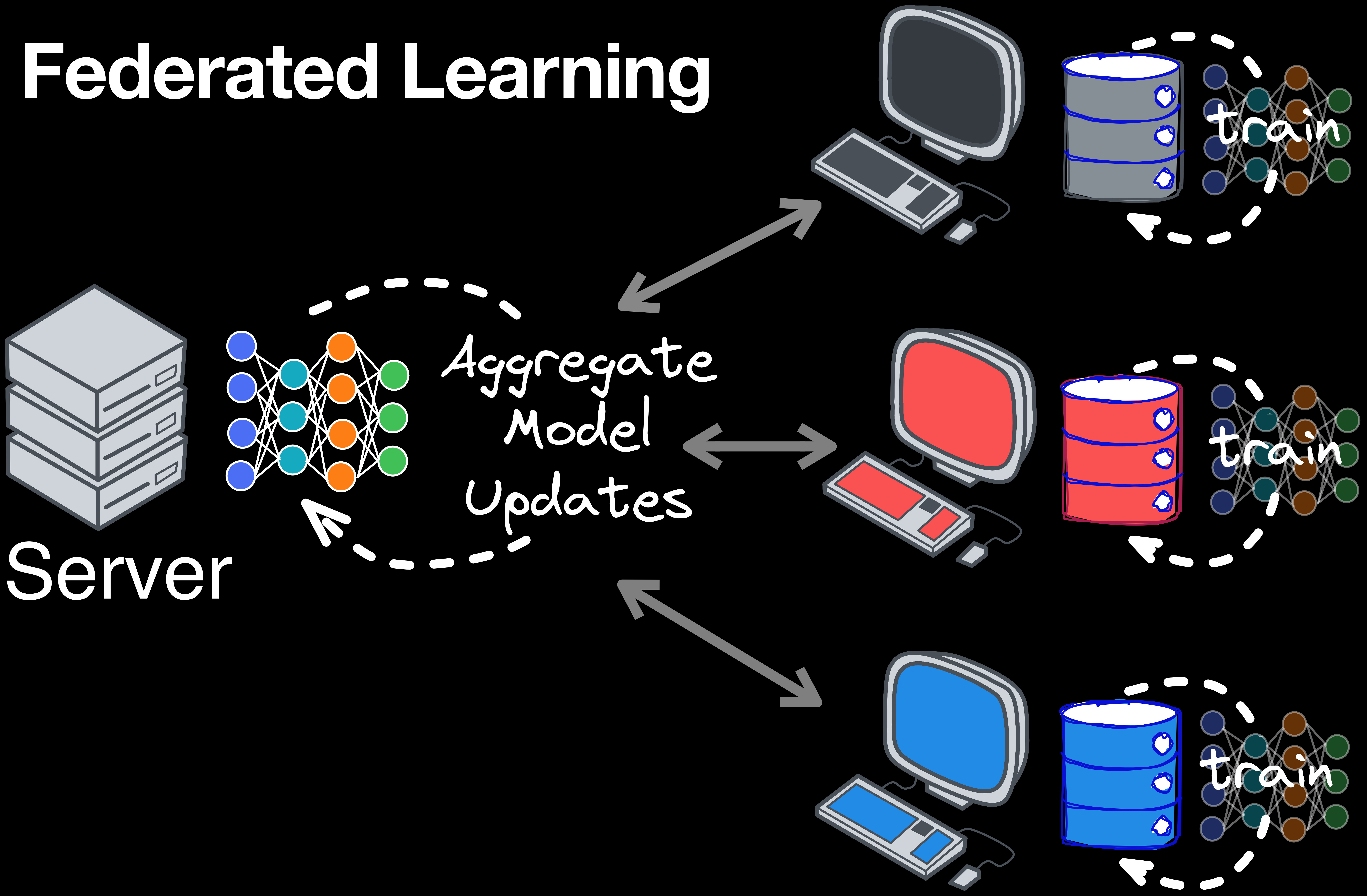
Asynchronous Federated Unlearning

Ningxin Su, Baochun Li

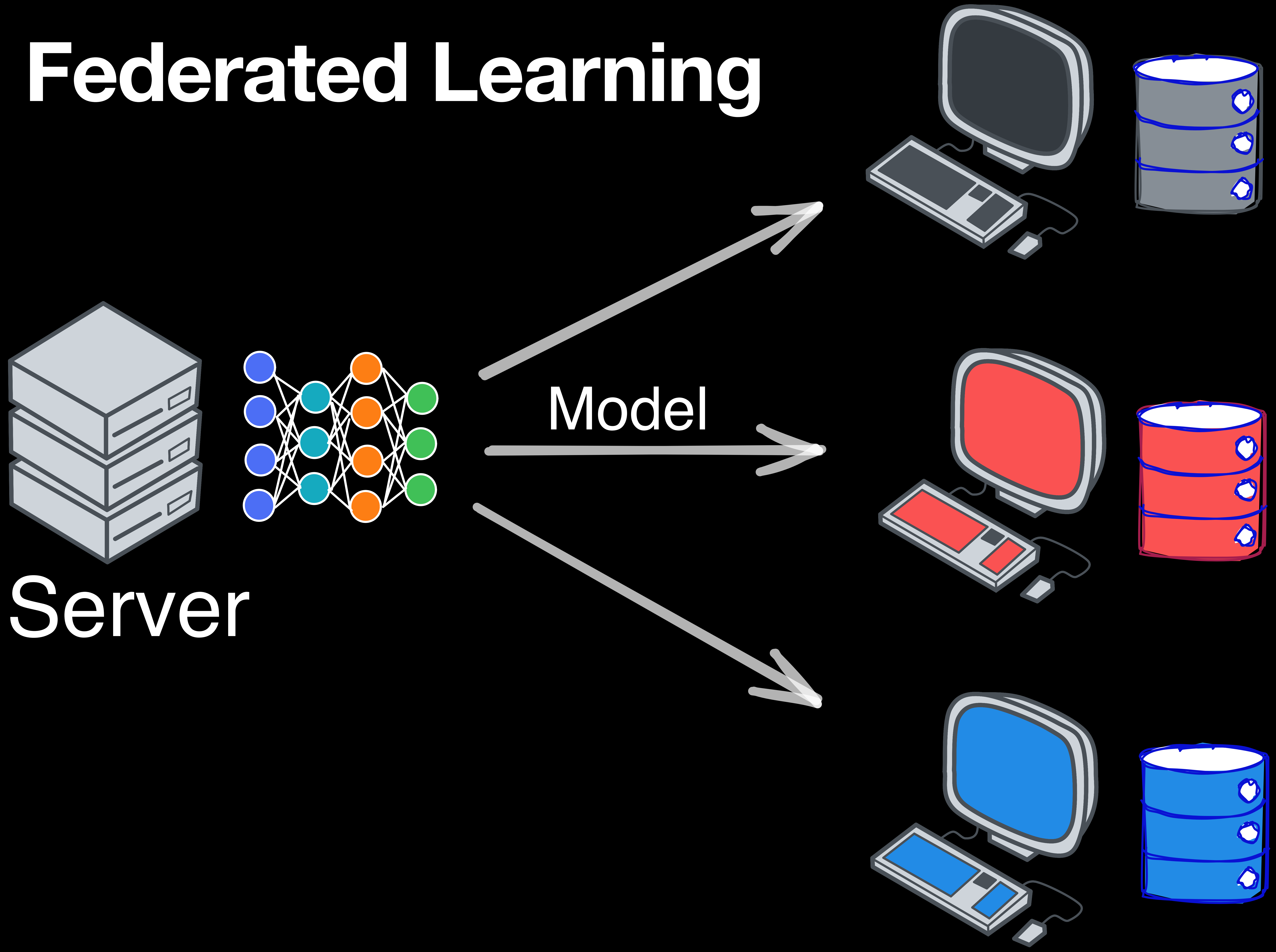
Department of Electrical and Computer Engineering

University of Toronto

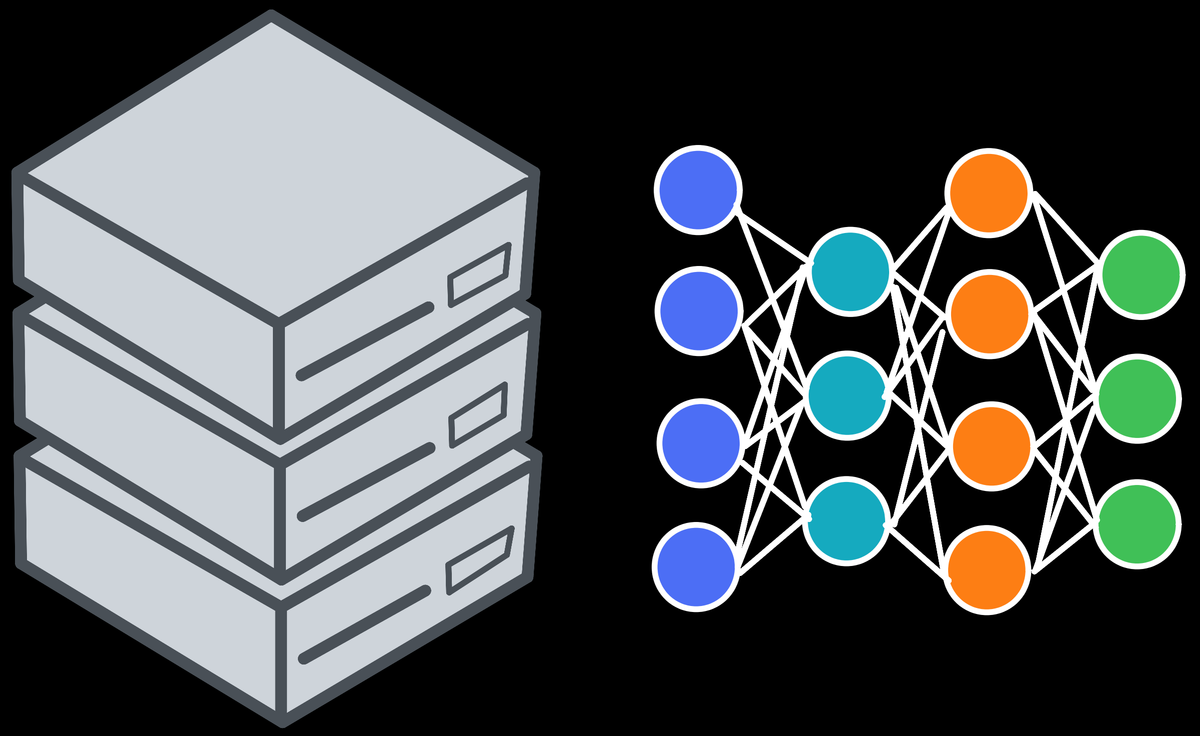
Federated Learning



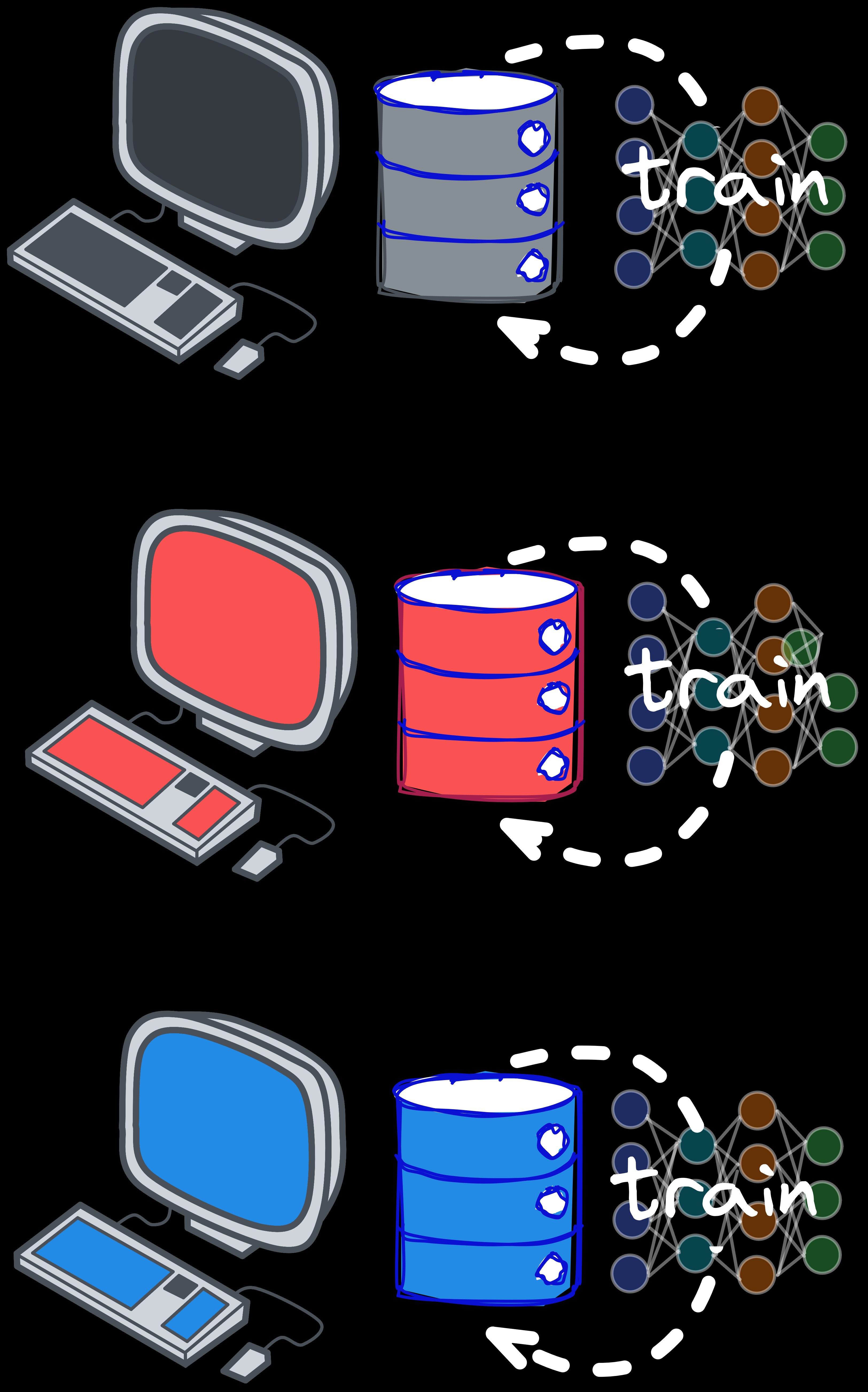
Federated Learning



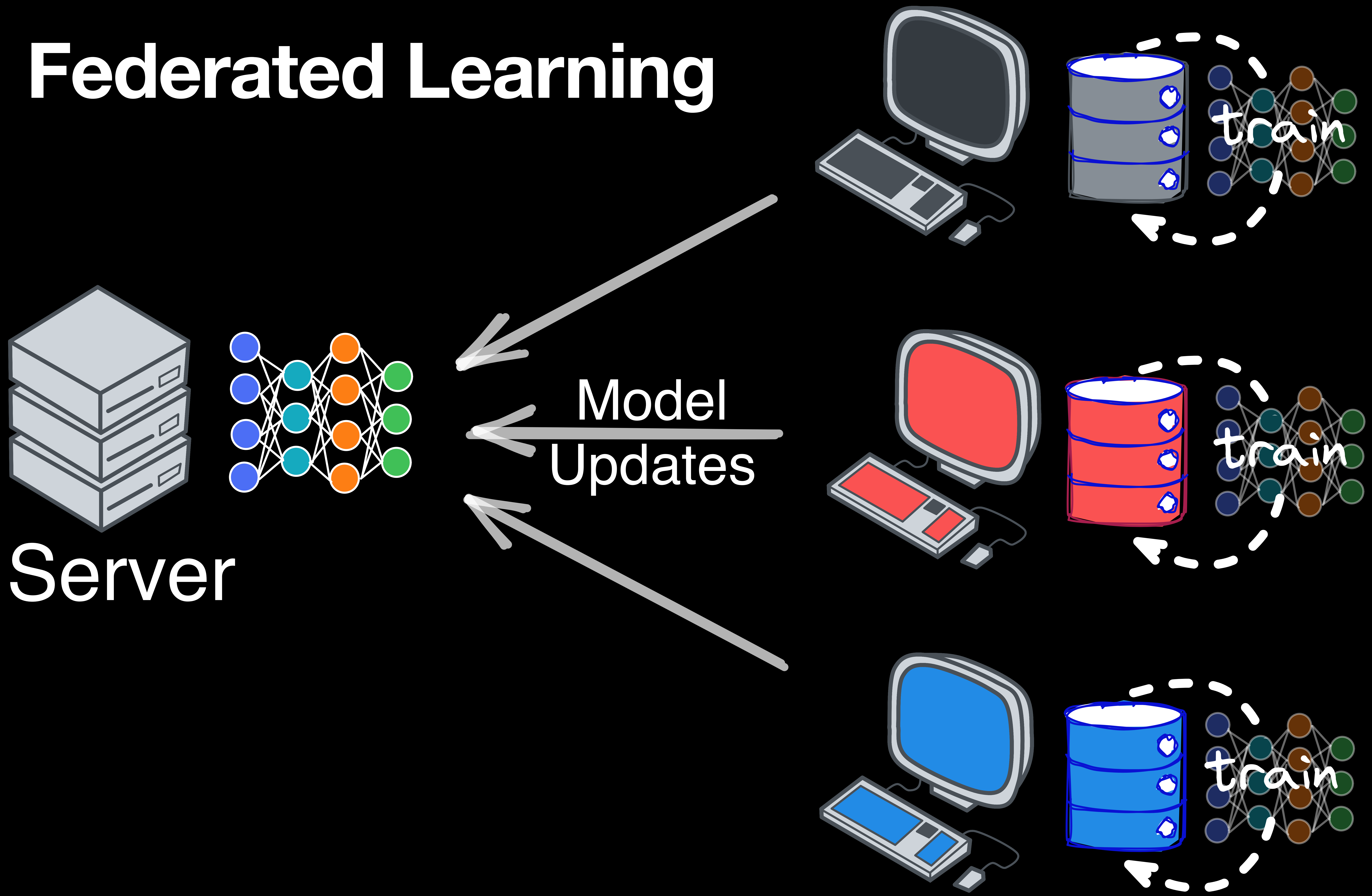
Federated Learning



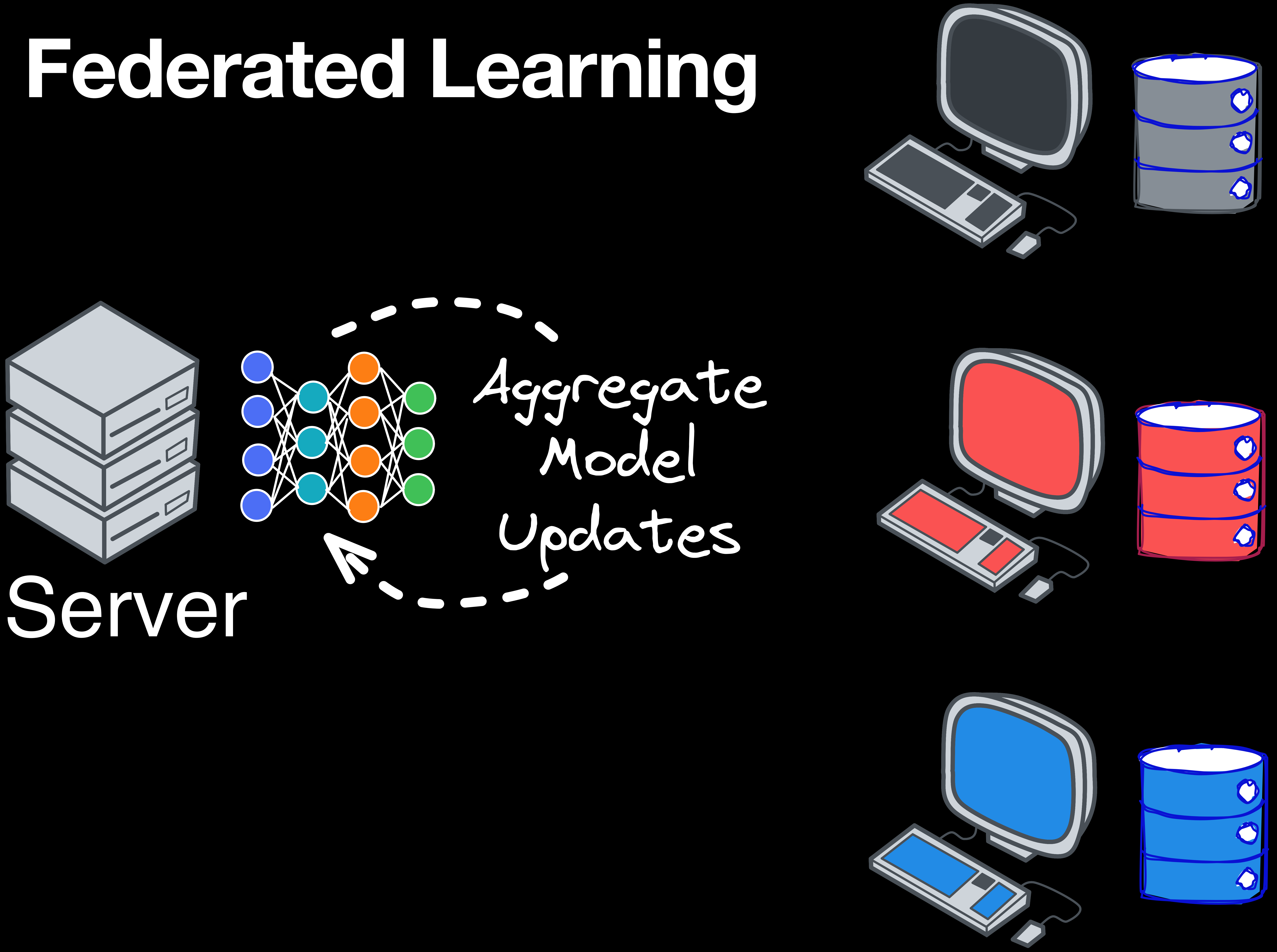
Server



Federated Learning

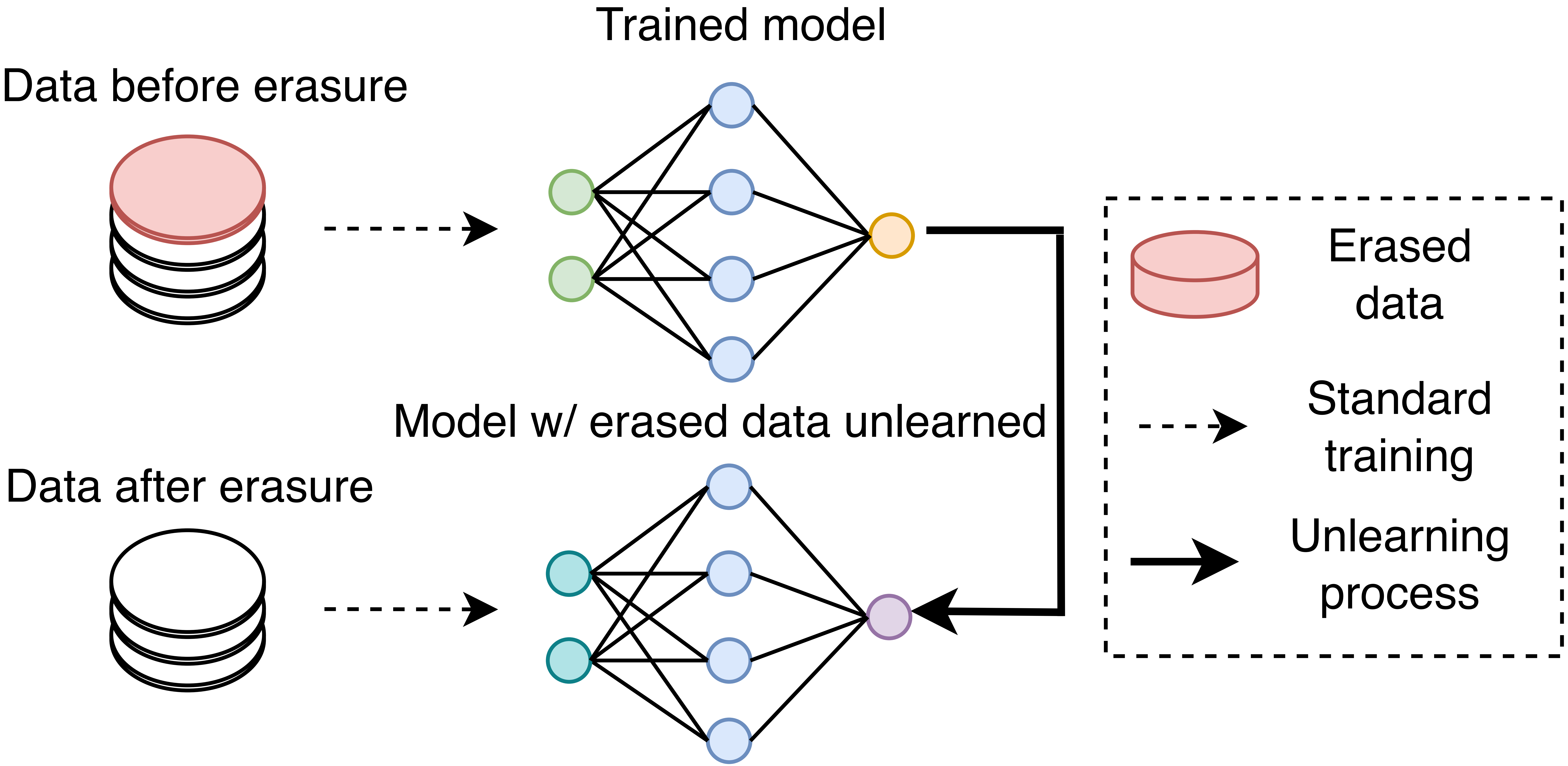


Federated Learning



Policy and regulation constraints such as GDPR required users to have “the right to erasure” — to erase effects of private data from a trained model

Machine Unlearning


Trained model
Data before erasure
Erased
data
Standard
training
Model w/ erased data unlearned
Data after erasure
Unlearning
process

Federated

~~Machine~~ Unlearning

Federated

~~Machine~~ Unlearning

Related Work

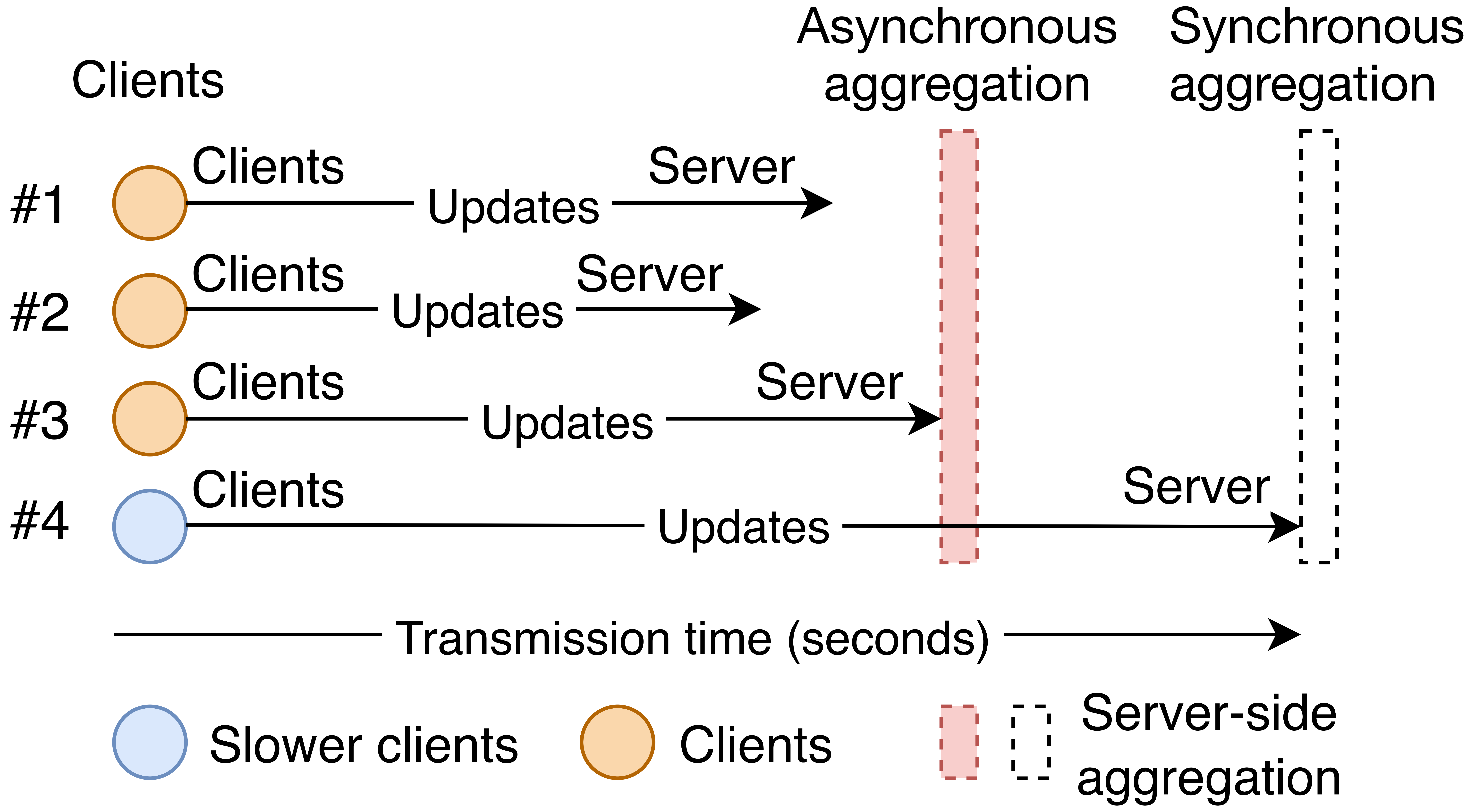
FedEraser: an approximation algorithm designed to speed up the unlearning process, but difficult to evaluate its effectiveness

G. Liu, *et al.* "**FedEraser**: Enabling Efficient Client-Level Data Removal from Federated Learning Models," *IEEE/ACM 29th International Symposium on Quality of Service (IWQoS)*, 2021.

INFOCOM'22: second-order optimization to speed up the unlearning process when retraining from scratch

Y. Liu, *et al.* "The Right to be Forgotten in Federated Learning: An Efficient Realization with Rapid Retraining," IEEE INFOCOM 2022.

Asynchronous
federated learning


Clients
Asynchronous aggregation
Synchronous aggregation
#1
Clients
Updates
Server
#2
Clients
Updates
Server
#3
Clients
Updates
Server
#4
Clients
Updates
Server
Transmission time (seconds)
Slower clients
Clients
Server-side aggregation

Asynchronous federate learning is faster!

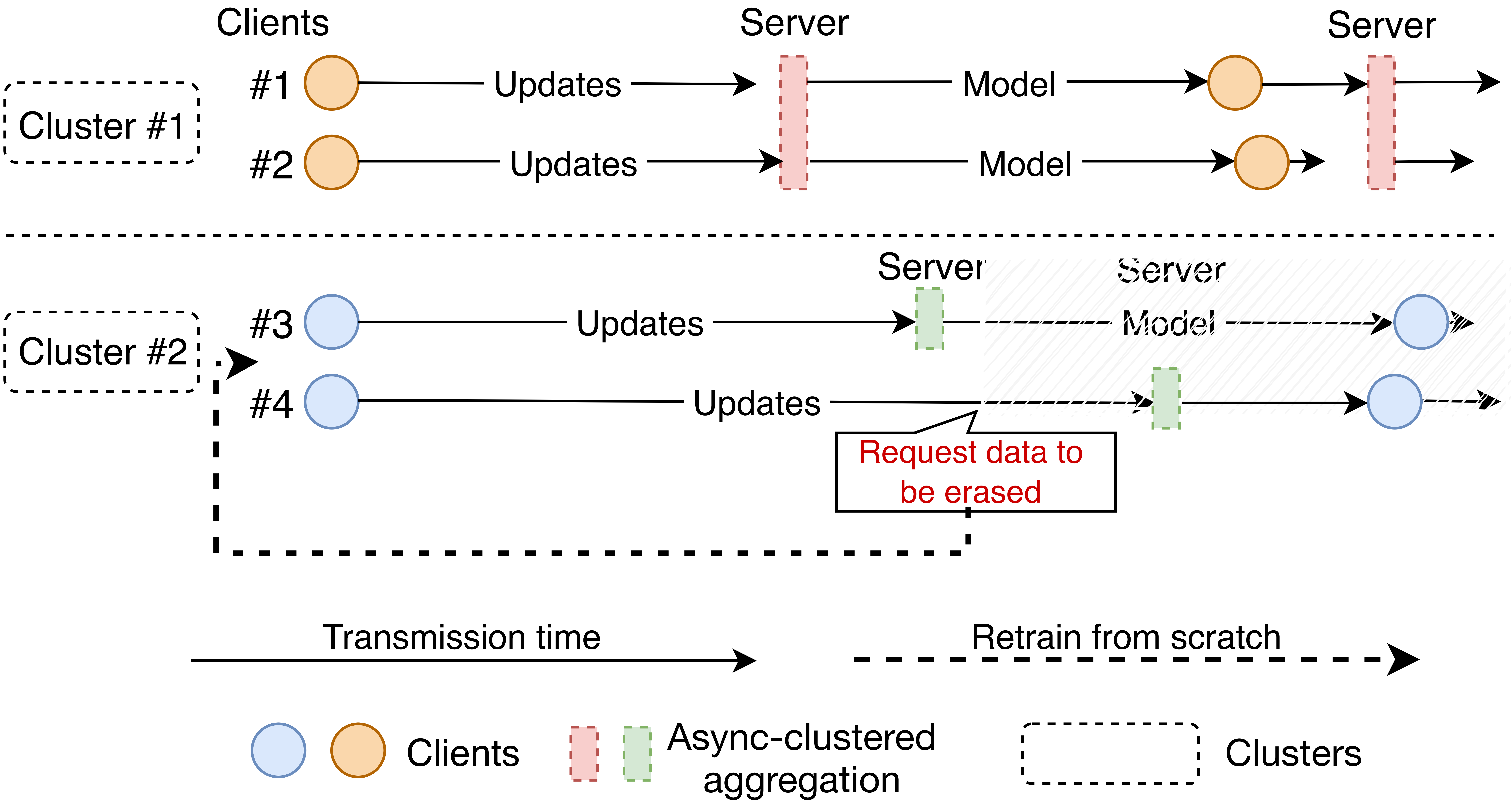
Can we combine asynchrony and unlearning?

Can we combine asynchrony and unlearning?

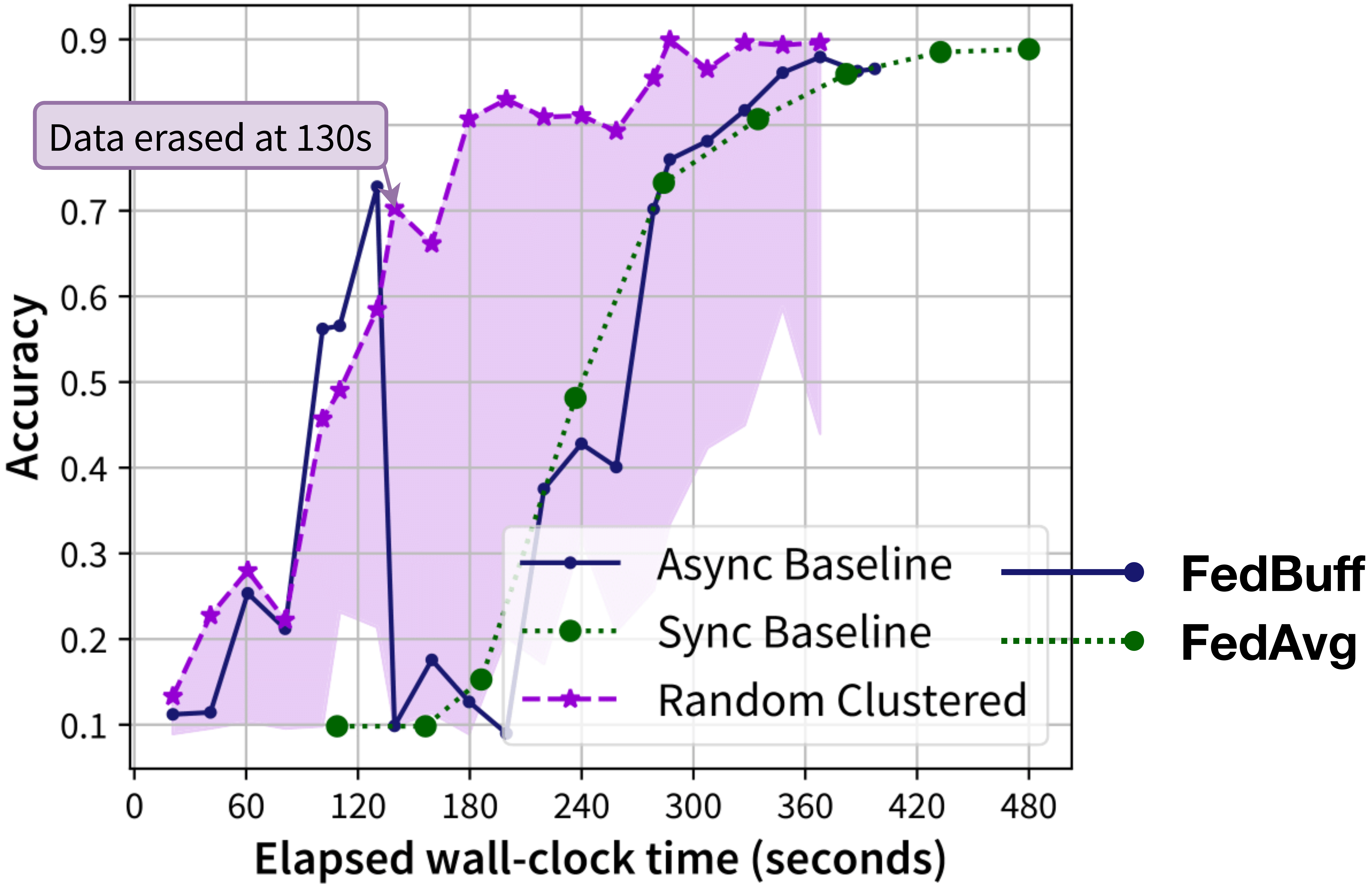
We can, and with benefits!

Knot

Combining asynchrony and unlearning



Clients
Server
Server
Cluster #1
#1
Updates
Model
#2
Updates
Model
Server
Server
Cluster #2
#3
Updates
Model
#4
Updates
Request data to be erased
Transmission time
Retrain from scratch
Clients
Async-clustered aggregation
Clusters



J. Nguyen, *et al.*, "Federated Learning with Buffered Asynchronous Aggregation," *Int'l Conference on Machine Learning (ICML)*, 2021.

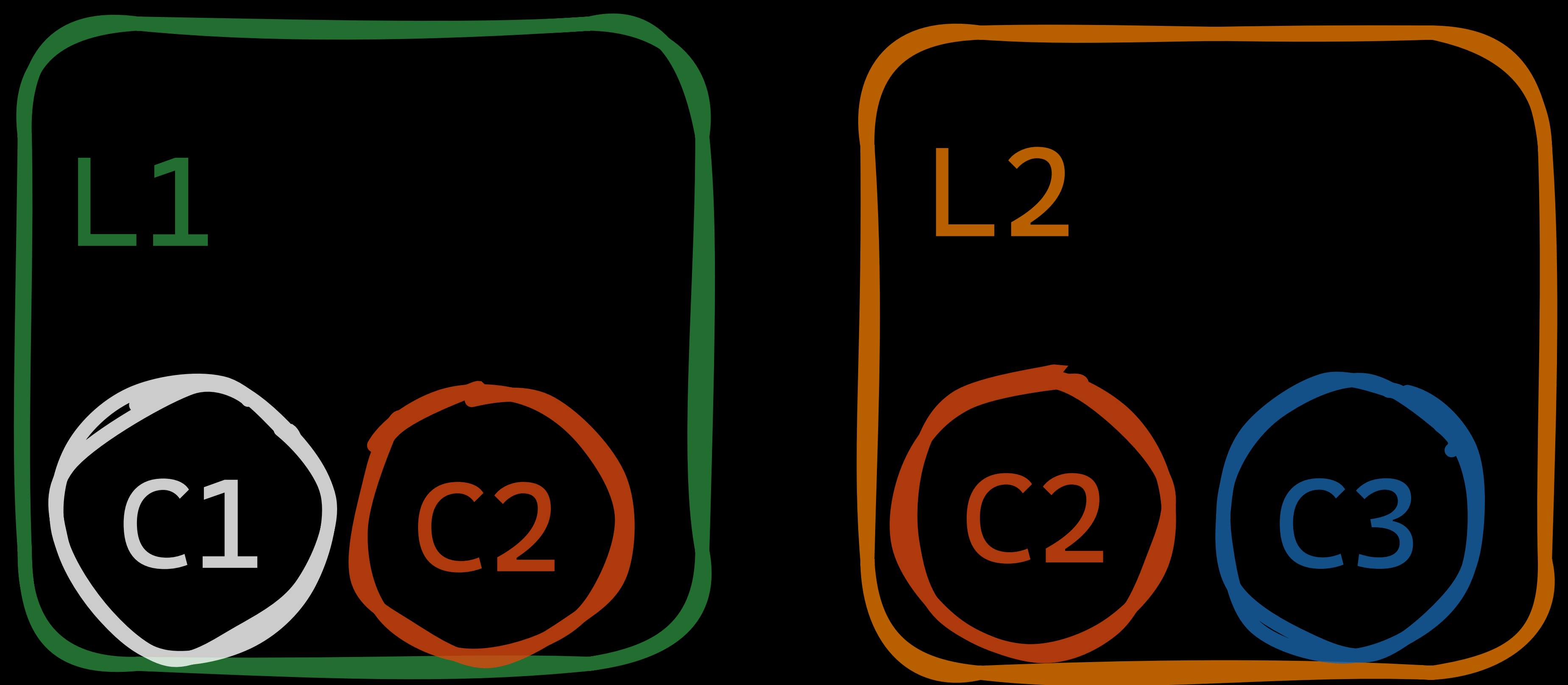
Intuitively, clustering helps reduce the risk of rolling back to retrain

Knot

Problem Formulation

Optimizing Client-Cluster Assignment

Assigning clients C_k into clusters L_n



$$d_{kn} = \| \, [a(\widetilde{T}_n - T_k), b(\widetilde{S}_n - S_k)] \, \|_2$$

training time

model disparity

Formulating the Optimization Problem

Formulating the Optimization Problem

$$\operatorname*{lexmin}_{x} f = (d_{11}x_{11}, \ldots, d_{kn}x_{kn}, \ldots, d_{KN}x_{KN})$$

Formulating the Optimization Problem

$$\operatorname*{lexmin}_{x} f = (d_{11}x_{11}, \ldots, d_{kn}\boxed{x_{kn}}, \ldots, d_{KN}x_{KN})$$

$$x \in \{0,1\}^{KN}, x = (x_{11}, \ldots, x_{KN})$$

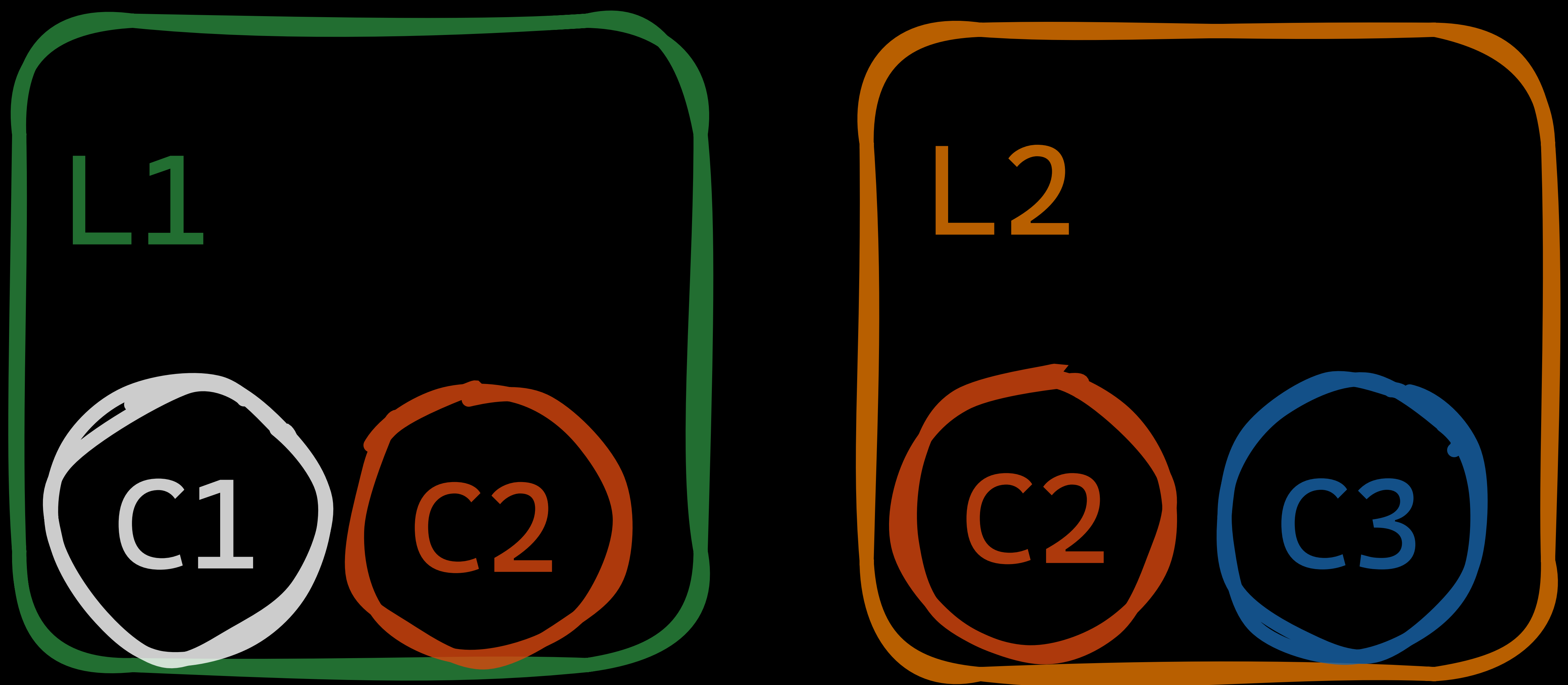
Formulating the Optimization Problem

$$\operatorname*{lexmin}_{x} f = (d_{11}x_{11}, \ldots, d_{kn}\boxed{x_{kn}}, \ldots, d_{KN}x_{KN})$$

$$x \in \{0,1\}^{KN}, x = (x_{11}, \ldots, x_{KN})$$

$x_{11} = 1$

$x_{12} = 0$



$x_{21} = 1$

$x_{22} = 1$

$$\operatorname*{lexmin}_{x} f = (d_{11}x_{11}, \ldots, d_{kn}x_{kn}, \ldots, d_{KN}x_{KN})$$

$$\text{s.t.} \quad \sum_{n=1}^{N} x_{kn} \leq c_1, \forall k \in \mathcal{K} \quad (2)$$

$$\sum_{n=1}^{N} x_{kn} \geq 1, \ \forall k \in \mathcal{K} \quad (3)$$

$$\sum_{k=1}^{K} x_{kn} \geq c_2, \ \forall n \in \mathcal{N} \ (4)$$

$$\sum_{k=1}^{K} x_{kn} \leq c_3, \ \forall n \in \mathcal{N} \ (5)$$

$$\operatorname*{lexmin}_{x} f = (d_{11}x_{11}, \ldots, d_{kn}x_{kn}, \ldots, d_{KN}x_{KN})$$

$$\text{s.t.} \quad \sum_{n=1}^{N} x_{kn} \leq \boxed{c_1}, \forall k \in \mathcal{K} \quad (2)$$

# of clusters a client belongs to

$$\sum_{n=1}^{N} x_{kn} \geq \boxed{1}, \forall k \in \mathcal{K} \quad (3)$$

$$\sum_{k=1}^{K} x_{kn} \geq c_2, \ \forall n \in \mathcal{N} \quad (4)$$

$$\sum_{k=1}^{K} x_{kn} \leq c_3, \ \forall n \in \mathcal{N} \quad (5)$$

$$\operatorname*{lexmin}_{x} f = (d_{11}x_{11}, \ldots, d_{kn}x_{kn}, \ldots, d_{KN}x_{KN})$$

$$\text{s.t.} \quad \sum_{n=1}^{N} x_{kn} \leq c_1, \forall k \in \mathcal{K} \quad (2)$$

$$\sum_{n=1}^{N} x_{kn} \geq 1, \forall k \in \mathcal{K} \quad (3)$$

# of clusters a client belongs to

$$\sum_{k=1}^{K} x_{kn} \geq c_2, \forall n \in \mathcal{N} \quad (4)$$

$$\sum_{k=1}^{K} x_{kn} \leq c_3, \forall n \in \mathcal{N} \quad (5)$$

# of clients a cluster can have

Transforming into an LP Problem

R. R. Meyer, “A Class of Nonlinear Integer Programs Solvable by a Single Linear Program,”
SIAM Journal on Control and Optimization, vol. 15, no. 6, pp. 935–946, 1977.

Transforming into an LP Problem

$$\operatorname*{lexmin}_{x} f = (d_{11}x_{11}, \ldots, d_{kn}x_{kn}, \ldots, d_{KN}x_{KN})$$

R. R. Meyer, “A Class of Nonlinear Integer Programs Solvable by a Single Linear Program,”
SIAM Journal on Control and Optimization, vol. 15, no. 6, pp. 935–946, 1977.

Transforming into an LP Problem

$$\operatorname*{lexmin}_{x} f = (d_{11}x_{11}, \ldots, d_{kn}x_{kn}, \ldots, d_{KN}x_{KN})$$

Separable convex objective

$$\min_x \sum_{k \in \mathcal{K}} \sum_{n \in \mathcal{N}} (KN)^{D_{kn} x_{kn}}$$

R. R. Meyer, "A Class of Nonlinear Integer Programs Solvable by a Single Linear Program," *SIAM Journal on Control and Optimization*, vol. 15, no. 6, pp. 935–946, 1977.

Transforming into an LP Problem

$$\operatorname*{lexmin}_{x} f = (d_{11}x_{11}, \ldots, d_{kn}x_{kn}, \ldots, d_{KN}x_{KN})$$

Separable convex objective

$$\min_x \sum_{k\in\mathcal{K}} \sum_{n\in\mathcal{N}} (KN)^{D_{kn}x_{kn}}$$

Totally unimodular matrix

Constraints (2), (3), (4) and (5)

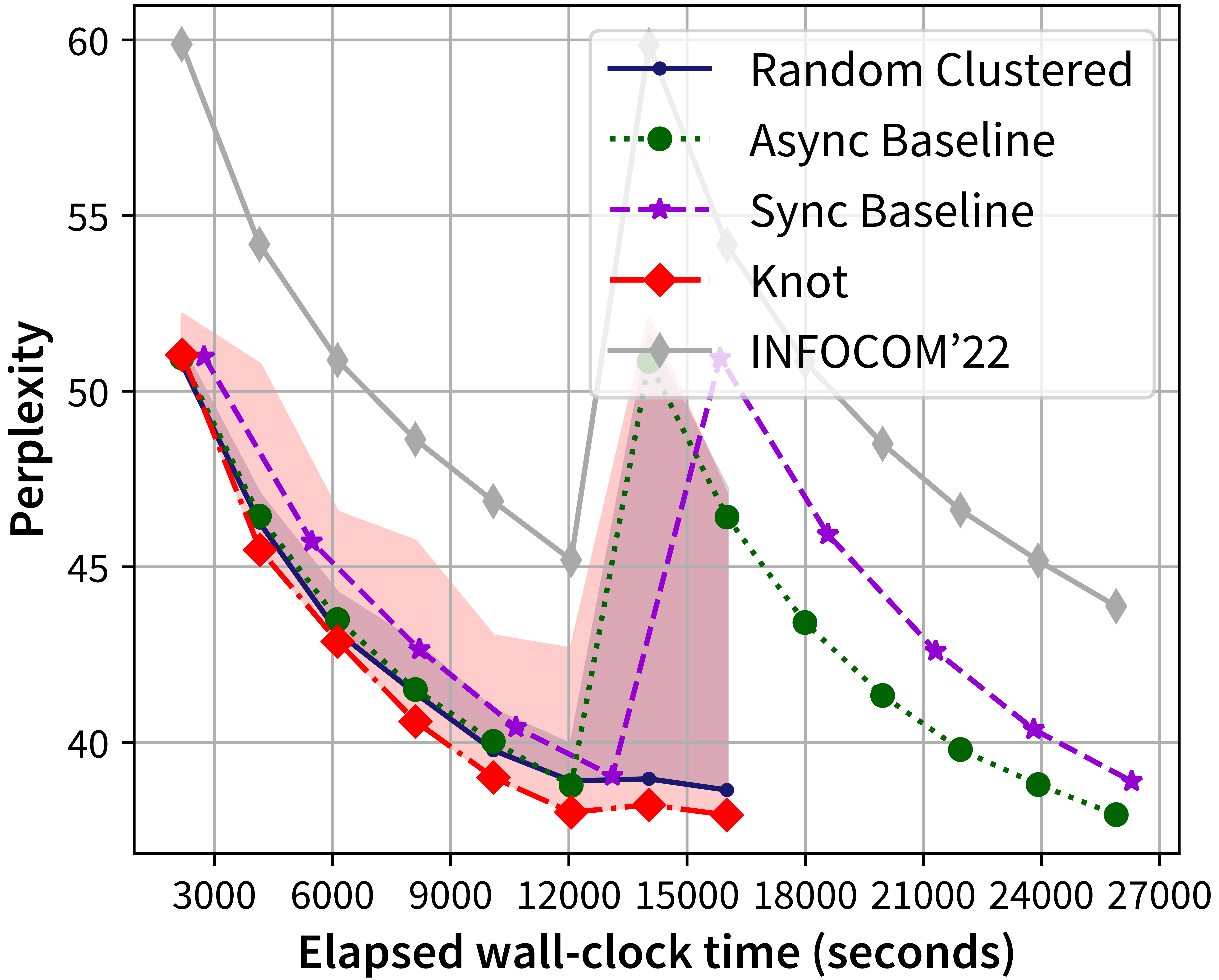
R. R. Meyer, “A Class of Nonlinear Integer Programs Solvable by a Single Linear Program,” *SIAM Journal on Control and Optimization*, vol. 15, no. 6, pp. 935–946, 1977.

Knot

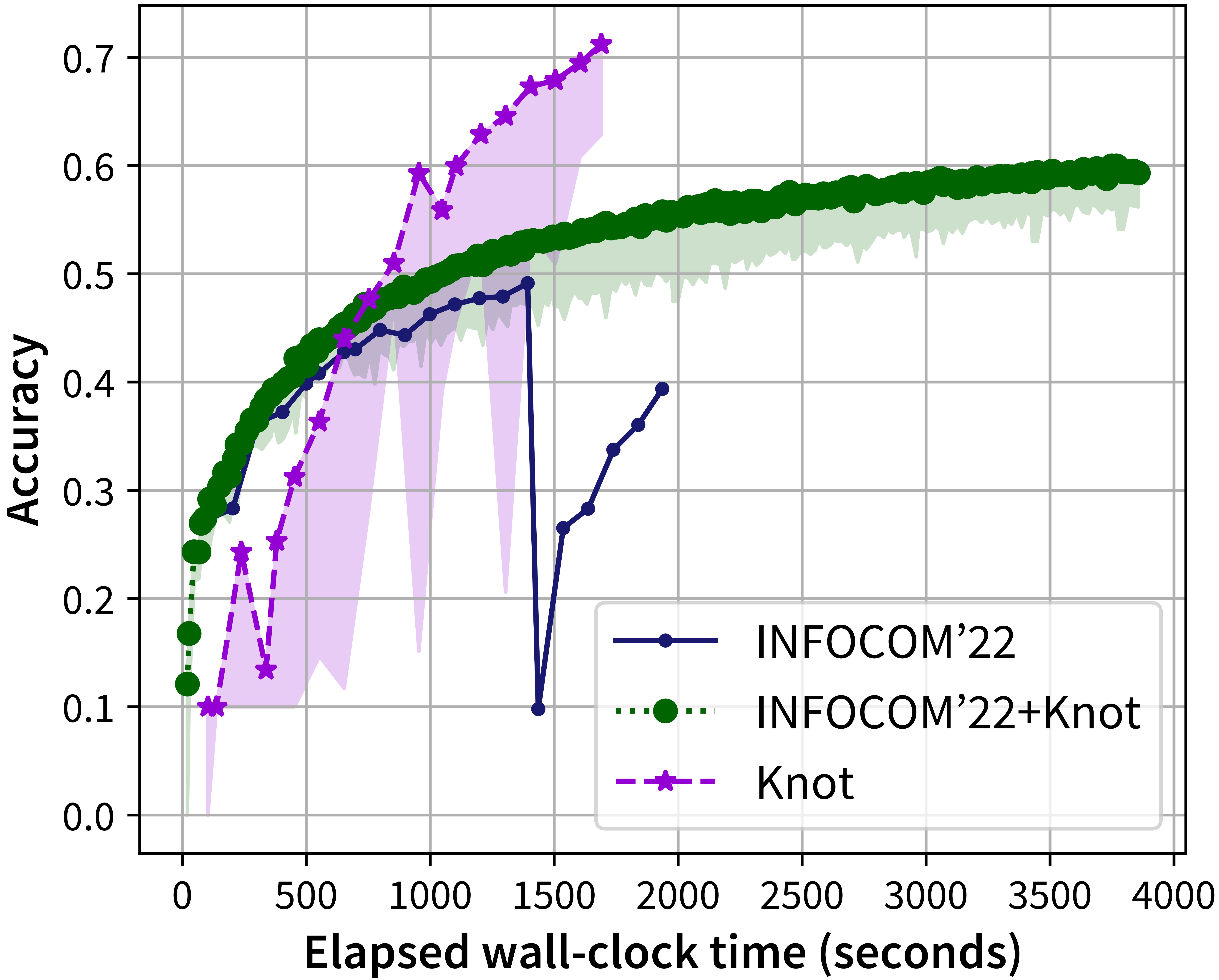
Experimental Results

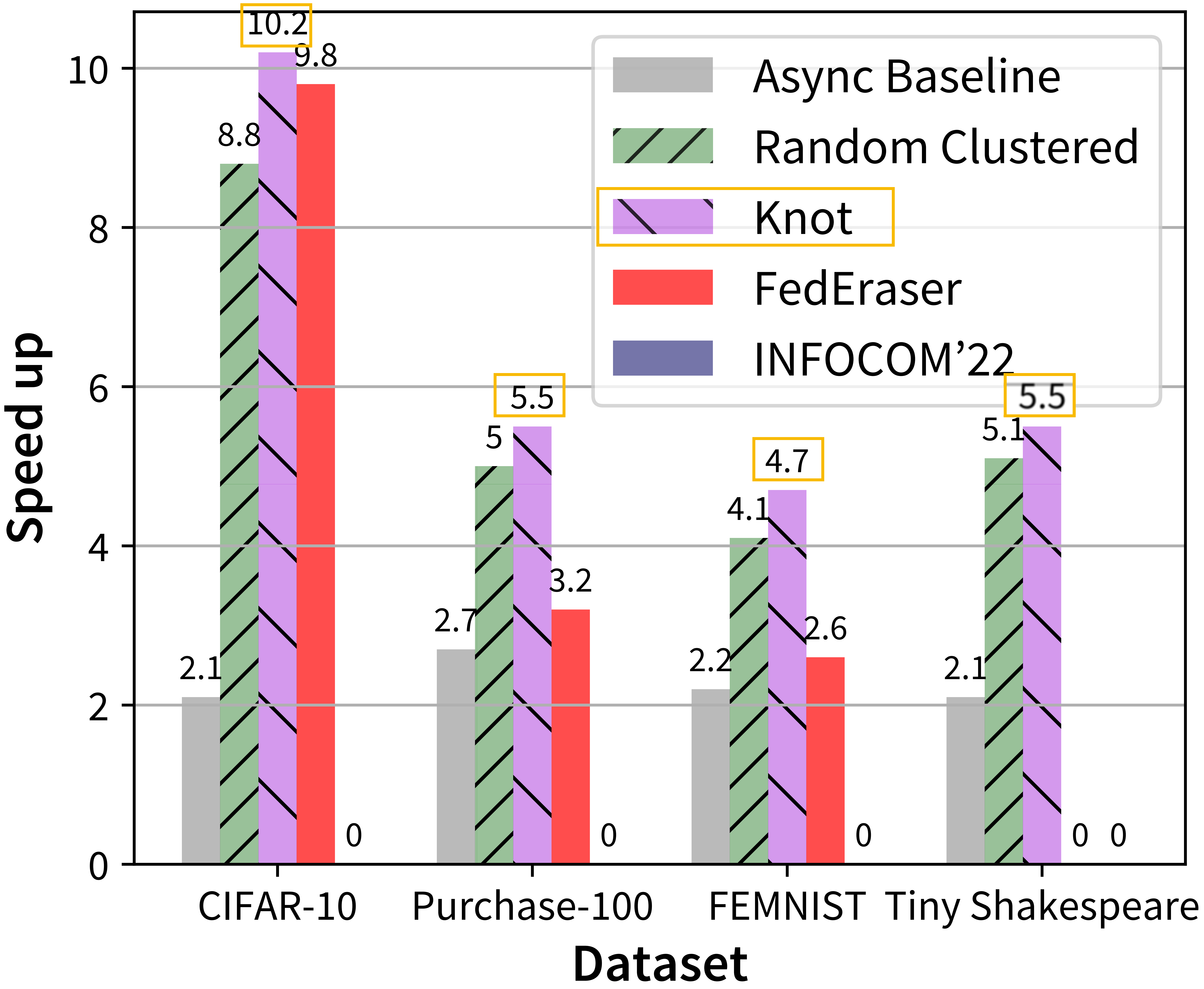
Experimental Settings

Parameter	`CIFAR-10`	`FEMNIST`	`Purchase-100`	`Tiny-Shakespeare`
K	100	250	100	70/50
# selected	20	200	60	50/30
# minimum	15	150	30	25
# erased	1/2	1/2	1/2	2/4
Samplers	non–i.i.d.	non–i.i.d.	non–i.i.d.	i.i.d.
Models	`VGG-16`	`LeNet-5`	`MLP`	`GPT-2`



Random Clustered
Async Baseline
Sync Baseline
Knot
INFOCOM'22
Perplexity
60
55
50
45
40
3000
6000
9000
12000
15000
18000
21000
24000
27000
Elapsed wall-clock time (seconds)


Accuracy
Elapsed wall-clock time (seconds)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
500
1000
1500
2000
2500
3000
3500
4000
INFOCOM’22
INFOCOM’22+Knot
Knot



Speed up
Dataset
Async Baseline
Random Clustered
Knot
FedEraser
INFOCOM'22
CIFAR-10
Purchase-100
FEMNIST
Tiny Shakespeare
0
2
4
6
8
10
2.1
8.8
10.2
9.8
0
2.7
5
5.5
3.2
0
2.2
4.1
4.7
2.6
0
2.1
5.1
5.5
0
0

Optimal clustering > random clustering > no clustering

Optimal and fast

~~Optimal~~ clustering > random

clustering > no clustering

ningxinsu.github.io