Seminar at Gachon University

Scalable Federated Learning on Real-World Edge Device Environments

Sunwoo Lee

sunwool@inha.ac.kr

Introduction

- Sunwoo Lee, Ph.D.
  - Assistant Professor
  - Department of Computer Science & Engineering
  - Inha University, 2022 ~ present

- **E**ducation & **E**xperiences
  - Assistant Professor at Inha University
  - Postdoc at University of Southern California
  - Ph.D. at Northwester University
  - M.S. & B.S. at Hanyang University
  - System Software Researcher

- **R**esearch Interests
  - Large-scale machine learning
  - Communication-efficient Federated Learning
  - Applied machine learning for electronic materials design and analysis

Outline

Research Background and Motivation

Practical Issues in Federated Learning

Our solution #1

FedLAMA: Layer-wise Adaptive Model Aggregation

Our solution #2

InclusiveFL: Scalable FL on heterogeneous edge devices

Wrap-up

FedML: an open-source software framework for FL

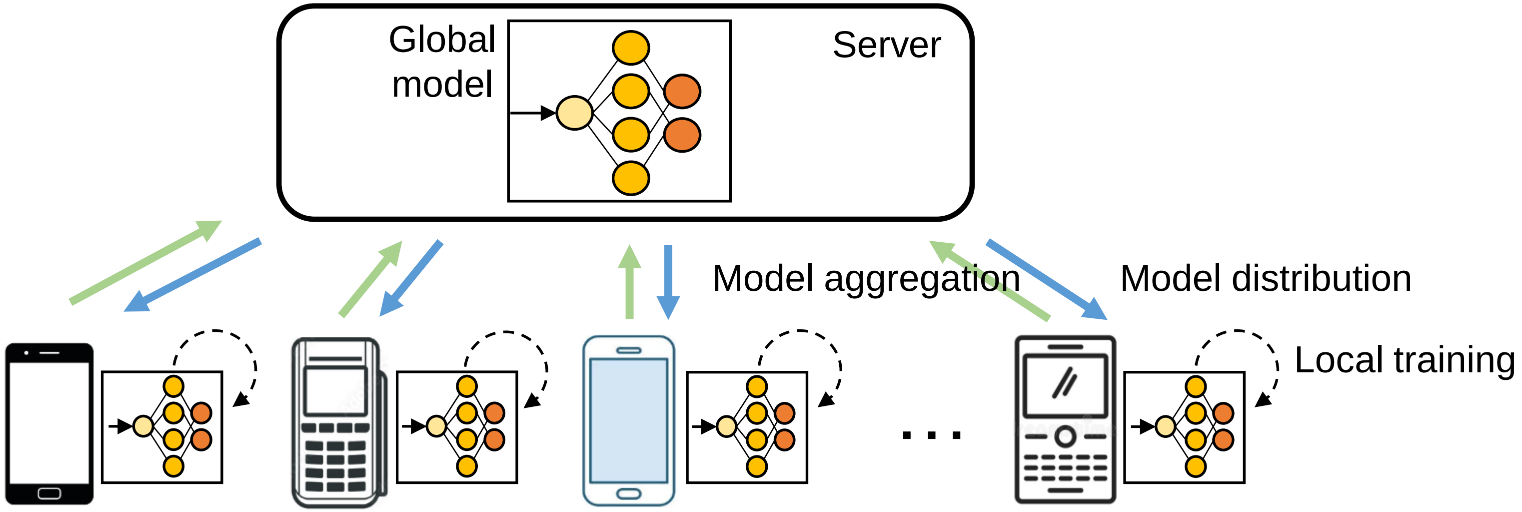
Massive Amount Data is Born at the Edge

- Data at the Edge is:
  - ***Distributed*** across many devices.
  - ***Non-sharable*** across different devices.
  - ***Heterogeneous*** across devices w.r.t. the size and the labels.

What is Federated Learning (FL)?

- A novel solution to analyzing the distributed, non-sharable, and heterogeneous datasets[1]

Main Principle: train locally & aggregate globally

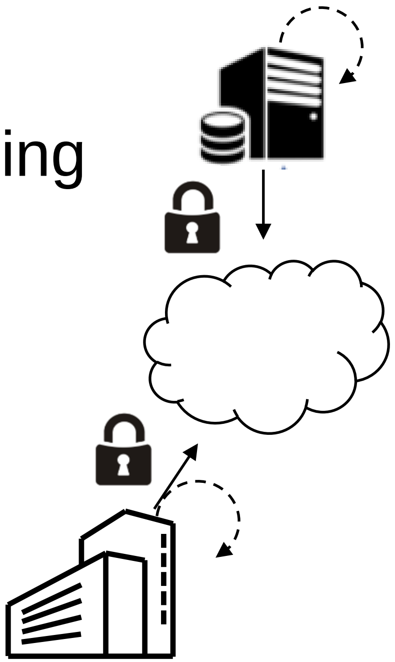


[1] McMahan et al., Communication-Efficient Learning of Deep Networks…. AISTATS 2017

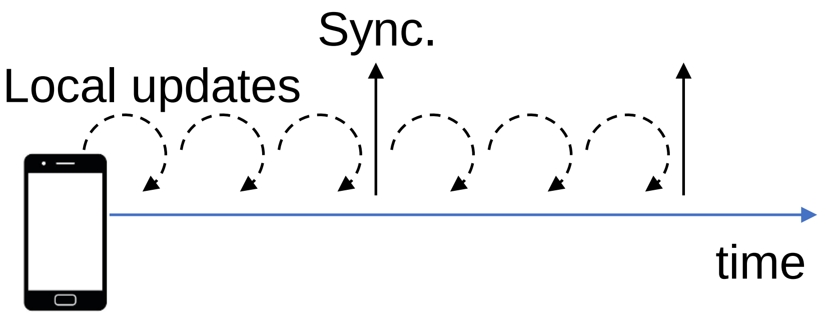
Why is FL Promising?

1. Effective Distributed Learning on Heterogeneous Datasets

2. Secure Learning



3. Communication-Efficient Distributed Learning



However, … Scalability Issues

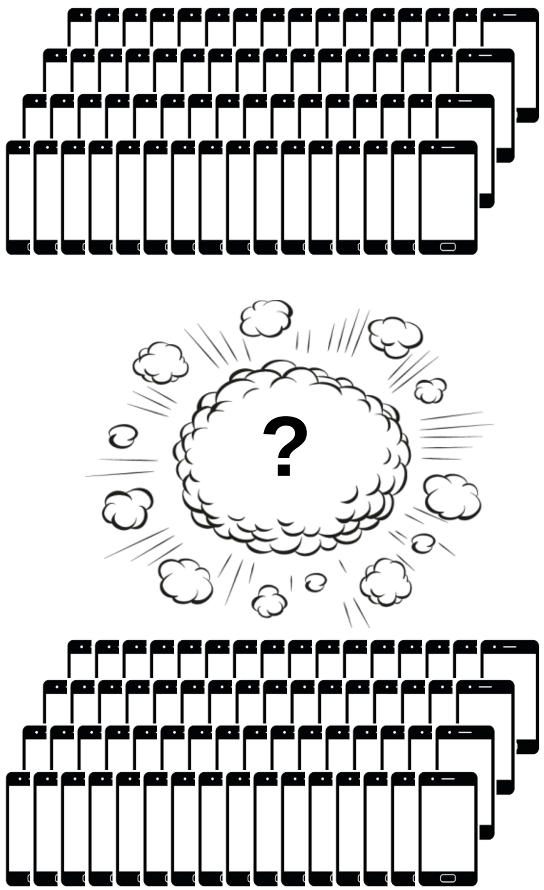
Theoretical Limitation

It converges slowly as more clients join the training.

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla F(u_t)\|^2\right] \leq O\left(\frac{1}{\sqrt{mT}}\right) + O\left(\frac{m}{T}\right)$$

$$\underbrace{O\left(\frac{1}{\sqrt{mT}}\right)}_{\textit{Linear speedup}} > O\left(\frac{m}{T}\right) \text{ only when } T > m^3$$

Implementation Issue



Limited resource at the edge.

↓

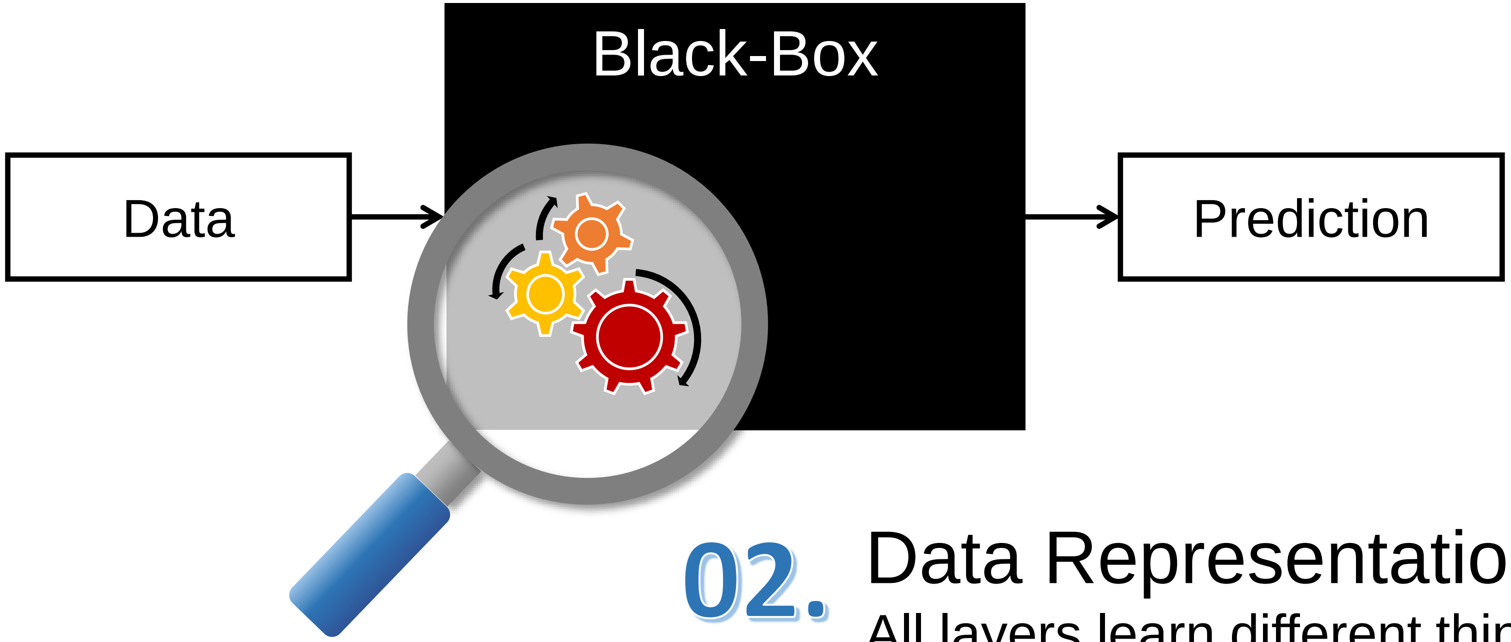
In large-scale FL, the comm. cost is still the bottleneck!

Solutions are in the Black-Box!

Neural network is a black-box (limited interpretability).

01. Layer-wise Model Discrepancy

The degree of divergence at each local model.



02. Data Representation Discrepancy

All layers learn different things!

Outline

Research Background and Motivation

Practical Issues in Federated Learning

Our solution #1

FedLAMA: Layer-wise Adaptive Model Aggregation

Our solution #2

InclusiveFL: Scalable FL on heterogeneous edge devices

Wrap-up

FedML: an open-source software framework for FL

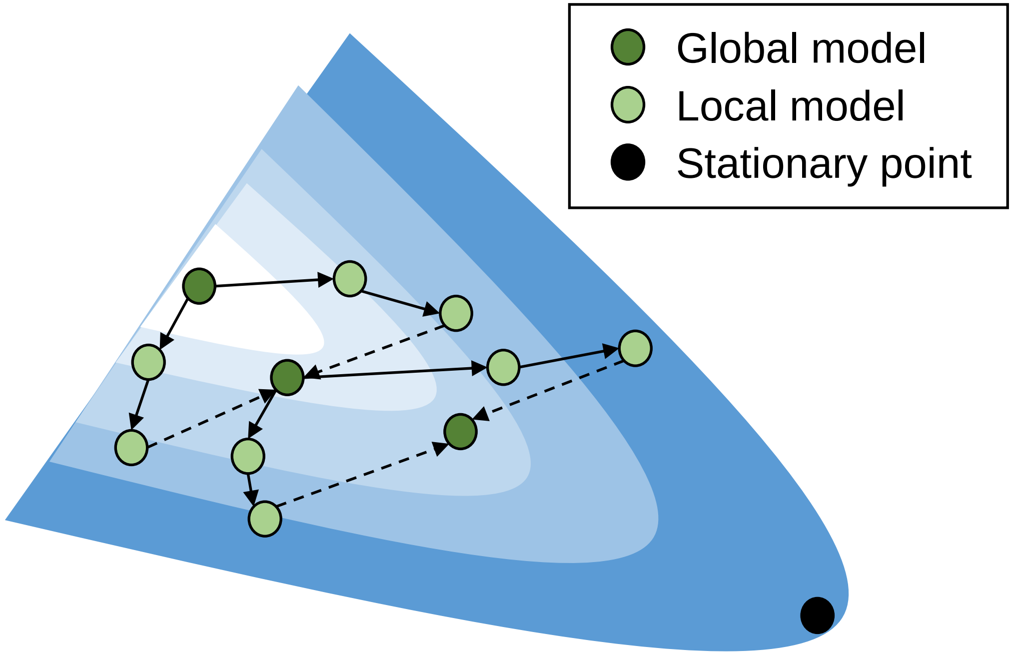
Periodic Model Averaging

- The most foundational model aggregation scheme in FL.
  - $u_t = u_{t-1} - \frac{1}{m}\sum_{i=1}^{m}\sum_{j=0}^{\tau-1}\eta\nabla f\left(x_{t-1,j},\xi_i\right)$

The average of m local accumulated updates

The local updates for τ steps

The model discrepancy among clients is eliminated by fully synchronizing the model after every τ local updates.



Example of FL with 2 clients

Model Discrepancy Matters!

- Model discrepancy
  - The average difference between the global model and local models.
  - The performance difference between centralized training and FL.

FedAvg convergence rate for smooth and non-convex problems

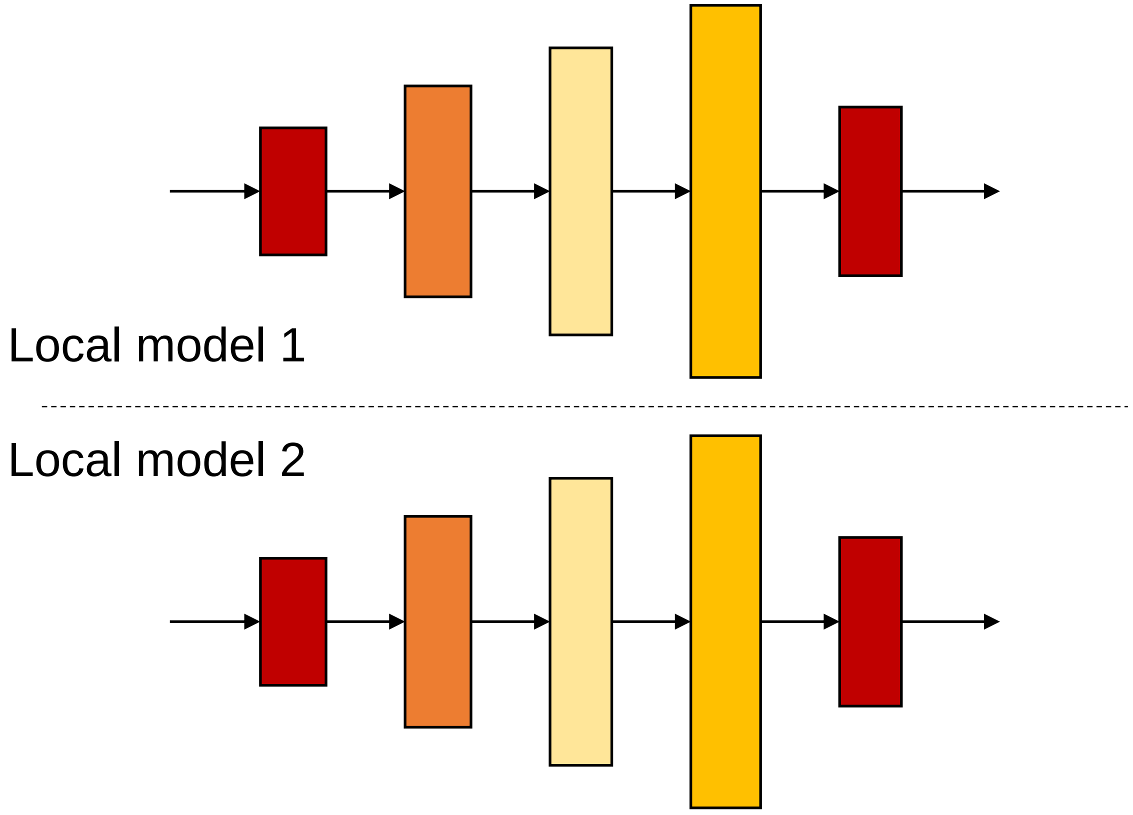
$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\|\nabla F(\mathbf{u}_k)\|^2\right] \leq \overbrace{\frac{2}{\eta K}\mathbb{E}\left[F(\mathbf{u}_1)-F(\mathbf{u}_*)\right]}^{\text{SGD loss}} + \overbrace{2\eta L\sigma^2\sum_{i=1}^{m}(p_i)^2}^{\text{variance}}$$

$$+ \underbrace{\frac{L^2}{K}\sum_{k=1}^{K}\sum_{i=1}^{m}p_i\,\mathbb{E}\left[\left\|\mathbf{u}_k-\mathbf{x}_k^i\right\|^2\right]}_{\textbf{\textit{model discrepancy}}}.$$

Note: Synchronous SGD does not have the model discrepancy because it synchronizes the local gradients every iteration!

Model Discrepancy within *Neural Networks*

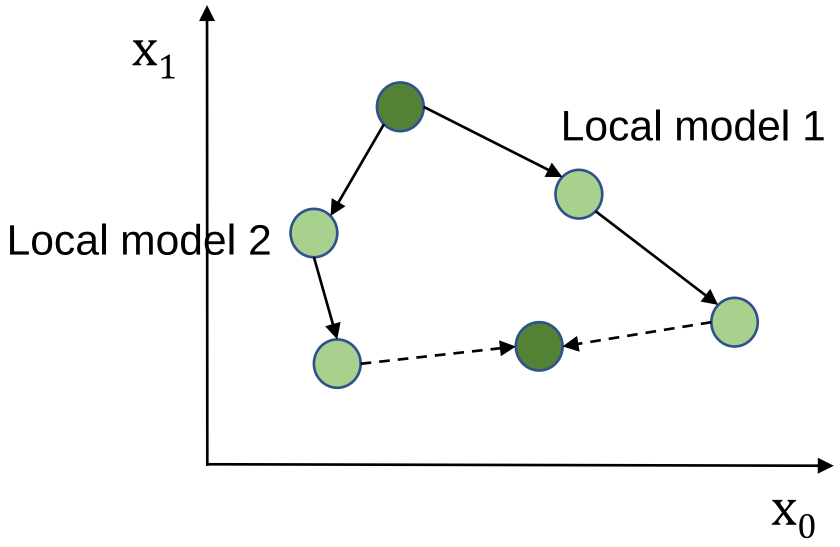
- When training neural networks, layers do not ‘equally‘ contribute to the model discrepancy!



Many factors affect the layer-wise degree of model discrepancy, such as parameter connection patterns, activation functions, and batch normalization.

Inefficient Network Bandwidth Consumption

Key Question: “Should we really synchronize the whole model at once every communication round?”



Aggregating similar parameters does not make any meaningful training progress while spending the network bandwidth!

Layer Prioritization (1/2)

- Layer-wise Model Discrepancy Metric

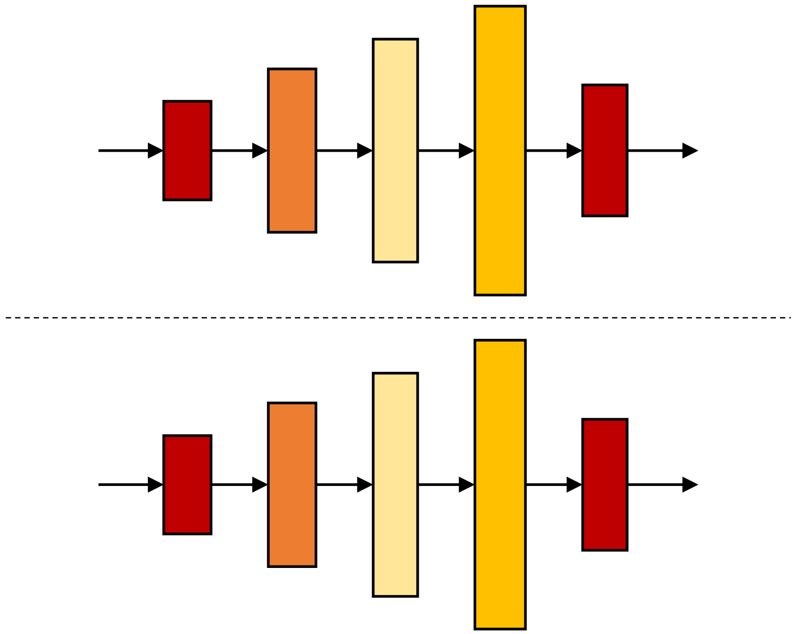
$$d_l = \frac{\overbrace{\frac{1}{m}\sum_{i=1}^{m}\|\mathbf{u}_l - \mathbf{x}_l^i\|^2}^{\text{Average model discrepancy}}}{\tau_l * \underbrace{\dim(\mathbf{u}_l)}_{\text{Number of parameters (communication cost)}}}, \qquad l \in \{1, \cdots, L\}$$

This metric estimates how much model discrepancy can be eliminated at a unit communication cost.

Layer Prioritization (2/2)

- All layers now can be prioritized based on the proposed discrepancy metric!
  - The higher the d_l value, the higher the priority.

$$d_l = \frac{\frac{1}{m}\sum_{i=1}^{m}\|\mathbf{u}_l - \mathbf{x}_l^i\|^2}{\tau_l * \dim(\mathbf{u}_l)}, \qquad l \in \{1, \cdots, L\}$$

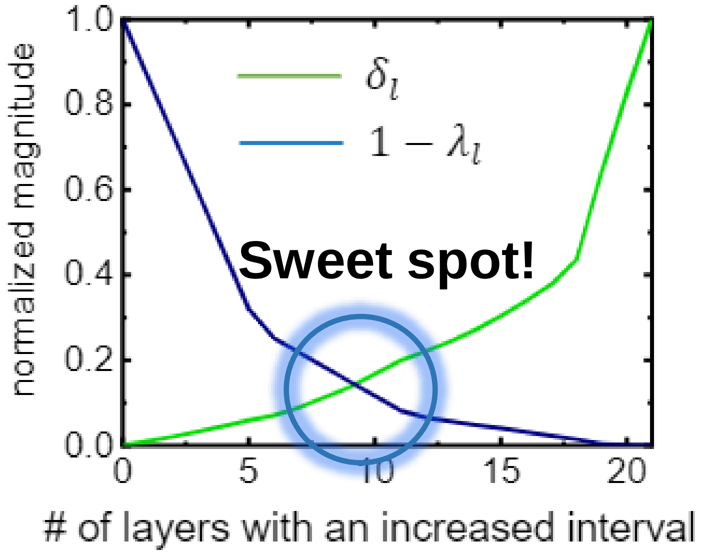


Impact of Layer-Wise Model Aggregation

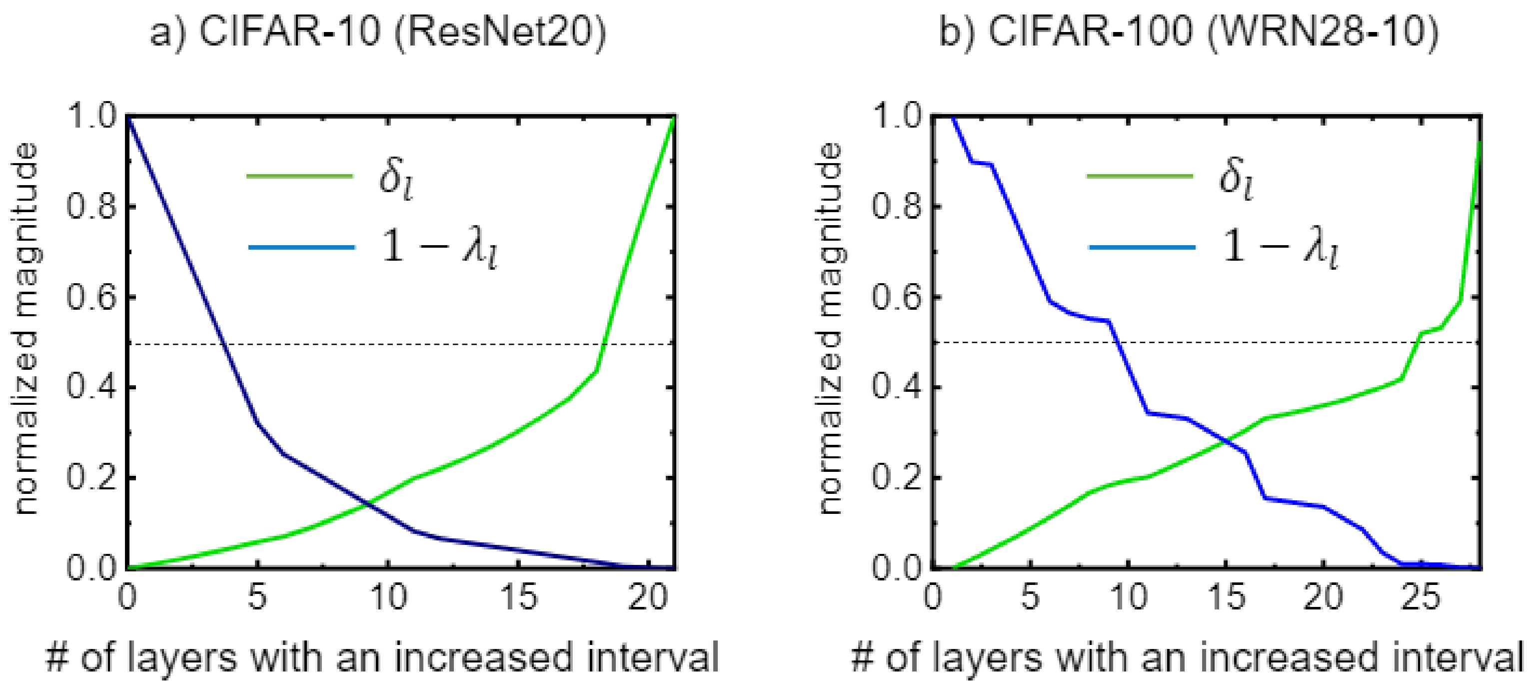
- Sort the layers based on the proposed discrepancy metric (low-to-high).
- Then, what if we increase the aggregation interval at the low-priority layers?

δ_l: the accumulated discrepancy
λ_l: the accumulated communication cost

Intuitively, the sweet spot shows how many layers can have a relaxed aggregation interval.



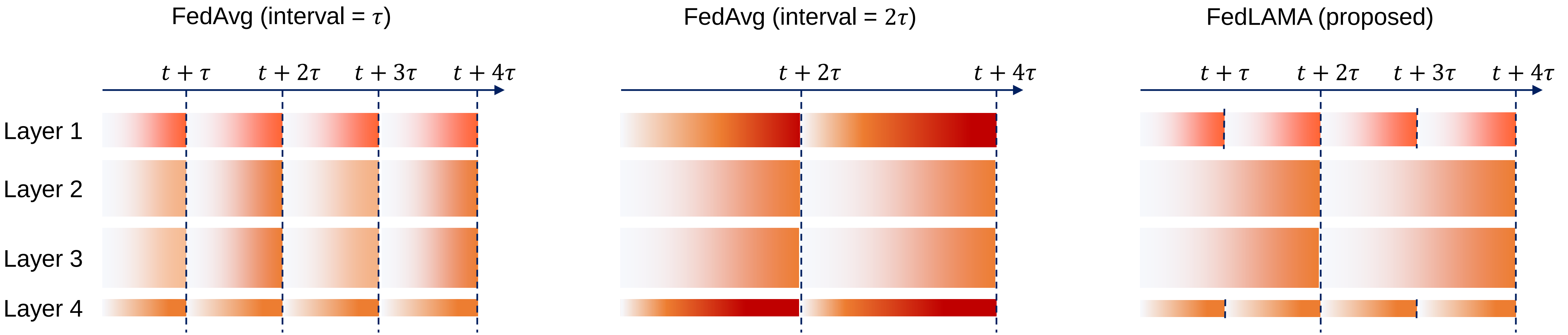
Impact of Layer-Wise Model Aggregation



The sweet spots below 0.5 indicate the improved scalability at the cost of minimal adverse impact on the convergence.

Our Solution

- FedLAMA (**Fed**erated **L**ayer-wise **A**daptive **M**odel **A**ggregation)
  - Find the sweet spot at run-time.
  - Increase the interval by a factor of ϕ at the low-priority layers.



The frequent full aggregations: low model discrepancy but high comm. cost

The less frequent full aggregations: low comm. cost but high model discrepancy

Layer-wise aggregations: low comm. cost and low model discrepancy

Results: Theoretical Analysis

Federated Optimization

$$\min_{\mathbf{x}\in\mathbb{R}^d}\left[F(\mathbf{x}) := \frac{1}{m}\sum_{i=1}^{m} F_i(\mathbf{x})\right]$$

Assumptions – Our analysis assumes the followings.

1. (Smoothness). Each local objective function is L-smooth, that is, $\|\nabla F_i(\mathbf{x}) - \nabla F_i(\mathbf{y})\| \leq L\|\mathbf{x}-\mathbf{y}\|, \forall i \in \{1,\cdots,m\}$.
2. (Unbiased Gradient). The stochastic gradient at each client is an unbiased estimator of the local full-batch gradient: $\mathbb{E}_{\xi_i}\left[\mathbf{g}^i_{t,j}\right] = \nabla F_i(\mathbf{x}^i_{t,j})$.
3. (Bounded Variance). The gradient variance is bounded: $\mathbb{E}_{\xi_i}\left[\|\mathbf{g}^i_{t,j} - \nabla F_i(\mathbf{x}^i_{t,j})\|^2\right] \leq \sigma^2, \forall i \in \{1,\cdots,m\}$.
4. (Bounded Dissimilarity). There exist constants $\beta^2 \geq 1$ and $\kappa^2 \geq 0$ such that $\frac{1}{m}\sum_{i=1}^{m}\|\nabla F_i(\mathbf{x})\|^2 \leq \beta^2\|\frac{1}{m}\sum_{i=1}^{m}\nabla F_i(\mathbf{x})\|^2 + \kappa^2$. If local objective functions are identical to each other, $\beta^2 = 1$ and $\kappa^2 = 0$.

Non-IID dataset

Convergence Rate

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla F(\mathbf{u}_t)\|^2\right] \leq \frac{4}{\eta\tau T}\left(F(\mathbf{u}_0) - F(\mathbf{u}_*)\right) + \frac{2L\eta}{m}\sigma^2 + 3L^2\eta^2(\tau-1)\sigma^2 + 6\eta^2L^2\tau(\tau-1)\kappa^2 \quad (6)$$

where $\mathbf{u}_*$ indicates a local minimum and τ is the largest averaging interval across all the layers ($\tau'\phi$).

Finite Horizon Result

If the learning rate diminishes like $\eta = \frac{\sqrt{m}}{\sqrt{T}}$,

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla F(\mathbf{u}_t)\|^2\right] \leq \mathcal{O}\left(\frac{1}{\sqrt{mT}}\right) + \mathcal{O}\left(\frac{m}{T}\right). \quad (7)$$

If $T > m^3$, the first term on the right-hand side becomes dominant and it achieves linear speedup. That is, `FedLAMA`

- `FedLAMA` provides a solid convergence guarantee.
- It achieves linear speedup when η is sufficiently small.
- It's as fast as FedAvg with the interval $\phi\tau$

Results: Empirical Study

- FL simulation with 128 Clients
  - Random 25% of the clients participate in each communication round.
  - 10,000 local steps in total.

Table 1: The CIFAR-10 (ResNet20) classification results. The total number of local steps is 10,000 and the local batch size is 32. The dataset is split based on a Dirichlet distribution ($\alpha = 0.1$) w.r.t the labels.

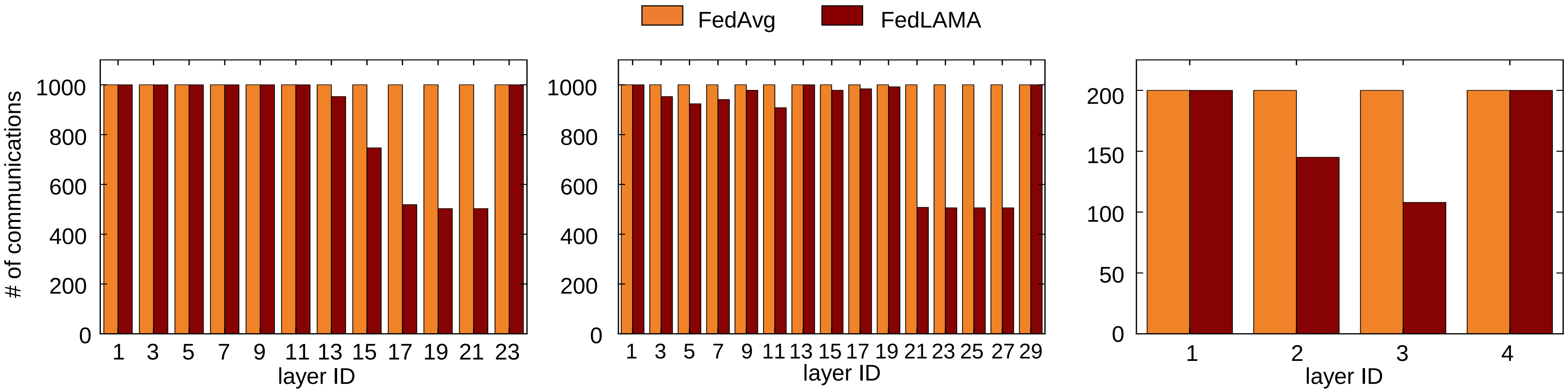
FedAvg (Periodic Full Avg.)					FedLAMA							
Full training					Early stopping						Full training	
LR	τ'	ϕ	Validation acc.	C ratio	LR	τ'	ϕ	Validation acc.	# of steps	C ratio	Validation acc.	C ratio
0.4	10	1	$81.66 \pm 0.3\%$	100%	0.4	10	1	$81.66 \pm 0.3\%$	9,860	100%	$81.66 \pm 0.3\%$	100%
0.3	20	1	$72.99 \pm 0.5\%$	50%	0.2	10	2	**77.33** $\pm 0.3\%$	**5,160**	**32.01%**	**81.46** $\pm 0.3\%$	**61.55%**
0.3	40	1	$66.64 \pm 0.5\%$	25%	0.2	10	4	**68.32** $\pm 0.4\%$	**4,120**	**18.65%**	**80.60** $\pm 0.4\%$	**44.36%**

As the interval increases, the periodic full averaging rapidly loses the accuracy.

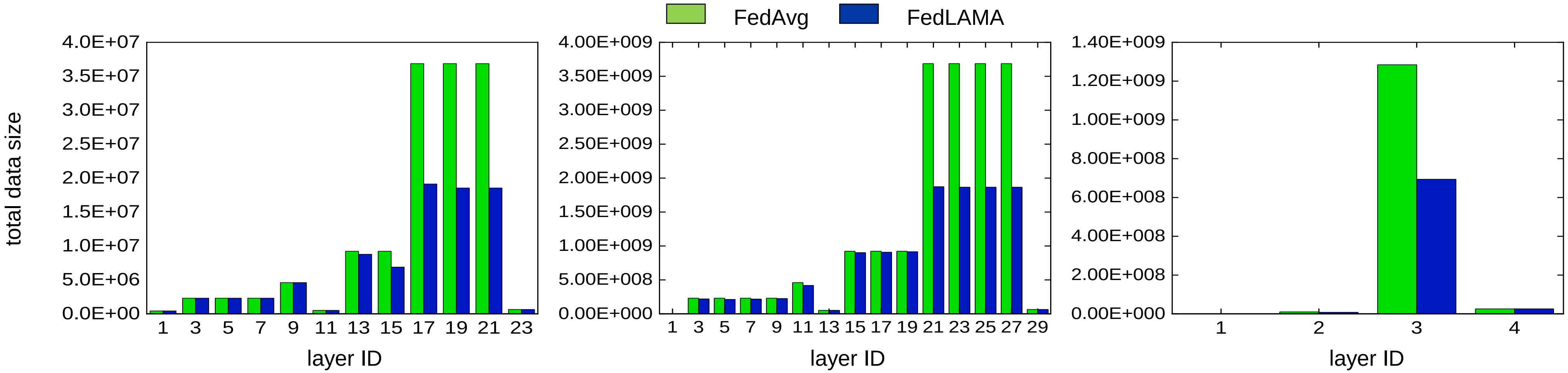
FedLAMA achieves the same accuracy within significantly fewer steps.

After the same 10,000 steps, FedLAMA achieves much higher accuracy!

Results: Communication Cost



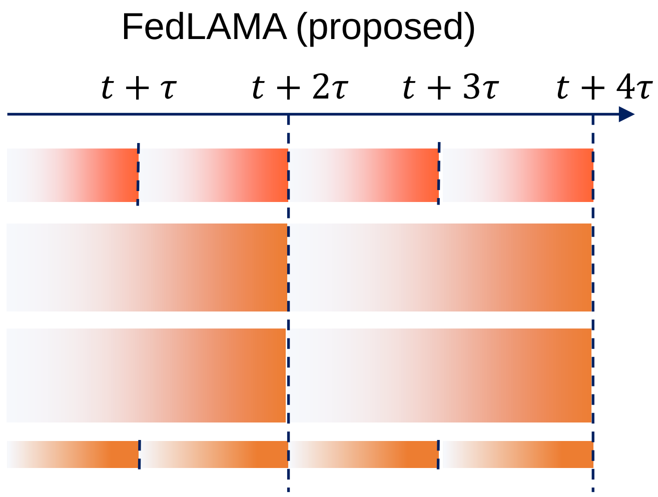
a) CIFAR-10 (ResNet20) b) CIFAR-100 (WRN28-10) c) FEMNIST (CNN)



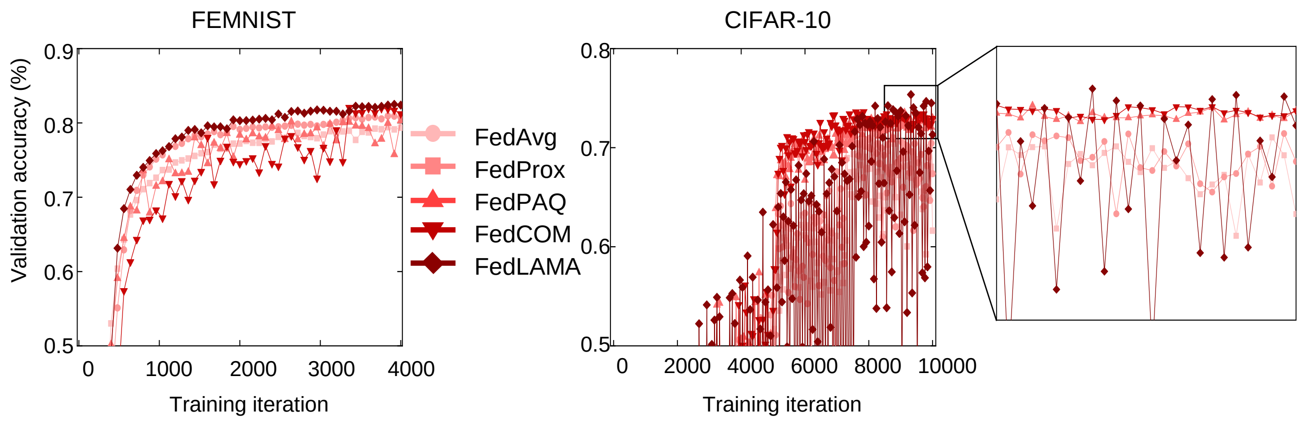
Summary

FedLAMA, a layer-wise adaptive model aggregation scheme shows the most efficient way of spending the network bandwidth in FL!

https://arxiv.org/abs/2110.10302



FedLAMA is a novel model aggregation scheme that can be generally applied to any FL applications!



Outline

Research Background and Motivation
Practical Issues in Federated Learning

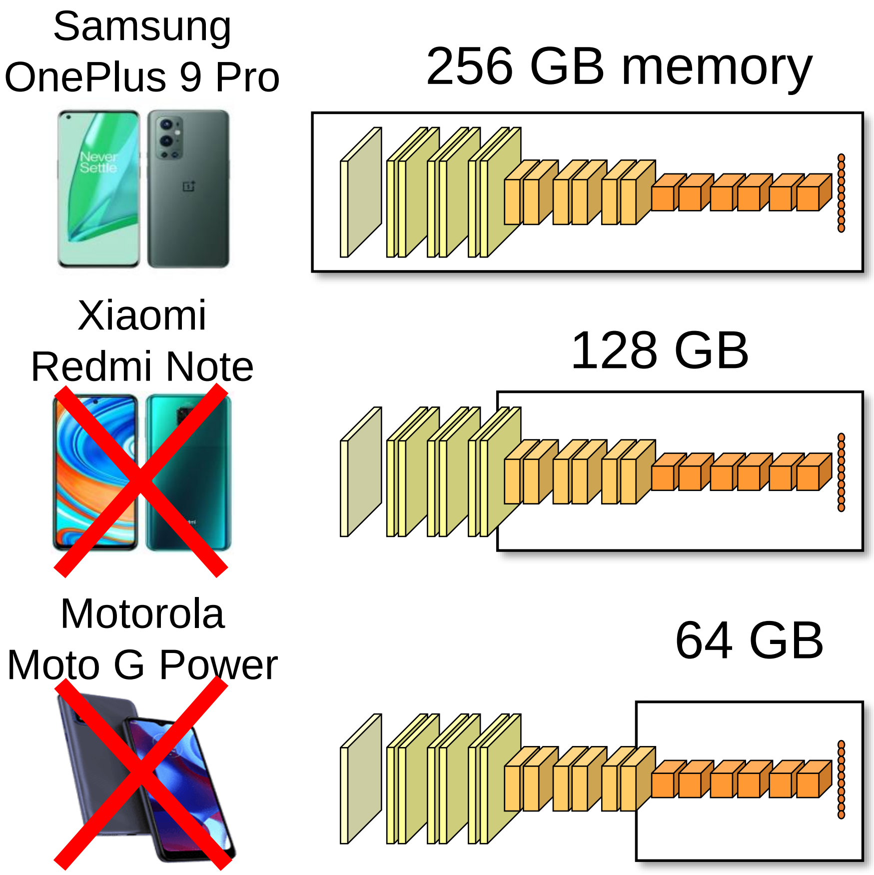
Our solution #1
FedLAMA: Layer-wise Adaptive Model Aggregation

Our solution #2
InclusiveFL: Scalable FL on heterogeneous edge devices

Wrap-up

FedML: an open-source software framework for FL

Heterogeneous Systems



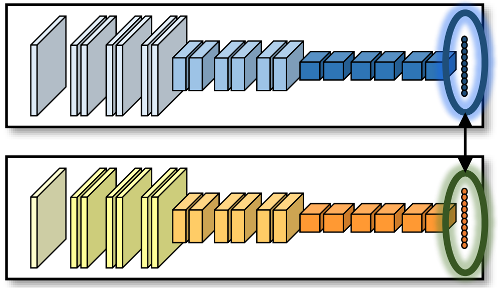
The conventional assumption of 'homogeneous' clients

What if the model is so large?

- Small and weak devices may not even hold the whole model in their memory space!

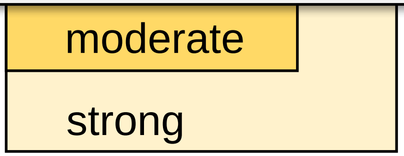
Heterogeneous Clients in FL

- Knowledge Distillation
  - Co-distillation
  - FedHe

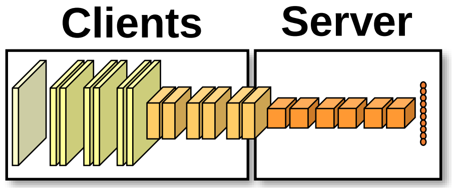


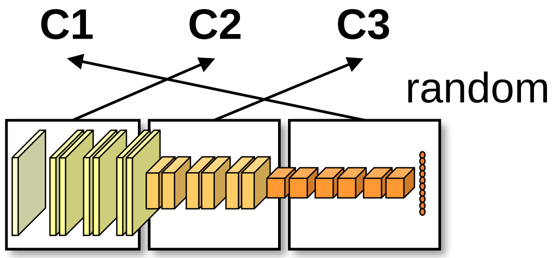
The errors are exchanged and then back-propagated!

No principled way of splitting and utilizing the ‘weak’ clients!



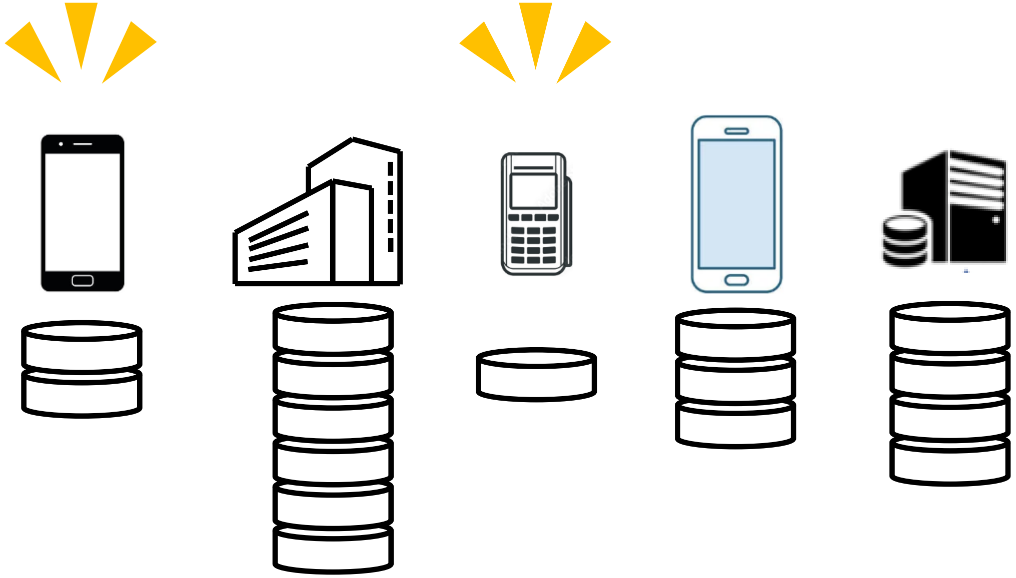
- Partial training strategies
  - ResIST
  - SplitFed
  - Model approximation / decomposition





Goal: Enable Weak Client Participation in FL

- We consider ***weak clients*** that cannot effectively train the full model on its own.
  - Limited memory space
  - Too weak compute power



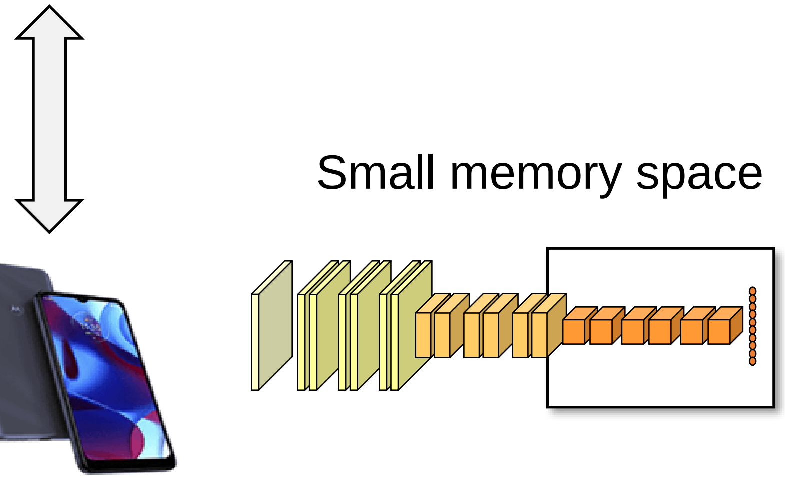
In order to utilize distributed data in the real-world, we should make all available devices participate in the training!

Key Question

If a weak client takes in charge of a part of neural network, which part should be assigned?

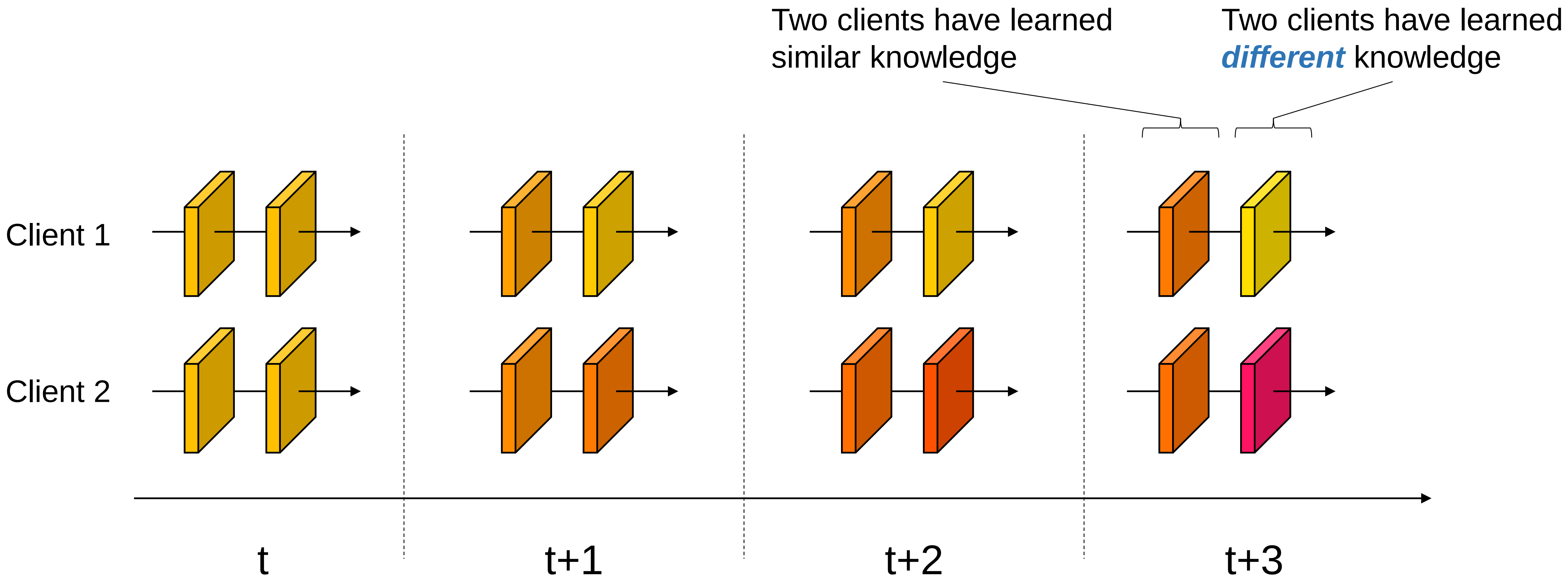
Limited network bandwidth

Small memory space

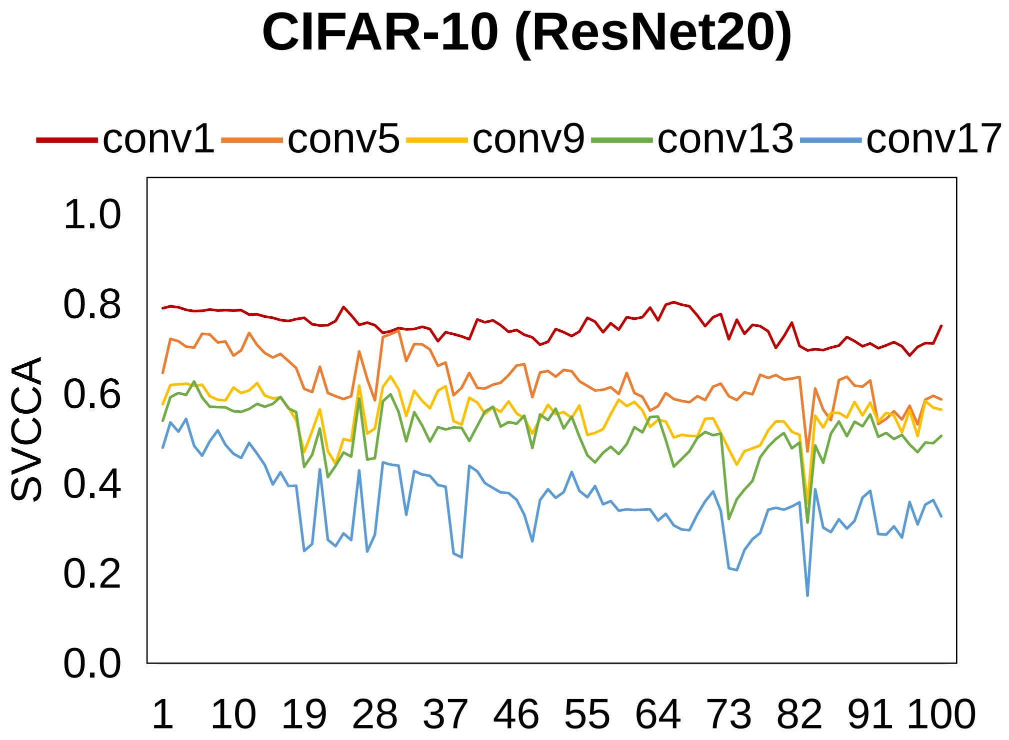


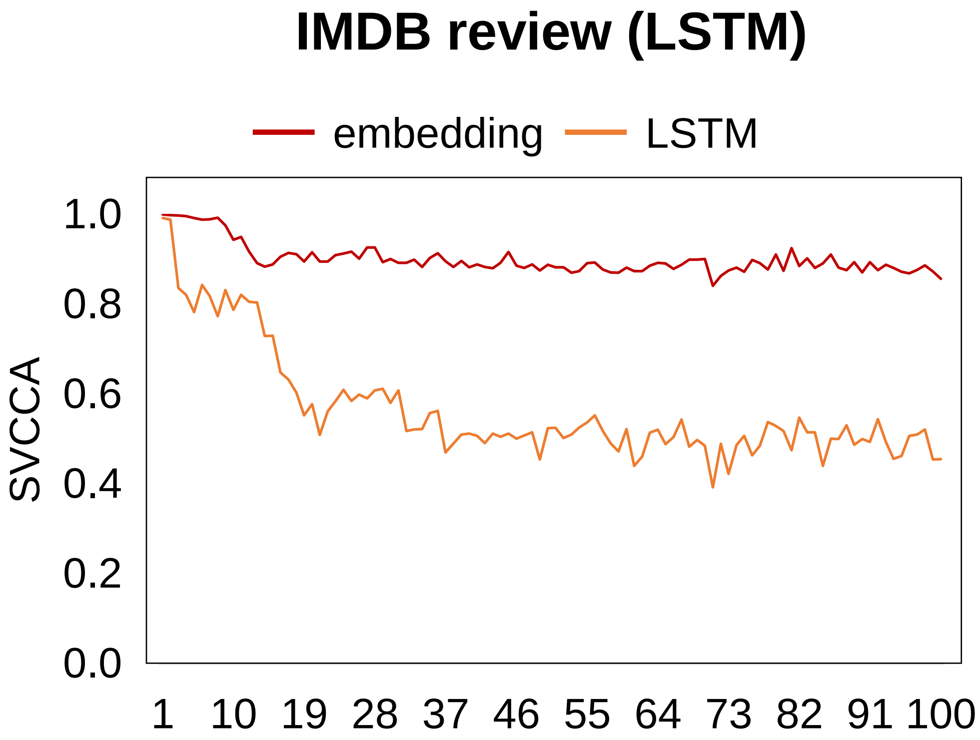
Weak compute power

Hypothesis: layers may have different data characteristics



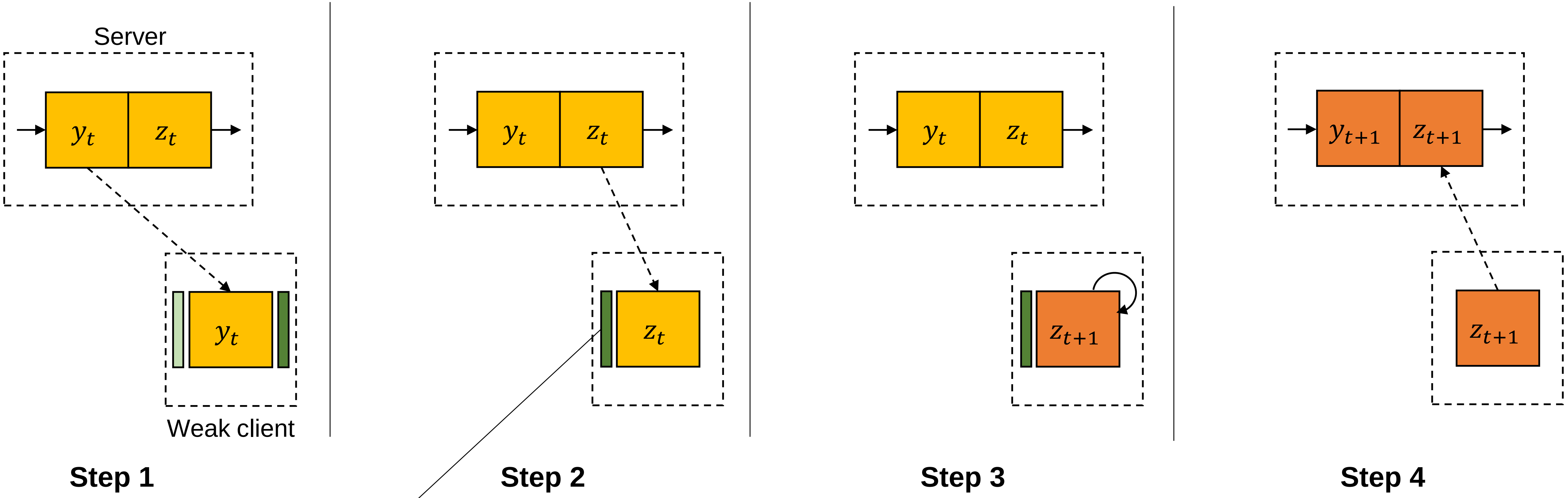
Empirical Study: Layer-wise Data Representation Analysis





The input-side layers learn similar data across independent clients!

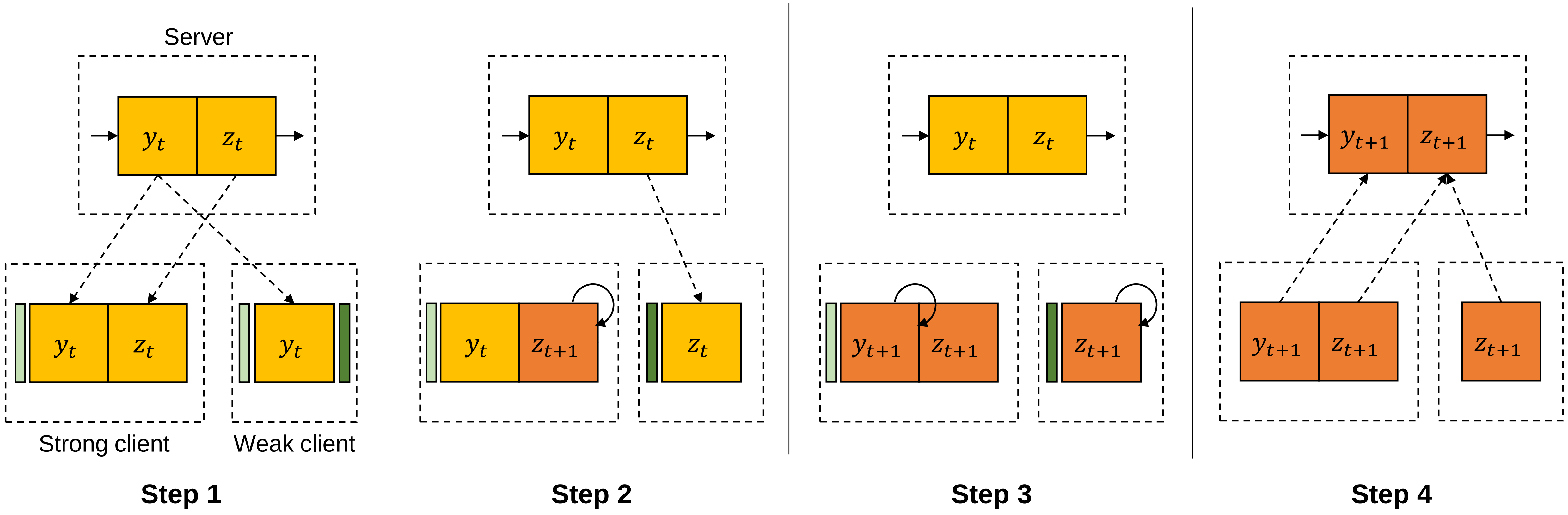
Partial Training at Weak Clients



The intermediate output is recorded and reused multiple steps!

Weak clients contribute only to the output-side subset of layers.

InclusiveFL: Heterogeneous System-Aware FL



Result: Theoretical Analysis

$$\mathbf{x_t} = (\mathbf{y_t}, \mathbf{z_t})$$

Full model

$$\mathbf{y_{t+1}} = \mathbf{y}_t - \frac{\eta}{s}\sum_{i=1}^{s}\sum_{j=0}^{\tau-1}\nabla f(\mathbf{y}_{t,j}^{i})$$

Input-side sub-model

$$\mathbf{z_{t+1}} = \mathbf{z}_t - \frac{\eta}{m}\sum_{i=1}^{m}\sum_{j=0}^{\tau-1}\nabla f(\mathbf{z}_{t,j}^{i}),$$

Output-side sub-model

Theorem 1. *Suppose all m local models are initialized to the same point $\mathbf{x}_0$. Under Assumption $1 \sim 3$, if Algorithm 2 runs for T communication rounds and the learning rate satisfies $\eta \leq \min\left\{\frac{1}{\tau L_{max}}, \frac{1}{4L_{max}\sqrt{\tau(\tau-1)}}\right\}$, the average-squared gradient norm of $\mathbf{x}_t$ is bounded as follows.*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla F(\mathbf{x}_t)\|^2\right] \leq \frac{14}{3T\eta\tau}\left(F(\mathbf{x}_0) - F(\mathbf{x}_*)\right)$$
$$+ \left(\frac{7L_y\eta}{3s} + \frac{16L_y^2\eta^2(\tau-1)}{3}\right)\sigma_y^2 + \left(\frac{14}{3s} + \frac{64L_y^2\eta^2\tau(\tau-1)}{3}\right)\bar{\sigma}_y^2$$
$$+ \left(\frac{7L_z\eta}{3m} + \frac{8L_z^2\eta^2(\tau-1)}{3}\right)\sigma_z^2 + \left(\frac{32L_z^2\eta^2\tau(\tau-1)}{3}\right)\bar{\sigma}_z^2$$

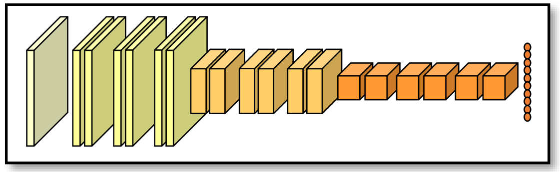
Does not go to zero even when η diminishes. Thus, it converges to **the neighborhood region** of a stationary point rather than the exact point.

Experimental Settings

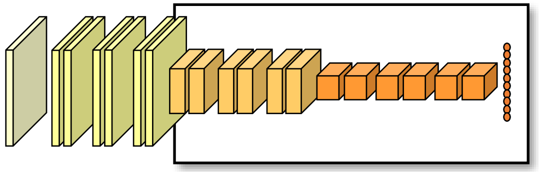
Table 1: The model size of the three different types of clients.

Removed layers of Resnet20 (CIFAR-10)	Number of parameters (p)	Number of activations (a)	Capacity
(Strong) -	272,762	6,947,136	1.00
(Moderate) The first conv. layer + the first 3 residual blocks	257,994	2,752,832	0.42
(Weak) The first conv. layer + the first 6 residual blocks	206,346	917,824	0.16

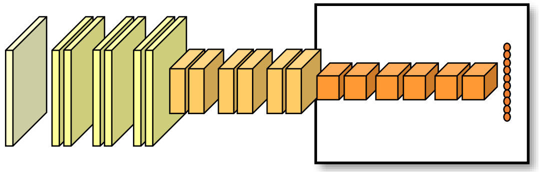
Strong Clients (100%)



Moderate Clients (42%)

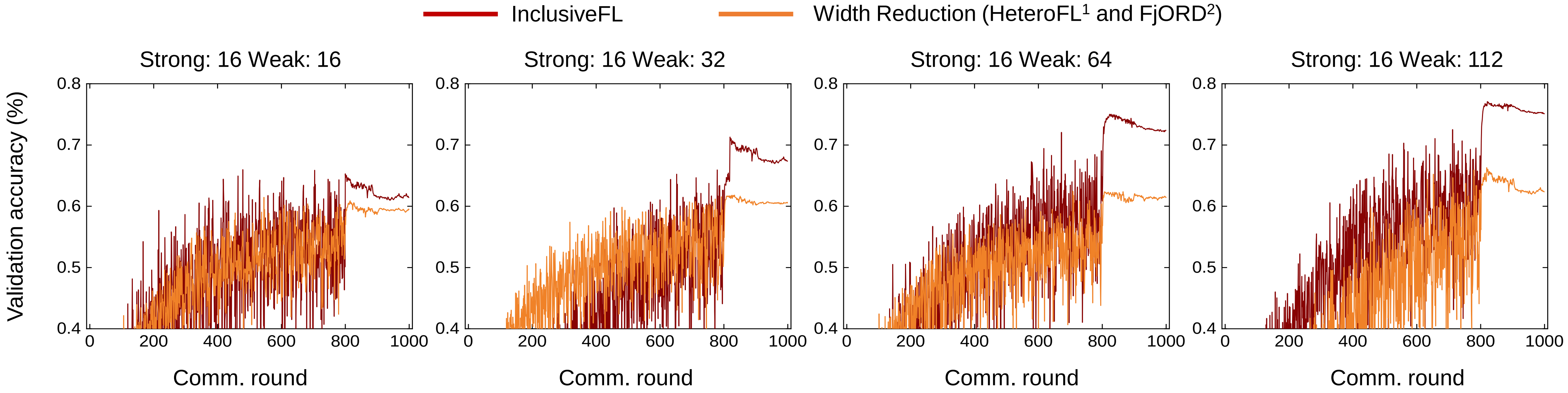


Weak Clients (16%)



Result: Comparison to SOTA

# of strong clients	# of weak clients	Width Reduction [5, 11]	InclusiveFL
16	0	60.15 ± 1.5%	
16	16	61.34 ± 2.1%	**66.62** ± 1.1%
16	32	62.09 ± 1.5%	**72.60** ± 1.2%
16	64	63.68 ± 3.3%	**74.79** ± 0.8%
16	112	65.01 ± 2.9%	**77.34** ± 1.6%



These results empirically proves that InclusiveFL better utilize the 'weak' clients than HeteroFL and FjORD!

1. Diao et al., HeteroFL: Computation and Communication Efficient Federated Learning for Heterogeneous Clients, arXiv
2. Horvath et al., FjORD: Fair and Accurate Federated Learning under heterogeneous targets with Ordered Dropout, NeurIPS 21

Result: Comprehensive Empirical Study

Table 9. The non-IID CIFAR-10 classification performance under various heterogeneous FL settings. 'Width Reduction' corresponds to the static version of HeteroFL and FjORD without local knowledge distillation.

	Strong client		Moderate client		Weak client		Avg. Capacity	Inclusive FL	Width Reduction
case 1	128	(100%)	0	(0%)	0	(0%)	1.00	80.35 ± 0.2%	
case 2	64	(50%)	64	(50%)	0	(0%)	0.71	**80.07** ± 0.2%	76.77 ± 1.3%
case 3	32	(25%)	96	(75%)	0	(0%)	0.57	**79.20** ± 0.3%	67.92 ± 2.1%
case 4	16	(12.5%)	112	(87.5%)	0	(0%)	0.49	**79.11** ± 0.2%	59.03 ± 0.8%
case 5	64	(50%)	0	(0%)	64	(50%)	0.58	**80.21** ± 0.4%	72.97 ± 2.5%
case 6	32	(25%)	0	(0%)	96	(75%)	0.37	**78.91** ± 0.4%	69.70 ± 2.1%
case 7	16	(12.5%)	0	(0%)	112	(87.5%)	0.27	**77.19** ± 1.6%	54.53 ± 2.9%
case 8	32	(25%)	32	(25%)	64	(50%)	0.44	**79.78** ± 0.1%	67.93 ± 1.5%
case 9	16	(12.5%)	32	(25%)	80	(62.5%)	0.33	**80.04** ± 0.1%	59.59 ± 0.8%
case 10	16	(12.5%)	16	(12.5%)	96	(75%)	0.30	**78.01** ± 0.2%	58.17 ± 2.2%

Result: Timings on Real Edge Devices

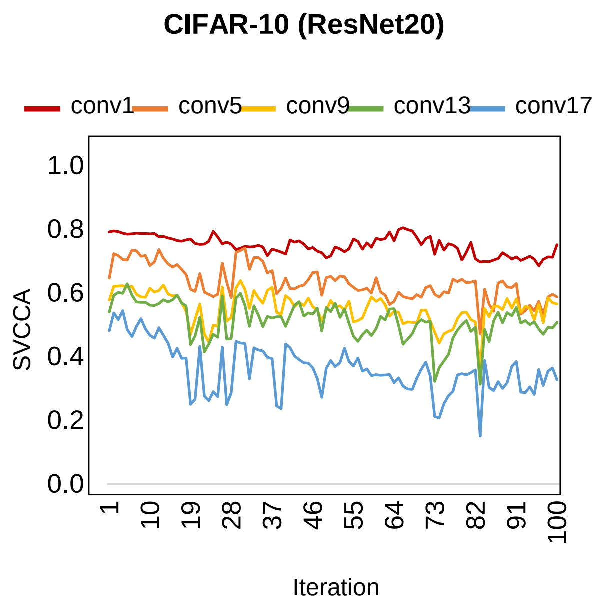
- ResNet20 training implemented using MNN software framework
- Samsung OnePlus 9 Pro (2021 model)

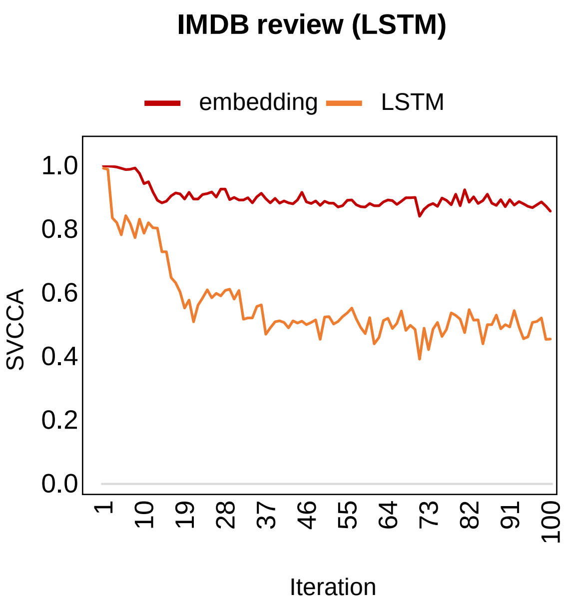
		InclusiveFL			Width Reduction		
Workload	Model Size	Computation	I/O	End-to-End	Computation	I/O	End-to-End
Feed-forward	Strong	2095.4 ms	-	2095.4 ms	2095.4 ms	-	2095.4 ms
	Moderate		678.4 ms	2773.8 ms	1431.0 ms	-	1431.0 ms
	Weak		1316.8 ms	3412.2 ms	936.7 ms	-	936.7 ms
Backpropagation	Strong	419,643.8 ms	-	419,643.8 ms	419,643.8 ms	-	419,643.8 ms
	Moderate	197,265.0 ms		197,265.0 ms	317,669.7 ms		317,669.7 ms
	Weak	85,448.3 ms		85,448.3 ms	187,580.4 ms		187,580.4 ms

InclusiveFL has extra I/O time, however it significantly reduces the backward pass time.

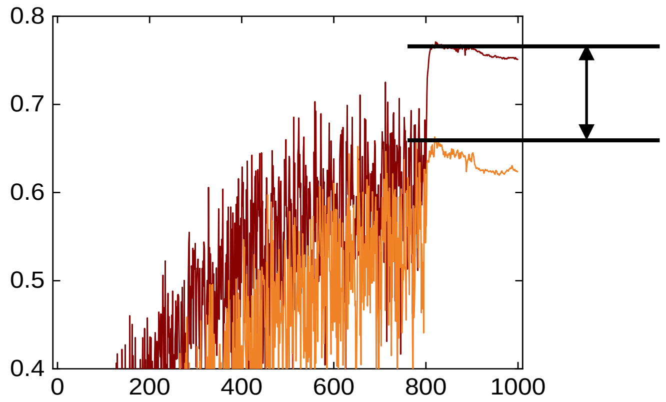
Summary

Our empirical study demonstrates that the input-side layers learn 'similar' data representations regardless of the data distribution.





Layer-wise Partial Model Training effectively enables the weak clients to contribute to the global model training!



Outline

Research Background and Motivation

Practical Issues in Federated Learning

Our solution #1

FedLAMA: Layer-wise Adaptive Model Aggregation

Our solution #2

InclusiveFL: Scalable FL on heterogeneous edge devices

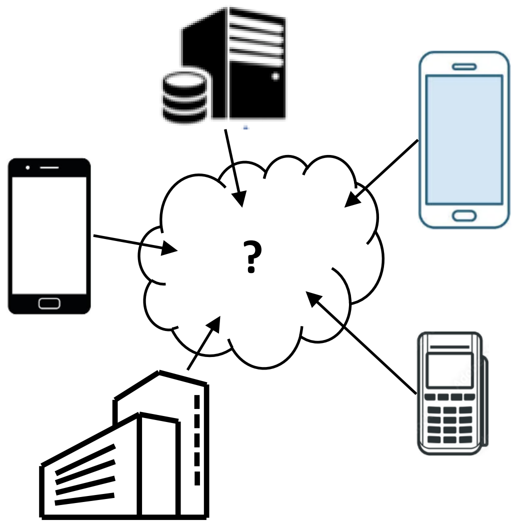
Wrap-up

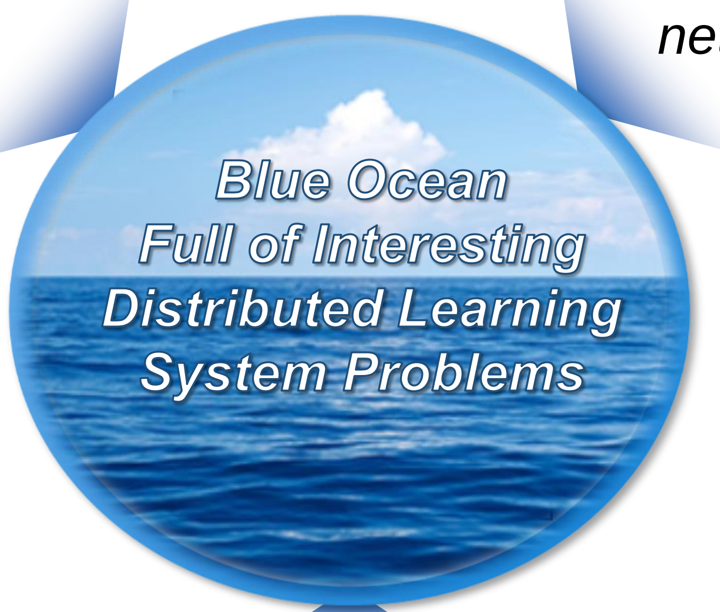
FedML: an open-source software framework for FL

Promising Research Directions

1. Extremely large-scale Federated Learning

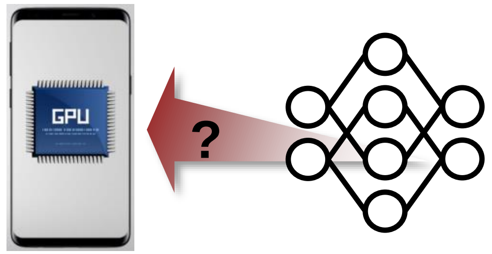
Edge devices + HPC systems





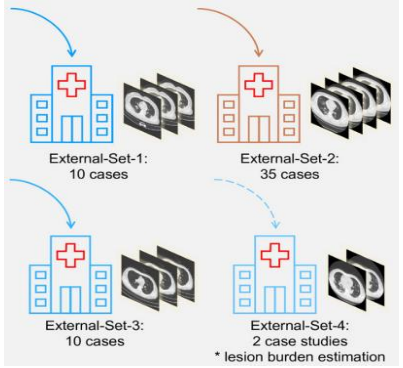
2. IoT systems for Machine Learning

Efficient system software for neural network training



3. Application-specific Federated Learning

Large-scale Fusion (inter-disciplinary) Research



Outline

Research Background and Motivation
Practical Issues in Federated Learning

Our solution #1
FedLAMA: Layer-wise Adaptive Model Aggregation

Our solution #2
InclusiveFL: Scalable FL on heterogeneous edge devices

Wrap-up

FedML: an open-source software framework for FL

Landing FL in Real World!

An open source Federated Learning framework developed by University of Southern California Ph.D. students.

FedML

FedML

Open Source ▼ Platform ▼ Solutions ▼ Technology ▼ Contact About Us

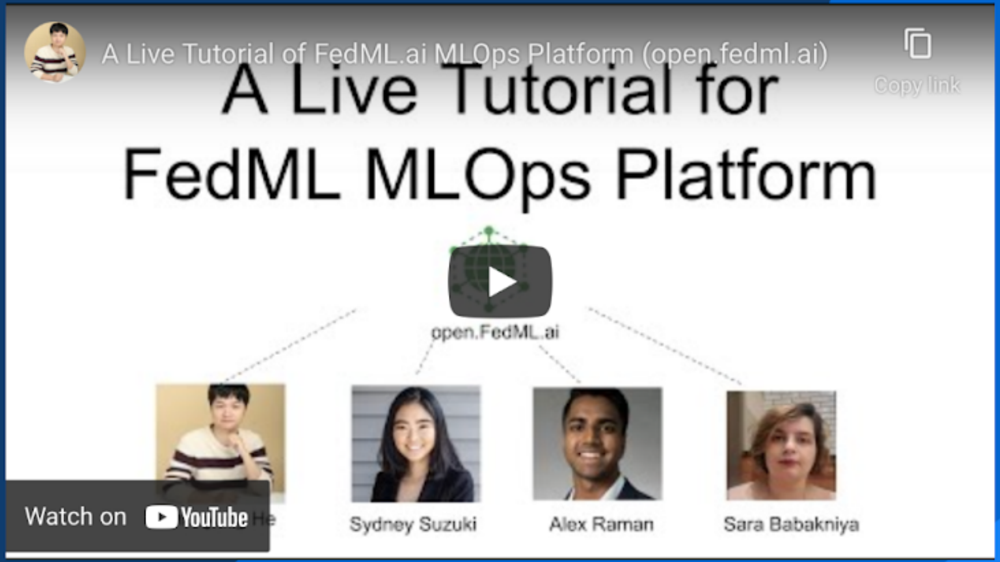
Social, Secure, Scalable, and Efficient

Federated Learning/Analytics and Edge AI Platform in Open Collaboration

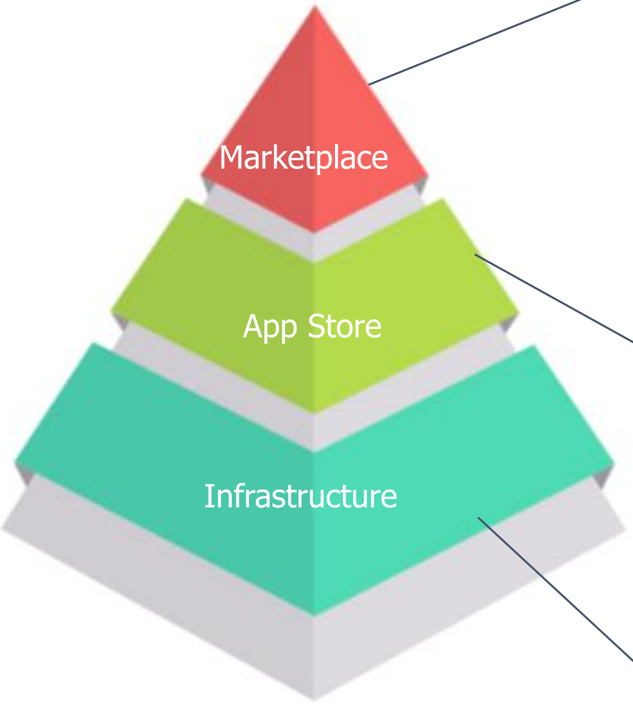
Enable machine learning everywhere

- Cutting-edge algorithms backed by years of **Open Source**-oriented research (50+ scientific publications, 900+ early slack users, and 300+ GitHub forks)
- Lightweight and cross-platform **Edge AI SDK** for GPUs, smartphones, and IoTs
- User-friendly **MLOps** platform to simplify collaboration and real-world deployment
- Platform-supported vertical **Solutions** *across a broad range of industries*

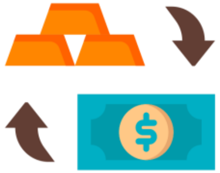
LIVE DEMO SIGN UP JOIN OUR COMMUNITY



FedML = Decentralized AI x Web3

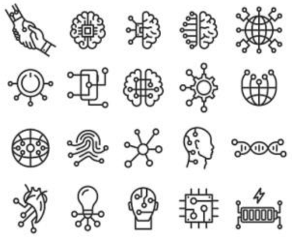


Web 3-based AI Marketplace



transparency, traceability, low cost, high performance

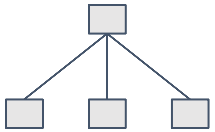
AI App Store



Democratizing AI with out-of-the-box experience
AI applications for diverse verticals

AI Infrastructure

open and secure



decentalized/federated

deployable and monitorable

FedML Platform

Open Source

An international community for cutting-edge algorithms

Wide Adoption by AI Community

FedML open source library has been used widely in the world, including researchers and engineers from the *United States, Canada, China, Germany, Denmark, Korea, and Singapore*. Some of them are from big companies *Google, Amazon, Adobe, Cisco, and Huawei*, as well as well-known research-oriented universities such as *Stanford, Princeton, USC, HKUST, Tsinghua*, etc. They published in top-tier AI conferences including ICML, NeurIPS, ICLR, and AAAI.

3	50+	1500+	10+
Products including open-source, edge AI SDK, and MLOps platform	Scientific Publications in ML/FL algorithms, security/privacy, systems, and applications	Open-source Community Users from all over the world	Industrial Collaborators from top-tier companies

GitHub: https://github.com/FedML-AI
Documentation: https://doc.fedml.ai
Join Slack Community

Edge AI SDK

A lightweight and cross-platform design for secure edge training

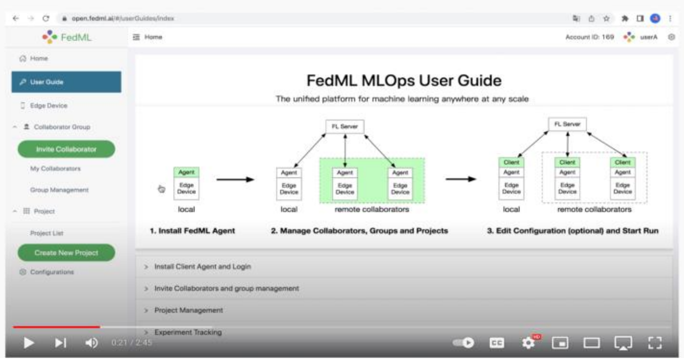
write once, run everywhere: enabling a smooth migration from in-lab simulation (open source) to real-world distributed system

Engineering Stack: python, Java, C++, android, ubuntu, node, Vue.js, PyTorch, docker, CLOUD NATIVE COMPUTING FOUNDATION, amazon web services

MLOps Cloud

A user-friendly design for zero-code real-world deployment

user-friendly, zero-code, deployment

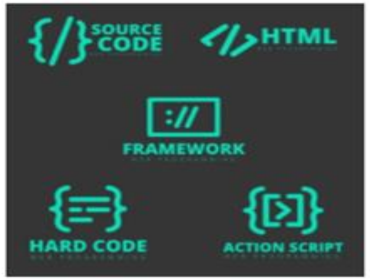


https://fedml.ai/platform-tutorial/

FedML Four Solutions

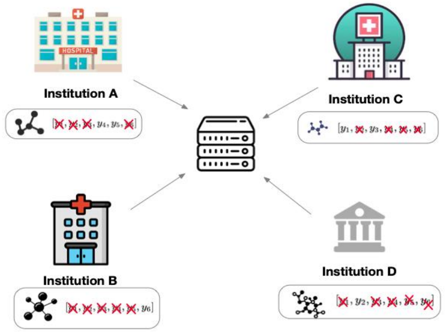
FedML Parrot
(strong imitation)

FL Simulator in Real-World



constantly evolves and brings innovation via open-source contributions

Strong and Simple Connector for Federated Learning from Data Silos (hospitals, banks, factories, …)

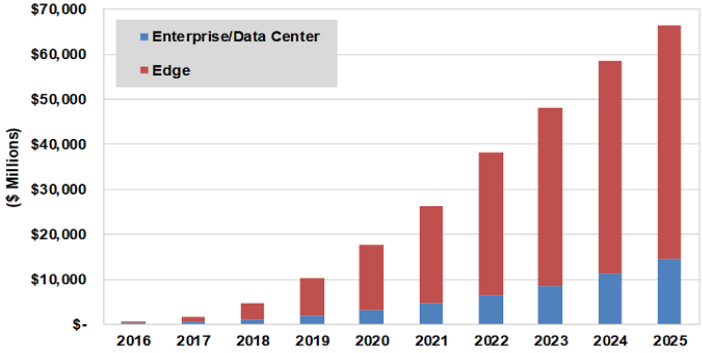


FedML Octopus
(perfect adsorption)

FedML BeeHive
(swarm intelligence)

Collaborative Learning from Scattered Data on Edge Devices (mobile, IoT,...)

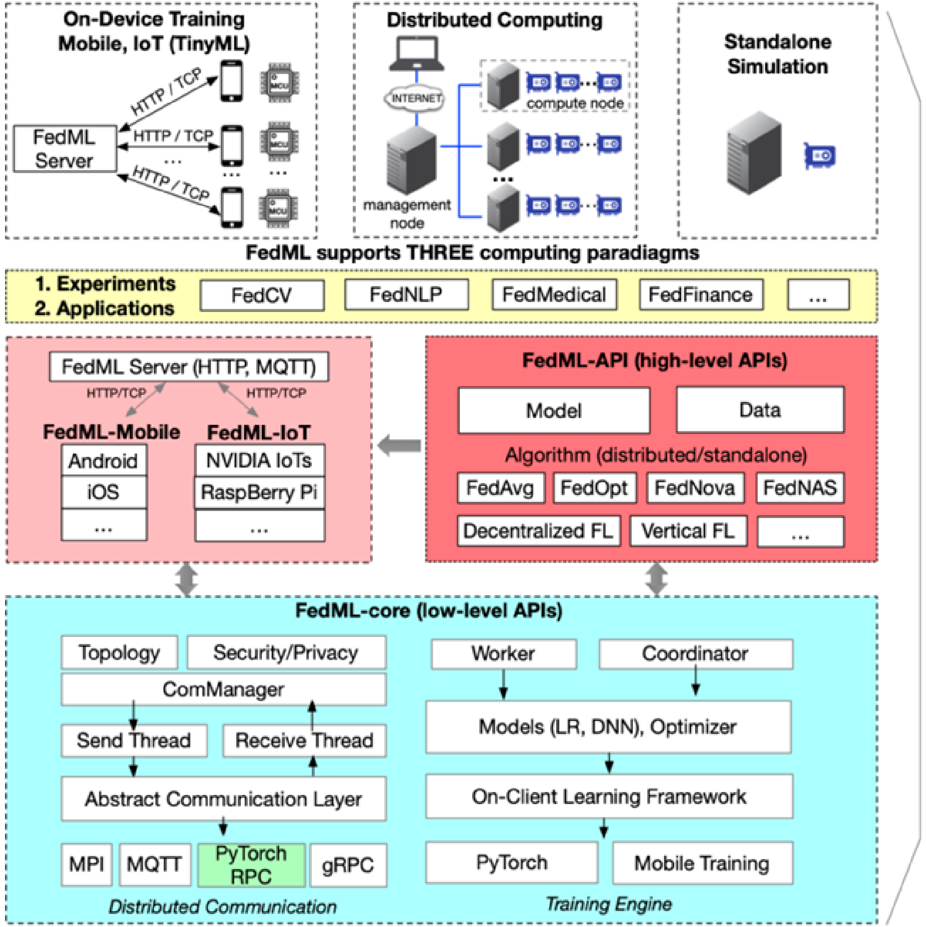
Speedy Training of Large Models (Harnessing the Explosion of Compute-power at the Edge)



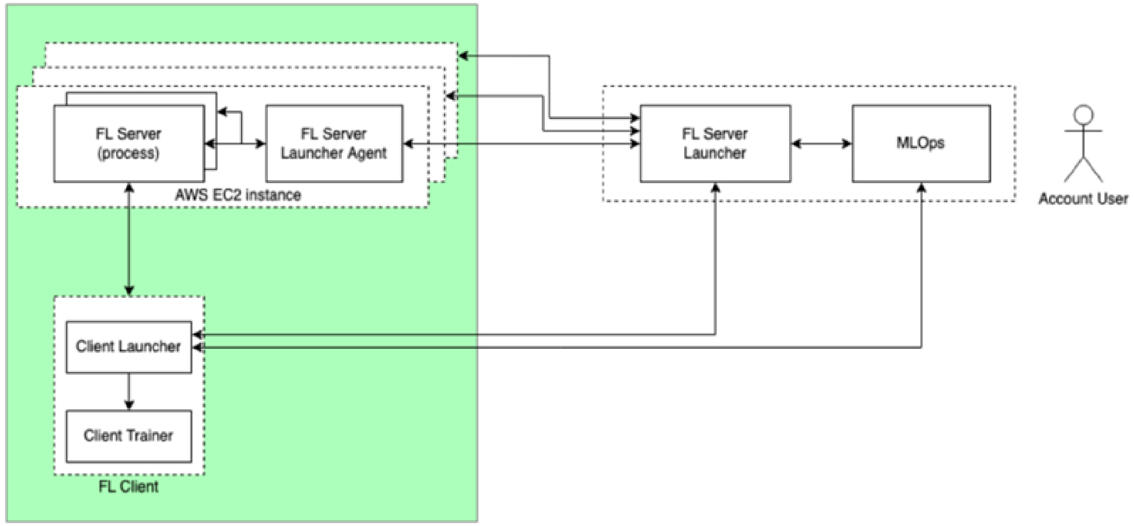
FedML Cheetah
(aim at speed)

FedML Overview

FedML Open Source Library



FedML Edge-Cloud Platform

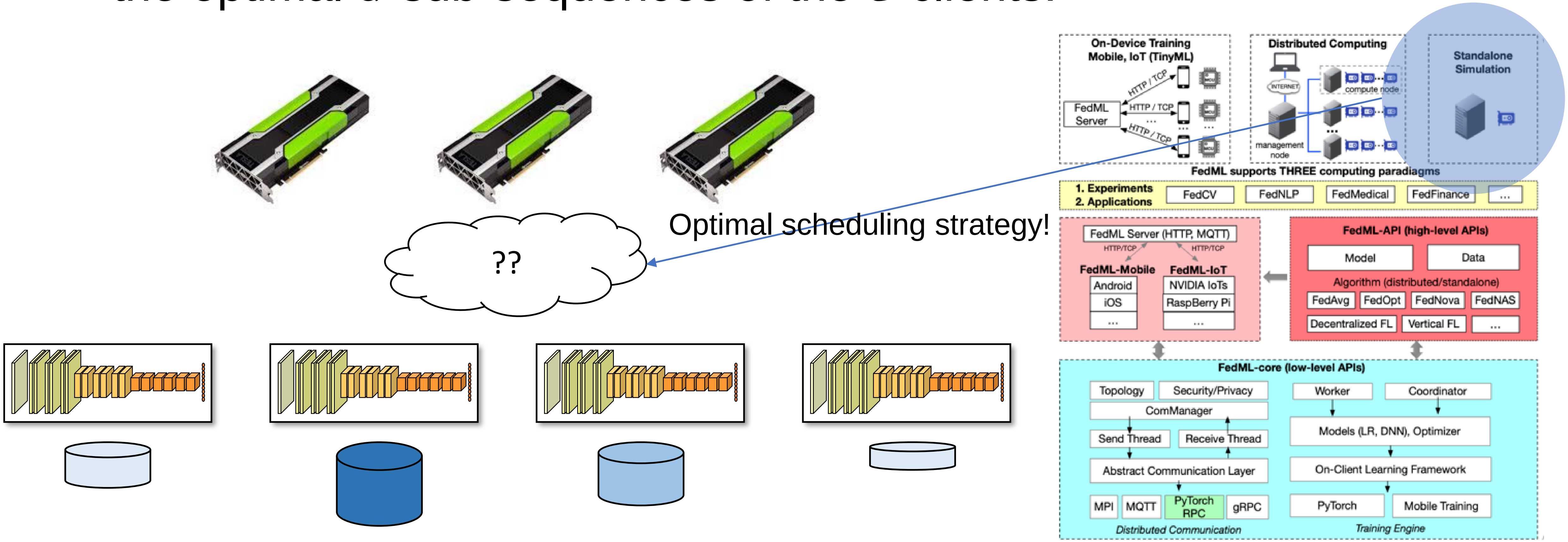


write once, run everywhere: enabling a smooth migration from in-lab simulation (open source) to real-world distributed system

Engineering Stack:

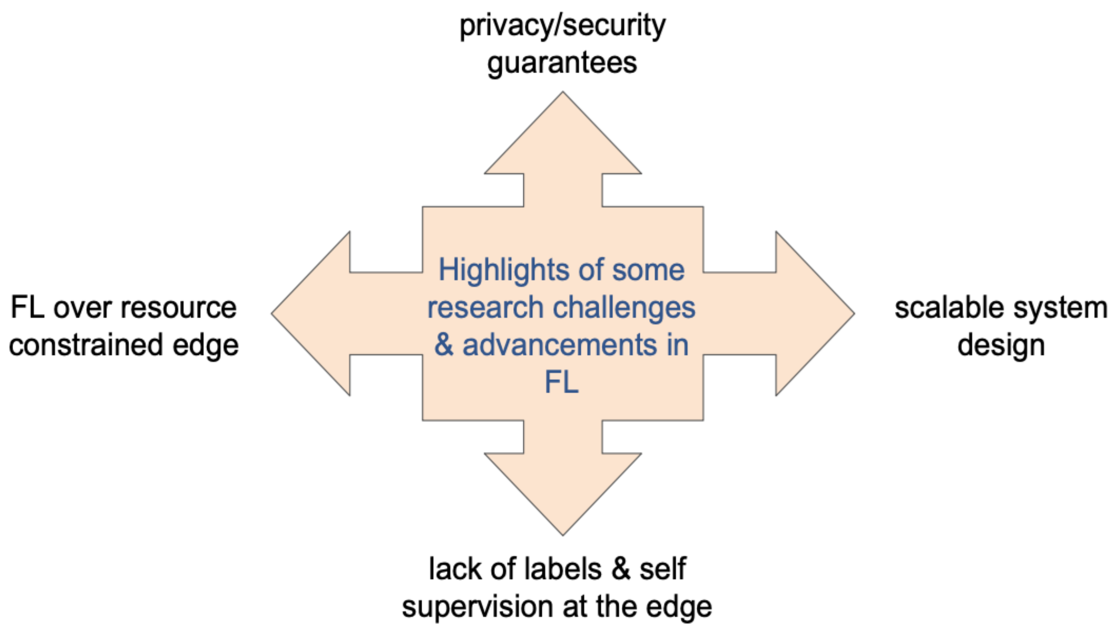
Dynamic Programming-based Resource Scheduler

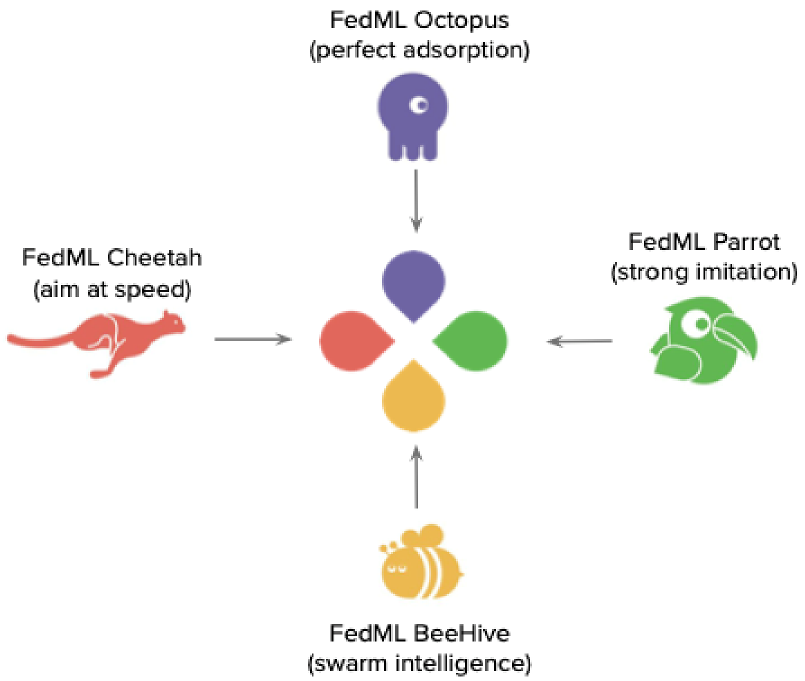
- Given G GPUs and C clients (local models), the scheduler finds the optimal G sub-sequences of the C clients.



Summary

- FL is revolutionizing the ML ecosystem by pushing learning to the 'edge'!
- FedML is a powerful platform that enables many solutions and real-world deployment.





Any Questions?

Thank you for your attention!