Edge Computing in Age of Machine Learning and Energy Constraints

Presenter: Klara Nahrstedt (klara@illinois.edu)

Siebel School of Computing and Data Science
University of Illinois Urbana-Champaign

e

ILLINOIS

Outline

- Motivation
- Federated Learning Training Description
- BoFL
  - Problem Statement
  - Solution and Evaluation
- FedCore
  - Problem Statement
  - Solution and Evaluation
- Conclusion

Large scale Machine Learning Systems

ML Training System

City-scale
Surveillance Video Analysis

CV

World-scale
Question Answering Services

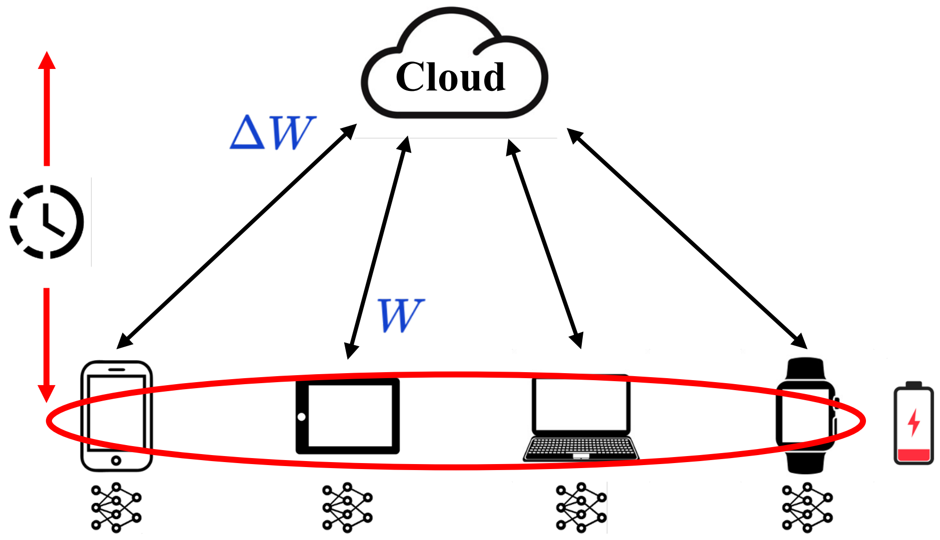
NLP

Large-scale
Healthcare Model Training

Medical AI

Multifaceted Challenges of Federated Lerning Training Systems

Federated Learning Systems (Training)



compute-intensive | **energy-consuming** | **latency-sensitive**

Multifaceted Challenges of Federated Lerning Training Systems

Federated Learning Systems

Multifaceted Resource Challenges

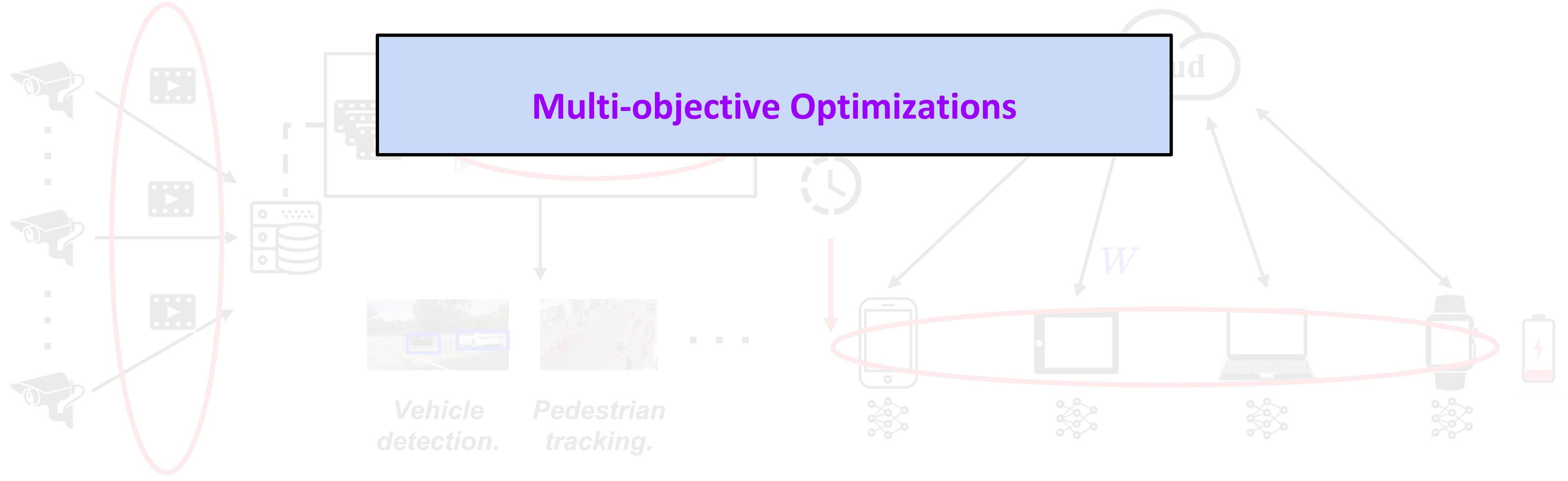
compute-intensive

energy-consuming

latency-sensitive

Multifaceted Challenges of Federated Lerning Training Systems

Federated Learning Systems



Multi-objective Optimizations

computation-throughput

energy-consumption

execution-latency

Optimization Opportunities in ML Training Systems

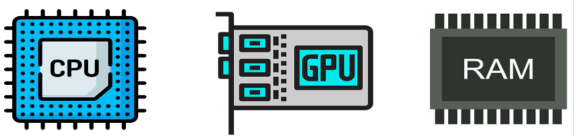
Data Redundancy

- Frame Filtering
- Video Compression
- Resolutions & Bitrates
- ...

bandwidth

throughput

Hardware Configurability

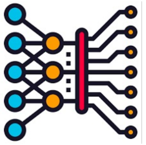


- Voltages
- Frequencies
- Heterogeneity
- ...

energy

latency

ML Model Sparsity



- Quantization
- Model Pruning
- Gradient Compression
- ...

latency

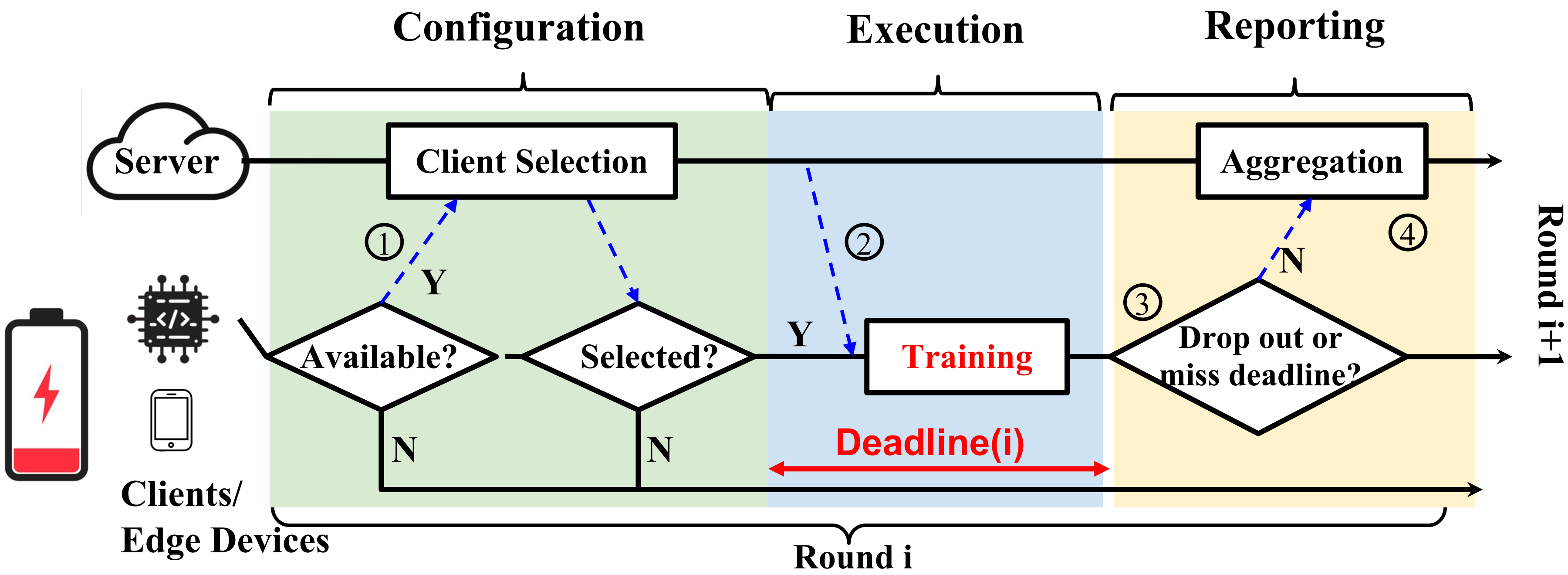
bandwidth

energy

Outline

- Motivation
- Federated Learning Training Description
- BoFL
    - Problem Statement
    - Solution and Evaluation
- FedCore
    - Problem Statement
    - Solution and Evaluation
- Conclusion

Federated Learning Workflow



① Device check-in with server; then the server selects a subset of clients

② Model and training parameters are sent to selected devices

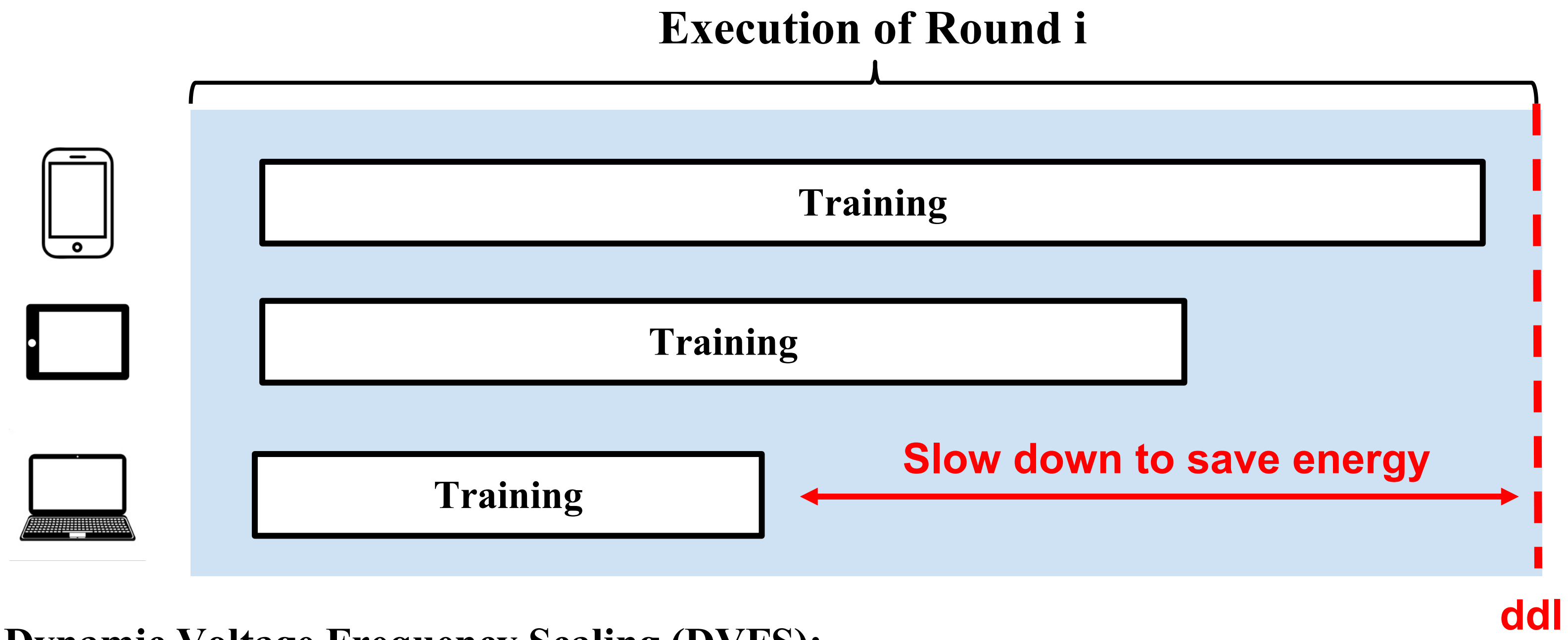
③ On-device training is executed; the model gradients are reported if training succeeds

④ Server aggregates updates into the global model; training moves to the next round

Outline

- Motivation
- Federated Learning Training Description
- BoFL
  - Problem Statement
  - Solution and Evaluation
- FedCore
  - Problem Statement
  - Solution and Evaluation
- Conclusion

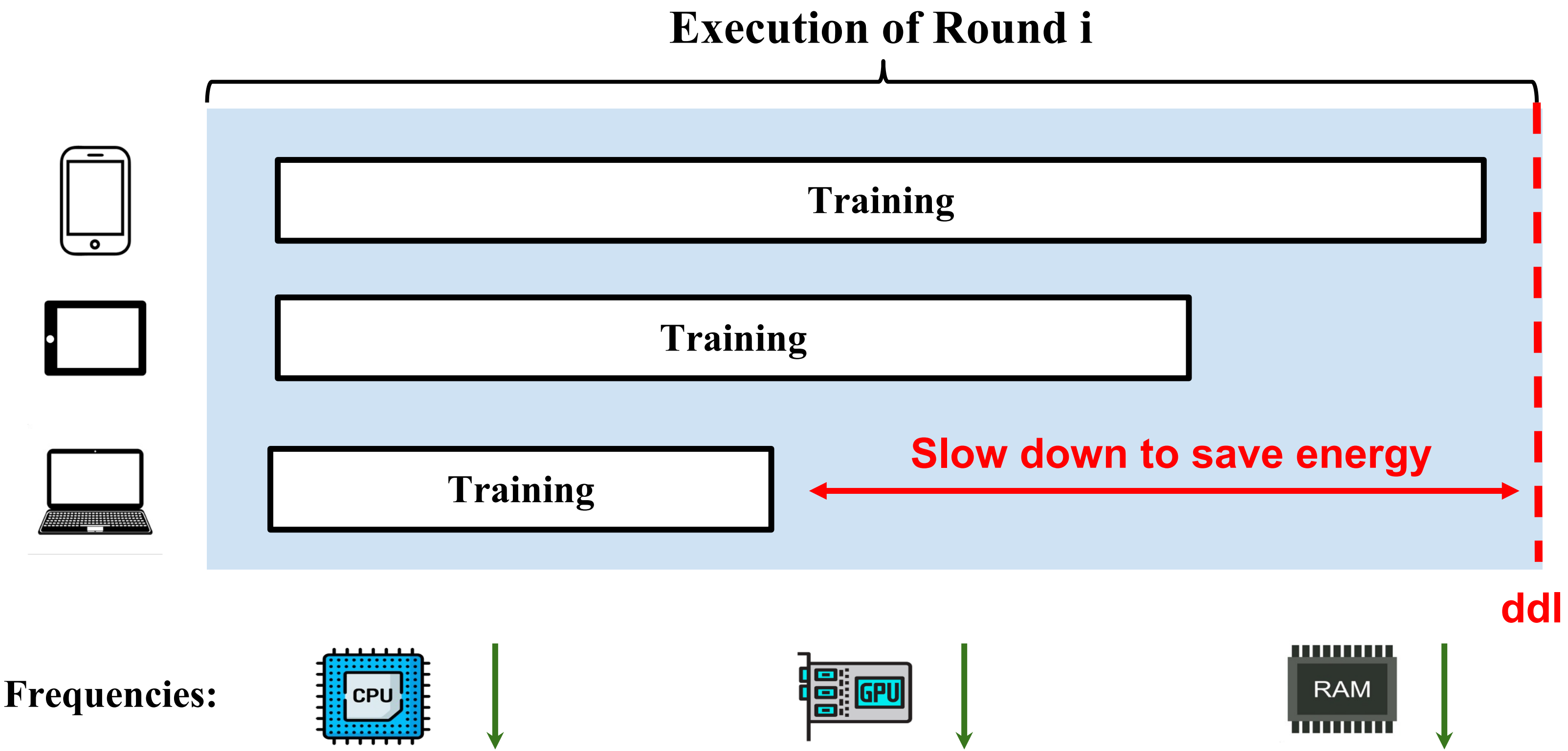
Pace Control with DVFS



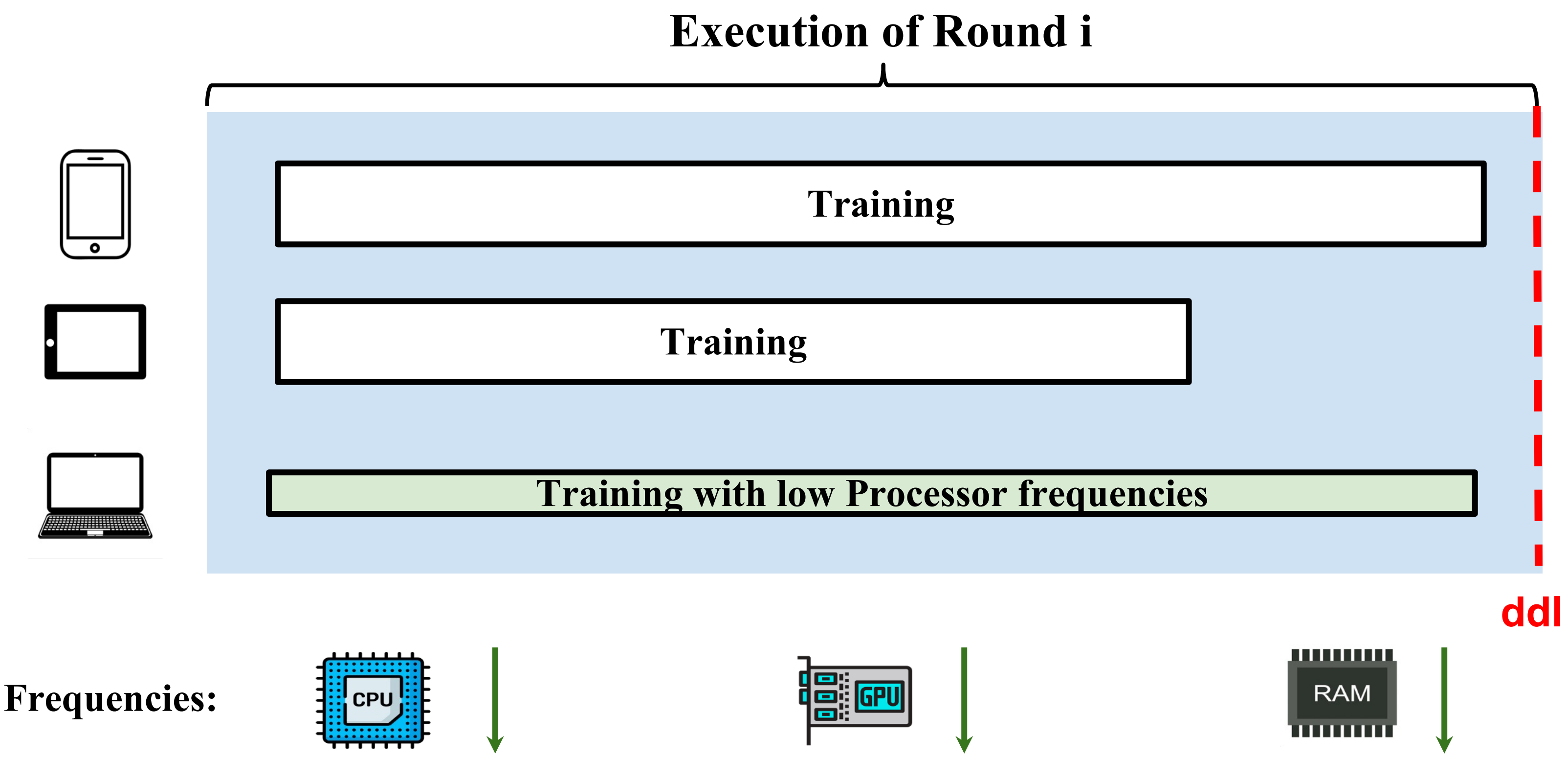
Dynamic Voltage Frequency Scaling (DVFS):
the adjustment of power and speed settings on a computing devices' various processors for power saving when those resources are not needed.

H. Guo et al., "BoFL: Bayesian optimized local training pace control for energy efficient federated learning", **ACM/IFIP Middleware '22**: 23rd ACM/IFIP International Middleware Conference, November 2022

Pace Control with DVFS



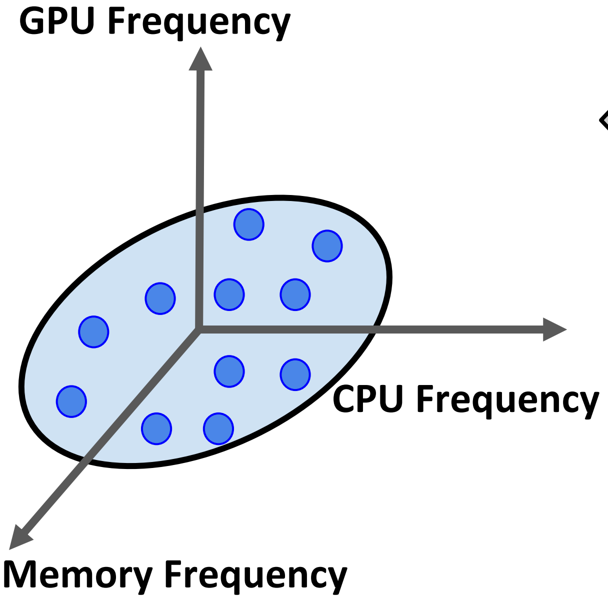
Pace Control with DVFS

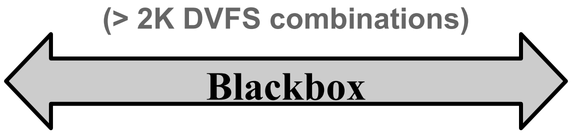


Energy consumption is reduced & Training deadline is satisfied. ✅

Finding the Best Training Pace is Challenging

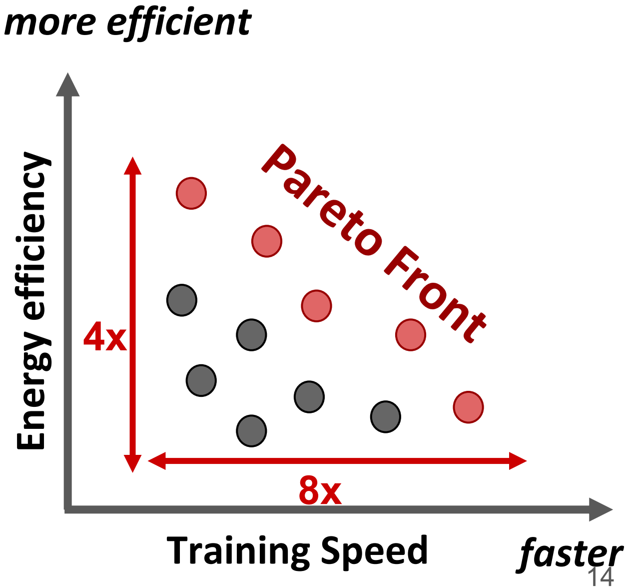
Question:
How to select the best DVFS configurations for each round of local model training?





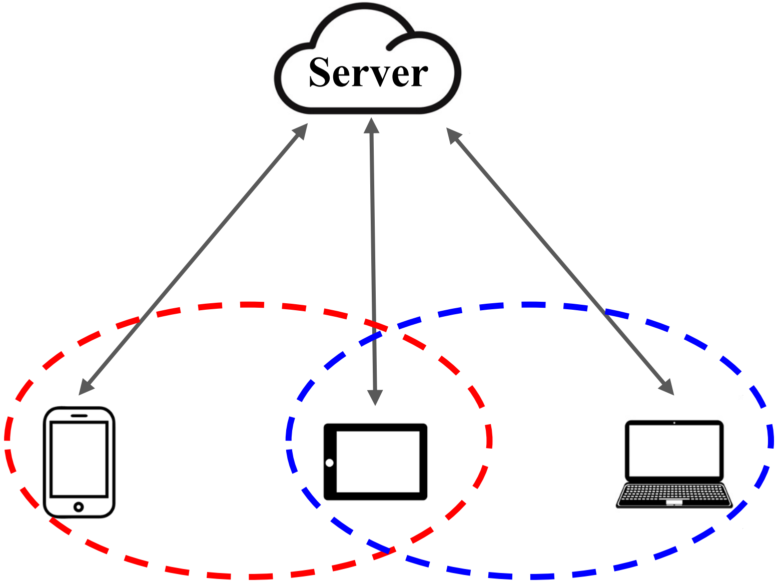
Challenges:

- Non-linearity;
- NN-model dependence;
- Hardware dependence



Finding the Best Training Pace is Challenging

Question:
How to select the best DVFS configurations for each round of local model training?

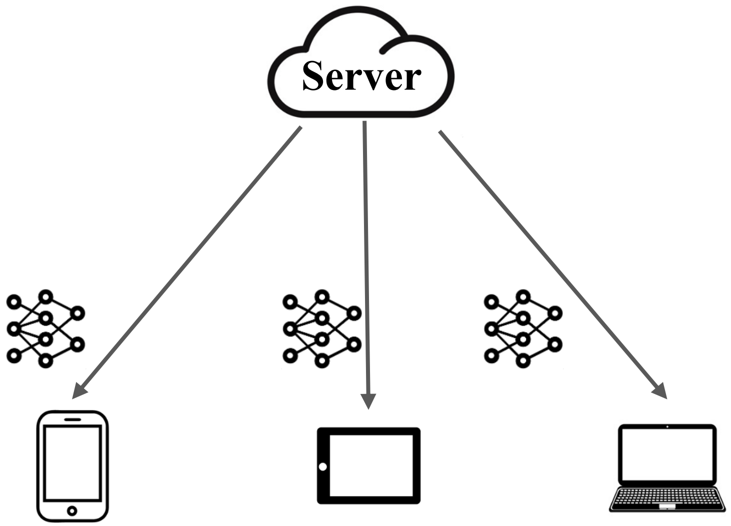


Challenges:

- **Different execution deadlines for each execution round;**

Finding the Best Training Pace is Challenging

Question:
How to select the best DVFS configurations for each round of local model training?



Challenges:

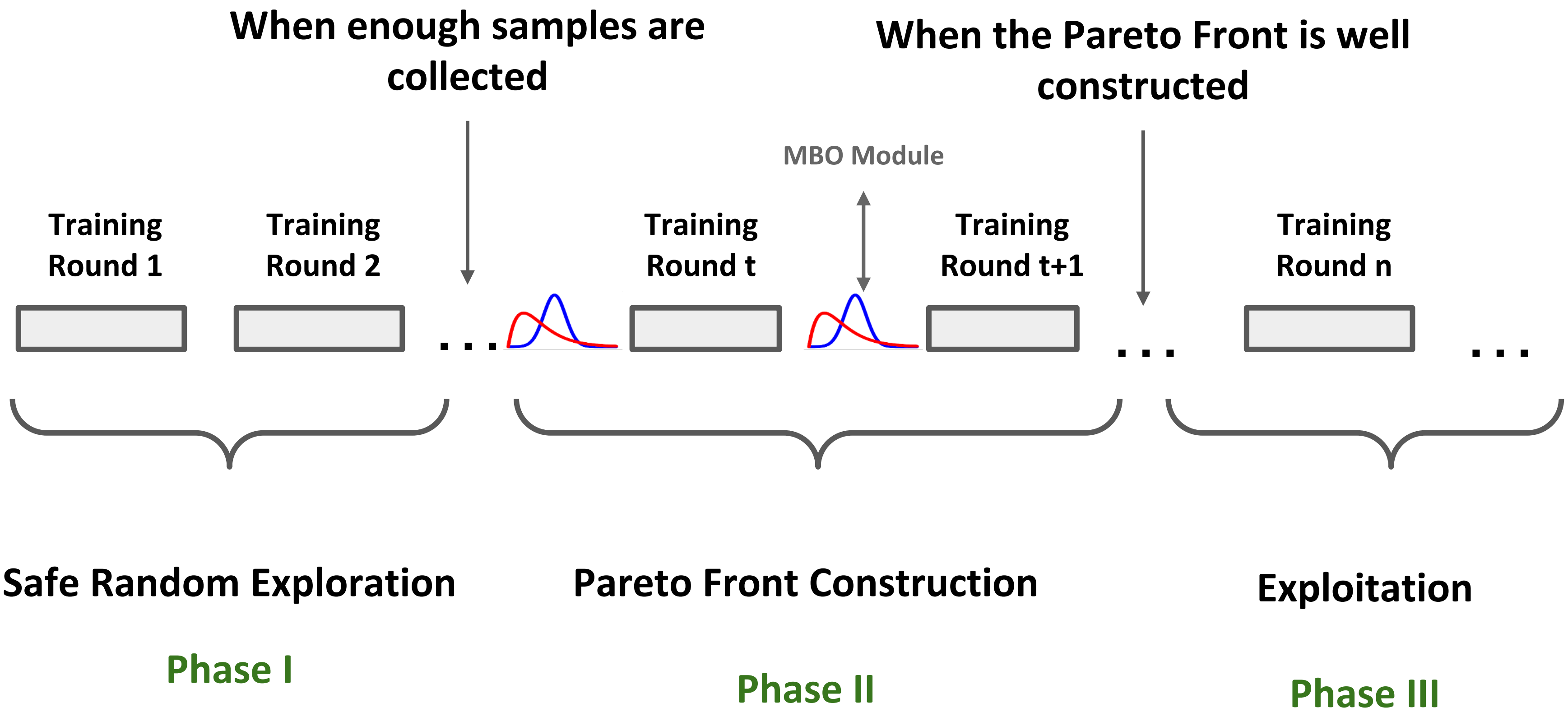
- **Different execution deadlines for each execution round;**
- **No access to the NN-model before FL task for performance profiling.**

Goal:
Find the Pareto DVFS configurations of a Blackbox optimization in an online form.

Outline

- Motivation
- Federated Learning Training Description
- BoFL
    - Problem Statement
    - Solution and Evaluation
- FedCore
    - Problem Statement
    - Solution and Evaluation
- Conclusion

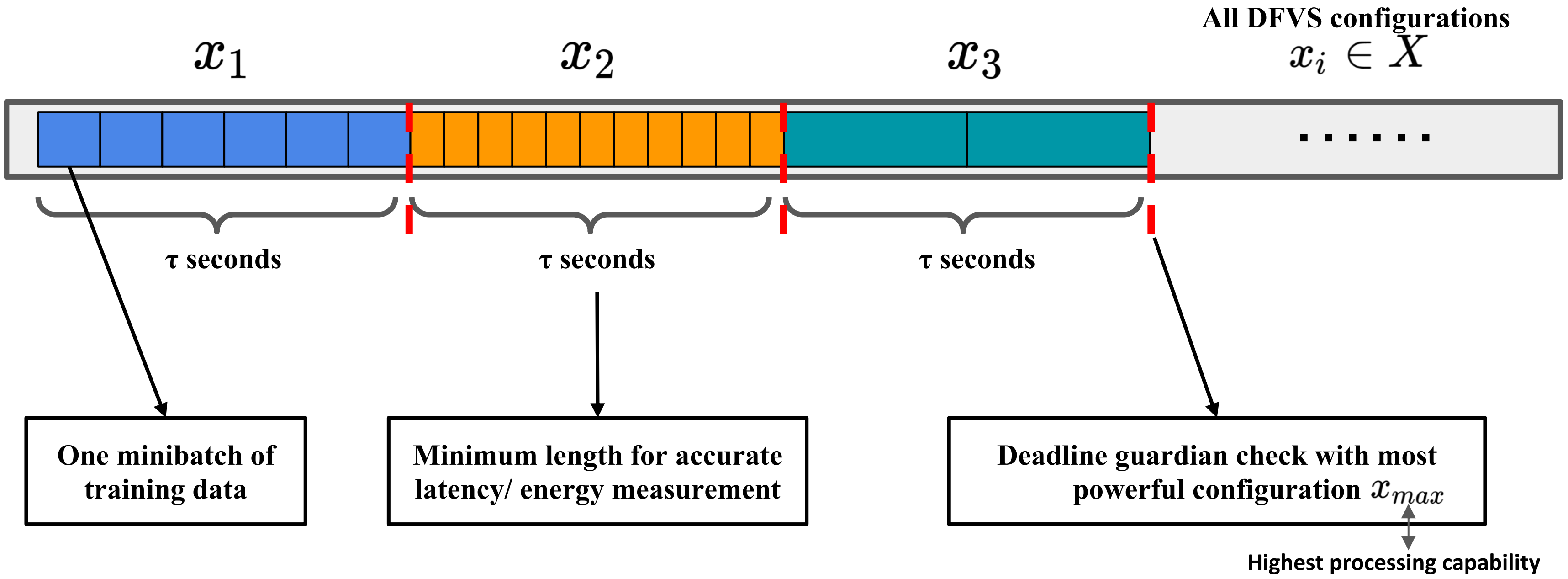
Our Solution: BoFL



BoFL: Safe Random Exploration

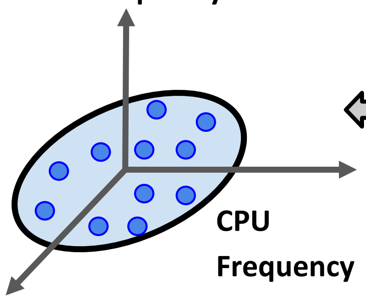
The execution length of each FL round usually take several minutes:

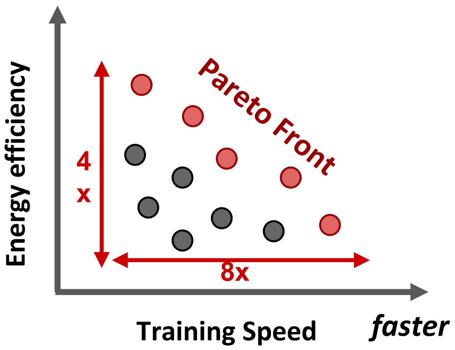
- **Try to explore as many configurations as possible;**
- **Make sure to finish all training data before the deadline.**

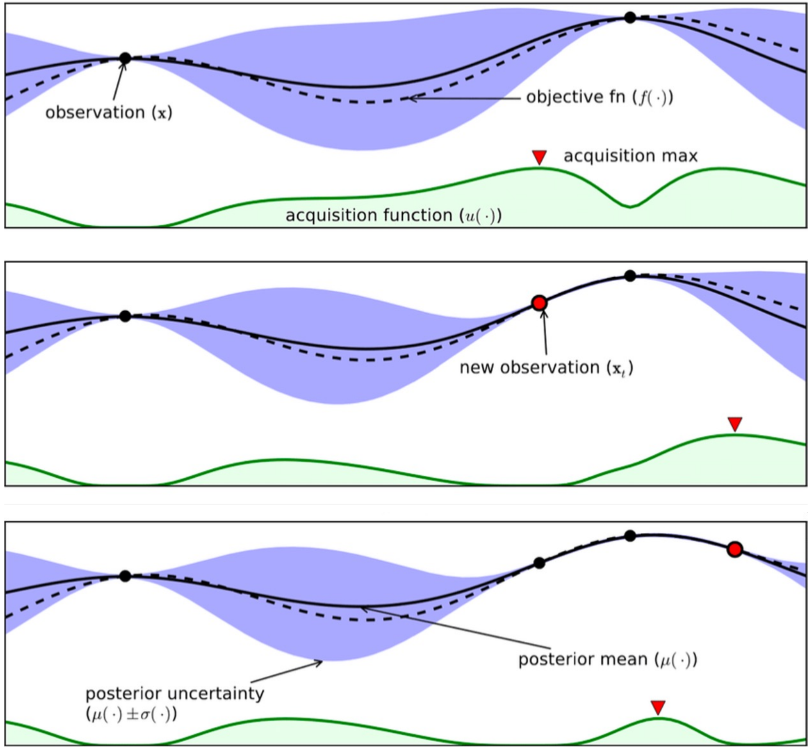


Pareto Construction with Bayesian Optimization

Bayesian optimization (BO) is a sample-efficient methodology for optimizing expensive-to-evaluate black-box functions.





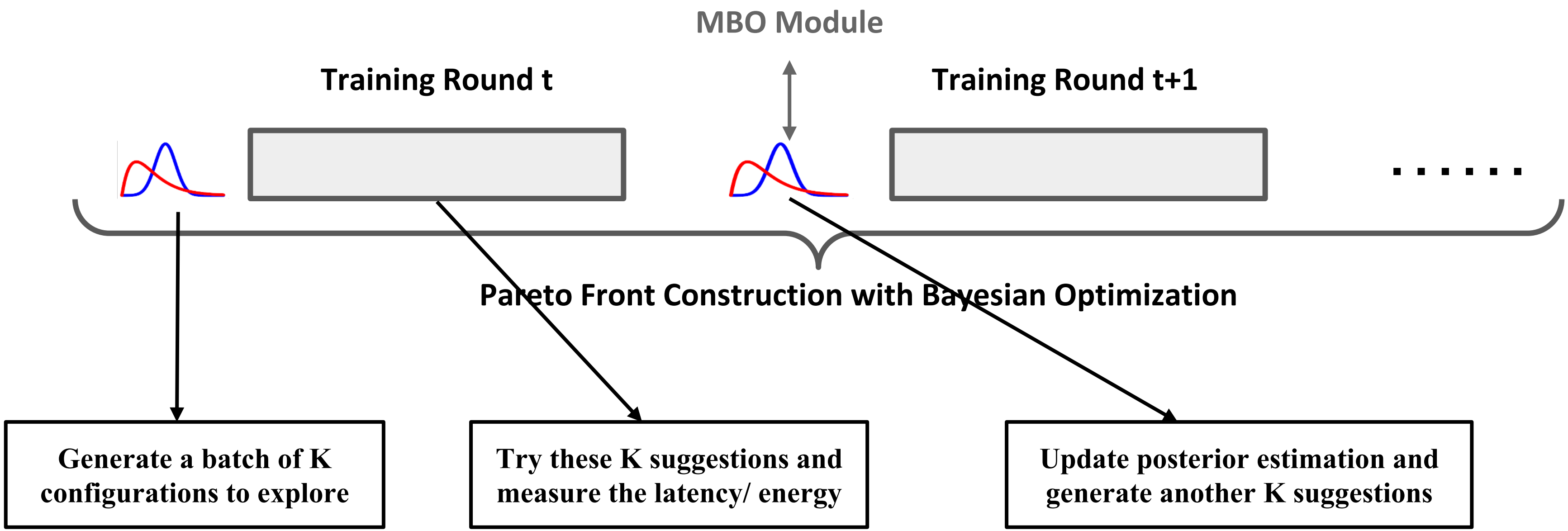


Multi-Objective Bayesian optimization (MBO)

BoFL: Pareto Front Construction

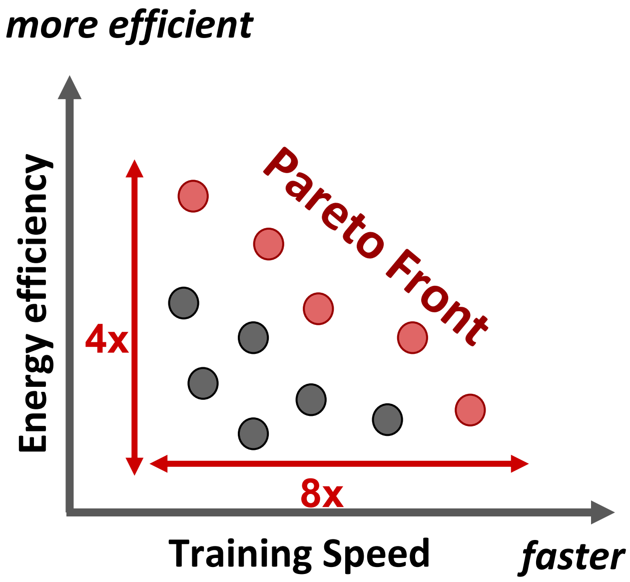
The execution length of each FL round usually takes several minutes:

- **Generate batched exploration suggestions.**



BoFL: Exploitation

After *Pareto Front Construction Phase*.



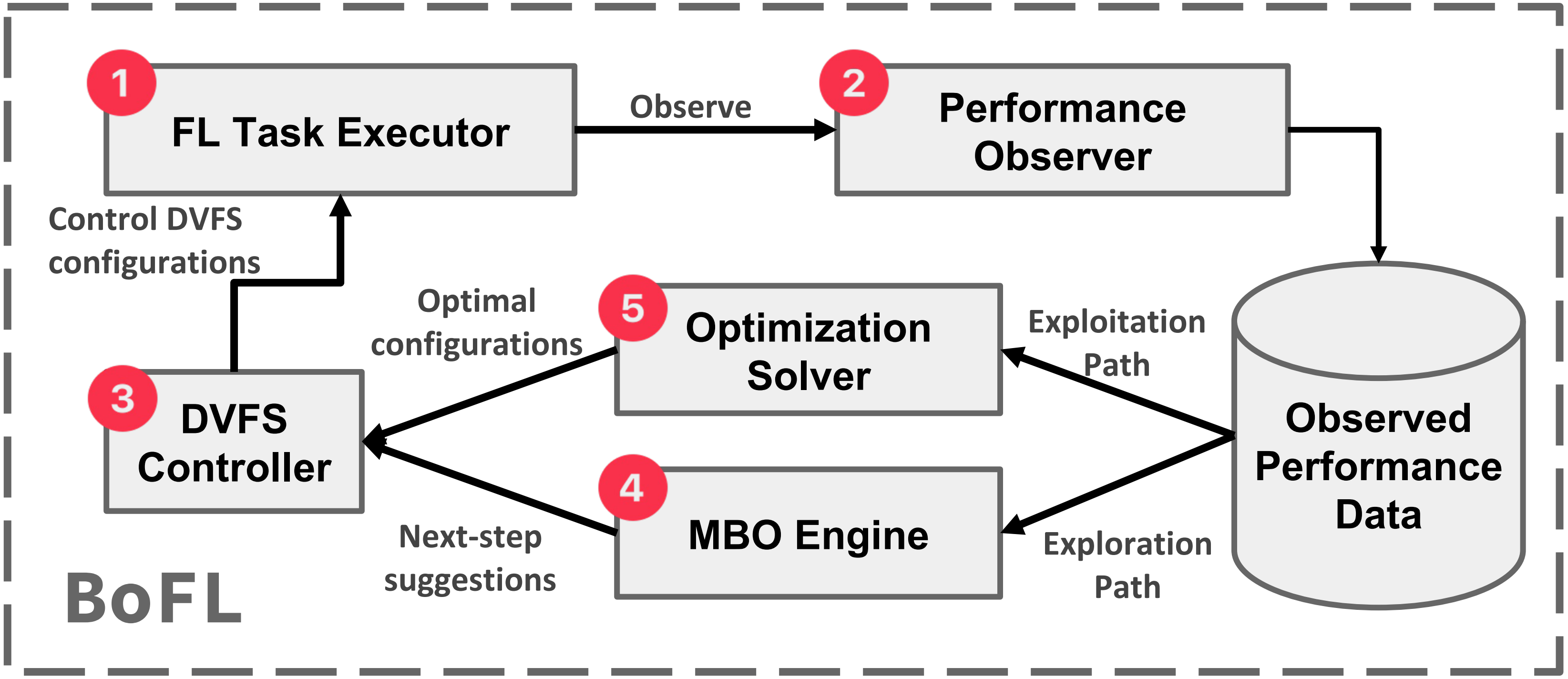
Select configurations from the Pareto Front:

To Minimize energy consumption;

S.T. DDLs are not missed;

An ILP problem that can be efficiently solved

BoFL Architecture



BoFL Evaluation

Hardware Testbeds:

	Jetson AGX	Jetson TX2
CPU	8-core ARM v8.2	2-core Nvidia Denver2 + 4-core ARM Cortex-A57
Frequencies	**0.42GHz → 2.26GHz (25 steps)**	**0.34GHz → 2.03GHz (12 Steps)**
GPU	512-core Volta GPU	256-core Pascal GPU
Frequencies	**0.11GHz → 1.38GHz (14 steps)**	**0.11GHz → 1.30GHz (13 steps)**
Memory	32GB 256-bit LPDDR4x	8GB 128-bit LPDDR4
Frequencies	**0.20GHz → 2.13GHz (6 steps)**	**0.41GHz → 1.87GHz (6 steps)**

Jetson AGX

Jetson TX2

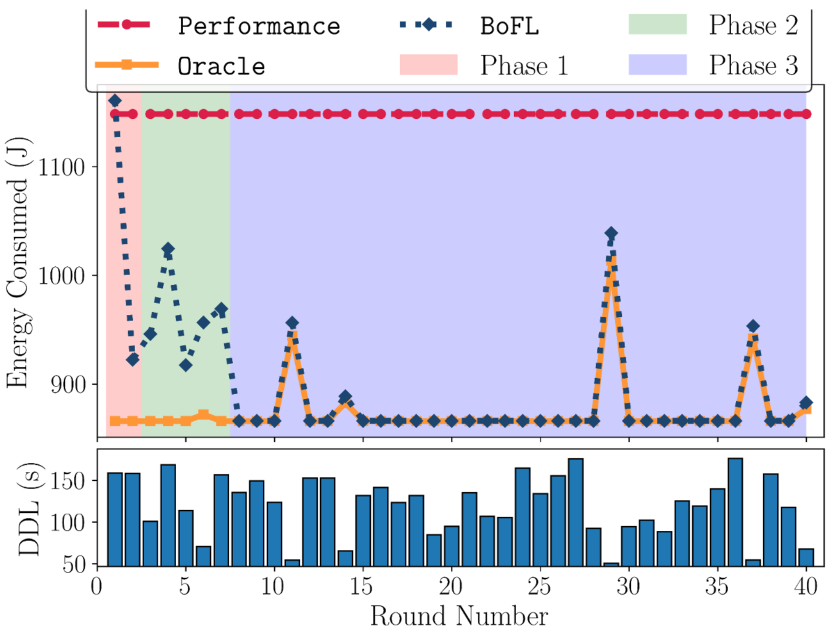
Federated Learning Tasks:

3 different FL task of 100 rounds:

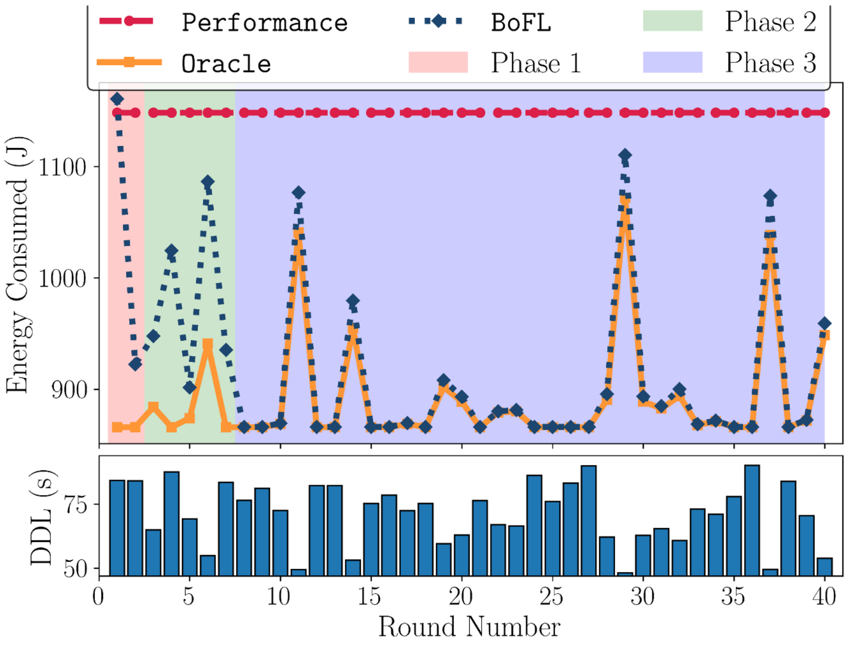
Datasets	**NN-Model**
CIFAR10	Vision-Transformer
ImageNet	ResNet-50
IMDB	LSTM

Evaluation of Energy Efficiency

(AGX, ImageNet-ResNet50)



$$ddl_{max} = 4 \times ddl_{min}$$

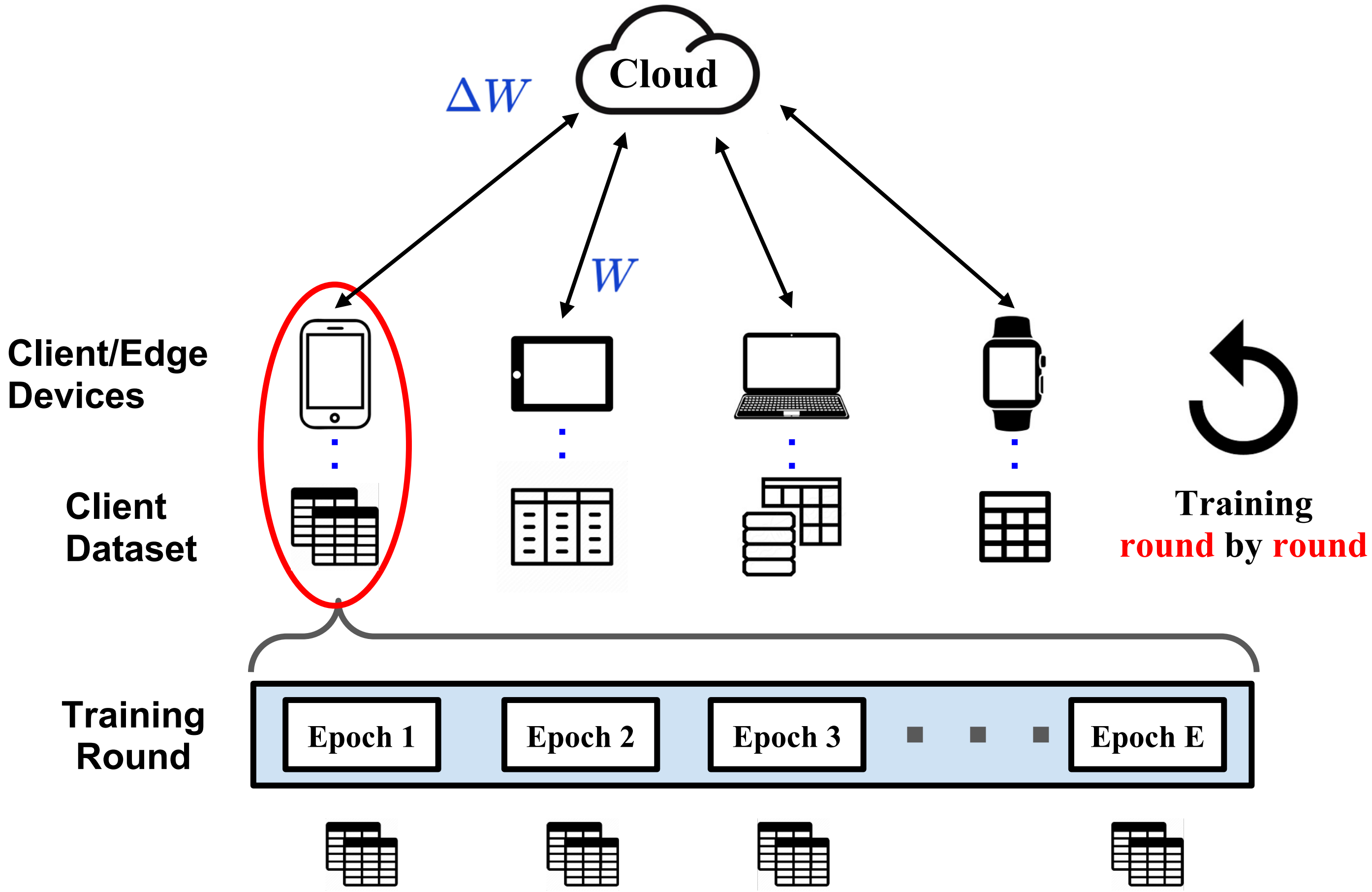


$$ddl_{max} = 2 \times ddl_{min}$$

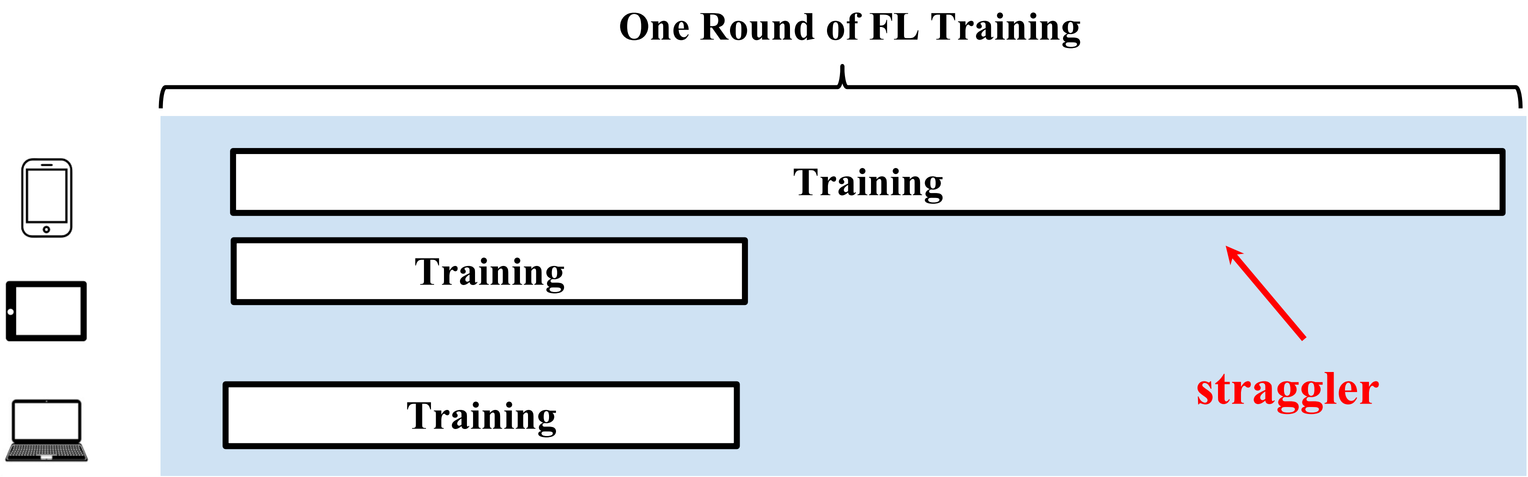
Outline

- Motivation
- Federated Learning Training Description
- BoFL
    - Problem Statement
    - Solution and Evaluation
- FedCore
    - Problem Statement
    - Solution and Evaluation
- Conclusion

Federated Learning System



Straggler Effect in FL Systems

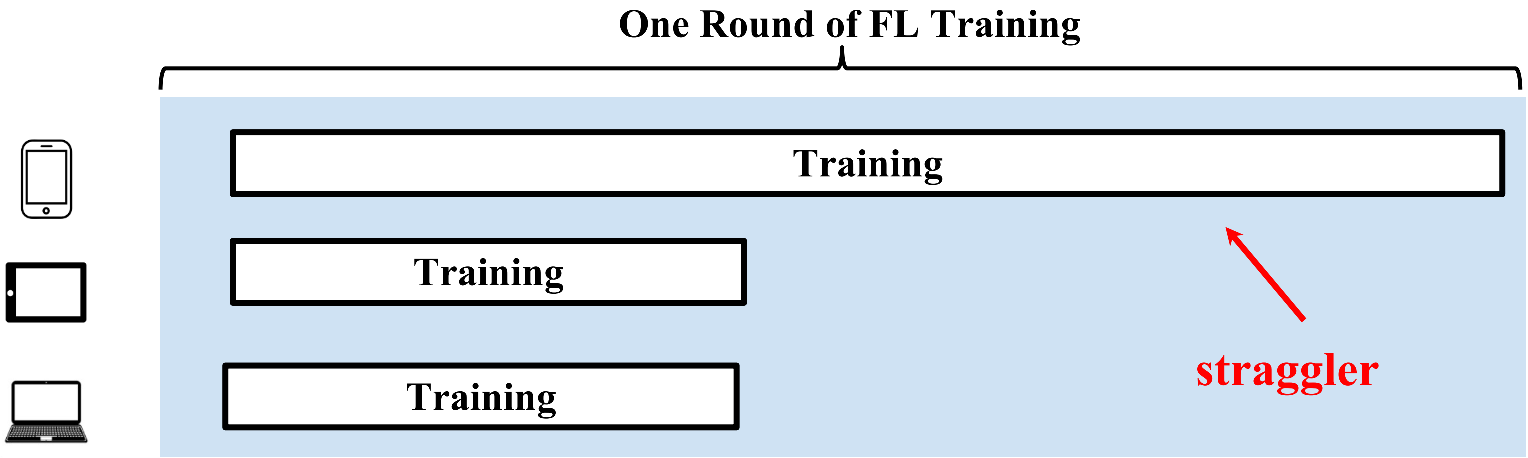


Straggler Problem in Meta's FL systems:

Meta's million-client FL system, Papaya, demonstrated that per-client training time distribution spans over **two orders of magnitude**, and the round completion time is **21x** larger than the average training time due to stragglers' delays.

H. Guo et al. "FedCore: Straggler-Free Federated Learning with Distributed Coresets", **IEEE International Conference on Communications (ICC) 2024**, June 2024.

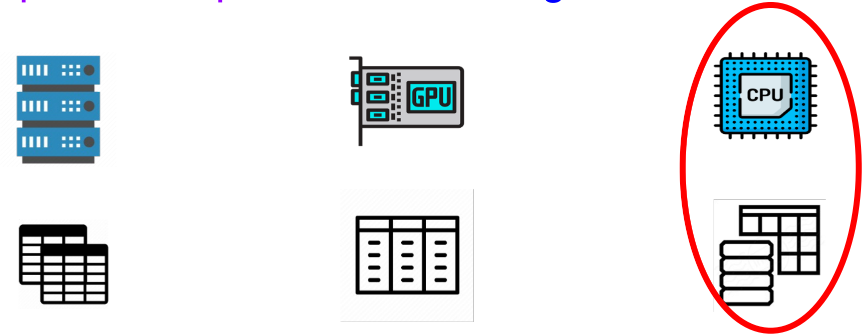
Root Causes of Straggler Problem



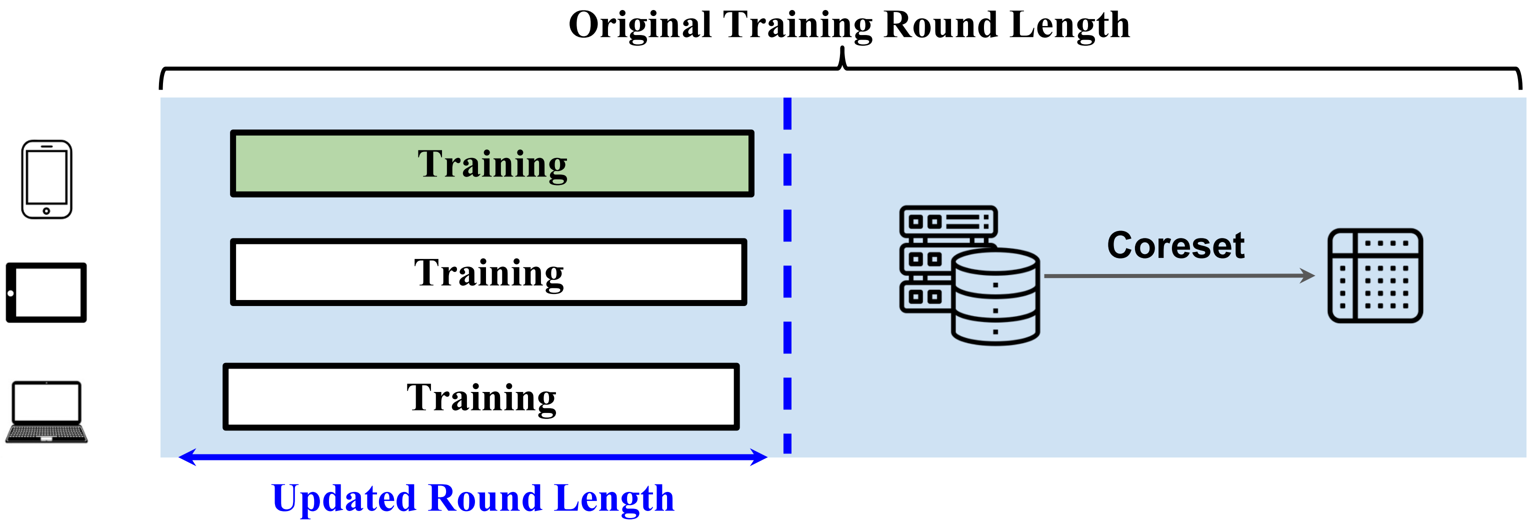
Mismatch between clients' computational power and training data size.

❌ **Hardware:**

✅ **Data Size:**



Straggler Free FL with Training Coresets

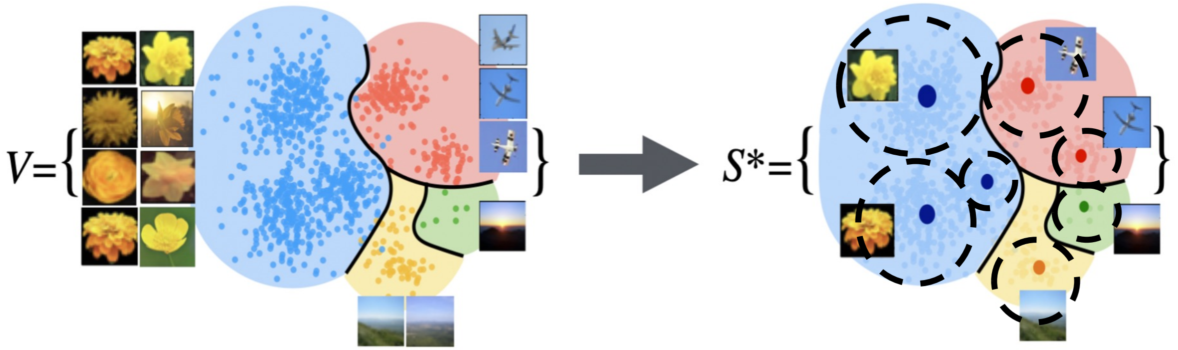


Data Efficient FL with Coreset:

A subset **S** of the whole Training dataset **V**, where the ML model trained on **S**, i.e., θ_s, has similar performance as the model trained on the whole dataset, i.e., θ_v.

Challenges & Solutions

Challenge #1:
How to select data samples that best represent the whole dataset?

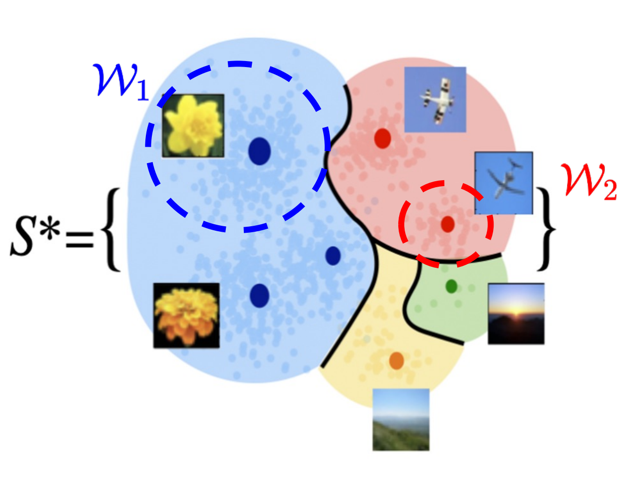


Solutions:

- **Cluster training samples based on their per-sample gradient similarities;**
- **Form a Coreset using the cluster centroids.**

Challenges & Solutions

Challenge #1:
How to select data samples that best represent the whole dataset?



Solutions:

- **The coreset is a weighted collection of the cluster centers. Sample weights equal to the corresponding cluster sizes.**
- **The federated gradient update will be a weighted sum of the coreset gradients.**

Challenges & Solutions

Challenge #2:
How to adaptively update the Coreset while the model parameters are envolving?

Round 1 **Round 2** **Round 3** ... **Round N**

$$\underbrace{S_1 \qquad S_2 \qquad S_3 \qquad \cdots \qquad S_N}$$

Different Coresets for each round while model is updating.

Solutions:

- **Generating per-sample gradients over full-set periodically.**
- **Update Coreset with the updated gradients.**

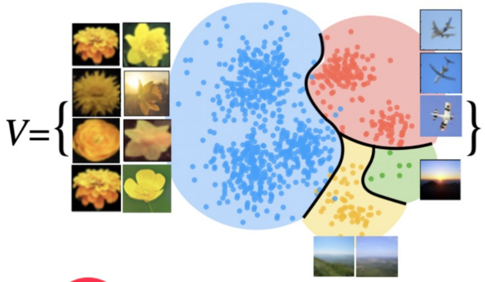
Outline

- Motivation
- Federated Learning Training Description
- BoFL
  - Problem Statement
  - Solution and Evaluation
- FedCore
  - Problem Statement
  - Solution and Evaluation
- Conclusion

FedCore System Overview

In Each Round of FL Training

Full-set Training Epoch

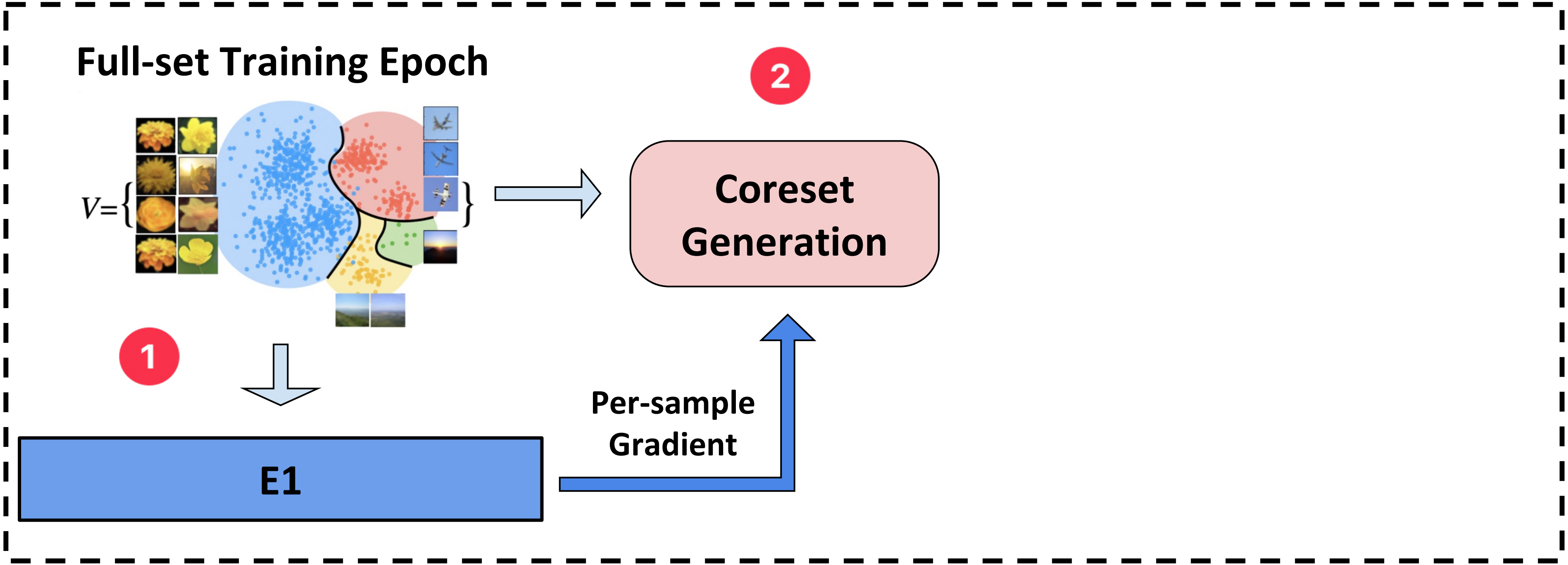


1

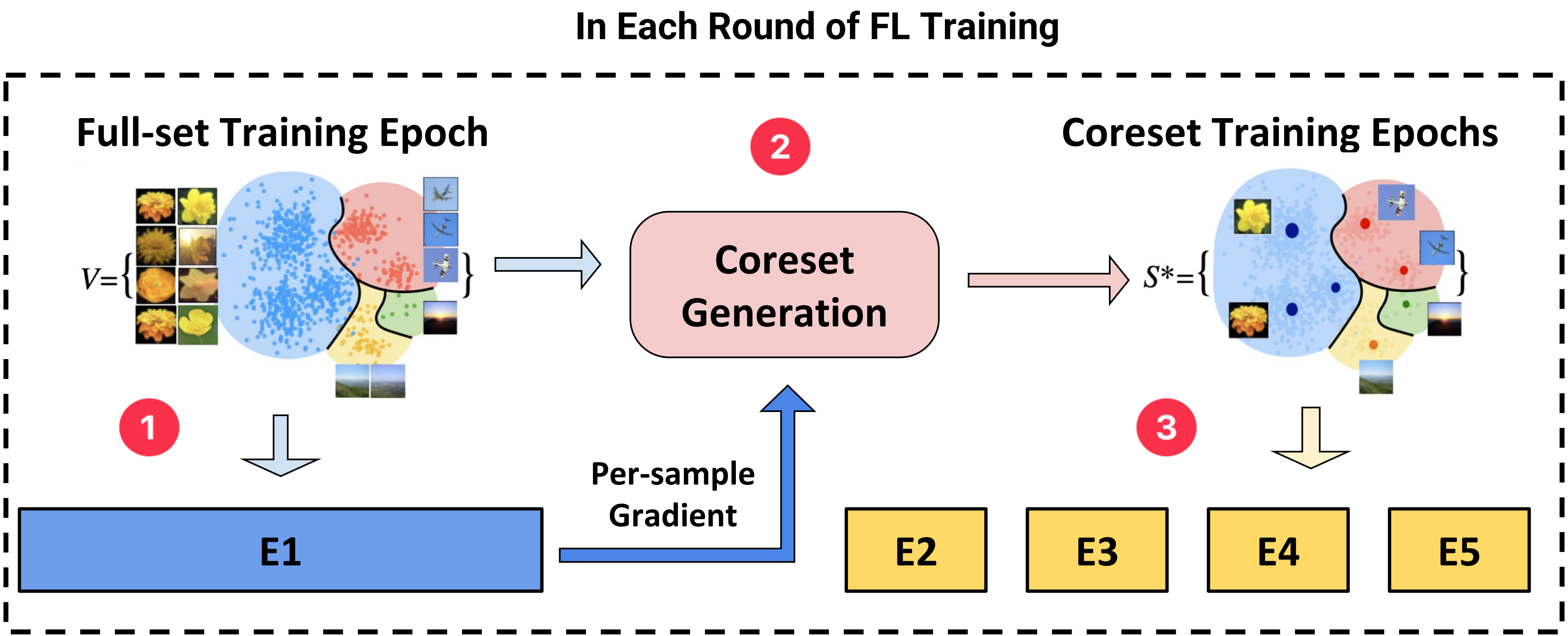
E1

FedCore System Overview

In Each Round of FL Training



FedCore System Overview



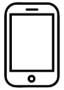
Coresets Generation Via Optimization

OPT: ***K-Medoids Clustering*** in the gradient space.

$$\min_{S^* \subseteq V} \sum_{i \in \underbrace{V}_{\text{full-set}}} \min_{j \in \underbrace{S^*}_{\text{coreset}}} \underbrace{d(i,\, j)}_{\text{gradient distance}}$$

$$\text{st.},\ |S^*| \le \underbrace{\text{threshold}}_{\text{hardware \& round length}}$$

Target round length: 5 mins



[50 samples / min]

[300 samples / min]

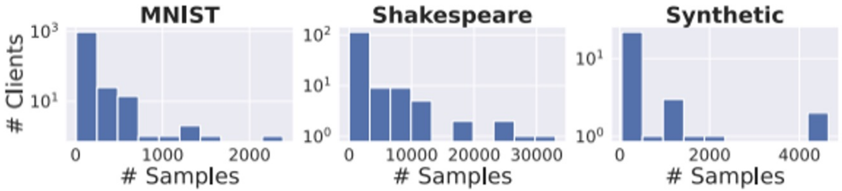
Thresholds: 250

1500

FedCore Evaluation

Statistics of the Evaluation Benchmarks:

Dataset	Clients	Samples	Samples / Client	
			mean	std
MNIST	1,000	69,035	69	106
Shakespare	143	517,106	3,616	6,808
Synthetic	30	20,101	670	1,148



Distribution of training samples per client

Comparison Baselines:

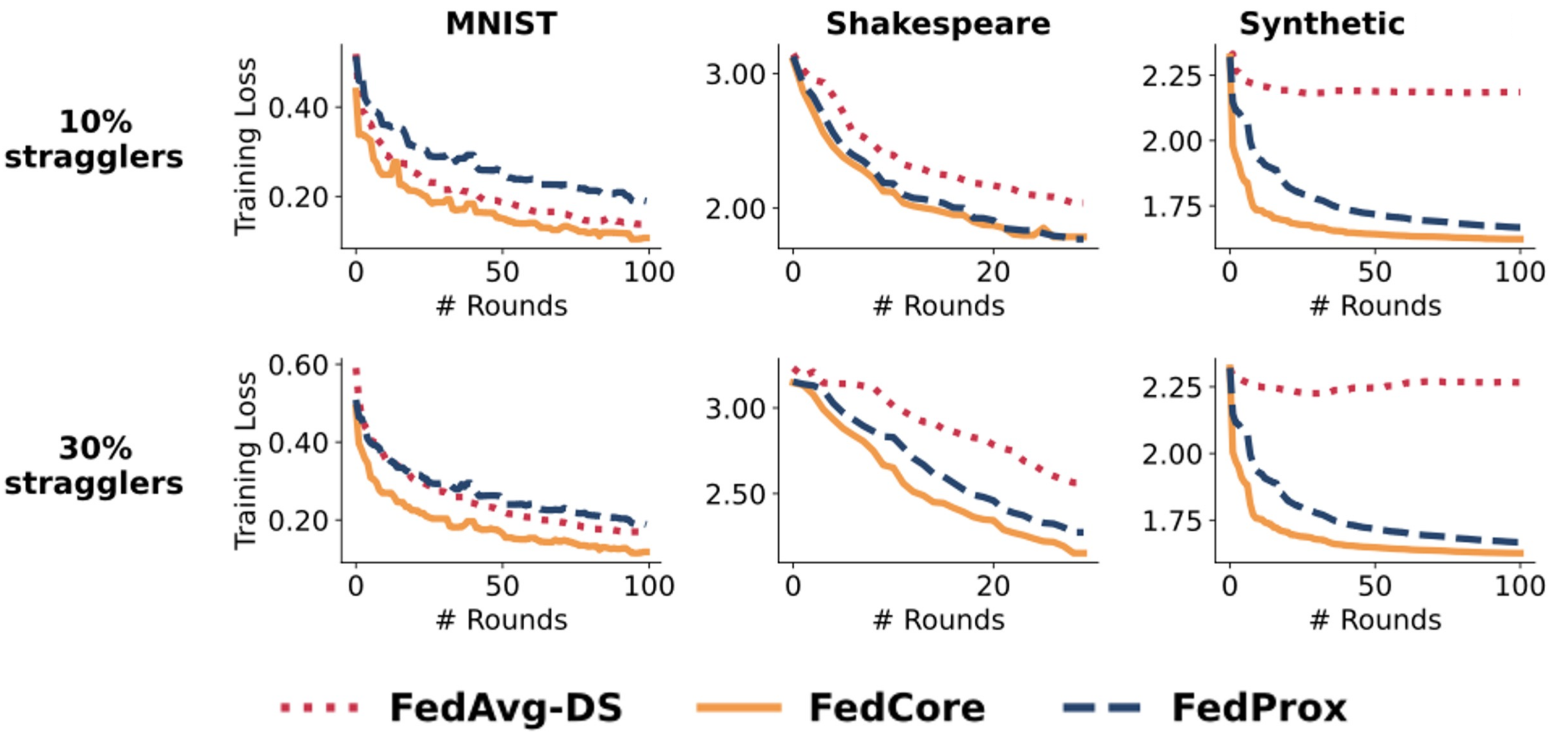
- **FedAvg [1]:** the vanilla FL algorithm without stagger prevention;
- **FedAvg-DS:** deadline sensitive version of FedAvg, drop all the stragglers;
- **FedProx [2]:** handles partial training results from stragglers that may finish less epochs before the deadline.

Implementation:

https://github.com/hongpeng-guo/FedCore

[1] McMahan, Brendan, et al. "Communication-efficient learning of deep networks from decentralized data." *Artificial intelligence and statistics*. PMLR, 2017.
[2] Li, Tian, et al. "Federated optimization in heterogeneous networks." *Proceedings of Machine learning and systems,* 2020.

Evaluation of Training Loss



Evaluation of Accuracy & Training Time

FedCore increases FL training speed by up to 8x without loss of model accuracy

		MNIST		Shakespeare		Synthetic (1, 1)	
		10%	30%	10%	30%	10%	30%
Test Accuracy	FedAvg	94.7		44.9		71.8	
	FedAvg-DS	94.1	93.1	39.0	25.2	23.0	19.9
	FedProx	92.6	92.7	44.1	31.3	**72.3**	72.2
	FedCore	**94.6**	**94.5**	**44.7**	**34.8**	72.2	**72.8**
Mean Training Time per Round (normalized)	FedAvg	3.27	8.48	1.38	4.09	1.37	4.80
	FedAvg-DS	0.94	0.95	0.60	0.67	0.69	0.79
	FedProx	0.98	0.99	0.85	0.94	0.86	0.95
	FedCore	0.99	0.99	0.90	0.99	0.93	0.99

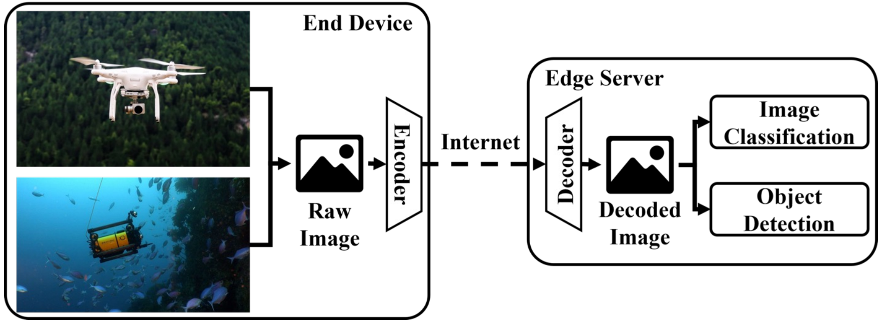
Comparison of test accuracy and training time for FedCore and the Baselines at 10% and 30% stragglers.

Bold: top accuracy; Red: exceeded deadline. Normalized time of 1 is round deadline.

Outline

- Motivation
- Federated Learning Training Description
- BoFL
    - Problem Statement
    - Solution and Evaluation
- FedCore
    - Problem Statement
    - Solution and Evaluation
- Conclusion

Conclusion – Other Edge Apps - Visual Analytics

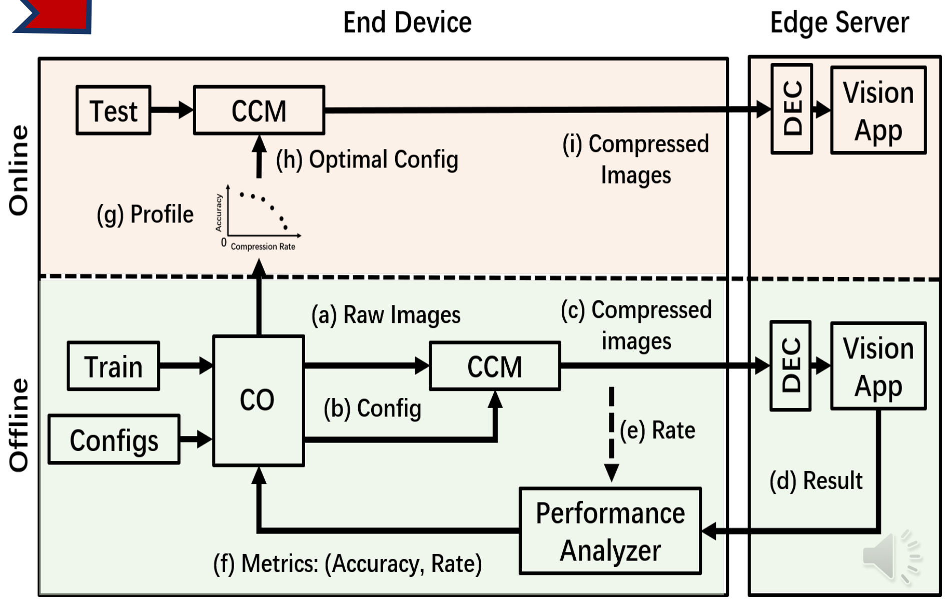


Efficient image compression is critical.

Two Stages:

- **Offline Profiling**: CO interacts with CCM and vision app to derive profile (g)
- **Online Compression**: CCM selects optimal configuration (h) from profile (g) based on bandwidth condition or accuracy requirement

- **Contextualized Compression Module** (CCM): compression based on context
- **Compression Optimizer** (CO): derive a profile assisting CCM



B. Chen, Z. Yan, K. Nahrstedt "Context-aware Image Compression Optimization for Visual Analytics Offloading", ACM MMSys 2022, **Best Paper Award**

Acknowledgement

Joint work: Hongpeng Guo[1], Haotian Gu[2], Zhe Yang[1], Xiaoyang Wang[1], Eun Kyung Lee[3], Nandhini Chandramoorthy[3], Tamar Eilam[3], Deming Chen[1]

Funding: IBM Illinois Discovery Accelerator Institute (IIDAI), 2021-2031 at University of Illinois Urbana-Champaign

Publications:

H. Guo et al., "BoFL: Bayesian optimized local training pace control for energy efficient federated learning", **ACM/IFIP Middleware '22**: 23rd ACM/IFIP International Middleware Conference, November 2022,

H. Guo et al., "FedCore: Straggler-Free Federated Learning with Distributed Coresets", **IEEE International Conference on Communications (ICC) 2024**, June 2024.

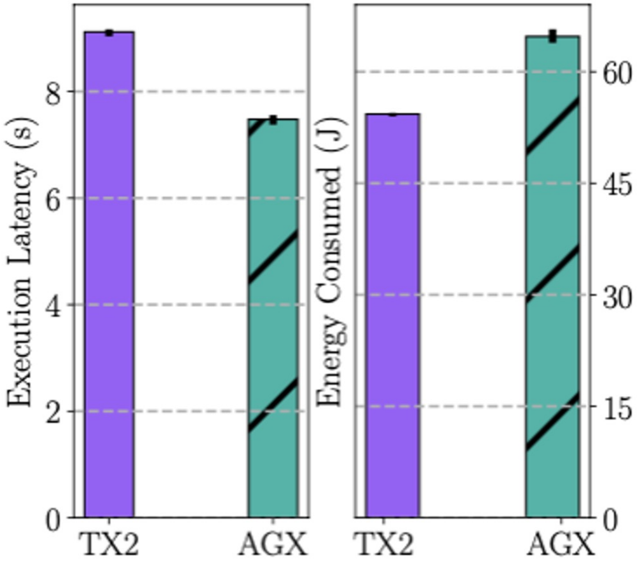
ILLINOIS [1]

Berkeley UNIVERSITY OF CALIFORNIA [2]

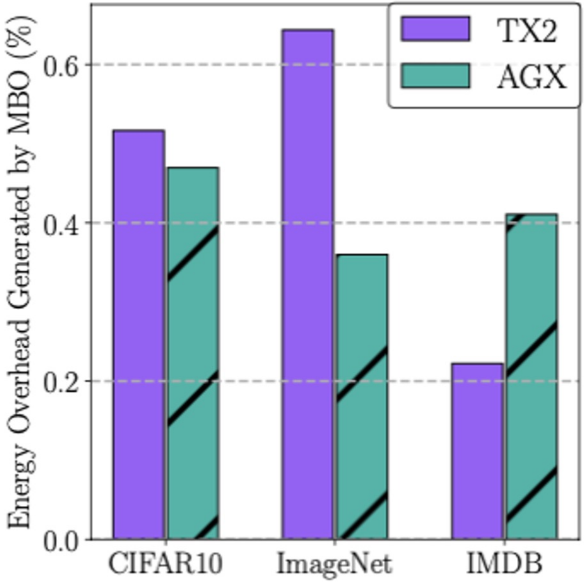
IBM Research [3]

Additional Slides

BoFL Evaluation of System Overhead



(a) MBO overhead per round.



(b) Overall energy overhead.

FedCore Coresets Generation Via Optimization

OPT: ***K-Medoids Clustering*** in the gradient space.

$$\min_{S^* \subseteq V} \sum_{i \in \underbrace{V}_{\text{full-set}}} \min_{j \in \underbrace{S^*}_{\text{coreset}}} \underbrace{d(i, j)}_{\text{gradient distance}}$$

$$\text{st.}, \; |S^*| \leq \underbrace{\text{threshold}}_{\text{hardware \& round length}}$$

- OPT is applied to each client to create distributed coresets.
- OPT can be solved with ***FasterPAM*** [1]. The cluster centers form a coreset.

If $\|\text{OPT}\| \leq \epsilon$ holds for every client, every round:

$$\mathbb{E}[\mathcal{L}(w_{fedcore}) - \mathcal{L}(w_*)] \leq \mathcal{O}(\epsilon) + \mathcal{O}\left(1/\underbrace{\mathcal{R}}_{training\ rounds}\right)$$

[1] Erich Schubert and Peter J Rousseeuw. Faster k-medoids clustering: improving the pam, clara, and clarans algorithms. In International Conference on Similarity Search and Applications, 2019.