




Federated Learning and Democratized Learning

Choong Seon Hong

Professor, Department of Computer Science
and Engineering, Kyung Hee University



- Introduction: Motivation of Federated Learning
- Federated Learning
 - FL Formulation
 - FL Algorithms
 - Ongoing Research Problems
 - Federated Learning: at the Edge
 - Summary
- Democratized Learning
 - Introduction
 - Key Components
 - Ongoing Research Problems
- Multimodal Federated Learning
 - Introduction
 - Key Components
 - Ongoing Research Problems

“Data is the New Oil”



Data is born decentralized

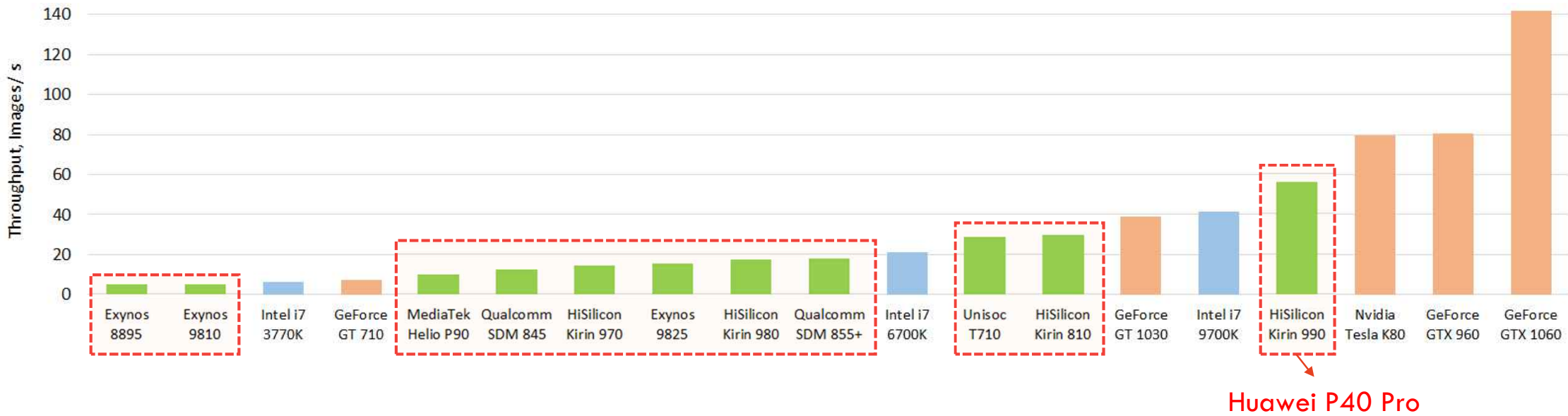
- Billions of phones & IoT devices constantly generate data
- Data enables better products and smarter models

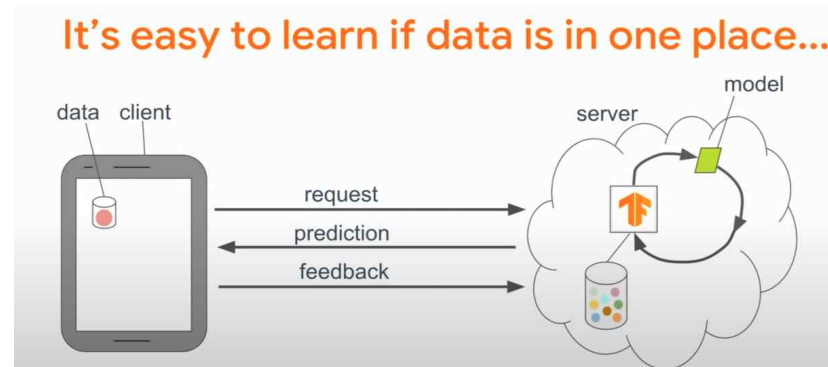
Data processing is moving on device

- Improved latency
- Works offline
- Better battery life
- Privacy advantages
- AI-powered mobile processors

Performance Review of all Mobile SoCs with AI capabilities

The performance of **mobile AI accelerators** has been evolving rapidly in the recent years, **nearly doubling with each new generation of SoCs**. The current **4th generation of mobile Neural Processing Units (NPUs)** is already approaching the results of CUDA-compatible Nvidia graphics cards presented not long ago.





Don't use data to improve products and services

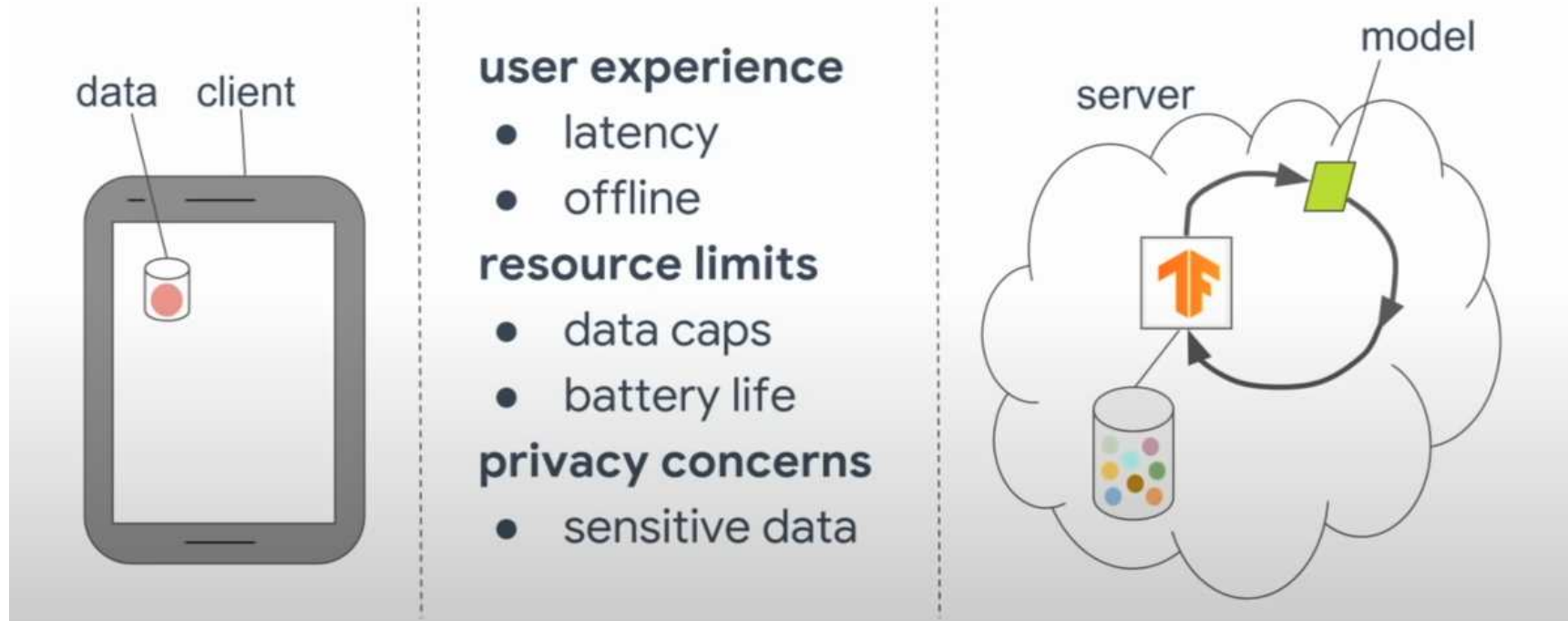
Log data centrally

Centralized data analysis and learning

Federated analysis and learning



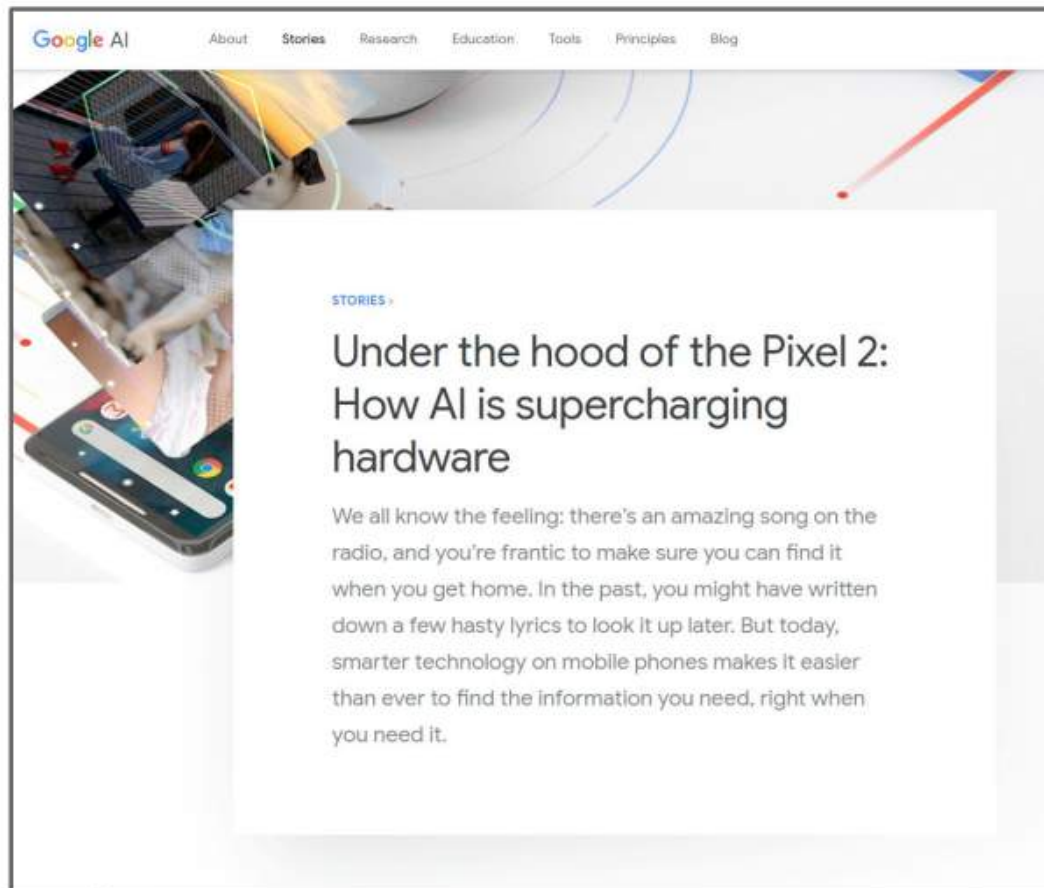
...but centralization has disadvantages





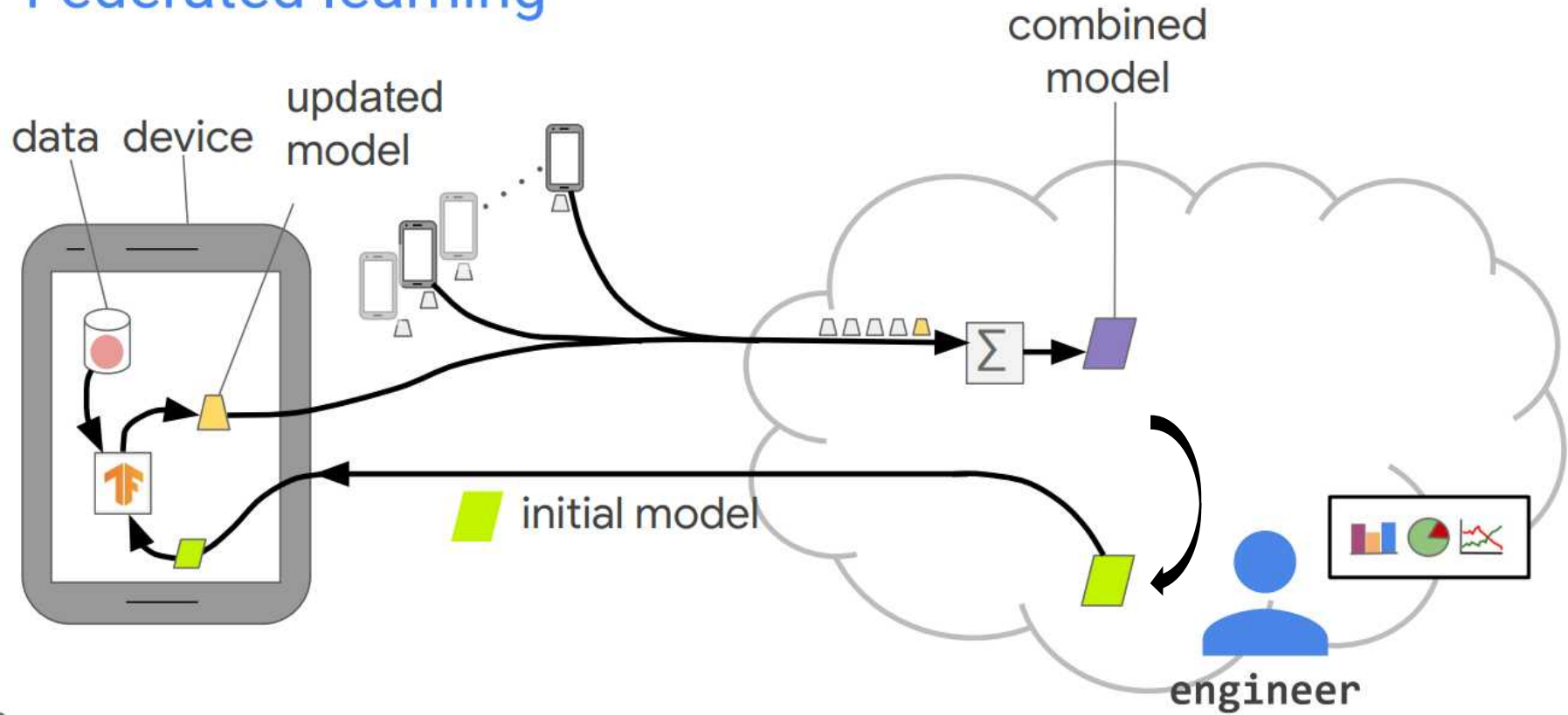
<https://youtu.be/gbRJPa9d-VU>

Federated Learning on Pixel Phones



- Federated Learning are used to improve several Google products.
- Replace hard-coded ranking system with a model trained on mobile phone usage
- Each phone contributed improvements to the global model without sending any training data to Google's servers
- Keep data from user interaction with the phones in private

Federated learning



Google

Gboard: language modeling

- Predict the next word based on typed text so far
- Powers the predictions strip

When should you consider federated learning?

- On-device data is more relevant than server-side proxy data
- On-device data is privacy sensitive or large
- Labels can be inferred naturally from user interaction

Google

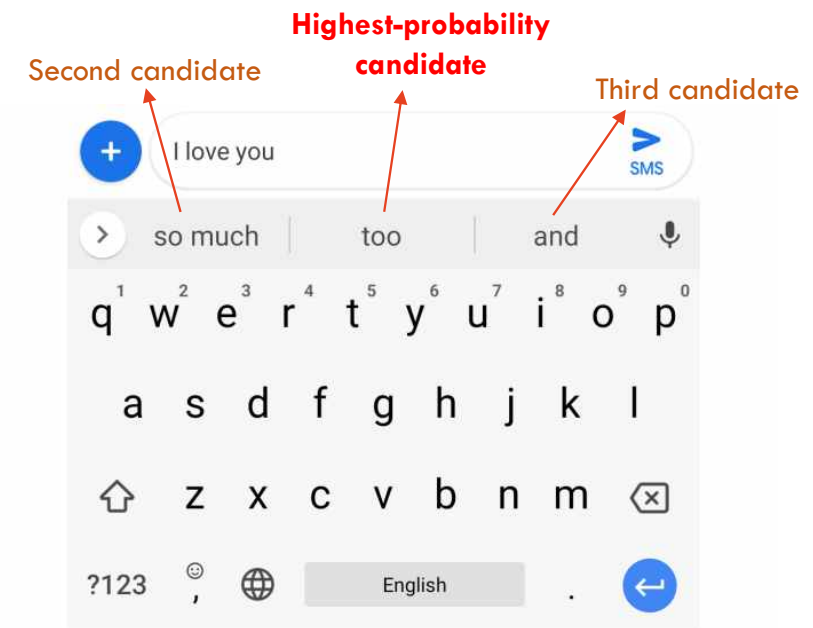
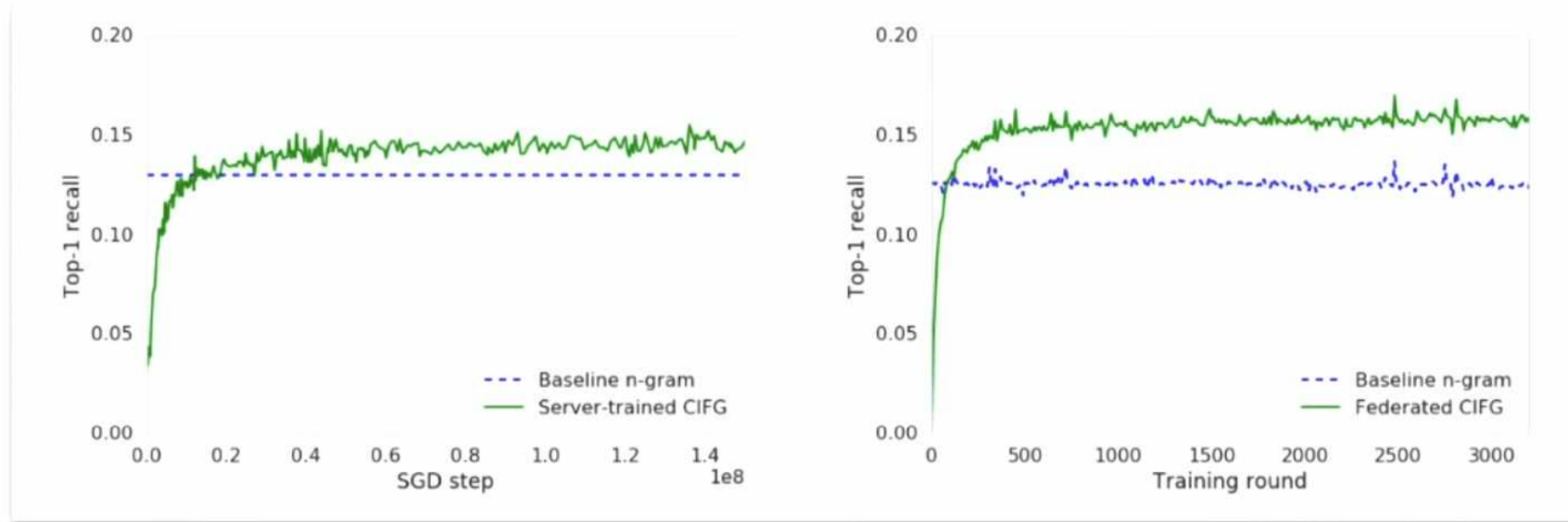


Fig. 1. Next word predictions in Gboard. Based on the context “I love you”, the keyboard predicts “and”, “too”, and “so much”.



Gboard: language modeling



Server-trained model
compared to **baseline**

Federated model
compared to **baseline**

A. Hard, *et al.* **Federated Learning for Mobile Keyboard Prediction.**
arXiv:1811.03604

Federated RNN (compared to prior n-gram model):

- Better next-word prediction accuracy: +24%
- More useful prediction strip: +10% more clicks

Characteristics of **federated learning**

vs. traditional *distributed learning*

Data locality and distribution

- **massively decentralized, naturally arising (non-IID) partition**
- Data is siloed, held by a small number of coordinating entities
- *system-controlled* (e.g. shuffled, balanced)

Data availability

- **limited availability, time-of-day variations**
- *almost all data nodes always available*

Addressability

- **data nodes are anonymous and interchangeable**
- *data nodes are addressable*

Node statefulness

- **stateless (generally no repeat computation)**
- *stateful*

Node reliability

- **unreliable (~10% failures)**
- *reliable*

Wide-area communication pattern

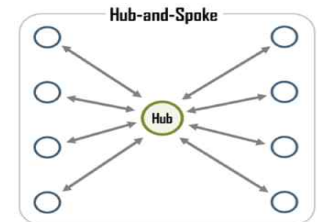
- **hub-and-spoke topology**
- *peer-to-peer topology (fully decentralized)*
- *none (centralized to one datacenter)*

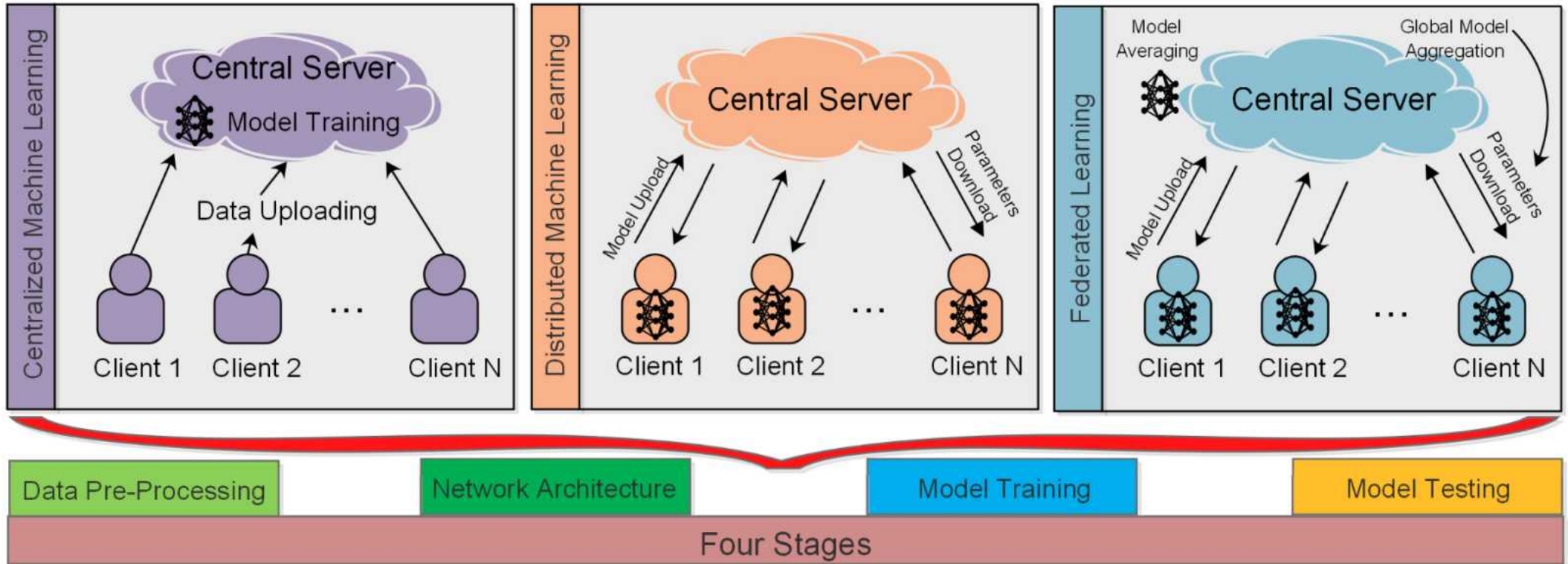
Distribution scale

- **massively parallel (1e9 data nodes)**
- *single datacenter*

Primary bottleneck

- **communication**
- *computation*





- Unique characteristics of Federated Learning
 - Non-IID
 - The data generated by each user are quite different
 - Unbalanced
 - Some users produce significantly more data than others
 - Massively distributed
 - Training data is stored across a very large number of devices
 - Limited communication
 - Unstable mobile network connections

- Recall traditional learning problem

- For a training dataset containing n samples (x_i, y_i) , $1 \leq i \leq n$, the training objective is:

$$\min_{w \in \mathbb{R}^d} f(w) \quad \text{where} \quad f(w) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^n f_i(w)$$

$f_i(w) = l(x_i, y_i, w)$ is the loss of the prediction on example (x_i, y_i)

- Deep learning optimization relies on SGD and its variants, through mini-batches

$$w_{t+1} \leftarrow w_t - \eta \nabla f(w_t; x_k, y_k)$$

Federated Learning problem

- Suppose n training samples are distributed to K clients, where P_k is the set of indices of data points on client k , and $n_k = |P_k|$.
- For training objective: $\min_{w \in \mathbb{R}^d} f(w)$

Algorithm 1 FederatedAveraging. The K clients are indexed by k ; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:

```

initialize  $w_0$ 
for each round  $t = 1, 2, \dots$  do
     $m \leftarrow \max(C \cdot K, 1)$ 
     $S_t \leftarrow$  (random set of  $m$  clients)
    for each client  $k \in S_t$  in parallel do
         $w_{t+1}^k \leftarrow$  ClientUpdate( $k, w_t$ )
     $w_{t+1} \leftarrow \sum_{k=1}^K \frac{n_k}{n} w_{t+1}^k$ 
    
```

ClientUpdate(k, w): // Run on client k

```

 $\mathcal{B} \leftarrow$  (split  $\mathcal{P}_k$  into batches of size  $B$ )
for each local epoch  $i$  from 1 to  $E$  do
    for batch  $b \in \mathcal{B}$  do
         $w \leftarrow w - \eta \nabla \ell(w; b)$ 
return  $w$  to server
    
```

The Federated Averaging Algorithm

$$f(w) = \sum_{k=1}^K \frac{n_k}{n} F_k(w)$$

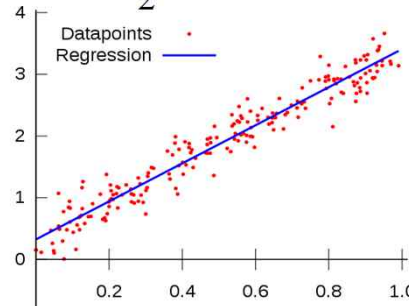
Global loss function

$$\text{where } F_k(w) \stackrel{\text{def}}{=} \frac{1}{n_k} \sum_{i \in P_k} f_i(w)$$

Local loss function at each client k

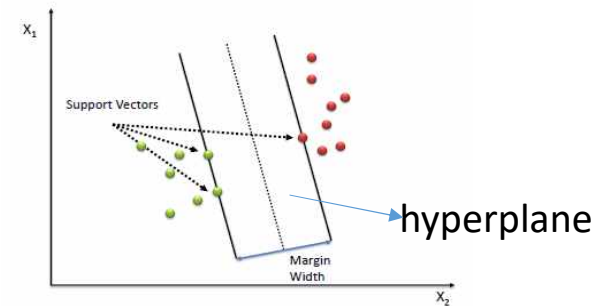
Linear regression

$$f_i(w) = \frac{1}{2} (x_i^T w - y_i)^2, y_i \in \{0, 1\}$$

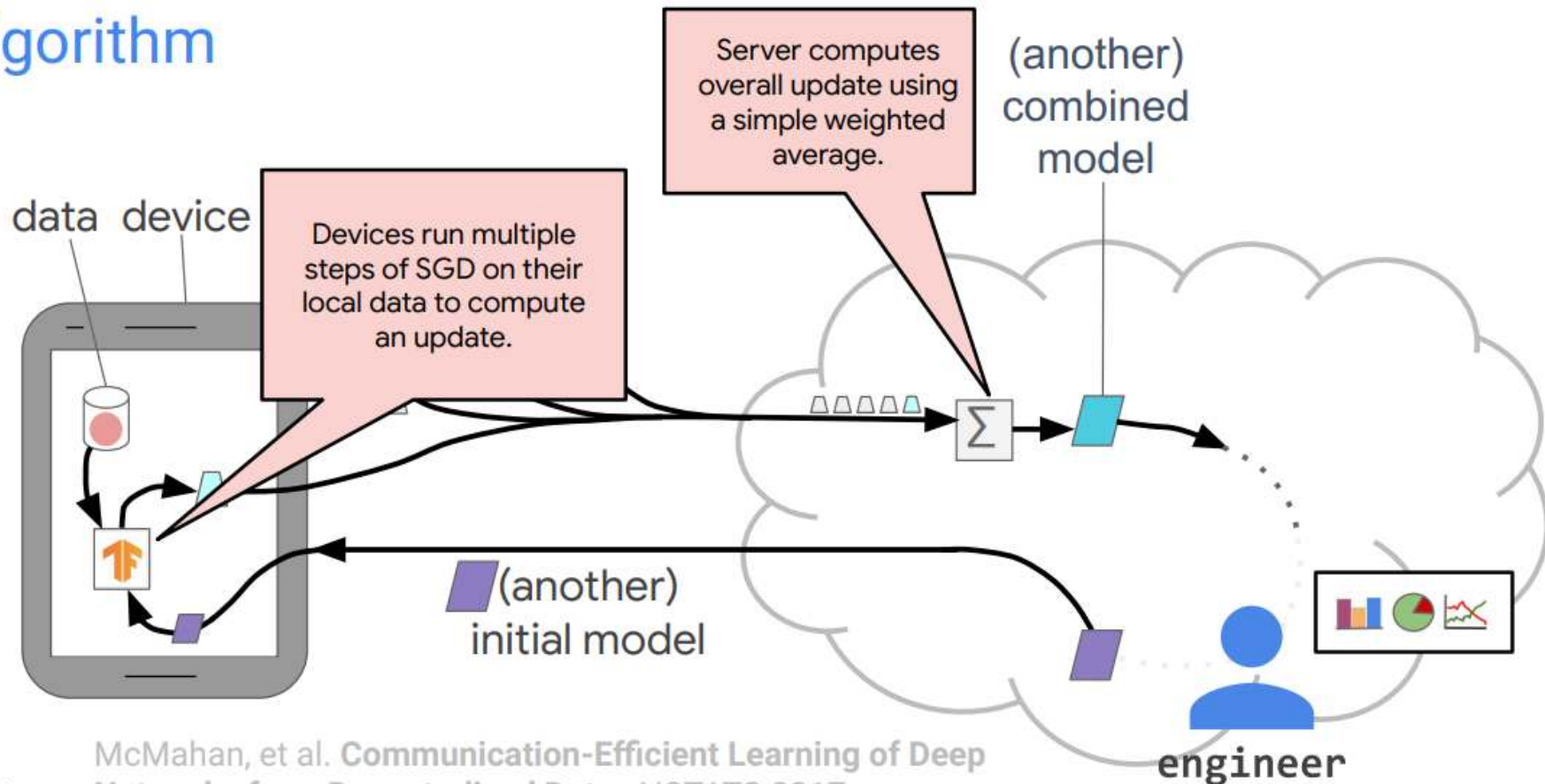


Support vector machine

$$f_i(w) = \{0, 1 - y_i x_i^T w\}, y_i \in \{-1, 1\}$$



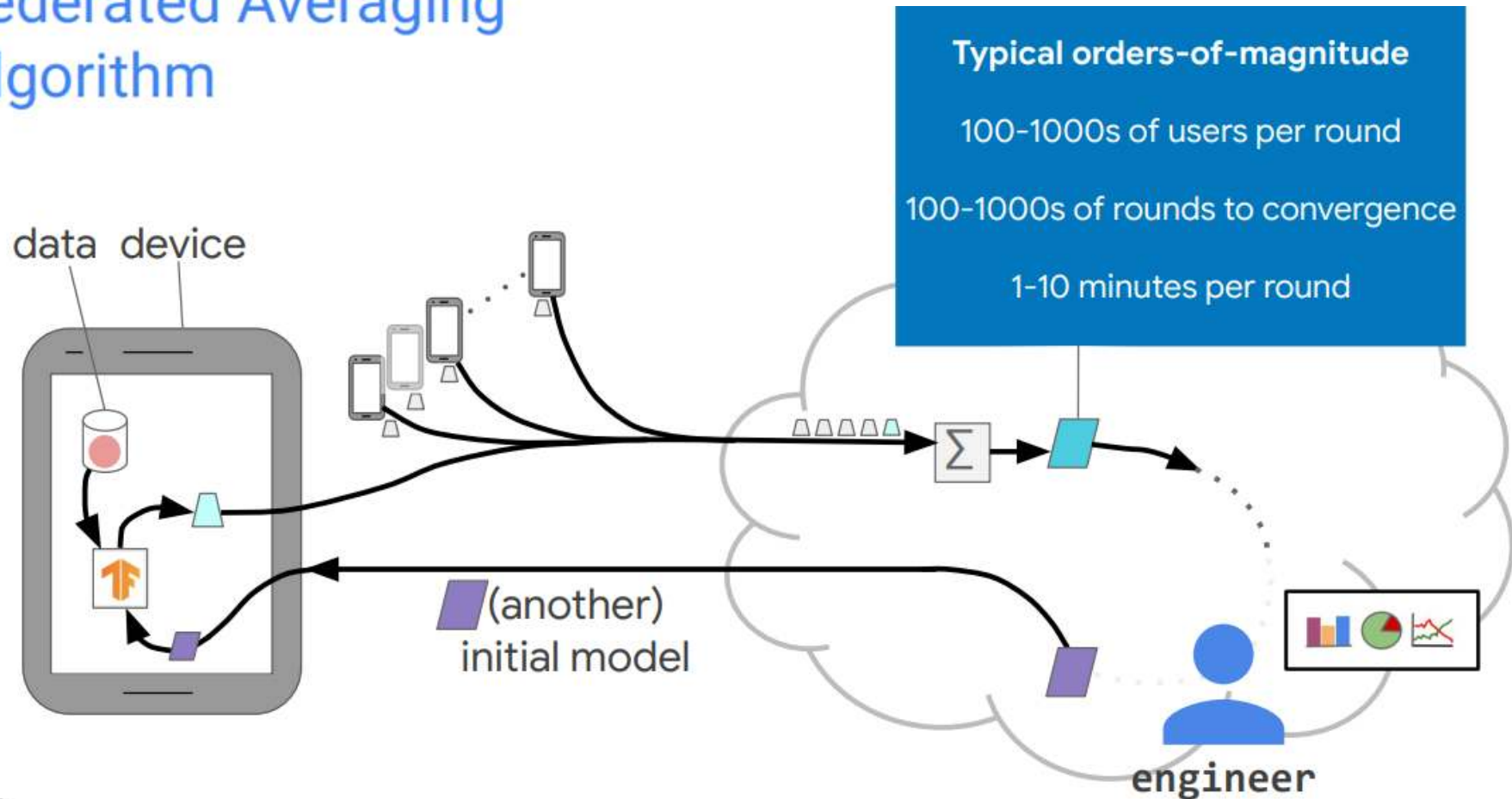
Federated Averaging Algorithm



Google

McMahan, et al. **Communication-Efficient Learning of Deep Networks from Decentralized Data.** AISTATS 2017.

Federated Averaging Algorithm



Google

The Federated Averaging algorithm

Server

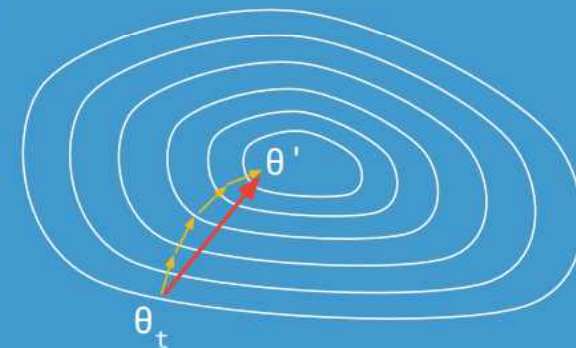
Until Converged:

1. Select a random subset (e.g. 1000) of the (online) clients
2. In parallel, send current parameters θ_t to those clients

Selected Client k

1. Receive θ_t from server.
2. Run some number of minibatch SGD steps, producing θ'
3. Return $\theta' - \theta_t$ to server.

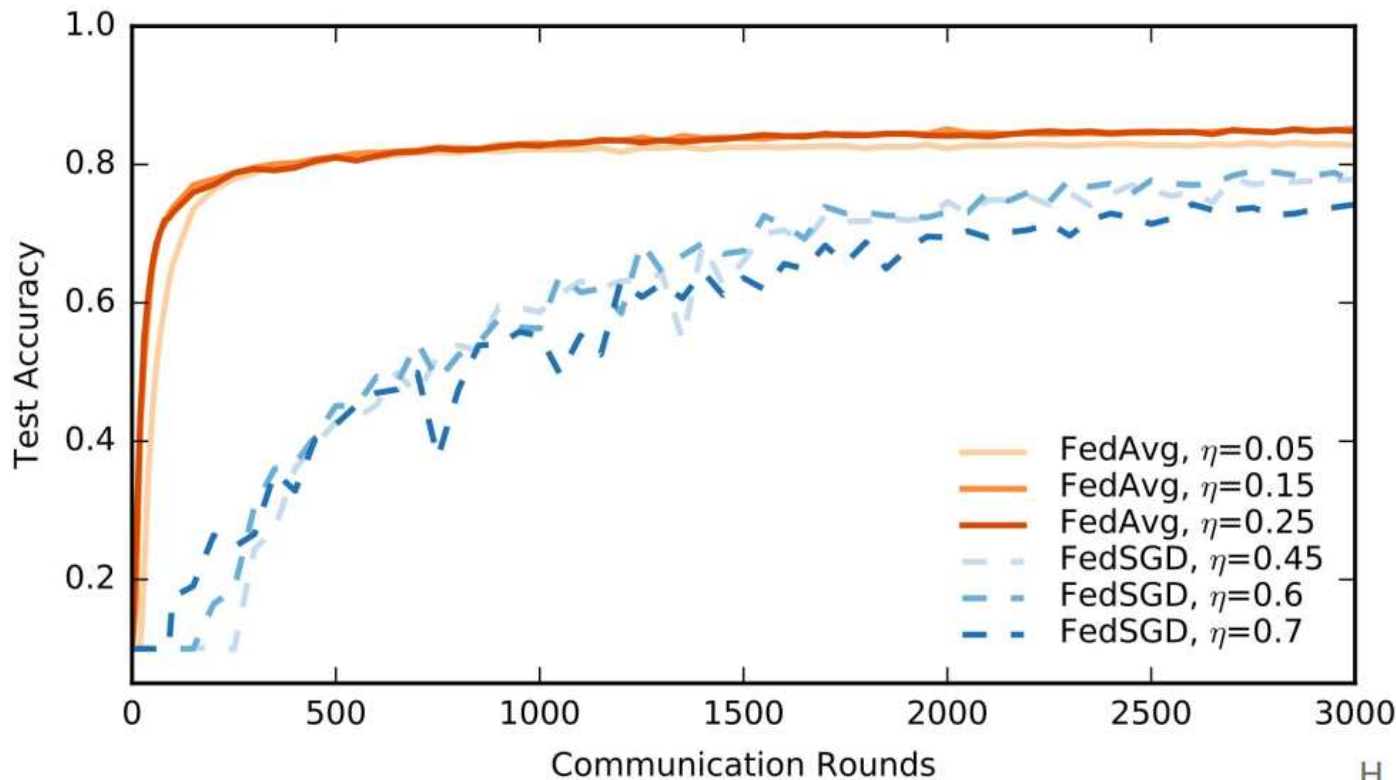
FedSGD: 1 step



3. $\theta_{t+1} = \theta_t + \text{data-weighted average of client updates}$

H. B. McMahan, et al.
Communication-Efficient Learning of
Deep Networks from Decentralized
Data. AISTATS 2017

Using the convolutional model for CIFAR-10



Updates to reach 82%
SGD 31,000
FedSGD 6,600
FedAvg 630

49x decrease in communication (updates) vs SGD

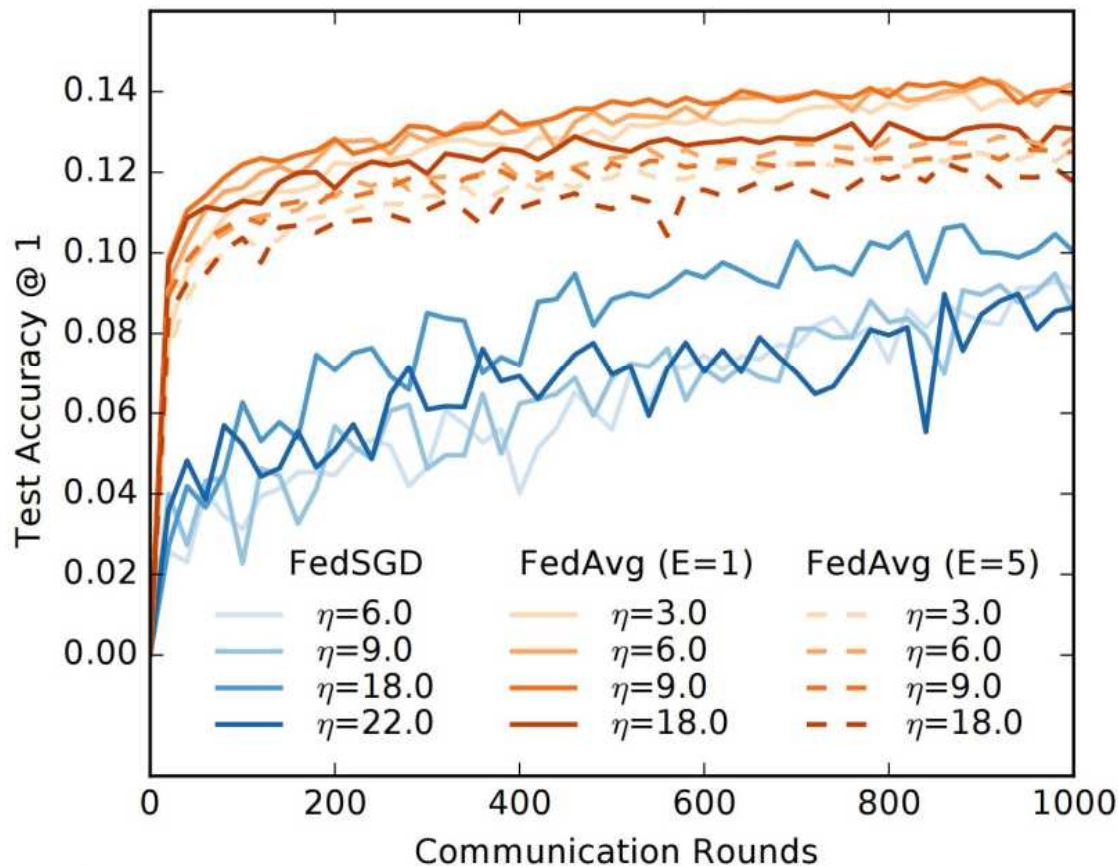
(IID and balanced data)

H. B. McMahan, et al.
Communication-Efficient Learning of Deep Networks from Decentralized Data. AISTATS 2017

Google

Large-scale LSTM for next-word prediction

Dataset: Large Social Network, 10m public posts, grouped by author.



Rounds to reach 10.5% Accuracy

FedSGD 820

FedAvg 35

23x decrease in communication rounds

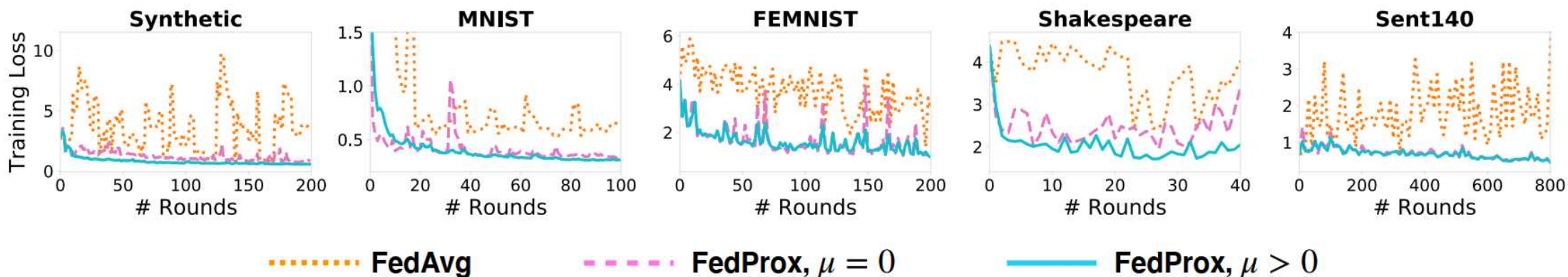
H. B. McMahan, *et al.*
Communication-Efficient Learning of Deep Networks from Decentralized Data. AISTATS 2017

FedProx:

Modified Local Subproblem:
$$\min_{w_k} F_k(w_k) + \frac{\mu}{2} \left\| w_k - w^t \right\|^2$$

a proximal term

- The proximal term explicitly limits the impact of heterogeneous local updates
- Don't drop straggler devices: instead [safely] incorporate partial work
- Generalization of FedAvg and is **more stable** in the **heterogeneous** setting



FedAdagrad (Momentum based algorithms)

Algorithm 3 FEDADAGRAD

Initialization: $x_0, \tau > 0$ and $v_{-1} \geq \tau^2$

for $t = 0, \dots, T - 1$ **do**

 Sample subset \mathcal{S} of clients

$x_{i,0}^t = x_t$

for each client $i \in \mathcal{S}$ **in parallel do**

for $k = 0, \dots, K - 1$ **do**

 Compute an unbiased estimate $g_{i,k}^t$ of $\nabla F_i(x_{i,k}^t)$

$x_{i,k+1}^t = x_{i,k}^t - \eta l g_{i,k}^t$

$\Delta_i^t = x_{i,K}^t - x_t$

$\Delta_t = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \Delta_i^t$

$v_t = v_{t-1} + \Delta_t^2$
 $x_{t+1} = x_t + \eta \frac{\Delta_t}{\sqrt{v_t + \tau}}$

Local SGD
updates at
clients

Adagrad updates on
model delta at server

Send the changes in local models
to the server for aggregation



Momentum based update



FedYogi and FedAdam (Momentum based algorithms)

Algorithm 4 **FEDYOGI** (and **FEDADAM**)

Initialization: $x_0, v_{-1} \geq \tau^2$, decay $\beta_2 \in (0, 1)$

for $t = 0, \dots, T - 1$ **do**

 Sample subset \mathcal{S} of clients

$$x_{i,0}^t = x_t$$

for each client $i \in \mathcal{S}$ **in parallel do**

for $k = 0, \dots, K - 1$ **do**

 Compute an unbiased estimate $g_{i,k}^t$ of $\nabla F_i(x_{i,k}^t)$

$$x_{i,k+1}^t = x_{i,k}^t - \eta g_{i,k}^t$$

$$\Delta_i^t = x_{i,K}^t - x_{t-1}$$

$$\Delta_t = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \Delta_i^t$$

$$v_t = v_{t-1} - (1 - \beta_2) \Delta_t^2 \text{sign}(v_{t-1} - \Delta_t^2) \text{ (FEDYOGI)}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) \Delta_t^2 \text{ (FEDADAM)}$$

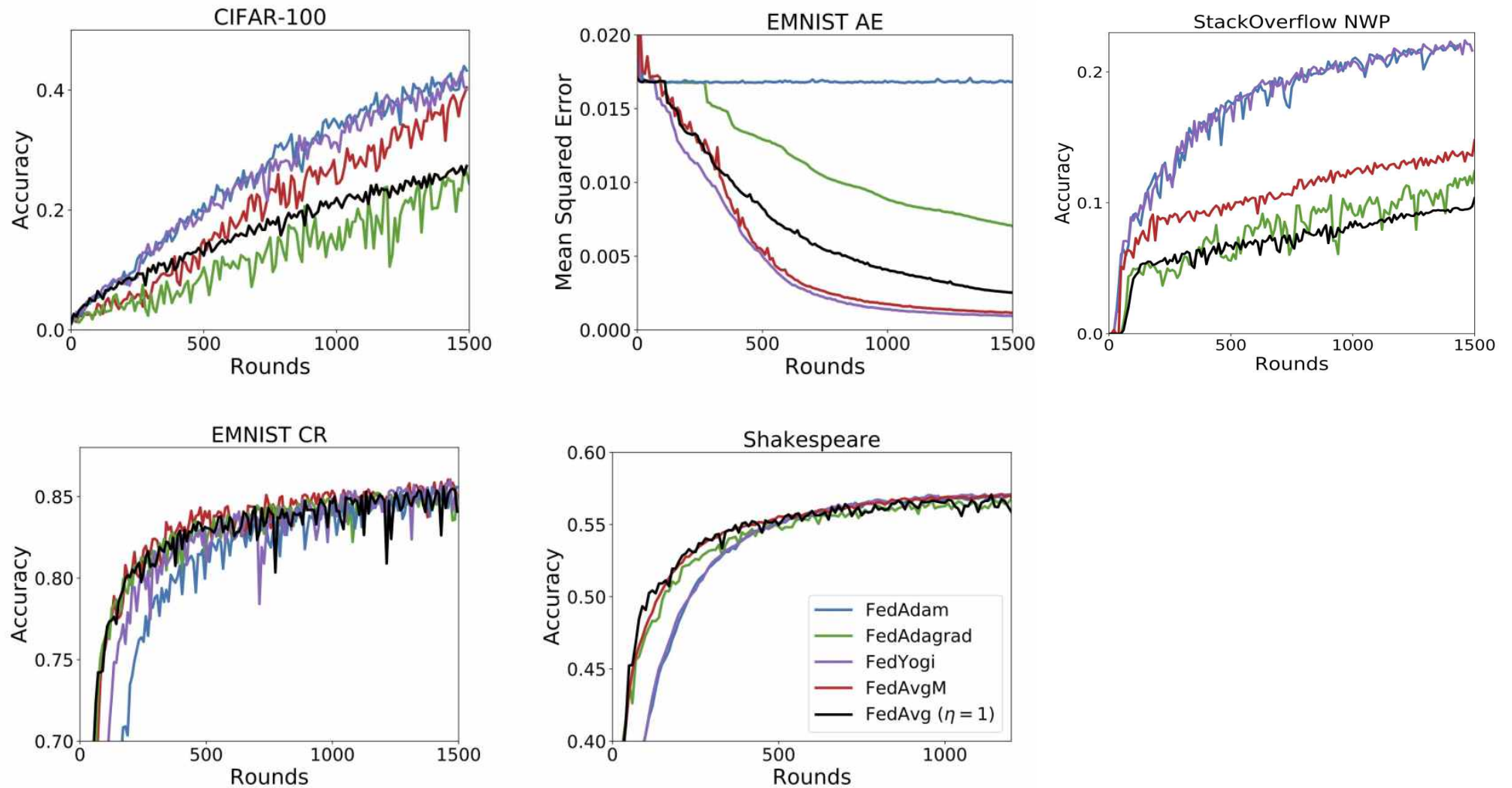
$$x_{t+1} = x_t + \eta \frac{\Delta_t}{\sqrt{v_t + \tau}}$$

Local SGD updates
at clients

Adam/Yogi updates
on model delta at
server

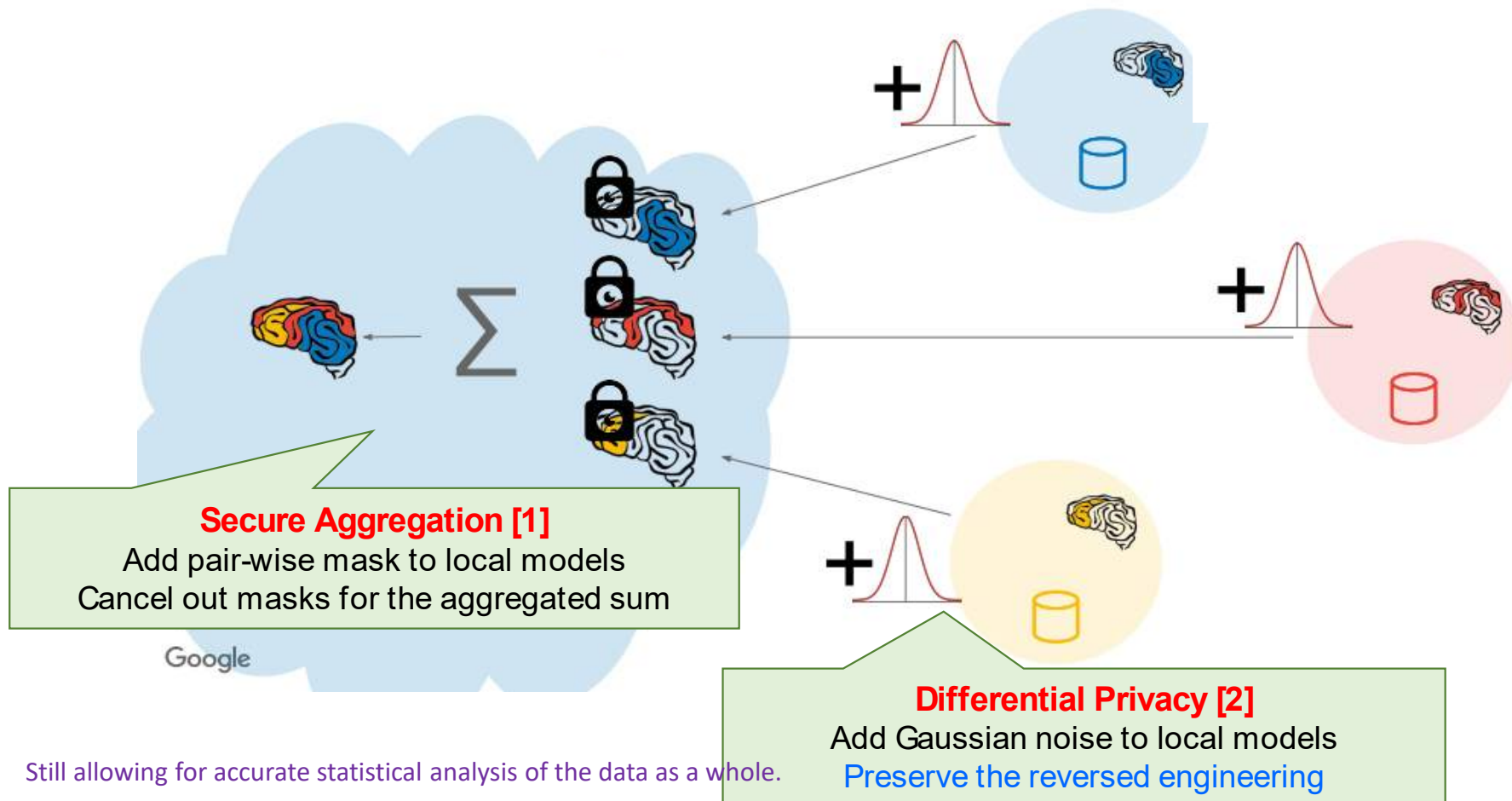
Momentum based update

the momentum term is a weighted sum of the previous gradient updates; β_1, β_2 : Momentum parameters; 0.9, 0.99



Reddi, Sashank, et al. "Adaptive Federated Optimization." *arXiv preprint arXiv:2003.00295* (2020).

Enhancing the **security** and **privacy** of FL



[1] K. Bonawitz, et al. "Practical Secure Aggregation for Privacy-Preserving Machine Learning." CCS 2017.

[2] H. B. McMahan, et al. "Learning Differentially Private Recurrent Language Models" arXiv preprint arXiv:1710.06963 (2017).

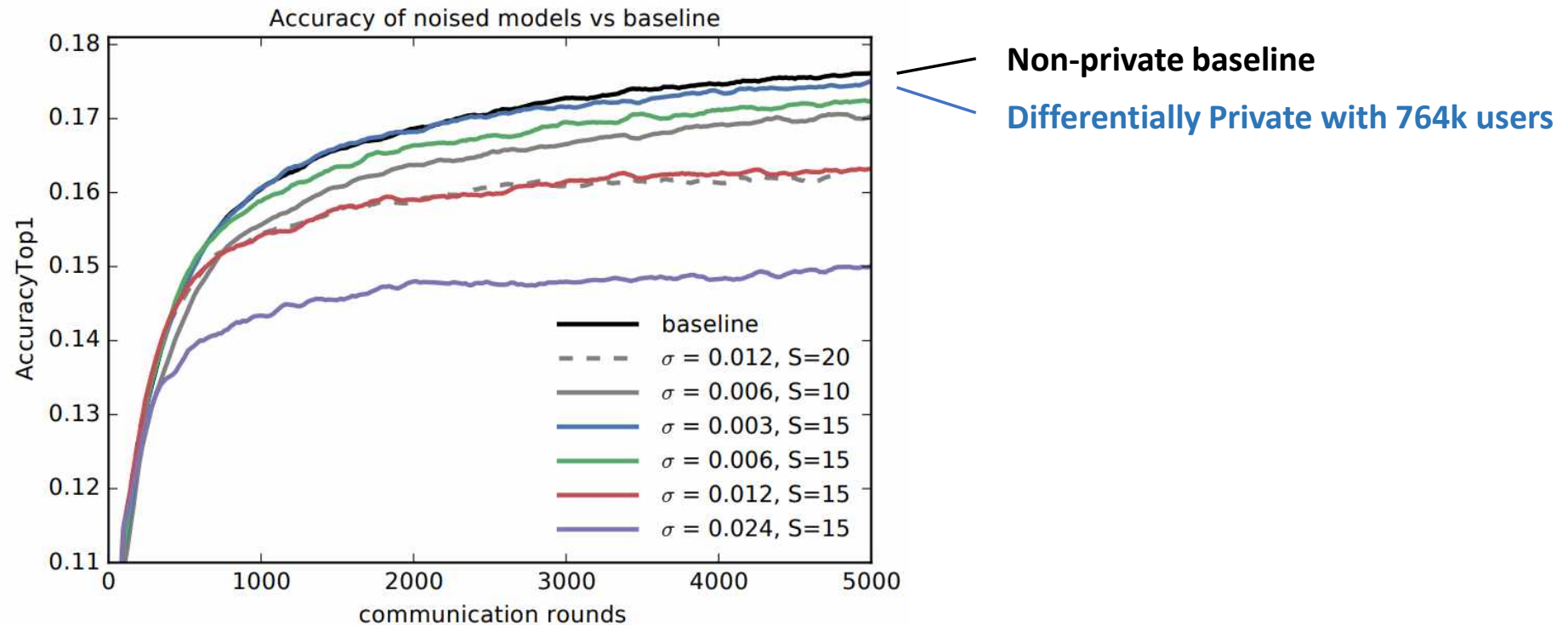
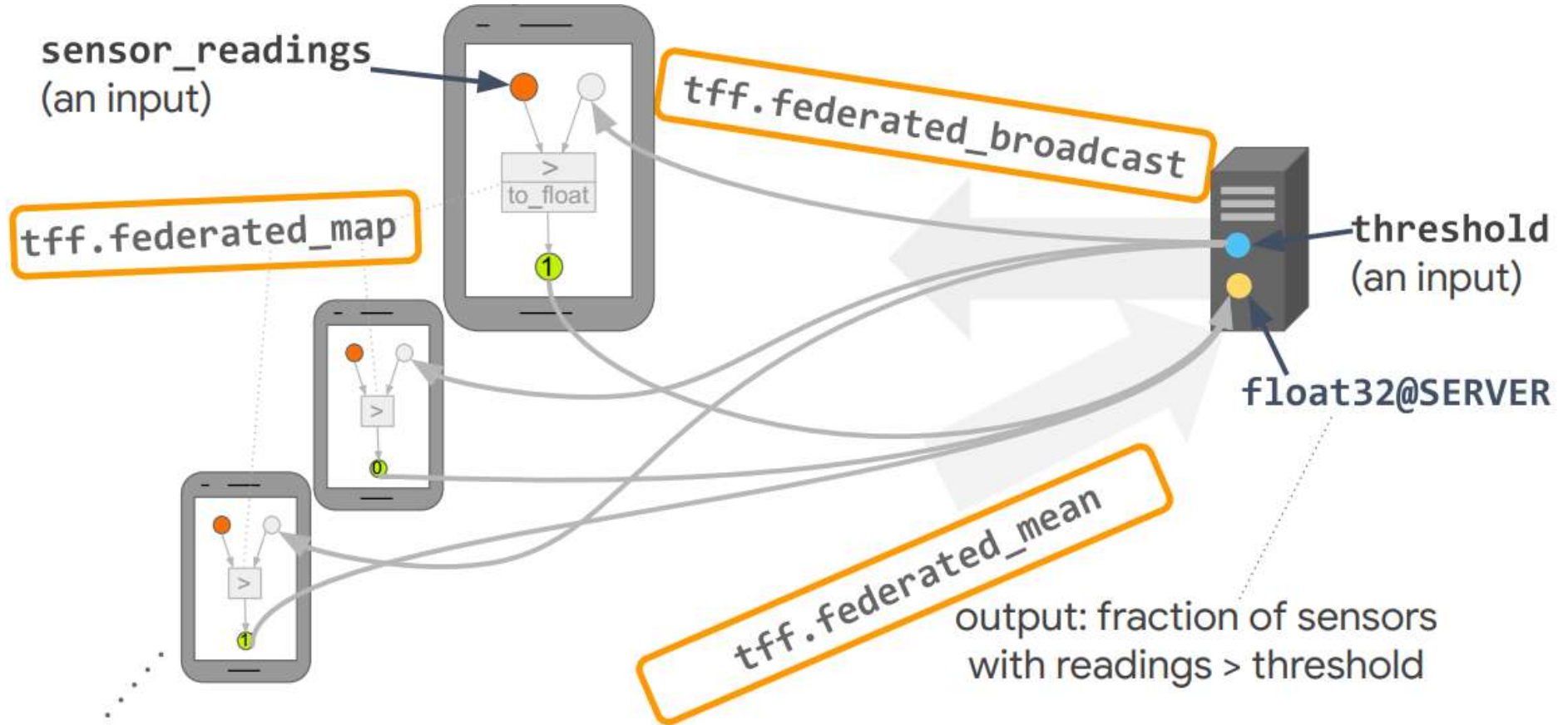


Figure 1: Noised training versus the non-private baseline. The model with $\sigma = 0.003$ nearly matches the baseline.

Federated Learning: Implementation

TensorFlow Federated (TFF): Machine Learning on Decentralized Data



* without the need to share the data with a central server.

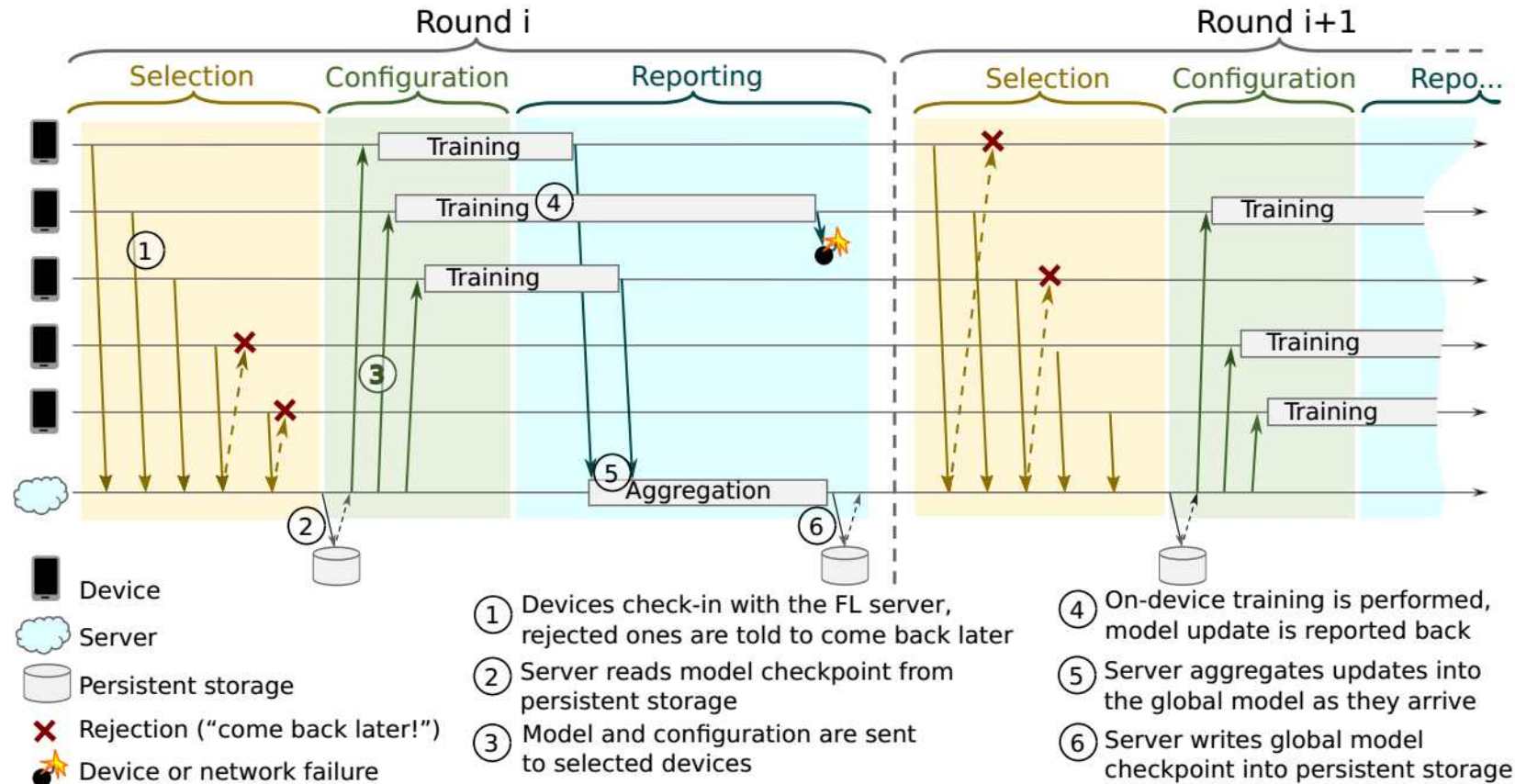
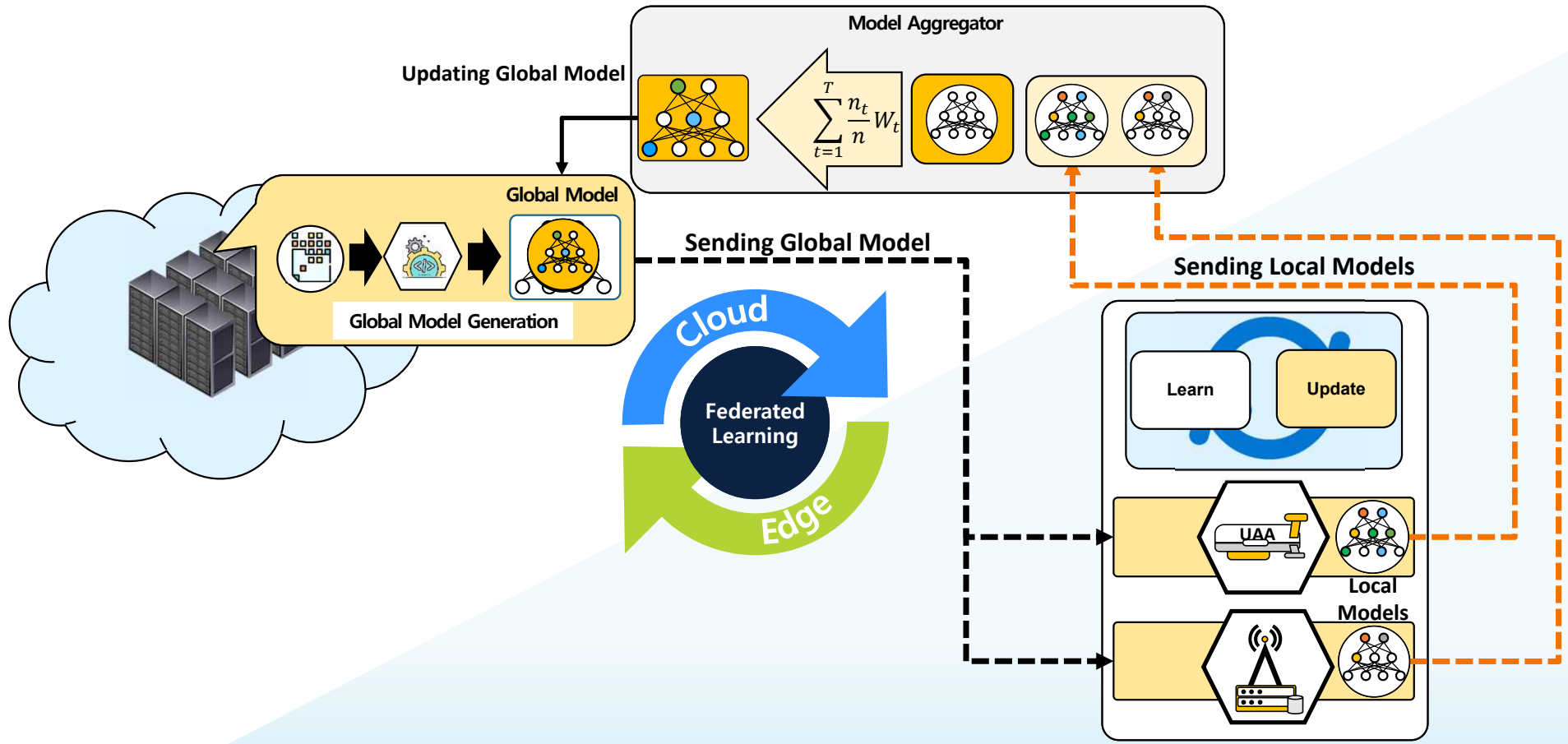


Figure 1: Federated Learning Protocol

- Approaches that scale FL to larger models, including model and gradient **compression** techniques
- Novel applications of FL, extension to new learning algorithms and model classes.
- **Theory for FL** (i.e., convergence analysis)
- **Communications for FL** (e.g., resource allocation, scheduling, etc)
- **FL for communications** (e.g., Federated Deep Reinforcement Learning for offloading, caching, etc)
- Enhancing the **security** and **privacy** of FL, including cryptographic techniques and differential privacy
- **Bias and fairness** in the FL setting
- Not everyone has to have the same model (**multi-task** and pluralistic learning, **personalization**, domain adaptation)
- Generative models, transfer learning, semi-supervised learning



Deployment scenario at the edge

- **The under-explored resource allocation for the Federated Learning scheme:**
 - The uncertainty of wireless channels
 - UEs with heterogeneous power constraints
 - The difference in local training data size
- **Contributions of [1]:**
 - Formulate a Federated Learning over wireless network problem, namely (FEDL)
 - Decompose the non-convex FEDL problem and transform it to three convex sub-problems and obtain the globally optimal solution
 - Trade-off between **computation** and **communication latencies** determined by learning accuracy level
 - Trade-off between the **Federated Learning time** and **UE energy consumption**.

[1] Nguyen H. Tran, Wei Bao, Albert Zomaya, Minh N.H. Nguyen and Choong Seon Hong, “**Federated Learning over Wireless Networks: Optimization Model Design and Analysis**,” *IEEE International Conference on Computer Communications (INFOCOM 2019)*, April 29 - May 2, 2019, Paris, France

Iterative Process

Step 1. Local Computation: Every UE needs to solve the local learning problem

$$w_n^{(t)} = \arg \min_{w_n \in \mathbb{R}^d} F_n(w_n | w^{(t-1)}, \nabla J^{(t-1)})$$

with the local error $0 \leq \theta \leq 1$

Step 2. Transmit Learning Parameters: UEs send their weight parameters $w_n^{(t)}$, the gradient $\nabla J_n^{(t)}$ to the controller via a **shared wireless environment** (TDMA for uplink).

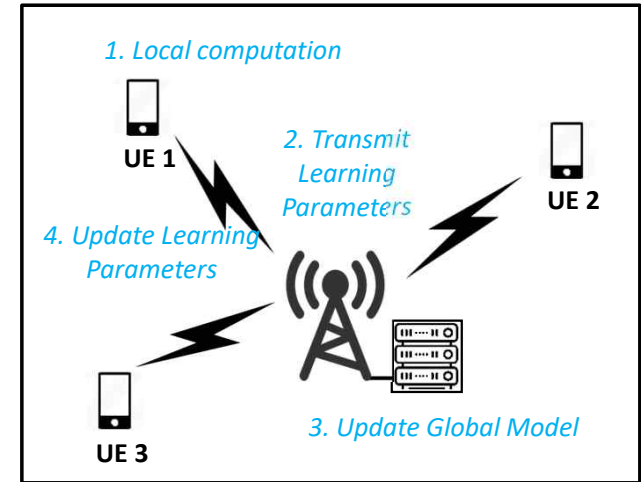
Step 3. Update Global Model: The local model parameters and gradients are aggregated at the controller

$$w^{(t+1)} = \frac{1}{N} \sum_{n=1}^N w_n^{(t)}$$

$$\nabla J^{(t+1)} = \frac{1}{N} \sum_{n=1}^N \nabla J_n^{(t)}$$

Step 4. Update Learning Parameters: These updated learning parameters then are broadcast to all UEs.

Until a global error $0 \leq \varepsilon \leq 1$ is achieved.



Federated Learning Scheme

- Convergence analysis for number of global rounds and local iterations

- Consider an iterative optimization algorithm \mathcal{A} is used to solve local learning problem, then the overall learning time

$$\text{TIME}(\mathcal{A}, \theta) = \underbrace{K_{\mathcal{A}}(\epsilon, \theta)}_{\text{\#global rounds}} \times (\underbrace{c}_{\text{Communication}} + \underbrace{\mathcal{T}_{\mathcal{A}}(\theta)}_{\text{Computation}}),$$

- According to [1], the general bound on the number of global rounds is

$$K(\epsilon, \theta) = \frac{O(\log(1/\epsilon))}{1-\theta}, \quad \text{Normalize} \quad \gg \quad K(\theta) = \frac{1}{1-\theta}$$

where controllable local error θ given the relative global error ϵ .

- The number of local iterations is upper bounded by $O(\log(1/\theta))$

- The proposed **FEDL** optimization problem

➤ Minimizing the **UEs' energy consumption** and **learning time**

FEDL: $\min. K(\theta) [E_{glob}(f, \tau, \theta) + \kappa T_{glob}(T_{cmp}, T_{com}, \theta)]$ (13)

s.t. $\sum_{n=1}^N \tau_n \leq T_{com},$ (14) Communication Time

$\max_n \frac{c_n D_n}{f_n} = T_{cmp},$ (15) Computational Time

$f_n^{min} \leq f_n \leq f_n^{max}, \forall n \in \mathcal{N},$ (16) CPU cycle of UEs

$p_n^{min} \leq p_n(s_n/\tau_n) \leq p_n^{max}, \forall n \in \mathcal{N},$ (17) Transmission power

$0 \leq \theta \leq 1.$ (18) Local error

D_n : a local data set

c_n : the number of CPU cycles for UE n to execute one sample of data

S_n : signal strength (watts or dBm)

• Solution Approach

- The non-convex **FEDL** problem is decomposed into three convex subproblems and obtains closed-form solutions

CPU-cycle control



$$\text{SUB1: } \min. \quad \sum_{n=1}^N E_n^{cmp}(f_n) + \kappa T_{cmp} \quad (19)$$

$$\text{s.t.} \quad \frac{c_n D_n}{f_n} \leq T_{cmp}, \quad \forall n \in \mathcal{N}, \quad (20)$$

$$f_n^{min} \leq f_n \leq f_n^{max}, \quad \forall n \in \mathcal{N}. \quad (21)$$

Uplink power control



$$\text{SUB2: } \min. \quad \sum_{n=1}^N E_n^{com}(\tau_n) + \kappa T_{com} \quad (22)$$

$$\text{s.t.} \quad \sum_{n=1}^N \tau_n \leq T_{com}, \quad (23)$$

$$p_n^{min} \leq p_n(s_n/\tau_n) \leq p_n^{max}, \quad \forall n \in \mathcal{N}. \quad (24)$$

Local accuracy control



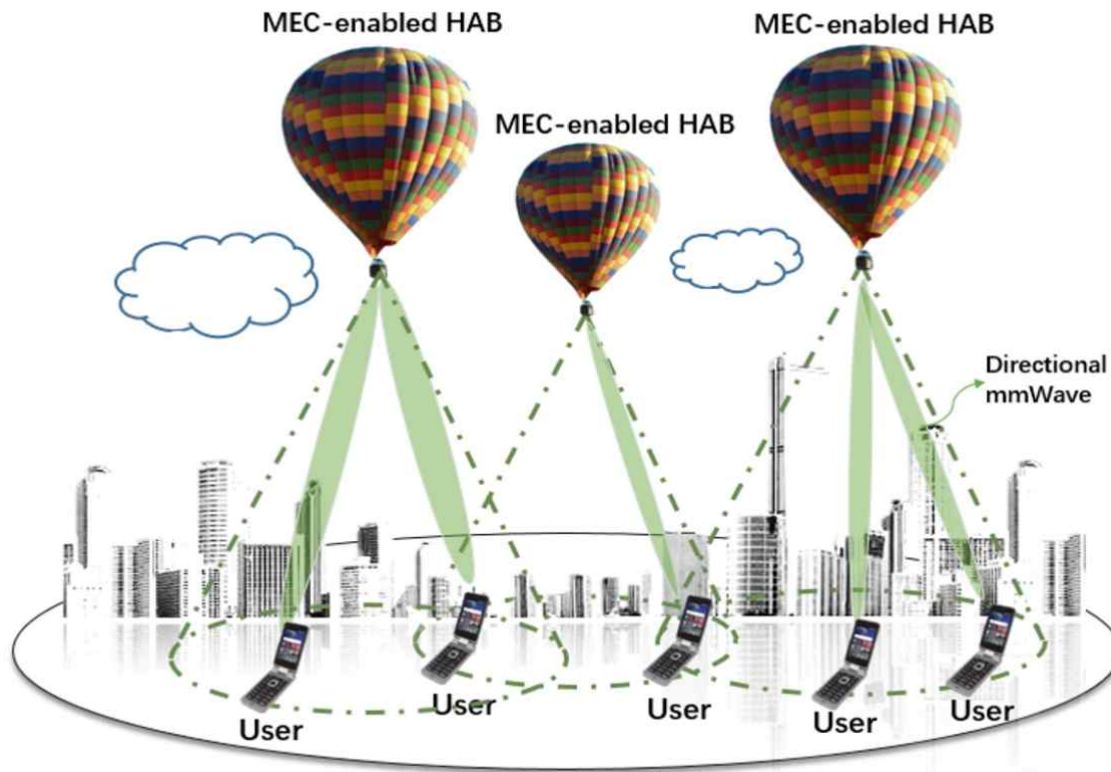
$$\text{SUB3:}$$

$$\min. \quad K(\theta) [E_{glob}(f^*, \tau^*, \theta) + \kappa T_{glob}(T_{cmp}^*, T_{com}^*, \theta)]$$

$$\text{s.t.} \quad 0 \leq \theta \leq 1. \quad (37)$$

□ **Theorem 1.** *The globally optimal solution to FEDL is the combined solutions to three sub-problems SUB1, SUB2, and SUB3.*

- Federated Learning for Task and Resource Allocation in Wireless High Altitude Balloon (HAB) Networks

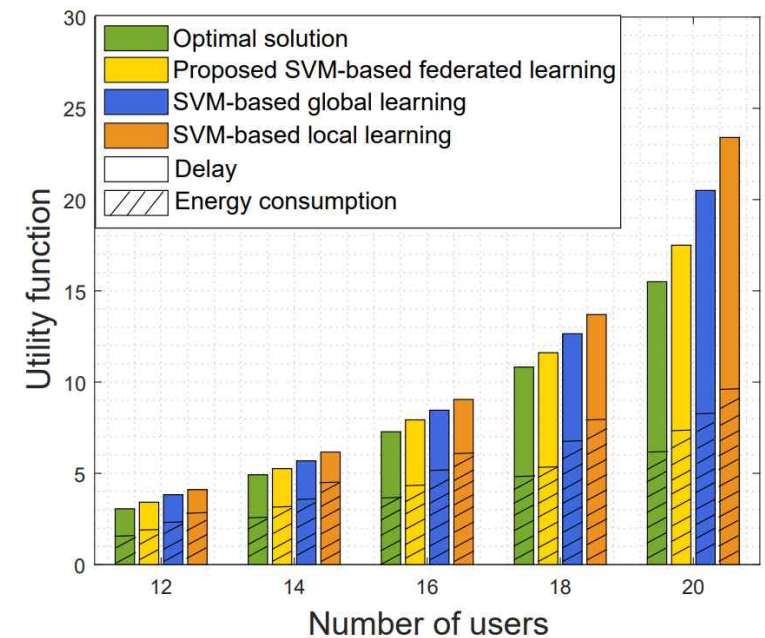
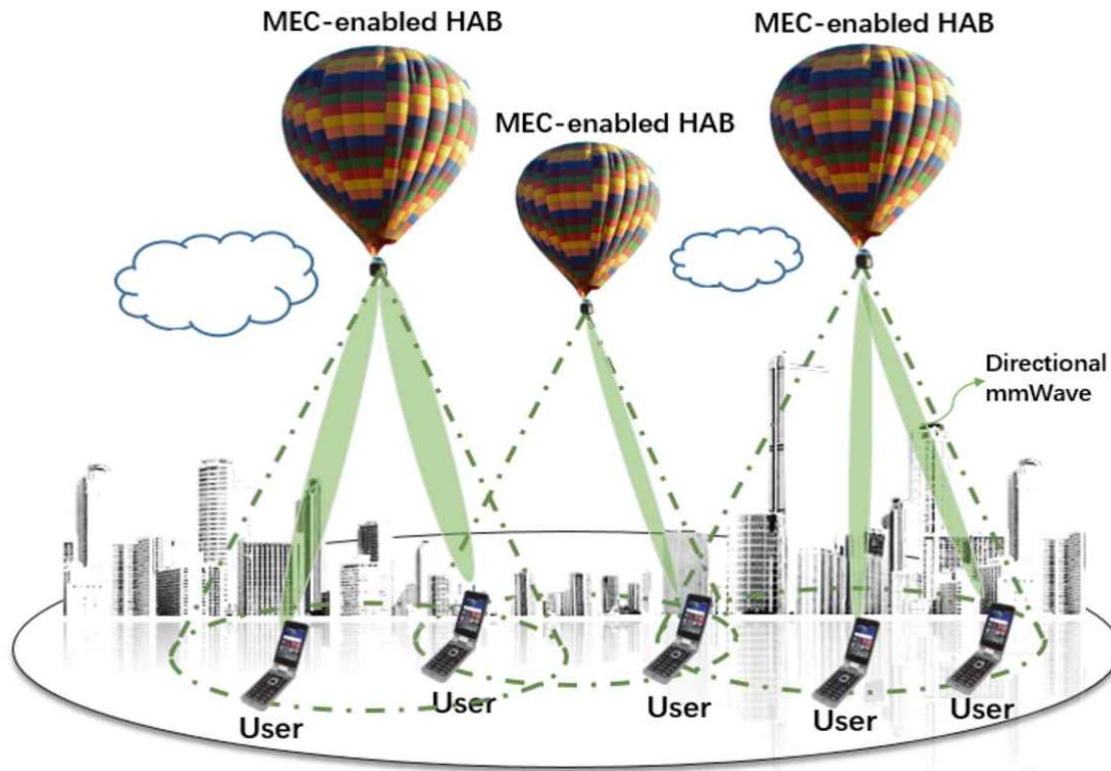


SVM : Finding the optimal decision boundary in the space by maximizing the margin.

- Support vector machine (SVM)-based federated learning (FL) algorithm method enables each HAB to cooperatively build an SVM model for proactive user associations.
 - Without any transmissions of historical user association results nor of the data size of the task requested
 - To
- Given the association decision, the service sequence and task allocation of each user can be optimized to minimize the weighted sum of the energy and time consumption.

[1] Wang, Sihua, Mingzhe Chen, Changchuan Yin, Walid Saad, Choong Seon Hong, Shuguang Cui, and H. Vincent Poor., "Federated Learning for Task and Resource Allocation in Wireless High Altitude Balloon Networks," IEEE INTERNET OF THINGS JOURNAL, VOL. 8, NO. 24, DECEMBER 15, 2021


- Federated Learning for Task and Resource Allocation in Wireless High Altitude Balloon (HAB) Networks



Beyond learning: federated analytics

- **Federated analytics** is the practice of **applying data science methods** to the **analysis of raw data that is stored locally** on users' devices.
- Like federated learning, it works by running **local computations** over each device's data, and only **making the aggregated results** - and never any data from a particular device - available to product engineers.
- Unlike federated learning, however, federated analytics aims to support basic data science needs.
 - Federated quantile estimation
 - Federated counting of distinct elements or events
 - Federated histogram estimation over closed sets
 - Federated heavy hitters discovery over open sets
 - Federated density estimation of vector spaces
 - Federated selection of random data subsets
 - Federated SQL?
 - etc...

definition proposed in

 <https://ai.googleblog.com/2020/05/federated-analytics-collaborative-data.html>

- AI is moving towards edge devices with the availability of massively distributed data sources and the increase in computing power for handheld and wireless devices.
- In Federated Learning (FL), on-device learning agents collaboratively train a global learning model without sharing their local datasets.
- The mainstream research of FL is proposing new algorithms to improve both theoretical and practical performance following their theoretical convergence analysis.
- FL is applied in a variety of mobile and edge devices applications.
- Communications for FL and FL for communications networks are our current research interests.

- Introduction: Motivation of Federated Learning
- Federated Learning
 - FL Formulation
 - FL Algorithms
 - Ongoing Research Problems
 - Federated Learning: at the Edge
 - Summary
- **Democratized Learning**
 - Introduction
 - Key Components
 - Ongoing Research Problems
- Multimodal Federated Learning
 - Introduction
 - Key Components
 - Ongoing Research Problems

Challenges of conventional FL scheme

- Large-scale, unbalanced, and highly personalized data is extremely challenge in practice such as hand writing and voice recognition applications
 - Limited number of samples in the local data
 - Heterogeneity of labels data in classification problem
 - Limited information (e.g., partial observation) about the environments
- The personalized learning performance at each learning agent can be declined (**negative impact**) due to inappropriate aggregation among exceedingly different learning agents characteristic
- The conventional FL cannot handily resolve the underlying **cohesive relation between global and personalized performance**

Motivations of Democratized Learning Philosophy

- **Hierarchical structure** of social or many complex systems
- **Individuals** cultivate personal objectives, skills and interact with each other to form many levels of social groups
- The higher-level and larger groups have more capabilities to solve complex problems via the **collective contributions** of their members

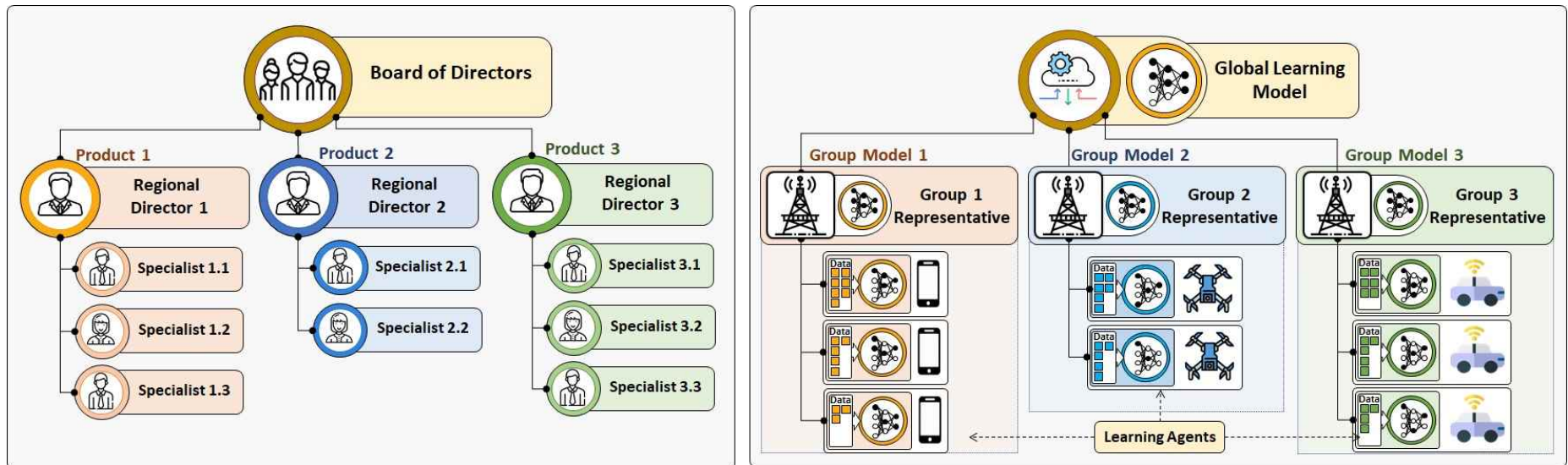


Fig. 1: Analogy of a hierarchical distributed learning system.

Motivations of Democratized Learning Philosophy

- **Unique features of the democracy in future distributed learning systems**
 - According to the **differences** in their characteristics, learning agents **form appropriate groups** that can be specialized for similar agents to deal with the learning tasks
 - Learning agents are **free to join** any of the appropriate groups and **exhibit equal power** in the construction of their groups' generalized learning model.
 - These specialized groups are **self-organized in a hierarchical structure** and collectively construct the shared generalized learning knowledge to improve their learning performance by reducing individual biases
 - The **power of each group** can be represented by the number of its members which varies over the training time.

Motivations of Democratized Learning Philosophy

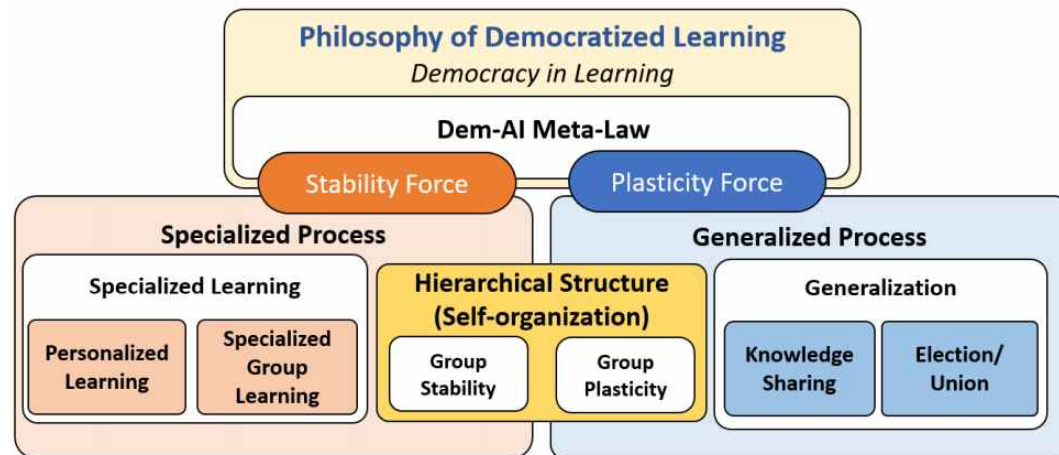
- Recent neuroscience research studies the continual life-long learning capabilities for a **general artificial intelligence** as in biological intelligence
 - **Generalization capabilities** due to high synaptic plasticity level allow easier to adapt and learn new knowledge
 - **Specialization capabilities** increase the specific complex skills

- **The duality of processes in distributed learning**
 - The **generalized process**
 - The high-level of **plasticity**, the easier to change the group members
 - **Generalization** broadens the knowledge by sharing among members
 - The **specialized process**
 - **Specialized learning** exploits the personalized data at learning agents
 - Encourage a separation of groups due to the personalized characteristics
=> Groups become **stable**.

Concepts

- **Definition:** Democratized Learning (**Dem-AI** in short) focuses on the study of a **dual** (coupled and working together) **specialized-generalized processes** in a **self-organizing hierarchical structure** of **large-scale distributed** learning systems.
- The specialized and generalized processes operate jointly towards an ultimate learning goal identified as performing **collective learning** from biased learning agents.

- **Specialized Process**
- **Generalized Process**
- **Hierarchical structuring**



Conceptual architecture of the democratized learning philosophy

M. N. H. Nguyen, S. R. Pandey, K. Thar, N. H. Tran, M. Chen, W. Saad, and C. S. Hong, "Distributed and democratized learning: Philosophy and research challenges," *IEEE Computational Intelligence Magazine*, Vol. 16, Issue 1, pp. 49-62, Jan. 2021.

Democratized Learning vs Federated Learning

Criteria	Democratized Learning	Federated Learning
Structure	<ul style="list-style-type: none"> • Self-organizing hierarchical structure 	<ul style="list-style-type: none"> • Not fully analyze for hierarchical structures
Performance	<ul style="list-style-type: none"> • Global, groups, personal local learning performance 	<ul style="list-style-type: none"> • More focus global model learning performance • Algorithms are developed based on convergence analysis of algorithms and strong theoretical assumptions.
Learning capabilities	<ul style="list-style-type: none"> • Specialization and Generalization capabilities • Multiple learning tasks 	<ul style="list-style-type: none"> • Personalization/ Generalization/ Global performance
Scalability	<ul style="list-style-type: none"> • Nature-inspired large-scale • Easier for decentralized management 	<ul style="list-style-type: none"> • Large-scale • Centralized management
Others	<ul style="list-style-type: none"> • Enhancing privacy and security based on hierarchical grouping • Diversity of models, democracy in learning features 	<ul style="list-style-type: none"> • Distillation, Generative adversarial network, Fairness, Meta-Learner

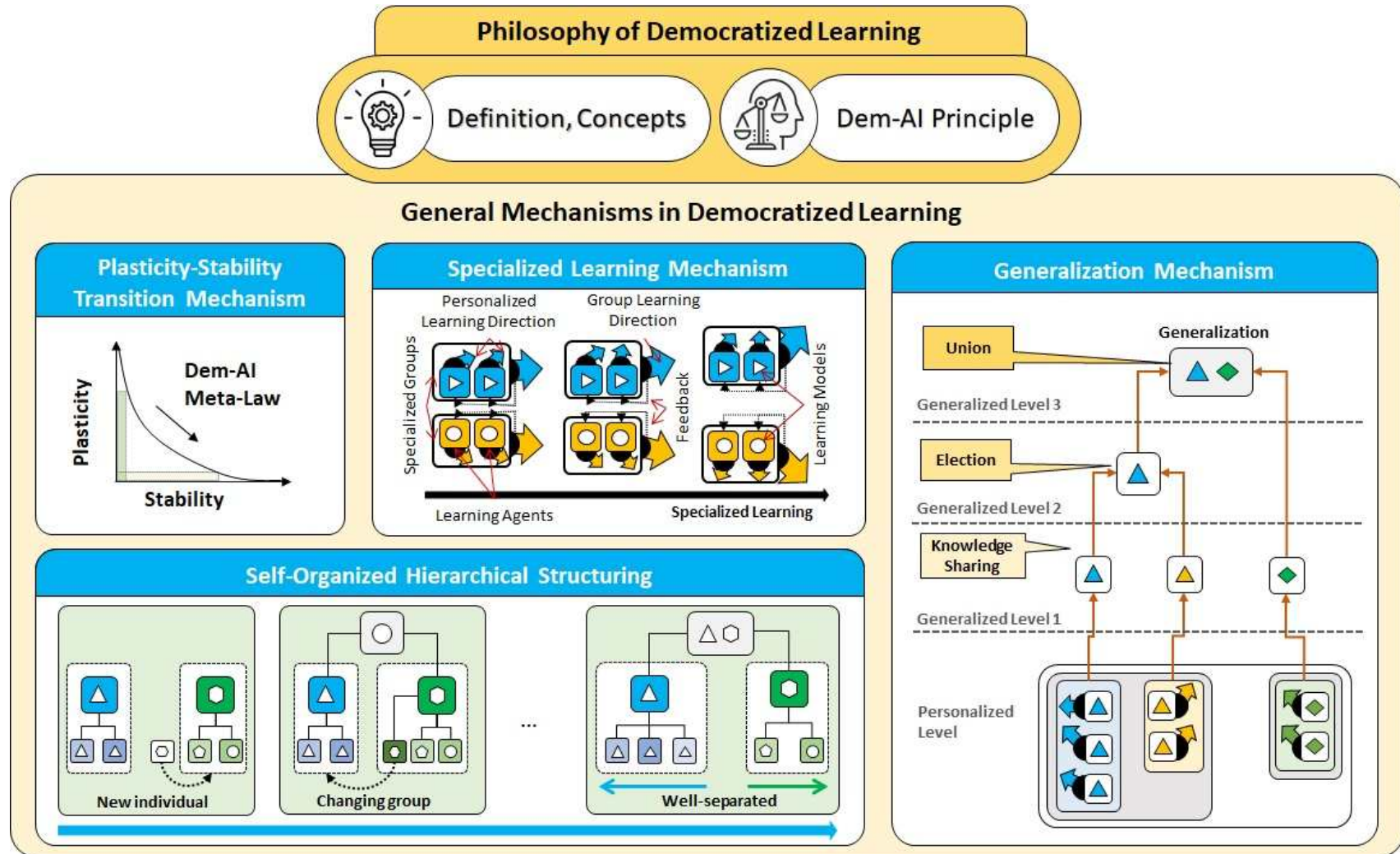


Fig. 2: Anatomy of Democratized Learning.

Democratized Learning: Key Components

- The self-organizing hierarchical structure of the Dem-AI system evolves to adapt to the training environment.

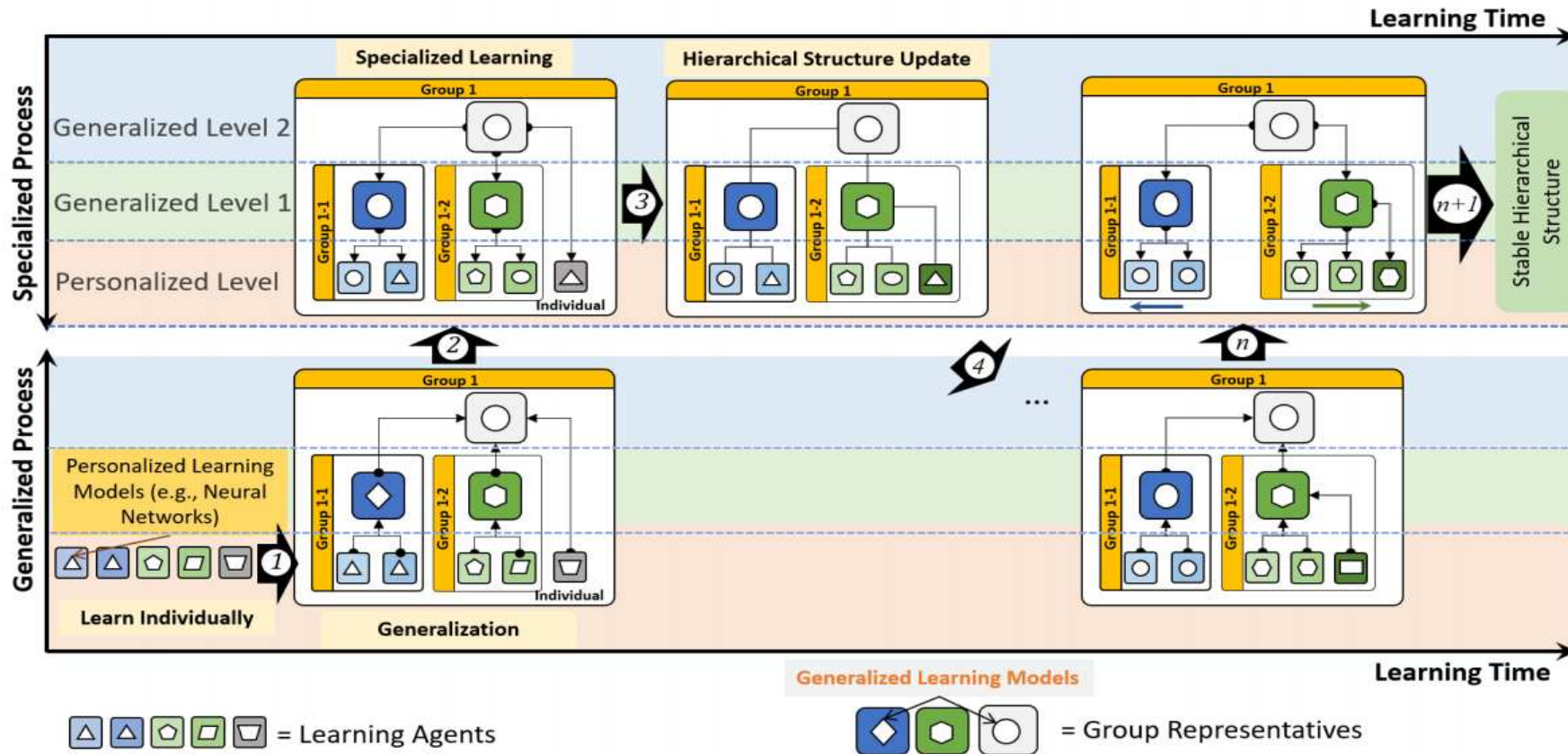
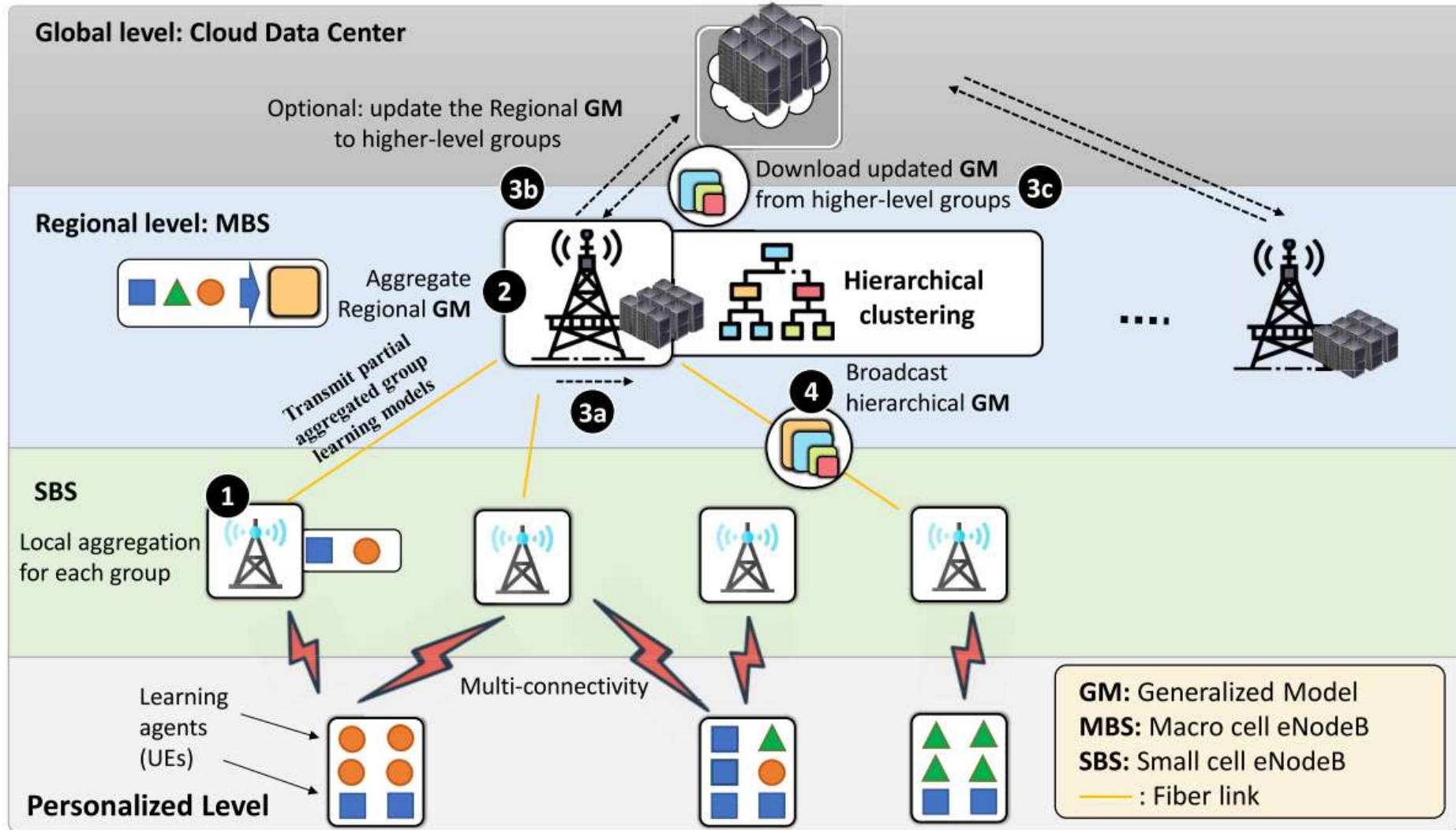


Fig. 4: The illustration of the transition in Dem-AI principle.

Edge-assisted Democratized Learning



Edge-assisted Democratized Learning

- **Decomposition of Edge-cloud Operation:**
 - Three steps of hierarchical aggregation in MEC servers at SBS, MBS, and Cloud.
 - MEC server at MBS takes the role of a **regional controller** and a cloud server acts as a **global coordinator** to manage this system.
 - **Regional edge learning using DemLearn algorithm [2]**
 - **Personalized Learning problem** at each learning agent:

$$\mathbf{PM}_n = \min. (1 - \beta_t)\mathbf{PLO}(\mathcal{D}_n) + \beta_t \sum_{h=1}^K \frac{1}{N_{g,n}^{(h)}} \mathbf{GM}_n^{(h)},$$

- **Hierarchical averaging** to construct groups' learning models (*level k=1*) and regional learning model (*level k=2*)

$$\mathbf{GM}^{(k)} = \sum_{c \in \mathcal{C}} \frac{N_{g,c}^{(k-1)}}{N_g^{(k)}} \mathbf{GM}_c^{(k-1)},$$

Update groups structure

PLO : personalized learning objective

[2] M. N. H. Nguyen, S. R. Pandey, D. T. Nguyen, N. H. Tran, E. N. Huh, W. Saad, and C. S. Hong, "Self-organizing democratized learning: Towards large-scale distributed learning systems," IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, DOI: [10.1109/TNNLS.2022.3170872](https://doi.org/10.1109/TNNLS.2022.3170872)

Algorithm 1 Democratized Learning (DemLearn)

- 1: **Input:** K, T, τ .
- 2: **for** $t = 0, \dots, T - 1$ **do**
- 3: **for** learning agent $n = 1, \dots, N$ **do**
- 4: Agent n receives and updates its local model from the higher-level generalized models $\mathbf{w}_{n,t}^{(1)}, \dots, \mathbf{w}_{n,t}^{(K)}$ of its super groups as

$$\mathbf{w}_{n,t+1}^{(0)} = (1 - \beta_t)\mathbf{w}_{n,t}^{(0)} + \frac{\beta_t}{B} \sum_{k=1}^K \frac{1}{N_{g,n}^{(k)}} \mathbf{w}_{n,t}^{(k)}, \text{ where } B = \sum_{k=1}^K \frac{1}{N_{g,n}^{(k)}}; \quad (6)$$

Local Model
Intializaiton

- 5: Agent n iteratively updates the personalized learning model $\mathbf{w}_n^{(0)}$ as an in-exact minimizer (i.e., *gradient based*) of the following problem:

$$\mathbf{w}_{n,t+1}^{(0)} \approx \arg \min_{\mathbf{w}} L_n^{(0)}(\mathbf{w} | \mathcal{D}_n) + \frac{\mu}{2} \sum_{k=1}^K \frac{1}{N_{g,n}^{(k)}} \|\mathbf{w} - \mathbf{w}_{n,t}^{(k)}\|^2; \quad (7)$$

Local Learning

- 6: Agent n sends updated learning model to the server;
- 7: **end for**
- 8: **if** ($t \bmod \tau = 0$) **then**
- 9: Server reconstructs the hierarchical structure by the clustering algorithm;
- 10: **end if**
- 11: Each group i at each generalized level k performs an update for its learning model as follows

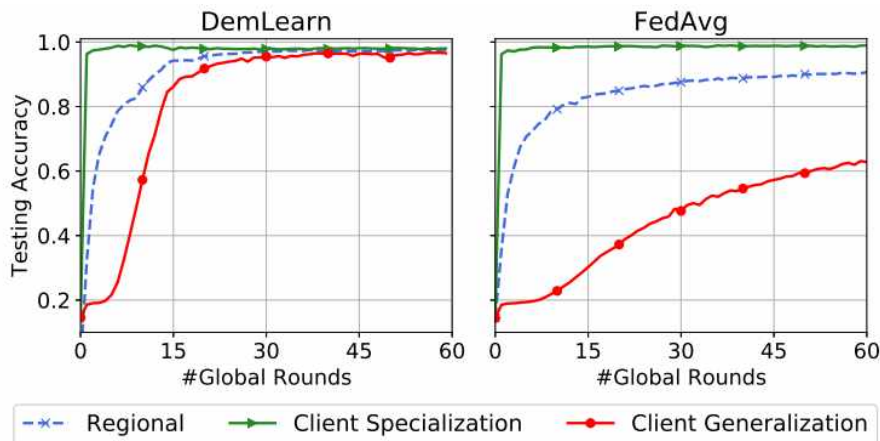
$$\mathbf{w}_{i,t+1}^{(k)} = \sum_{j \in \mathcal{S}_{i,k}} \frac{N_{g,j}^{(k-1)}}{N_{g,i}^{(k)}} \mathbf{w}_{j,t+1}^{(k-1)}; \quad (8)$$

Hierrachical Averaging

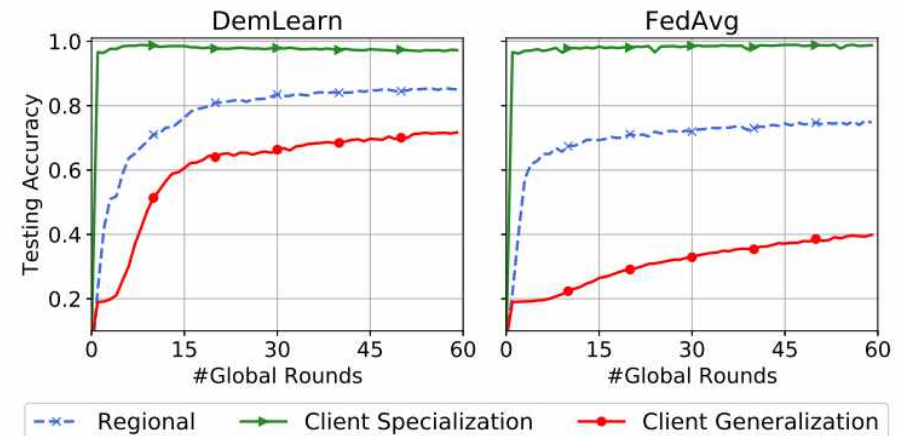
- 12: **end for**
-

Edge-assisted Democratized Learning

- Implementation of DemLearn algorithm is available at <https://github.com/nhatminh/Dem-AI/>
- Developing the device association and resource allocation for this system



(a) Experiment with MNIST dataset.



(b) Experiment with Fashion-MNIST dataset.

Regional Dem-AI learning performance for 100 users..

[2] M. N. H. Nguyen, S. R. Pandey, D. T. Nguyen, N. H. Tran, E. N. Huh, W. Saad, and C. S. Hong, “Self-organizing democratized learning: Towards large-scale distributed learning systems,” IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, DOI: [10.1109/TNNLS.2022.3170872](https://doi.org/10.1109/TNNLS.2022.3170872)

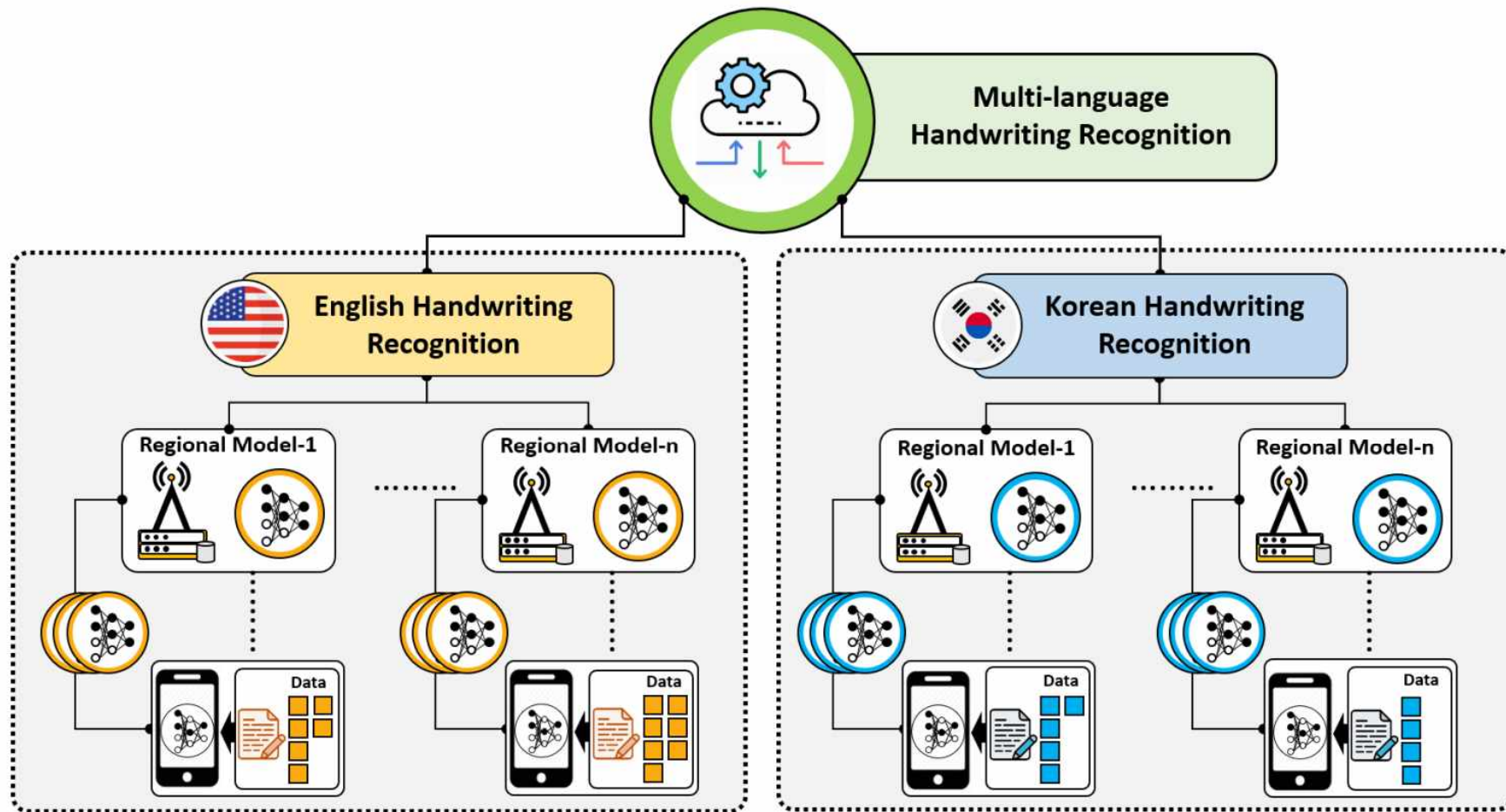


Fig. 7: An example of Dem-AI systems: Multi-language handwriting recognition.

Research Opportunities

- Develop a novel algorithm design for **multi-task distributed learning** setting
- Enhancement of **privacy** and **security issues** in distributed learning systems:
 - ✓ Information exploitation: reverse the personal data
 - ✓ Free-riding
 - ✓ Model/data poisoning attacks
- Optimization design regarding the synergy of **Resource Allocation** and **Learning Performance**
 - ✓ Group structure changing
- **Future Personalized Applications**
 - ✓ Learn the **unique features** and **personalized characteristics** during the daily activities of each user and make appropriate decisions
 - ✓ VR/AR services: regional edge intelligence is used to predict the future gaze direction, motion, and mobility patterns, which are exceedingly different among users.

Research Opportunities

- Develop a novel algorithm design for **multi-task distributed learning** setting
- Enhancement of **privacy** and **security issues** in distributed learning systems:
 - ✓ Information exploitation: reverse the personal data
 - ✓ Free-riding
 - ✓ Model/data poisoning attacks
- Optimization design regarding the synergy of **Resource Allocation** and **Learning Performance**
 - ✓ Group structure changing
- **Future Personalized Applications**
 - ✓ Learn the **unique features** and **personalized characteristics** during the daily activities of each user and make appropriate decisions
 - ✓ VR/AR services: regional edge intelligence is used to predict the future gaze direction, motion, and mobility patterns, which are exceedingly different among users.

- Introduction: Motivation of Federated Learning
- Federated Learning
 - FL Formulation
 - FL Algorithms
 - Ongoing Research Problems
 - Federated Learning: at the Edge
 - Summary
- Democratized Learning
 - Introduction
 - Key Components
 - Ongoing Research Problems
- **Multimodal Federated Learning**
 - Introduction
 - Key Components
 - Ongoing Research Problems

What is multimodal learning ?

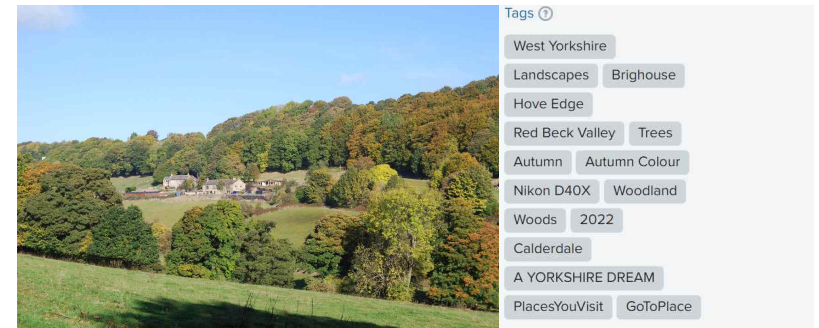
- A modality is just sort of a kind of data and we have achieved lots of success in single modality: classify digits for MNIST, speech recognition with audio waveforms



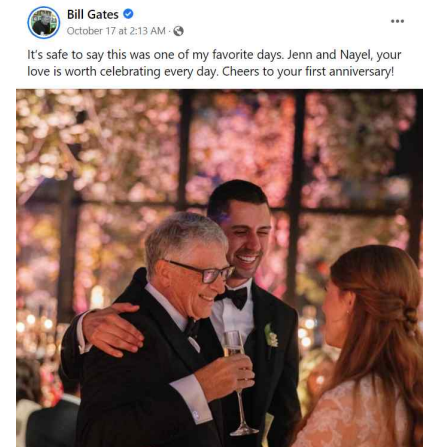
- **Multimodal learning** is a learning scheme that involves multiple modalities, which can manifest itself in different ways:
 - Input is one modality, output is another
 - Multiple modalities are learned jointly
 - One modality assists in the learning of another

- Most of the data we see in our daily lives is in **multiple modalities** at the same time.

- On Flickr, A picture associated with some tags



- On Facebook, you might see a post that has a **text description and image**.

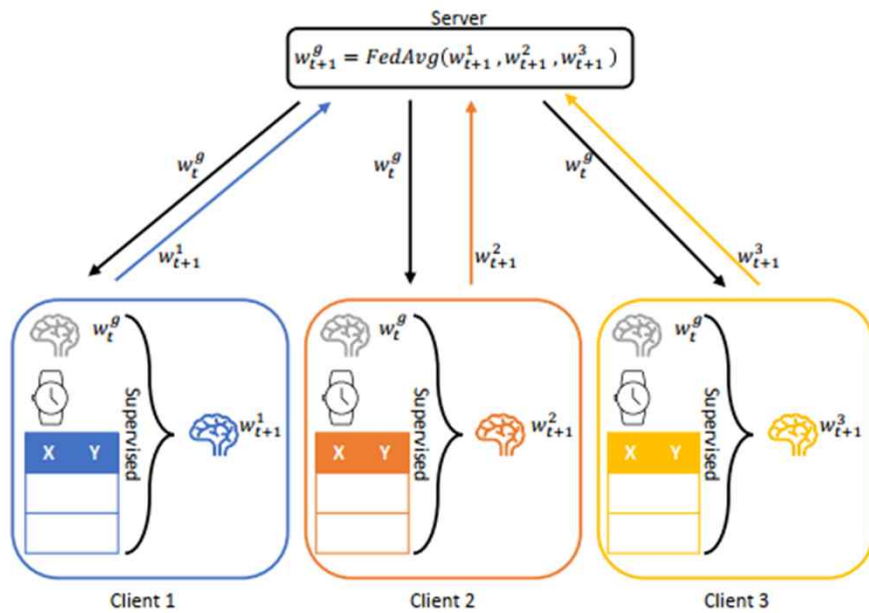


- With the available of modalities data, we want to take advantage of it and study how can we **improve performance of downstream tasks**.

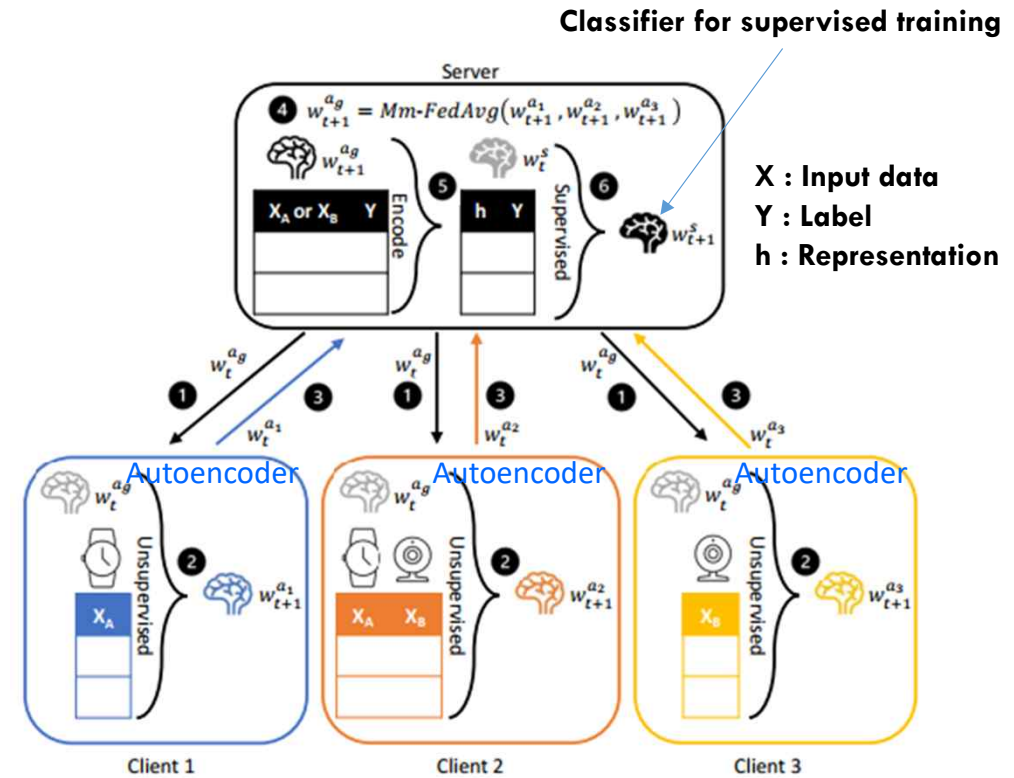
Motivation of multimodal federated learning

- Despite many advantages in preserving privacy, existing FL methods consider a scenario where clients hold only single-modal data, restricting the use of multimodal data in various equipment.
 - IoT applications often deploy **different types of devices** (e.g, smartphones, smartwatches)
- The **common features** from multimodal data provides more accuracy and robustness performance than single-modal data.
- The design for FL framework using multimodal data becomes more **practical** where users own data generated from multiple data sources.

Traditional FL vs Multimodal FL



Federated Averaging (FedAvg)



Multimodal Federated Averaging (MM-FedAvg)

Main Components

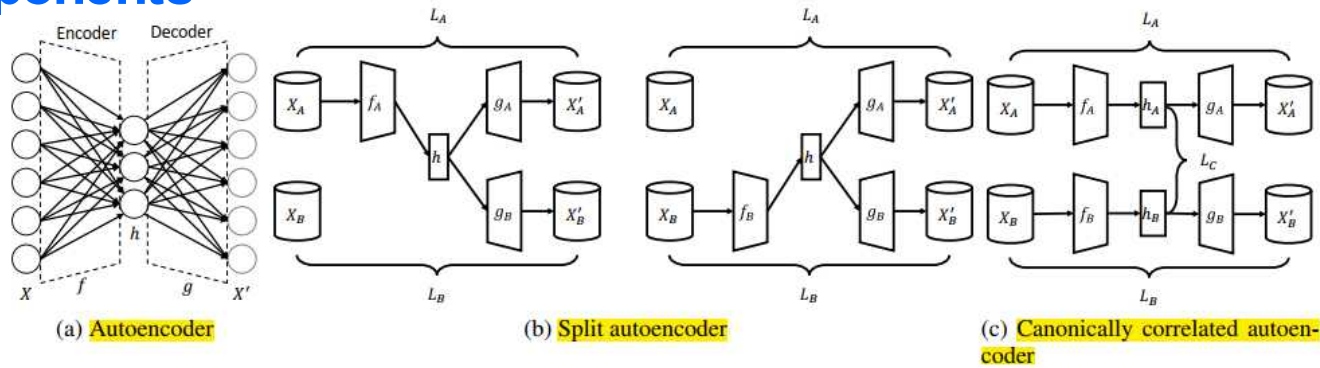


Figure 2: In an autoencoder (a), an encoder f maps input data X into a hidden representation h . A decoder g maps h into a reconstruction X' . In split autoencoders (b), for aligned input (X_A, X_B) from two modalities, data from one modality are input into its encoder to generate an h , which is then used to reconstruct the data for both modalities through two decoders. In a canonically correlated autoencoder (c), data from both modalities are input into their encoders to generate two representations.

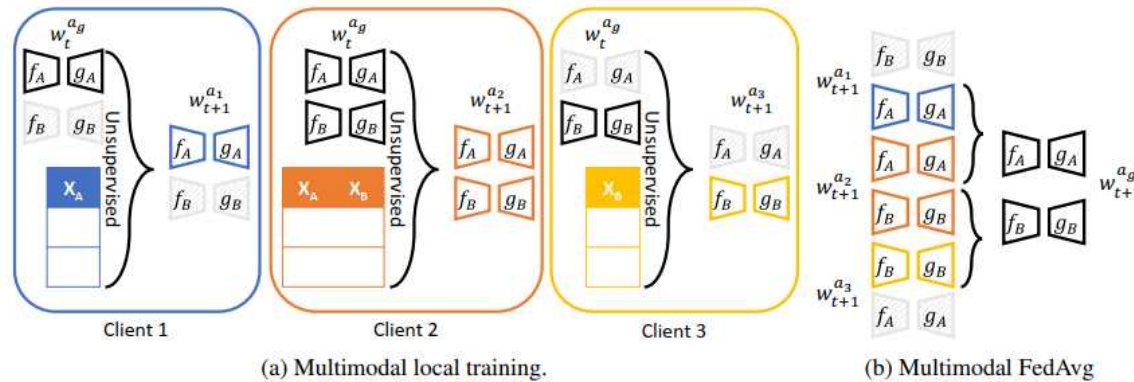


Figure 3: During local training (a), clients only update the f and g that are related to the modalities of their data. When conducting multimodal FedAvg (b) on the server, only the updated parts of each local model will be aggregated.

Research Opportunities

- Develop a novel design for **multimodal FL** setting with more modalities and more powerful backbone models.
- Develop the **edge AI** mechanisms for multimodal FL
 - ✓ Large model is not practical for on-device learning
 - ✓ Can be applicable for more downstream tasks (vision, language tasks).

- We have explored state-of-the-art distributed learning frameworks such as **Federated Learning, Democratized Learning and Multimodal Federated Learning** to provide solutions for large-scale private and personal learning services
- We have introduced several federated learning algorithms, reference implementation, and applications
- We have introduced concepts and an initial implementation of democratized learning
- We have introduced distributed learning frameworks in a variety of research domains and applications.

- [1] Hard Andrew et al. "Federated learning for mobile keyboard prediction." *arXiv preprint arXiv:1811.03604* (2018).
- [2] McMahan, Brendan, et al. "Communication-efficient learning of deep networks from decentralized data." *Artificial Intelligence and Statistics*. PMLR, 2017.
- [3] Li, Tian, et al. "Federated optimization in heterogeneous networks." *arXiv preprint arXiv:1812.06127* (2018).
- [4] Reddi, Sashank, et al. "Adaptive Federated Optimization." *arXiv preprint arXiv:2003.00295* (2020).
- [5] K. Bonawitz, et al. "Practical Secure Aggregation for Privacy-Preserving Machine Learning." *CCS 2017*.
- [6] H. B. McMahan, et al. "Learning Differentially Private Recurrent Language Models" *arXiv preprint arXiv:1710.06963* (2017).
- [7] Bonawitz, Keith, et al. "Towards federated learning at scale: System design." *arXiv preprint arXiv:1902.01046* (2019).
- [8] Nguyen H. Tran, Wei Bao, Albert Zomaya, Minh N.H. Nguyen and Choong Seon Hong, "Federated Learning over Wireless Networks: Optimization Model Design and Analysis," *IEEE International Conference on Computer Communications (INFOCOM 2019)*, April 29 - May 2, 2019, Paris, France.
- [9] C. Ma, J. Konecny, M. Jaggi, V. Smith, M. I. Jordan, P. Richtarik, and M. Takac, "Distributed Optimization with Arbitrary Local Solvers," *Optimization Methods Software*, vol. 32, no. 4, pp. 813–848, Jul. 2017.
- [10] Wang, Sihua, Mingzhe Chen, Changchuan Yin, Walid Saad, Choong Seon Hong, Shuguang Cui, and H. Vincent Poor., "Federated Learning for Task and Resource Allocation in Wireless High Altitude Balloon Networks," *arXiv:2003.09375*, March 2020.
- [11] M. N. H. Nguyen, S. R. Pandey, K. Thar, N. H. Tran, M. Chen, W. Saad, and C. S. Hong, "Distributed and democratized learning: Philosophy and research challenges," *arXiv:2003.09301*, 2020.
- [12] M. N. H. Nguyen, S. R. Pandey, D. T. Nguyen, N. H. Tran, E. N. Huh, W. Saad, and C. S. Hong, "Self-organizing democratized learning: Towards large-scale distributed learning systems," *arXiv preprint arXiv:2007.03278* (2020).
- [13] Yuchen Zhao, Payam Barnaghi, and Hamed Haddadi. Multimodal federated learning on iot data. In *2022 IEEE/ACM Seventh International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pp. 43–54. IEEE, 2022

Thank You!

Q & A