Federated and Split Learning

Prof. Joongheon Kim
Korea University, School of Electrical Engineering
Artificial Intelligence and Mobility Laboratory
https://joongheon.github.io
joongheon@korea.ac.kr

01. Introduction

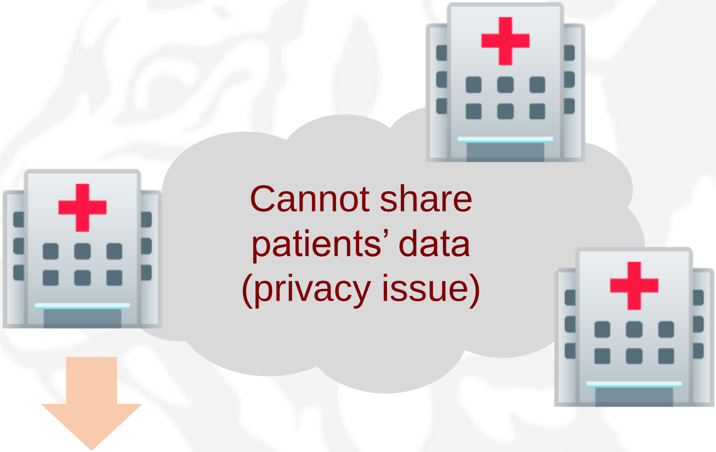
02. Federated Learning vs. Split Learning

03. Split Learning for AI: Spatio-temporal Split Learning

04. SplitFed: Federated Learning Meets Split Learning

05. Mixup

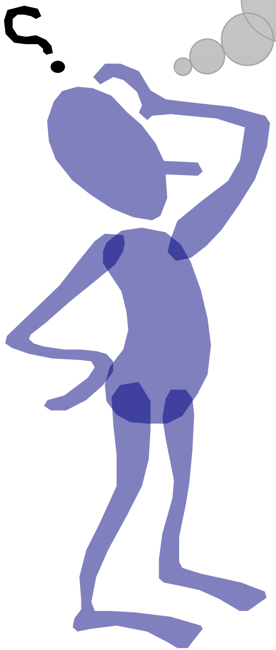
- It's not possible to gather all data in a single hospital/medical-cloud for deep learning computation (due to patients' privacy).
  Then, following problems can occur:
  - **Overfitting** in each hospital
  - **Training Performance Degradation**



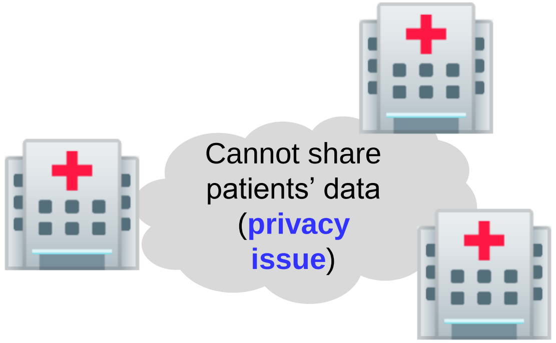
Goals

- Maintaining Deep Learning Computation Performance
- Prohibiting Duplicated Patients' Data

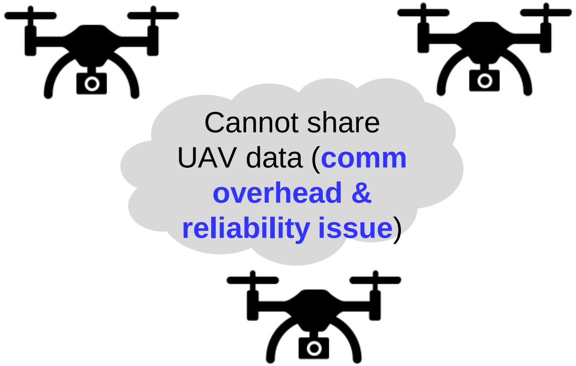
Having all data in a single storage is very hard in real-world applications!



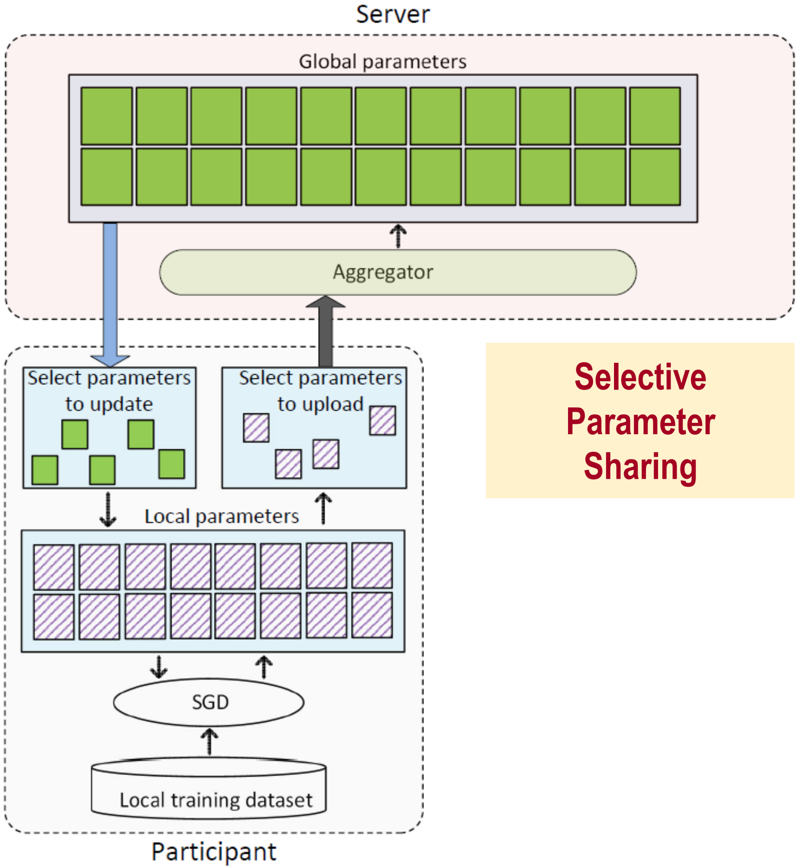
Medical Platforms



Network Platforms



- **Collaborative Deep Learning**
  - How it works?
    - All clouds share the model at first.
    - Each cloud trains its own model (Data is not shared among clouds for privacy-preserving).
    - Each cloud shares weight values (not the data itself).
      → **Selective Parameter Sharing**
  - Disadvantages
    - Performance degradation
    - Synchronization (No network delays are assumed.)

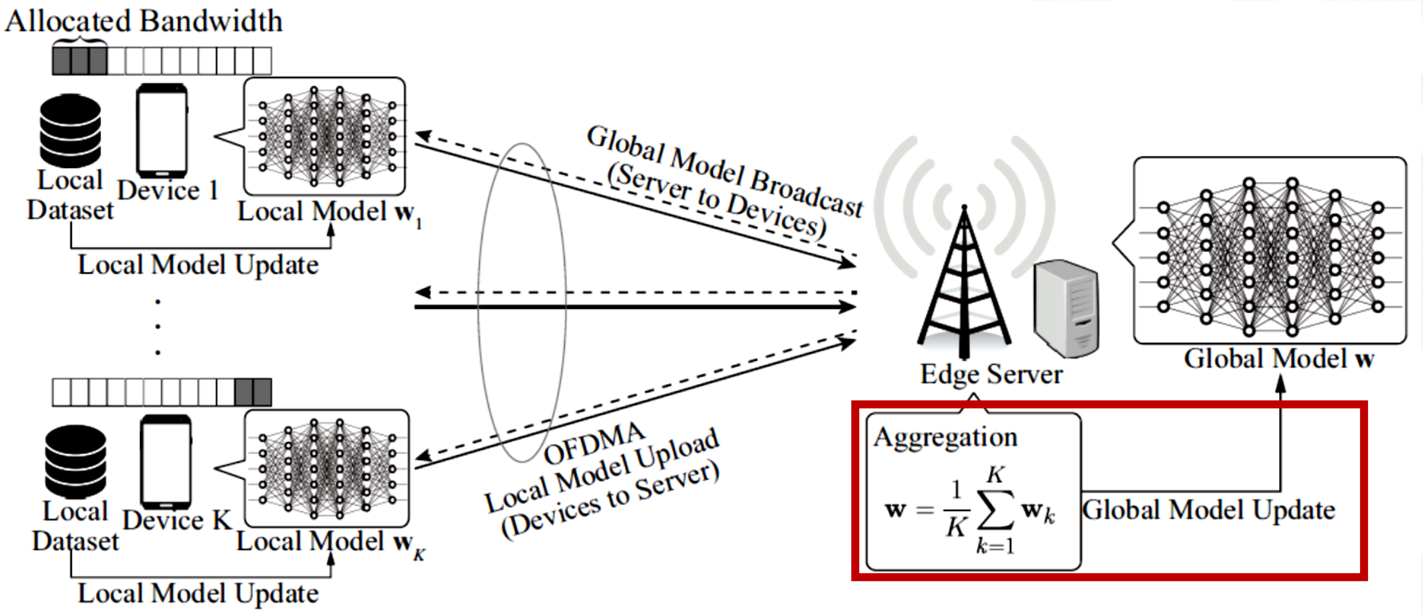


Selective Parameter Sharing

R. Shokri and V. Shmatikov, "Privacy-Preserving Deep Learning," *ACM CCS 2015*. (Citation: 1200+)

Artificial Intelligence and Mobility Lab

- Energy-Efficient FL System, Intuitive Averaging



Q. Zeng, Y. Du, K. K. Leung, and K. Huang, "Energy-Efficient Radio Resource Allocation for Federated Edge Learning," https://arxiv.org/abs/1907.06040, July 2019.

- Federated Averaging
  - Weighted Averaging

Algorithm 1 Federated averaging algorithm

Require: Local minibatch size B, number of participants m per iteration, number of local epochs E, and learning rate η.
Ensure: Global model $\mathbf{w}_G$.

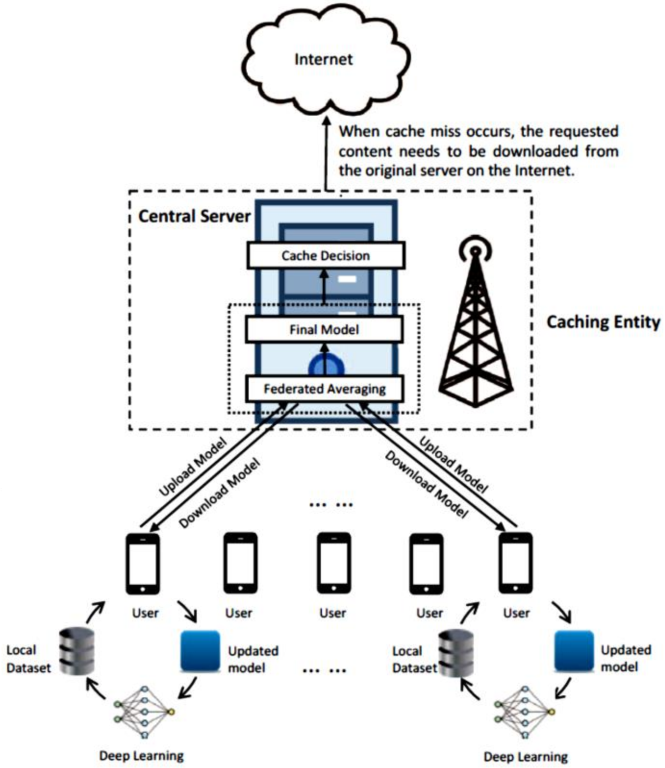
```
[Participant i]
LocalTraining(i, w):
Split local dataset D_i to minibatches of size B which are included into the set 𝓑_i.
for each local epoch j from 1 to E do
    for each b ∈ 𝓑_i do
        w ← w − ηΔL(w; b)      (η is the learning rate and ΔL is the gradient of L on b.)
    end for
end for

[Server]
Initialize w_G^0
for each iteration t from 1 to T do
    Randomly choose a subset 𝒮_t of m participants from 𝒩
    for each partipant i ∈ 𝒮_t parallely do
        w_i^{t+1} ← LocalTraining(i, w_G^t)
    end for
    w_G^t = (1 / Σ_{i∈𝒩} D_i) Σ_{i=1}^{N} D_i w_i^t      (Averaging aggregation)
end for
```

$$\boxed{\mathbf{w}_G^t = \frac{1}{\sum_{i\in\mathcal{N}} D_i}\sum_{i=1}^{N} D_i \mathbf{w}_i^t \qquad \text{(Averaging aggregation)}}$$

W. Yang, *et. al.*, "Federated Learning in Mobile Edge Networks: A Comprehensive Survey," https://arxiv.org/abs/1909.11875v1, September 2019.

Artificial Intelligence and Mobility Lab

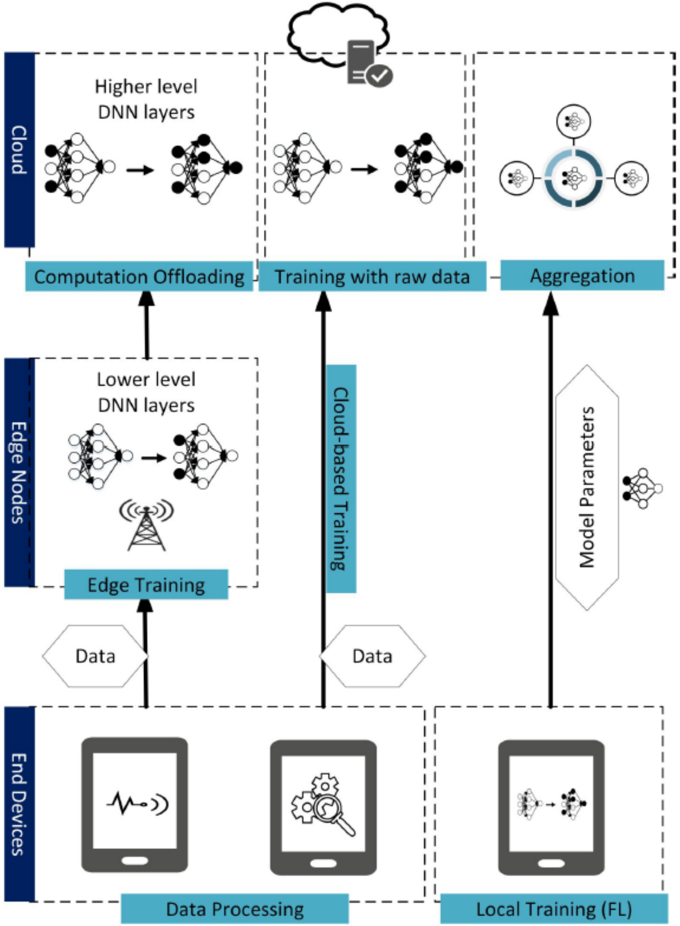
- System Model (Initial Starting)



Z. Yu, J. Hu, G. Min, H. Lu, Z. Zhao, H. Wang, and N. Georgalas, "Federated Learning Based Proactive Content Caching in Edge Computing," in *Proc. of IEEE GLOBECOM*, Abu Dhabi, UAE, December 2018.

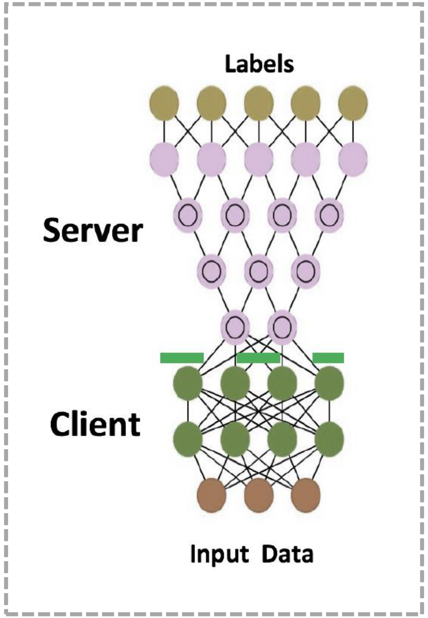
Federated Learning: FL Applications to Networks

- Edge AI approach brings AI processing closer to where data is produced.
- FL allows training on devices where the data is produced.

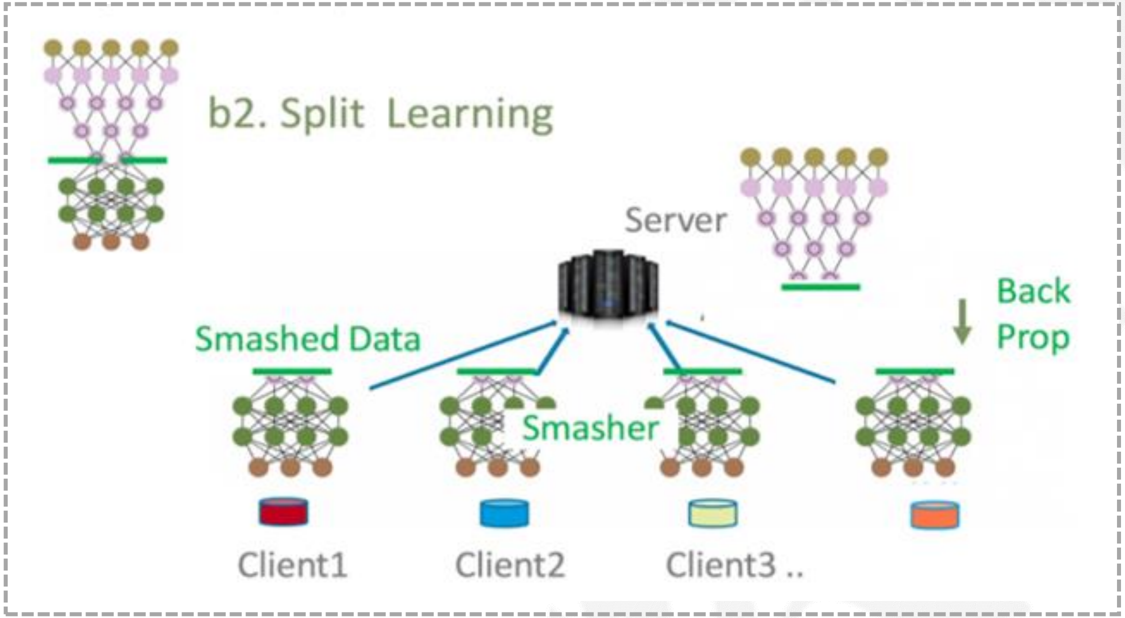


W. Yang, *et. al.*, "Federated Learning in Mobile Edge Networks: A Comprehensive Survey," https://arxiv.org/abs/1909.11875v1, September 2019.

- Communication efficiency of split learning and federated learning



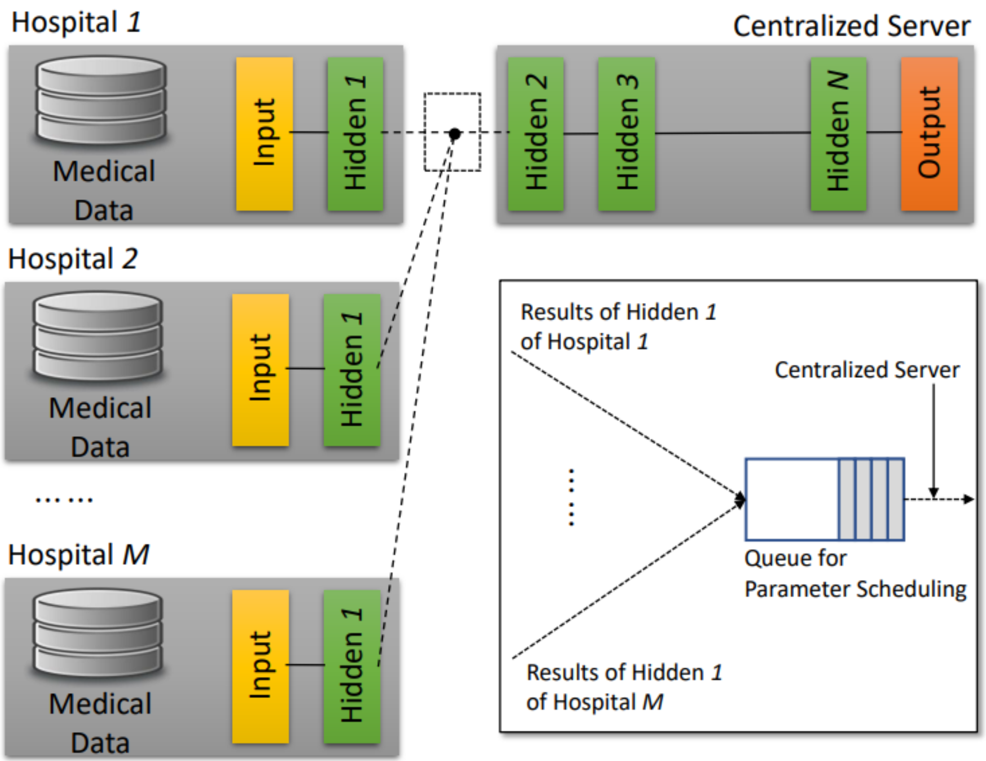
Vanilla split learning setup



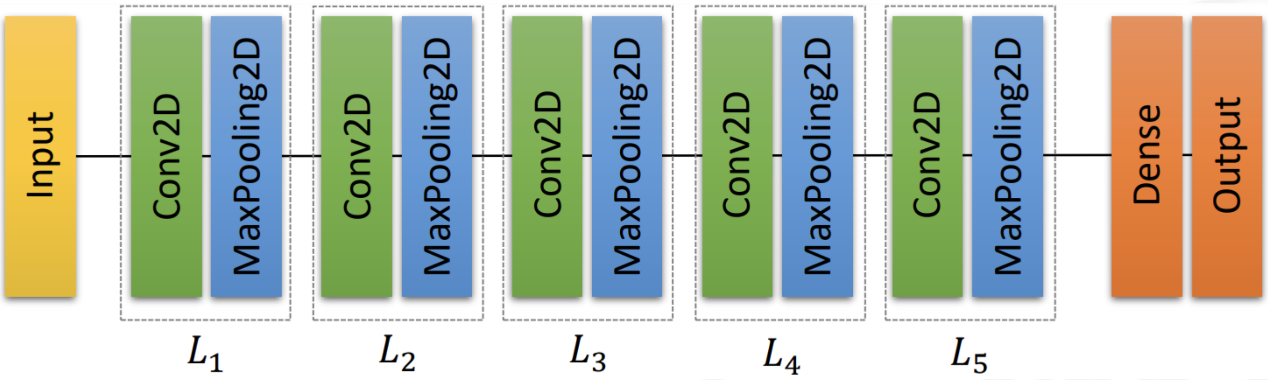
Split learning setup with multiple clients and a server

Singh, P. Vepakomma, O. Gupta, and R. Raskar, "Detailed Comparison of Communication Efficiency of Split Learning and Federated Learning," https://arxiv.org/abs/1909.09145, September 2019

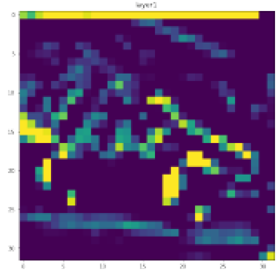
Spatio-temporal Split Learning: Concept



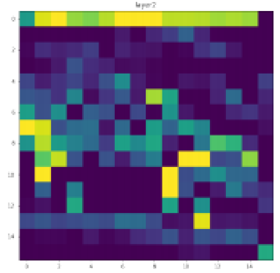
- Is an innovative deep learning approach to **preserve the privacy** of personal health data through split learning
- Resolves the issue of **data-imbalance** and **overfitting**
- Our model is **versatile**: proposed split learning works with both numerical and image data
- Utilize **real-world data** provided by SNUH



(a) Original



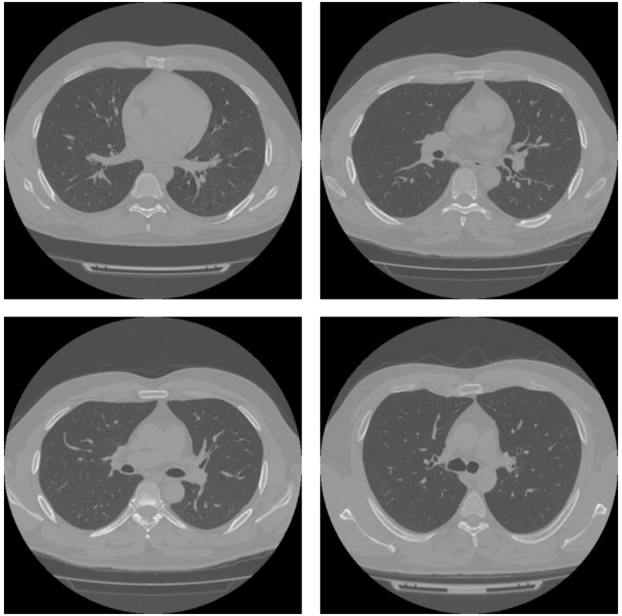
(b) Conv2D in L_1



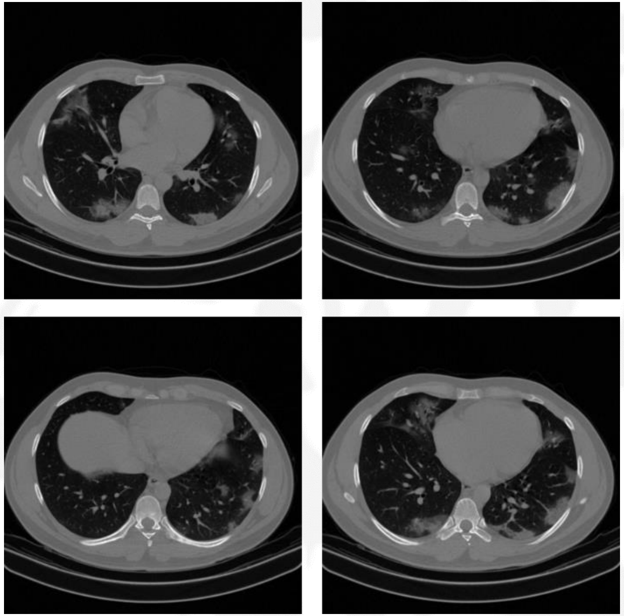
(c) L_1

Layers at end-systems	Accuracy
Nothing (All layers are in the server)	71.09 %
L_1	68.18 %
L_1, L_2	67.92 %
L_1, L_2, L_3	66.00 %
L_1, L_2, L_3, L_4	65.66 %

Medical Data for Evaluation: COVID-19 Chest CT Scan

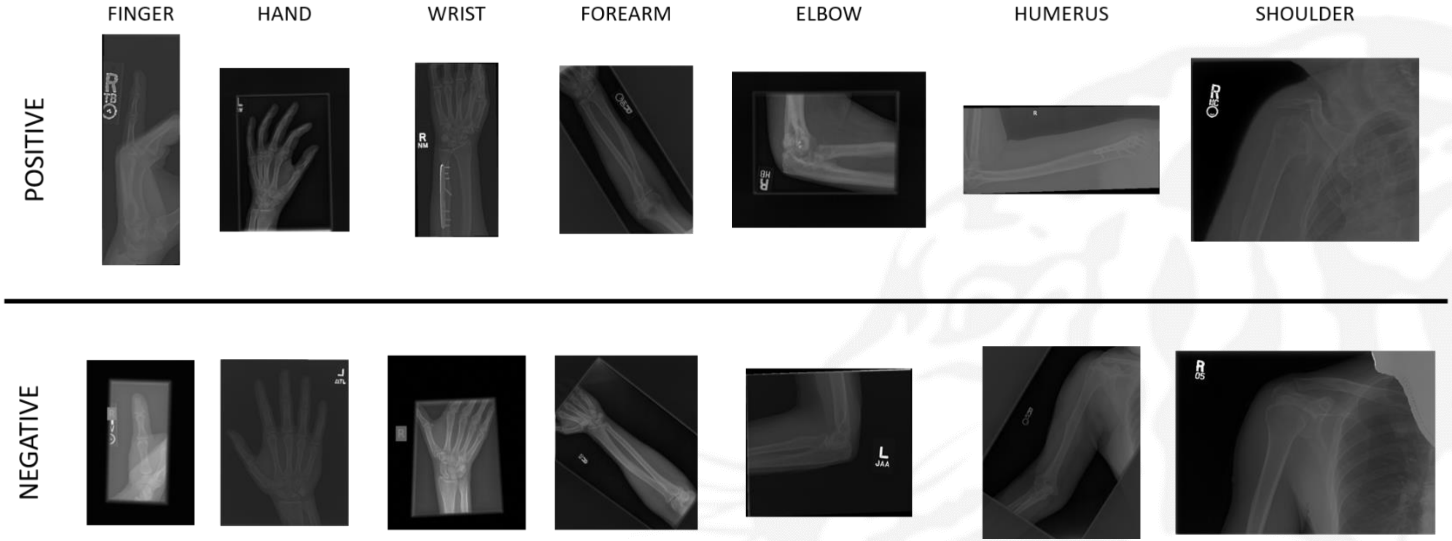


COVID-19 patient CT scan images
(7,593 images)



Non-COVID CT scan images
(6,893 images)

Medical Data for Evaluation: MURA



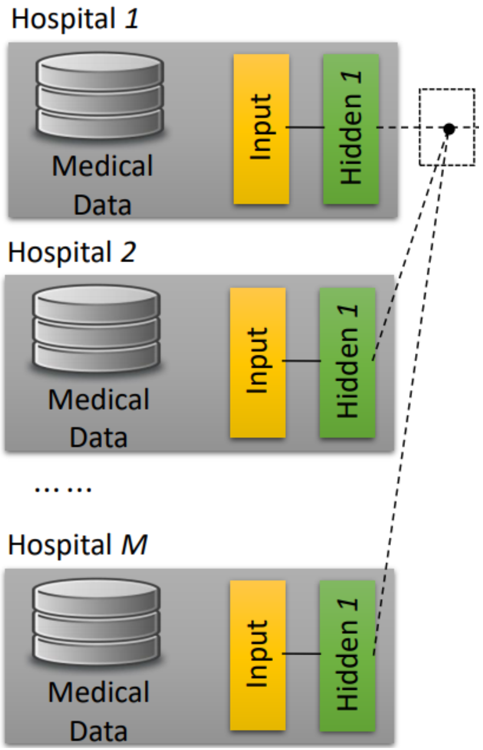
Category	Total case	Positive case	Negative case
Finger	5,106	1,968	3,138
Hand	5,543	1,484	4,059
Wrist	9,752	3,987	5,765
Forearm	1,825	661	1,164
Elbow	4,931	2,006	2,925
Humerus	1,272	599	673
Shoulder	8,379	4,168	4,211

Medical Data for Evaluation: Cholesterol Data

Index	Age	Sex	Height	Weight	TC	HDL-C	TG
Case 1	62	Male	175.0	68.20	178	50	83
Case 2	80	Male	168.0	78.70	104	22	148
Case 3	56	Male	178.0	80.85	207	55	158
Case 4	73	Female	144.8	50.45	144	30	100
Case 5	66	Male	167.7	62.80	138	60	74

407,540 patients' medical record
provided by SNUH

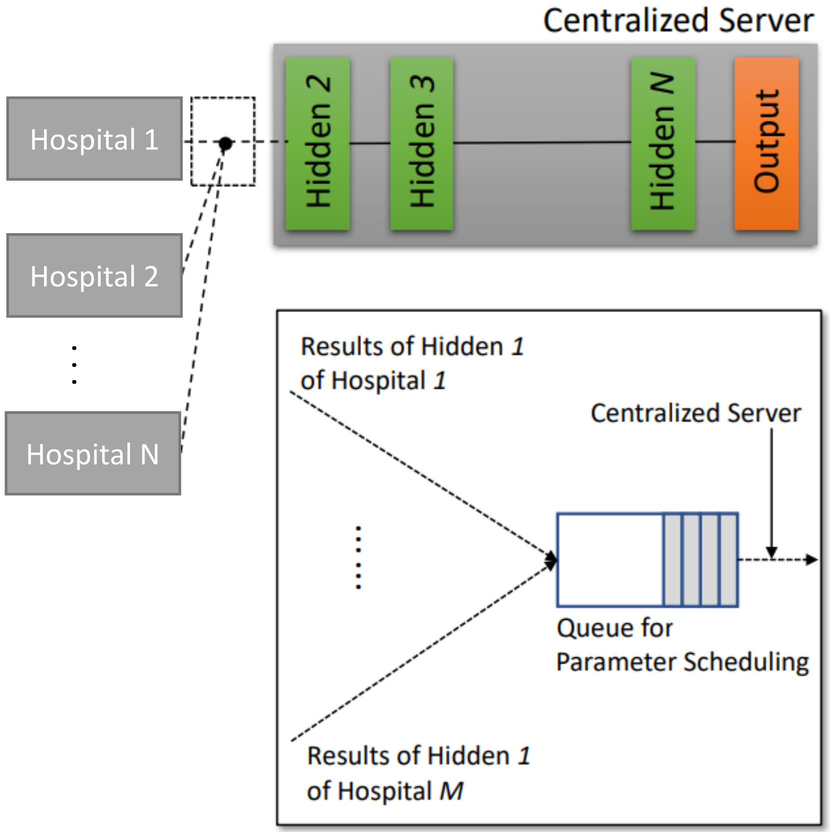
Algorithm for Client



Require: Batch size B, clients C, number of clients n, number of epoch E, learning rate α, target value y, predicted value $\hat{y}$, input data I, number of input data I_n, number of label l_a, and output convolution layer O^l.

procedure CLIENT
For Client = $\{1, \cdots, n\}$ **do**
  For *Training data set* = $\{1, \cdots, x\}$ **do**
    Calculate Conv. ▷ Eq. (1)
    ▷ $f_c = \text{Conv}(O^{l-1}, w^l, I_n, l_a) = net^l_{I_n, l_a}$
  ▷ Send feature f_c to server.
  End For
End For

Algorithm for Server



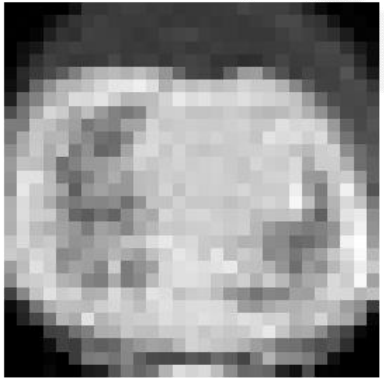
Require: Batch size B, clients C, number of clients n, number of epoch E, learning rate α, target value y, predicted value $\hat{y}$, input data I, number of input data I_n, number of label l_a, and output convolution layer O^l.

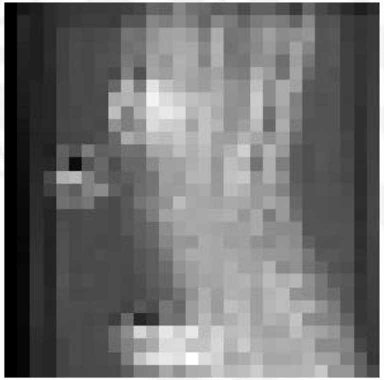
```
procedure SERVER
  Receive input data from client : f_c
  Concatenate all features Σ_{k=1}^{n} f_c^k
  For epoch = 1, E do
      For Training data set do
          Calculate Conv, and Pool                          ▷ Eq. (1), Eq. (2)
          ▷ f_c = Conv(O^{l-1}, w^l, I_n, l_a) = net^l_{I_n,l_a}
          ▷ f_p = Pool(f_c, I_m, I_a)
          ▷ ŷ is calculated using I, (1), and (2).
      ▷ Calculate loss.                                     ▷ Eq. (3)
      ▷ Update the model: update weights w ← w · α.
      End For
  End For
```

Preserving Privacy

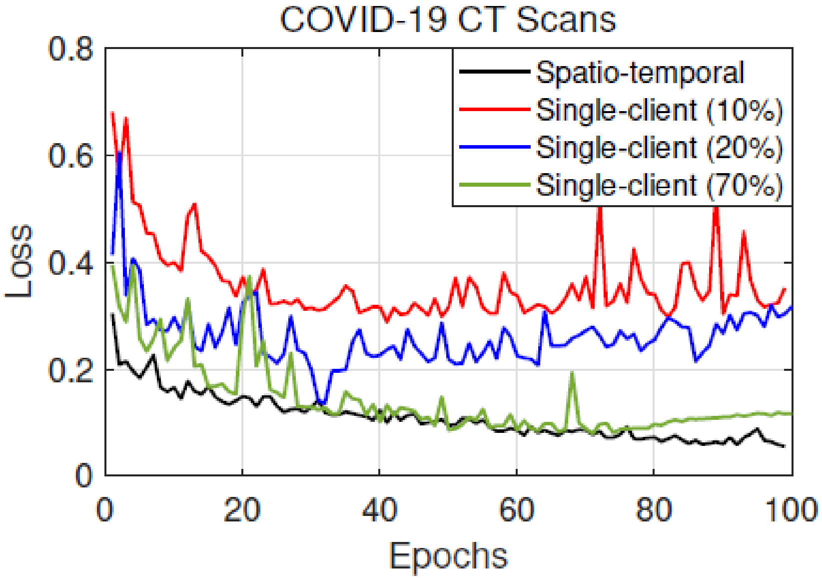
Original image

Image after passing through one hidden layer

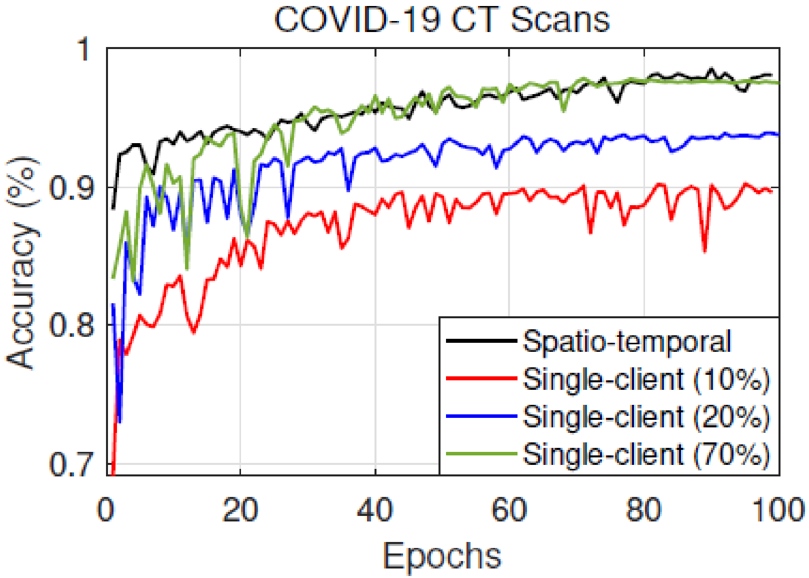




Experiment Result: COVID-19 chest CT scan

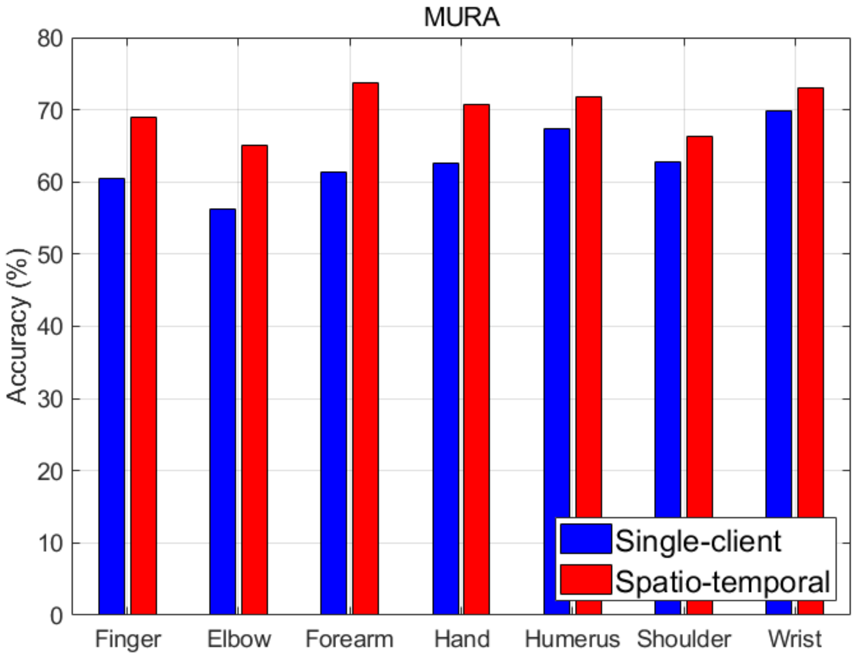


(a) Loss



(b) Accuracy

Experiment result: MURA

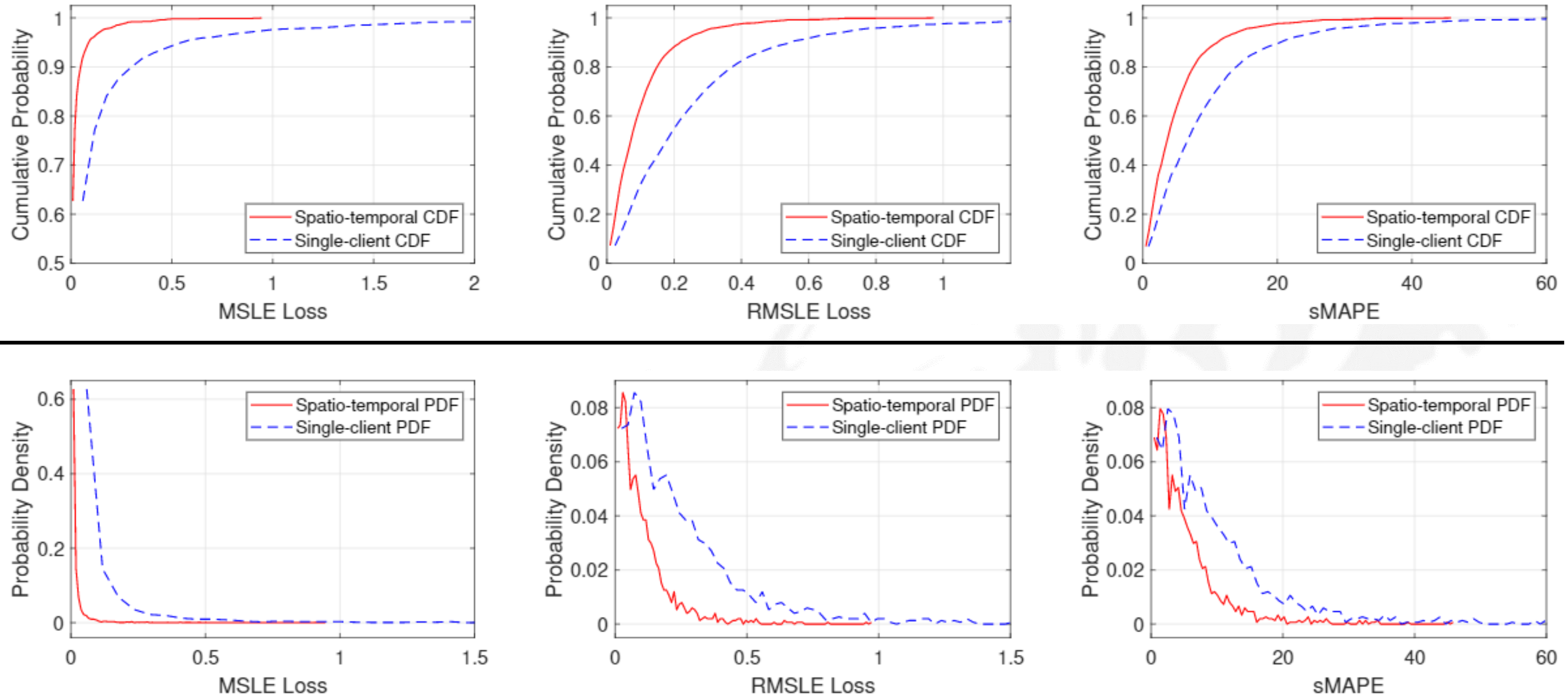


		Finger	Elbow	Forearm	Hand	Humerus	Shoulder	Wrist
Accuracy (%)	Single-client	60.5	56.3	61.4	62.6	67.3	62.8	69.9
	Spatio-temporal	68.9	65.1	73.7	70.8	71.8	66.4	73.1

Experiment result: Cholesterol data

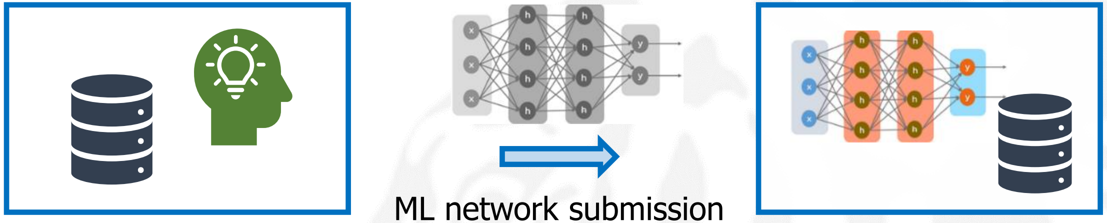
CDF

PDF



Artificial Intelligence and Mobility Lab

- SplitFed: Federated Learning Meets Split Learning
  - Model to data approach

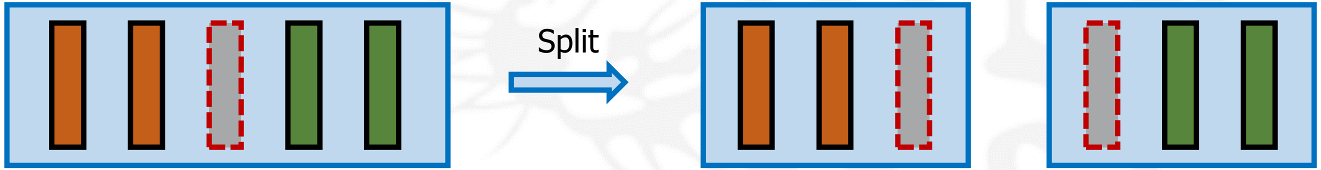


Modeler/Analyst

ML network submission for training

Data curator platform (private)

  - Network split



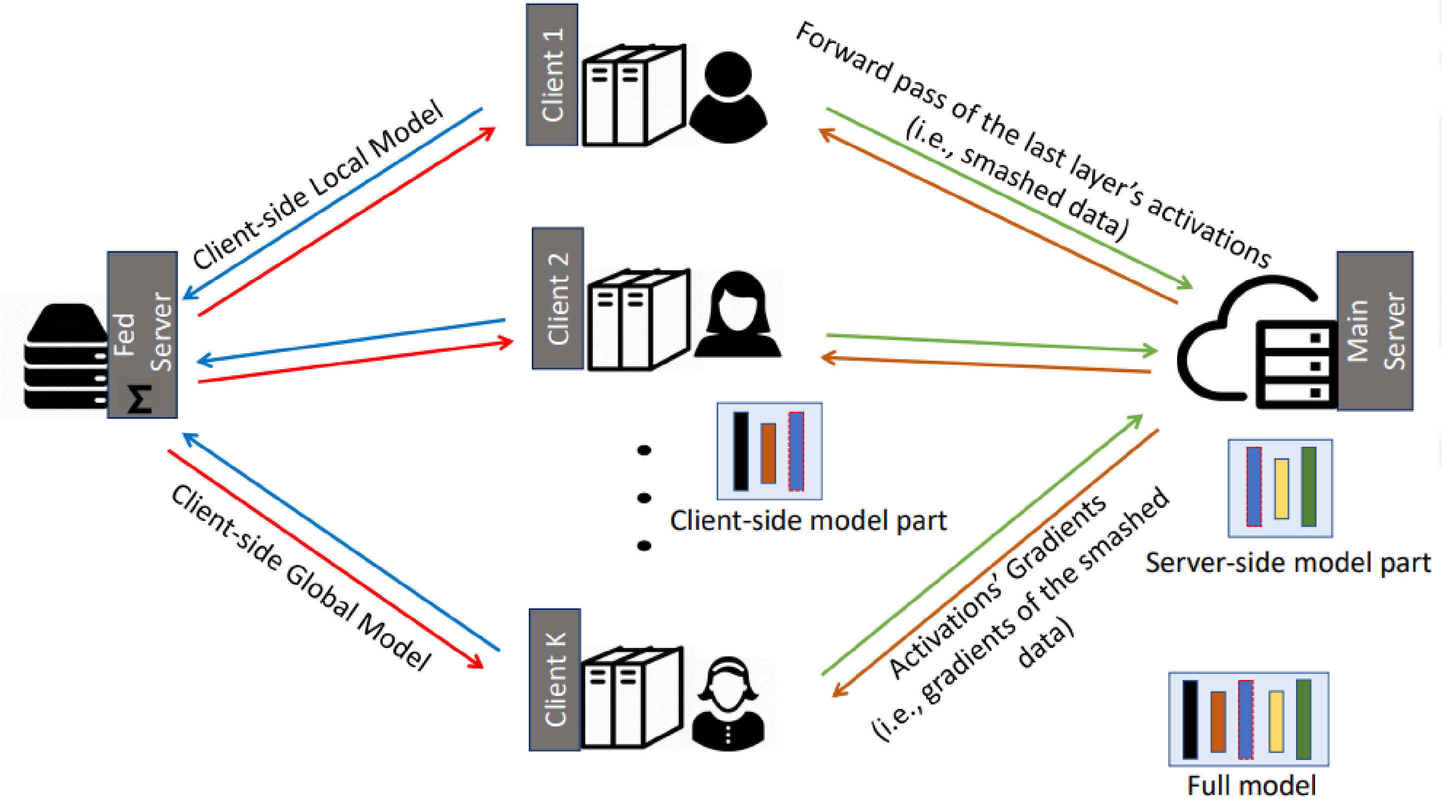
Full model

Client-side model portion

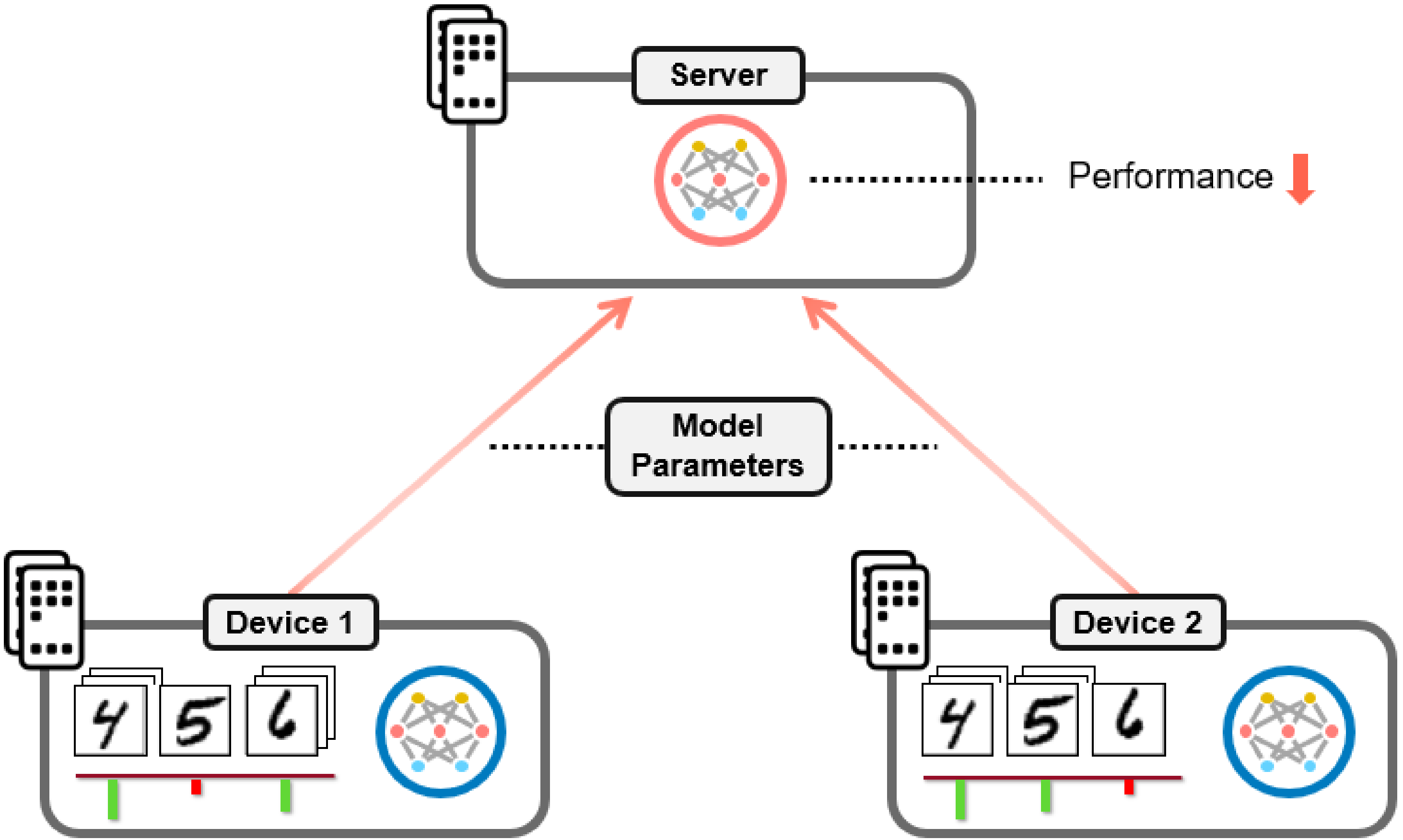
Server-side model portion

C. Thapa, M.A.P. Chamikara, and S. Camtepe, "SplitFed: When Federated Learning Meets Split Learning," https://arxiv.org/abs/2004.12088, April 2020.

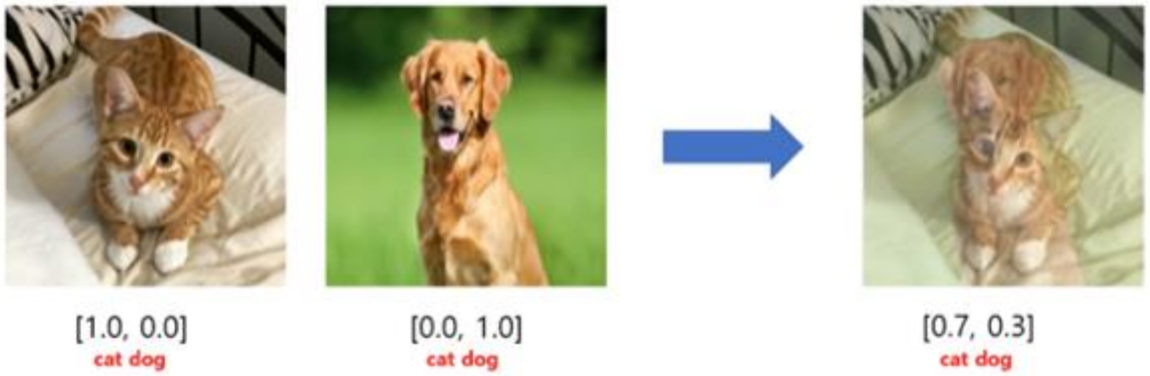
- SplitFed: Federated Learning Meets Split Learning

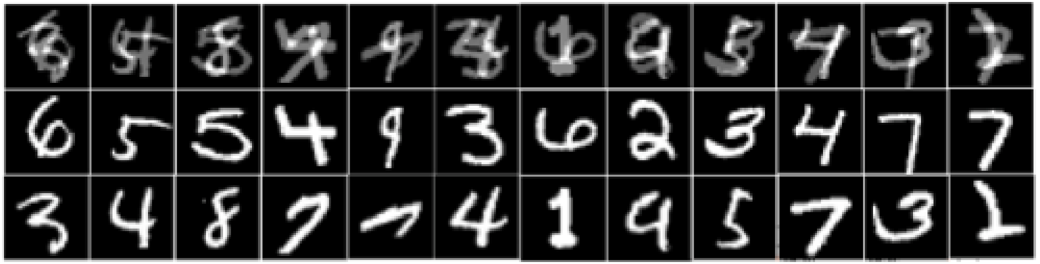


Issue (non-IID)



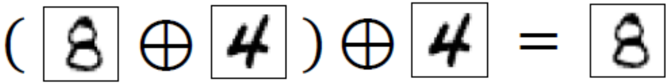
- Data dependent augmentation technique.
- Create a new sample by weighted linear interpolation of the images and labels of two data respectively.



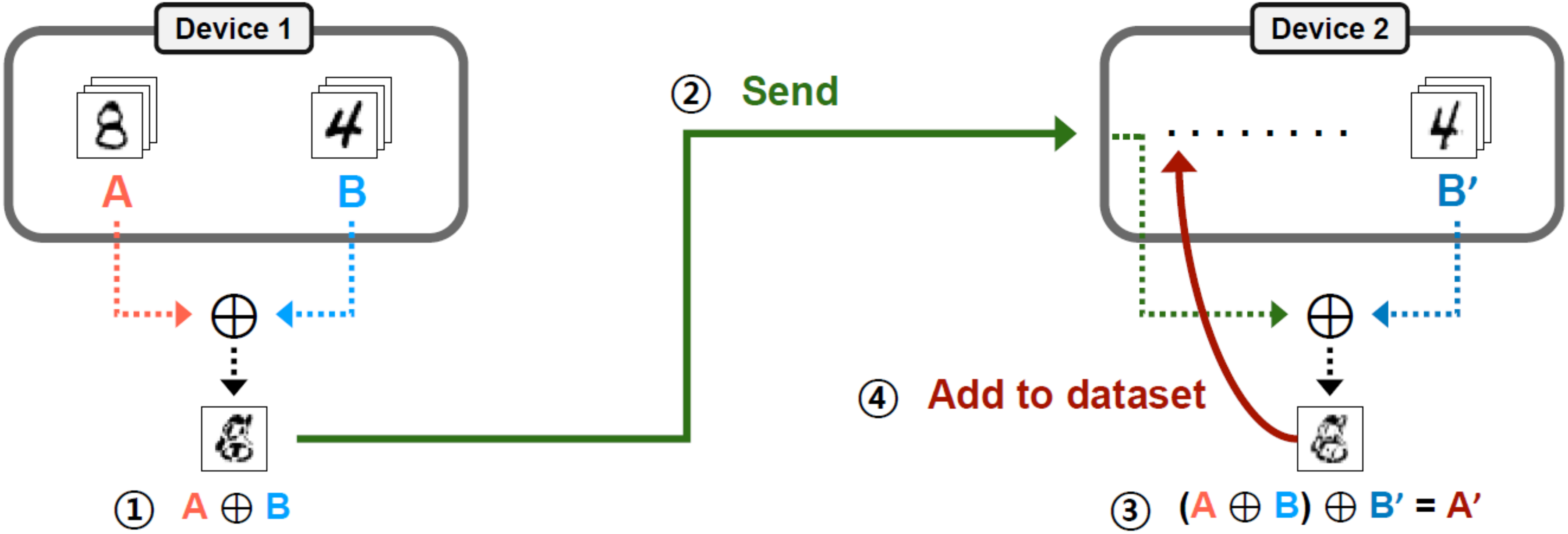


XOR Mixup

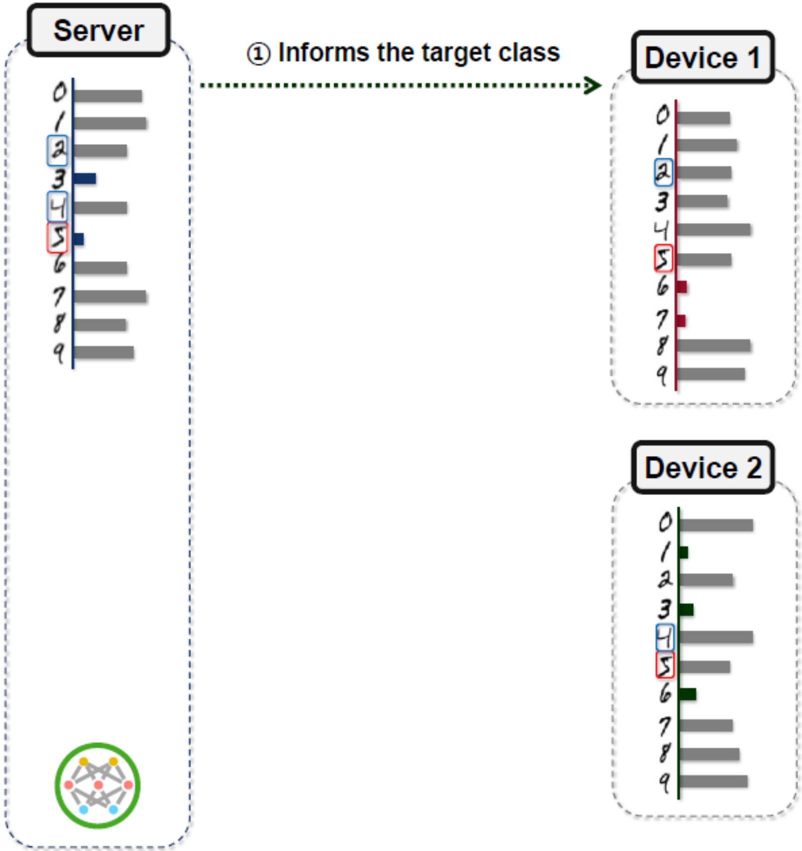
Key idea: $(A \oplus B) \oplus B = A$



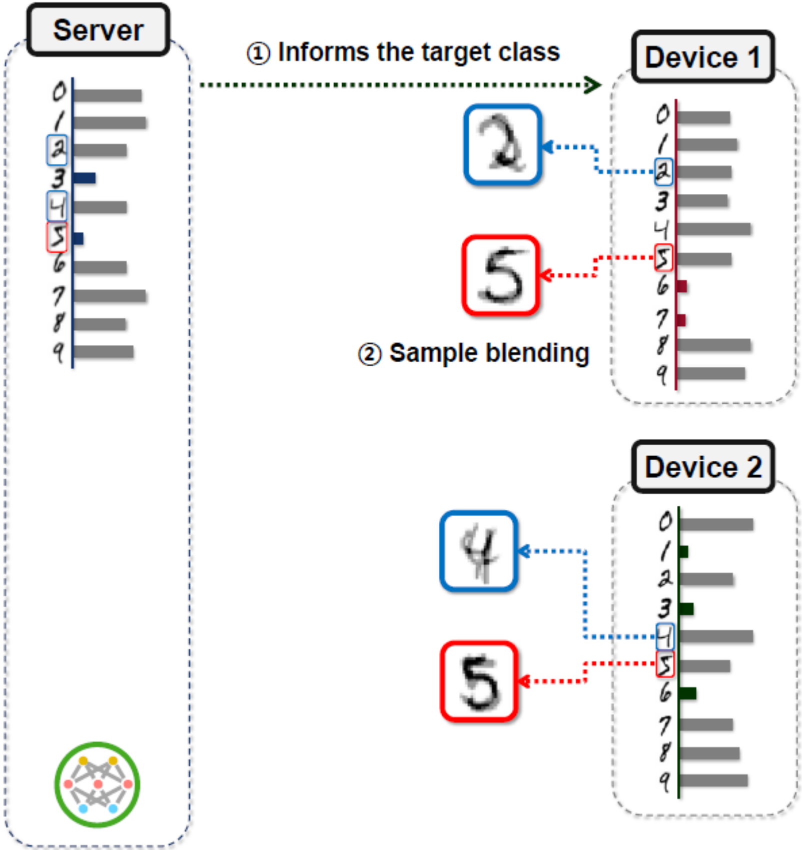
$\oplus$: XOR operation



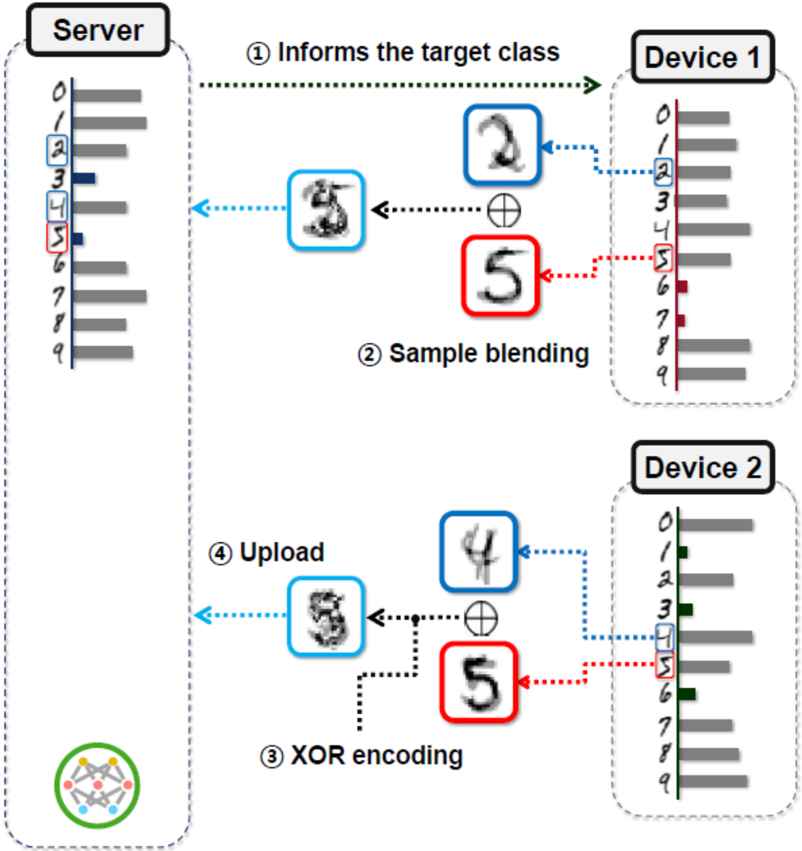
XOR Mixup


Server
① Informs the target class
Device 1
Device 2
0
1
2
3
4
5
6
7
8
9

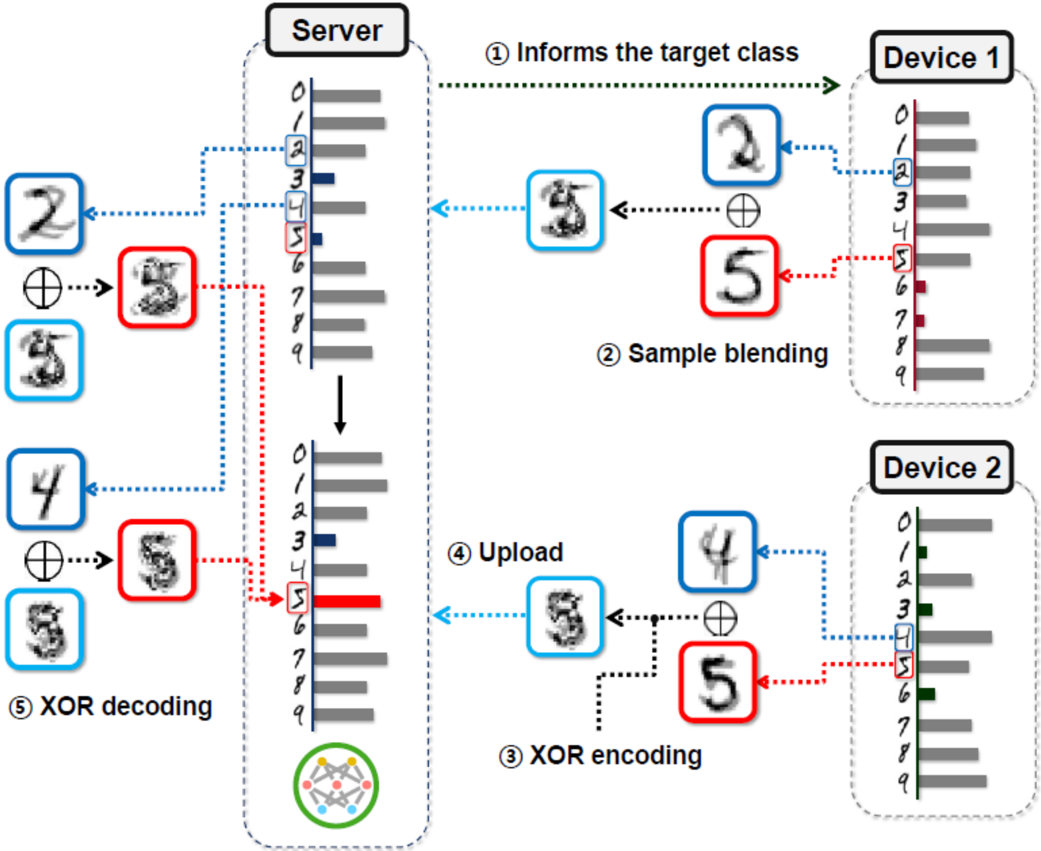
XOR Mixup


Server
① Informs the target class
Device 1
② Sample blending
Device 2
0
1
2
3
4
5
6
7
8
9

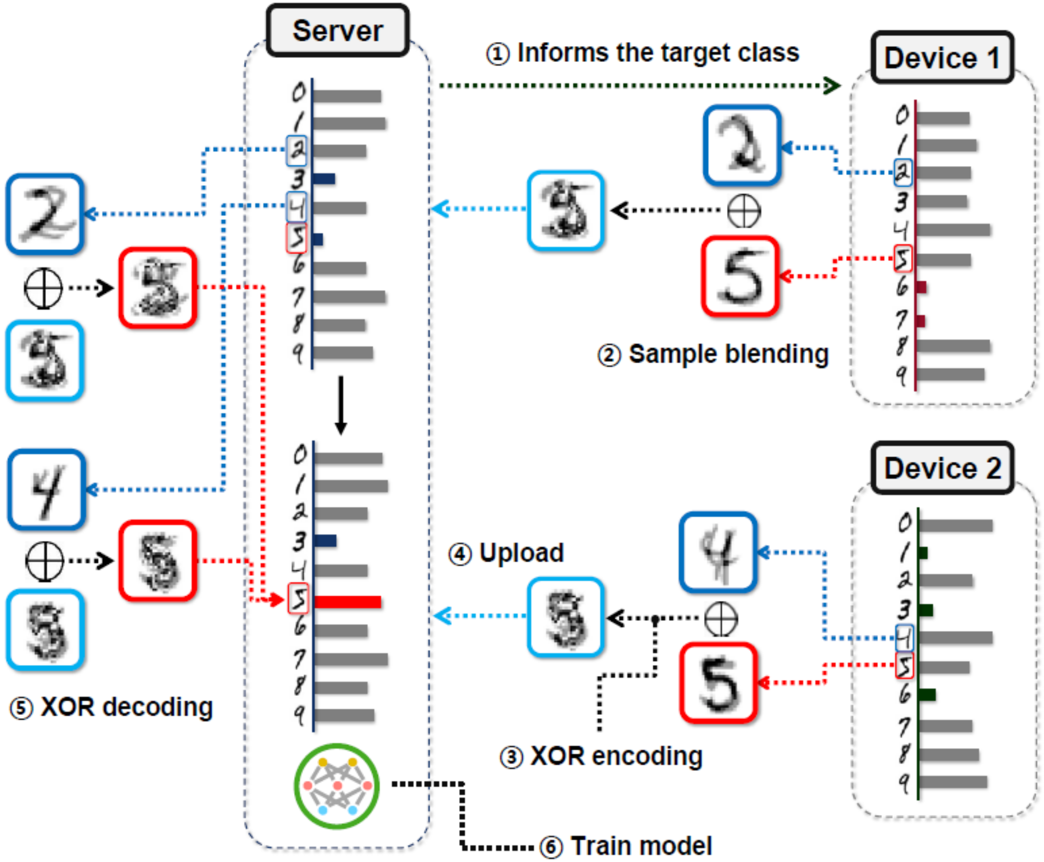
XOR Mixup


Server
① Informs the target class
Device 1
② Sample blending
Device 2
④ Upload
③ XOR encoding
0
1
2
3
4
5
6
7
8
9

XOR Mixup


Server
① Informs the target class
Device 1
② Sample blending
Device 2
④ Upload
③ XOR encoding
⑤ XOR decoding

XOR Mixup



Server
① Informs the target class
Device 1
② Sample blending
Device 2
④ Upload
③ XOR encoding
⑤ XOR decoding
⑥ Train model

- Reference
  - **XOR Mixup: Privacy-Preserving Data Augmentation for One-Shot Federated Learning**
    MyungJae Shin, Chihoon Hwang, Joongheon Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim
    ICML Workshop on Federated Learning for User Privacy and Data Confidentiality (Virtual, July 2020)
  - https://arxiv.org/abs/2006.05148

Thank you for your attention!

- More questions?
  - joongheon@korea.ac.kr