RingSFL: An Adaptive Split Federated Learning Towards Taming Client Heterogeneity

Webinar - IEEE ComSoC TCCN, SIG on AI empowered Internet of Vehicles

Nan Cheng

School of Telecommunications Engineering,
Xidian University

Sep. 1, 2023

[1] J. Shen, N. Cheng, X. Wang, F. Lyu, W. Xu, Z. Liu, K. Aldubaikhy, and X. Shen (2023). "RingSFL: An Adaptive Split Federated Learning Towards Taming Client Heterogeneity". *IEEE Transactions on Mobile Computing, Accepted*.

Roadmap

Part I

Background

Federated Learning

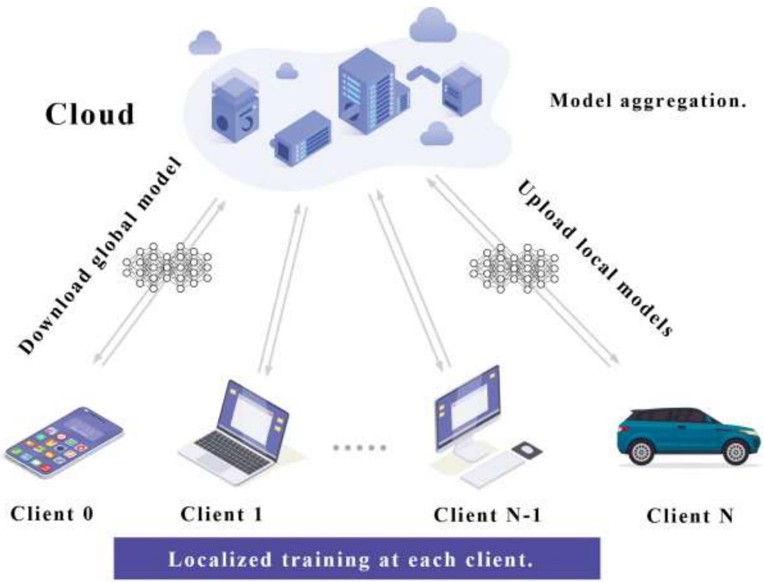


Figure. The training process of federated learning.

Training Process

- Cloud distribute initialized global model.
- Each client conducts training using their local datasets.
- Each client uploads trained local model to cloud for aggregation.
- Cloud distribute aggregated model.
- Repeat step 2 – 4 until converge.

Challenge

Client Heterogeneity

The clients in the FL system may differ significantly in terms of computational capability and battery level.

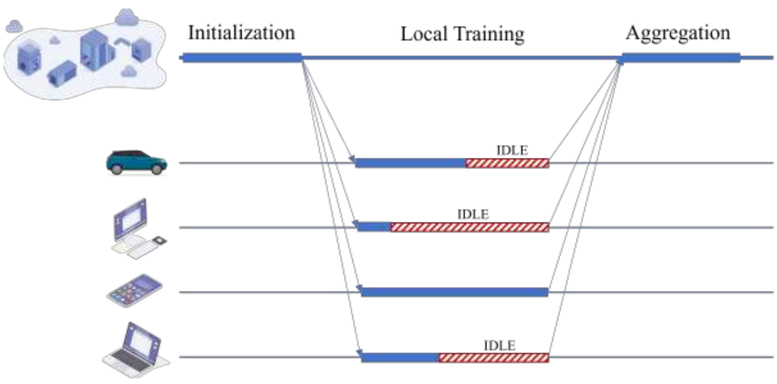


Figure. Straggler effect.

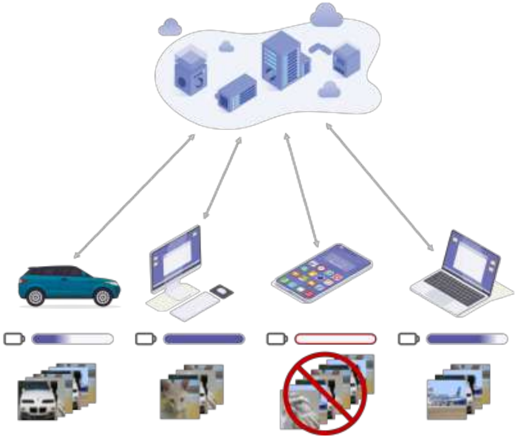


Figure. Client dropout.

Data heterogeneity leads to poor convergence and may cause clients with important data to drop out of training.

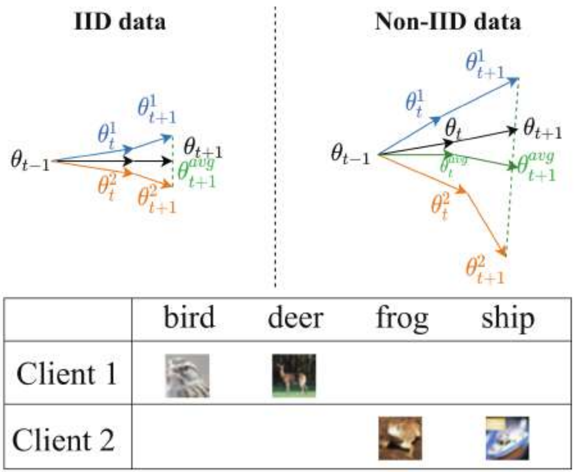


Figure. Non-IID data.

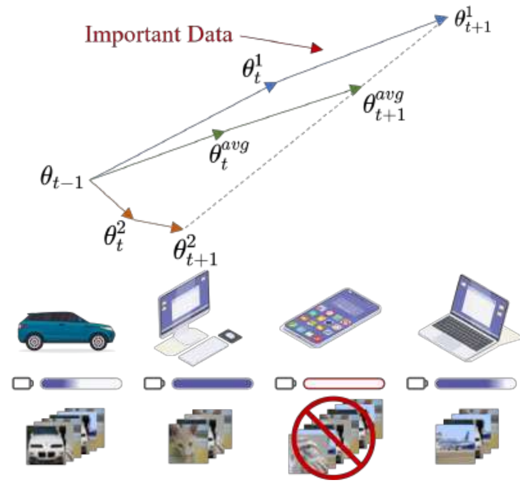


Figure. Important data absence.

Challenge
Privacy Leakage

Sensitive information can still be revealed from model parameters/gradients by a third-party entity or the server.

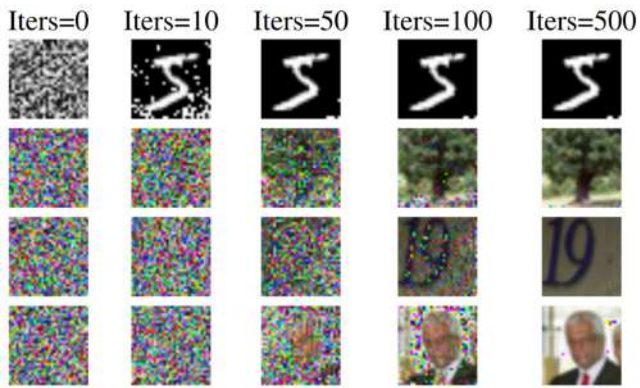


Figure. MIT at 2019.[a]

[a] L. Zhu, Z. Liu, and S. Han (2019). "Deep leakage from gradients". In: *Advances in neural information processing systems* 32.

Figure. Nvidia at 2021.[a]

[a] H. Yin, A. Mallya, A. Vahdat, J.M. Alvarez, J. Kautz, and P. Molchanov (2021). "See through gradients: Image batch recovery via gradinversion". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16337–16346.

SPLIT LEARNING

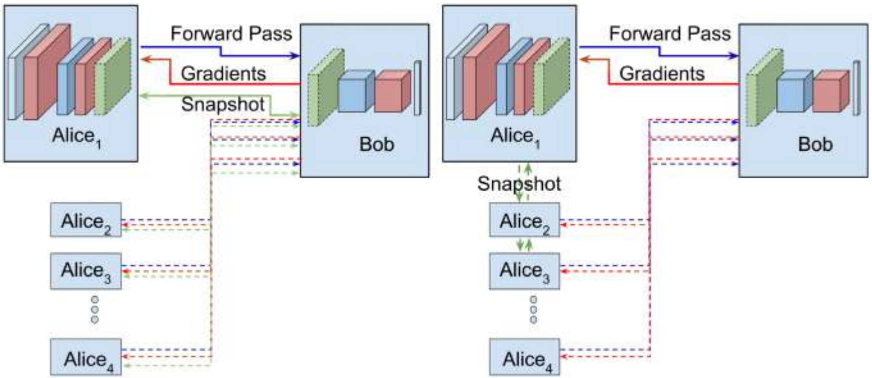


Figure. The training process of split learning.

Advantages

- Lower client computation load.
- Improved security.

Limitations

- Encounter convergence issues in Non-IID datasets.
- Cannot parallelize.

Part II

RingSFL: A Ring-shaped Split Federated Learning

Architecture

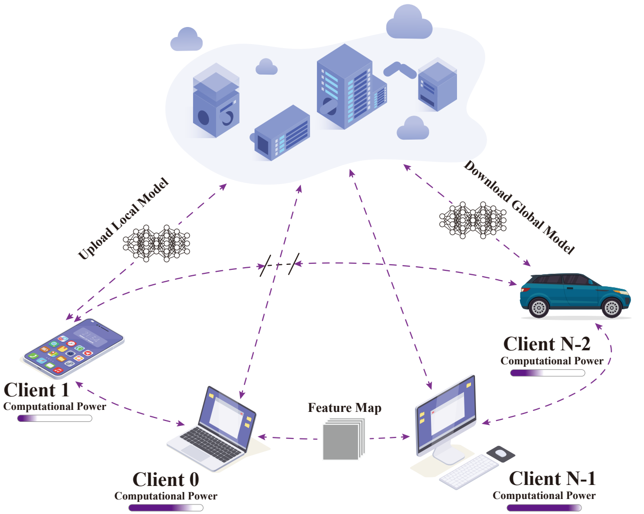
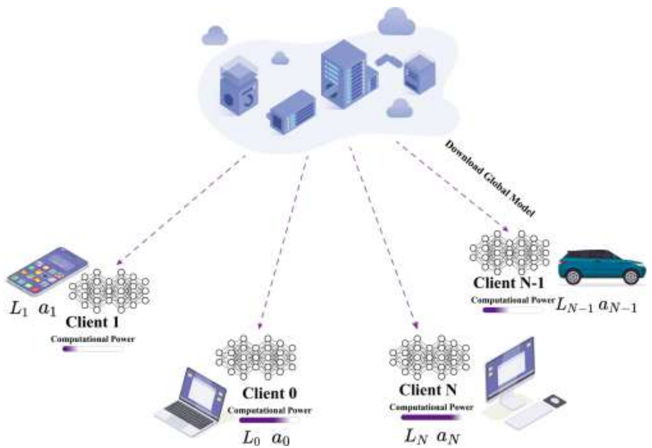


Figure. The architecture of RingSFL.

- The system consists of a server for model aggregation and N clients for cooperative training.
- The clients form a ring topology, where adjacent clients can communicate with each other through direct communication technologies such as device-to-device (D2D) communication.
- The clients can also communicate with the server for model downloading and uploading as in FL.

Training Process

Initialization



The server distributes the initialized global model with W layers and configuration parameters (L_i, a_i).

Propagation Length

$$L_i = \frac{C_i}{\sum_{j=0}^{N-1} C_j} W \tag{1}$$

C_i: computational power of u_i.

Aggregation Weight

$$a_i = \frac{D_i}{\sum_{j=0}^{N-1} D_j} \tag{2}$$

D_i: dataset size of u_i.

Training Process

Forward Propagation

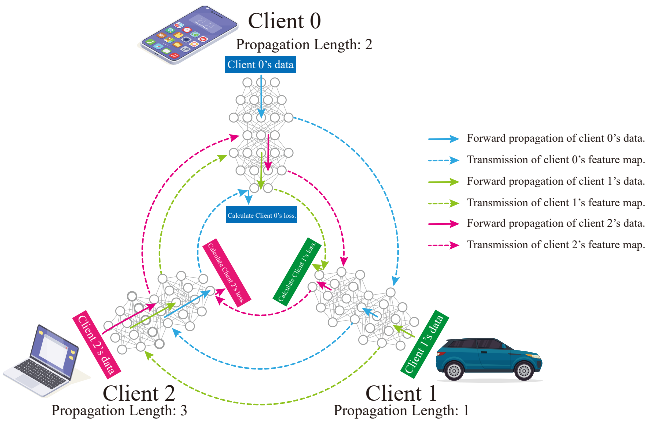


Figure. Forward propagation processes for RingSFL with 3 clients. A multilayer perceptron (MLP) containing 6 fully connected layers is trained, and the propagation length is set to: $L_0 : L_1 : L_2 = 2 : 1 : 3$.

- **Starting Phase**: Clients sample a batch from their respective datasets and enter it into the local model to get the feature map for the relay phase.
- **Relay Phase**: Clients receive the feature map from the previous node, propagate it forward in the local model and then send it to the next node.
- **Stop Phase**: When the feature map traverses all the clients, the clients receive their model output. Clients calculate loss values based on model output and local labels for back propagation.

Training Process
Backward Propagation

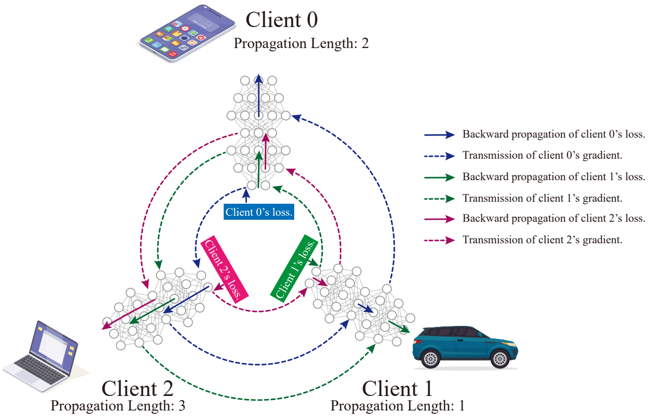
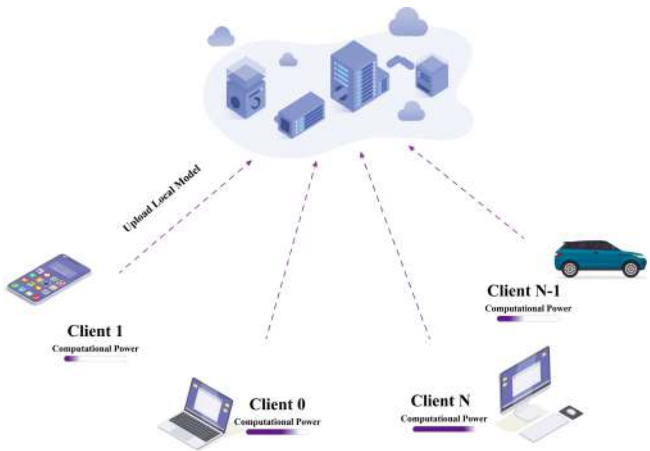


Figure. Backward propagation processes for RingSFL with 3 clients. A multilayer perceptron (MLP) containing 6 fully connected layers is trained, and the propagation length is set to: $L_0 : L_1 : L_2 = 2 : 1 : 3$.

- **Starting Phase**: Clients send the loss value to the previous node and start back propagation.
- **Relay Phase**: Clients receive the gradients from the next node in the ring, back propagate locally, and pass the gradients of the smashed layer to the previous node in the ring.
- **Stop Phase**: Clients use the locally cached model gradient to update the local model.

Training Process

Model Aggregation



- In each communication round, the trained local model parameters $\mathcal{W}_i^{t+1}$ are uploaded to the server for aggregation.
- Since the gradients are already weighted during the training process, model aggregation can be achieved by direct averaging

$$\mathcal{W}_g^{t+1} = \frac{1}{N}\sum_{i=0}^{N-1}\mathcal{W}_i^{t+1} \tag{3}$$

Model Split Scheme

The computation time of client u_i can be denoted by $\dfrac{p_i MN}{C_i}$, where p_i denotes the ratio of the training load assigned to u_i, $\sum_{i=0}^{N-1} p_i = 1$, and M denotes the computation volume of a model to update once.

$$\min_{p_0,\cdots,p_{N-1}} \ \max\left\{\frac{p_0 MN}{C_0}, \frac{p_1 MN}{C_1}, \cdots, \frac{p_{N-1} MN}{C_{N-1}}\right\} \tag{4}$$

$$\text{s.t.} \quad \sum_{i=0}^{N-1} p_i = 1, \tag{4a}$$

$$0 \le p_i \le 1, \quad \forall i = 0, \cdots, N-1. \tag{4b}$$

$$\Rightarrow \begin{cases} p_i^* = \dfrac{C_i}{\sum_{j=0}^{N-1} C_j}, & \forall i = 0, \cdots, N-1, \\ m^* = \dfrac{MN}{\sum_{j=0}^{N-1} C_j}. \end{cases} \tag{5}$$

So we set the propagation length of u_i to: $L_i = p_i^* W = \dfrac{C_i}{\sum_{j=0}^{N-1} C_j} W$

Overlapping Layers Can Improve Model Performance

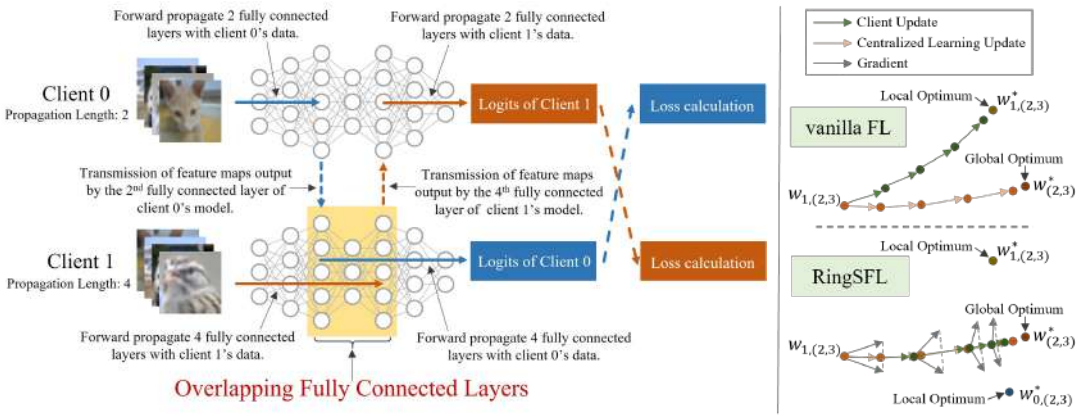


Figure. Forward propagation processes for RingSFL with 2 clients. A multilayer perceptron (MLP) containing 6 fully connected layers is trained,and the propagation length is set to: $L_0 : L_1 = 2 : 4$.

Higher aggregation frequency of overlapping layers, leading to more reliable gradient.

$$\mathcal{W}_{i,(j)}^{t} = \mathcal{W}_{i,(j)}^{t} - \eta|\mathcal{U}_{i,(j)}| \sum_{k \in \mathcal{U}_{i,(j)}} a_k g_{k,(j)}^{t}, \tag{6}$$

Privacy Enhancement

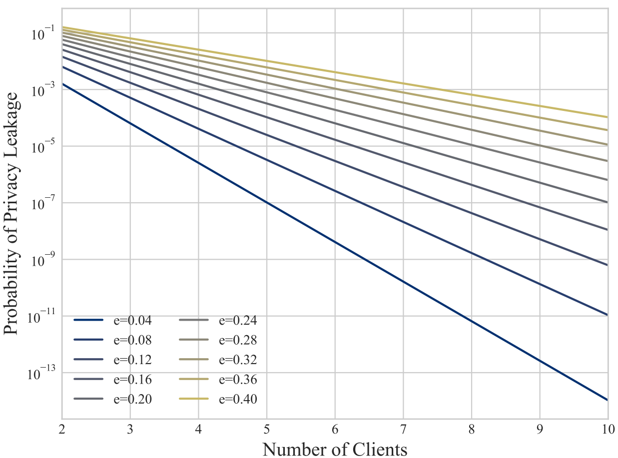
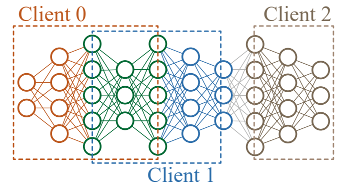


Figure. Impact of the number of clients and the probability of communication links being eavesdropped on the probability of privacy leakage.



- Since clients upload blended models to the server, an eavesdropper must reassemble these blended models based on propagation lengths to obtain the complete models belonging to each client.
- Using e_i to denote the probability that the communication link between u_i and the server is eavesdropped, the probability of privacy leakage can be expressed as

$$P = \prod_{i=0,\cdots,N-1} e_i. \tag{7}$$

Part III

Experimental Results

Setup

Simulation Environment

- Python 3.9.12
- Pytorch 1.11.0

Prototype System

- ARM Cortex-A72 @ 1.5GHz 6.4W
- 11th Gen Intel(R) Core(TM) i7-11700 @ 2.50GHz 65W
- Central Frequency: 5440MHz
- Bandwidth: 40MHz
- D2D rate: 135 ± 5.83 Mbps

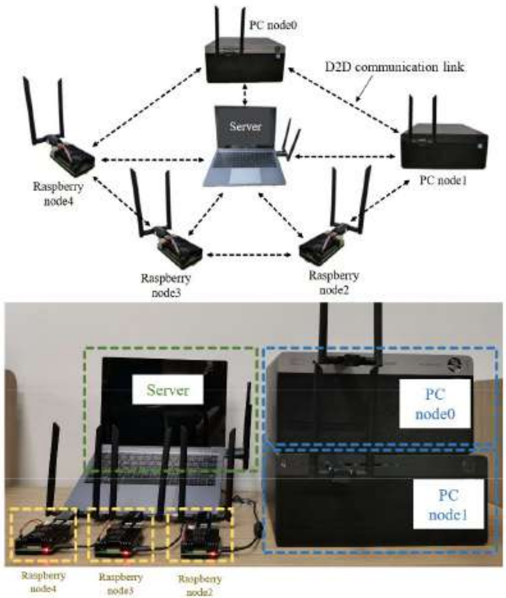


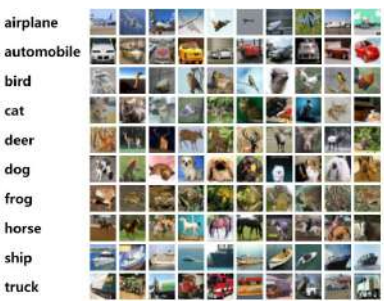
Figure. The prototype system of RingSFL.

Datasets and Models

Datasets

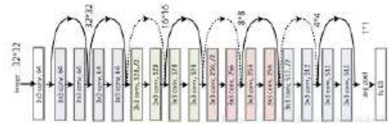
- MNIST

- CIFAR10

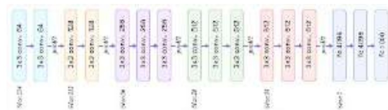


Models

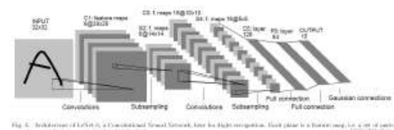
- ResNet18



- VGG16



- LeNet-5



- AlexNet

Experimental Results

Convergence Performance of ResNet18

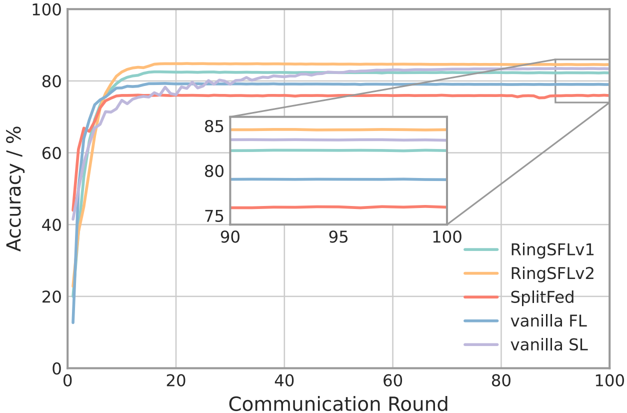


Figure. Trained on IID CIFAR10 dataset.

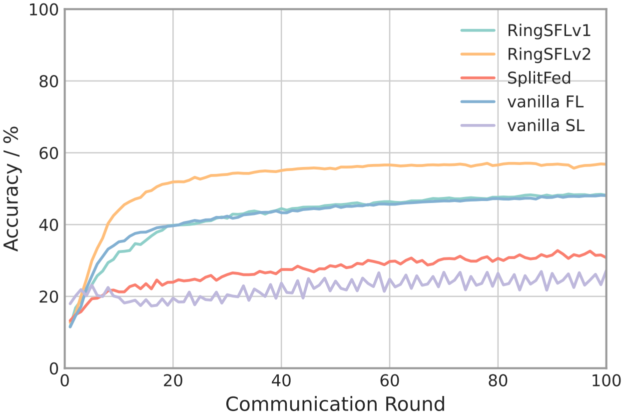


Figure. Trained on Non-IID CIFAR10 dataset.

Experimental Results

Convergence Performance of Other Models

Top-1 Accuracy (%) of Each Model under Different Algorithms. The best accuracy is marked in bold, and the secondary is marked in underline.

	ResNet18 (IID / Non-IID)	VGG16 (IID / Non-IID)	AlexNet (IID / Non-IID)	LeNet-5 (IID / Non-IID)
RingSFLv1	82.35 ± 0.36 / $\underline{48.30 \pm 0.57}$	$\underline{79.30 \pm 0.20}$ / $\underline{40.35 \pm 0.99}$	$\underline{98.83 \pm 0.11}$ / 89.58 ± 0.55	98.82 ± 0.19 / 94.34 ± 0.56
RingSFLv2	$\mathbf{84.57 \pm 0.17}$ / $\mathbf{56.80 \pm 0.78}$	$\mathbf{84.33 \pm 0.10}$ / $\mathbf{41.26 \pm 1.29}$	$\mathbf{99.13 \pm 0.07}$ / $\underline{94.31 \pm 0.88}$	$\mathbf{99.10 \pm 0.04}$ / $\underline{95.75 \pm 0.73}$
SplitFed	75.92 ± 0.51 / 30.16 ± 4.49	72.86 ± 0.62 / 28.17 ± 2.15	98.76 ± 0.09 / 84.00 ± 4.39	98.74 ± 0.24 / 93.64 ± 0.70
vanilla FL	78.93 ± 0.27 / 48.02 ± 1.28	77.02 ± 0.34 / 39.52 ± 0.81	98.81 ± 0.07 / 91.60 ± 1.14	$\underline{98.84 \pm 0.08}$ / 94.77 ± 0.29
vanilla SL	$\underline{83.41 \pm 0.44}$ / 26.96 ± 3.58	78.50 ± 0.69 / 35.33 ± 1.29	98.69 ± 0.10 / $\mathbf{98.84 \pm 0.08}$	98.80 ± 0.14 / $\mathbf{98.86 \pm 0.09}$

Experimental Results

Effect of Overlapping Layers

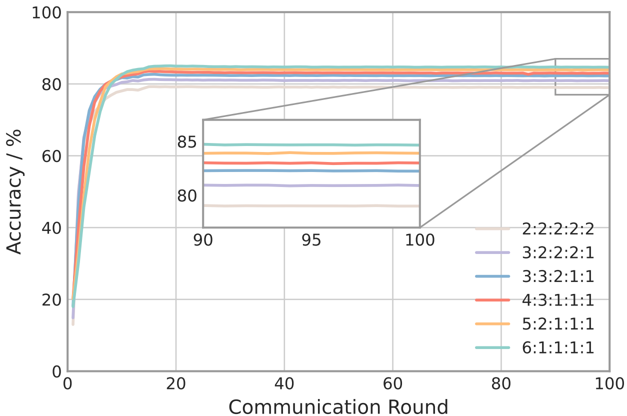


Figure. Trained on IID CIFAR10 dataset.

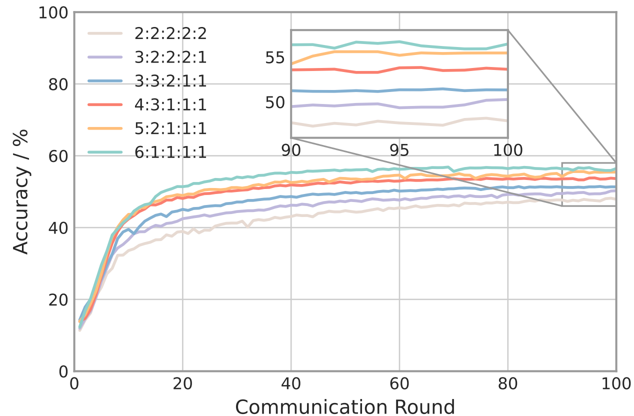


Figure. Trained on Non-IID CIFAR10 dataset.

Experimental Results

Effect of Overlapping Layers

Top-1 Accuracy (%) of Each Model under Different Propagation Lengths. The best accuracy is marked in bold, and the secondary is marked in underline.

Propagation Lengths	ResNet18 (IID / Non-IID)	Propagation Lengths	VGG16 (IID / Non-IID)	Propagation Lengths	AlexNet (IID / Non-IID)	Propagation Lengths	LeNet-5 (IID / Non-IID)
6:1:1:1:1	**84.66 ± 0.33** / **56.45 ± 1.10**	**12:1:1:1:1**	**84.29 ± 0.14** / 41.48 ± 1.08	**13:1:1:1:1**	99.00 ± 0.16 / **94.49 ± 0.67**	**8:1:1:1:1**	**99.10 ± 0.07** / **95.85 ± 0.32**
5:2:1:1:1	<u>83.90 ± 0.29</u> / <u>55.45 ± 0.47</u>	**11:2:1:1:1**	<u>83.98 ± 0.24</u> / **42.56 ± 0.69**	**11:3:1:1:1**	<u>99.05 ± 0.12</u> / <u>94.28 ± 0.59</u>	**7:2:1:1:1**	<u>99.04 ± 0.06</u> / <u>95.79 ± 0.30</u>
4:3:1:1:1	83.00 ± 0.16 / 53.63 ± 0.62	**10:3:1:1:1**	83.78 ± 0.53 / <u>41.68 ± 0.69</u>	**9:5:1:1:1**	**99.11 ± 0.10** / 93.79 ± 0.30	**6:3:1:1:1**	99.02 ± 0.05 / 95.66 ± 0.19
3:3:2:1:1	82.24 ± 0.20 / 51.34 ± 0.74	**8:3:3:1:1**	82.81 ± 0.25 / 39.25 ± 0.62	**7:5:3:1:1**	99.00 ± 0.14 / 93.00 ± 0.11	**5:3:2:1:1**	99.00 ± 0.06 / 95.65 ± 0.18
3:2:2:2:1	80.90 ± 0.19 / 50.27 ± 0.53	**6:3:3:3:1**	80.53 ± 0.32 / 37.57 ± 0.94	**5:5:3:3:1**	98.91 ± 0.07 / 92.14 ± 0.61	**4:3:2:2:1**	98.96 ± 0.04 / 95.51 ± 0.16
2:2:2:2:2	79.00 ± 0.50 / 47.89 ± 0.64	**4:3:3:3:3**	77.58 ± 0.25 / 39.76 ± 0.96	**4:4:3:3:3**	98.84 ± 0.12 / 92.53 ± 1.03	**3:3:2:2:2**	98.97 ± 0.03 / 95.45 ± 0.16

Experimental Results

Effect of D2D Communication

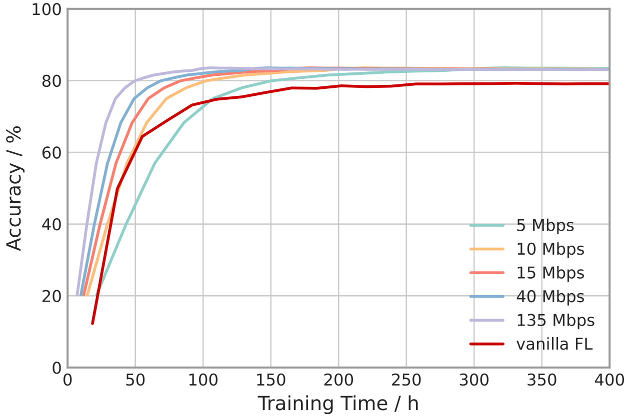


Figure. Testing convergence of ResNet18 on Cifar10 under different D2D communication rates.

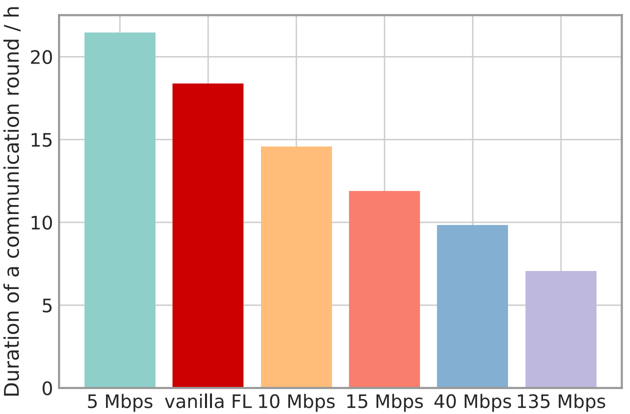


Figure. Time cost of ResNet18 in a communication round under different D2D communication rates.

Experimental Results

Convergence Time Reduction and Energy Efficiency

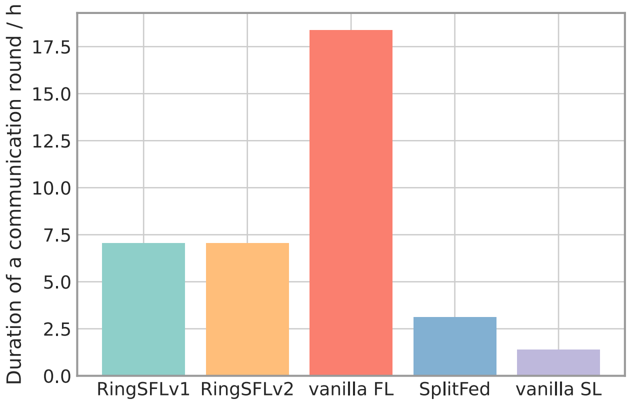


Figure. Time cost of ResNet18 in a communication round under different algorithms.

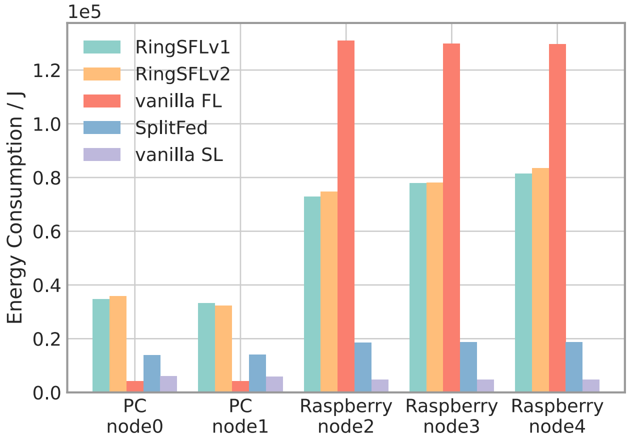


Figure. Energy consumption of different devices in a communication round.

Experimental Results

Effect of Client Dropout

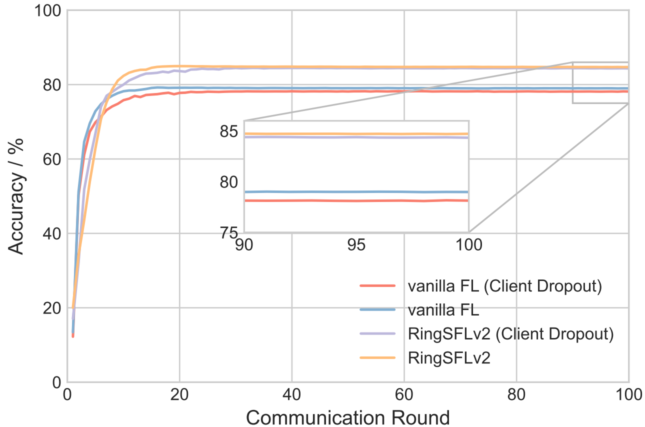


Figure. Testing convergence of ResNet18 on CIFAR10 (IID) with randomly two clients dropping out in each communication round.

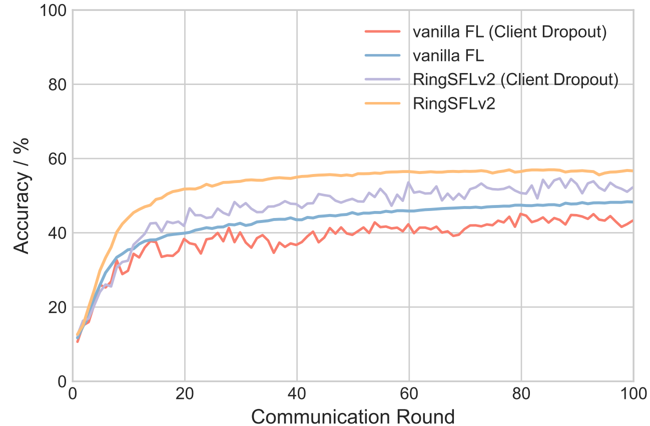


Figure. Testing convergence of ResNet18 on CIFAR10 (Non-IID) with randomly two clients dropping out in each communication round.

Experimental Results
Privacy Preservation

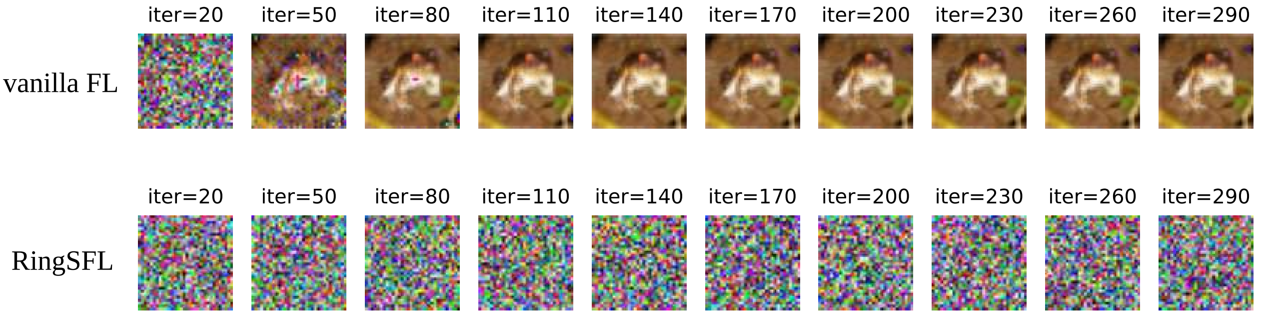


Figure. Reconstructed data after attacking vanilla FL and RingSFL.[2]

[2] L. Zhu, Z. Liu, and S. Han (2019). "Deep leakage from gradients". In: *Advances in neural information processing systems* 32.

References

- J. Shen, N. Cheng, X. Wang, F. Lyu, W. Xu, Z. Liu, K. Aldubaikhy, and X. Shen (2023). "RingSFL: An Adaptive Split Federated Learning Towards Taming Client Heterogeneity". In: *IEEE Transactions on Mobile Computing, Accepted*.
- H. Yin, A. Mallya, A. Vahdat, J.M. Alvarez, J. Kautz, and P. Molchanov (2021). "See through gradients: Image batch recovery via gradinversion". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16337–16346.
- L. Zhu, Z. Liu, and S. Han (2019). "Deep leakage from gradients". In: *Advances in neural information processing systems* 32.

THANKS