西安電子科技大學
XIDIAN UNIVERSITY

School of Cyber Engineering

Compressed Federated Learning Based on Adaptive Local Differential Privacy

Yinbin Miao[1], Rongpeng Xie[1], Xinghua Li[1], Ximeng Liu[2], Zhuo Ma[1], and Robert H.Deng[3]

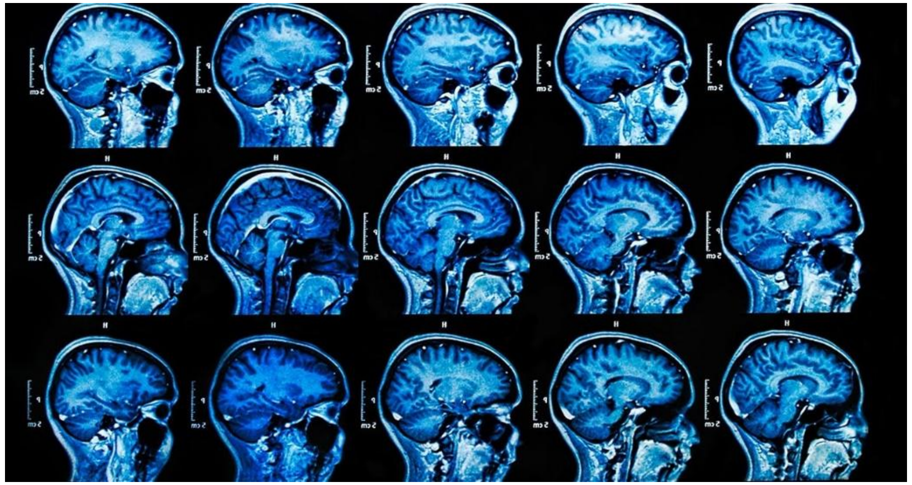
[1]Xidian University, China; [2]Fuzhou University, China; [3]Singapore Management University, Singapore

Annual Computer Security Applications Conference

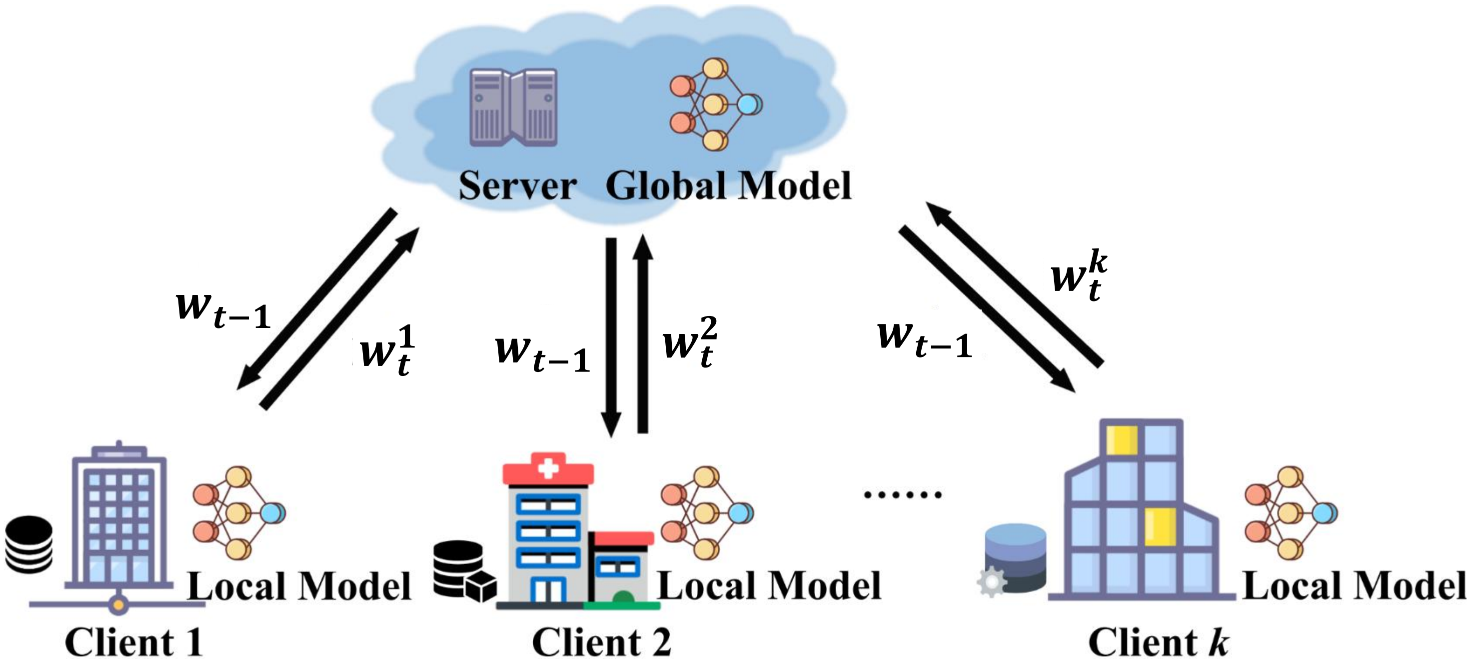
ACSAC 2022
December 5–9, 2022 • Austin, Texas, USA


BANK
HOSPITAL
Data Silo

Federated Learning

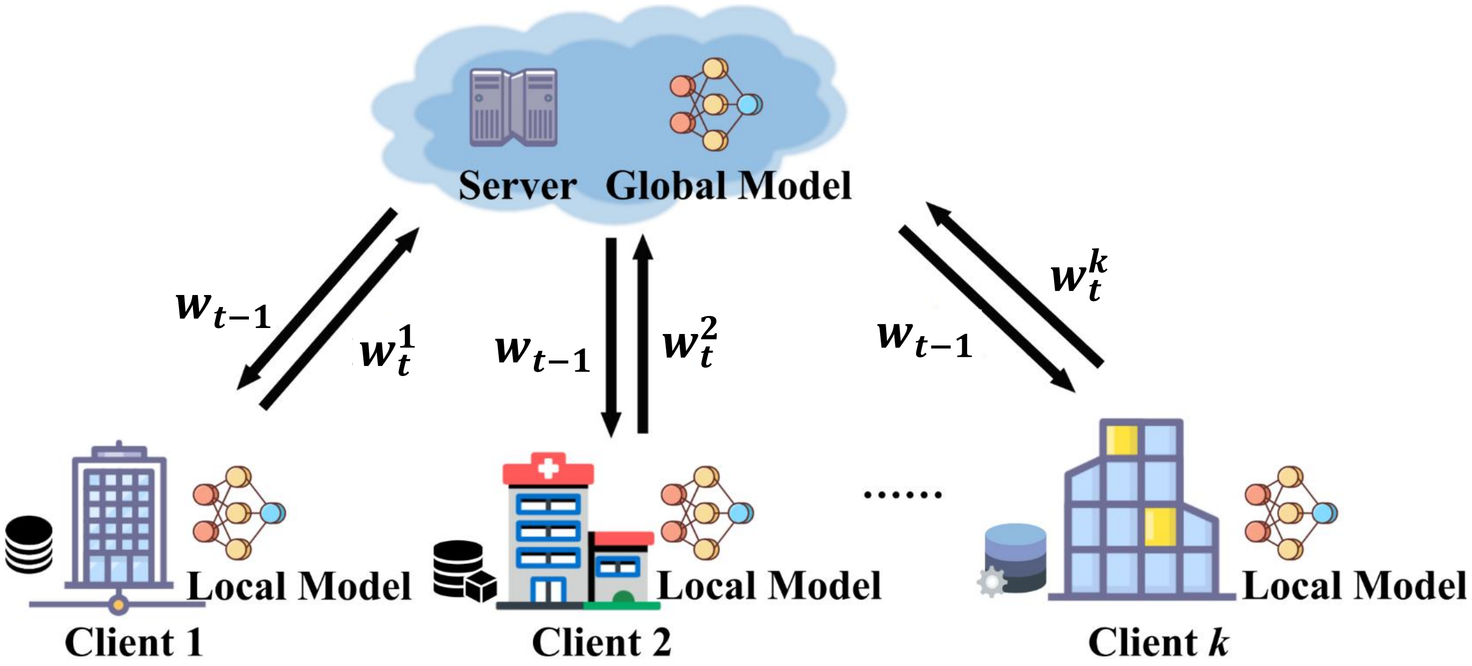


FEDERATED LEARNING

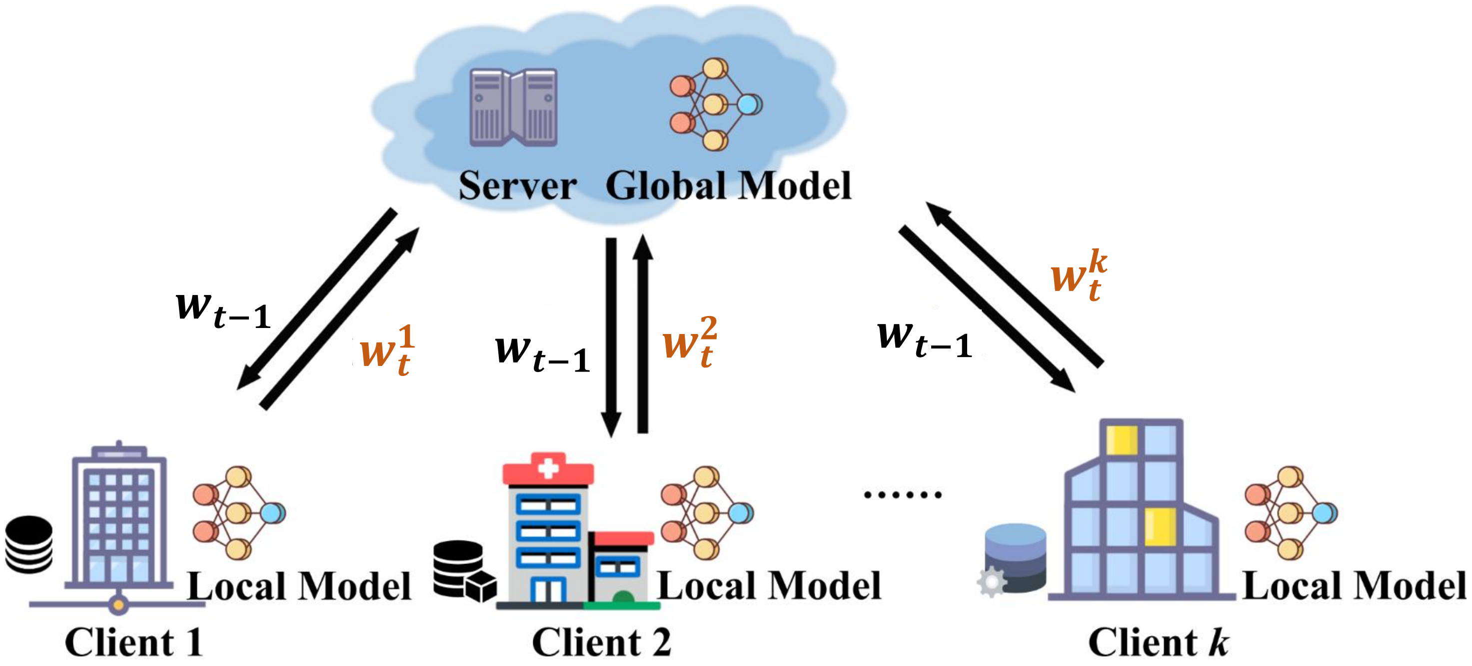


- Step-I : Server distributes global model
- Step-II : Clients train local models
- Step-III : Clients upload local models and server aggregates the results

ISSUES

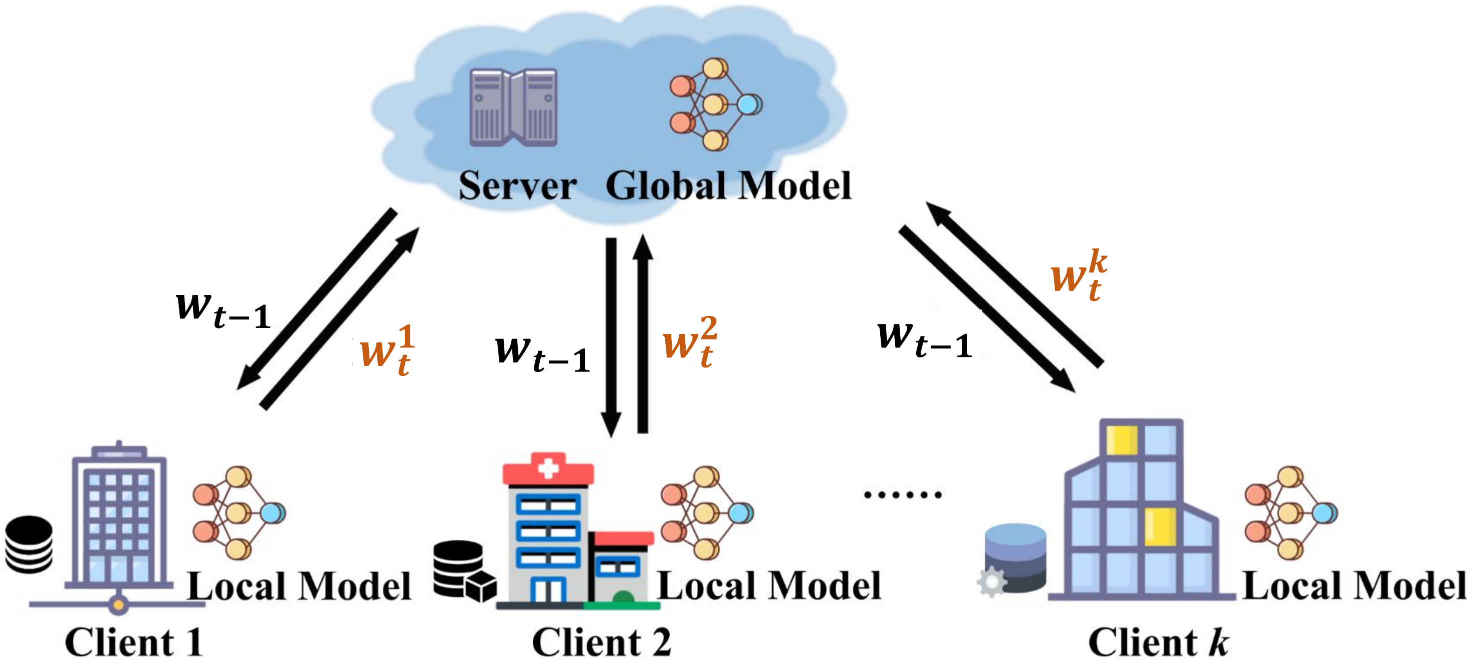

Server
Global Model
w_{t-1}
w_t^1
w_{t-1}
w_t^2
w_{t-1}
w_t^k
Local Model
Client 1
Local Model
Client 2
......
Local Model
Client k

ISSUES

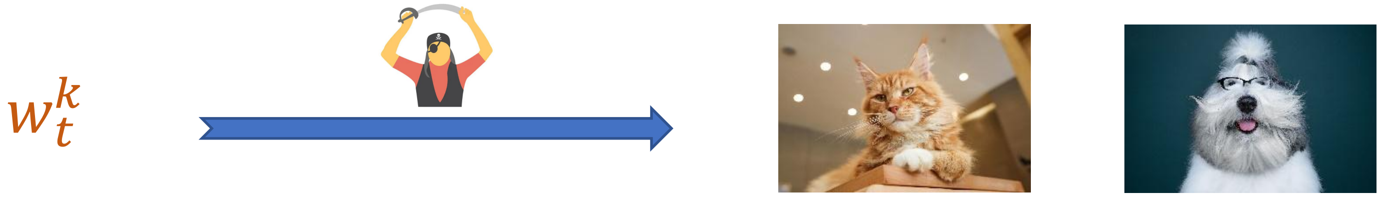


- Privacy disclosure of clients' training data due to plaintext model parameters or gradients

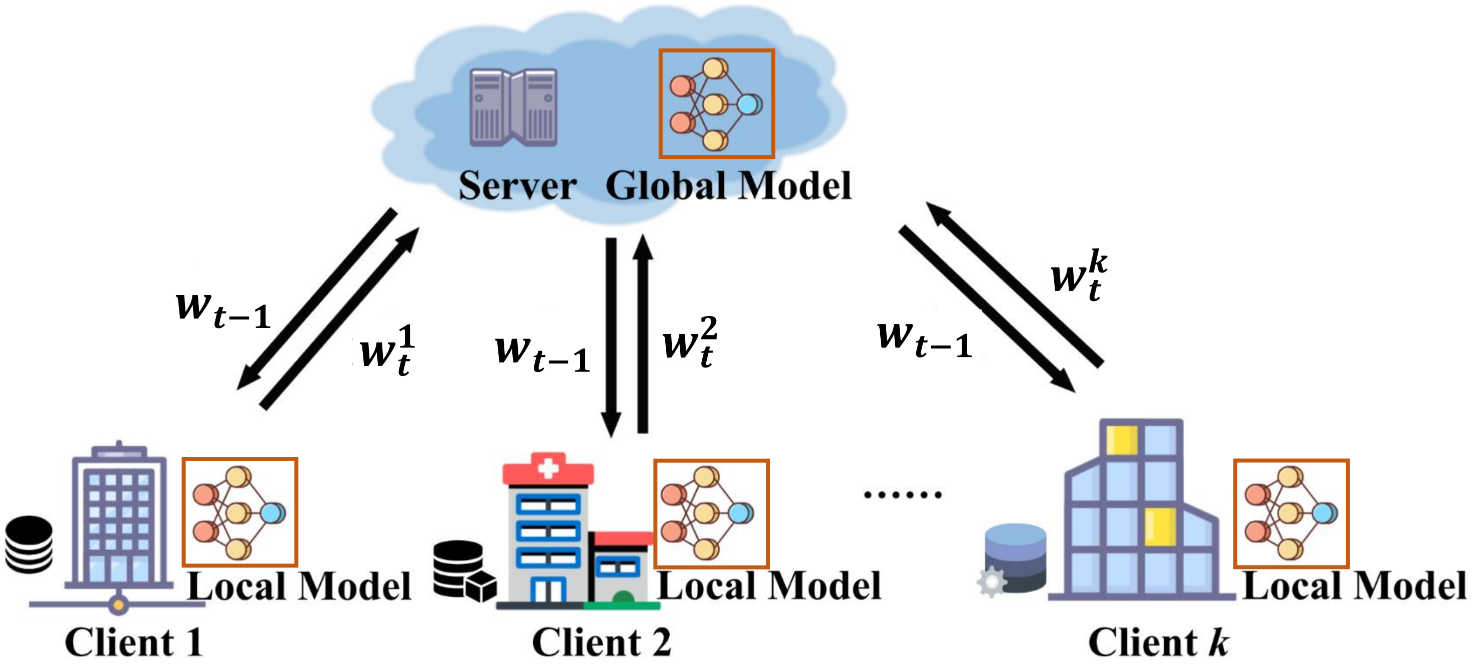
ISSUES



- Privacy disclosure of clients' training data due to plaintext model parameters



ISSUES



■ The Curse of Dimensionality of DNN

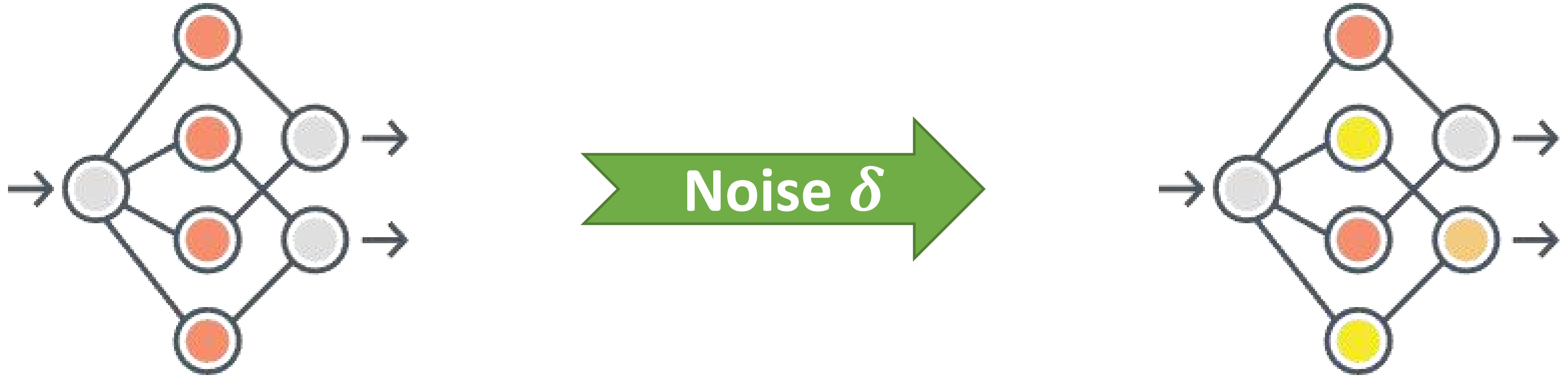
METHODOLOGY & CHALLENGES

■ Local Differential Privacy (LDP)

Let $\mathcal{M}$ be a randomized perturbation mechanism, for any pair input x and z in $\mathcal{D}$ and any output Y of $\mathcal{M}$, $\mathcal{M}$ satisfies ε-LDP such that

$$Pr[\mathcal{M}(x) = Y] \leq e^{\varepsilon} \cdot Pr[\mathcal{M}(z) = Y],$$

where $\mathcal{D}$ is a dataset and ε is the privacy budget of $\mathcal{M}$.

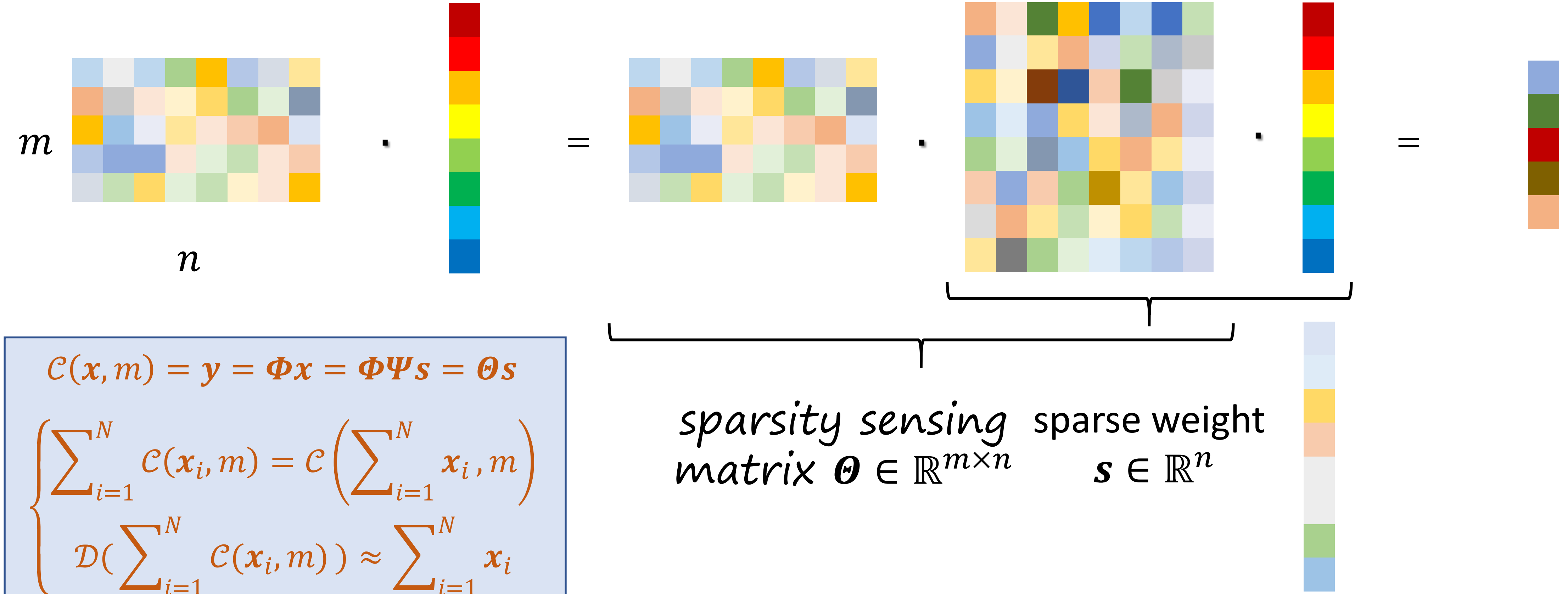


$w^* = w + \delta$, where $\boxed{\delta \sim \mathcal{N}(0,\ \sigma^2)}$

random *distribution*

METHODOLOGY & CHALLENGES

Compressive Sensing (CS)

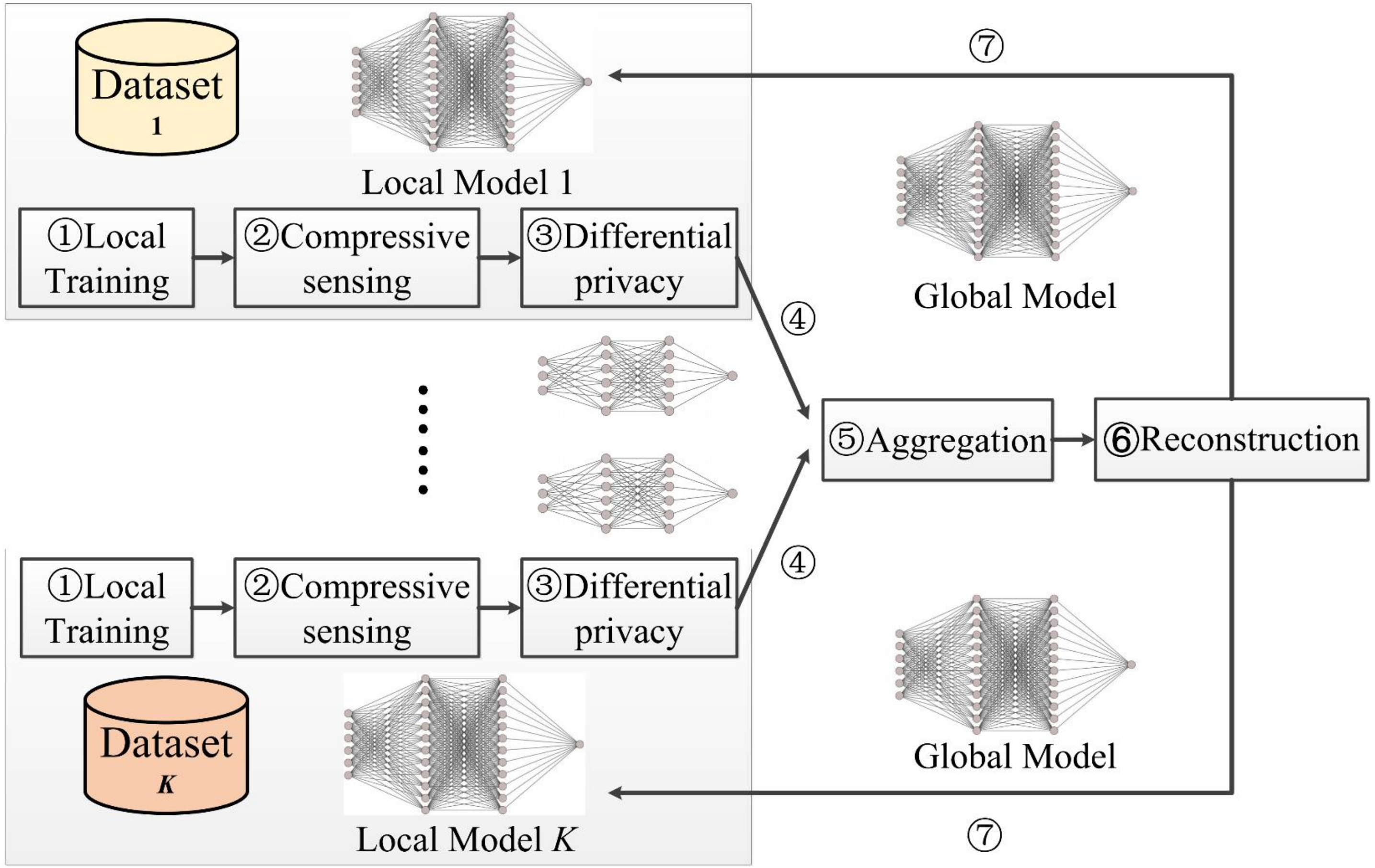


$$\mathcal{C}(\boldsymbol{x}, m) = \boldsymbol{y} = \boldsymbol{\Phi x} = \boldsymbol{\Phi \Psi s} = \boldsymbol{\Theta s}$$

$$\begin{cases} \sum_{i=1}^{N} \mathcal{C}(\boldsymbol{x}_i, m) = \mathcal{C}\left(\sum_{i=1}^{N} \boldsymbol{x}_i, m\right) \\ \mathcal{D}\left(\sum_{i=1}^{N} \mathcal{C}(\boldsymbol{x}_i, m)\right) \approx \sum_{i=1}^{N} \boldsymbol{x}_i \end{cases}$$

OUR SCHEME

■ **Technical Overview :**



OUR SCHEME

- **Local Compressive Sensing:**

Algorithm 2 : Local Compressive Sensing

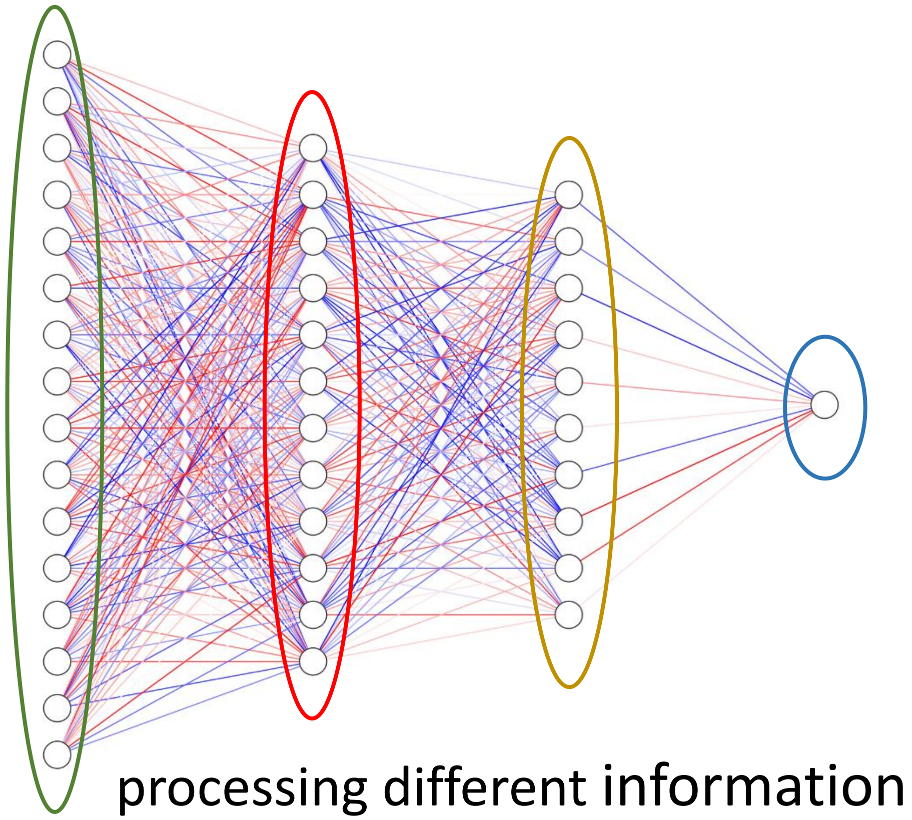
Input: the trained local model $\mathbf{w}_t^k$; the total layers L; a certain layer l; compression ratio CR

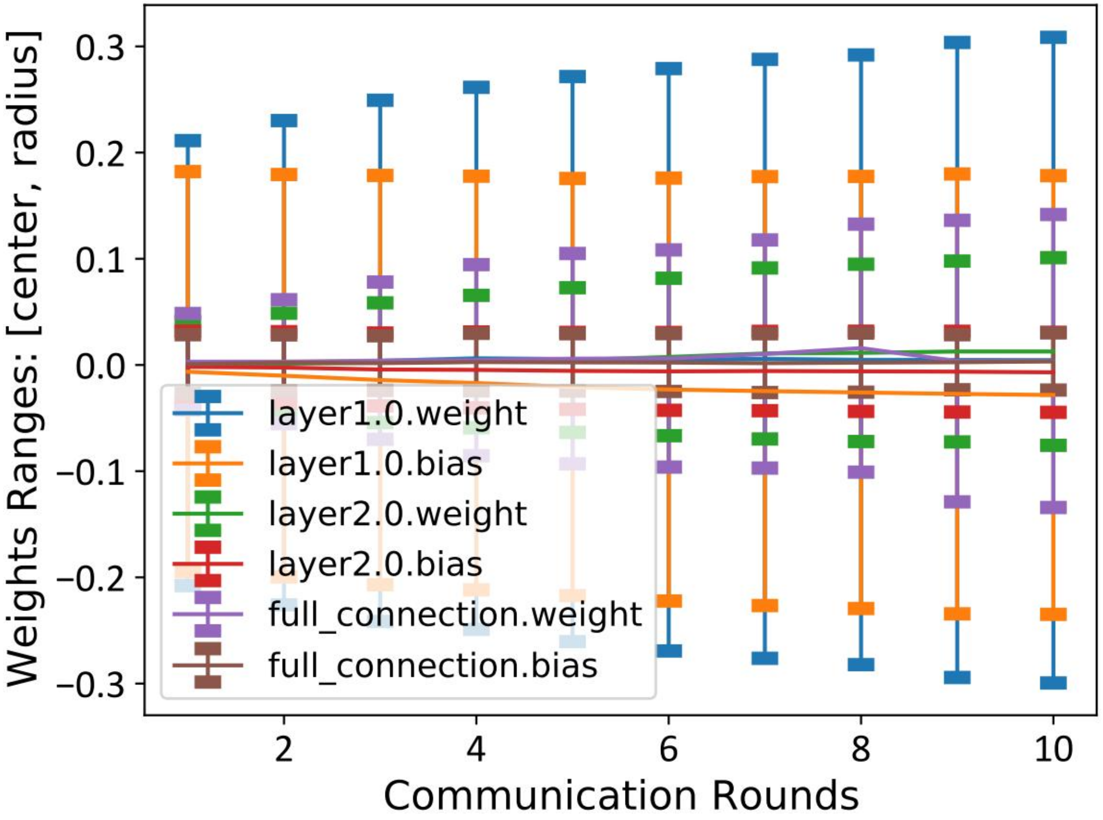
Client$_k(\mathbf{w}_t^k)$:

$\mathbf{c}_t^k = \mathbf{w}_t^k$
for each $l \in L$ in $\mathbf{c}_t^k$ **do**
    $n_l = \text{len}(\mathbf{c}_{t,l}^k)$ // the number of weights for this layer
    $m_l = n_l \times CR$
    $\mathbf{c}_{t,l}^k = C(\mathbf{c}_{t,l}^k, m_l)$ // including DCT and interception
end for
Return: $\mathbf{c}_t^k$

OUR SCHEME

■ Adaptive Local Differential Privacy:





- The weight ranges $[c_l - r_l,\ c_l + r_l]$ of different layers $l \in [1,\ L]$ are different, where c_l is the range center and r_l is the range radius.

OUR SCHEME

■ **Adaptive Local Differential Privacy:**

$$w^* = w + \delta = \boxed{\mathcal{M}(c_l, r_l)}$$

Layer-level adaptability

$$c_l = \frac{(max_l + min_l)}{2}, \quad r_l = max_l - c_l$$

FIXED

Finer-grained

Weight-level adaptability

Even if they are weights of the same layer, their offsets will be different

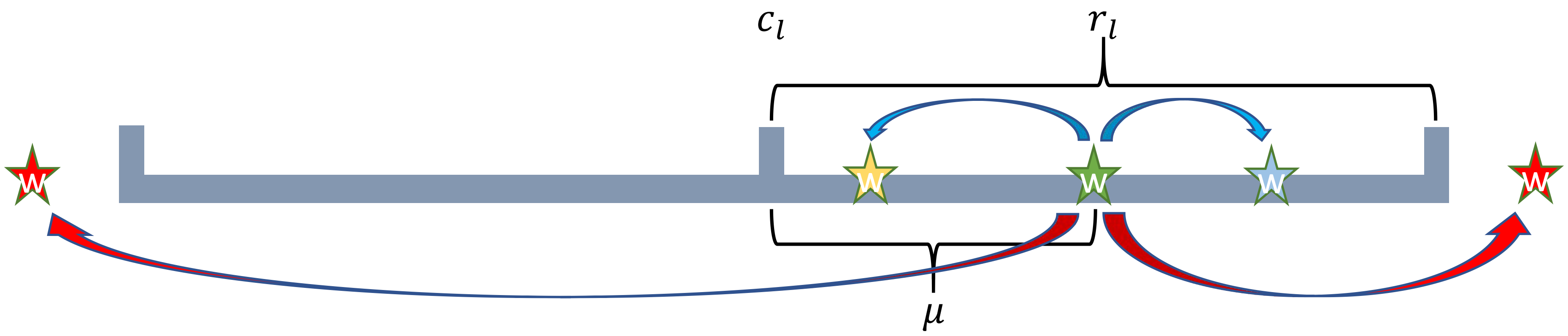
offset: $\mu = w - c_l \leq r_l$

$$w^* = \mathcal{M}(c_l, \mu)$$

OUR SCHEME

■ **Adaptive Local Differential Privacy:**

$$\mathcal{M}(w) = w^* = \begin{cases} c_l + \mu \cdot \dfrac{e^{\varepsilon}+1}{e^{\varepsilon}-1}, & with\ probability \quad \dfrac{e^{\varepsilon}-1}{2e^{\varepsilon}} \\ c_l + \mu \cdot \dfrac{e^{\varepsilon}-1}{e^{\varepsilon}+1}, & with\ probability \quad \dfrac{e^{\varepsilon}+1}{2e^{\varepsilon}} \end{cases}$$



EXPERIMENTS

■ SETUP:

- Datasets: MNIST, Fashion-MNIST
- Model: CNN(2 convolutional layers + 1 fully connected layer)
- Super-parameters:

Epochs E	50/100/200
Number of clients	10
Learning rate	0.1
Compression Ratio (CR)	1/0.5/0.1/0.05
Privacy budget	$+\infty$/1/2

- Runtime environment: Pytorch 1.10.0, Numpy 1.21.5, a single CPU @ 3.30 GHz, 16.0GB RAM

EXPERIMENTS

- **Analysis of Adaptive Perturbation:**

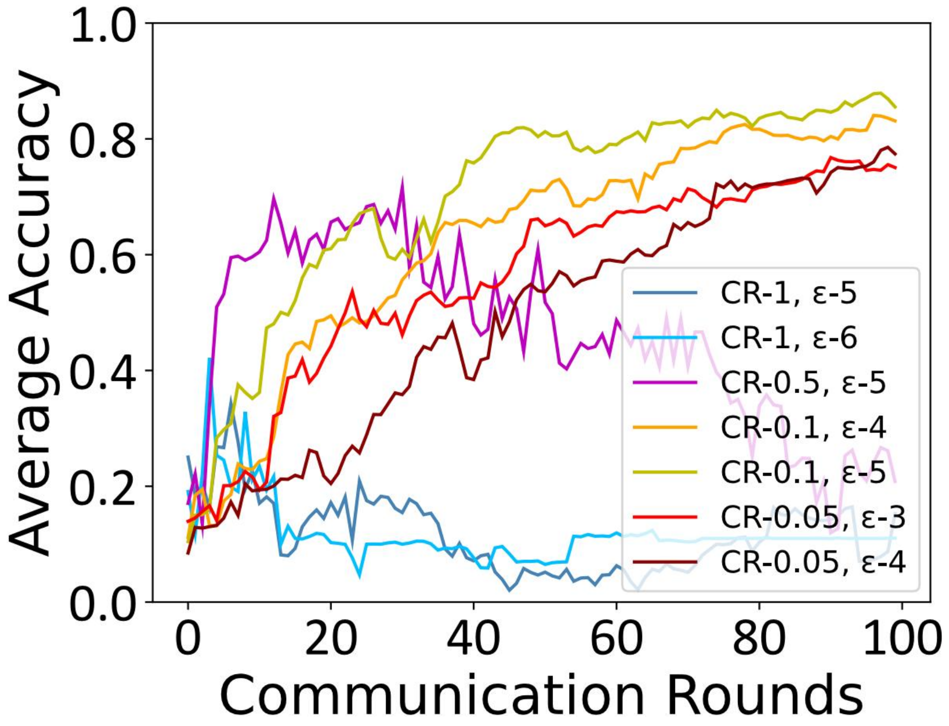


Fig. 1. Apply r_l in MNIST

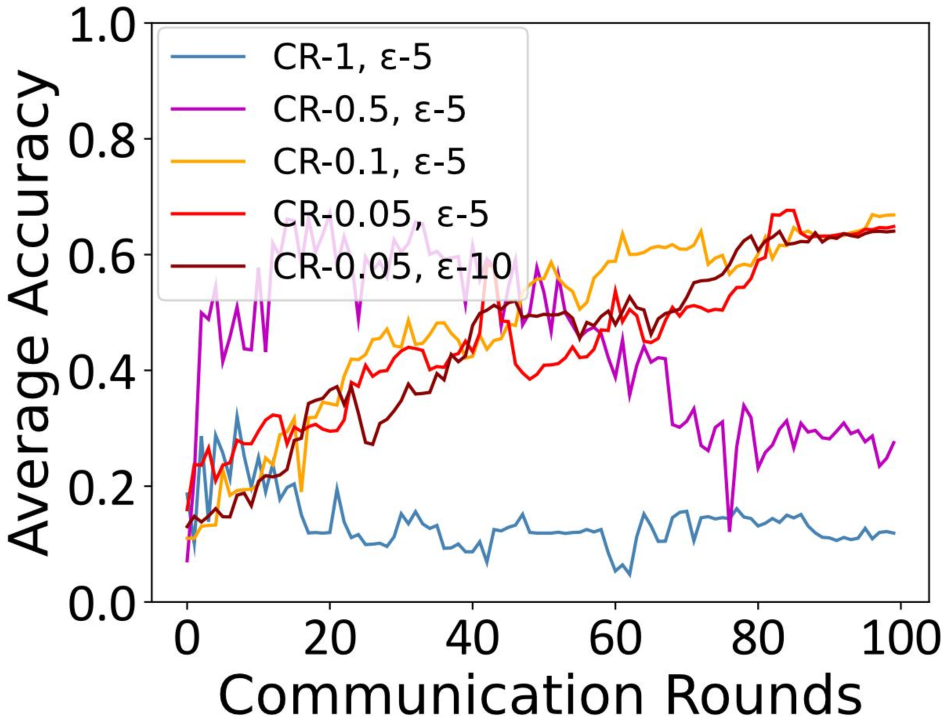
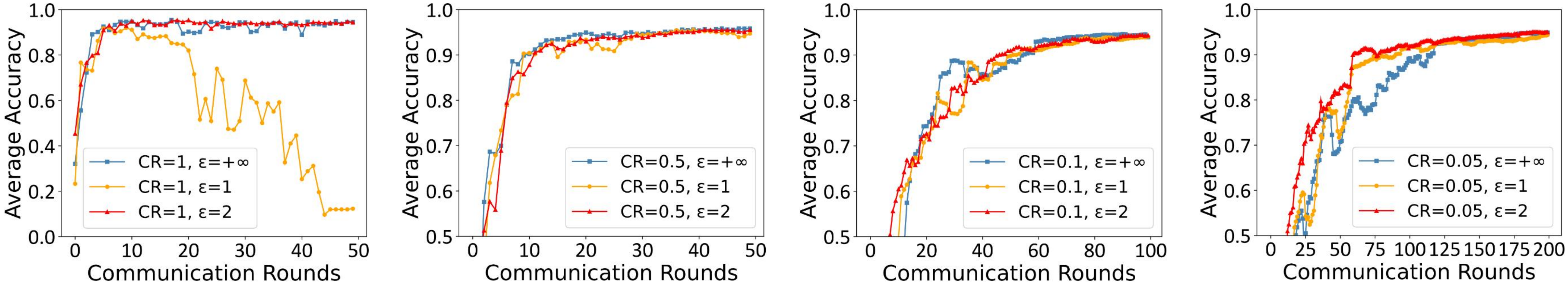


Fig. 2. Apply r_l in Fashion-MNIST

EXPERIMENTS

Analysis of Adaptive Perturbation:



(a) (b) (c) (d)

Fig. 3. Apply μ in MNIST

EXPERIMENTS

■ Analysis of Adaptive Perturbation:

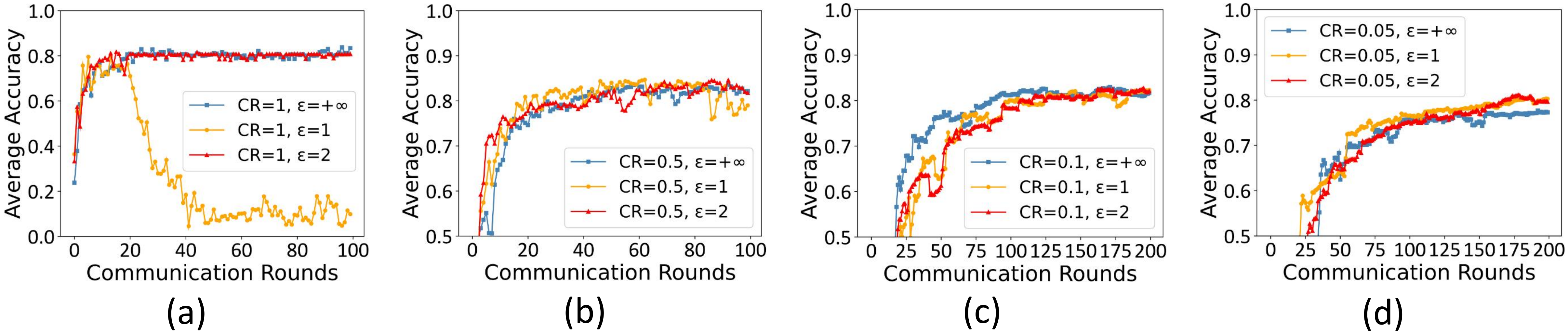


Fig. 4. Apply μ in Fashion-MNIST

■ Analysis of Compression Ratio:

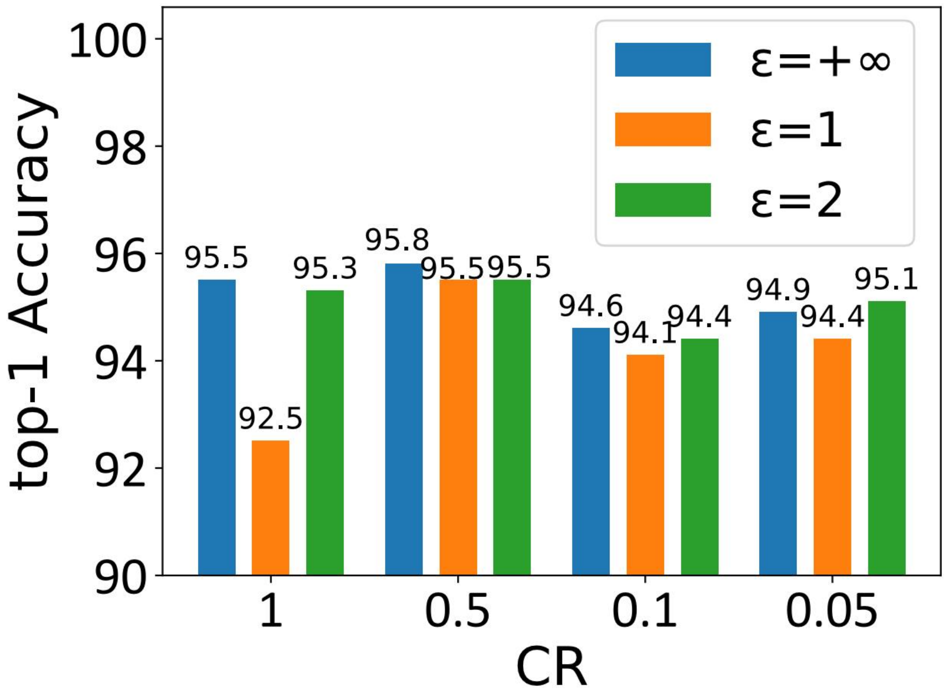


Fig. 5. MNIST

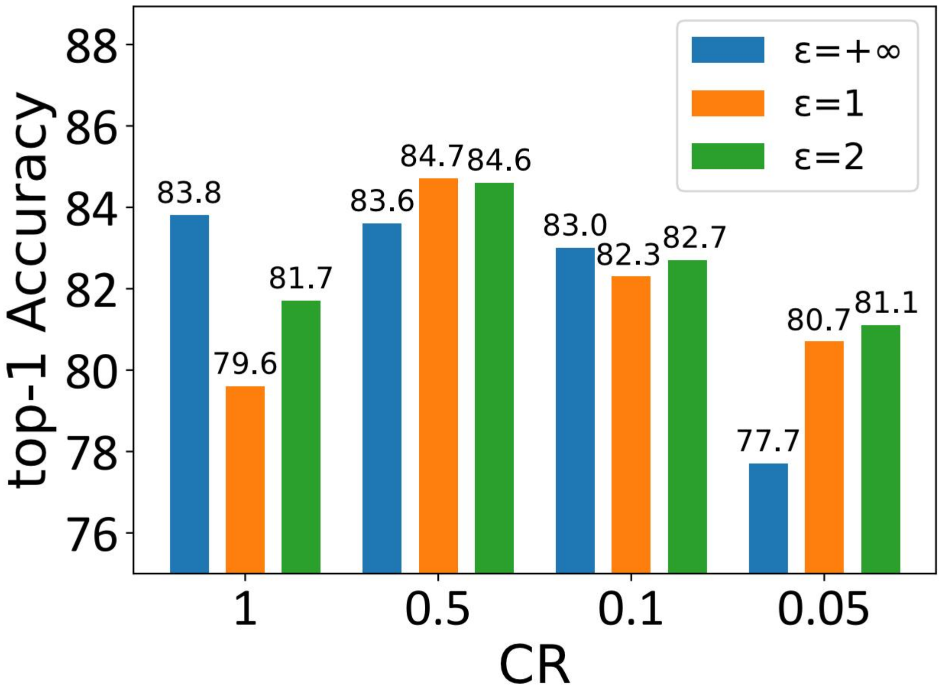


Fig. 6. Fashion-MNIST

EXPERIMENTS

■ Traffic and Running Time:

Privacy budget ε		$\varepsilon = +\infty$		$\varepsilon = 1$	
Metrics		Traffic(MB)	Runtime(mins)	Traffic(MB)	Runtime(mins)
Dataset		MNIST			
CR	1	12.69	13.76	12.69($\varepsilon = 2$)	14.45($\varepsilon = 2$)
	0.5	6.35	16.79	6.35	24.80
	0.1	1.27	52.26	1.27	53.66
	0.05	0.63	72.84	0.63	105.27
Dataset		Fashion-MNIST			
CR	1	12.69	27.93	12.69($\varepsilon = 2$)	28.37($\varepsilon = 2$)
	0.5	6.35	33.55	6.35	33.52
	0.1	1.27	70.35	1.27	90.16
	0.05	0.63	72.51	0.63	130.12

CONCLUSION

- We use the compressive sensing to compress the local model, which reduces not only the size of the model but also the amount of noises
- We apply adaptive Local Differential Privacy to add controllable noises for protecting data privacy and ensuring high model performance
- Our experiments demonstrate that our scheme improves the accuracy of the model with lower privacy budget, and reduces the communication overhead by 95% at most

Thank you!
Q&A