Efficient and Straggler-Resistant Homomorphic Encryption for Heterogeneous Federated Learning

Nan Yan[1], Yuqing Li[1], Jing Chen[1], Xiong Wang[2], Jianan Hong[3], Kun He[1], and Wei Wang[4]

[1]Wuhan University

[2]Huazhong University of Science and Technology

[3]Shanghai Jiao Tong University

[4]The Hong Kong University of Science and Technology

Federated Learning (FL)

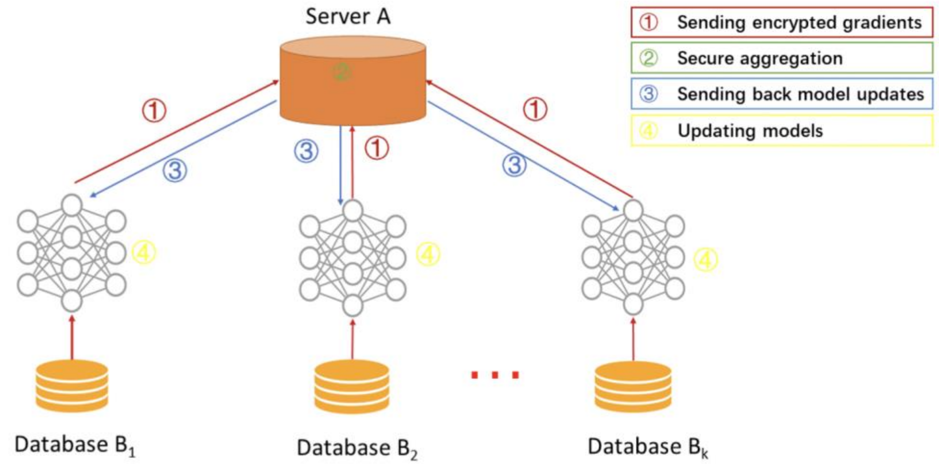
Privacy Concerns:

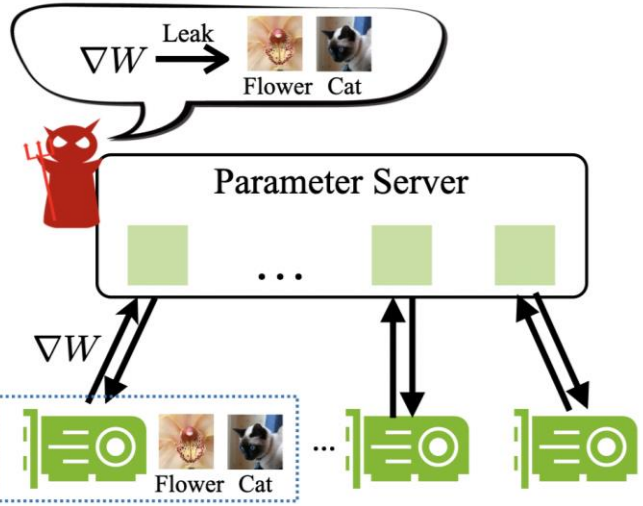
- GDPR
- CCPA

Solution: Federated Learning [1]

Collaborative training without sharing private data

Leakage of parameters or gradients (eg.DLG [2])





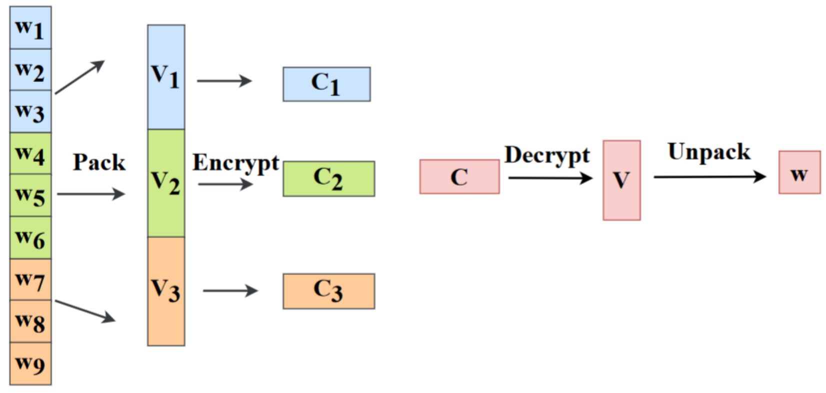
[1] Yang Q, Liu Y, Chen T, et al. Federated machine learning: Concept and applications[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2019, 10(2): 1-19.
[2] Zhu L, Liu Z, Han S. Deep leakage from gradients[J]. Advances in neural information processing systems, 2019, 32

(Packed) Homomorphic Encryption (PHE)

Homomorphic Encryption (HE):

- Encrypt model updates [3]
- Operate directly on ciphertext

PHE: Packing multiple plaintexts into a single ciphertext [4]



- Why is HE expensive:
  - Computation
  - Communication

Plaintext size	(Packed) HE scheme	Ciphertext size	Encryption time (s)	Decryption time (s)
109.89KB	Paillier	21.97 MB	63.46	39.63
	PackedPaillier	264.96 KB	3.18	2.60
	BFV		Memory out	
	PackedBFV	22.68 MB	0.04	0.02
	CKKS		Memory out	
	PackedCKKS	4.54 MB	0.06	0.04

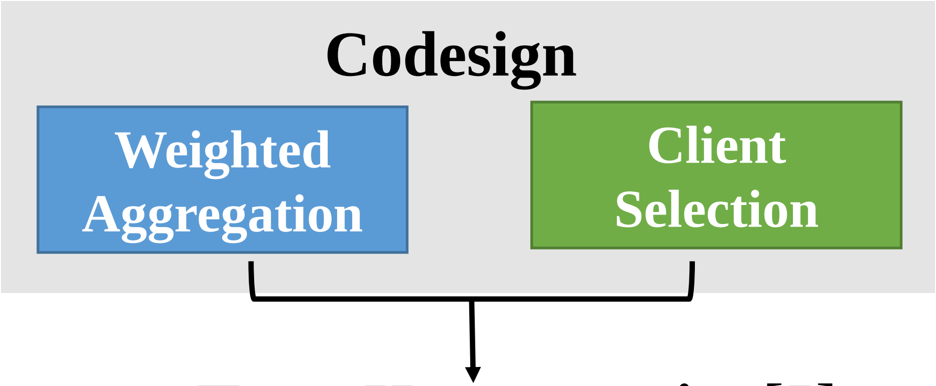
[3] Aono Y, Hayashi T, Wang L, et al. Privacy-preserving deep learning via additively homomorphic encryption[J]. IEEE transactions on information forensics and security, 2017, 13(5): 1333-1345.
[4] Zhang C, Li S, Xia J, et al. {BatchCrypt}: Efficient homomorphic encryption for {Cross-Silo} federated learning[C]//2020 USENIX annual technical conference (USENIX ATC 20). 2020: 493-506.

Limitations of Packed Homomorphic Encryption (PHE)

Causes and Challenges

- Why high costs

Time (s) / Clients	Training	Encryption	Idle	Decryption
Normal clients	3.24	6.68	8.25	4.65
Stragglers	6.19	12.24	2.00	9.69



Tame Heterogeneity [5]

- Statistical Heterogeneity
  - Difference in local models
  - Bias (Non-IID)

Slow Convergence

- System Heterogeneity
  - Computational Capabilities
  - Communication Bandwidth

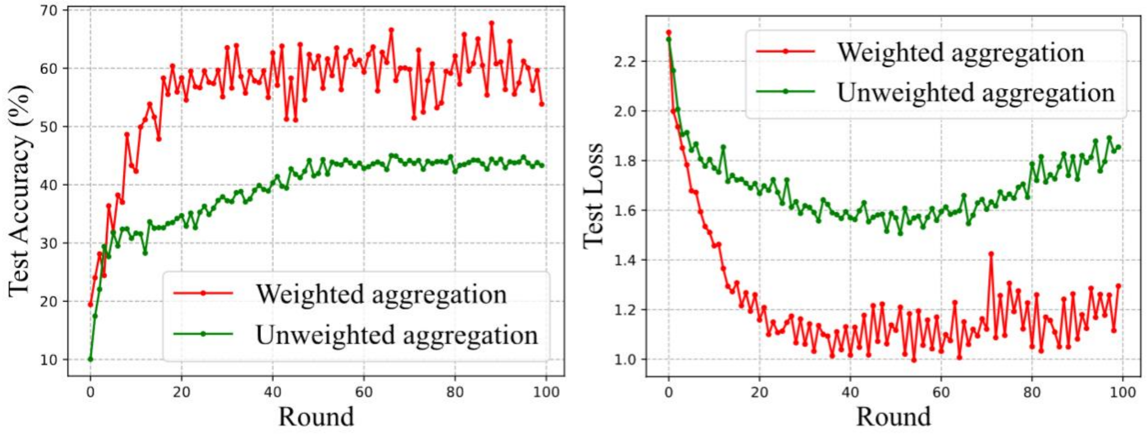
Straggler

[5] B. Luo, W. Xiao, S. Wang, J. Huang, and L. Tassiulas, "Tackling system and statistical heterogeneity for federated learning with adaptive client sampling," in Proc. IEEE INFOCOM, 2022, pp. 1739–1748.

Potential Solutions and Challenges

Causes and Challenges

- What to do with *weighted aggregation*



Weighted Aggregation [6]

- Average aggregation exacerbates bias
- Private data volume ≠ Contribution
- Client-side misreporting of weights

Server-side weighted is needed

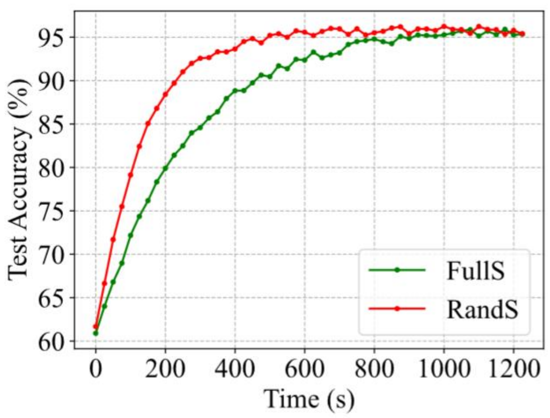
Challenge 1: contribution-based weighted aggregation in ciphertext

[6] Y. Deng, F. Lyu, J. Ren, Y.-C. Chen, P. Yang, Y. Zhou, and Y. Zhang, “Improving federated learning with quality-aware user incentive and auto-weighted model aggregation,” IEEE TPDS, vol. 33, no. 12, pp. 4515–4529, 2022.

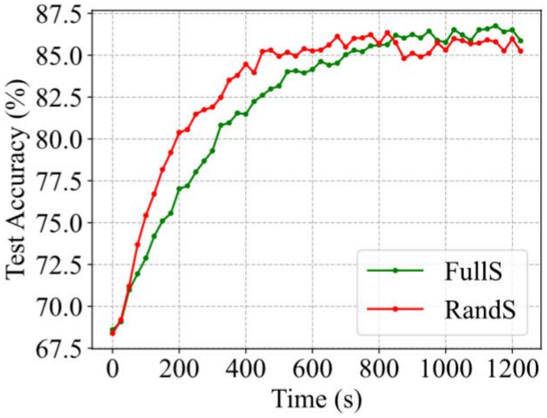
Potential Solutions and Challenges

Causes and Challenges

- What to do with *client selection*



(a) MNIST



(b) FashionMNIST

Client Selection [7]

- Different importance of clients
- Selection depends on model updates in plaintext

Privacy protection is more challenging

Challenge 2: efficient and secure client selection

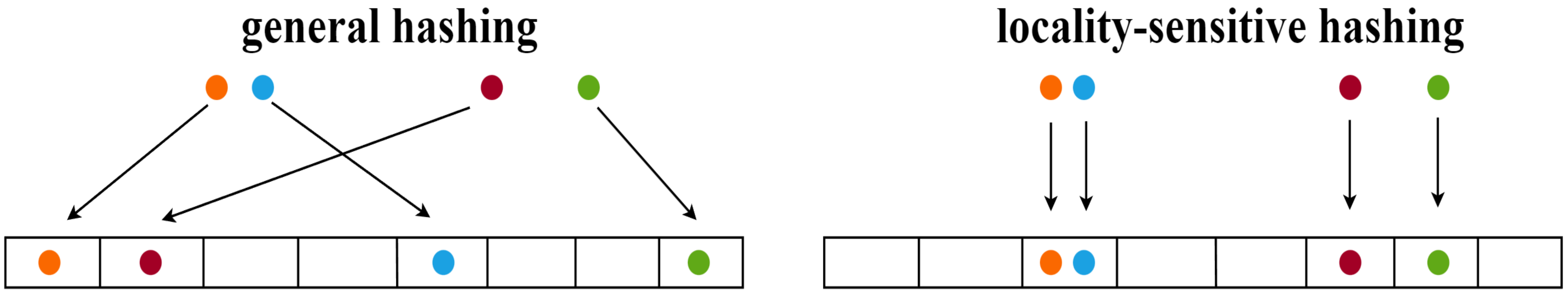
[7] F. Lai, X. Zhu, H. V. Madhyastha, and M. Chowdhury, "Oort: Efficient federated learning via guided participant selection," in Proc. USENIX OSDI, 2021, pp. 19–35.

System Overview: Preliminary Knowledge

Locality-Sensitive Hashing (LSH)

- Why is LSH [8]
  - Approximate nearest neighbor search
  - **Similarity is maintained**
  - Condensed representation

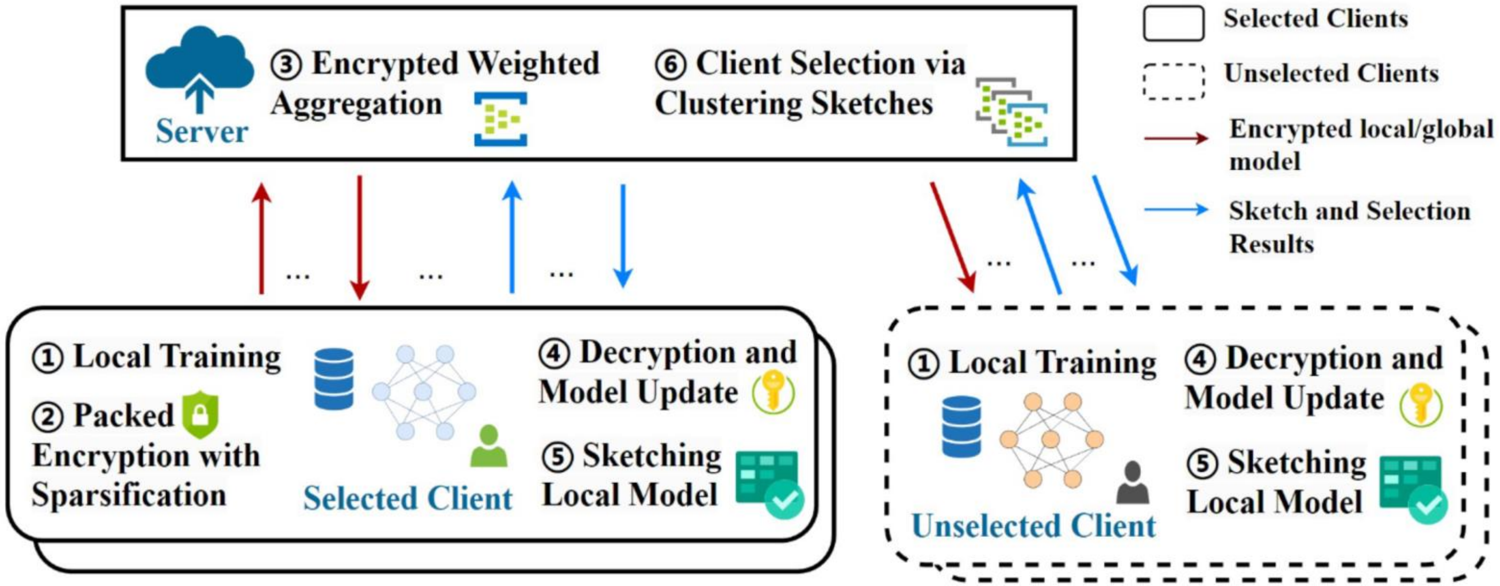
$$\text{If } d(w_p, w_q) < R, \text{ then } \Pr_{\mathcal{H}}(h(w_p) = h(w_q)) \geq p_1;$$
$$\text{If } d(w_p, w_q) \geq cR, \text{ then } \Pr_{\mathcal{H}}(h(w_p) = h(w_q)) \leq p_2;$$



[8] B. Neyshabur and N. Srebro, "On symmetric and asymmetric lshs for inner product search," in Proc. ICML, 2015, pp. 1926–1934.

System Overview: FedPHE Architecture

FedPHE



1. Clients produce gradients
2. Encrypt gradients and upload to Server
3. Server performs weighted aggregation on ciphertext
4. Clients receive aggregated ciphertext and update
5. Clients compute the sketch of local model
6. Server performs client selection

System Overview: CKKS Homomorphic Encryption

Packed HE Scheme

- Why is CKKS

	Paillier [9]	BFV [10]	CKKS [11]
Real Vector	✗	✗	☑
Homomorphic Multiplication	✗	☑	☑
Not Overflow	✗	✗	☑

[9] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” in Proc. Eurocrypt, 1999, pp. 223–238.
[10] J. Fan and F. Vercauteren, “Somewhat practical fully homomorphic encryption,” Cryptology ePrint Archive, 2012.
[11] J. H. Cheon, A. Kim, M. Kim, and Y. Song, “Homomorphic encryption for arithmetic of approximate numbers,” in Proc. ASIACRYPT, 2017, pp. 409–437.

System Overview: FedPHE Architecture

FedPHE: Contribution-aware encrypted weighted aggregation

Algorithm 1: FedPHE

Input: Clients $\mathcal{N}$, global round T, local steps E, learning rate η

Output: Global model $\boldsymbol{w}^T$

Initialize models $\{\boldsymbol{w}_i\}_{\mathcal{N}}$ and selected clients $\mathcal{S}^0 \leftarrow \mathcal{N}$;

// **Server**

for *each round* $t \in \{0, \cdots, T-1\}$ **do**

- Receive encrypted local models $\mathcal{C}_i^t$ and masks M_i^t from selected clients $i \in \mathcal{S}^t$;
- $\boldsymbol{C}^t, \boldsymbol{M}^t \leftarrow$ Run weighted aggregation by Alg. 2;
- Dispatch $\boldsymbol{C}^t$ and $\boldsymbol{M}^t$ to all clients;
- Receive sketches $\{h_i^t\}_{i \in \mathcal{N}}$ of clients' local models;
- $\mathcal{S}^{t+1} \leftarrow$ Run client selection by Alg. 3;
- Send $\mathcal{S}^{t+1}$ to clients;

// **Client** $i \in \mathcal{N}$

for *each round* $t \in \{0, \cdots, T-1\}$ **do**

- **for** $j = 0, \cdots, E-1$ **do**
  - $g_i(\boldsymbol{w}_{i,j}^t) \leftarrow \nabla f_i(\boldsymbol{w}_{i,j}^t)$;
  - $\boldsymbol{w}_{i,j+1}^t \leftarrow \boldsymbol{w}_{i,j}^t - \eta g_i(\boldsymbol{w}_{i,j}^t)$;
- $\boldsymbol{w}_i^t \leftarrow \boldsymbol{w}_{i,E}^t$;
- **if** $i \in \mathcal{S}^t$ **then**
  - $\mathcal{C}_i^t, M_i^t \leftarrow$ Run PHE and sparsification by Alg. 2;
  - Send $\mathcal{C}_i^t, M_i^t$ to the PS;
- Receive encrypted global model $\boldsymbol{C}^t$ and mask $\boldsymbol{M}^t$;
- $\boldsymbol{w}_i^t \leftarrow$ Decrypt and update with global model $\boldsymbol{w}^t$;
- Send sketch h_i^t of $\boldsymbol{w}_i^t$ to the PS by Alg. 3;
- Receive the selection set $\mathcal{S}^{t+1}$ from the PS;

- **Server: encrypted weighted aggregation**

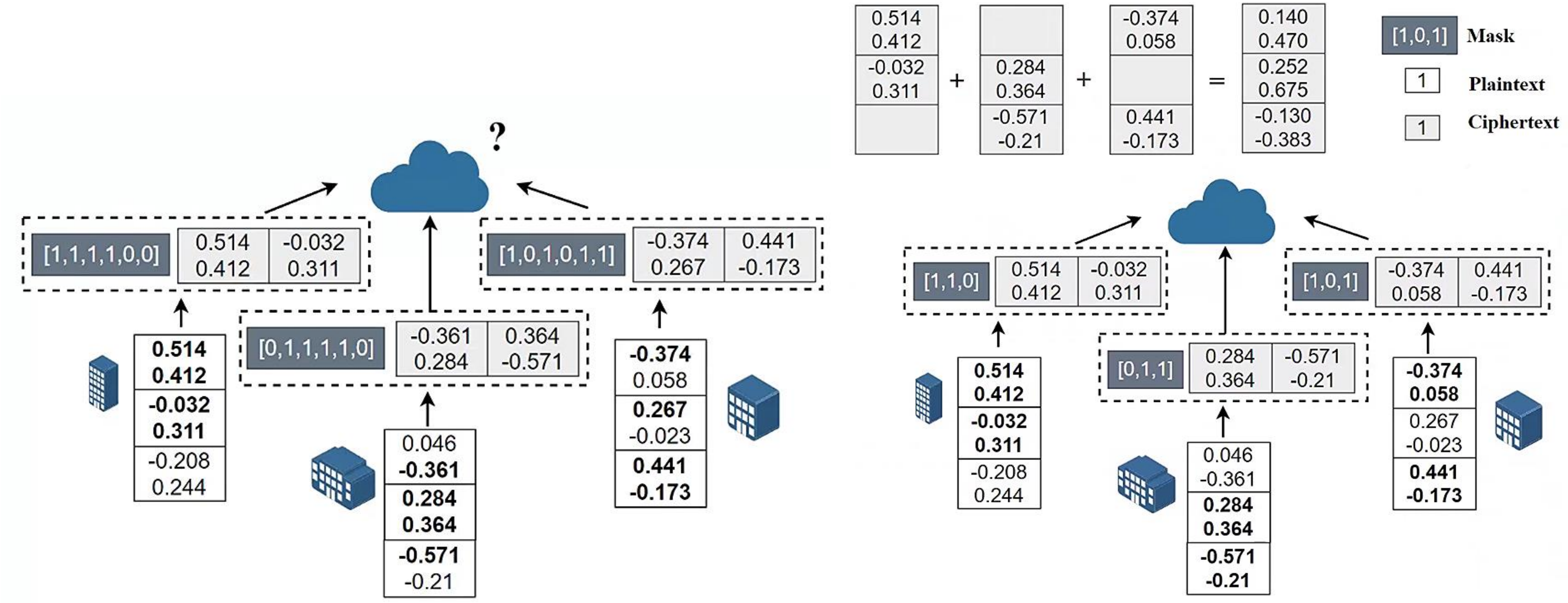
- **Server: selects clients**

- **Clients: packed homomorphic encrytion and sparsification**
- **Clients: sketch of local models**

System overview: FedPHE Architecture

FedPHE: Contribution-aware encrypted weighted aggregation

- Pack level sparsification



System overview: FedPHE Architecture

FedPHE Contribution-aware encrypted weighted aggregation

- Encrypted weighted aggregation
  - Hash collisions probability --> Jaccard similarity

$$\Pr_{\mathcal{H}}(h_i^{t-1} = h_i^t) = JS(w_i^{t-1}, w_i^t) \quad JS(X,Y) = |X \cap Y| / |X \cup Y|$$

Low similarity --> High contribution

$$p_i^t = \frac{\exp(-\beta \cdot JS(w_i^{t-1}, w_i^t))}{\sum_{j \in \mathcal{S}^t} \exp(-\beta \cdot JS(w_j^{t-1}, w_j^t))}$$

$$\boldsymbol{E}(\boldsymbol{w}^{t+1}) = \sum\nolimits_{i \in \mathcal{S}^t} \boldsymbol{E}\left(p_i^t\right) \times \boldsymbol{E}\left(\boldsymbol{w}_i^t\right) \Longrightarrow \boldsymbol{E}(\boldsymbol{w}^{t+1}) = \sum\nolimits_{i \in \mathcal{S}^t} \boldsymbol{E}\left(p_i^t \times \boldsymbol{w}_{\boldsymbol{i}}^t\right)$$

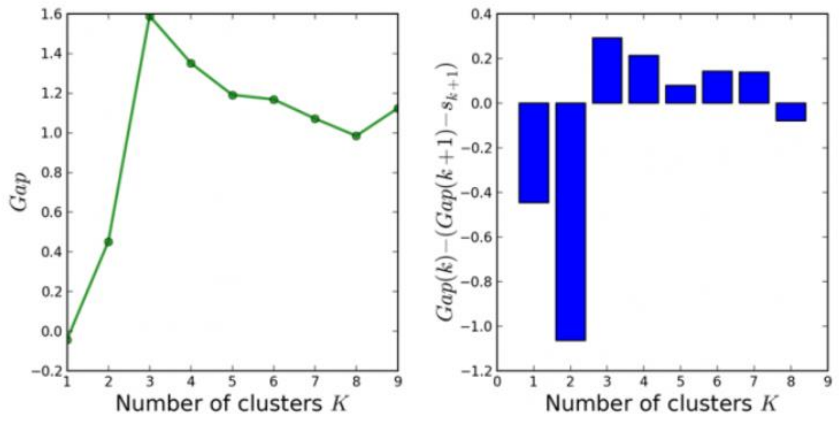
Sketching-based Client Selection

Clustering sketches

- Gap statistic [12]
  - Optimal number k of clusters
  - *Monte Carlo* simulation and intra-class variation

$$\mathbb{G}_n(k) = \mathbb{E}_n^*(\log W_k) - \log W_k \qquad \mathbb{G}_k \geq \mathbb{G}_{k+1} - s_{k+1}$$

- LSH properties
  - **Similar sketches** mean **similar models**



[12] R. Tibshirani, G. Walther, and T. Hastie, “Estimating the number of clusters in a data set via the gap statistic,” Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 63, no. 2, pp. 411–423, 2001

Sketching-based Client Selection

Selecting Clients

- Priority function
  - Selecting clients to train ***quickly***
  - **Historical** engagement performance
- Compared to traditional *cosine* similarity
  - Computation and communication **efficient**
  - **Privacy** protection

$$\mathbb{F}_i^t = \frac{1}{\alpha\delta_i^{t-1} + (1-\alpha) \times T_i^t}$$

$$\delta_i^{t-1} = \frac{1}{t}\sum_{j=0}^{t} T_i^j$$

Performance Evaluation

Evaluation Setup

- Dirichlet Non-IID setting ($\alpha = 1$)
- LSH setting $k = 200$

Datasets and models

- MNIST, LeNet-5
- FashionMNIST, CNN
- CIFAR-10, ResNet-20

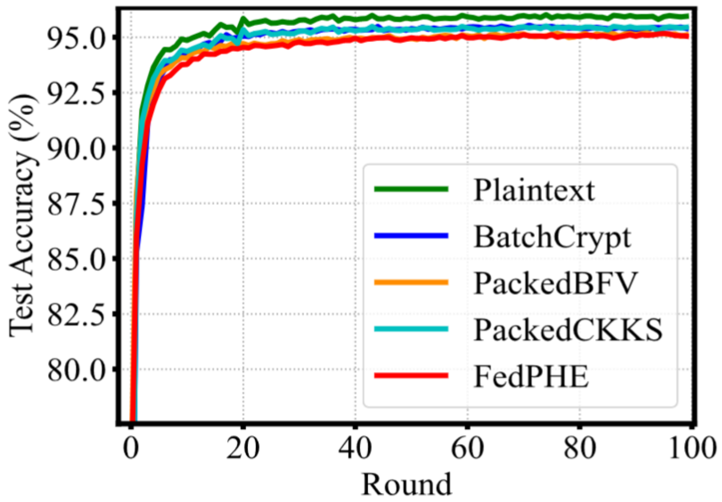
Benchmarks

Baseline	Method
Plaintext	No encryption (Upper bounds)
BatchCrypt	Packed HE based on Paillier
PackedBFV	Packed HE based on BFV
PackedCKKS	Packed HE based on CKKS
FedAvg	Randomly select clients
FLANP	Adaptively add clients

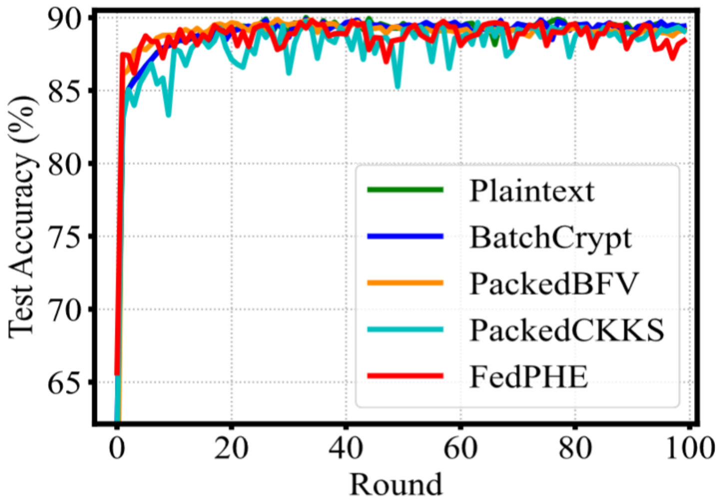
Performance Evaluation

Accuracy

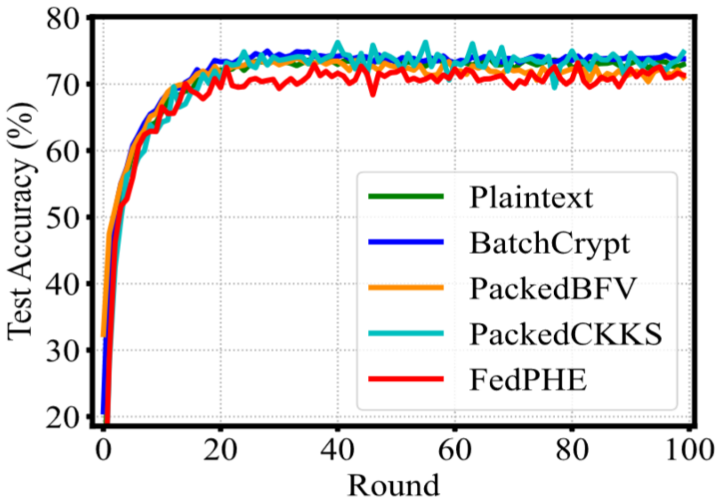
- Packing encryption will not cause accuracy decrease
- FedPHE fluctuates 0.26% − 1.58% due to sparsification and client selection



(a) MNIST



(b) FashionMNIST



(c) CIFAR-10

Performance Evaluation

Network traffic and training time

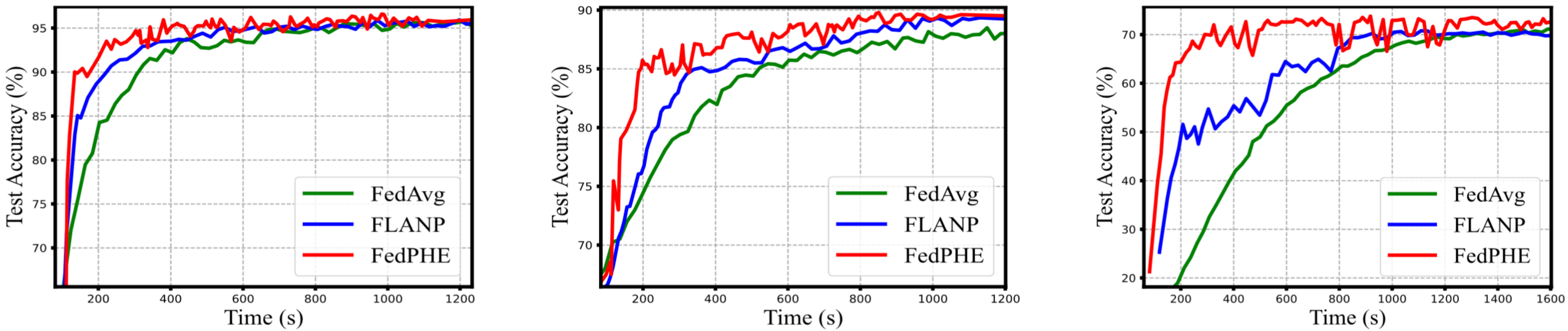
- Training acceleration 1.85-4.44×
- Reduce communication overhead by 1.24-22.62×

Dataset	Metric	Plaintext	BatchCrypt	PackedBFV	PackedCKKS	FedPHE
MNIST	Traffic (MB)	81	**217**	3959	2886	**175**
	Accuracy	95.94%	95.50%	95.10%	95.44%	95.04%
	Time (s)	342.34	**1377.03**	652.78	885.90	**743.26**
FashionMNIST	Traffic (MB)	73	**196**	3300	2550	**151**
	Accuracy	89.22%	89.42%	89.10%	89.07%	88.96%
	Time (s)	333.15	**1590.23**	650.68	823.88	**690.65**
CIFAR-10	Traffic (MB)	523	**1256**	22107	17089	**5165**
	Accuracy	72.95%	73.79%	71.02%	74.77%	71.37%
	Time (s)	1016.62	**7126.14**	1491.74	2419.18	**1605.71**

Performance Evaluation

Network traffic and training time

- FedPHE achieved faster convergence speed than FedAvg and FLANP
- FedPHE is more effective against stragglers

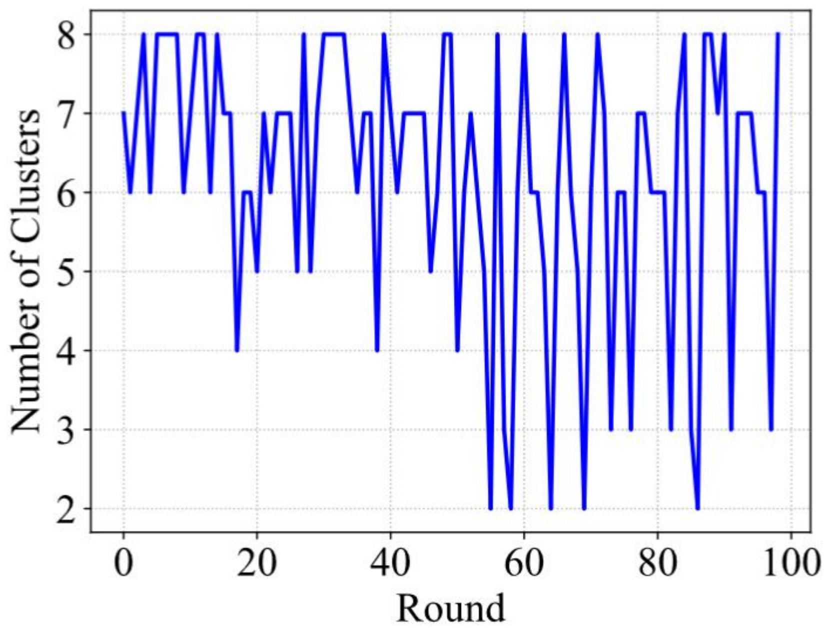


(a) MNIST (b) FashionMNIST (c) CIFAR-10

Performance Evaluation

Comparison of number of clusters

- Decreasing number indicates higher similarity between local models
- Dynamically determine the cluster number

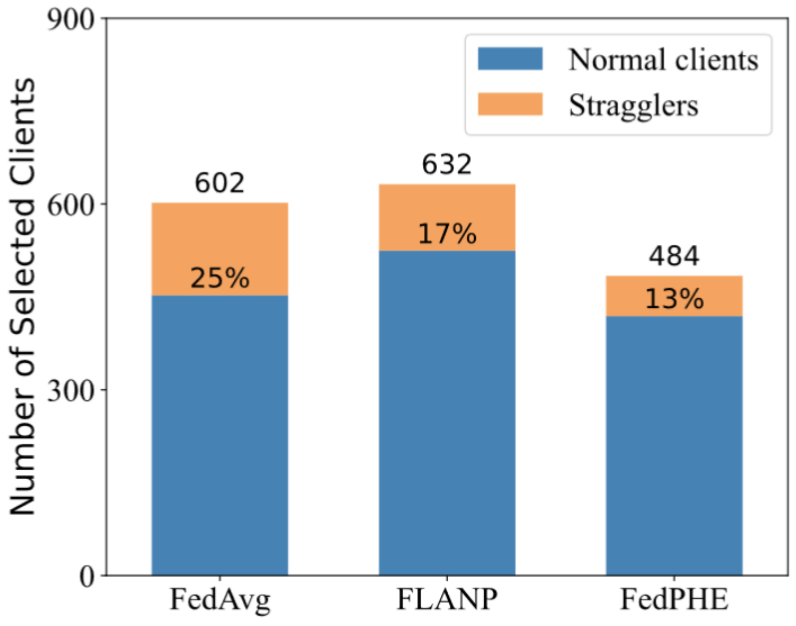


(a) Number of Clusters

Performance Evaluation

Mitigate Straggler

- FedAvg still has 25% stragglers
- FLANP still has 17% stragglers
- FedPHE has only 13% of stragglers
- FedPHE selects a representative subset of clients and can also optimize communications
- Mitigate the impact of stragglers 1.71-2.39×



(b) Number of Selected Clients

Conclusions

Design Goal

- Privacy Protection
- Efficiency
- Straggler Resistance

FedPHE Architecture

- Encrypted weighted aggregation
  - CKKS-based PHE (privacy)
  - Contribution-aware (accuracy)
  - Pack-level sparsification (efficiency)
- Sketch-based client selection
  - Sketching local models (privacy)
  - Clustering sketches (efficiency)
  - Selecting clients (straggler resistance)

Conclusions

FedPHE Results

- Training speed is increased by 1.85-4.44×
- Communication overhead is reduced by 1.24-22.62×
- Model accuracy only dropped by 1.58%

Thank you for coming!   FedPHE is open sourced at https://github.com/lunan0320/FedPHE

nanyan@whu.edu.cn

Q&A