清華大学智能产业研究院
Institute for AI Industry Research,Tsinghua University

Knowledge Transfer Federated Learning

Yang Liu
Institute for AI Industry Research （AIR），Tsinghua University

Outline

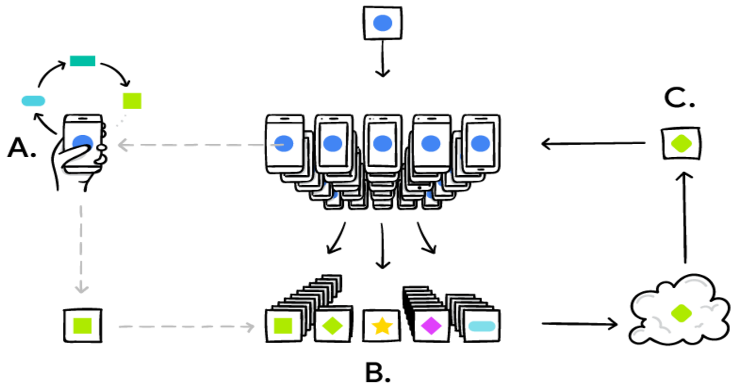
I. Knowledge Transfer (KT)- Federated Learning (FL)

II. Addressing challenges in KD-based FL

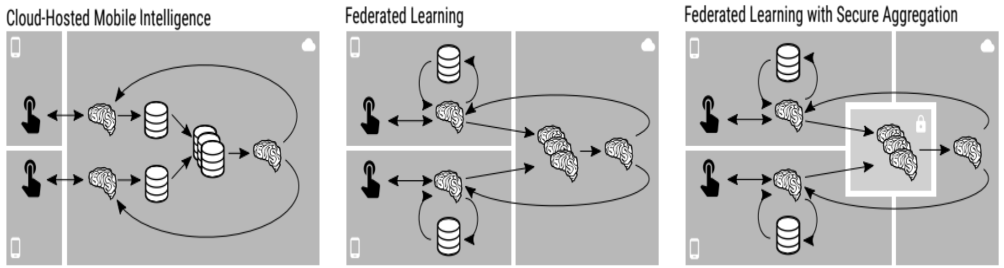
III. Vertical FL

Cross-device vs Cross-silo FL

Google's FML (Cross-device)

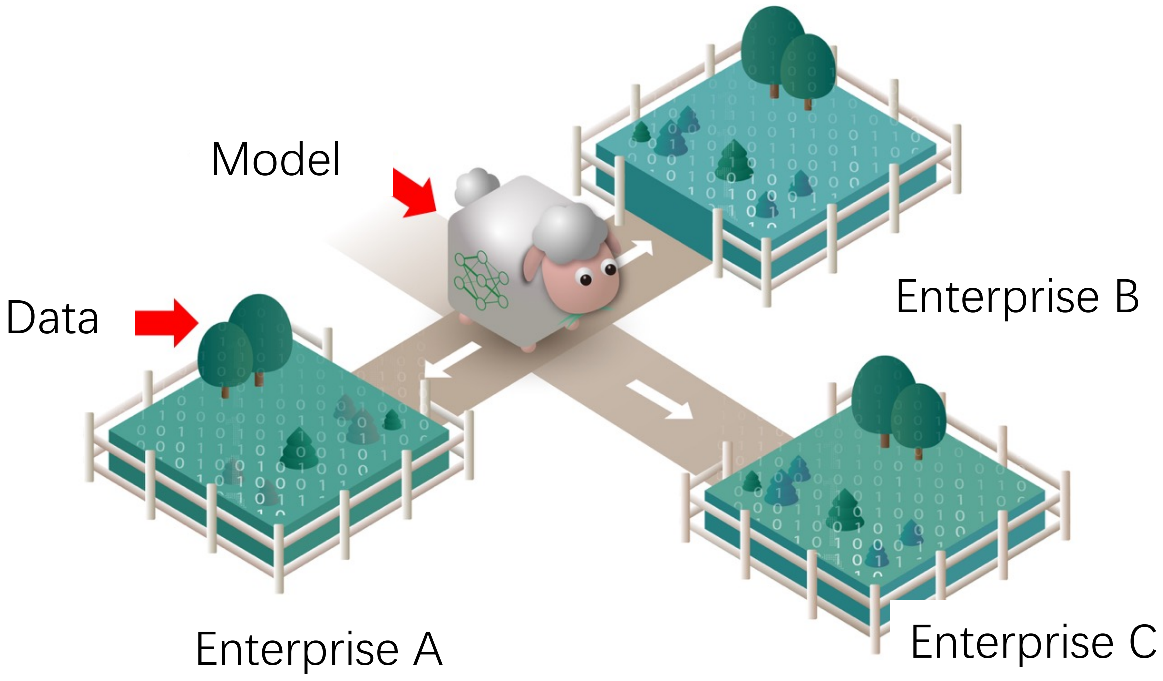


H. Brendan McMahan et al, *Communication-Efficient Learning of Deep Networks from Decentralized Data*, Google, 2017



Keith Bonawitz et al, *Practical Secure Aggregation for Privacy-Preserving Machine Learning*, Google, 2017

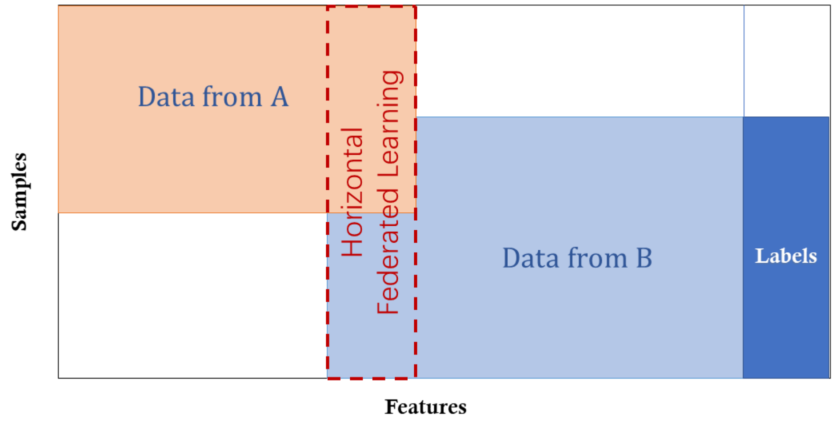
Cross-silo Federated Learning



Advances and open problems in Federated Learning, *Foundations and Trends in Machine Learning: Vol. 14: No. 1–2, pp 1-210*

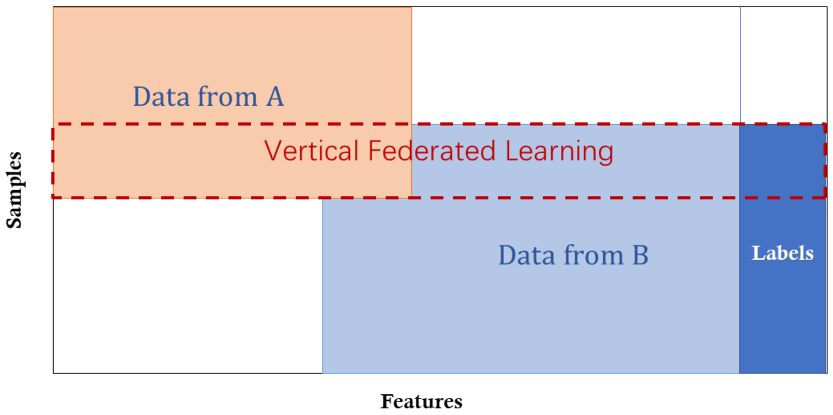
Horizontal, Vertical FL and FTL

Horizontal FL



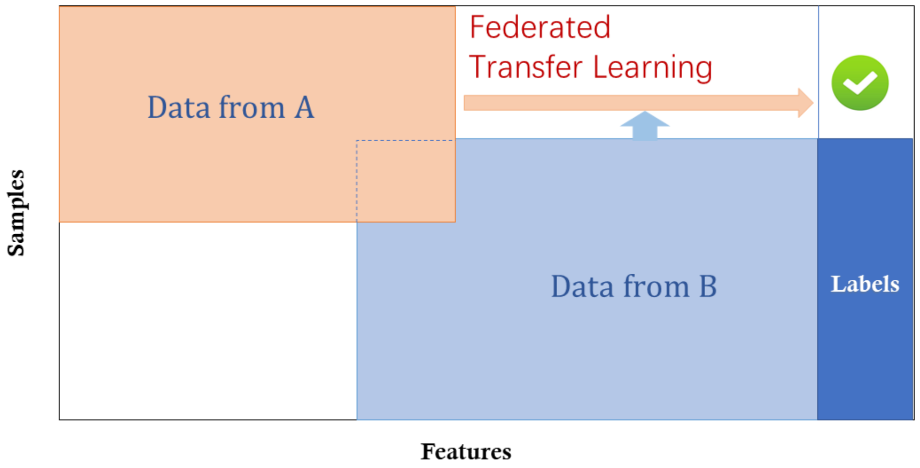
- Large overlap of features of the two data sets

Vertical FL



- Large overlap of sample IDs (users) of the two data sets

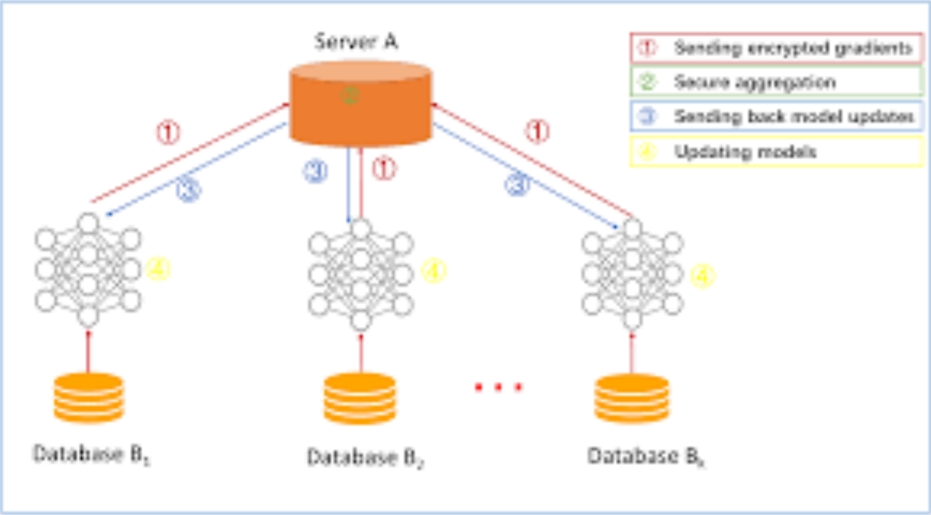
Federated Transfer Learning



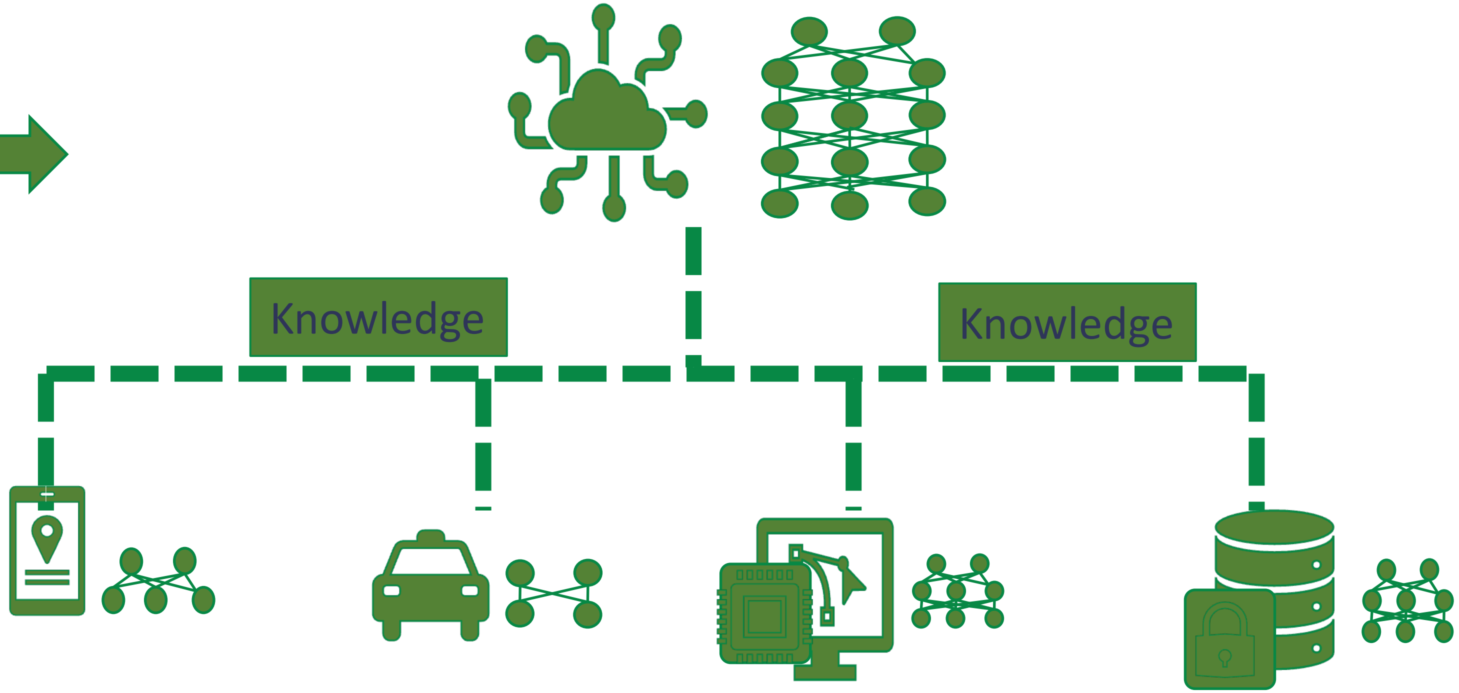
Q. Yang, Y. Liu, T. Chen & Y. Tong, Federated machine learning: Concepts and applications, *ACM Transactions on Intelligent Systems and Technology (TIST)* **10**(2), 12:1-12:19, 2019

Model Transfer and Knowledge Transfer FL

Model Transfer FL



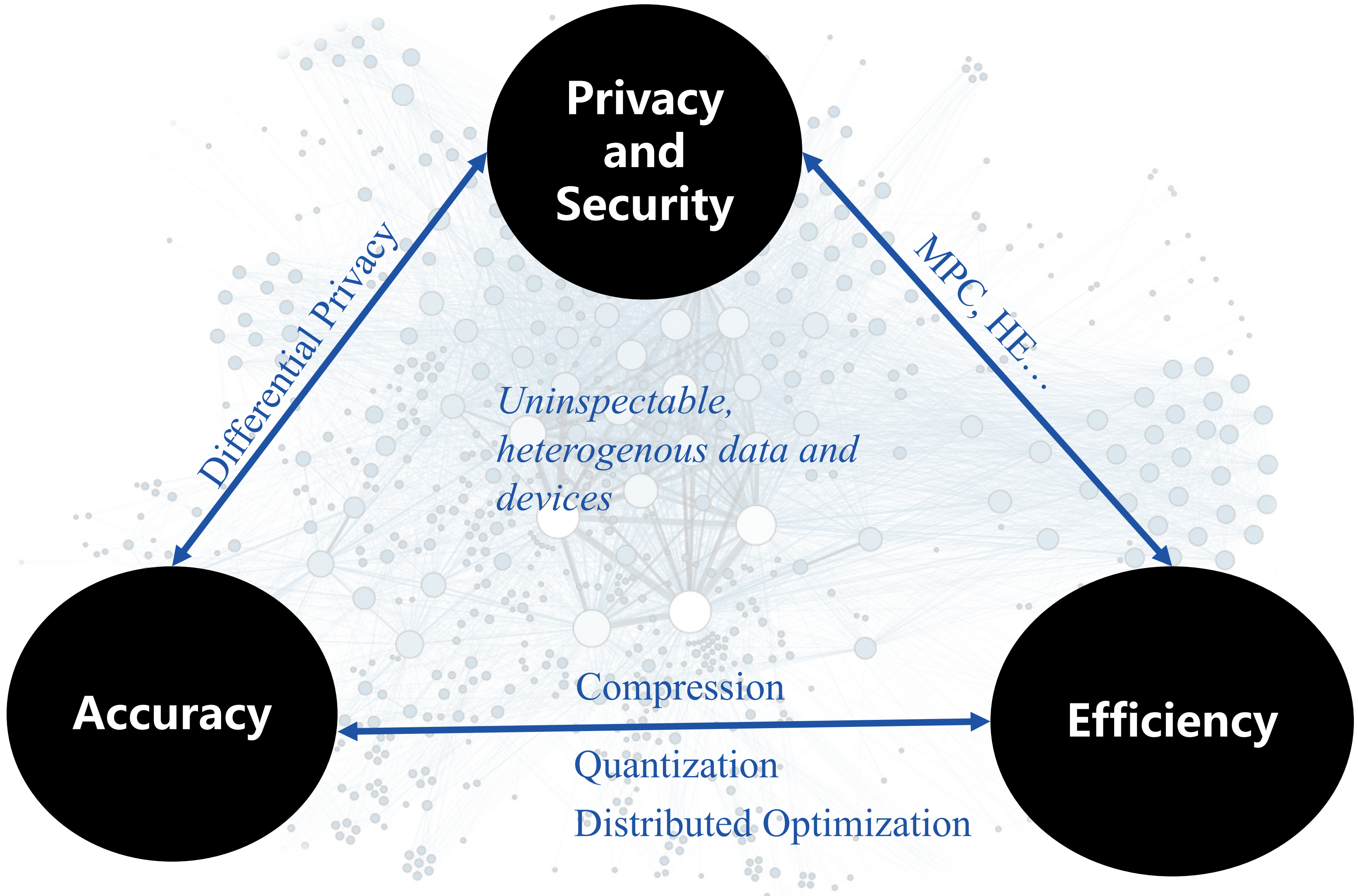
Knowledge Transfer FL



Examples of Knowledge Transfer FL

- Knowledge Distillation(KD)-based FL
- Vertical Federated Learning
- Federated Transfer Learning
- ....

Addressing Privacy-Accuracy-Efficiency Trilemma over various heterogeneity



Outline

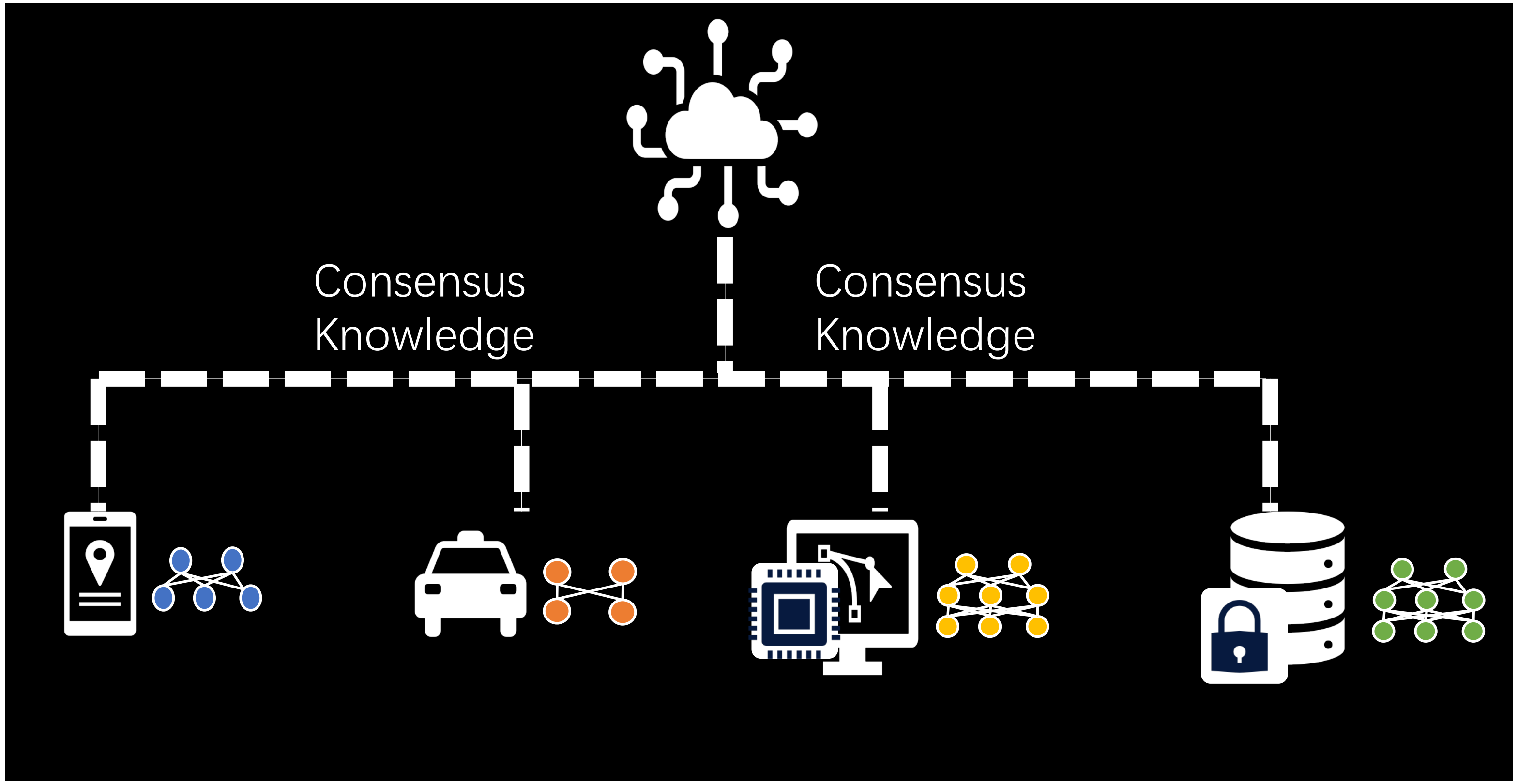
I. Knowledge Transfer (KT)- Federated Learning (FL)

II. Addressing challenges in KD-based FL

III. Vertical FL

KD-based Federated Learning

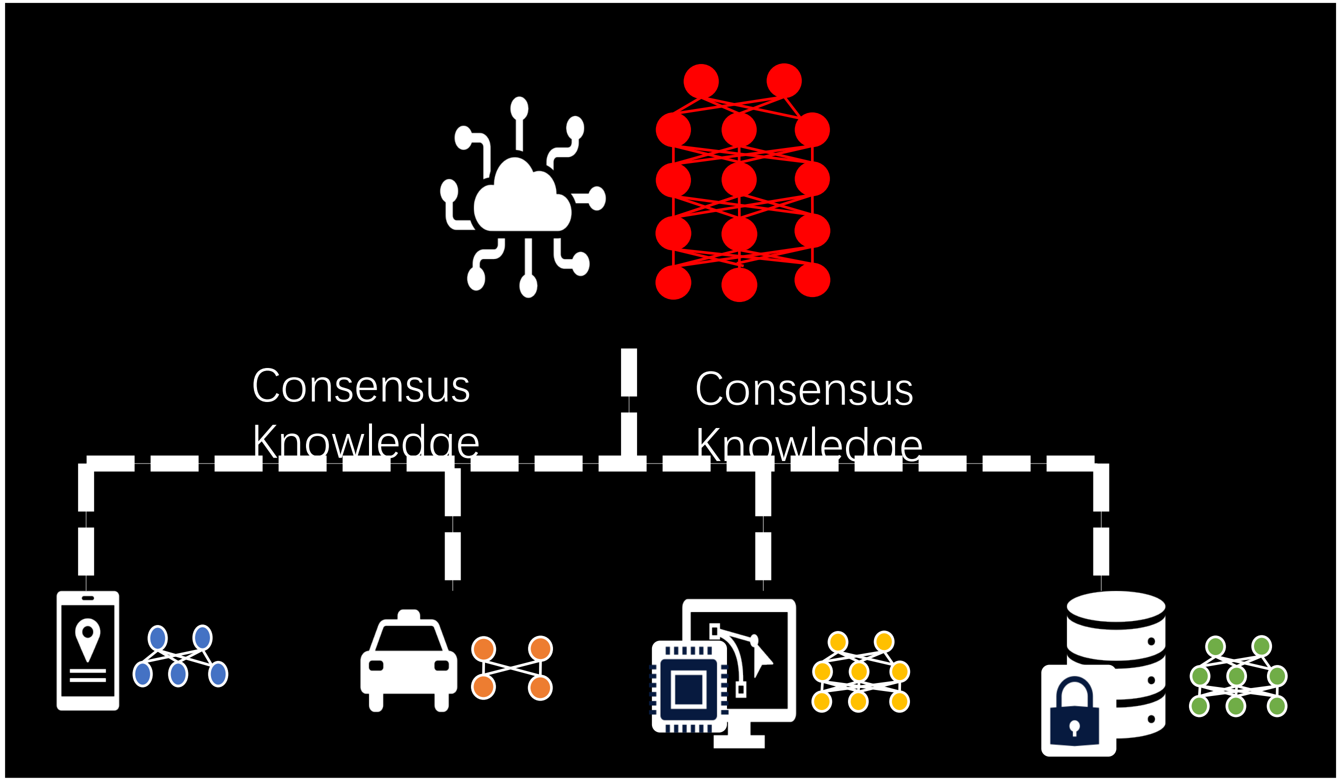
Transfer **<u>Knowledge</u>** instead of **model parameters.**

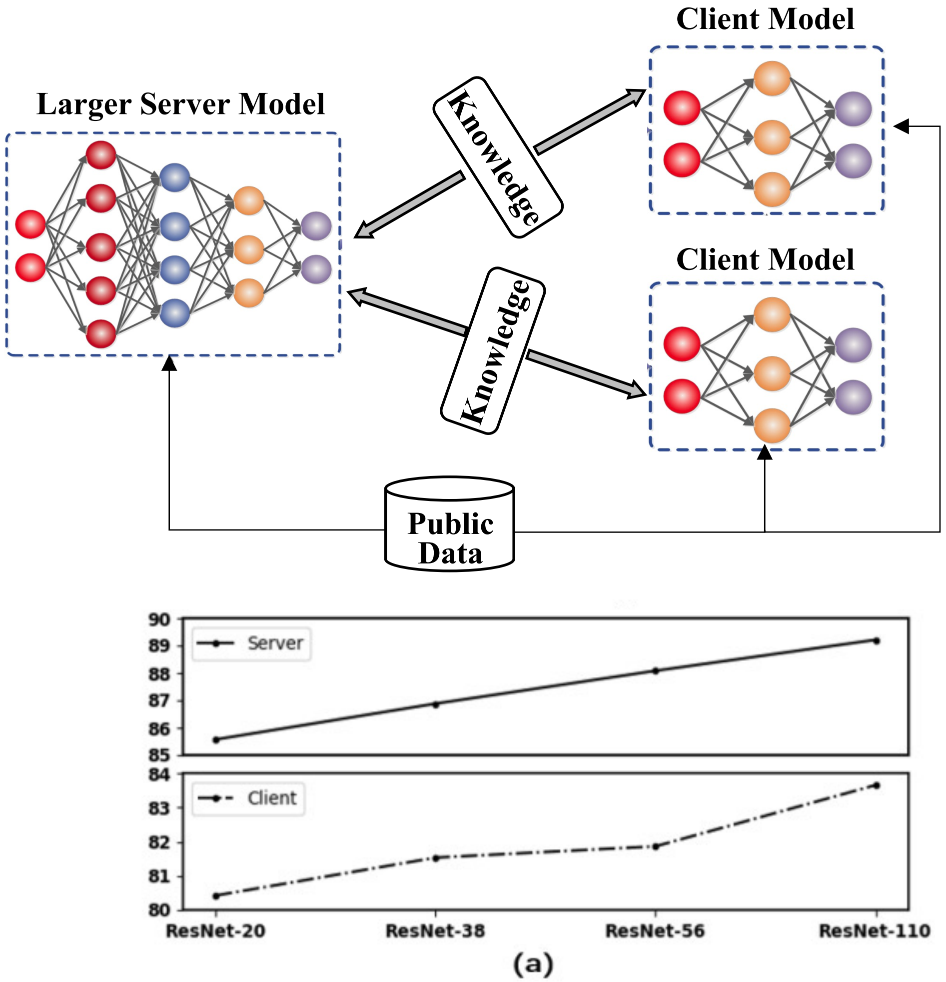


Advantages

- ✓ Heterogeneity
- ✓ Privacy
  - Gradient Leakage

FedGEMS: Federated Learning of Larger Server Models





Model performances of different server model sizes.

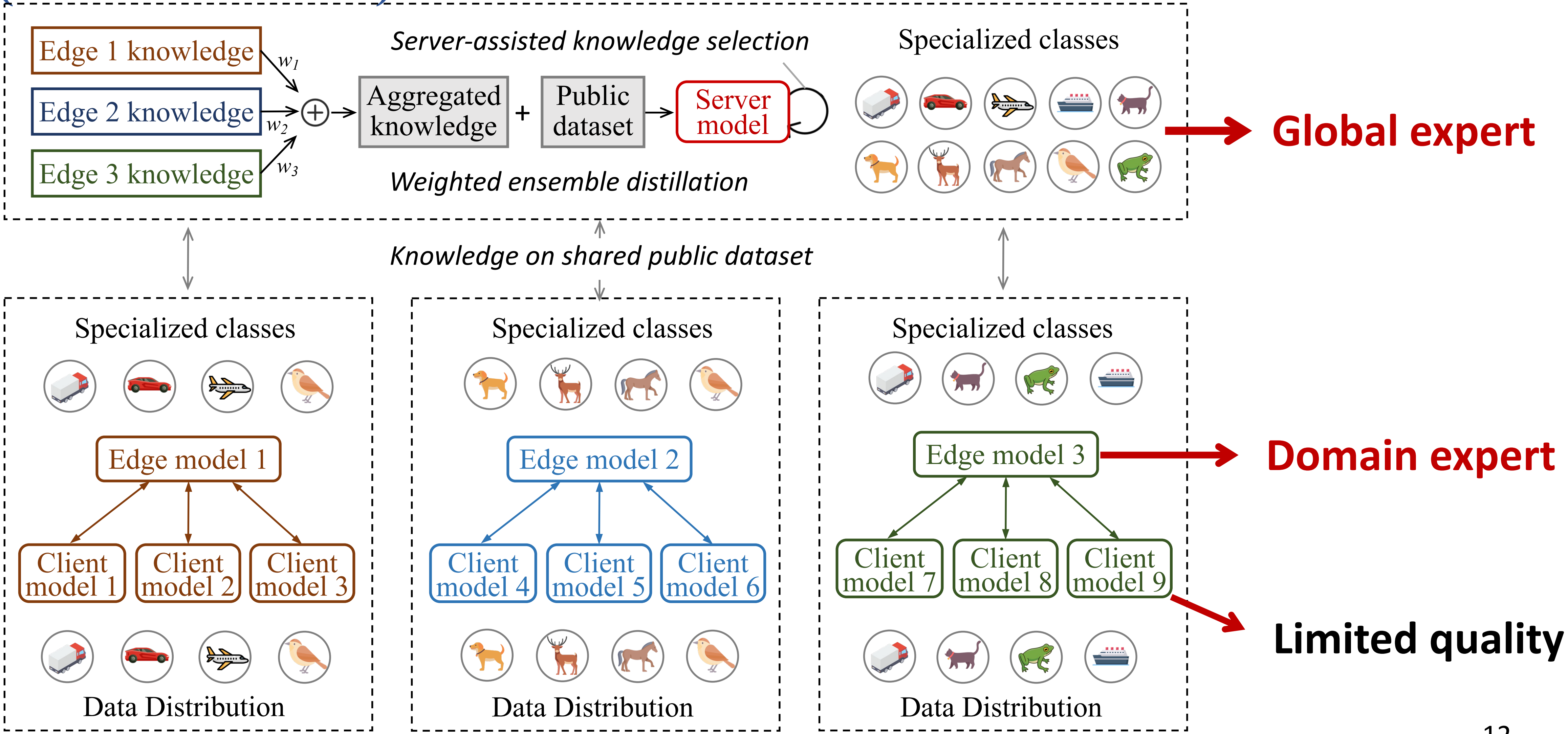
S Cheng, J Wu, Y Xiao, Y Liu*, Y Liu*, FedGEMS: Federated Learning of Larger Server Models via Selective Knowledge Fusion, https://arxiv.org/abs/2110.11027

Efficient knowledge transfer using unlabeled public dataset

➢ Challenges

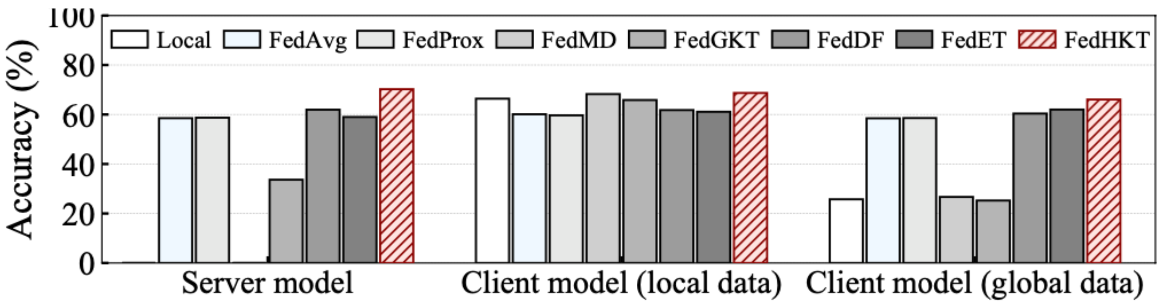
1. **Knowledge from clients: limited quality**
   - The learning models of clients are small
   - The training data size and categories of clients are limited
2. **Knowledge fusion methods: limited efficacy**
   - Clients have **diverse classification expertise** on various labels of samples
   - **Knowledge quality** provided by a client varies from samples
   - Unlabeled public samples **lack ground truths** to evaluate knowledge quality

FedHKT: A Hierarchical Knowledge Transfer Framework for Heterogeneous Federated Learning (INFOCOM'23)

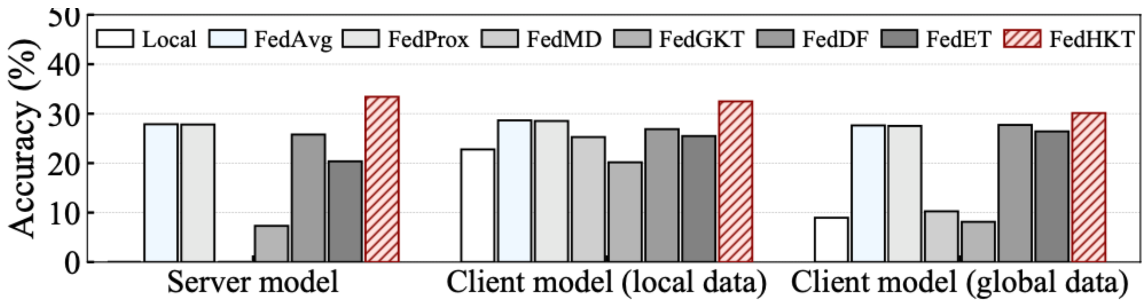


Evaluation Results

➢ Homogeneous model settings



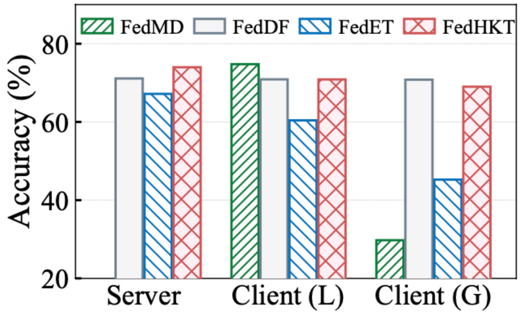
(c) CIFAR-10, 50 clients



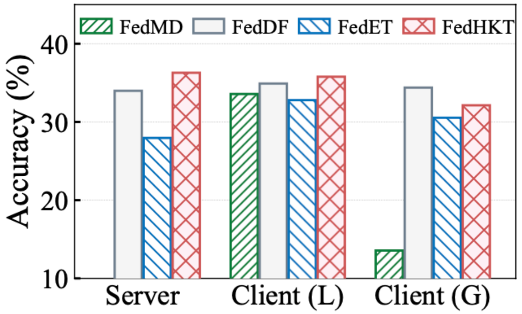
(d) CIFAR-100, 50 clients

- **Significant accuracy gain for server model**
- **Improved personalization and generalization performance for client model**

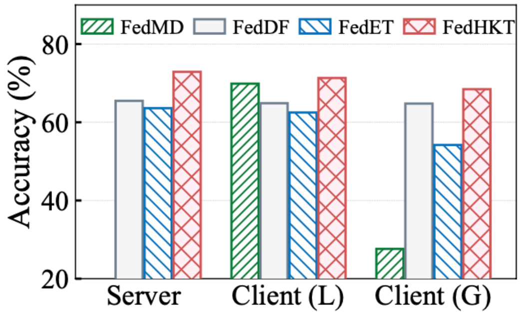
➢ Heterogeneous model settings



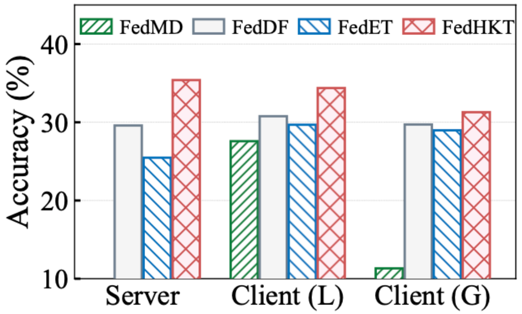
(a) CIFAR-10, 20 clients



(b) CIFAR-100, 20 clients



(c) CIFAR-10, 50 clients

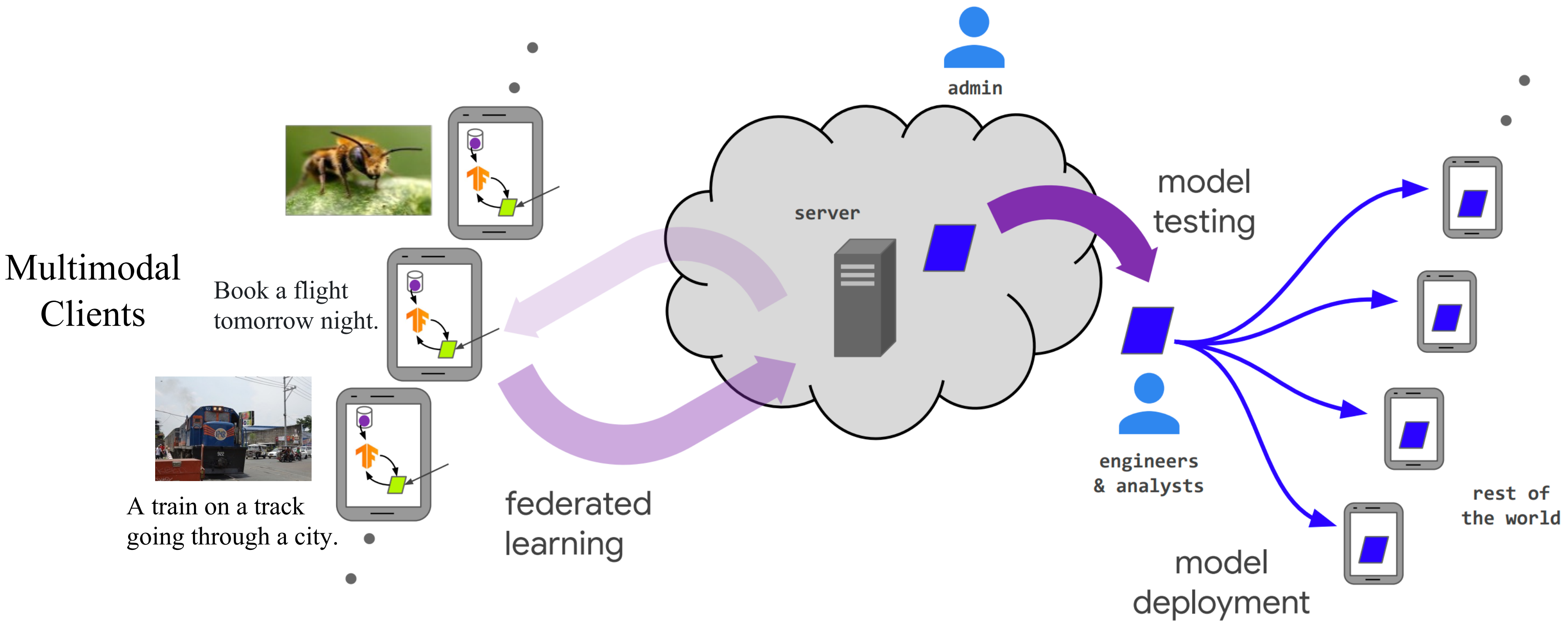


(d) CIFAR-100, 50 clients

- **Efficient knowledge transfer between server model and heterogeneous client models**

Multimodal Federated Learning

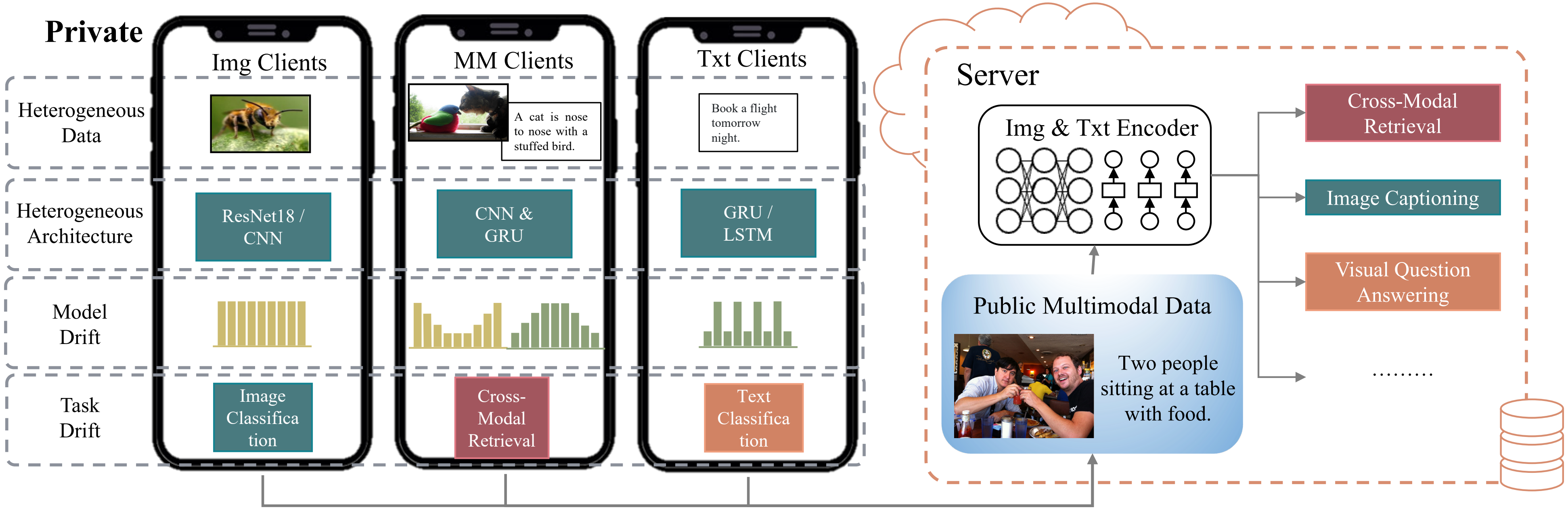
With the increasing amount of **multimedia** data on modern mobile systems and IoT infrastructures, harnessing these rich data without breaching user privacy becomes a critical issue.



- Figure adapted from "Advances and open problems in federated learning."

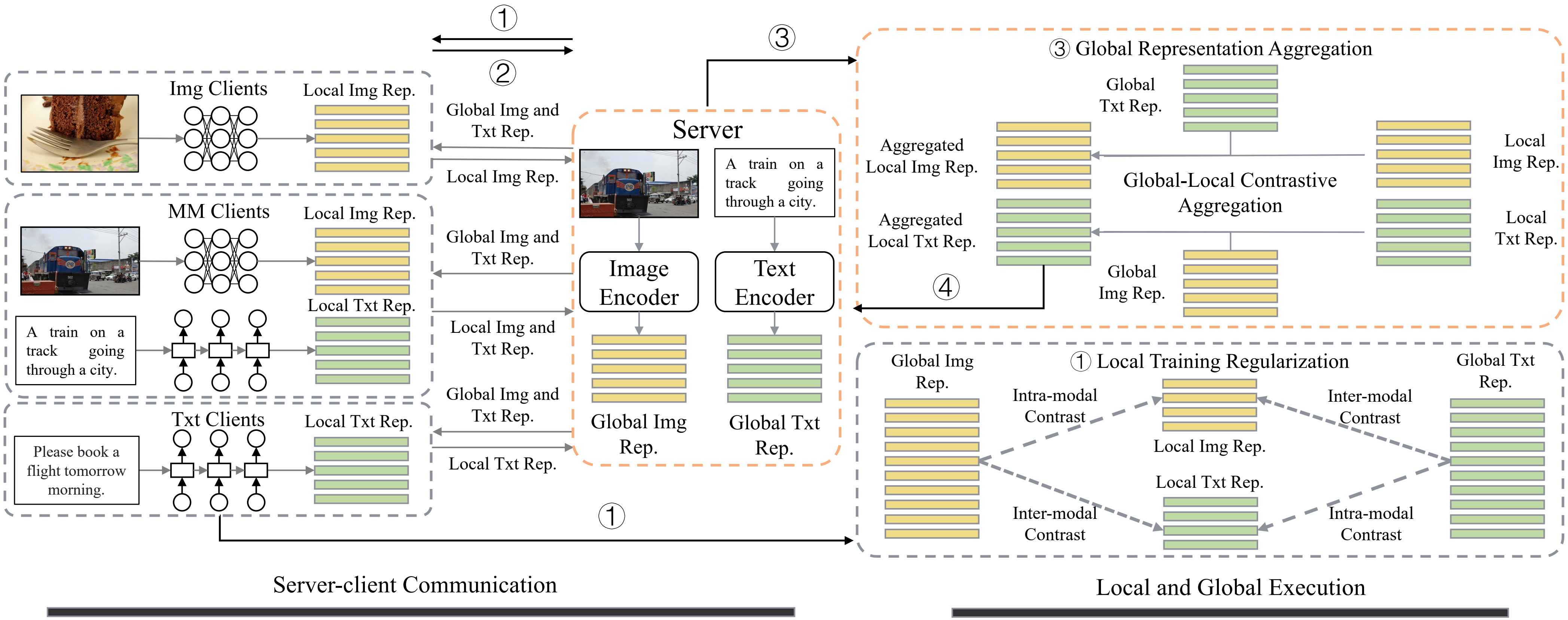
Challenges in Multimodal FL (MMFL)

- Model drift: two new unprecedented heterogeneous factors arise from multimodal discrepancy, modality gap and task gap.
- Existing MMFL methods all adopt FedAvg framework by using homogeneous models for each modality, restraining the complexity of the global model to smaller scales.
- Existing algorithms for larger server model training rely on knowledge distillation through logit, which only limited to classification tasks and not suitable for representation-based tasks like retrieval.



Contrastive Representation Ensemble and Aggregation (CreamFL， ICLR 2023)

CreamFL enables training larger server models from clients with heterogeneous model architectures and data modalities through representation ensemble transfer on public data, meanwhile effectively addressing the model drift challenge.

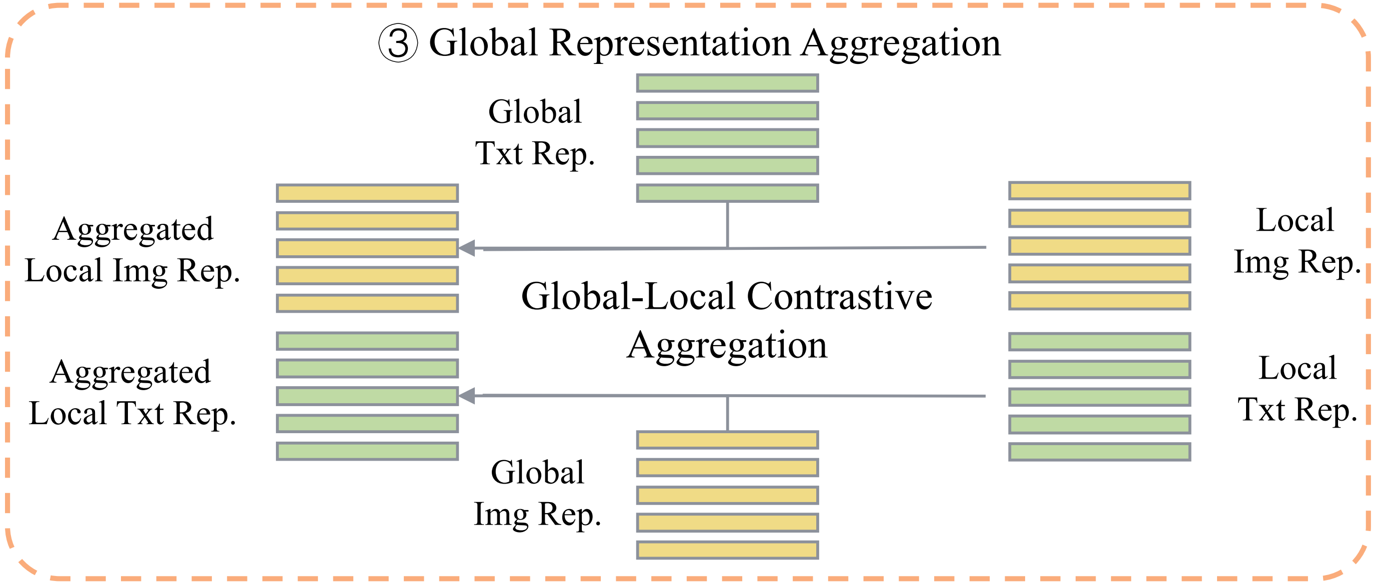


CreamFL: Global Contrastive Aggregation (GCA)

For global representations aggregation, we design a global-local cross-modal contrastive score for weighting purposes. The score for kth image of cth client is computed as:

$$s^{(k,c)} = \log \frac{\exp\left({\boldsymbol{i}_{\text{local}}^{(k,c)}}^{\top} \cdot \boldsymbol{t}_{\text{global}}^{(k)}\right)}{\sum_{j=1}^{|\mathcal{P}|} \mathbf{1}_{[j \neq k]} \exp\left({\boldsymbol{i}_{\text{local}}^{(k,c)}}^{\top} \cdot \boldsymbol{t}_{\text{global}}^{(j)}\right)}$$

We assign a higher weight to the local representation $\boldsymbol{i}_{\text{local}}^{(k,c)}$ that better matches its counterpart's global representation $\boldsymbol{t}_{\text{global}}^{(k)}$ (nominator), and less approximates other texts $\boldsymbol{t}_{\text{global}}^{(j)}, j \neq k$ (denominator).



Experiments

Settings:
- Public dataset: a subset of COCO
- Private datasets: CIFAR-100, AG_NEWS, Flicker30k
- Clients: 10 image clients, 10 text clients, 15 multimodal clients. 10 of them are randomly chosen to participate in each round training.

CreamFL achieves noticeable performance improvement over all baselines in all settings.

Table 1: Comparison of CreamFL with baselines on image-text retrieval task.

Types	Methods	1K Test Images						
		i2t_R@1	i2t_R@5	i2t_R@10	t2i_R@1	t2i_R@5	t2i_R@10	R@1_sum
w/o larger server model	FedAvg	29.38	59.84	73.52	23.71	56.86	72.95	74.20
	FedIoT	28.62	59.90	73.82	23.36	58.14	74.55	72.15
w/ larger server model	FedMD	32.88	66.64	80.02	28.26	64.23	79.58	86.07
	FedET	33.42	67.28	80.20	28.29	64.56	79.62	87.17
	FedGEMS	34.44	67.52	80.50	28.73	64.82	80.00	88.92
	reamFL+Avg	34.01	67.56	79.72	28.52	64.36	79.57	88.13
	reamFL+IoT	33.90	66.28	80.18	28.44	64.70	80.03	88.05
	CreamFL (ours)	**35.76**	**68.28**	**81.52**	**29.06**	**65.19**	**80.36**	**92.43**
Types	**Methods**	**5K Test Images**						
		i2t_R@1	i2t_R@5	i2t_R@10	t2i_R@1	t2i_R@5	t2i_R@10	
w/o larger server model	FedAvg	11.86	31.46	44.08	9.25	26.82	39.02	
	FedIoT	11.40	29.62	43.16	8.77	26.88	39.56	
w/ larger server model	FedMD	13.24	35.50	48.90	11.69	32.58	46.46	
	FedET	13.68	36.62	49.70	11.78	32.73	46.26	
	FedGEMS	13.94	37.32	50.78	11.81	33.01	46.54	
	reamFL+Avg	13.92	36.60	49.79	11.68	32.78	46.29	
	reamFL+IoT	14.06	36.58	49.14	11.65	33.01	46.64	
	CreamFL (ours)	**15.08**	**37.86**	**51.56**	**12.53**	**33.63**	**47.23**	

Experiments: Ablation

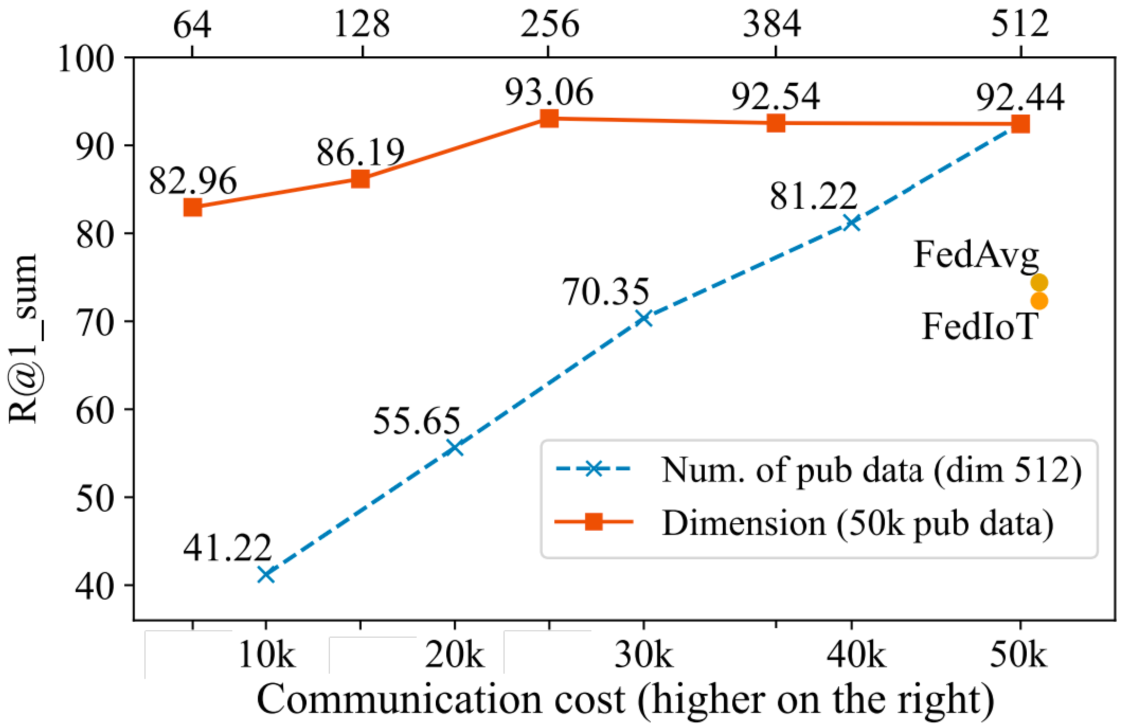
清華大学智能产业研究院
Institute for AI Industry Research,Tsinghua University

Ablation Studies for different components of CreamFL:

Methods	R@1_sum
reamFL+Mean	85.75
reamFL+Avg	88.13
reamFL+IoT	88.05
reamFL+GCA	90.03
reamFL+GCA + LCR.inter	91.98
reamFL+GCA + LCR.intra	90.84
CreamFL (reamFL+GCA+LCR)	**92.43**

GCA:global-local contrastive aggregation
LCR: local contrastive regularization
reamFL: vanilla representation ensemble (CreamFL without 'C')

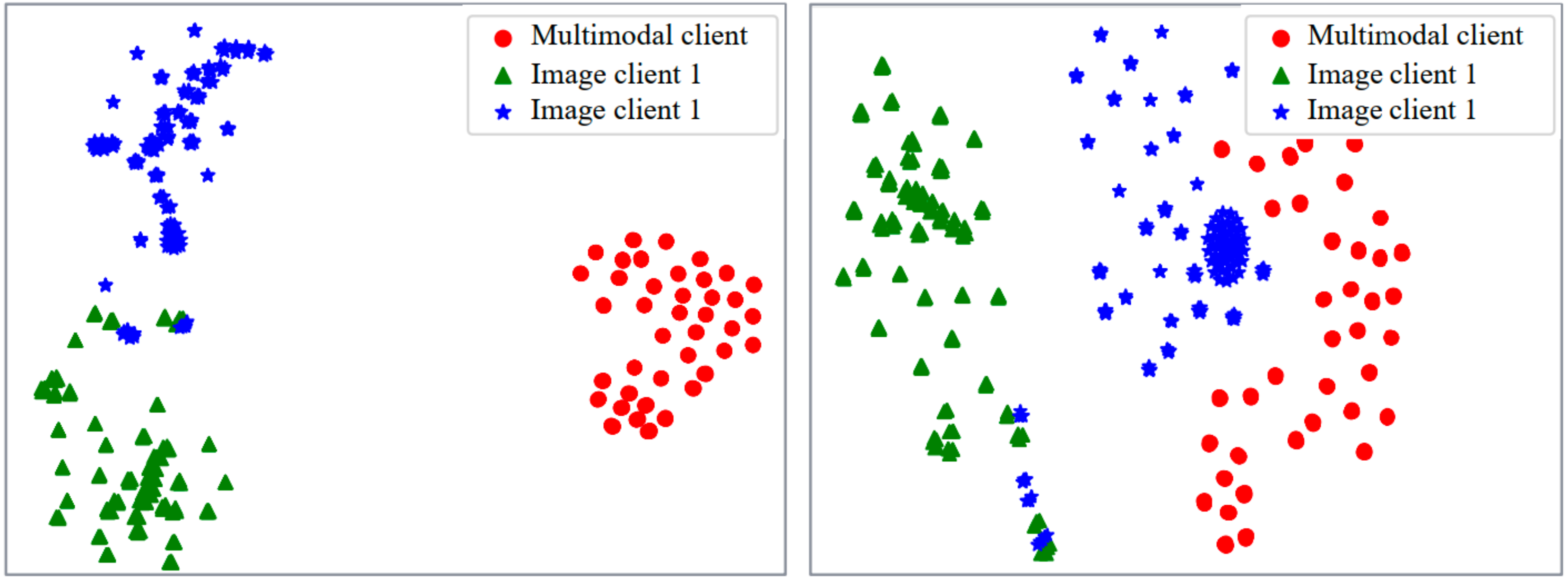
Trade-off between communication and performance:



Qualitative Study of Model Drift

Representations of 250 randomly chosen images from COCO are visualized.

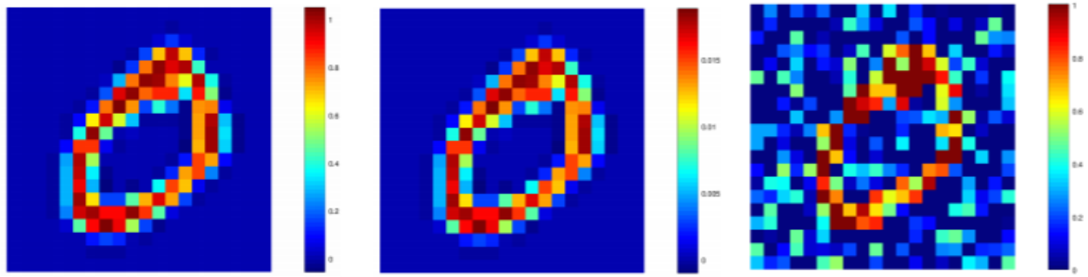
Model drift exists between two modality-identical text clients (blue and green), while this drift is much smaller than the gap between multimodal and uni-modal clients (red v.s. blue+green)



(a) Clients trained under vanilla reamFL+Mean (b) Clients trained under CreamFL

- **Qiying Yu, Yang Liu*, Yimu Wang, Ke Xu, Jingjing Liu***, Multimodal Federated Learning via Contrastive Representation Ensemble (ICLR 2023, code: https://github.com/FLAIR-THU/CreamFL)

Deep Leakage in Model Transfer FL



(a) Original 20x20 image of hand-written number 0, seen as a vector over $\mathbb{R}^{400}$ fed to a neural network.

(b) Recovered image using 400/10285 (3.89%) gradients (see Sect.3, Example 2). The difference with the original (a) is only at the value bar.

(c) Recovered image using 400/10285 (3.89%) gradients (see Sect.3, Example 3). There are noises but the truth label 0 can still be seen.

Fig. 3. Original data (a) vs. leakage information (b), (c) from a small part of gradients in a neural network.

Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. 2018. Privacy-Preserving Deep Learning via Additively Homomorphic Encryption. IEEE Trans. Information Forensics and Security， 13, 5 (2018),1333–1345

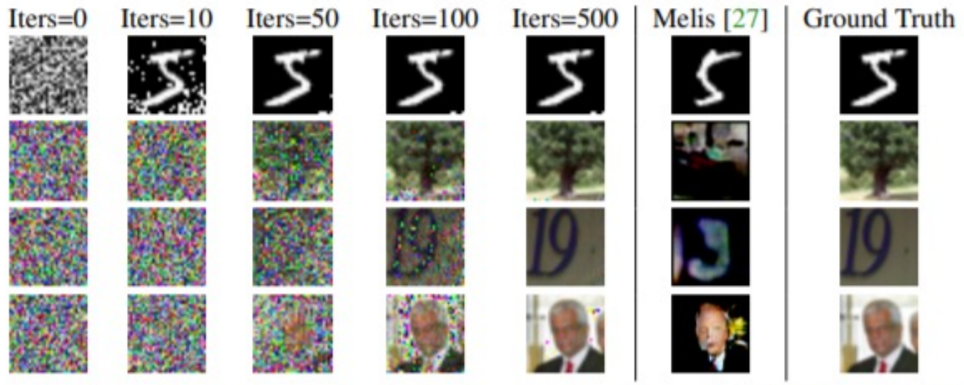


Figure 3: The visualization showing the deep leakage on images from MNIST [22], CIFAR-100 [21], SVHN [28] and LFW [14] respectively. Our algorithm fully recovers the four images while previous work only succeeds on simple images with clean backgrounds.

Ligeng Zhu, Zhijian Liu, Song Han, Deep Leakage from Gradients, Neurips 2019

original | results of independent optimization processes

Hongxu Yin et al , See through Gradients: Image Batch Recovery via GradInversion, CVPR 2021

Will there be *deep leakage from logits* in FedMD-like schemes?

Two necessary principles to attack FedMD

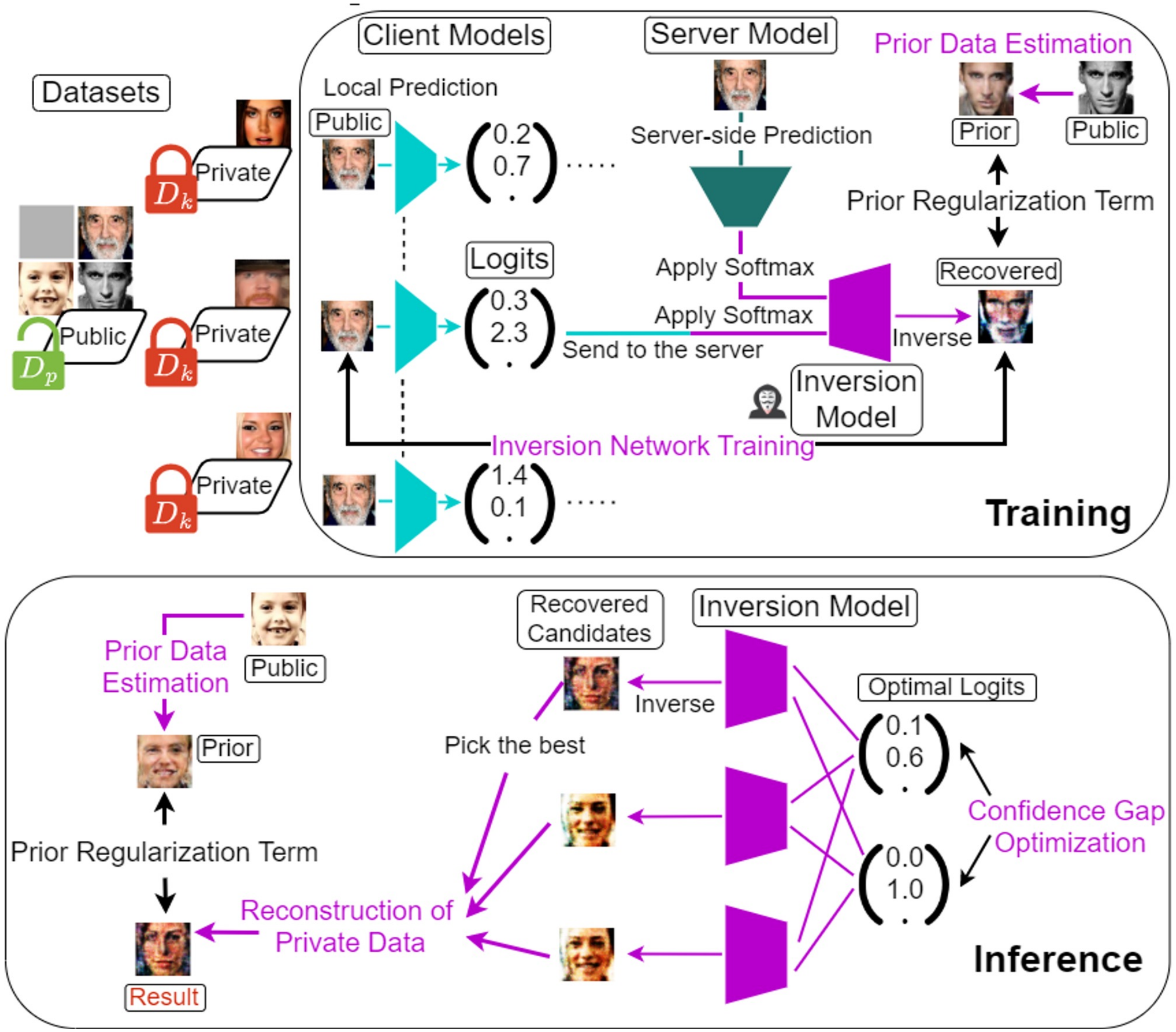
- ***Gradient-free***
- Since gradients are not shared in FedMD, our attack cannot use gradients-related information

- ***Knowledge-decoupling***
- Local models are trained on both private and public datasets,
- Our attack should recover only private data. (In the previous example, we do not want masked face)

None of existing methods meets both principles.

Paired-Logits-Inversion Attack (PLI,CVPR'23)

1. Train an inversion NN on public data
   - Input is the predicted logits of server-side and client-side models on the public data
   - Output is the original public data
2. Estimate output logits of server-side and client-side models on the target private data
3. Feed those estimated logits to the inversion NN to generate original private data
4. We also use prior generated from the public data for regularization



Hideaki Takahashi, Jingjing Liu, and Yang Liu, Breaching FedMD: Image Recovery via Paired-Logits Inversion Attack (CVPR 2023)

PLI - Inversion Neural Network

清華大学智能产业研究院
Institute for AI Industry Research,Tsinghua University

1.Server-side and client-side logits are

$$l_i^0 = f_0(W_0; x_i^0), \quad l_i^k = f_k(W_k; x_i^0)$$

2. Next, we train an inversion neural network G with

$$\min_{\theta} \sum_i ||G_\theta(p_{i,\tau}^0, p_{i,\tau}^k) - x_i^0||_2 + \gamma||G_\theta(p_{i,\tau}^0, p_{i,\tau}^k) - \bar{x}_i||_2$$

, where

$$p_{i,\tau}^0 = \text{softmax}(\boldsymbol{l}_i^0, \tau), \quad p_{i,\tau}^k = \text{softmax}(\boldsymbol{l}_i^k, \tau)$$

- the first term is reconstruction error
- the second term is regularization term

PLI - Estimate output logits of private data using Confidence Gap Optimization

The quality of recovered image is

$$Q(x_j^k) := p^k_{\underline{j},\tau} + p^0_{\underline{j},\tau} + \alpha H(p^0_{j,\tau})$$

Q is maximized with the bellow logits

$$\hat{p}^k_{\underline{u},\tau} = \begin{cases} 1 & (u = j) \\ 0 & (u \neq j) \end{cases}, \quad \hat{p}^0_{\underline{u},\tau} = \begin{cases} \frac{\sqrt[\alpha]{e}}{J-1+\sqrt[\alpha]{e}} & (u = j) \\ \frac{1}{J-1+\sqrt[\alpha]{e}} & (u \neq j) \end{cases}$$

Then, we can estimate the private data with

$$\arg\max_{x_j^k} Q(x_j^k) = G^k_\theta(\hat{p}^0_{j,\tau}, \hat{p}^k_{j,\tau})$$

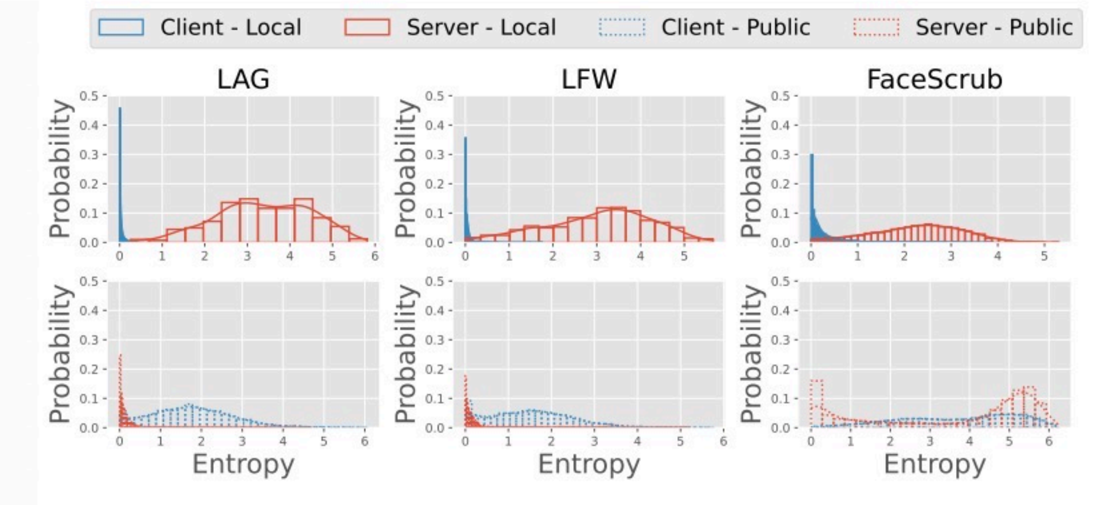


Figure 3. Confidence gap between the server and the client under FedMD setting on public and private data. This figure represents the normalized histogram of entropy on public and local datasets and estimated distribution. Lower entropy means that the model is more confident. Client consistently has higher confidence on private dataset than server, indicating a significant confidence gap.

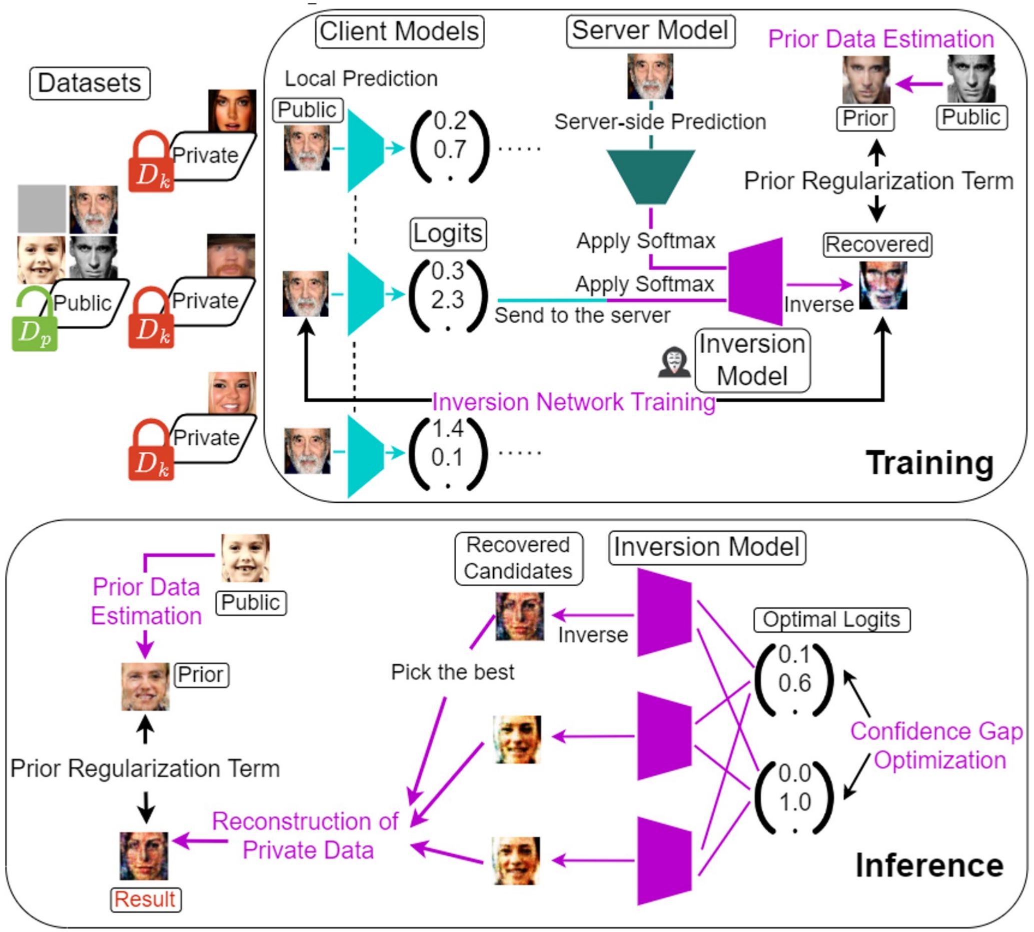
Prior Data Estimation

1. Naive Approach - same prior for all labels

$$\overline{x}_j = \frac{1}{|D_0|} \sum_{i \in D_0} x_i^0$$

1. GAN-based Translation Model - prior per label

$$\overline{x}_j = \frac{1}{|D_a|} \sum_{j \in D_a} A_\phi(x_j^a)$$



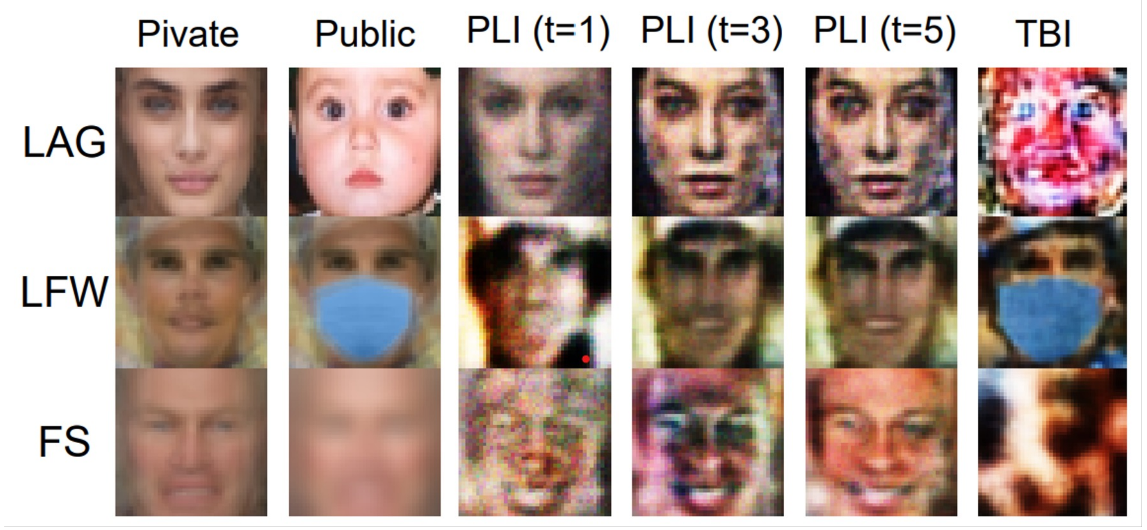
Results

1. The attack is success when SSIM between the reconstructed image and the average private image of the target label exceeds SSIM between any average private/public images of other labels.

2. Our PLI outperforms the prior method in most settings.

Dataset	FaceScrub			LAG			LFW		
Scheme	DS-FL	FedGEMS	FedMD	DS-FL	FedGEMS	FedMD	DS-FL	FedGEMS	FedMD
TBI ($K = 1$)	87.0	1.0	92.5	70.0	0.5	16.5	73.5	2.5	2.0
PLI ($K = 1$)	**91.5**	**29.0**	**94.0**	**71.0**	**17.0**	**60.0**	**99.5**	**91.0**	**99.5**
TBI ($K = 10$)	2.0	0.5	7.0	6.5	0.0	0.0	17.5	9.5	10.0
PLI ($K = 10$)	**62.5**	**20.0**	**74.5**	**15.0**	**26.5**	**63.5**	**15.5**	**71.5**	**79.0**

Table 2. Results on attack accuracy (%).



Hideaki Takahashi, Jingjing Liu, and Yang Liu， Breaching FedMD: Image Recovery via Paired-Logits Inversion Attack (CVPR 2023, code available at https://github.com/FLAIR-THU/PairedLogitsInversion)

t represents the number of communications.

Outline

I. Knowledge Transfer (KT)- Federated Learning (FL)

II. Addressing challenges in KD-based FL

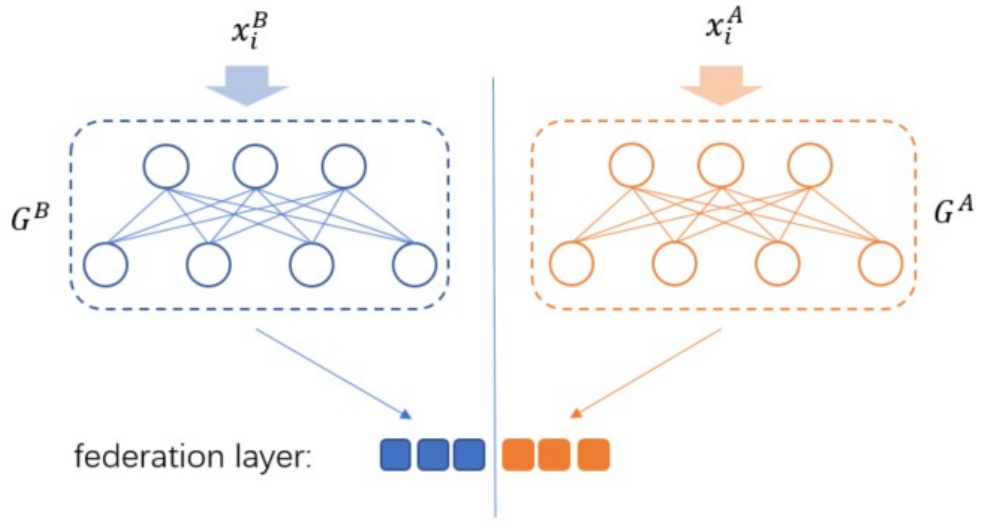
III. Vertical FL

Problem Definition for VFL

- The collaborative training problem is formulated:

$$\min_{\Theta} \mathcal{L}(\Theta;\mathcal{D}) \triangleq \frac{1}{N}\sum_{i=1}^{N} f(\theta_1,\ldots,\theta_K;\mathcal{D}_i) + \lambda\sum_{k=1}^{K}\gamma(\theta_k)$$

$$f(\theta_1,\ldots,\theta_K;\mathcal{D}_i) = f(\sum_{k=1}^{K} G^k(\mathbf{x}_i^k)\theta_k, y_i^K)$$



G^k denotes local feature transformation function that is unknown to other parties

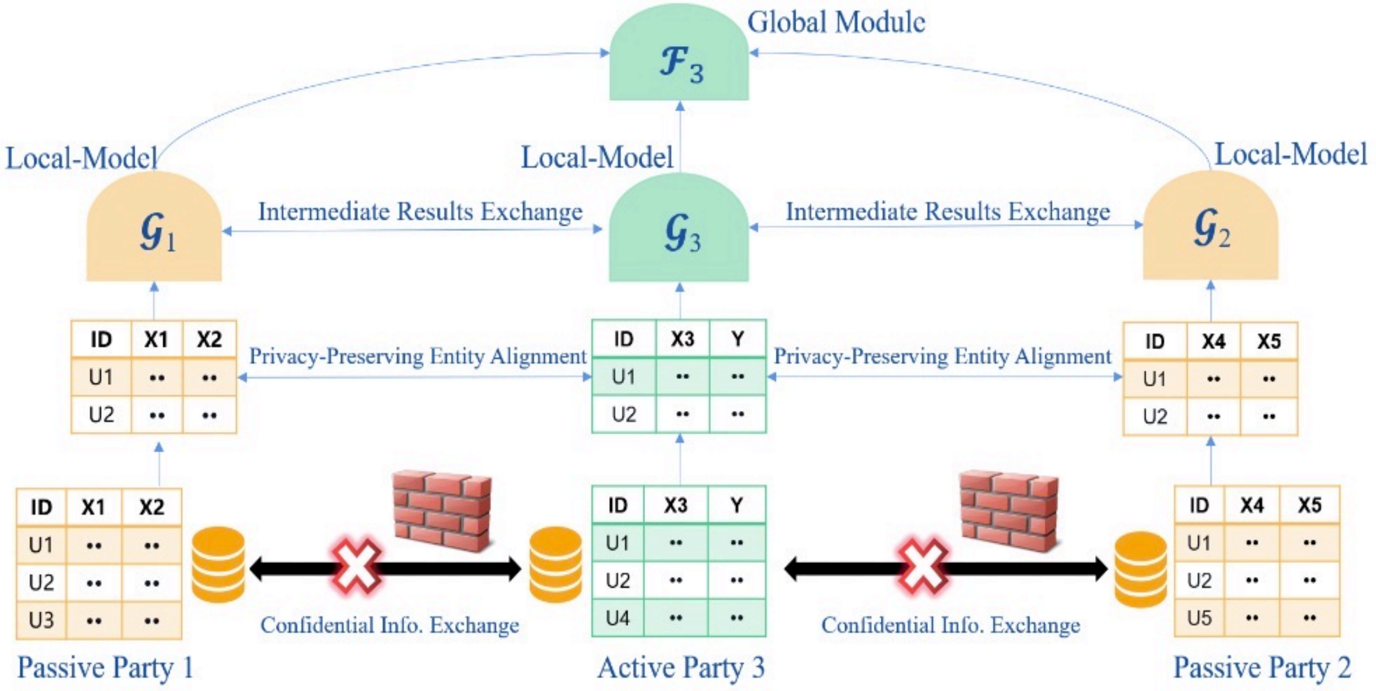
Assumptions:

- Features of the same sample are distributed across *K* parties.
- Samples referring to the same entity **are aligned (by encrypted entity alignment techniques)**
- Each party owns one part of a complete model
- Only one party has the label (the *Kth* party, “**active’ party**)

Constraints:

- Model parameters and data stay local

Training Vertical Federated Learning



$$f(\boldsymbol{\Theta}; \mathbf{x}_i, y_i) = \mathcal{L}\left(\mathcal{F}_K\left(\psi_K; \mathcal{G}_1(\mathbf{x}_{i,1}, \theta_1), \ldots, \mathcal{G}_K(\mathbf{x}_{i,K}, \theta_K)\right), y_{i,K}\right)$$

Algorithm 1 A General VFL Training Procedure.

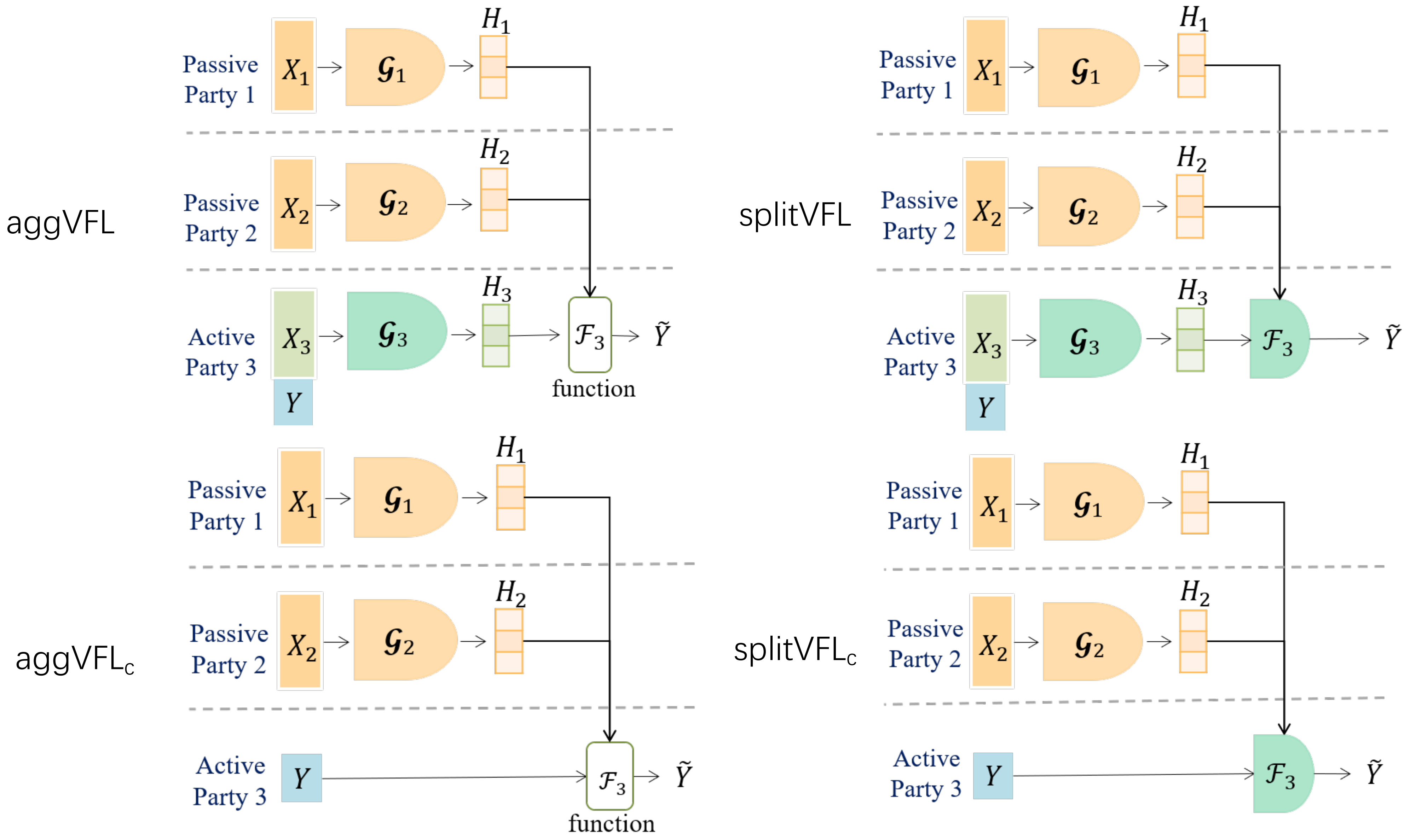
Input: learning rates η_1 and η_2

Output: Model parameters θ_1, θ_2 ... θ_K, ψ_K

```
Party 1,2,...,K, initialize θ1, θ2, ... θK, ψK.
for each iteration j = 1, 2, ... do
    Randomly sample a mini-batch of samples x ⊂ D
    for each party k=1,2,...,K in parallel do
        Party k computes H_k = G_k(x_k, θ_k);
        Party k sends {H_k} to party K;
    end for
    Active party K updates ψ_K^{j+1} = ψ_K^j − η1 ∂ℓ/∂ψ_K;
    Active party K computes and sends ∂ℓ/∂H_k to all other parties;
    for each party k=1,2,...,K in parallel do
        Party k computes ∇_{θ_k} ℓ with Equation (6);
        Party k updates θ_k^{j+1} = θ_k^j − η2 ∇_{θ_k} ℓ;
    end for
end for
```

Yang Liu et al, Vertical Federated Learning, https://arxiv.org/abs/2211.12814

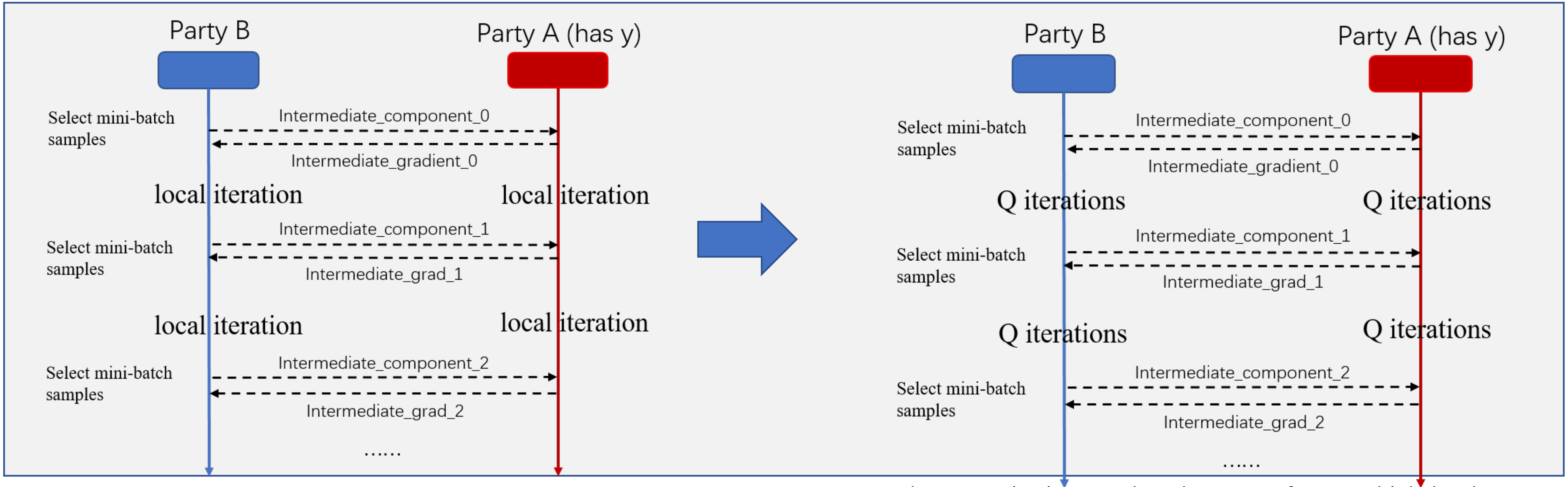
VFL Categorization



FedBCD: A FedAvg-like algorithm for VFL

FedSGD

FedBCD



- Communication at every round
- expensive especially when privacy-preserving protocol is applied.

- Each communication round, each party performs multiple local iterations,
- Each local iteration, each party locally computes gradient based on its own data and （staled) intermediate components from other parties in *the most recent synchronization*.

Y. Liu,. Kang, X. Zhang, L. Li, Y. Cheng, T. Chen, M. Hong, Q. Yang, FedBCD: A Communication-Efficient Collaborative Learning Framework for Distributed Features， IEEE Transaction on Signal Processing， 2022

FedBCD: Main Results

Algo.	MIMIC-LR AUC 84% Q	rounds	MNIST-CNN AUC 99.7% Q	rounds
FedSGD	1	334	1	46
FedBCD	5	71	3	16
	50	52	5	8
FedBCD-s	1	407	1	48
	5	74	3	15
	50	52	5	9

TABLE I

NUMBER OF COMMUNICATION ROUNDS TO REACH A TARGET AUC FOR FEDBCD-P, FEDBCD-S AND FEDSGD ON MIMIC-LR AND MNIST-CNN RESPECTIVELY.

Credit-FTL						
AUC	Algo.	Q	R	comp.	comm.	total
70%	FedSGD	1	17	11.33	11.34	22.67
	FedBCD	5	4	13.40	2.94	16.34
		10	2	10.87	2.74	13.61
75%	FedSGD	1	30	20.50	20.10	40.60
	FedBCD	5	8	26.78	5.57	32.35
		10	4	23.73	2.93	26.66
80%	FedSGD	1	46	32.20	30.69	62.89
	FedBCD	5	13	43.52	9.05	52.57
		10	7	41.53	5.12	46.65

TABLE II

NUMBER OF COMMUNICATION ROUNDS, COMPUTATION, COMMUNICATION AND TOTAL TRAINING TIME (MINS) TO REACH TARGET AUC FOR FEDSGD VERSUS FEDBCD-P.

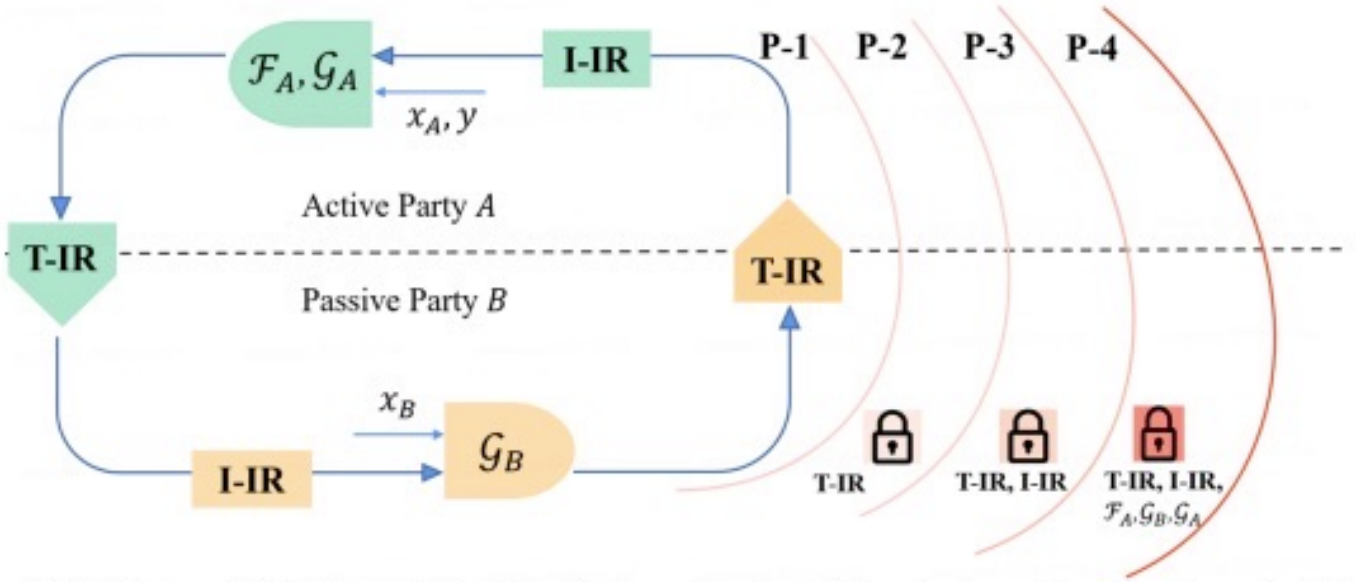
- The number of communication rounds required to reach ϵ

$$\frac{T}{Q} = \mathcal{O}\left(\frac{K^{1/2}}{S^{1/2}\epsilon^{3/2}}\right).$$

It is the first time that such rates have been proven for any algorithms with multiple local steps designed for the feature-partitioned federated learning problem

Compare with vanilla BCD, FedBCD saves communication by having multiple local updates

Security Protocols of VFL



T-IR: Transmitted Intermediate Results (e.g., local model outputs and backward gradients)

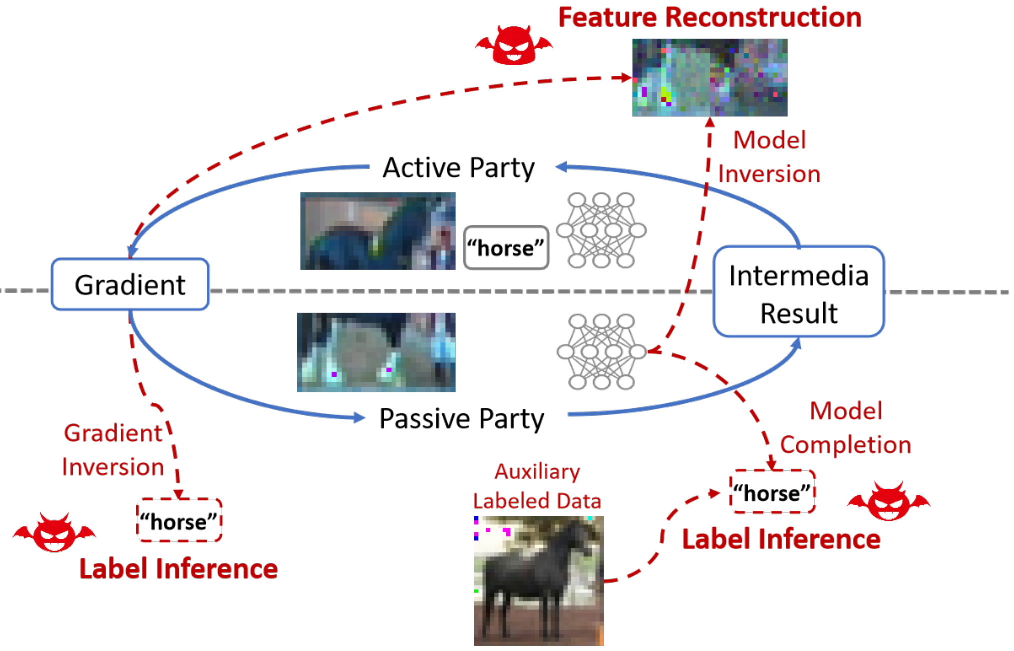
I-IR: Internal Intermediate Results (e.g., local trainable model parameters/gradients)

- Basic Protocol (P-1): Keeping Private data and models local.
- Standard Protocol (P-2): Protecting Exchanged Intermediate Results
- Enhanced Protocol (P-3): Protecting Entire Training Protocol
- Strict Protocol (P-4): Protecting Training Protocol and Results
- Relaxed Protocol (P-0): Nonprivate label or model.

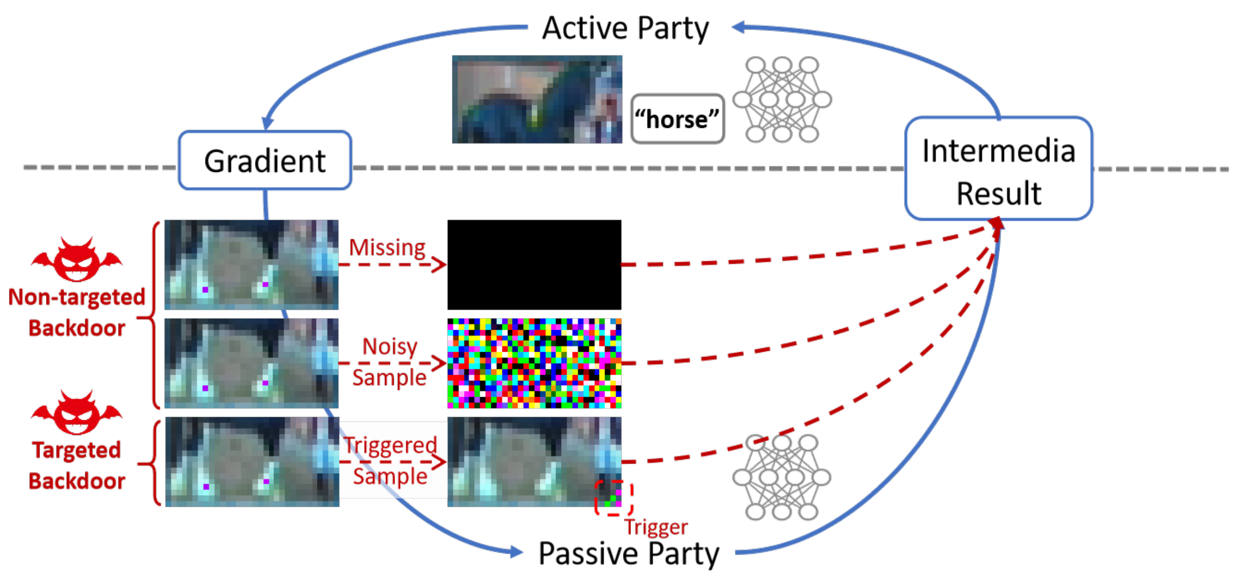
Yang Liu et al, Vertical Federated Learning, https://arxiv.org/abs/2211.12814

Attacks and Defenses

Data Reconstruction Attacks



Backdoor Attacks



Summary of Attacks

	Attacking Method	VFL Setting	Model	Against Protocol	Attacking Phase	Auxiliary Requirement
Label Inference Attack	Direct Label Inference (DLI) [19], [108]	aggVFL	NN	P-1	Training	–
	Norm Scoring (NS) [109]	splitVFL$_c$	NN	P-1	Training	–
	Direction Scoring (DS) [109]	splitVFL$_c$	NN	P-1	Training	–
	Residual Reconstruction (RR) [110]	aggVFL	LR	P-2	Training	–
	Gradient Inversion (GI) [108]	aggVFL	NN	P-2	Training	–
	Gradient Inversion (GI) [111]	splitVFL$_c$	NN	P-2	Training	Label Prior Distribution
	Passive Model Completion (PMC) [19]	splitVFL	NN	P-3	Inference	Labeled Data
	Active Model Completion (AMC) [19]	splitVFL	NN	P-3	Inference	Labeled Data
Feature Inference Attack	Binary Feature Inference Attack (BFIA) [112]	splitVFL	NN	P-1	Training	Binary Features
	Reverse Multiplication Attack (RMA) [113]	aggVFL	LR	P-2	Training	Corrupted Coordinator
	Protocol-aware Active Attack(PAA) [114]	aggVFL	LR	P-2	Training	–
	Reverse Sum Attack (RSA) [113]	aggVFL	GBDT	P-2	Training	–
	Equality Solving Attack (ESA) [100]	aggVFL	LR	P-0(g)	Inference	–
	Path Restriction Attack (PRA) [100]	aggVFL	Tree	P-0(g)	Inference	–
	Generative Regression Network (GRN) [100]	aggVFL	NN	P-0(g)	Inference	–
	White-Box Model Inversion (MI) [101], [102]	aggVFL & splitVFL$_c$	LR & NN	P-0(g)	Inference	–
	Black-box Model Inversion (MI) [101], [102]	aggVFL & splitVFL$_c$	LR & NN	P-1	Inference	Labeled Data
	Catastrophic Data Leakage in VFL (CAFE) [23]	aggVFL$_c$	NN	P-0(g)	Training	–

	Attacking Method	VFL Setting	Against Protocol	# of Classes	Attacking Phase	Auxiliary Requirement
Targeted Backdor Attack	Label Replacement Backdoor by replacing gradients (LRB) [138]	aggVFL	P-2	≥ 2	Training	At least one label of clean samples
	Adversarial Dominant Input attack (ADI) [139]	VLR/splitVFL$_c$	P-0(g)/P-1	≥ 2	Inference	a few samples from the other party
Non-targeted Backdoor Attack	Adversarial attack [24], [140]	splitVFL/aggVFL	P-1	≥ 2	Training	–
	Missing attack [24]	splitVFL/aggVFL	P-3	≥ 2	Training	–

Yang Liu et al, Vertical Federated Learning, https://arxiv.org/abs/2211.12814

Available online at

Summary of Defenses

清華大学智能产业研究院
Institute for AI Industry Research,Tsinghua University

Cryptographic Defense

Defense Work	VFL Setting	Model	Defense Scheme	Protocol	Party	Require Coordinator	Adversarial Assumption
GasconLR [17]	aggVFL	LR	GC+SS	P-3	≥ 2	✓	SH
HardyLR [94]	aggVFL	LR	HE	P-2	≥ 2	✓	SH
BaiduLR [107]	aggVFL	LR	HE	P-2	≥ 2	✗	SH
SecureLR [108]	aggVFL	LR	HE+SS	P-2	≥ 2	✗	SH
CAESAR [19]	aggVFL	LR	HE+SS	P-3	$= 2$	✗	SH
HeteroLR [97]	aggVFL	LR	HE+SS	a :P-3, p :P-4	$= 2$	✗	SH
FedV [21]	aggVFL	LR/SVM	FE	P-2	≥ 2	✓	SH
SecureBoost [15]	aggVFL	XGB	HE	P-2	≥ 2	✗	SH
SecureBoost+ [33]	aggVFL	XGB	HE	P-2	≥ 2	✗	SH
SecureXGB [35]	aggVFL	XGB	HE+SS	P-3	$= 2$	✗	SH
MP-FedXGB [38]	aggVFL	XGB	SS	P-3	≥ 2	✓	SH
SecureGBM [34]	aggVFL	LGBM	HE	P-2	$= 2$	✗	SH
Pivot [39]	aggVFL	RF / GBDT	HE+SS	P-3	≥ 2	✗	SH, $\leq K$-1 colluded parties
Enhanced Pivot [39]	aggVFL	DT	HE+SS	P-4	≥ 2	✗	SH, $\leq K$-1 colluded parties
FedSGC [109]	aggVFL$_c$	GNN	HE	P-2	$= 2$	✗	SH
ACML [110]	splitVFL$_c$	NN	HE	P-1	$= 2$	✗	SH
PrADA [79]	splitVFL	NN	HE	P-1	≥ 2	✗	SH
BlindFL [96]	splitVFL	NN	HE+SS	a :P-2, p :P-4	$= 2$	✗	SH
SFTL [77]	aggVFL	NN	HE	P-2	$= 2$	✗	SH
SFTL [77]	aggVFL	NN	SS	P-3	$= 2$	✗	SH
SEFTL [78]	aggVFL	NN	HE+SPDZ	P-3	$= 2$	✗	MA,dishonest majority
N-TEE [111]	aggVFL	XGB	TEE	P-3	≥ 2	✗	SH

Emerging Defense

	Defense Work	VFL Setting	Model	Defense Scheme	Against Attack	Defending Party
Defenses against Label Inference Attack	MARVELL[98]	splitVFL$_c$	NN	Add Noise	NS, DS	Active party
	Max-Norm[98]	splitVFL$_c$	NN	Add Noise	NS, DS	Active party
	CAE [30]	aggVFL	NN	HE+Disguise Label	DLI, MC	Active party
	DCAE [30]	aggVFL	NN	HE+Disguise Label+DG	DLI, MC	Active party
	PELoss [113]	splitVFL$_c$	NN	Potential Energy Loss	MC	Active party
	dCorr [101]	splitVFL$_c$	NN	Minimize Correlation	SA	Active party
	RM [114]	aggVFL	LR	HE+Random Mask	RR	Active party
Defenses against Feature Inference Attack	FG [28]	splitVFL	NN	Random Fake Gradients	CAFE	Passive party
	DRAVL [115]	splitVFL$_c$	NN	Adversarial Training	MI	Passive party
	MD [102]	splitVFL	NN	Masquerade	BFIA	Passive party
	DP-Paillier-MGD [104]	aggVFL	LR	HE+DP	PAA	Passive party

Table 9: Summary of defense strategies for defending against backdoor attacks.

Defense Work	VFL Setting	Defense Scheme	Against Attack
DP[30]	aggVFL	Add Noise	Targeted
GS [30]	aggVFL	Sparsify Gradient	Targeted
CAE [30]	aggVFL	HE+Disguise Label	Targeted
DCAE [30]	aggVFL	HE+Disguise Label+DG	Targeted
RVFR [29]	splitVFL	Robust Feature Sub-space Recovery	Targeted/Non-targeted

Available online at

Yang Liu et al, Vertical Federated Learning, https://arxiv.org/abs/2211.12814

Applications

Major Applications

- **Recommendation systems and Advertising**
- **Finance**
- **Healthcare**
- **Wireless Communication**

Open-Source Projects

- **FATE**
- **PyVertical**
- **FedLearner**
- **FedML**
- **Fedtree**
- **PaddleFL**

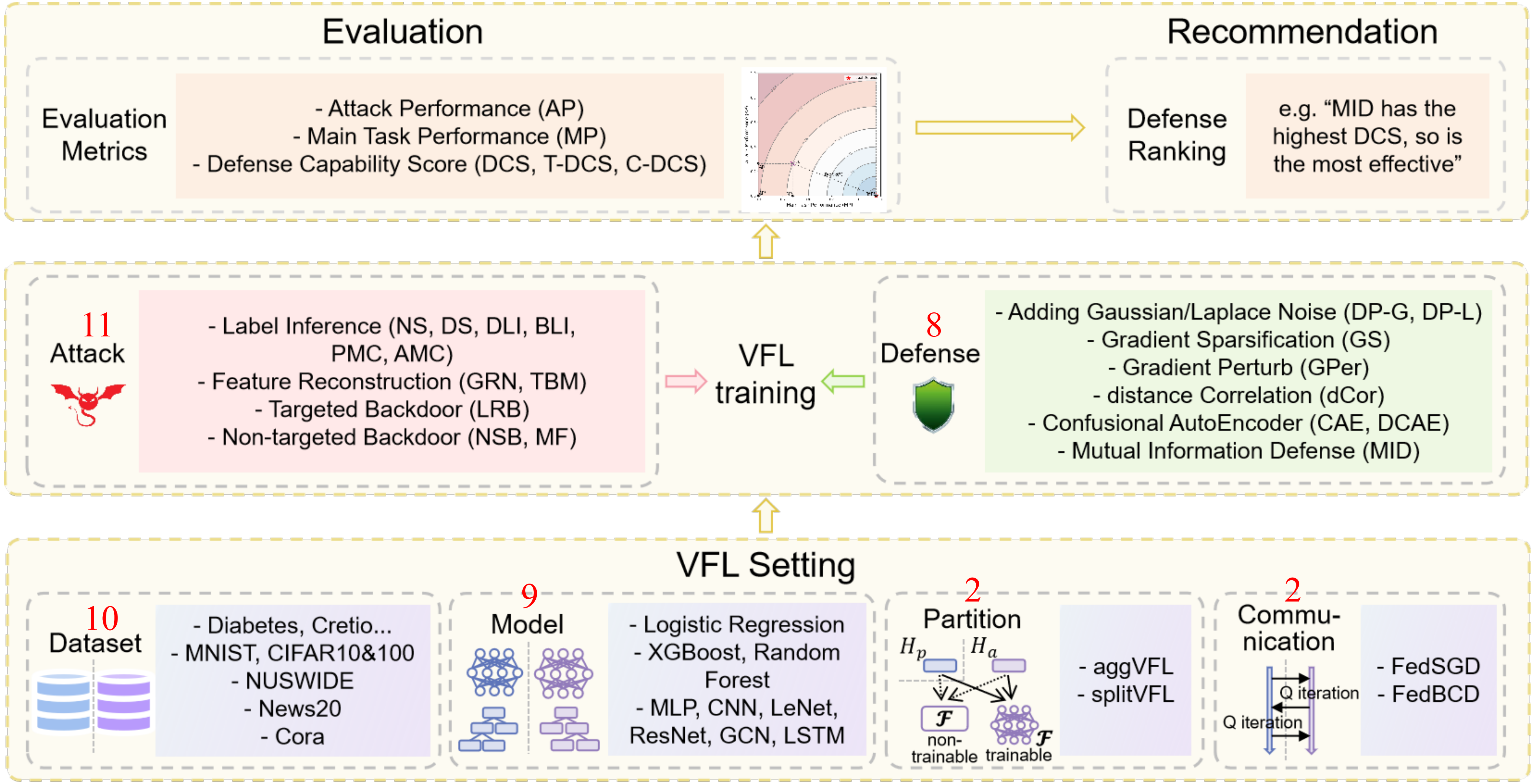
……

Challenges

- A substantial gap between the defense goal of VFL research and practice.
    - Research: achieving state-of-the-art performance on a targeted attack type.
    - Practice: effective yet simple defense solutions to thwart all possible attacks.
- Lack a light-weight and unified VFL framework designed for rapid testing new attack and defense algorithms

VFLAIR

GitHub Link: https://github.com/FLAIR-THU/VFLAIR



Evaluation Module

- **Defense *Depth***

1) Attack Performance (AP), Main Task Performance (MP)

- ideal Attack Performance (AP*), ideal Main Task Performance (MP*)

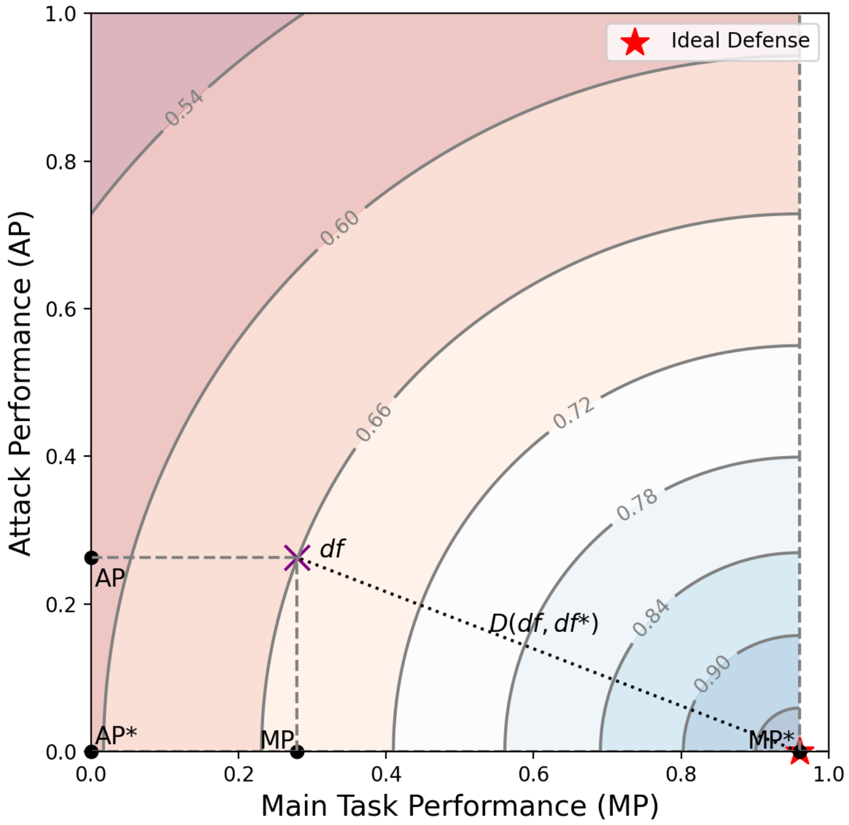
2) Defense Capability Score (DCS)

$$\mathrm{DCS} = \frac{1}{1 + D(df, df^*)} = \frac{1}{1 + \sqrt{(1-\beta)(\mathrm{AP} - \mathrm{AP}^*)^2 + \beta(\mathrm{MP} - \mathrm{MP}^*)^2}}$$

- **Defense *Breadth***

3) Type-level Defense Capability Score (T-DCS)

4) Comprehensive Defense Capability Score (C-DCS)

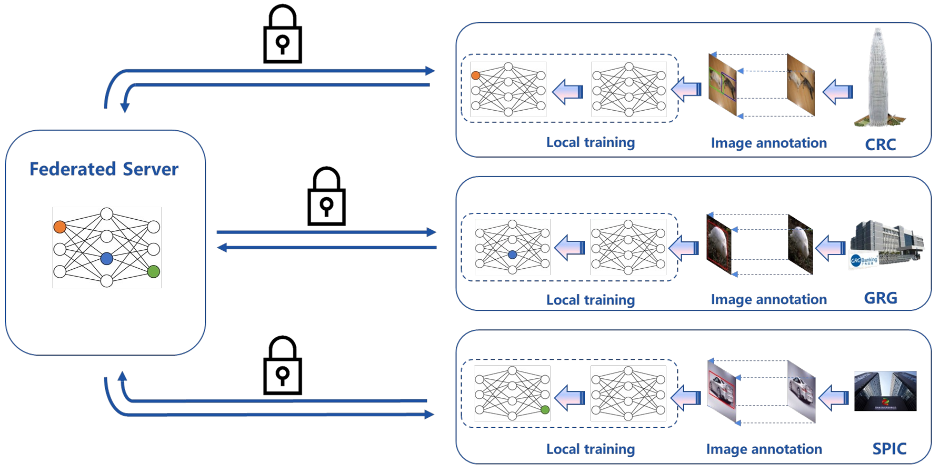


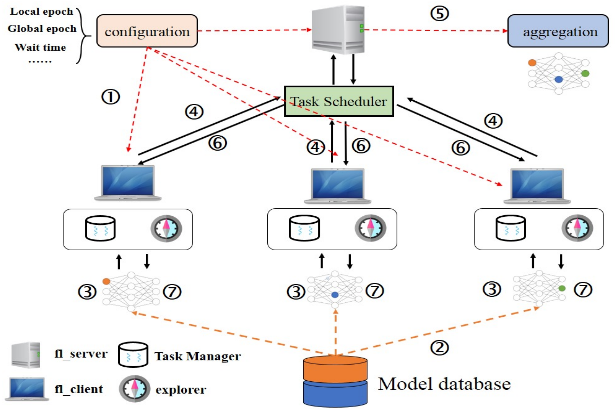
GitHub Link: https://github.com/FLAIR-THU/VFLAIR

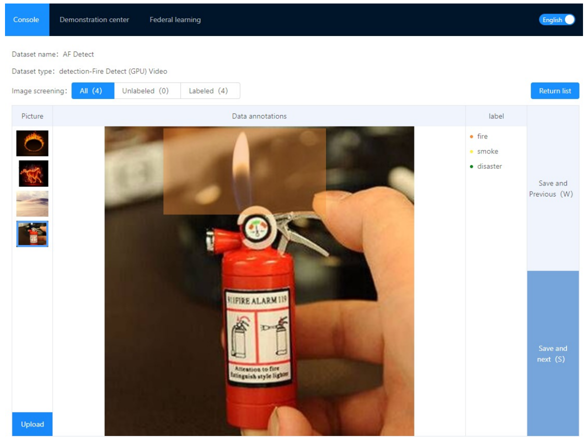
THANKS

FedVision

An online visual object detection platform powered by federated learning







Advantages:

- privacy
- Efficiency improved by ~ 200 times
- reducing labor cost by 60%