WeBank

CCF-TF 14

联邦学习的研究与应用

刘洋 范涛
微众银行高级研究员

yangliu@webank.com， dylanfan@webank.com

FedAI Ecosystem https://www.fedai.org/

2019.03

提纲

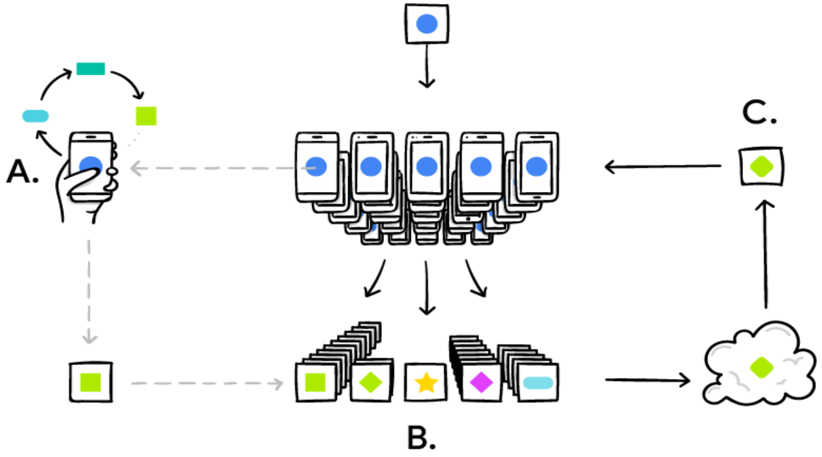
- 联邦学习 Federated Machine Learning
- 联邦迁移学习 Federated Transfer Learning
- 联邦学习的应用案例
- Federated AI Technology Enabler (FATE) 开源项目详解

01

Federated Learning

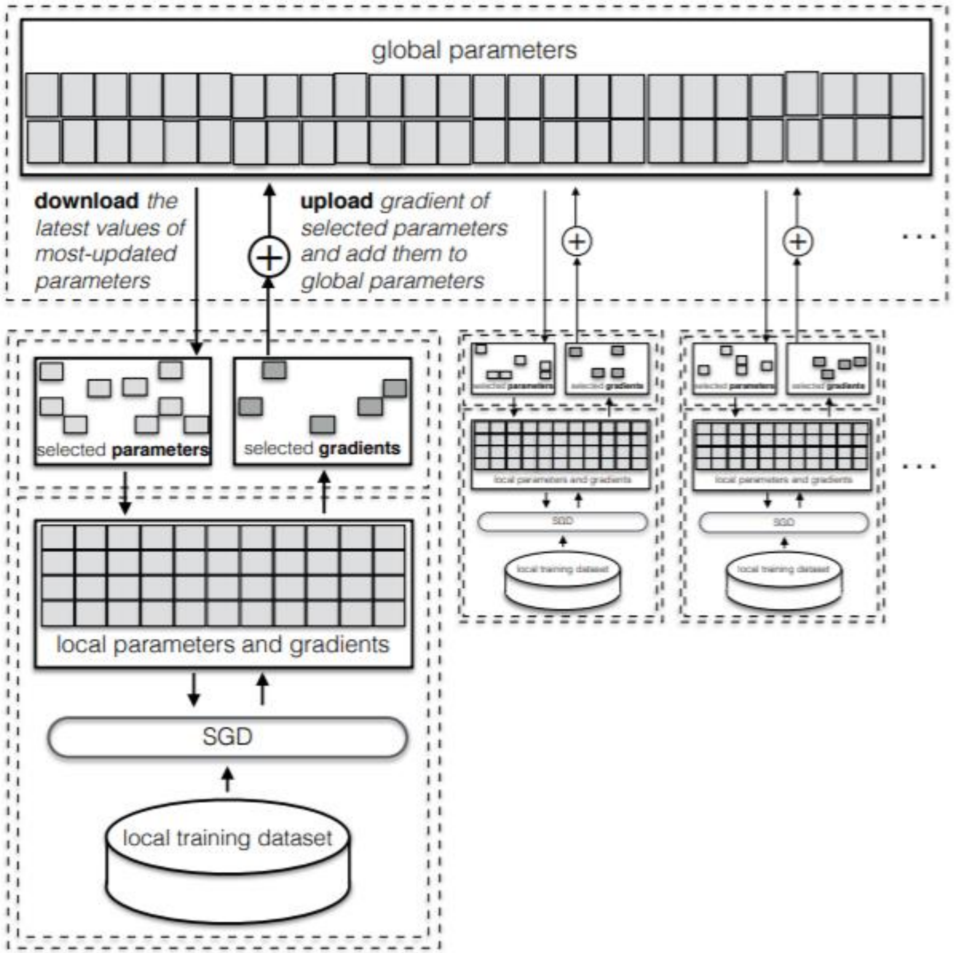
大规模用户在保护数据隐私下的协同学习

联邦学习（Federated Machine Learning）



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

- 手机终端，多个用户，1个中心
- 所有数据特征维度相同
- 本地训练
- 选择用户训练


global parameters
download the latest values of most-updated parameters
upload gradient of selected parameters and add them to global parameters
...
selected parameters
selected gradients
local parameters and gradients
SGD
local training dataset
...

Reza Shokri and Vitaly Shmatikov. 2015. *Privacy-Preserving Deep Learning*. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15). ACM, New York, NY, USA, 1310–1321.

- 选择参数更新

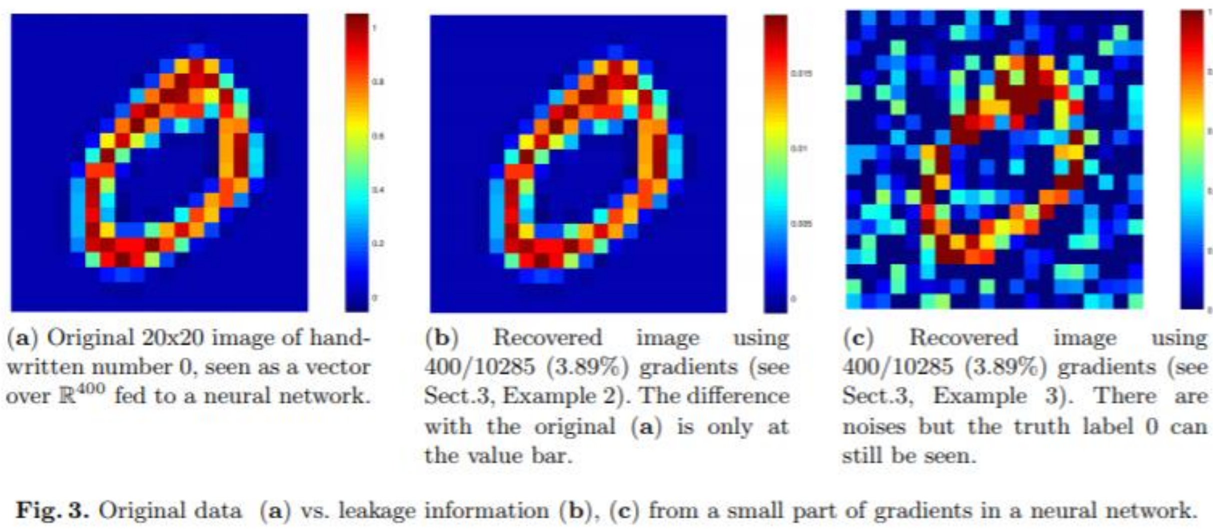
联邦学习（Federated Machine Learning）的研究进展

- **系统效率**
  - 模型压缩 Compression [KMY16]
  - 算法优化 Optimization algorithms [KMR16]
  - 参与方选取 Client selection [NY18]
  - 边缘计算 Resource constraint, IoT, Edge computing [WTS18]

- **模型效果**
  - 数据分布不均匀 Data distribution and selection [ZLL18]
  - 个性化 Personalization [SCS18]

- **数据安全**

梯度是否泄露信息?



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

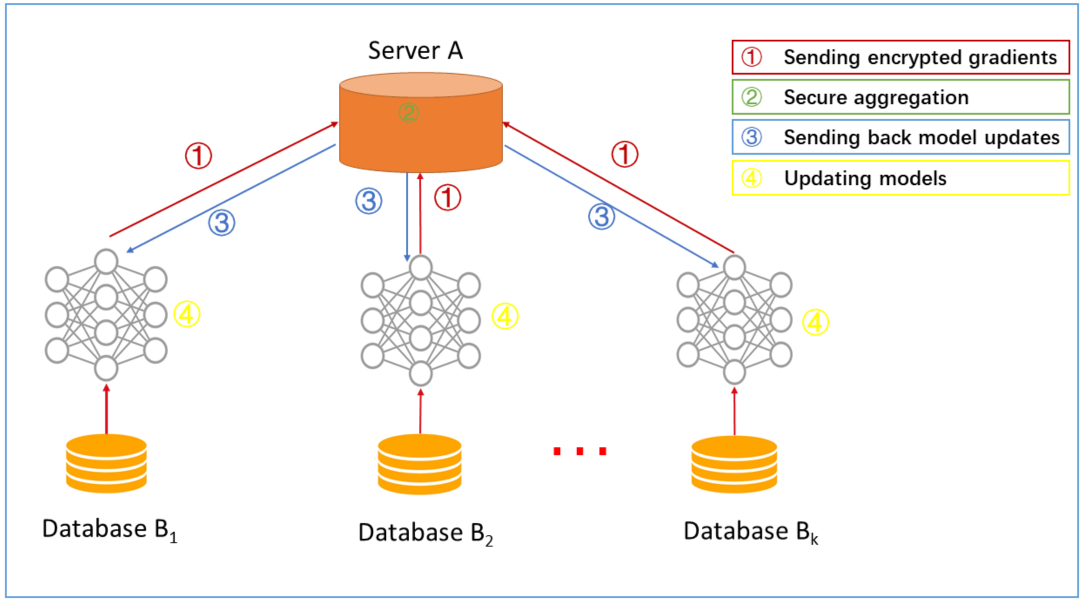
(b) Recovered image using 400/10285 (3.89%) gradients (see Sect.3, Example 2). The difference with the original (a) is only at the value bar.

(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

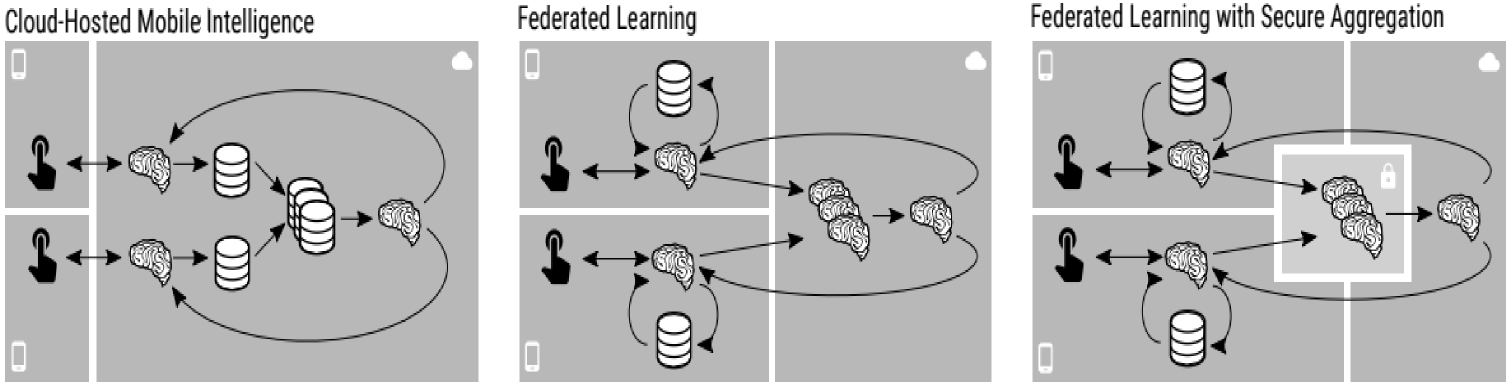
同态加密保护梯度更新



* 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

Secure Aggregation [BIK+17]

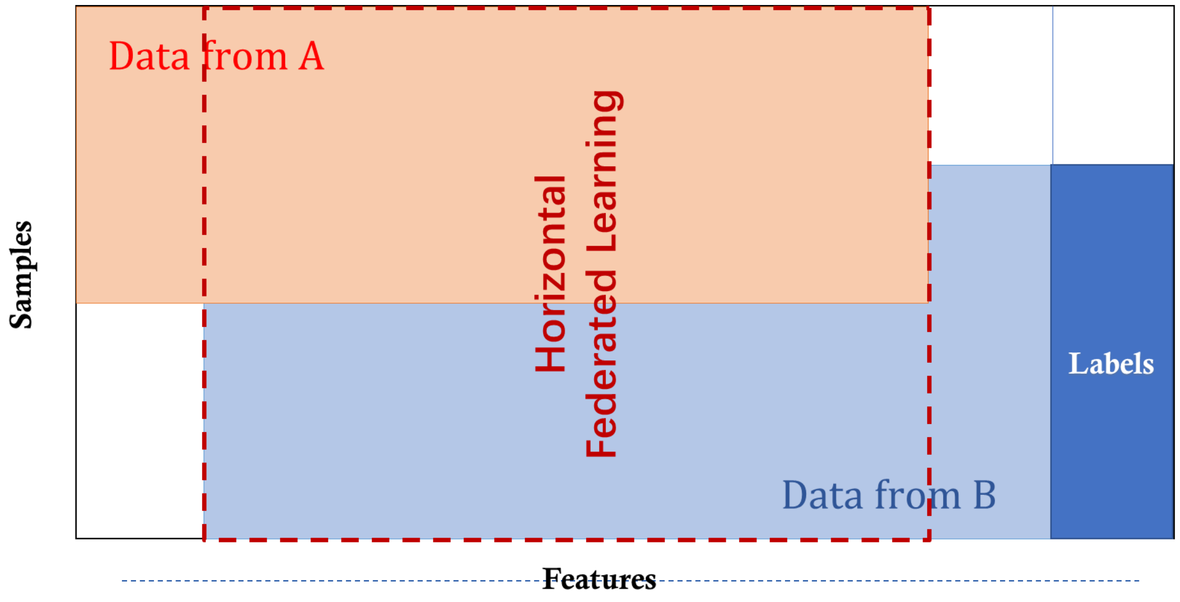
- 本地训练 Local training
- 秘密共享 Secret share aggregated update
- 稳定性 Robust to vanishing clients
- 无个人梯度信息泄露 Individual gradient not disclosed
- 半诚实假设 honest-but-curious setting, server does not collude with users


Cloud-Hosted Mobile Intelligence
Federated Learning
Federated Learning with Secure Aggregation

[BIK+17] Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., ... & Seth, K. (2017, October). Practical secure aggregation for privacy-preserving machine learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (pp. 1175-1191). ACM.

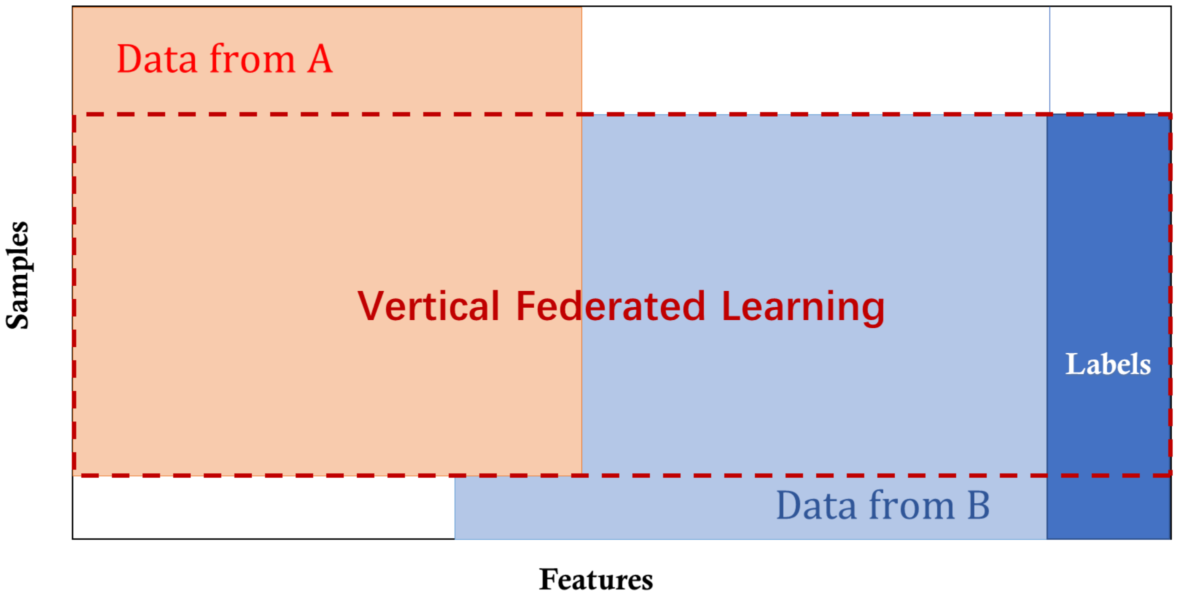
联邦学习分类 Categorization of Federated Machine Learning

横向联邦 Horizontal FML



- 数据方特征维度相同

纵向联邦 Vertical FML



- 数据方样本ID相同

WeBank

纵向联邦学习 Vertical Federated Learning

目标:

- ➢ A方 和 B 方 联合建立模型

假设：

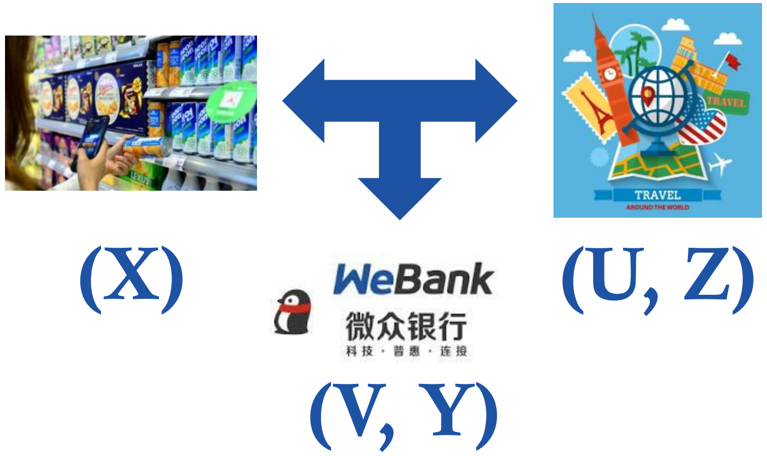
- ➢ 只有一方有标签 Y
- ➢ 两方均不暴露数据

挑战:

- ➢ 只有X的一方没有办法建立模型
- ➢ 双方不能交换共享数据

预期：

- ➢ 双方均获得数据保护
- ➢ 模型无损失（ LOSSLESS ）


(X)

WeBank 微众银行 科技 · 普惠 · 连接

(U, Z)

(V, Y)

ID	X1	X2	X3
U1	9	80	600
U2	4	50	550
U3	2	35	520
U4	10	100	600
U5	5	75	600
U6	5	75	520
U7	8	80	600

Retail A Data

ID	X4	X5	Y
U1	6000	600	No
U2	5500	500	Yes
U3	7200	500	Yes
U4	6000	600	No
U8	6000	600	No
U9	4520	500	Yes
U10	6000	600	No

Bank B Data

保护隐私的机器学习Privacy-Preserving Machine Learning

从安全定义开始…

半诚实
Honest-but-curious

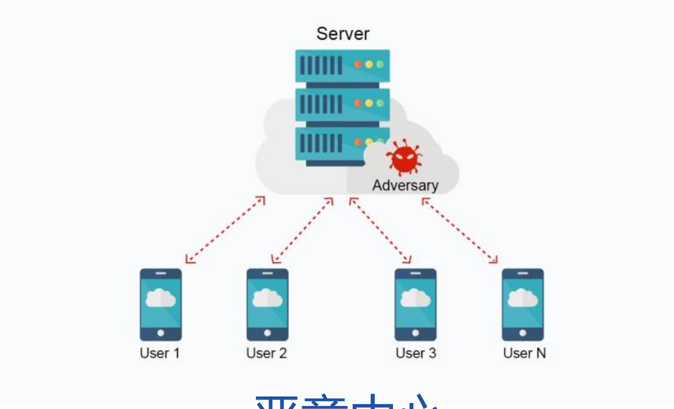
VS

恶意
Malicious

零知识
Zero knowledge

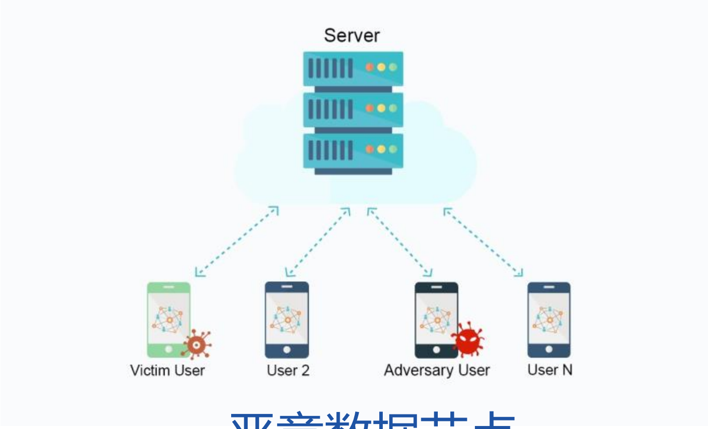
VS

一些知识
Some knowledge



恶意中心
Adversarial Server

VS



恶意数据节点
Adversarial Client

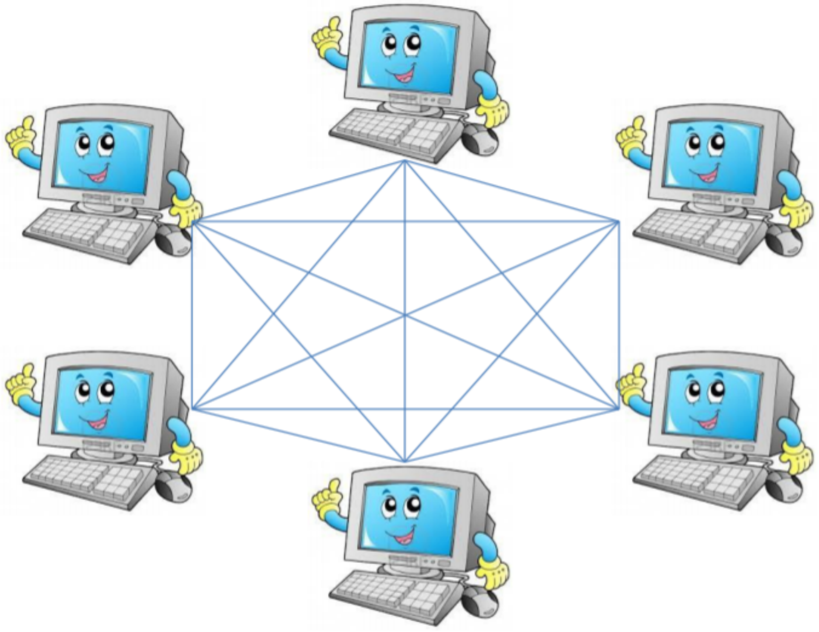
WeBank

隐私保护下的技术工具

- 多方安全计算 Secure Multi-party Computation (MPC)
- 同态加密 Homomorphic Encryption (HE)
- 姚式混淆电路 Yao’s Garbled Circuit
- 秘密共享 Secret Sharing
- 差分隐私 Differential Privacy (DP)

……

多方安全计算 (MPC)



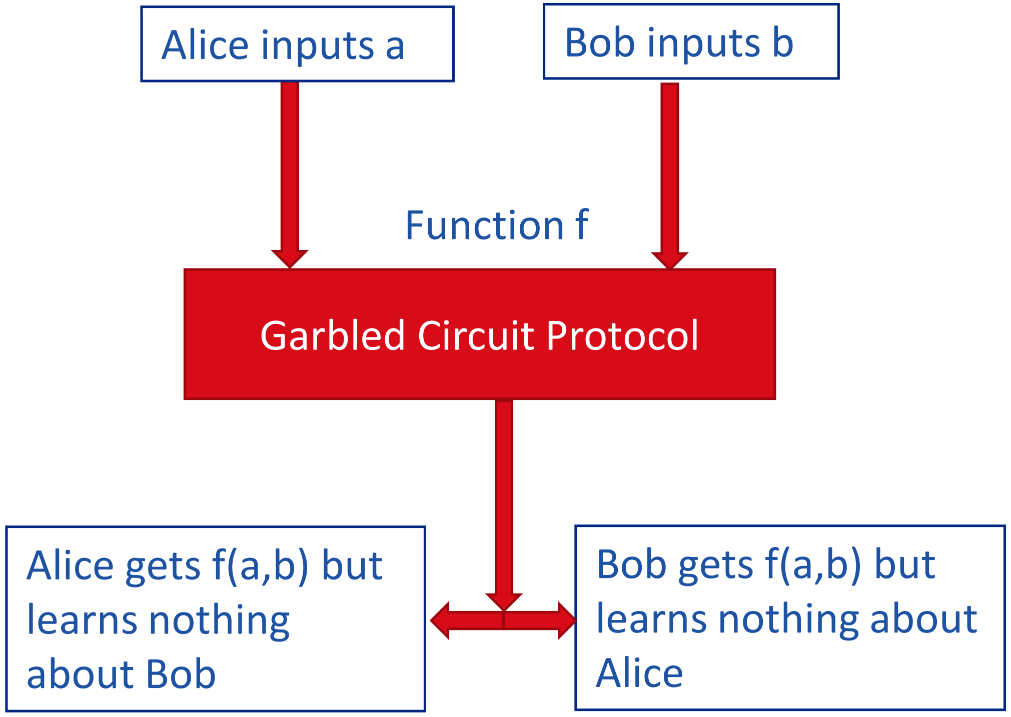
Ran Cohen ,Tel Aviv University, Secure Multiparty Computation: Introduction

- 保证信息层面的数据安全
- 零知识证明 zero knowledge
- 需要多方参与

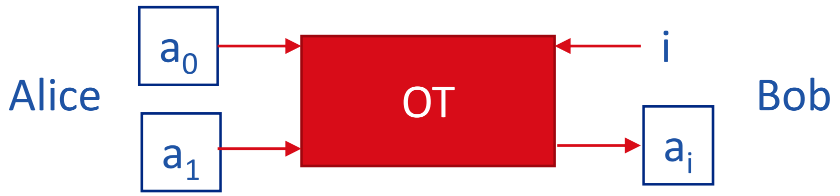
- 缺点：
  - 大量信息传输
  - 降低数据安全要求可以提高效率

WeBank

混淆电路 Yao’s Garbled Circuit Protocol (Andrew Yao, 1980s)


Alice inputs a
Bob inputs b
Function f
Garbled Circuit Protocol
Alice gets f(a,b) but learns nothing about Bob
Bob gets f(a,b) but learns nothing about Alice

- Oblivious Transfer

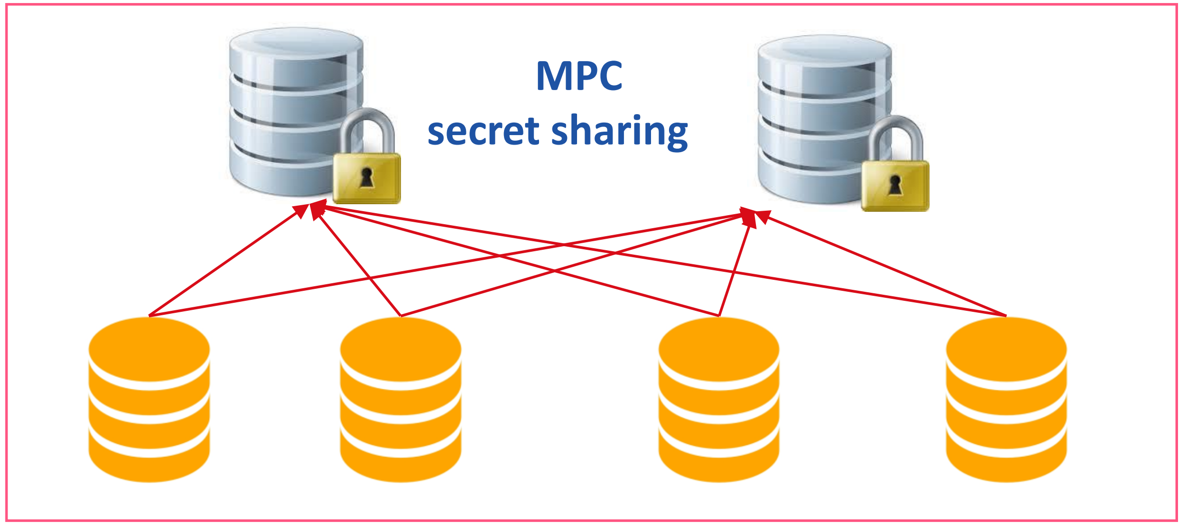


Steps

- Alice builds a garbled circuits;
- Alice sends her input keys;
- Alice and Bob perform Oblivious Transfer;
- Bob gets the output and sends back to Alice;
- Alice and Bob learns nothing about each other’s value.

SecureML [MZ17]

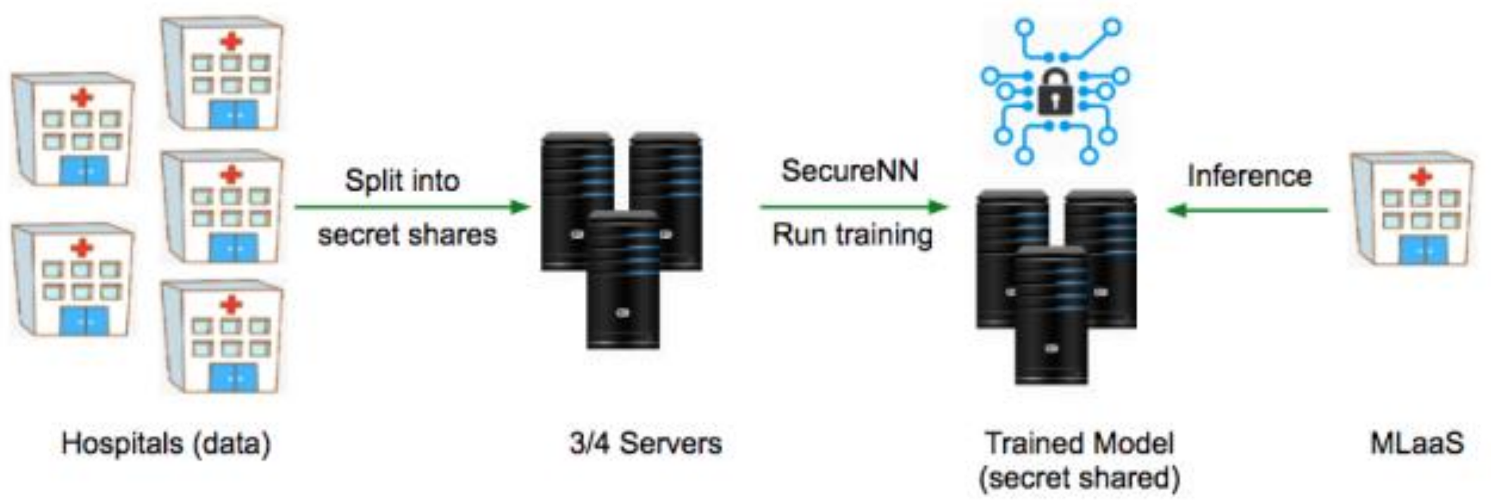
- 隐私保护下的机器学习 Privacy-preserving machine learning for linear regression, logistic regression and neural network training
- 秘密共享，混淆电路，不经意传输 Combine secret sharing, garbled circuits and oblivious transfer
- 需要两方计算 Learn via two untrusted, but non-colluding servers
- Computationally secure, but expensive



[MZ17] Mohassel, P., & Zhang, Y. (2017). SecureML: A system for scalable privacy-preserving machine learning. In *Proceedings of the 38th IEEE Symposium on Security and Privacy (SP)* (pp. 19-38). IEEE.

SecureNN [WGC18]

- 3-party or 4-party machine learning
- Protocols for training and prediction of CNNs, DNNs and other NNs
- Non-linear: private three-party comparison
- Information-theoretically secure



[WGC18] S. Wagh, D. Gupta, and N. Chandran, “Securenn: Efficient and private neural network training,” 2018, iACR ePrint Archive, https://eprint.iacr.org/2018/442.

同态加密 Homomorphic Encryption

- 全同态或者半同态 Full Homomorphic Encryption and Partial Homomorphic Encryption
- 数据层面的信息保护 Data-level information protection

Paillier 半同态加密 Partially homomorphic encryption

***Addition* :** $[[u]] + [[v]] = [[u+v]]$

Scalar multiplication: $n[[u]] = [[nu]]$

- For public key ***pk = n***, *the encoded form of* m ∈ {0, . . . , n − 1} is

$$\mathbf{Encode(m) = r^{n}\ (1 + n)^{\ m} \bmod n^{2}}$$

r is random selected from {0, . . . , n − 1}.

- For float q = (s, e) , encrypt [[q]] = ([[s]], e) , here $q = s\beta^{e}$, is base-β exponential representation.

Rivest, R. L.; Adleman, L.; and Dertouzos, M. L. 1978. On data banks and privacy homomorphisms. Foundations of Secure Computation, Academia Press, 169–179.

同态加密在机器学习上应用 Apply HE to Machine Learning

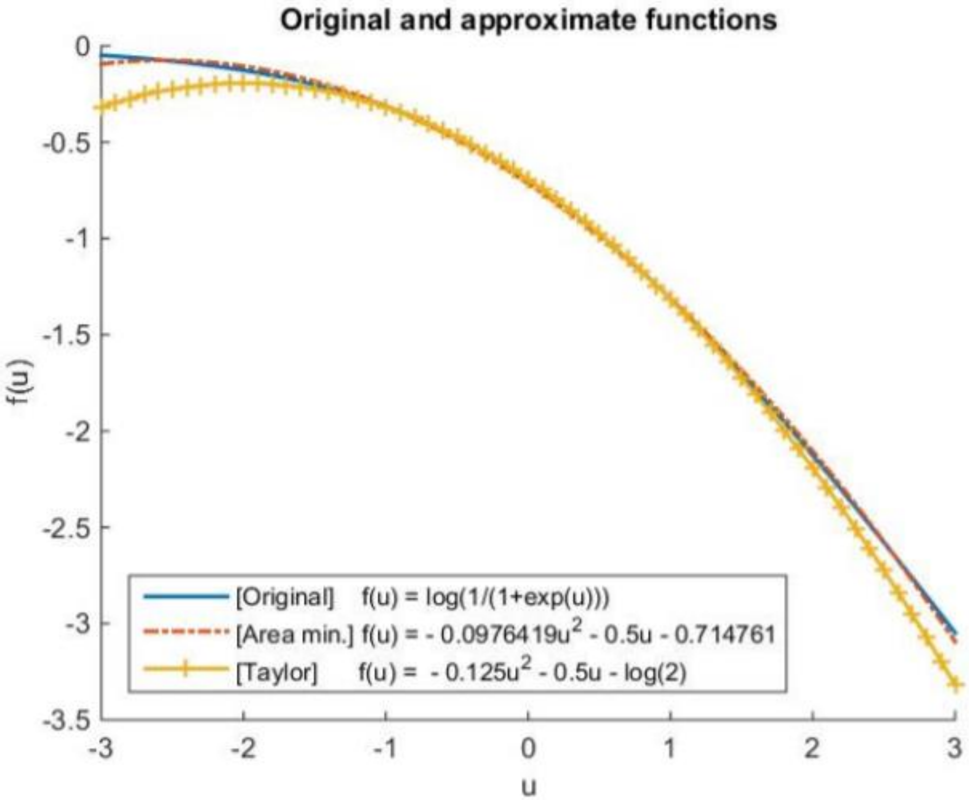
① 多项式近似 Polynomial approximation for logarithm function

$$\log\left(\frac{1}{1+\exp(u)}\right) \approx \sum_{j=0}^{k} a_j u^j$$

② 加密计算 Encrypted computation for each term in the polynomial function

$$loss = \log 2 - \frac{1}{2} y w^T x + \frac{1}{8}(w^T x)^2$$

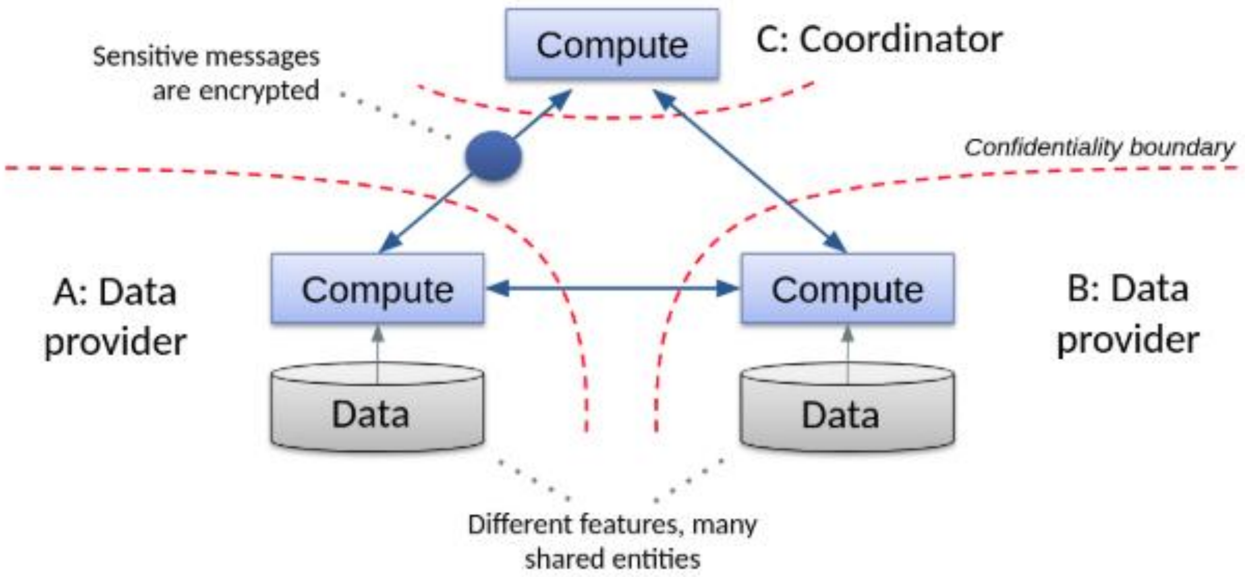
$$[[loss]] = [[\log 2]] + (-\frac{1}{2}) * [[y w^T x]] + \frac{1}{8}[[(w^T x)^2]]$$



- Kim, M.; Song, Y.; Wang, S.; Xia, Y.; and Jiang, X. 2018. Secure logistic regression based on homomorphic encryption: Design and evaluation. JMIR Med Inform 6(2)
- Y. Aono, T. Hayashi, T. P. Le, L. Wang, Scalable and secure logistic regression via homomorphic encryption, CODASPY16

Private Federated Learning on Vertically Partitioned Data via Entity Resolution and Additively HE [HHI+17]

- 纵向分割 Vertically partitioned dataset, one party knows label
- 需要可信第三方 Trusted Third Party (TTP) needed as the key pair distributor
- Paillier addictively homomorphic encryption
- 梯度信息、模型均暴露



Hardy, S., Henecka, W., Ivey-Law, H., Nock, R., Patrini, G., Smith, G., & Thorne, B. (2017). Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. *arXiv preprint arXiv:1711.10677*.

比较同态加密（HE），秘密共享（Secret Sharing) 与混淆电路 (Yao’s Garbled Circuit)

	Protocol by HE 同态加密	Protocol by secret sharing 秘密共享	Protocol by Garbled Circuit 混淆电路
Computations 计算成本	X	√	√
Communications 传输成本	√	X	X
Third party 需要第三方	X	√	X
Security 安全性	Computational security	Information theoretic security	Computational security

隐私保护的深度学习推断

CryptoNet

- Leveled homomorphic encryption
- Privately evaluate neural network

Ran Gilad-Bachrach et al. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In Proceedings of the 33nd International Conference on Machine Learning, ICML 2016

DeepSecure

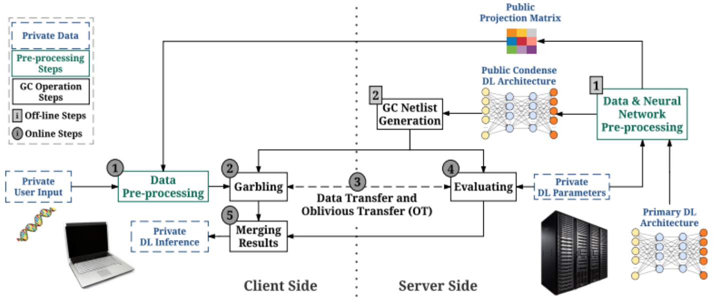


Figure 2: Global flow of DeepSecure framework including both off-line (indexed by rectangular icons) and online (indexed by oval icons) steps. The operations shown in the left hand side of the figure are executed by the client (Alice) while the operations on the right hand side are performed by the server (Bob).

- Yao’s GC

B. D. Rouhani, M. S. Riazi, F. Koushanfar, DeepSecure: Scalable Provably-Secure Deep Learning, CoRR, abs/1705.08963, 2017.

MiniONN

- Offline lattice-based AHE
- Online GC and secret-sharing

Jian Liu et al, Oblivious Neural Network Predictions via MiniONN transformations, In Proceedings of the 2017 ACM SIGSAC CCS 2017

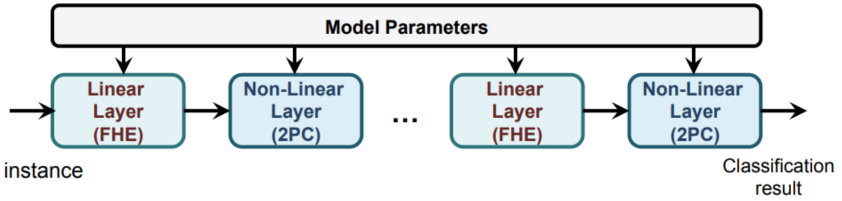
Chameleon

- 3-server model
- trusted third-party dealers

M. Sadegh Riazi et al. Chameleon: A hybrid secure computation framework for machine learning applications. Cryptology ePrint Archive, Report 2017/1164, 2017. https://eprint.iacr. org/2017/1164.

GAZELLE [GVC18]

- Lattice-based packed addictive HE for linear layer
- Garbled Circuit for non-linear layer
- Three orders of magnitude faster than CryptoNets [DBL^{+}16]

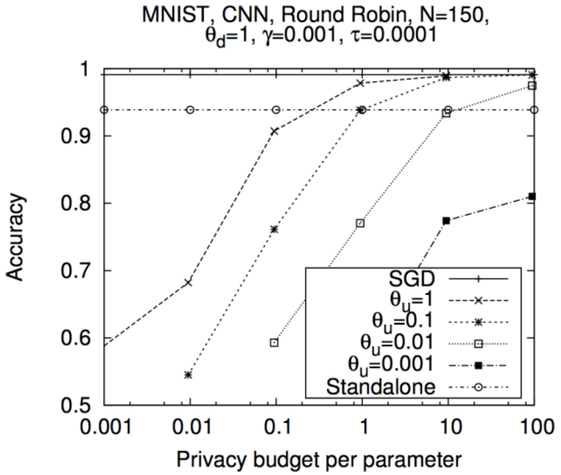


差分隐私

For any two datasets *D* and *D’* differing in a single item and any output *O* of function *f,*

ε controls the tradeoff between accuracy and privacy

$$\Pr\{f(D) \in O\} \leq exp(\epsilon) \cdot \Pr\{f(D') \in O\}.$$



Reza Shokri and Vitaly Shmatikov. 2015. Privacy-Preserving Deep Learning. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS ’15). ACM, New York, NY, USA, 1310–1321.

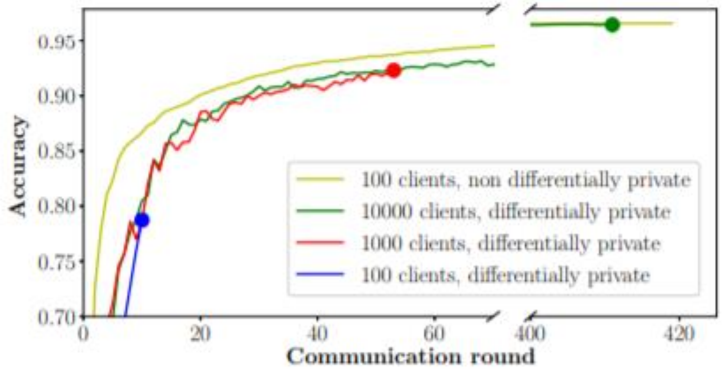


Figure 1: Accuracy of digit classification from non-IID MNIST-data held by clients over the course of decentralized training. For differentially private federated optimization, dots at the end of accuracy curves indicate that the δ-threshold was reached and training therefore stopped.

- From 100 to 1,000 clients, model accuracy does not converge and stays significantly below the non-differentially private approach

Robin C. Geyer, Tassilo Klein, and Moin Nabi. 2017. Differentially Private Federated Learning: A Client Level Perspective. CoRR abs/1712.07557 (2017)

纵向联邦学习 Vertical Federated Learning

目标:

- ➢ A方 和 B 方 联合建立模型

假设：

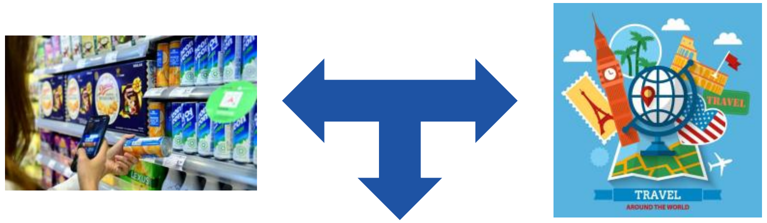
- ➢ 只有一方有标签 Y
- ➢ 两方均不暴露数据

挑战:

- ➢ 只有X的一方没有办法建立模型
- ➢ 双方不能交换共享数据

预期：

- ➢ 双方均获得数据保护
- ➢ 模型无损失（ LOSSLESS ）



(X)

(U, Z)

(V, Y)

ID	X1	X2	X3
U1	9	80	600
U2	4	50	550
U3	2	35	520
U4	10	100	600
U5	5	75	600
U6	5	75	520
U7	8	80	600

Retail A Data

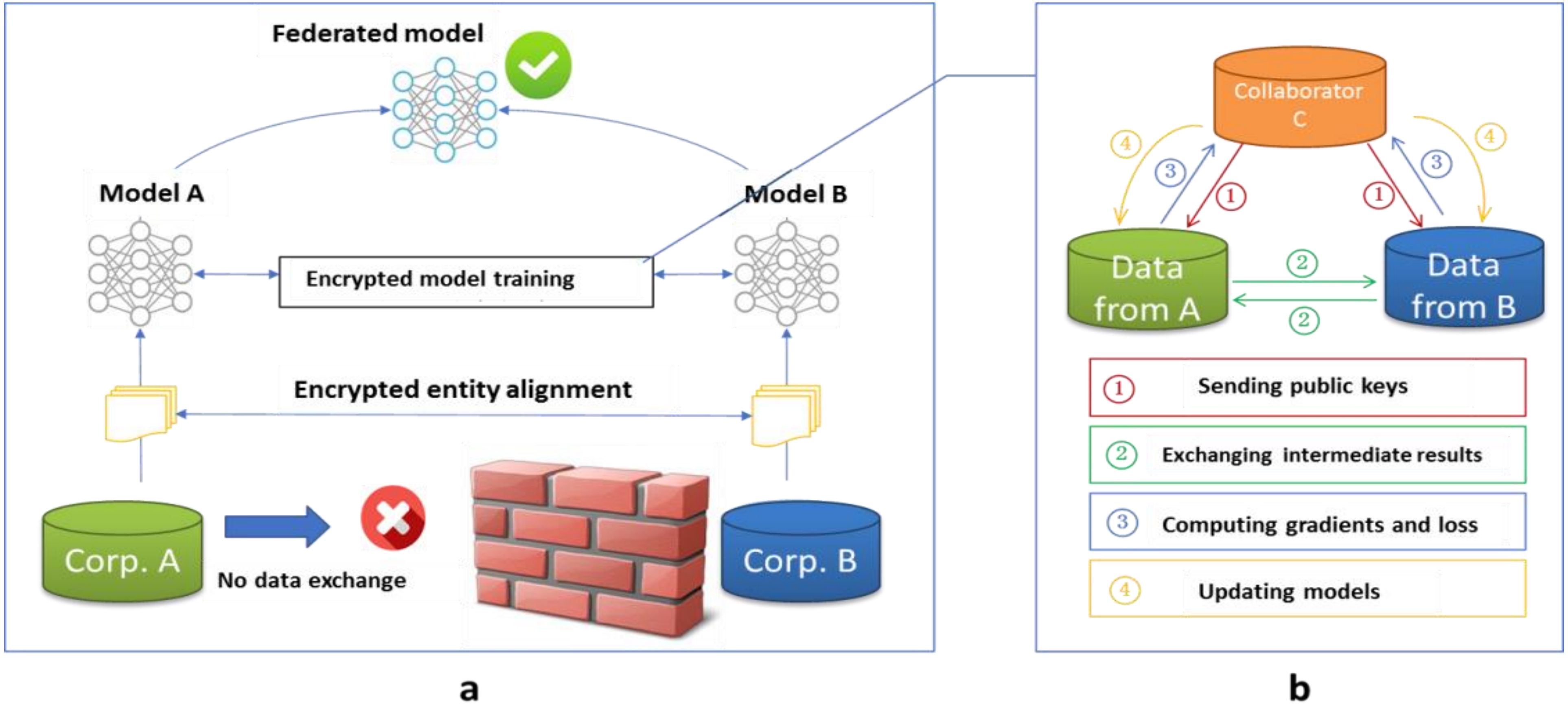
ID	X4	X5	Y
U1	6000	600	No
U2	5500	500	Yes
U3	7200	500	Yes
U4	6000	600	No
U8	6000	600	No
U9	4520	500	Yes
U10	6000	600	No

Bank B Data

安全定义 Security Definition

- All parties are ***honest-but-curious***.
- We assume a threat model with a semi-honest adversary who can corrupt at most one of the two data clients.
- For a protocol ***P*** performing $(O_A,O_B)=\boldsymbol{P}(I_A,I_B)$, where O_A and O_B are party A and B's output and I_A and I_B are their inputs, P is secure against A if there exists an infinite number of (I'_B,O'_B) pairs such that $(O_A,O'_B)=\boldsymbol{P}(I_A,I'_B)$.
- A practical solution to control information disclosure.

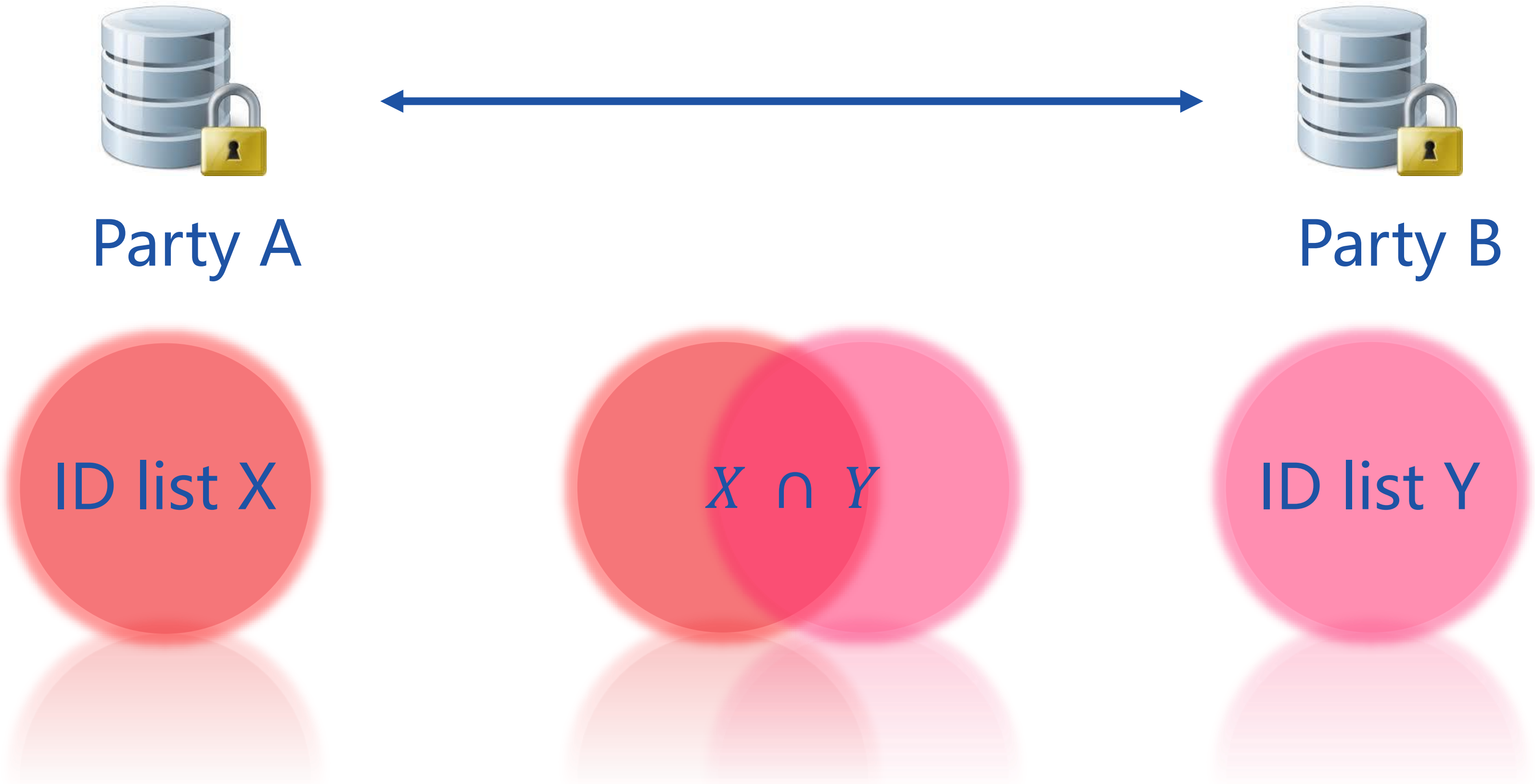
联邦学习系统 Architecture of Vertical Federated Learning



- **Each Site Holds Own Data**
- **Performance is LOSSLESS**
- **Solution for Common Users**

隐私保护下的样本ID匹配 privacy-Preserving Entity Match

- Party A and B learn their overlapping IDs but nothing else



Party A

Party B

ID list X

$X \cap Y$

ID list Y

隐私保护下的样本ID匹配 privacy-Preserving Entity Match

Party A

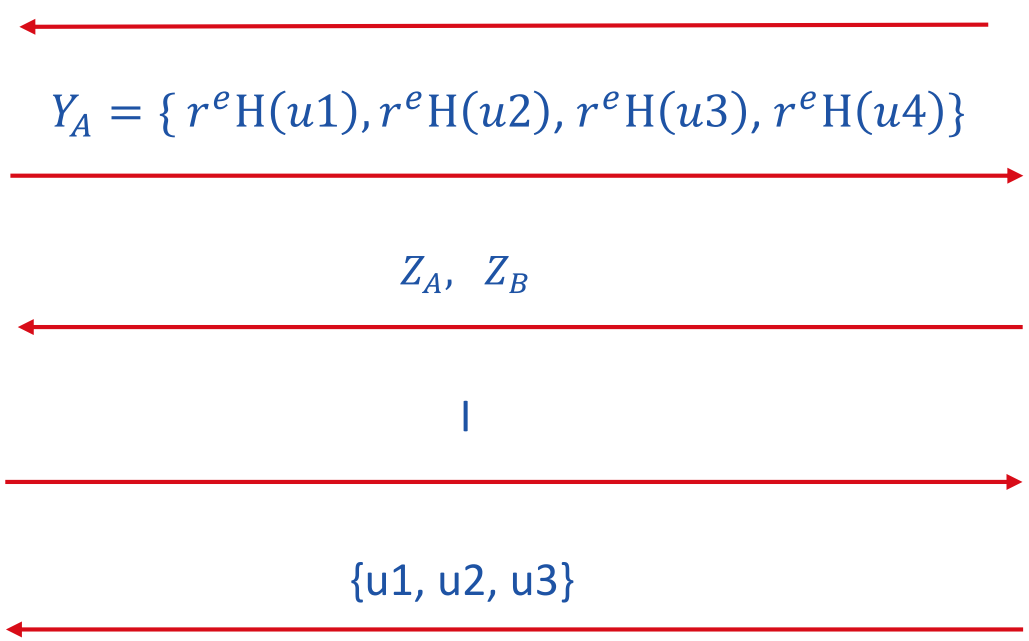
X_A: {u1, u2, u3, u4}

RSA + Hash

Party B

X_B: {u1, u2, u3, u5}

RSA：n, e, d



public key: (n, e)

Y_A = {r^e * H(u) | u ∈ X_A , r: rand }

$Y_A = \{\, r^e\mathrm{H}(u1), r^e\mathrm{H}(u2), r^e\mathrm{H}(u3), r^e\mathrm{H}(u4)\}$

Z_A=
$(r^e\mathrm{H}(ui))^d$=r*$(\mathrm{H}(ui))^d$ % n
| $r^e\mathrm{H}(ui)$ ∈ Y_A}

D_A=
{H(r*$(\mathrm{H}(ui))^d$/ r) =H($(\mathrm{H}(ui))^d$) |
r*$(\mathrm{H}(ui))^d$ ∈ Z_A}

Z_A, Z_B

Z_B = { H($(\mathrm{H(u)})^d$) | u ∈ X_B}

I

I = $D_A \cap Z_B$
= {H($(\mathrm{H}(u1))^d$), H($(\mathrm{H}(u2))^d$),
H($(\mathrm{H}(u3))^d$)}

{u1, u2, u3}

I, Z_B => {u1, u2, u3}

隐私保护下的训练过程: Linear regression

$$\min_{\Theta_A,\Theta_B} \sum_i ||\Theta_A x_i^A + \Theta_B x_i^B - y_i||^2 + \frac{\lambda}{2}(||\Theta_A||^2 + ||\Theta_B||^2)$$

$$u_i^A = \Theta_A x_i^A,\ u_i^B = \Theta_B x_i^B$$

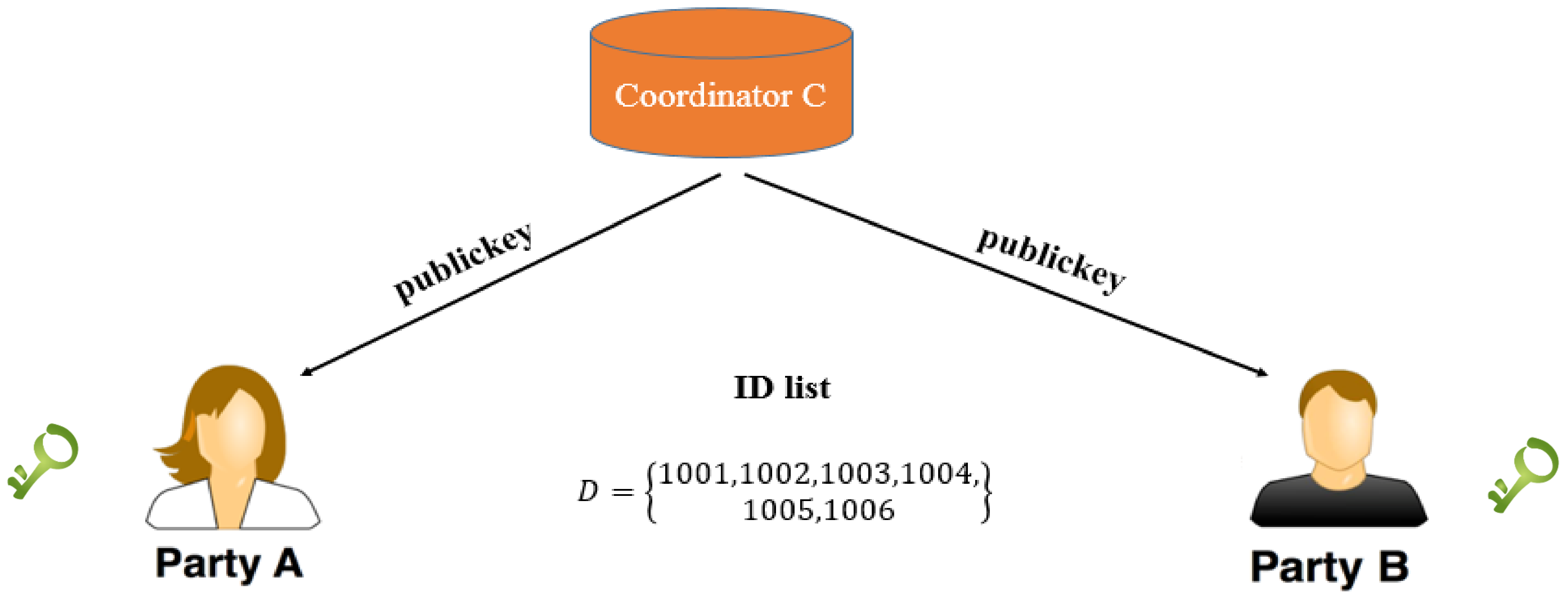
$$[[\mathcal{L}]] = [[\sum_i ((u_i^A + u_i^B - y_i))^2 + \frac{\lambda}{2}(||\Theta_A||^2 + ||\Theta_B||^2)]]$$

$$[[d_i]] = [[u_i^A]] + [[u_i^B - y_i]]$$

$$[[\frac{\partial \mathcal{L}}{\partial \Theta_A}]] = \sum_i [[d_i]] x_i^A + [[\lambda \Theta_A]]$$

$$[[\frac{\partial \mathcal{L}}{\partial \Theta_B}]] = \sum_i [[d_i]] x_i^B + [[\lambda \Theta_B]]$$

WeBank



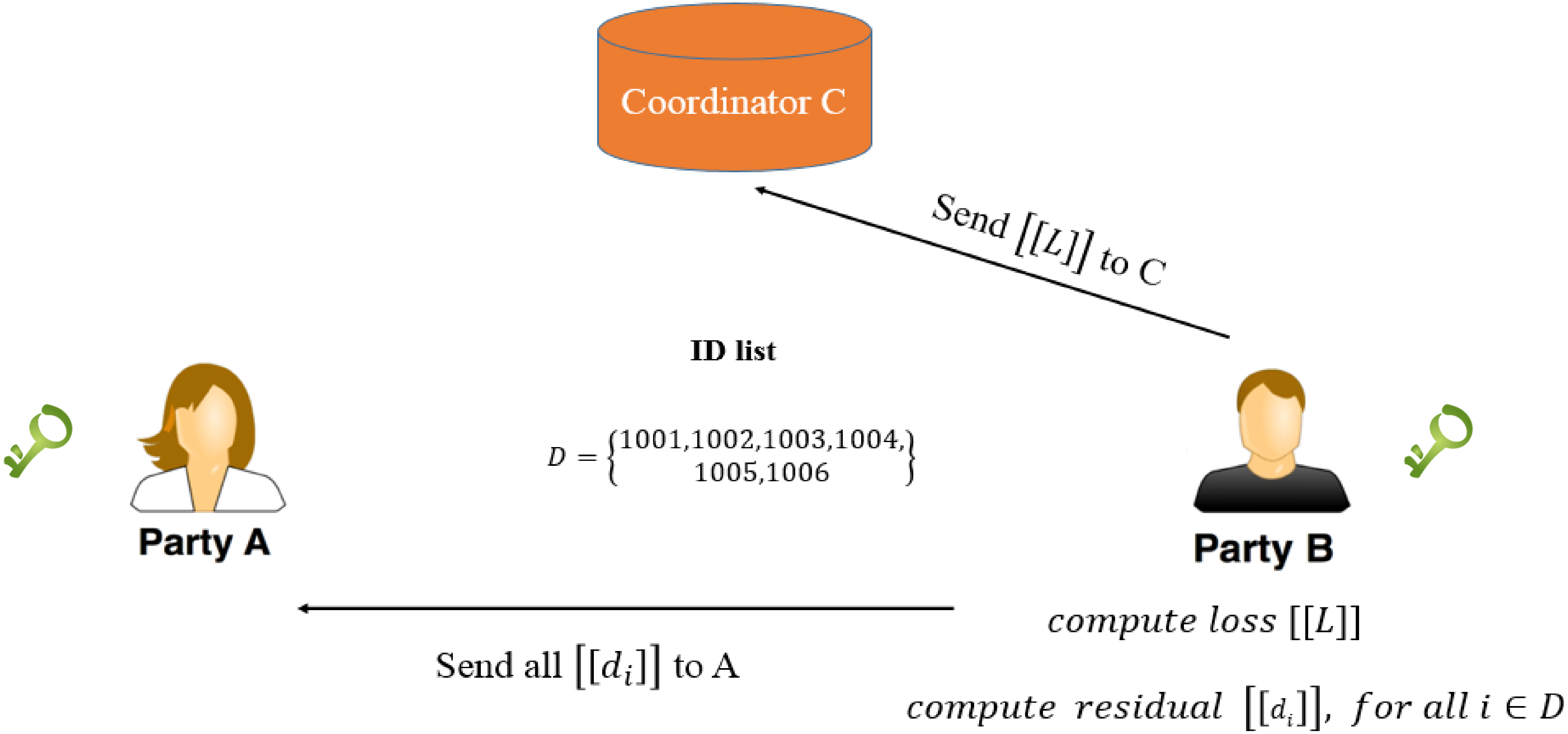
ID list

$$D = \begin{Bmatrix} 1001,1002,1003,1004, \\ 1005,1006 \end{Bmatrix}$$

1. Compute:
   $u_i^A = \Theta^A x_i^A = \Theta_1^A x_{i1}^A + \Theta_2^A x_{i2}^A \quad i \in D$

2. Compute and encrypt:
   $\left[\left[u_i^A\right]\right] \quad i \in D \quad \text{and} \quad \left[\left[\sum_{i \in D}\left(\left(u_i^A\right)^2\right) + \frac{\lambda}{2}\Theta_A^2\right]\right]$

3. send $\left[\left[u_i^A\right]\right] \quad i \in D \quad \text{and} \quad \left[\left[\sum_{i \in D}\left(\left(u_i^A\right)^2\right) + \frac{\lambda}{2}\Theta_A^2\right]\right]$ to B



Coordinator C

Send $[[L]]$ to C

ID list

$$D = \begin{Bmatrix} 1001,1002,1003,1004, \\ 1005,1006 \end{Bmatrix}$$

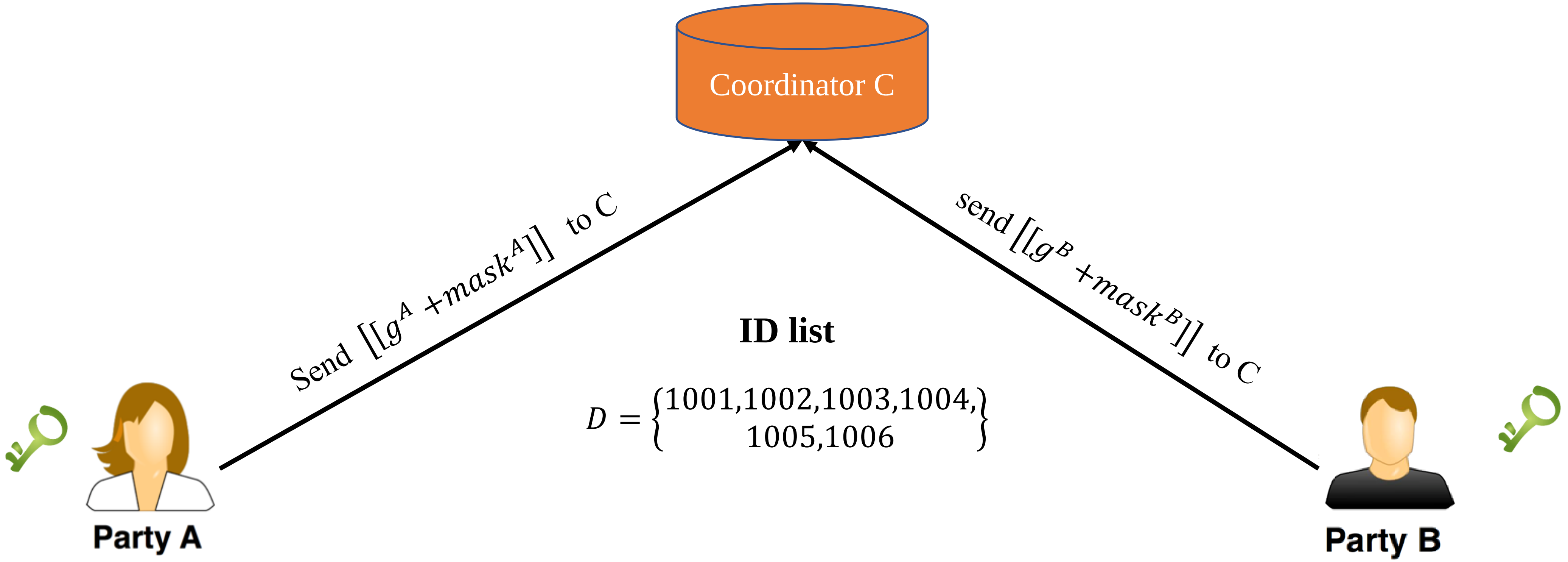
Party A

Party B

$compute\ loss\ [[L]]$

Send all $[[d_i]]$ to A

$compute\ \ residual\ \ [[d_i]],\ for\ all\ i \in D$



Coordinator C

Send $[[g^A + mask^A]]$ to C

send $[[g^B + mask^B]]$ to C

ID list

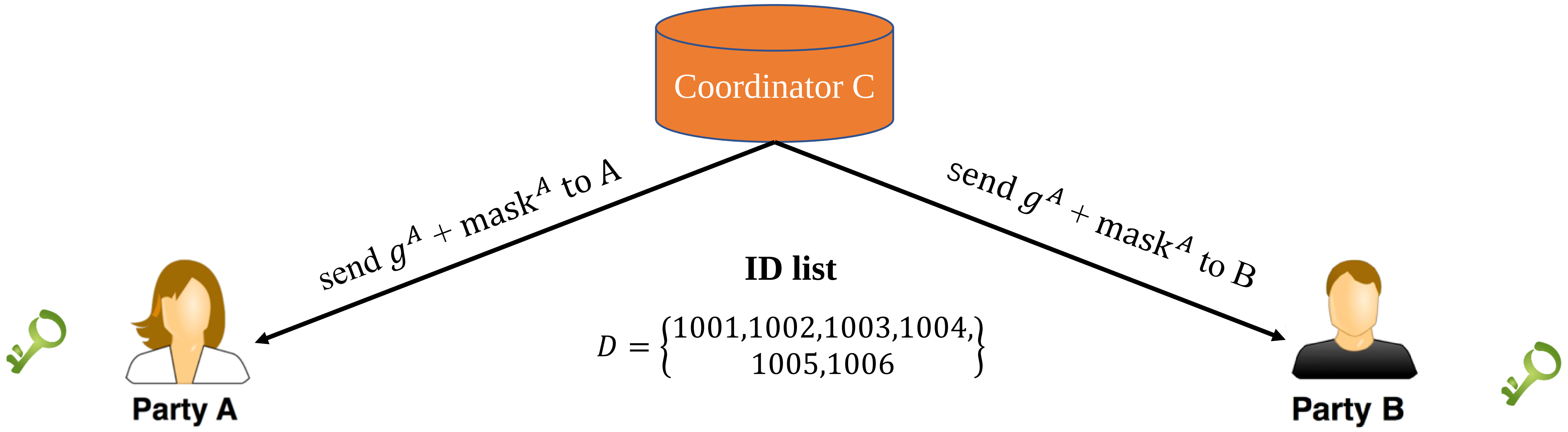
$$D = \begin{Bmatrix} 1001,1002,1003,1004, \\ 1005,1006 \end{Bmatrix}$$

Party A

Party B

$$compute\ local\ gradient \left[\left[\frac{\partial L}{\partial \Theta^A}\right]\right] = [[g^A]] + [[\ mask^A]]$$

$$compute\ local\ gradient \left[\left[\frac{\partial L}{\partial \Theta^B}\right]\right] = [[g^B]] + [[\ mask^B]]$$



Coordinator C

send $g^A + \text{mask}^A$ to A

send $g^A + \text{mask}^A$ to B

ID list

$$D = \begin{Bmatrix} 1001,1002,1003,1004, \\ 1005,1006 \end{Bmatrix}$$

Party A

Party B

update parameter Θ^A

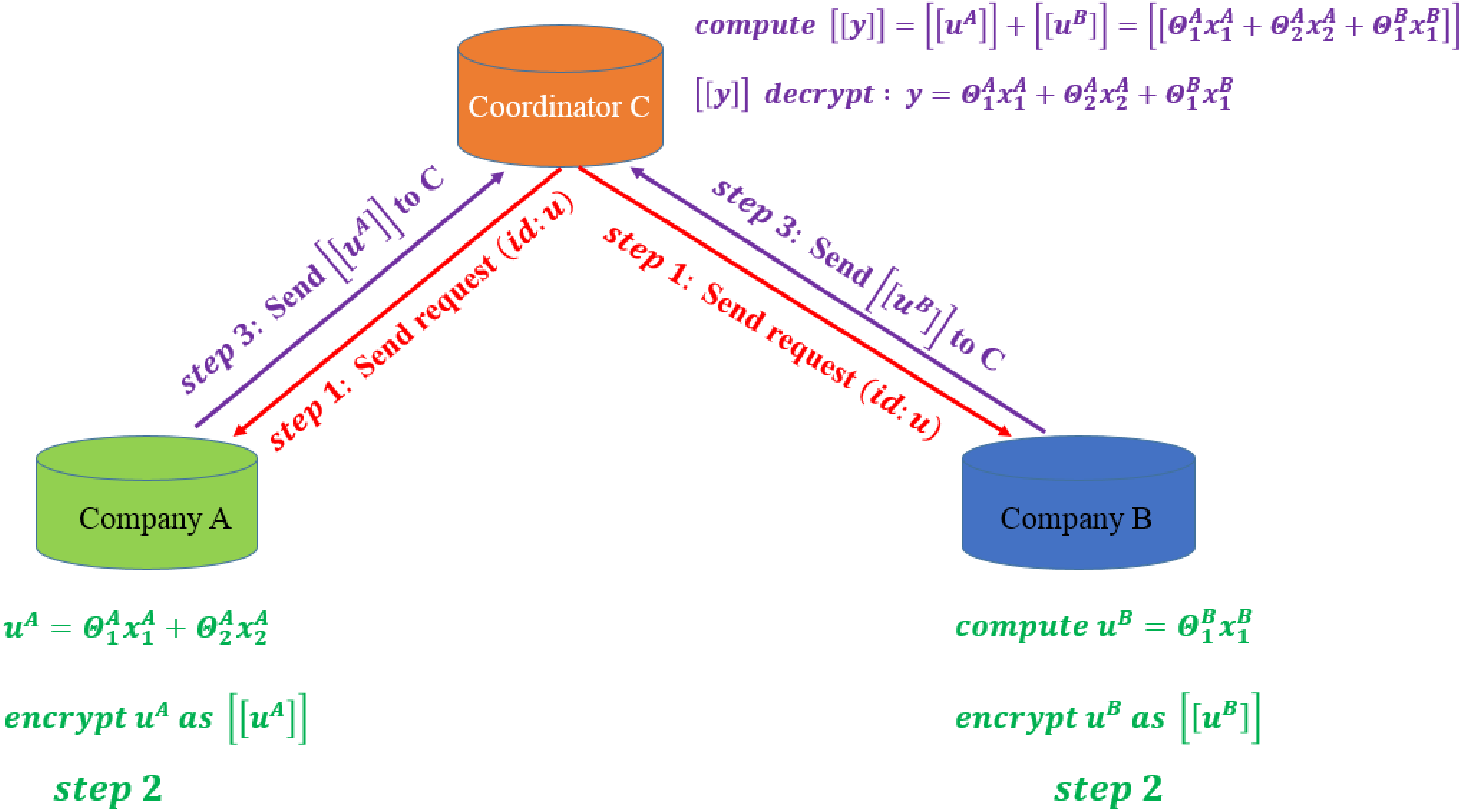
$$\Theta^A = \Theta^A - \eta\left(\frac{\partial L}{\partial \Theta^A} - \text{mask}^A\right)$$

update parameter Θ^B

$$\Theta^B = \Theta^B - \eta\left(\frac{\partial L}{\partial \Theta^B} - \text{mask}^B\right)$$

- 每个参与方并不知道另一方的数据和特征.
- 每个参与方只得到自己侧的模型参数（半模型）。

推断过程



compute $[[y]] = [[u^A]] + [[u^B]] = [[\Theta_1^A x_1^A + \Theta_2^A x_2^A + \Theta_1^B x_1^B]]$

$[[y]]$ *decrypt* : $y = \Theta_1^A x_1^A + \Theta_2^A x_2^A + \Theta_1^B x_1^B$

Coordinator C

step 3: Send $[[u^A]]$ *to C*

step 1: Send request (id: u)

step 3: Send $[[u^B]]$ *to C*

step 1: Send request (id: u)

Company A

Company B

$u^A = \Theta_1^A x_1^A + \Theta_2^A x_2^A$

encrypt u^A *as* $[[u^A]]$

step 2

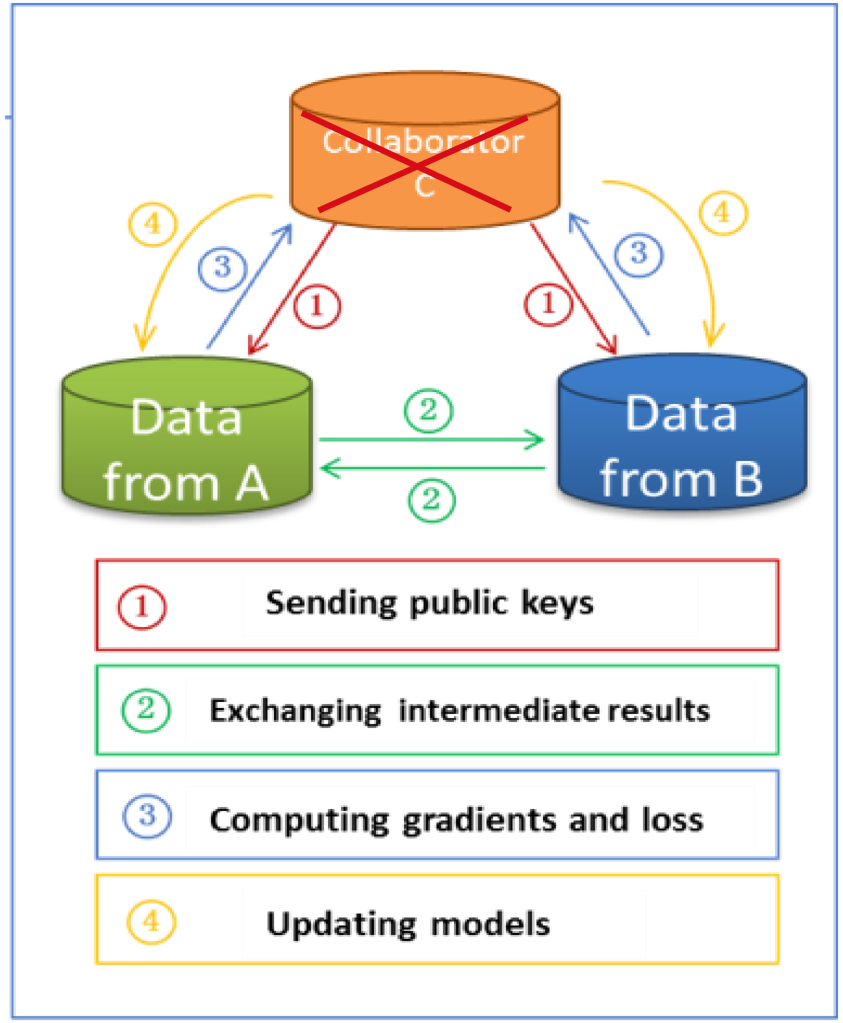
compute $u^B = \Theta_1^B x_1^B$

encrypt u^B *as* $[[u^B]]$

step 2

可否去掉第三方?

Yes!



① Sending public keys

② Exchanging intermediate results

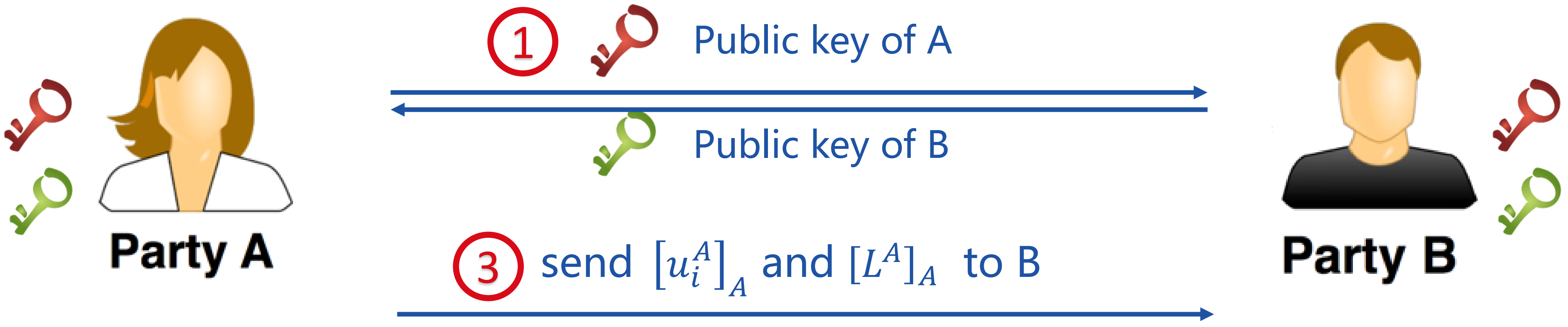
③ Computing gradients and loss

④ Updating models

b

WeBank

两方解决方案



① Public key of A

Public key of B

Party A

Party B

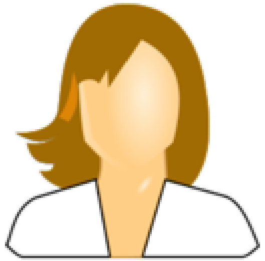
③ send $[u_i^A]_A$ and $[L^A]_A$ to B

② compute $u_i^A = \Theta_A x_i^A$,

$L^A = \sum_{i \in D}\left(\left(u_i^A\right)^2\right) + \frac{\lambda}{2}\Theta_A^2$

encrypt $[u_i^A]_A$, $[L^A]_A$

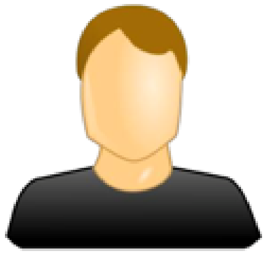
两方解决方案



Party A

5 send $[L+M^B]_A$, $\left[\frac{\partial L}{\partial \Theta_B}+R^B\right]_A$ and $[[d_i]_A]_B$ to A

7 Send $L+M^B, \frac{\partial L}{\partial \Theta_B}+R^B$ 和 $\left[\left[\frac{\partial L}{\partial \Theta_A}\right]_A\right]_B$ to B



Party B

6

$[L+M^B]_A \rightarrow L+M^B$

$\left[\frac{\partial L}{\partial \Theta_B}+R^B\right]_A \rightarrow \frac{\partial L}{\partial \Theta_B}+R^B$

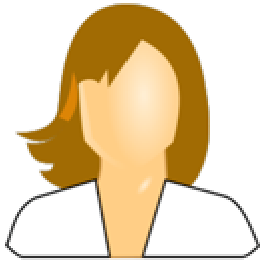
compute $\left[\left[\frac{\partial L}{\partial \Theta_A}\right]_A\right]_B$

4 compute

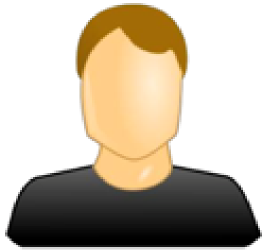
$[L]_A$, $\left[\frac{\partial L}{\partial \Theta_B}\right]_A$ 和$[d_i]_A$

WeBank

两方解决方案



Party A



Party B

8

$$L + M^B \to L$$

$$\frac{\partial L}{\partial \Theta_B} + R^B \to \frac{\partial L}{\partial \Theta_B}$$

$$\left[\left[\frac{\partial L}{\partial \Theta_A}\right]_A\right]_B \to \left[\frac{\partial L}{\partial \Theta_A}\right]_A$$

$$\Theta_B = \Theta_B - \gamma \frac{\partial L}{\partial \Theta_B}$$

9 send $\left[\frac{\partial L}{\partial \Theta_A}\right]_A$ to A

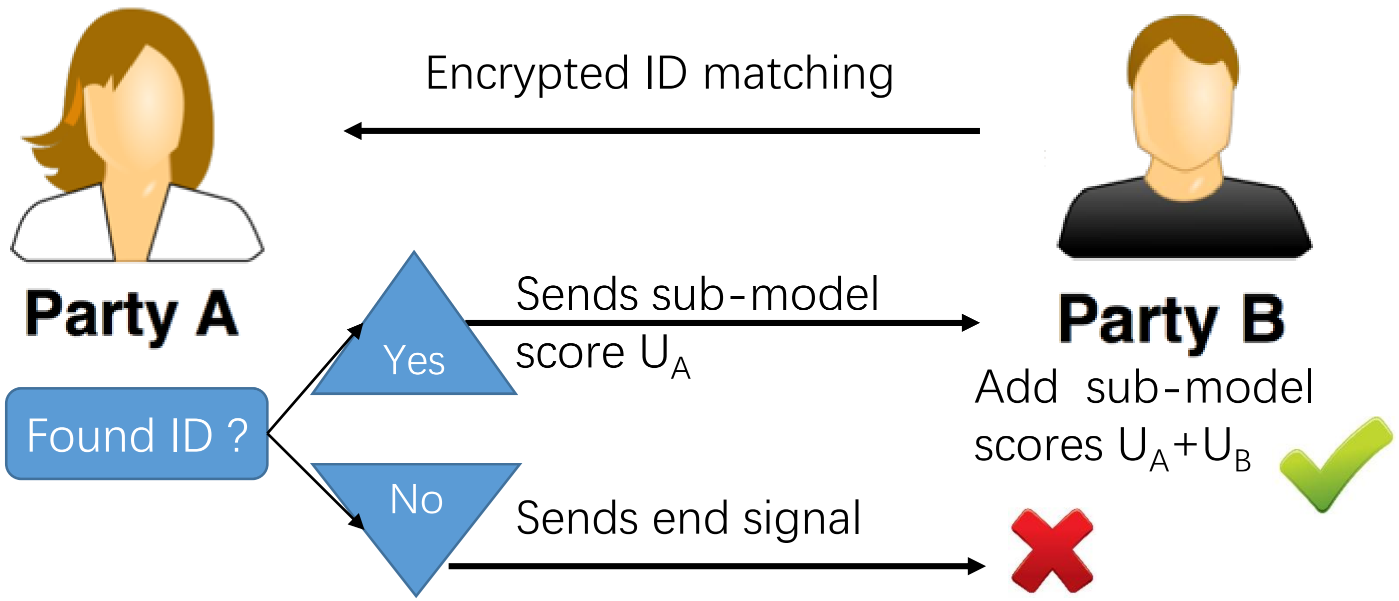
10

$$\left[\left[\frac{\partial L}{\partial \Theta_A}\right]_A\right]_B \to \left[\frac{\partial L}{\partial \Theta_A}\right]_A$$

$$\Theta_A = \Theta_A - \gamma \frac{\partial L}{\partial \Theta_A}$$

两方推理方案

- Suppose a new user ID arrives at Party B,



Encrypted ID matching

Party A

Found ID ?

Yes

Sends sub-model score U_A

No

Sends end signal

Party B

Add sub-model scores U_A+U_B

Step 1
Party B sends encrypted ID to A

Step 2
Party A and B compute local results

Step 3
Party A sends its results to B

Step 4
Party B combines results

安全性分析

- Security against third-party C
  - All C learns are the masked gradients and the randomness and secrecy of the masked matrix are guaranteed
- Security against each other
  - Party A learns its gradient at each step, but this is not enough for A to learn any information about B
  - Inability of solving *n* equations with more than *n* unknowns
- Security in the *semi-honest* setting

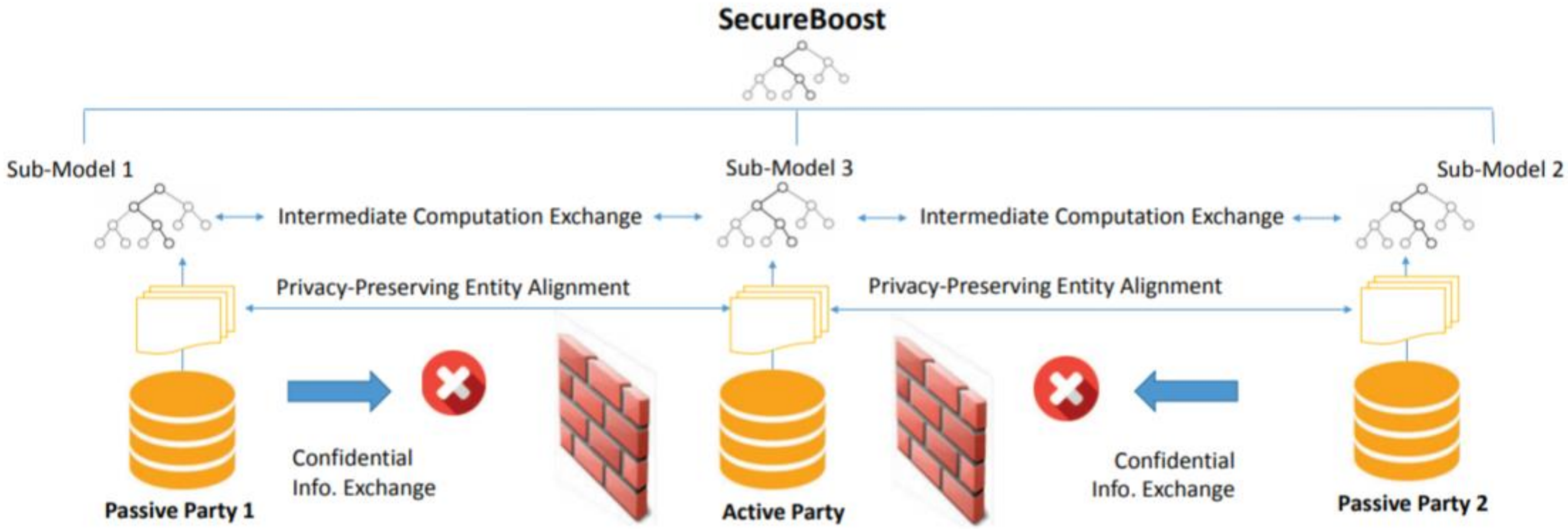
SecureBoost

1 **隐私保护下的样本ID匹配** Privacy-Preserving Entry-id Match

2 **联合学习** Collaboratively learn a shared gradient-tree boosting model

- using the summation of encrypted gradient to determine the best split over different data owners

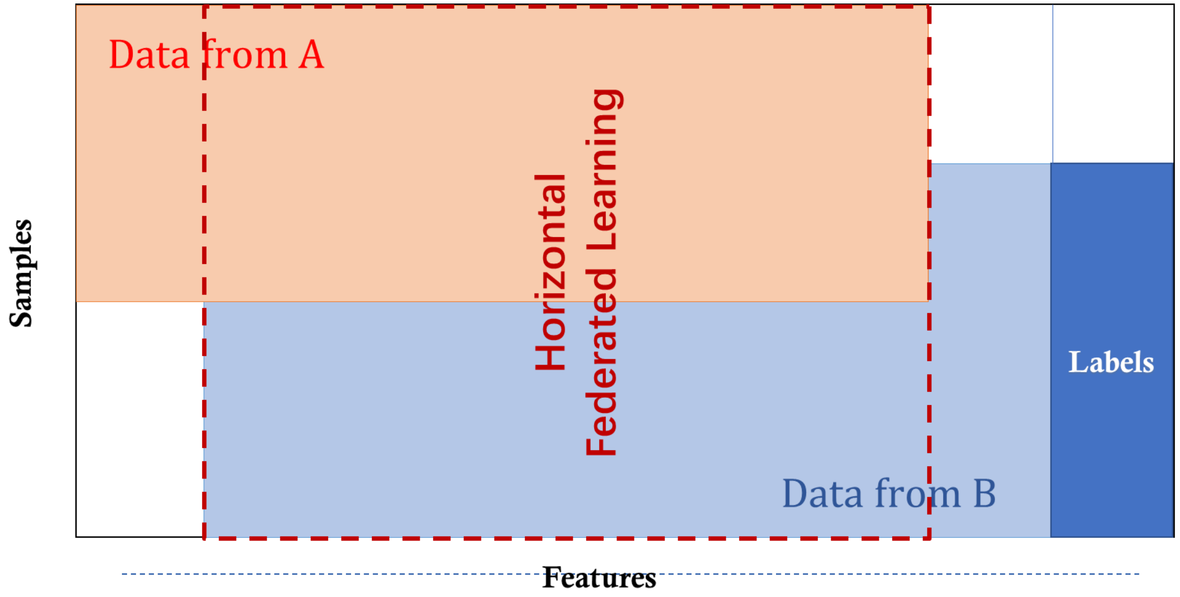
Contribution: lossless meanwhile secure



Kewei Cheng et al, **SecureBoost: A Lossless Federated Learning Framework, https://arxiv.org/abs/1901.08755**

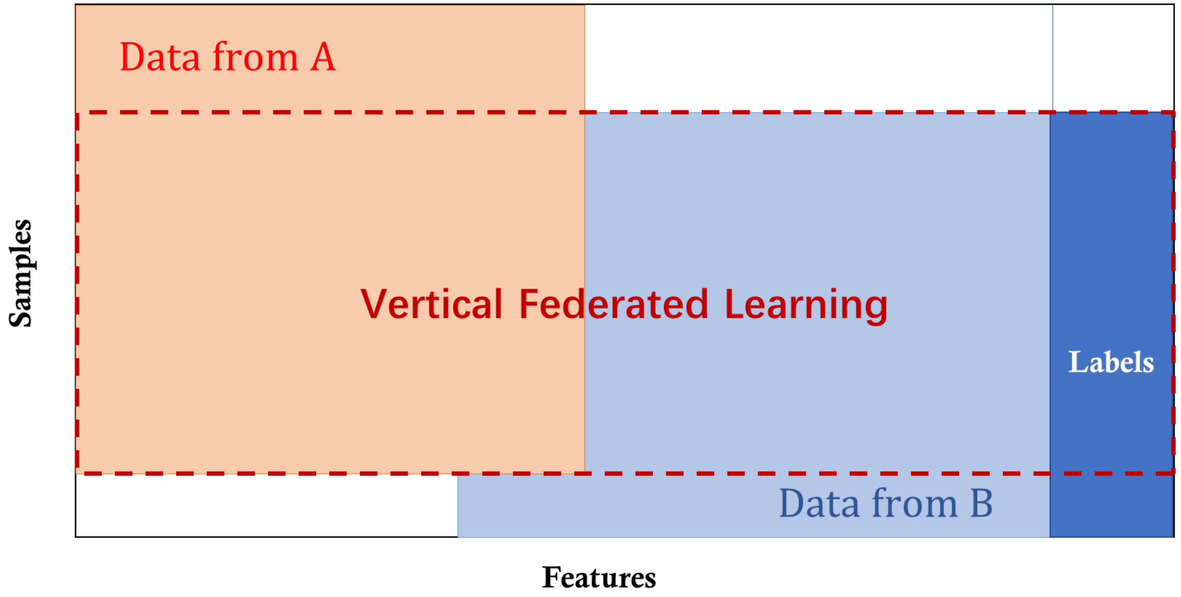
联邦学习分类 Categorization of Federated Machine Learning

横向联邦 Horizontal FML ✓



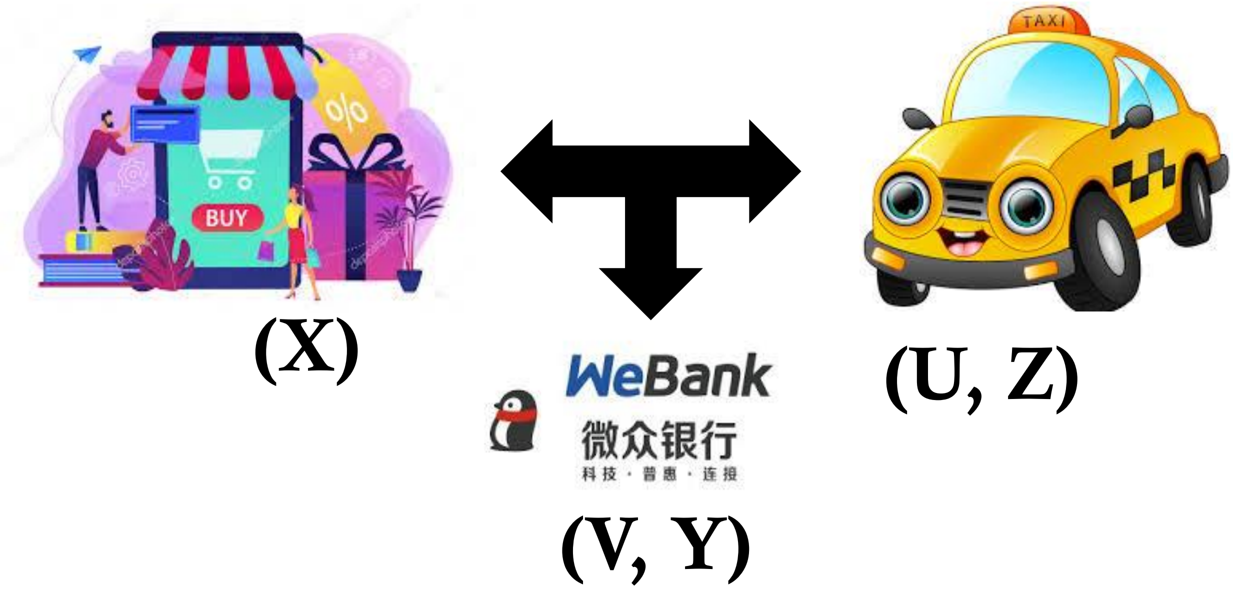
- 数据方特征维度相同

纵向联邦 Vertical FML ✓



- 数据方样本ID相同

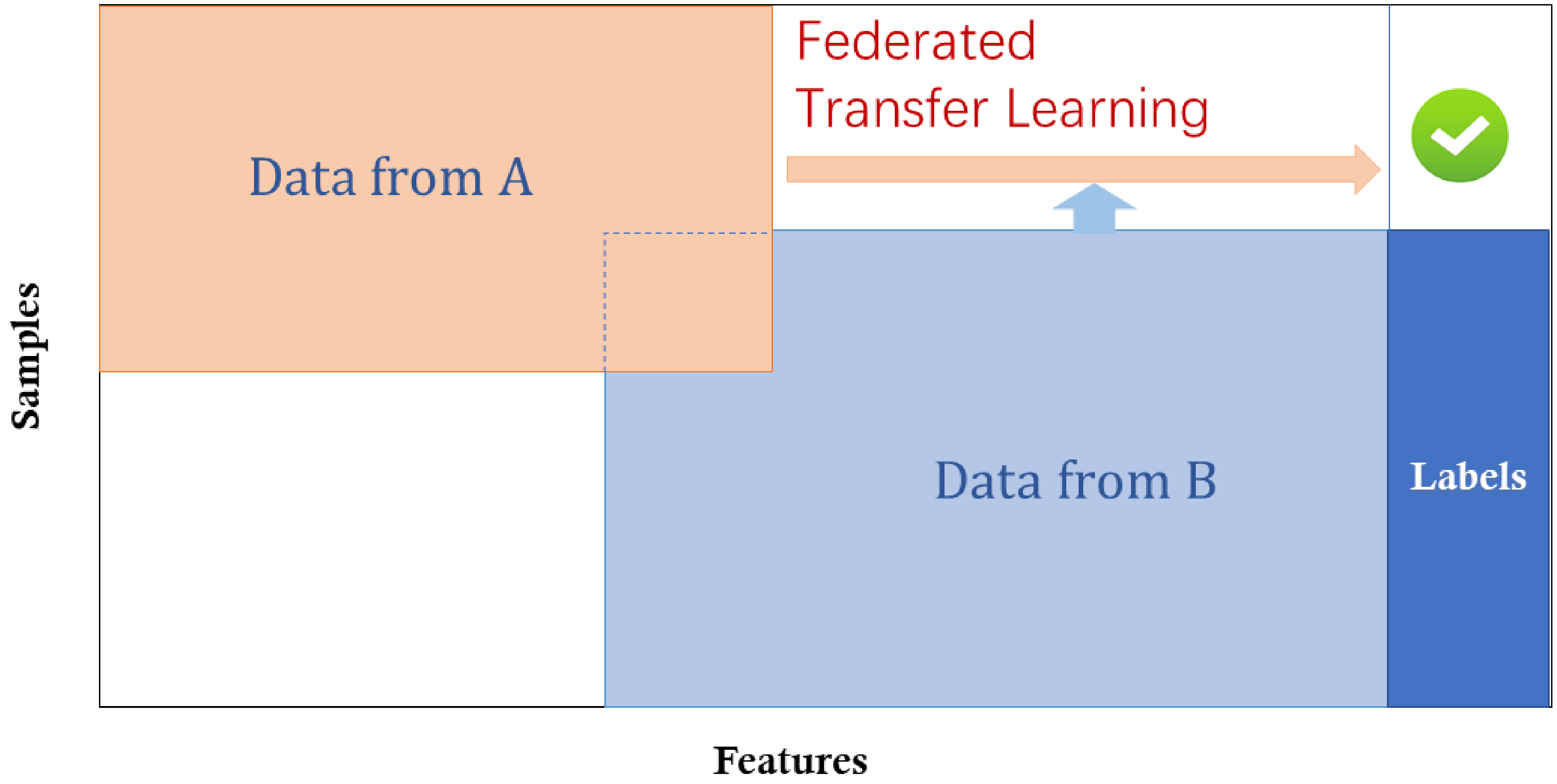
- 样本ID和特征没有足够的匹配怎么办?



(X)

(U, Z)

(V, Y)

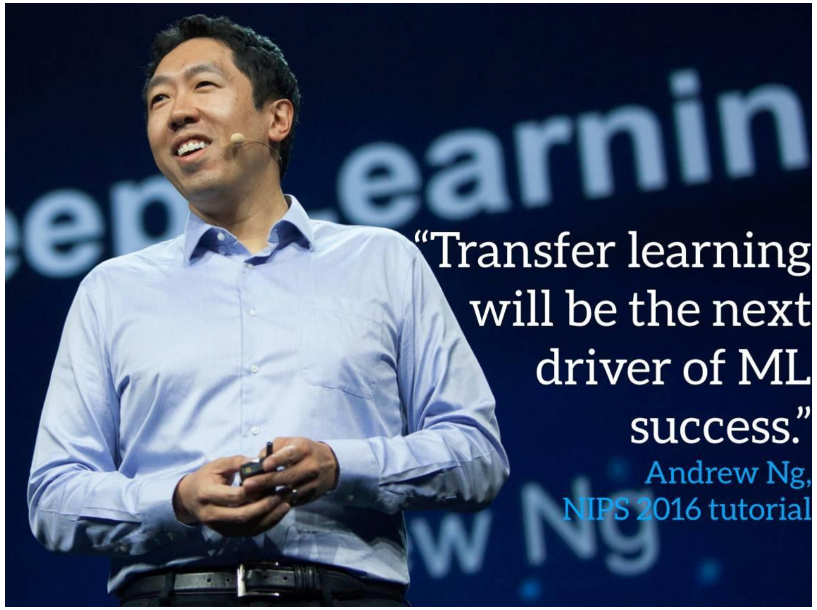

Federated
Transfer Learning
✓
Data from A
Data from B
Labels
Samples
Features

02

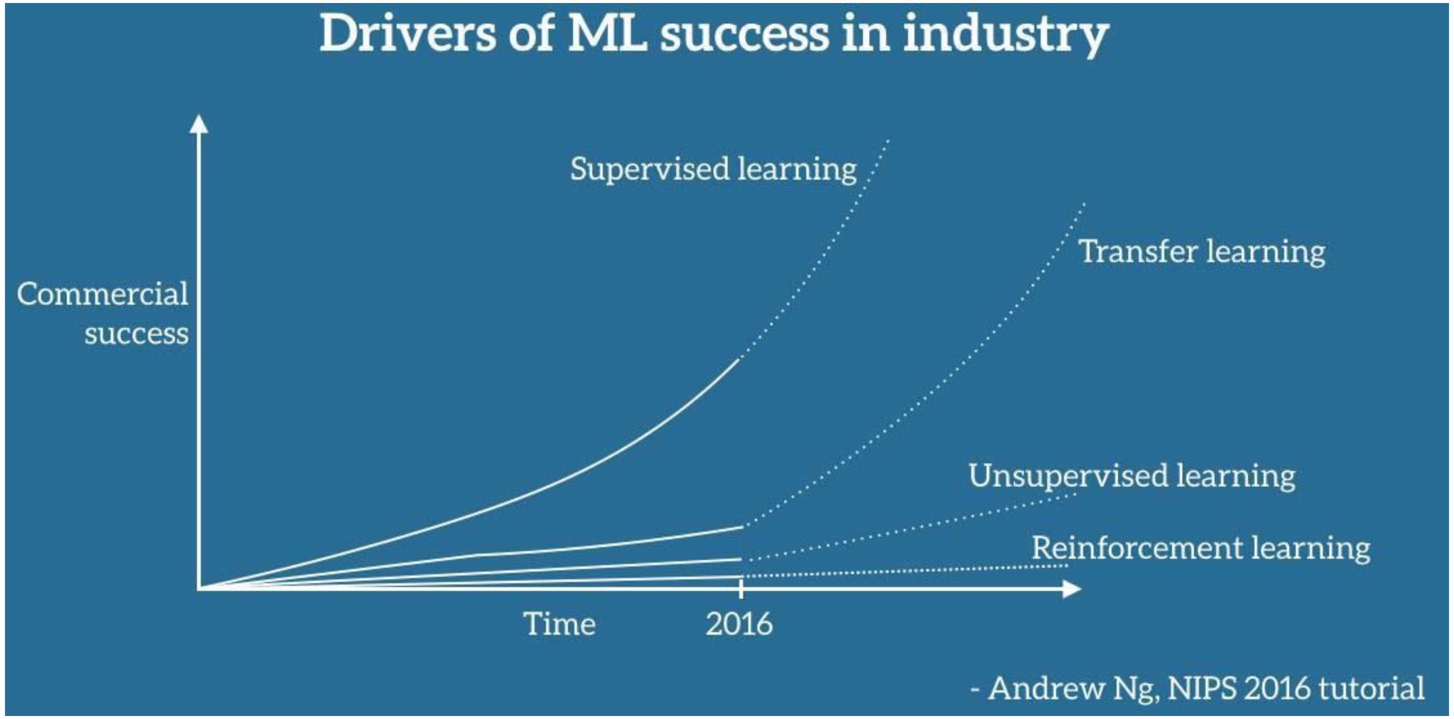
Federated Transfer Learning

实现保护用户隐私下的知识共享

迁移学习 Transfer Learning



“Transfer learning will be the next driver of ML success.”
Andrew Ng, NIPS 2016 tutorial


Drivers of ML success in industry

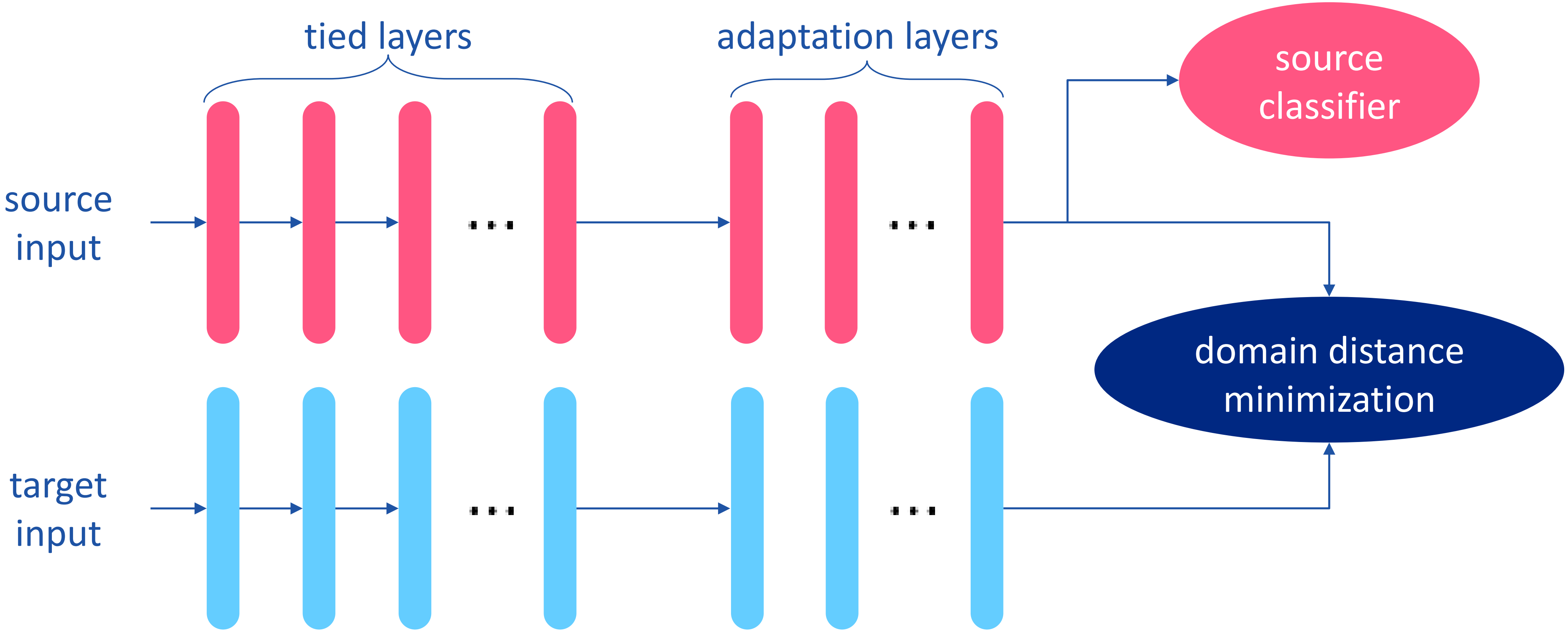
- Andrew Ng, NIPS 2016 tutorial

S. Ruder，*Transfer Learning - Machine Learning's Next Frontier*

迁移学习的核心: 找到不变量

- Objective

$$\mathcal{L} = \mathcal{L}_{\mathrm{source}} + \mathcal{L}_{\mathrm{distance}}$$

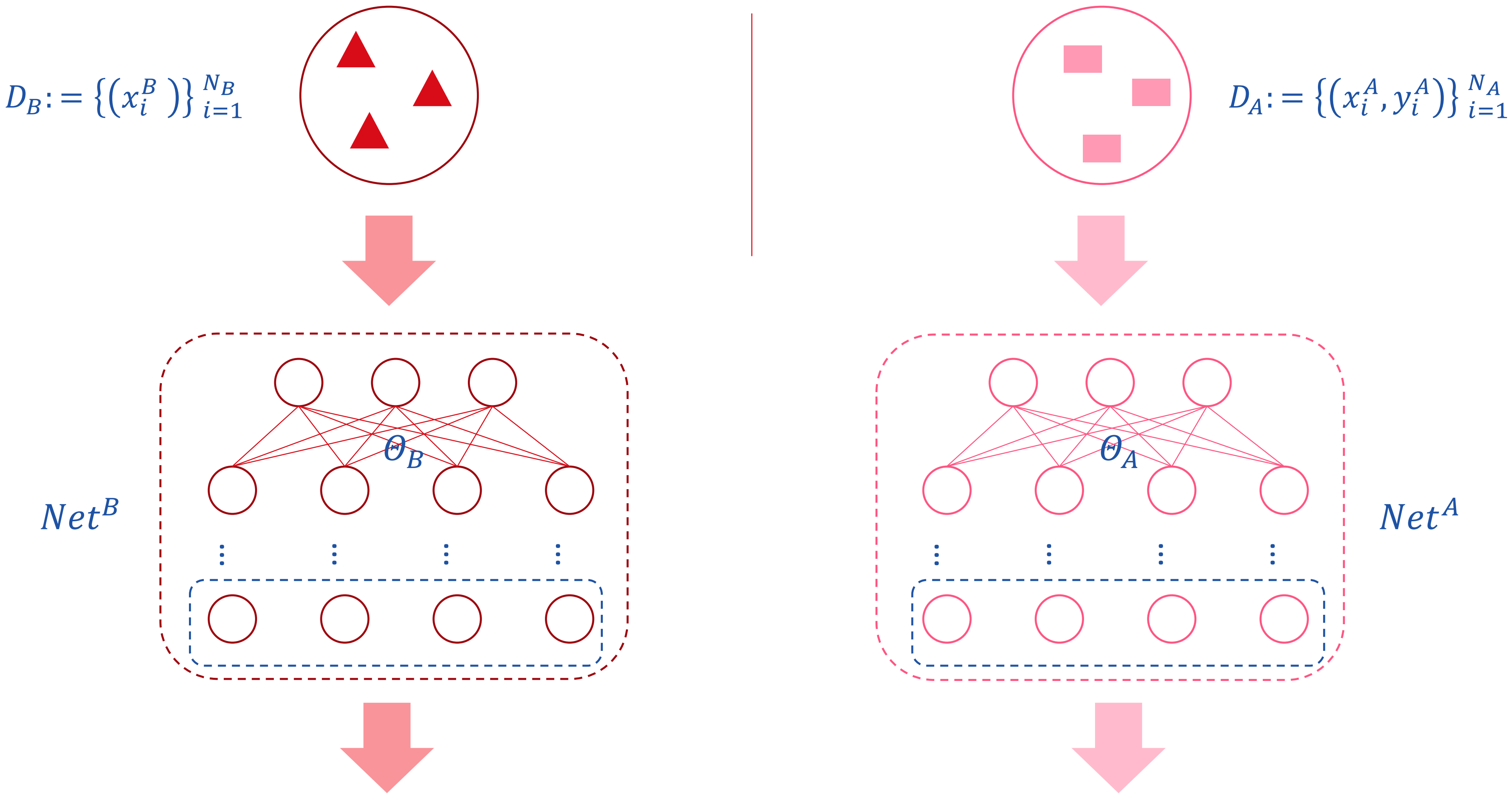


WeBank

如何在不共享数据的前提下做迁移学习？

target domain

source domain



$D_B := \{(x_i^B)\}_{i=1}^{N_B}$

$D_A := \{(x_i^A, y_i^A)\}_{i=1}^{N_A}$

Net^B

θ_B

Net^A

θ_A

$u_i^B, i \in \{1, \dots, N_B\}$

$u_i^A, i \in \{1, \dots, N_A\}$

通用性解决方案 A General Heterogeneous Transfer Learning Approach

Prediction Function

$$\varphi(u_j^B) = \varphi(u_1^A, y_1^A \ldots u_{N_A}^A, y_{N_A}^A, u_j^B).$$

For example, a translator function $\varphi(u_j^B) = \frac{1}{N_A} \sum_i^{N_A} y_i^A u_i^A (u_j^B)'$

Without loss of generality, assume linearly separable, $\varphi(u_j^B) = \Phi^A \mathcal{G}(u_j^B)$

Prediction Loss

$$\underset{\Theta^A, \Theta^B}{\operatorname{argmin}} \mathcal{L}_1 = \sum_i^{N_c} \ell_1(y_i^A, \varphi(u_i^B))$$

For example, logistic regression $\ell_1(y, \varphi) = log(1 + exp(-y\varphi))$.

Alignment Loss

$$\underset{\Theta^A, \Theta^B}{\operatorname{argmin}} \mathcal{L}_2 = -\sum_i^{N_{AB}} \ell_2(u_i^A, u_i^B)$$

For example $-u_i^A (u_i^B)'$ or $||u_i^A - u_i^B||_F^2$.

Without loss of generality $\ell_2(u_i^A, u_i^B) = \ell_2^A(u_i^A) + \ell_2^B(u_i^B) + \kappa u_i^A (u_i^B)'$

Taylor Expansion

$$\ell_1(y, \varphi) \approx \ell_1(y, 0) + \frac{1}{2} C(y)\varphi + \frac{1}{8} D(y)\varphi^2$$

$$C(y) = \tfrac{\partial \ell_1}{\partial \varphi}|_{\varphi=0},\ D(y) = \tfrac{\partial^2 \ell_1}{\partial \varphi^2}|_{\varphi=0}$$

$$\frac{\partial \ell}{\partial \varphi} = \frac{1}{2} C(y) + \frac{1}{4} D(y)\varphi$$

Total Loss

$$\underset{\theta^A, \theta^B}{\operatorname{argmin}} \mathcal{L} = \mathcal{L}_1 + \gamma \mathcal{L}_2 + \frac{\lambda}{2}(\mathcal{L}_3^A + \mathcal{L}_3^B)$$

$$\mathcal{L}_3^B = \sum_l^{L_B} ||\theta_l^B||_F^2$$

WeBank

Vertical Federated Transfer Learning

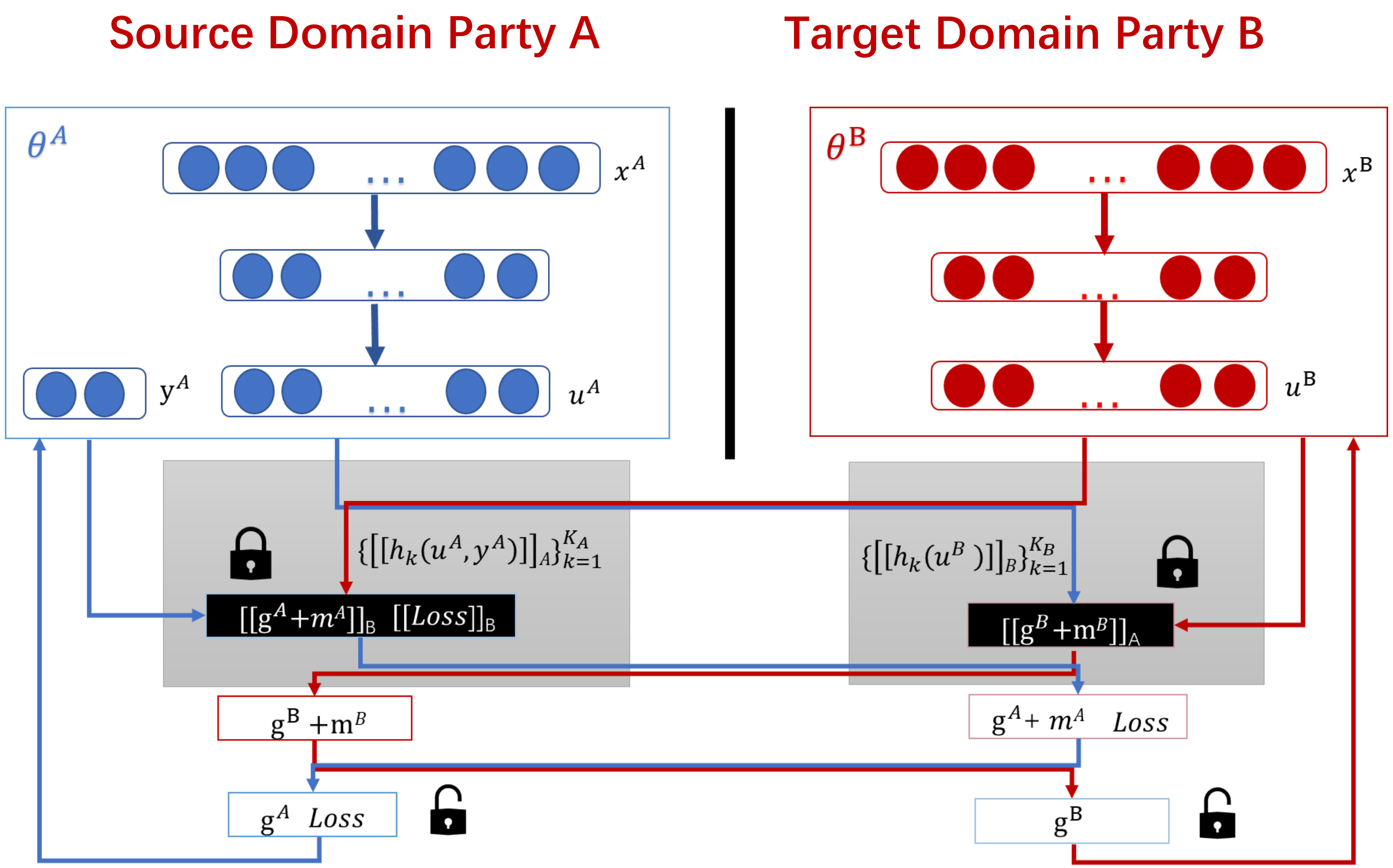
$$\begin{aligned}
[[\mathcal{L}]] = {} & \sum_i^{N_c}([[\ell_1(y_i^A,0)]] + \frac{1}{2}C(y_i^A)\Phi^A[[\mathcal{G}(u_i^B)]] \\
& + \frac{1}{8}D(y_i^A)\Phi^A[[(\mathcal{G}(u_i^B))'\mathcal{G}(u_i^B)]](\Phi^A)') \\
& + \gamma\sum_i^{N_{AB}}([[\ell_2^B(u_i^B)]] + [[\ell_2^A(u_i^A)]] + \kappa u_i^A[[(u_i^B)']]) \\
& + [[\frac{\lambda}{2}\mathcal{L}_3^A]] + [[\frac{\lambda}{2}\mathcal{L}_3^B]]
\end{aligned}$$

$$\begin{aligned}
[[\frac{\partial\mathcal{L}}{\partial\theta_l^B}]] = {} & \sum_i^{N_c}\frac{\partial(\mathcal{G}(u_i^B))'\mathcal{G}(u_i^B)}{\partial u_i^B}[[(\frac{1}{8}D(y_i^A)(\Phi^A)'\Phi^A]]\frac{\partial u_i^B}{\partial\theta_l^B} \\
& + \sum_i^{N_c}[[\frac{1}{2}C(y_i^A)\Phi^A]]\frac{\partial\mathcal{G}(u_i^B)}{\partial u_i^B}\frac{\partial u_i^B}{\partial\theta_l^B} \\
& + \sum_i^{N_{AB}}([[\gamma\kappa u_i^A]]\frac{\partial u_i^B}{\partial\theta_l^B} + [[\gamma\frac{\partial\ell_2^B(u_i^B)}{\partial\theta_l^B}]]) + [[\lambda\theta_l^B]]
\end{aligned}$$

$$\begin{aligned}
[[\frac{\partial\mathcal{L}}{\partial\theta_l^A}]] = {} & \sum_j^{N_A}\sum_i^{N_c}(\frac{1}{4}D(y_i^A)\Phi^A[[\mathcal{G}(u_i^B)'\mathcal{G}(u_i^B)]] \\
& + \frac{1}{2}C(y_i^A)[[\mathcal{G}(u_i^B)]])\frac{\partial\Phi^A}{\partial u_j^A}\frac{\partial u_j^A}{\partial\theta_l^A} \\
& + \gamma\sum_i^{N_{AB}}([[\kappa u_i^B]]\frac{\partial u_i^A}{\partial\theta_l^A} + [[\frac{\partial\ell_2^A(u_i^A)}{\partial\theta_l^A}]]) + [[\lambda\theta_l^A]]
\end{aligned}$$

Security Proof based on the inability of solving *n* equations with more than *n* unknowns.

系统原理 Architecture



Step 1

Party A and B send public keys to each other

Step 2

Parties compute, encrypt and exchange intermediate results

Step 3

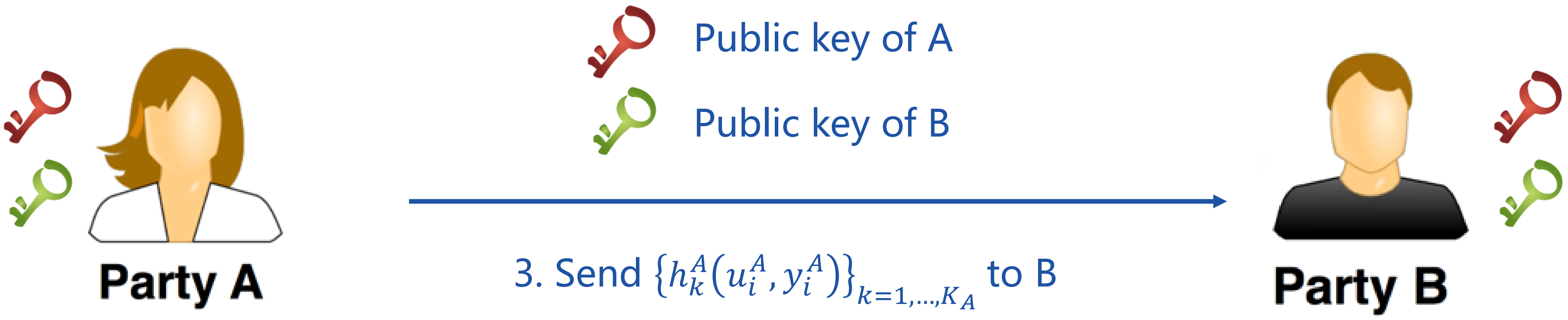
Parties compute encrypted gradients, add masks and send to each other

Step 4

Parties decrypt gradients and exchange, unmask and update model locally

WeBank

模型训练 (I)



Public key of A

Public key of B

Party A

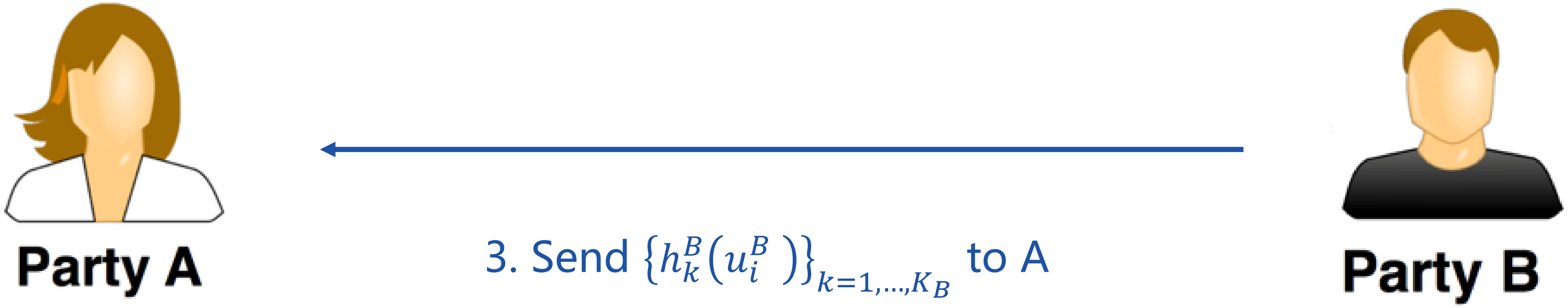
3. Send $\{h_k^A(u_i^A, y_i^A)\}_{k=1,\ldots,K_A}$ to B

Party B

1. compute $u_i^A \leftarrow Net^A(\Theta_A, x_i^A)$

2. compute and encrypt:$\{h_k^A(u_i^A, y_i^A)\}_{k=1,\ldots,K_A}$

模型训练 (II)



Party A

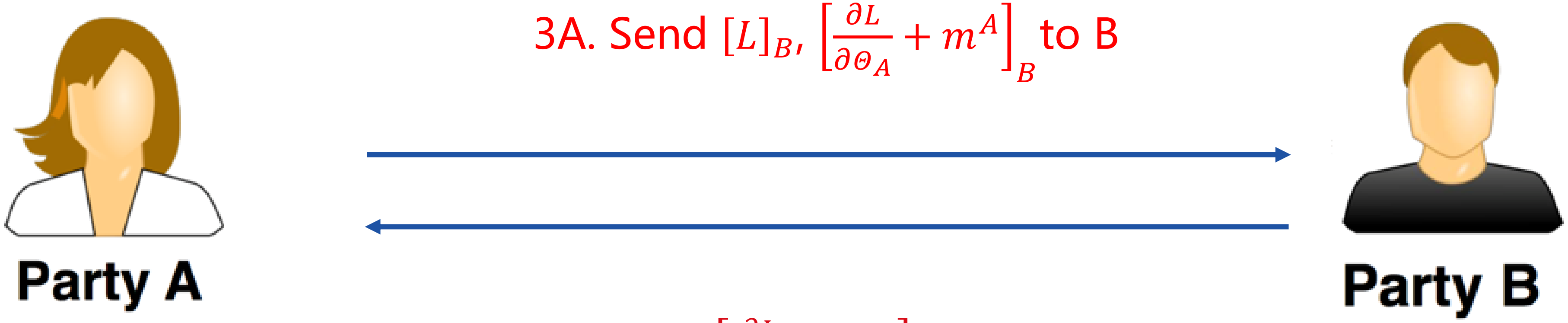
Party B

3. Send $\{h_k^B(u_i^B\ )\}_{k=1,\dots,K_B}$ to A

1. compute$u_i^B \leftarrow Net^B(\Theta_B, x_i^B)$

2. compute and encrypt:$\{h_k^B(u_i^B\ )\}_{k=1,\dots,K_B}$

模型训练 (III)



3A. Send $[L]_B$, $\left[\frac{\partial L}{\partial \Theta_A}+m^A\right]_B$ to B

3B. Send $\left[\frac{\partial L}{\partial \Theta_B}+m^B\right]_A$ to A

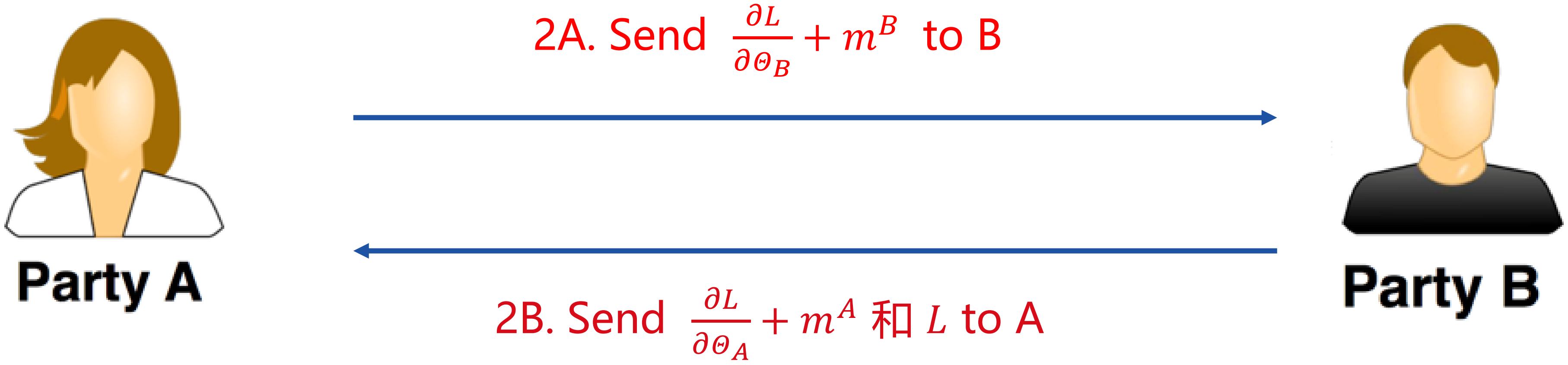
1A. Create random mask m^B

1B. create random mask m^B

2A. Computes and encrypts: $[L]_B$, $\left[\frac{\partial L}{\partial \Theta_A}+m^A\right]_B$

2B. Computes and encrypts:$\left[\frac{\partial L}{\partial \Theta_B}+m^B\right]_A$

模型训练 (IV)



Party A

Party B

2A. Send $\frac{\partial L}{\partial \Theta_B} + m^B$ to B

2B. Send $\frac{\partial L}{\partial \Theta_A} + m^A$ 和 L to A

1A. decrypts : $\left[\frac{\partial L}{\partial \Theta_B} + m^B\right]_A \rightarrow \frac{\partial L}{\partial \Theta_B} + m^B$

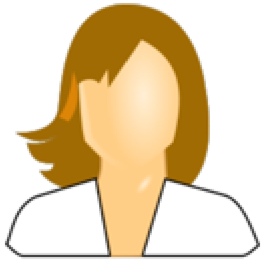
3A. Remove mask:$\frac{\partial L}{\partial \Theta_A} + m^A \rightarrow \frac{\partial L}{\partial \Theta_A}$

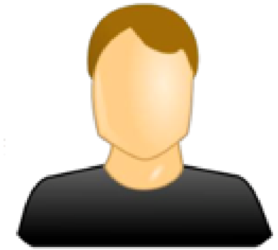
1B. decrypts : $\left[\frac{\partial L}{\partial \Theta_A} + m^A\right]_B \rightarrow \frac{\partial L}{\partial \Theta_A} + m^A$

$$[L]_B \rightarrow L$$

3B. Remove mask:$\frac{\partial L}{\partial \Theta_B} + m^B \rightarrow \frac{\partial L}{\partial \Theta_B}$

模型训练 (V)





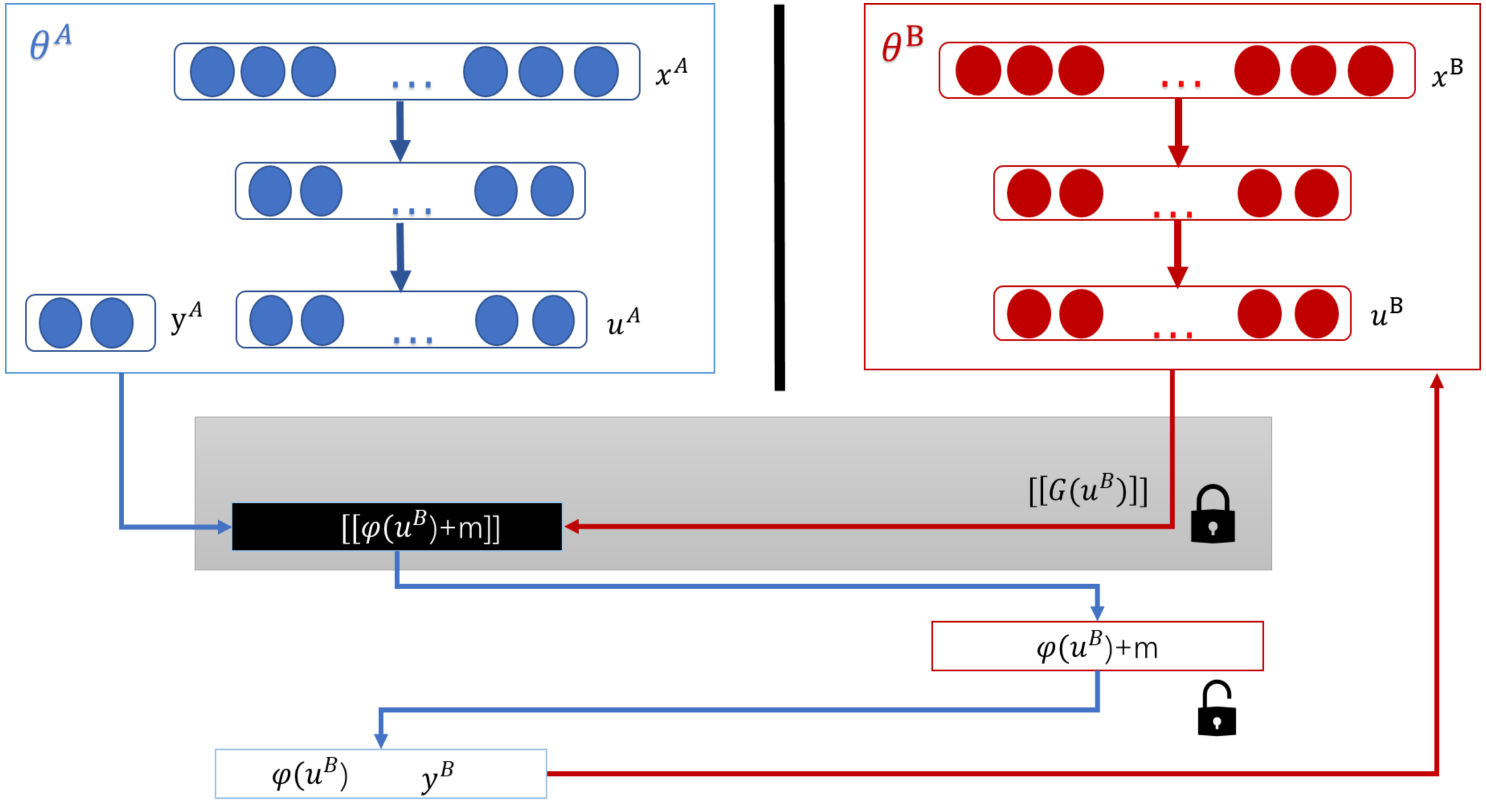
updates： $\mathit{\Theta}_A = \mathit{\Theta}_A - \gamma \frac{\partial L}{\partial \mathit{\Theta}_A}$

updates： $\mathit{\Theta}_B = \mathit{\Theta}_B - \gamma \frac{\partial L}{\partial \mathit{\Theta}_B}$

At the end of the training process, each party remains oblivious to the data structure of the other party, and only obtains the model parameters associated with its own features.

模型推断

- Suppose a new user ID arrives at Party B



Step 1

Party B computes and sends intermediate results to A

Step 2

Party A computes encrypted result, adds mask and send back to B

Step 3

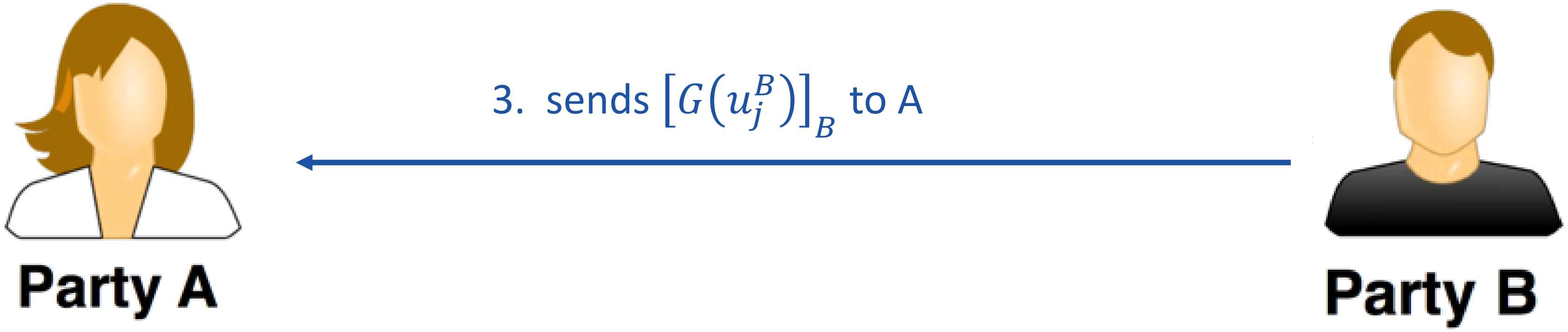
Party B decrypts masked result and sends back to A

Step 4

Party A unmasks results, predicts, and sends to B

WeBank

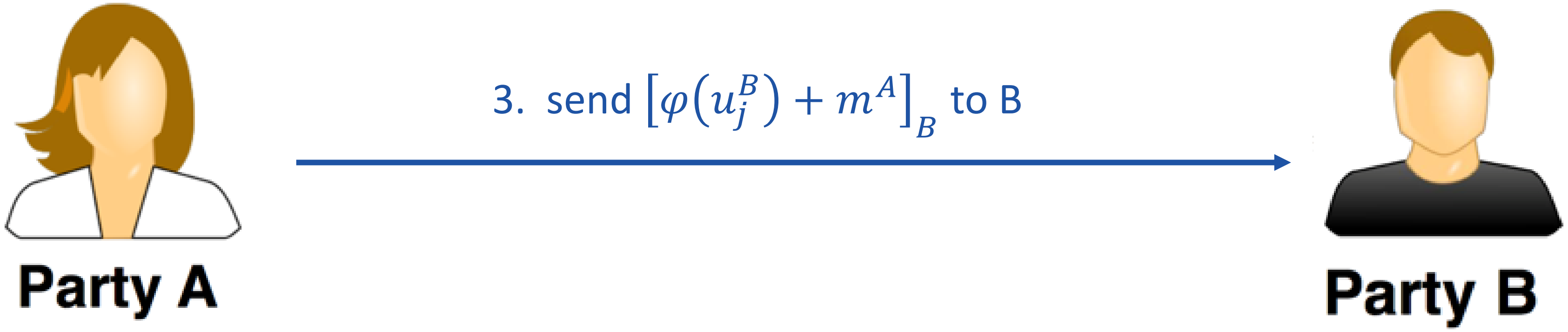
模型推断 (I)



1. Computes $u_j^B = Net^B\left(\Theta_B, x_j^B\right)$

2. Computes $G\left(u_j^B\right) = \left(u_j^B\right)^T$, and encrypts

模型推断 (II)

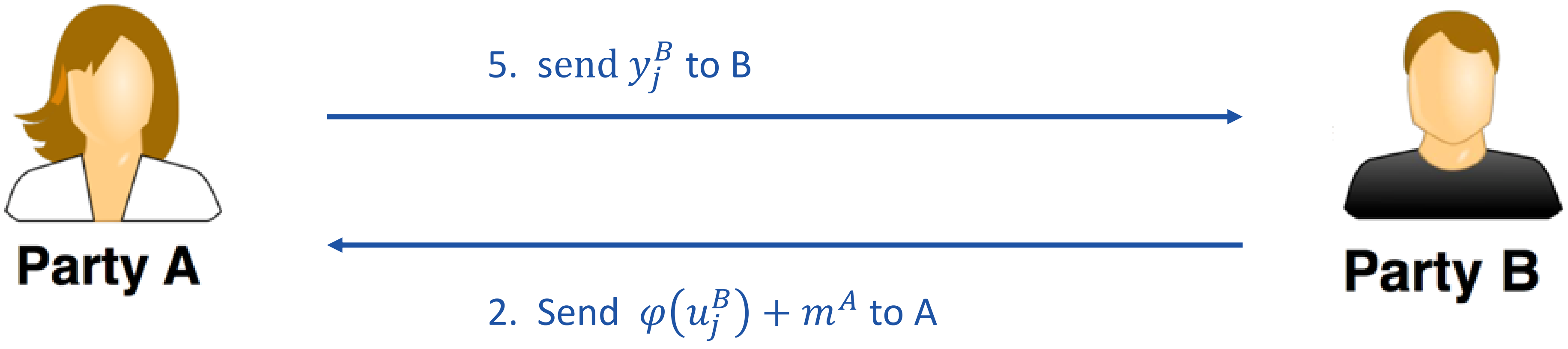


3. send $\left[\varphi\left(u_j^B\right)+m^A\right]_B$ to B

1. Create random mask m^A

2. computes $\left[\varphi\left(u_j^B\right)+m^A\right]_B$

模型推断 (III)



Party A

5. send y_j^B to B

2. Send $\varphi\left(u_j^B\right)+m^A$ to A

Party B

3. Unmask $\varphi\left(u_j^B\right)+m^A \rightarrow \varphi\left(u_j^B\right)$

4. $\varphi\left(u_j^B\right) \rightarrow y_j^B$

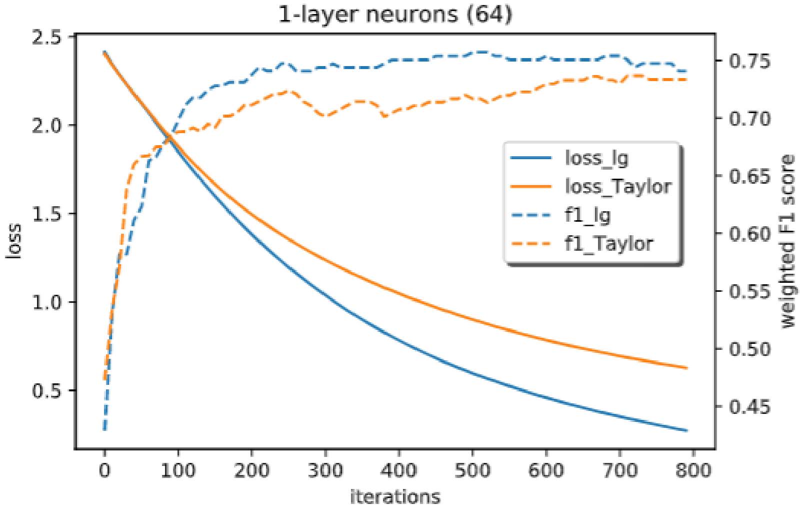
1. decrypts: $\left[\varphi\left(u_j^B\right)+m^A\right]_B \rightarrow \varphi\left(u_j^B\right)+m^A$

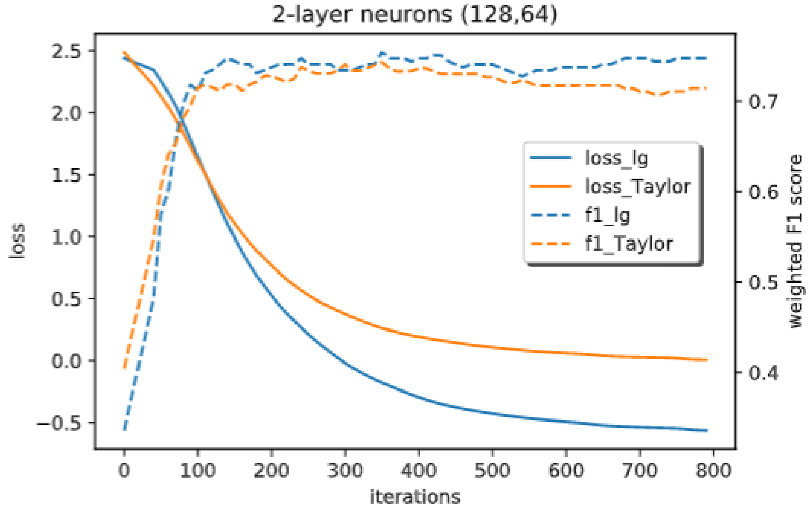
优势 Advantages

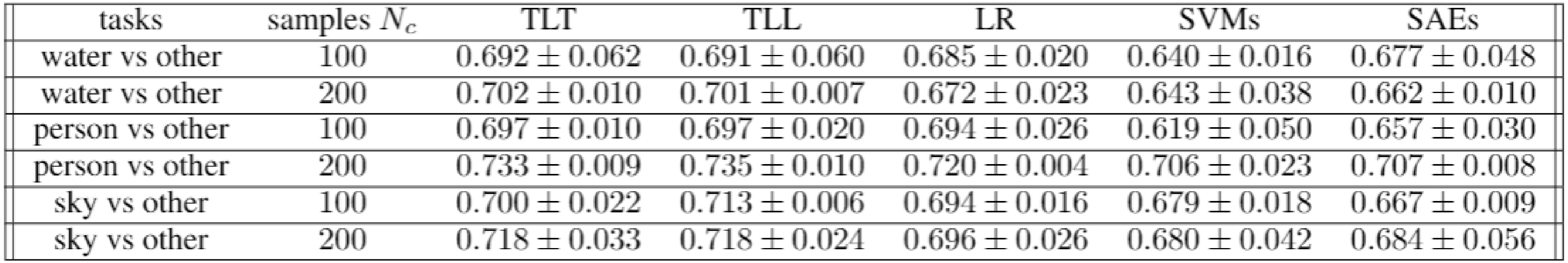
- 没有泄露原始数据 No exposure of raw data
- 没有泄露原始数据的加密形式 No exposure of encrypted raw data
- 没有第三方 No Third Party
- 模型几乎无损失 Almost lossless accuracy

效果评估 Performance

The **NUSWIDE** data set : low-level features from Flickr images, their associate tags and ground truth labels


1-layer neurons (64)


2-layer neurons (128,64)

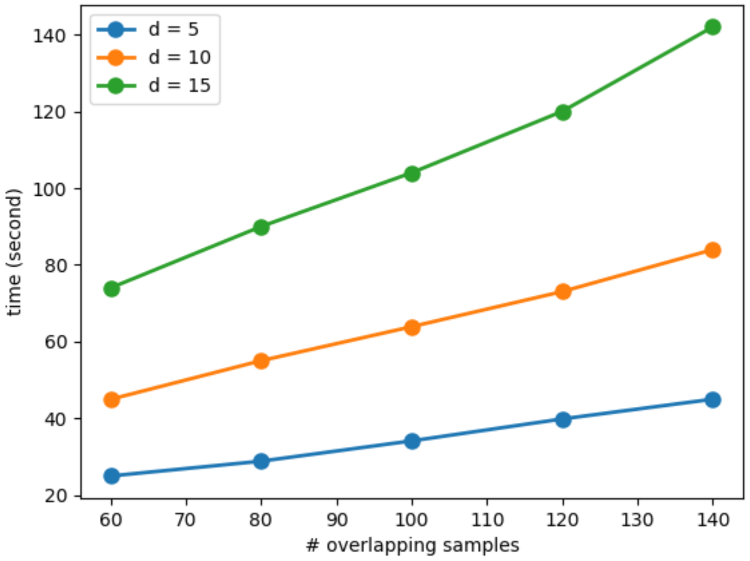


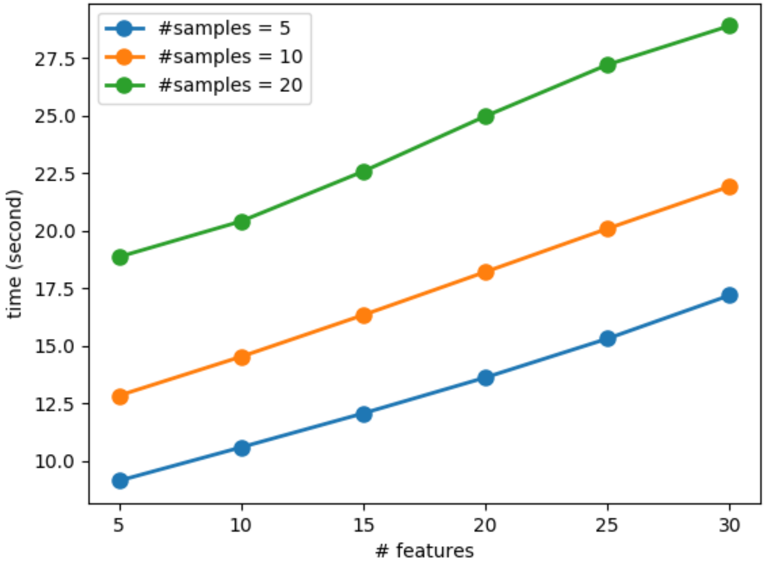
tasks	samples N_c	TLT	TLL	LR	SVMs	SAEs
water vs other	100	0.692 ± 0.062	0.691 ± 0.060	0.685 ± 0.020	0.640 ± 0.016	0.677 ± 0.048
water vs other	200	0.702 ± 0.010	0.701 ± 0.007	0.672 ± 0.023	0.643 ± 0.038	0.662 ± 0.010
person vs other	100	0.697 ± 0.010	0.697 ± 0.020	0.694 ± 0.026	0.619 ± 0.050	0.657 ± 0.030
person vs other	200	0.733 ± 0.009	0.735 ± 0.010	0.720 ± 0.004	0.706 ± 0.023	0.707 ± 0.008
sky vs other	100	0.700 ± 0.022	0.713 ± 0.006	0.694 ± 0.016	0.679 ± 0.018	0.667 ± 0.009
sky vs other	200	0.718 ± 0.033	0.718 ± 0.024	0.696 ± 0.026	0.680 ± 0.042	0.684 ± 0.056

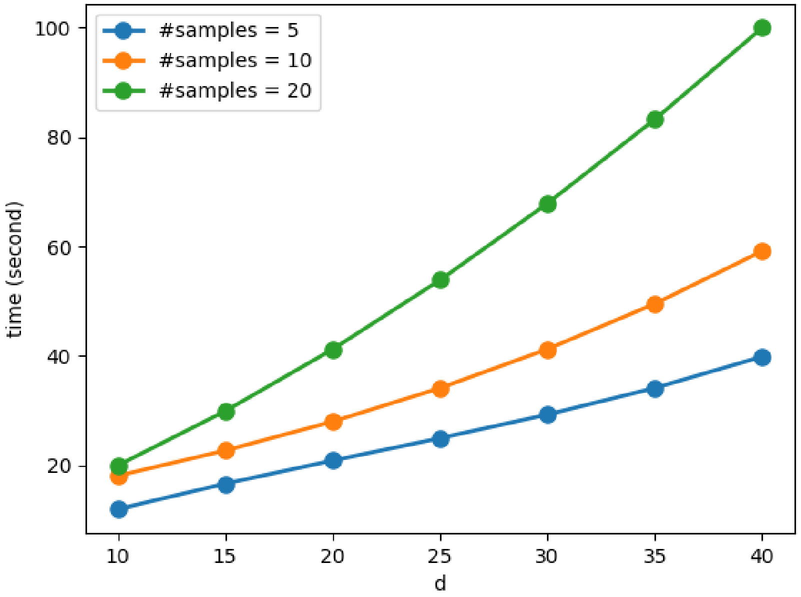
Table 1: Comparison of weighted F1 score of transfer learning with Taylor loss (TLT), with logistic loss (TLL) and self-learning with logistic regression (LR), with support vector machines(SVMs), and with stacked auto encoders(SAEs).

[NUSWIDE dataset] Chua, T.-S.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; and Zheng, Y. 2009. NUS-WIDE: A real-world web image database from national university of Singapore. In In CIVR.

延伸性 Scalability

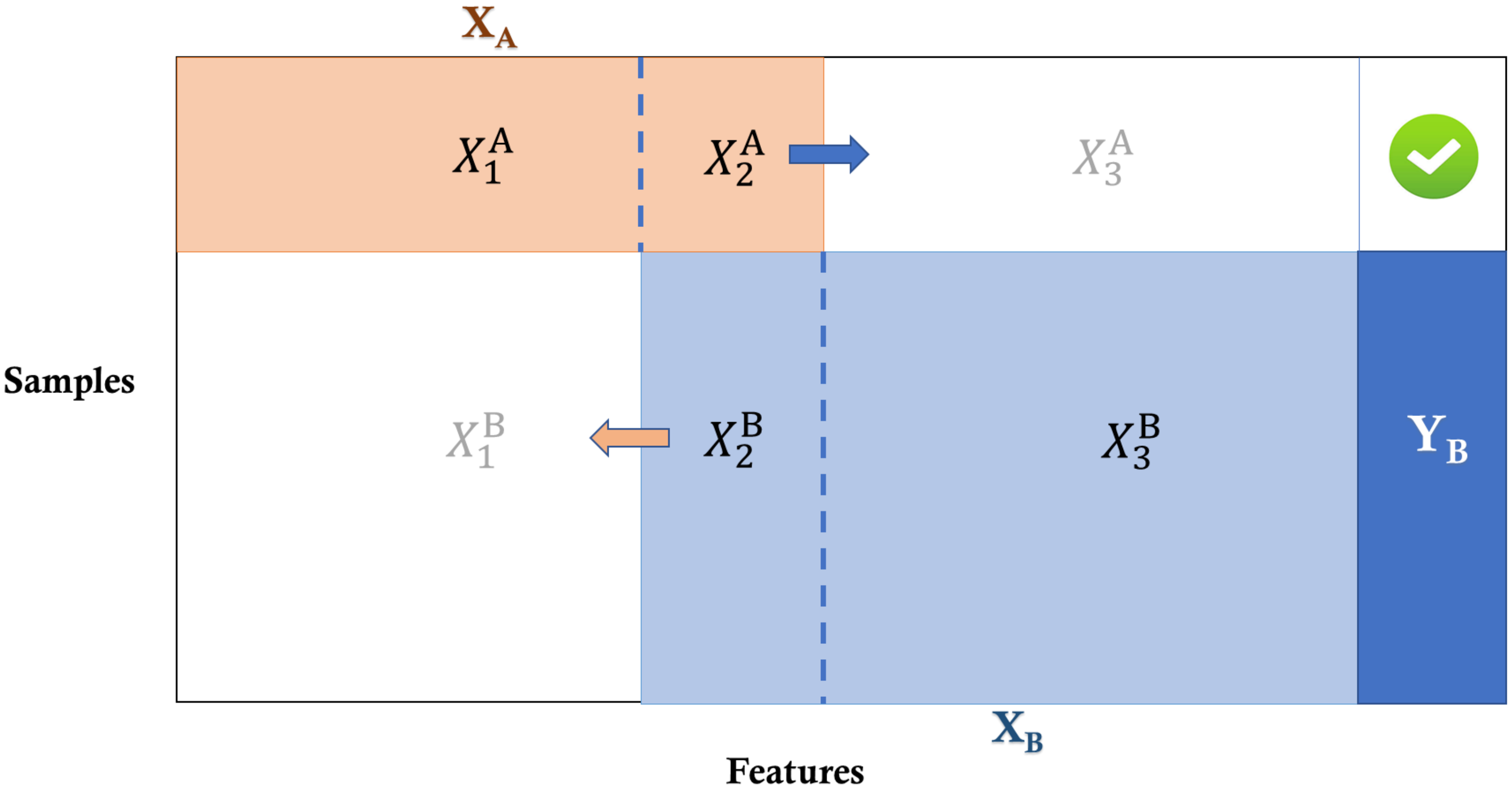






- 加密数据传输和加密运算是最大影响因素;
- 数据传输与数据量成正比;
- 加密运算代价与模型复杂度（参数量）成正比；

假如零样本重叠怎么办？



应用：

- 不用城市的用户行为数据分布
- 时间序列数据样本

.....

Feature-based Heterogeneous FTL（HFTL）

Step 1 ：Private transfer learning

$$\langle X^{k,m}\rangle = f^{m}_{c,\bar{c}}(\langle\theta^{m}\rangle, X^{k}_{c})$$

Step 2 : Private federated learning

Party	Feature space			Label
S	$\overline{X}^{S}_{c}$	X^{S}_{c}	$\langle X^{S,T}\rangle$	Y^{S}
T	$\langle X^{T,S}\rangle$	X^{T}_{c}	$\overline{X}^{T}_{c}$	Y^{T}

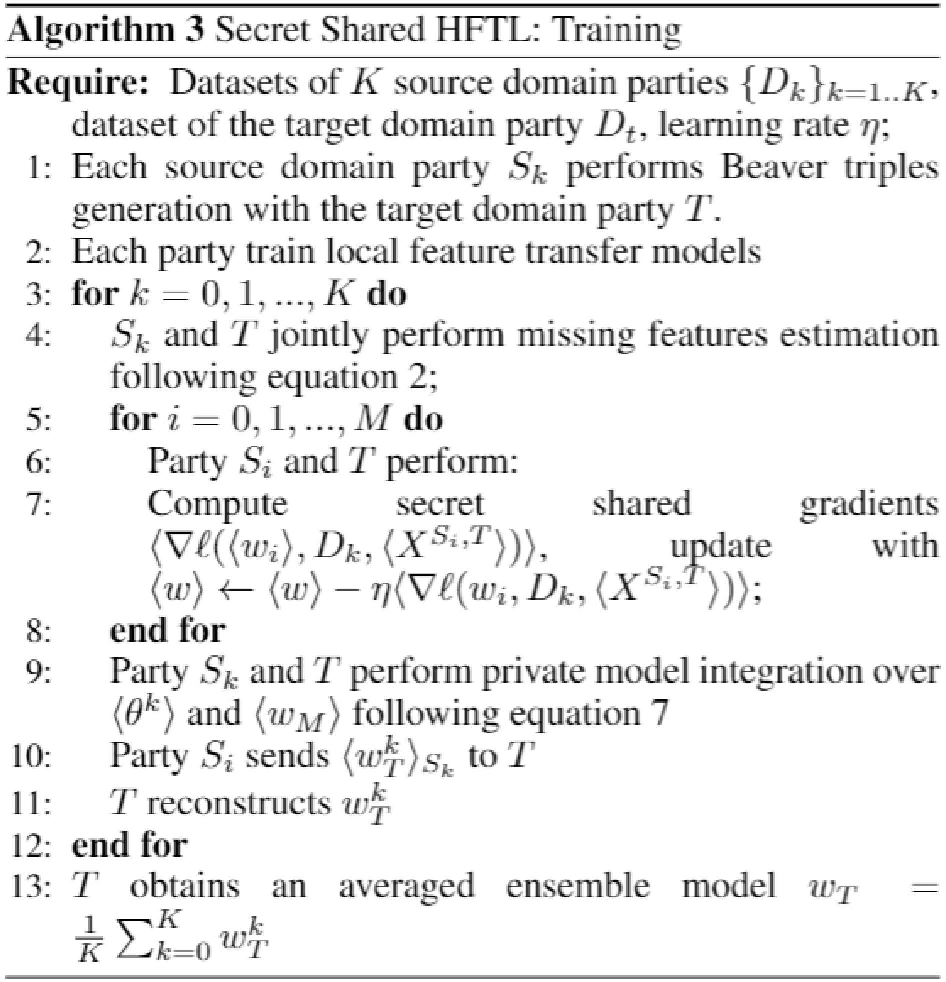
$$\langle w\rangle \leftarrow \langle w\rangle - \eta\langle\nabla\ell(w_i, D_k, \langle X^{S_i,T}\rangle)\rangle$$

Step 3 : Private model integration

$$\langle w^{k}_{T}\rangle = [(\langle\theta^{k}\rangle\cdot\langle w_{S_k}\rangle + \langle w_c\rangle) \quad \langle w_t\rangle]^{T}$$

Step 4: Private model inference

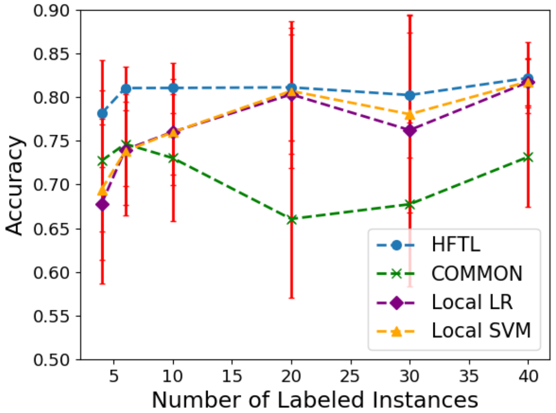
$$y = \sum_{i=1}^{K} f(w^{k}_{T}, X)$$



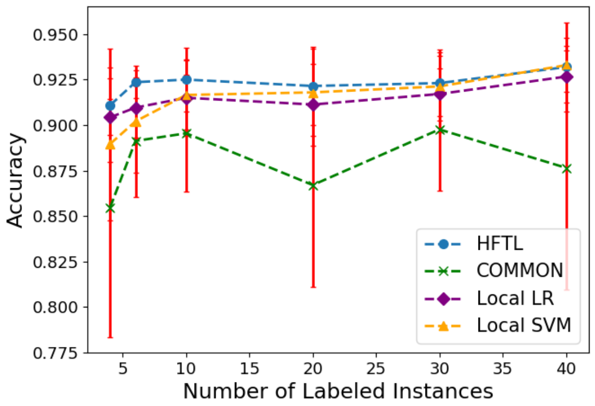
Algorithm 3 Secret Shared HFTL: Training

```
Require: Datasets of $K$ source domain parties $\{D_k\}_{k=1..K}$,
    dataset of the target domain party $D_t$, learning rate $\eta$;
Each source domain party $S_k$ performs Beaver triples
    generation with the target domain party $T$.
Each party train local feature transfer models
for $k = 0, 1, ..., K$ do
    $S_k$ and $T$ jointly perform missing features estimation
    following equation 2;
    for $i = 0, 1, ..., M$ do
        Party $S_i$ and $T$ perform:
        Compute secret shared gradients
        $\langle\nabla\ell(\langle w_i\rangle, D_k, \langle X^{S_i,T}\rangle)\rangle$, update with
        $\langle w\rangle \leftarrow \langle w\rangle - \eta\langle\nabla\ell(w_i, D_k, \langle X^{S_i,T}\rangle)\rangle$;
    end for
    Party $S_k$ and $T$ perform private model integration over
    $\langle\theta^k\rangle$ and $\langle w_M\rangle$ following equation 7
    Party $S_i$ sends $\langle w^k_T\rangle_{S_k}$ to $T$
    $T$ reconstructs $w^k_T$
end for
$T$ obtains an averaged ensemble model $w_T = \frac{1}{K}\sum_{k=0}^{K} w^k_T$
```

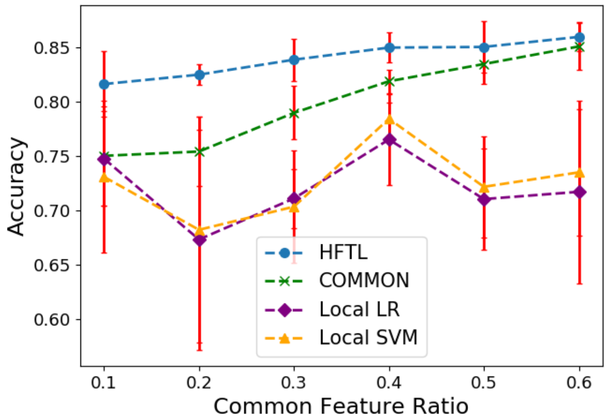
结果比较



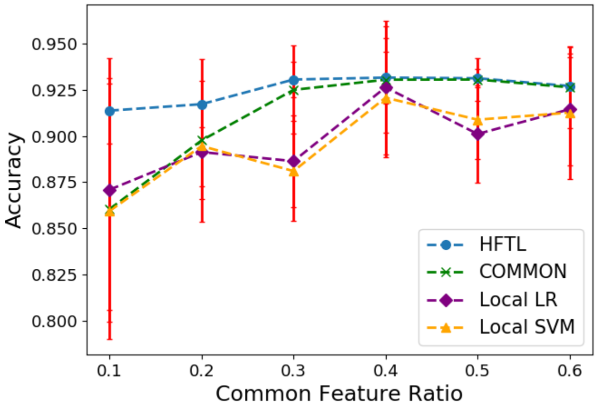
Spambase



WDBC



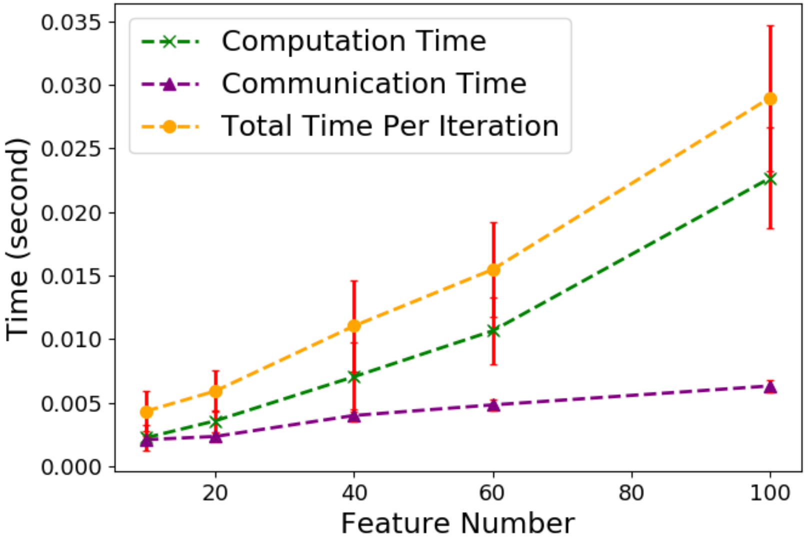
Spambase



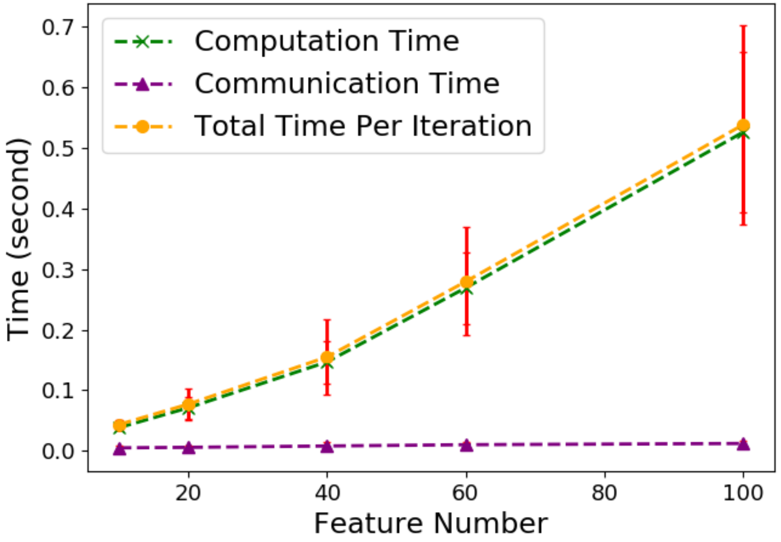
WDBC

效率比较

Secret Sharing



HE



联邦学习（Federated Machine Learning）的挑战

模型攻击[BVH+18]

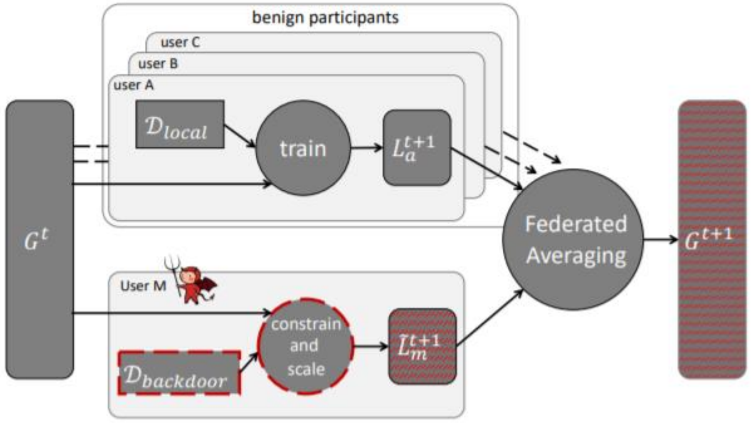
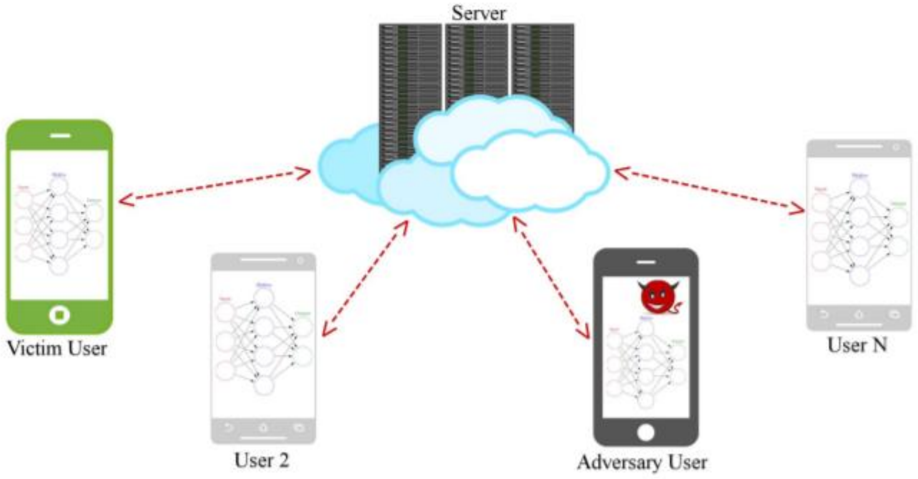


Fig. 1: **Overview of the attack.** The attacker compromises one or more of the participants, trains a model on the backdoor data using our new constrain-and-scale technique, and submits the resulting model. After federated averaging, the global model is replaced by the attacker's backdoored model.

Eugene B et al. 2018. ***How To Backdoor Federated Learning.*** arXiv:cs.CR/1807.00459

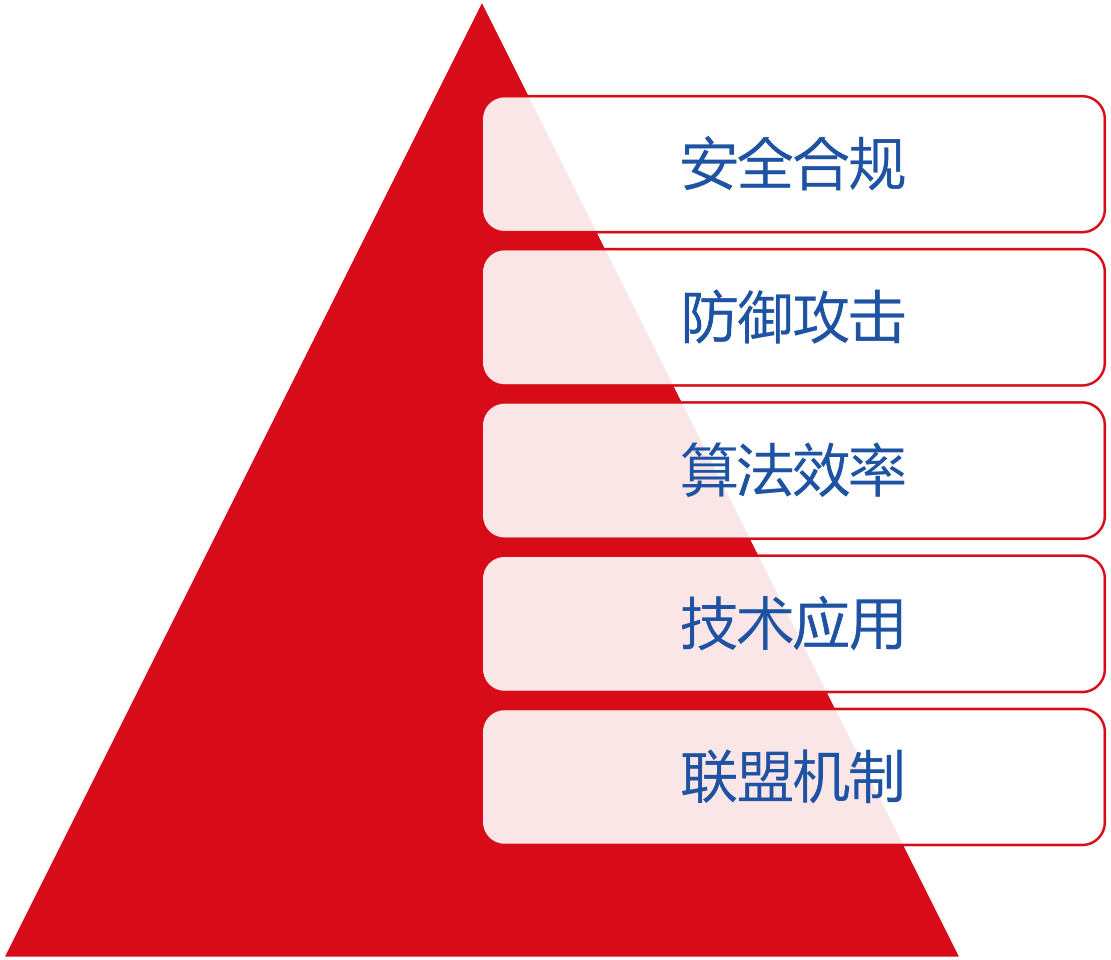
数据攻击[HAP17]



(b) Collaborative Learning

Briland H et al. 2017. ***Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning***

联邦学习（Federated Machine Learning）的研究展望



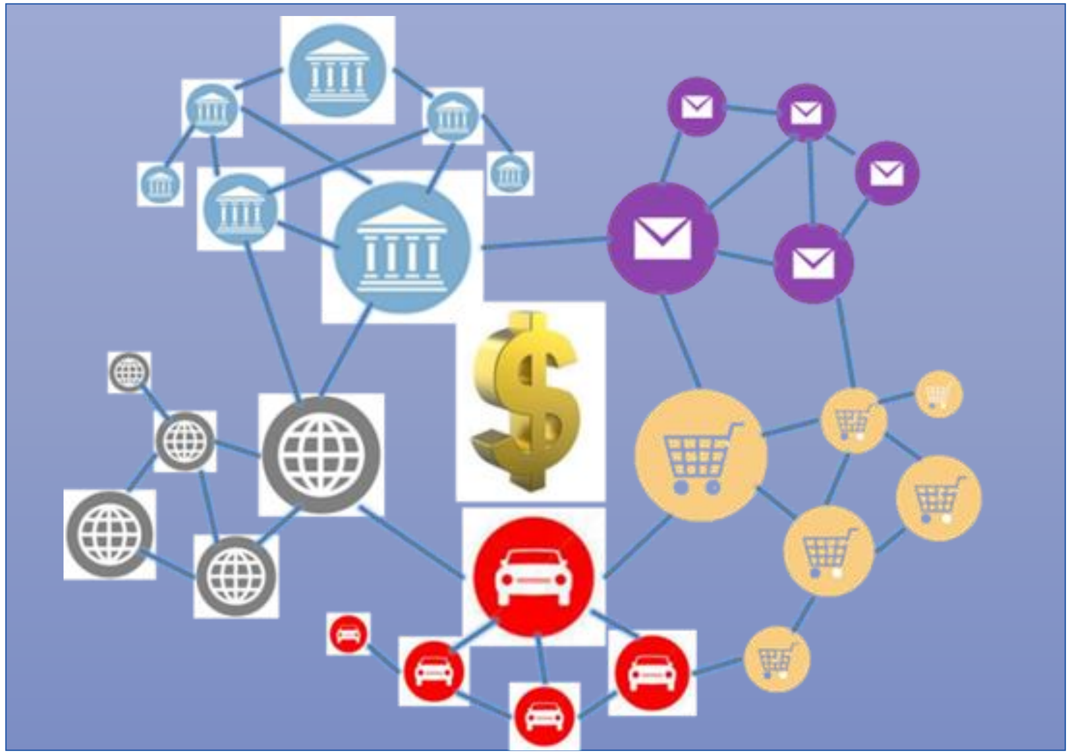
安全合规

防御攻击

算法效率

技术应用

联盟机制



References

- H. Brendan McMahan et al, Communication-Efficient Learning of Deep Networks from Decentralized Data, Google, 2017
- Reza Shokri and Vitaly Shmatikov. 2015. Privacy-Preserving Deep Learning. In Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security (CCS ’15). ACM, New York, NY, USA, 1310–1321.
- 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
- Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: concepts and applications, ACM TIST, ,2018
- Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., ... & Seth, K. (2017, October). Practical secure aggregation for privacy-preserving machine learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (pp. 1175-1191). ACM
- Ran Cohen ,Tel Aviv University, Secure Multiparty Computation: Introduction
- Du, W., Han, Y. S., & Chen, S. (2004, April). Privacy-preserving multivariate statistical analysis: Linear regression and classification. In *Proceedings of the 2004 SIAM international conference on data mining* (pp. 222-233). Society for Industrial and Applied Mathematics.
- Han, S., Ng, W. K., Wan, L., & Lee, V. C. (2010). Privacy-preserving gradient-descent methods. *IEEE Transactions on Knowledge and Data Engineering*, *22*(6), 884-899.
- Mohassel, P., & Zhang, Y. (2017, May). SecureML: A system for scalable privacy-preserving machine learning. In *2017 38th IEEE Symposium on Security and Privacy (SP)* (pp. 19-38). IEEE
- S. Wagh, D. Gupta, and N. Chandran, “Securenn: Efficient and private neural network training,” 2018, iACR ePrint Archive, https://eprint.iacr.org/2018/442
- Kim, M.; Song, Y.; Wang, S.; Xia, Y.; and Jiang, X. 2018. Secure logistic regression based on homomorphic encryption: Design and evaluation. JMIR Med Inform 6(2)

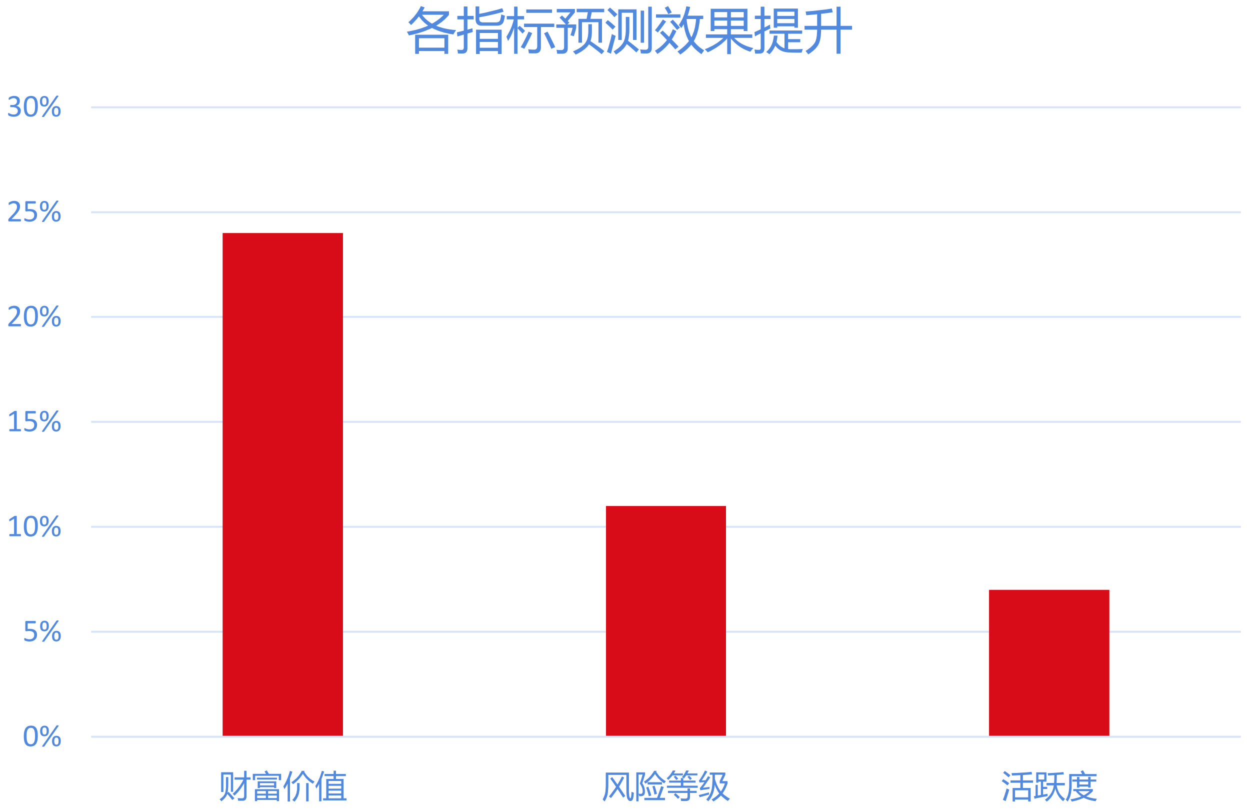
References

- Y. Aono, T. Hayashi, T. P. Le, L. Wang, Scalable and secure logistic regression via homomorphic encryption, CODASPY16
- Hardy, S., Henecka, W., Ivey-Law, H., Nock, R., Patrini, G., Smith, G., & Thorne, B. (2017). Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. arXiv preprint arXiv:1711.10677
- Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan. “GAZELLE: A Low Latency Framework for Secure Neural Network Inference”. In: IACR Cryptology ePrint Archive 2018 (2018), p. 73 (pp. 4–6).
- E Hesamifard et al, “CryptoDL: Deep Neural Networks over Encrypted Data”,2017
- Robin C. Geyer, Tassilo Klein, and Moin Nabi. 2017. Differentially Private Federated Learning: A Client Level Perspective. CoRR abs/1712.07557 (2017)
- Yang Wang , Quanquan Gu , and Donald Brown, Differentially Private Hypothesis Transfer Learning, 2018
- Virginia Smith et al. 2017. Federated Multi-Task Learning. In Advances in Neural Information Processing Systems
- S. Ruder ，Transfer Learning - Machine Learning's Next Frontier
- Yang Wang , Quanquan Gu , and Donald Brown, Differentially Private Hypothesis Transfer Learning, 2018

03

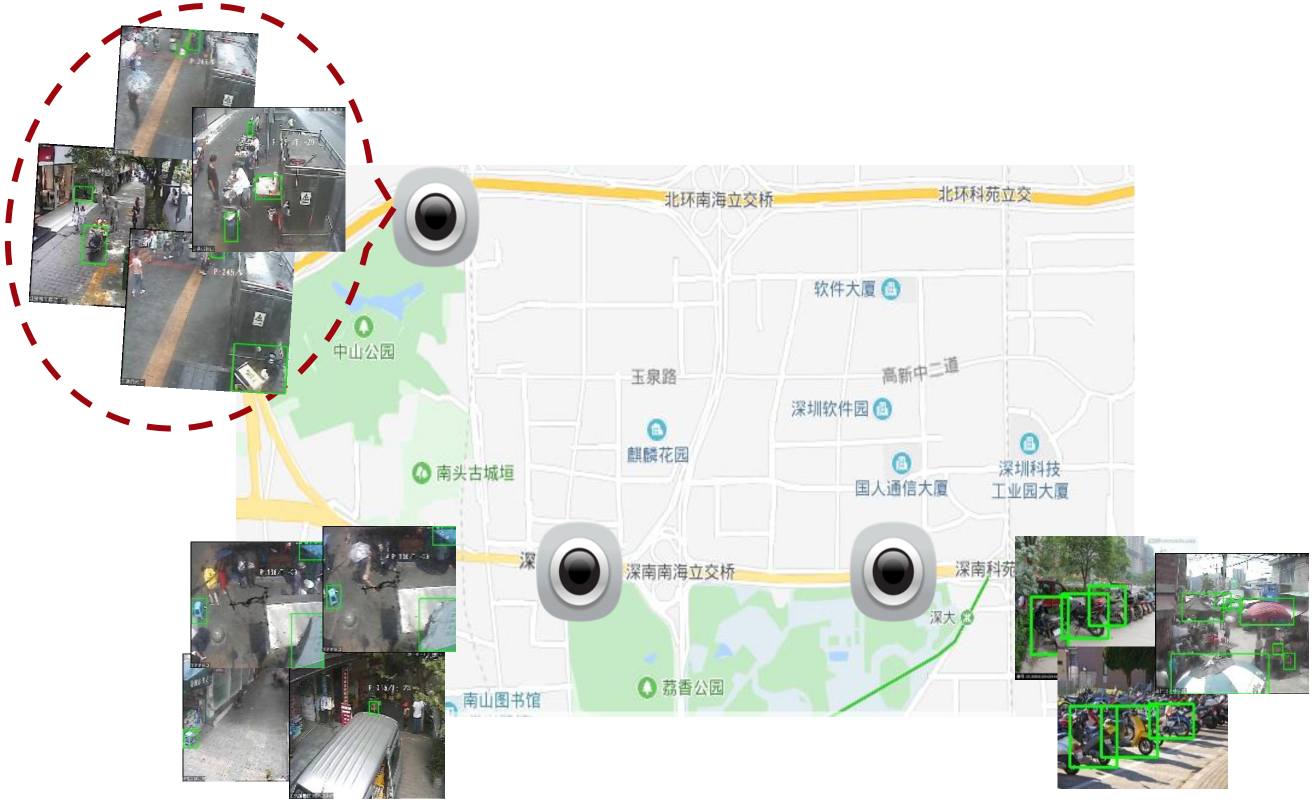
应用案例 Building Federated AI Applications

Federated Learning 在金融领域的应用



Federated Learning 视觉应用 - 城市管理

*微众 Webank AI 联合极视角 Extreme Vision 项目

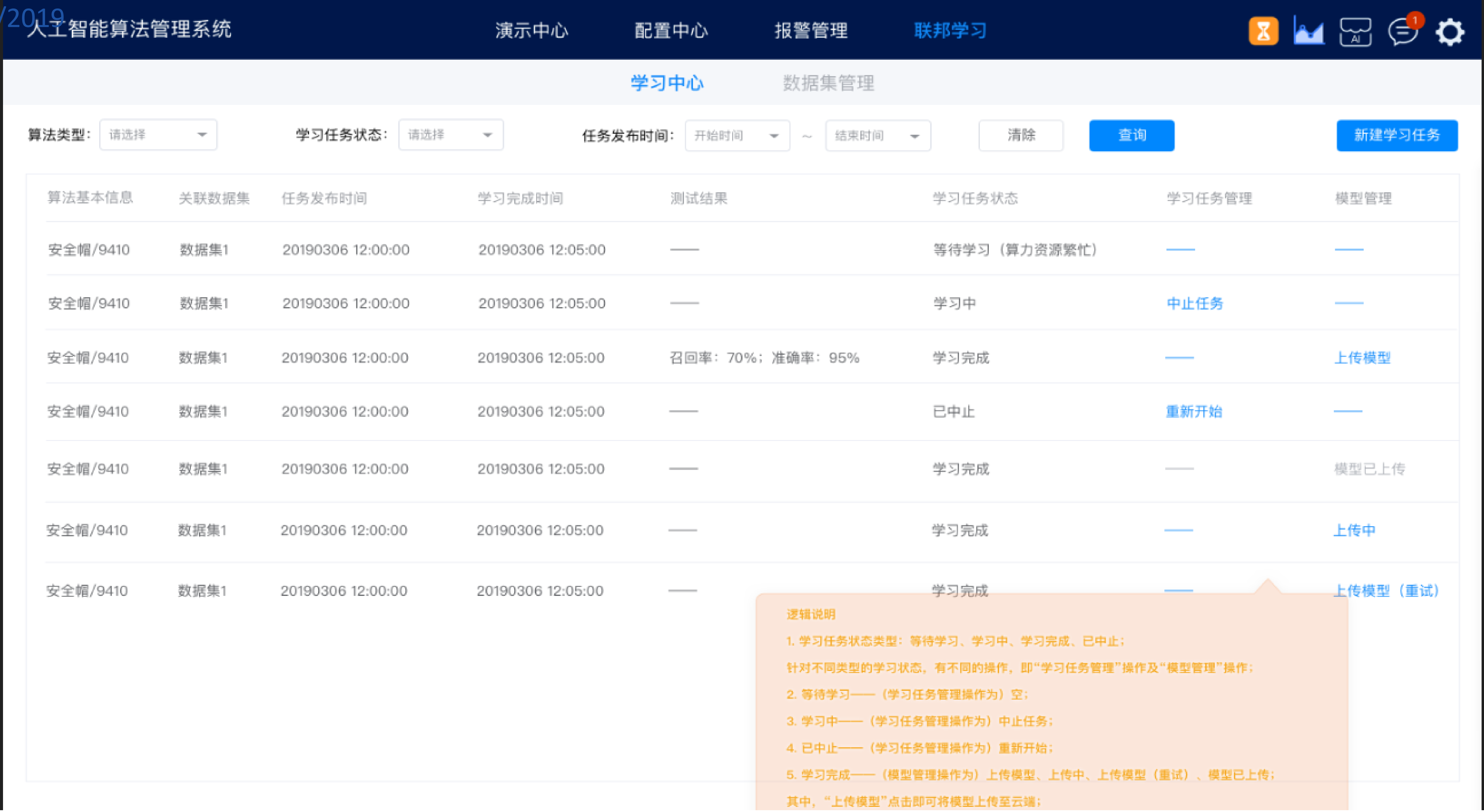


挑战

- 标签数量少
- 数据分散，集中管理成本高
- 离线延迟的模型更新和反馈

联邦学习

- 在线模型更新和反馈
- 无需集中上传数据
- 数据保护，隐私性高



算法类型：请选择

学习任务状态：请选择

任务发布时间：开始时间 ~ 结束时间

算法基本信息	关联数据集	任务发布时间	学习完成时间	测试结果	学习任务状态	学习任务管理	模型管理
安全帽/9410	数据集1	20190306 12:00:00	20190306 12:05:00	——	等待学习（算力资源繁忙）	——	——
安全帽/9410	数据集1	20190306 12:00:00	20190306 12:05:00	——	学习中	中止任务	——
安全帽/9410	数据集1	20190306 12:00:00	20190306 12:05:00	召回率：70%；准确率：95%	学习完成	——	上传模型
安全帽/9410	数据集1	20190306 12:00:00	20190306 12:05:00	——	已中止	重新开始	——
安全帽/9410	数据集1	20190306 12:00:00	20190306 12:05:00	——	学习完成	——	模型已上传
安全帽/9410	数据集1	20190306 12:00:00	20190306 12:05:00	——	学习完成	——	上传中
安全帽/9410	数据集1	20190306 12:00:00	20190306 12:05:00	——	学习完成	——	上传模型（重试）

逻辑说明

1. 学习任务状态类型：等待学习、学习中、学习完成、已中止；

针对不同类型的学习状态，有不同的操作，即“学习任务管理”操作及“模型管理”操作；

2. 等待学习——（学习任务管理操作为）空；

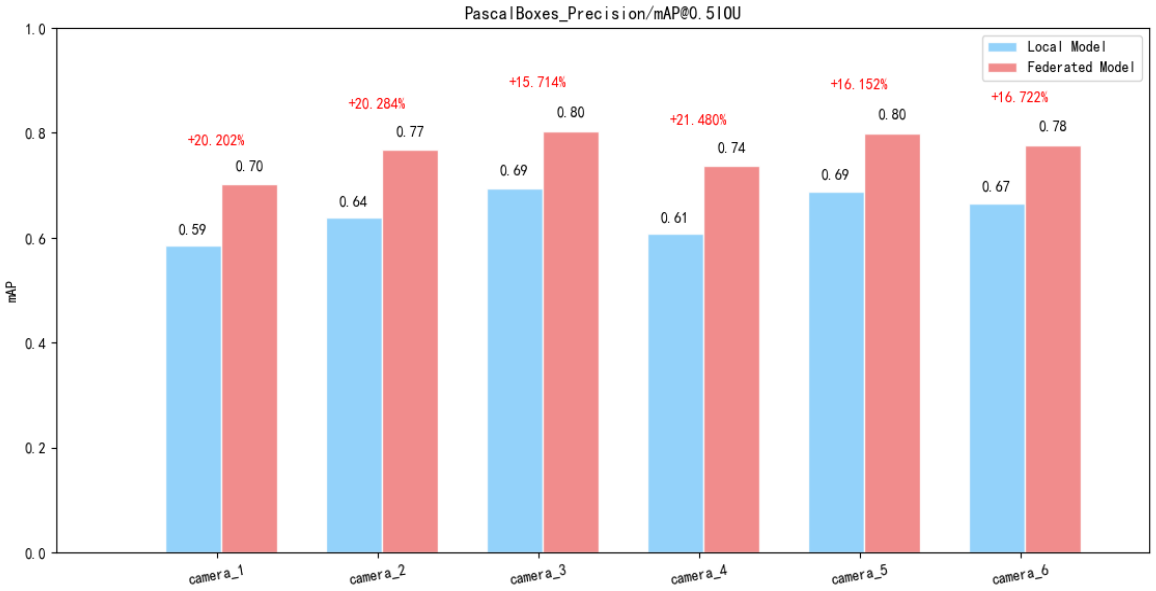
3. 学习中——（学习任务管理操作为）中止任务；

4. 已中止——（学习任务管理操作为）重新开始；

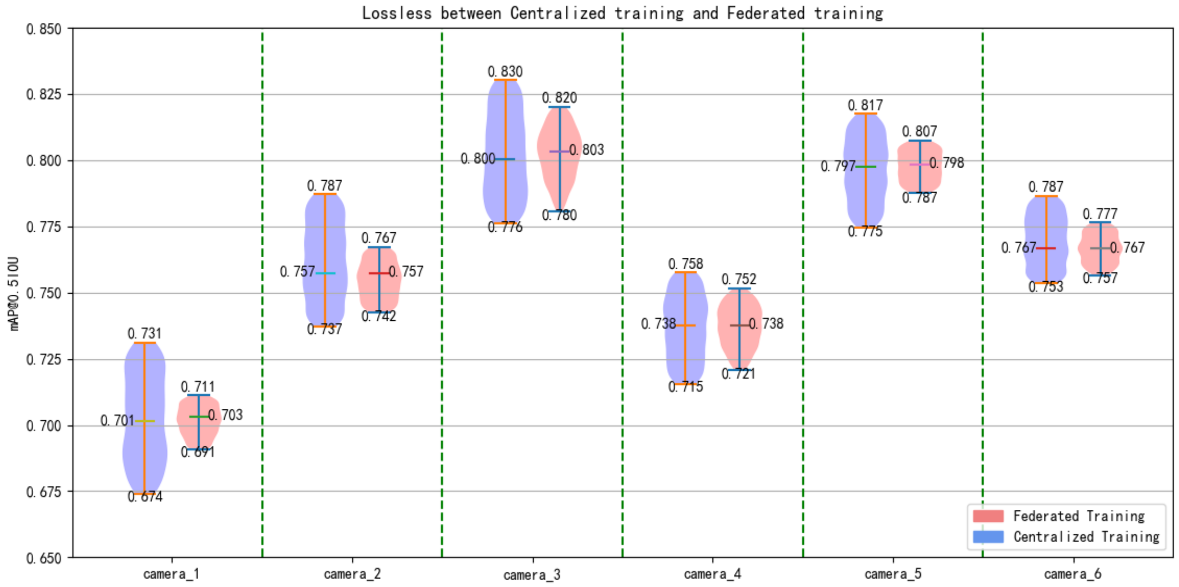
5. 学习完成——（模型管理操作为）上传模型、上传中、上传模型（重试）、模型已上传；

其中，“上传模型”点击即可将模型上传至云端；

联邦学习对模型提升率15%

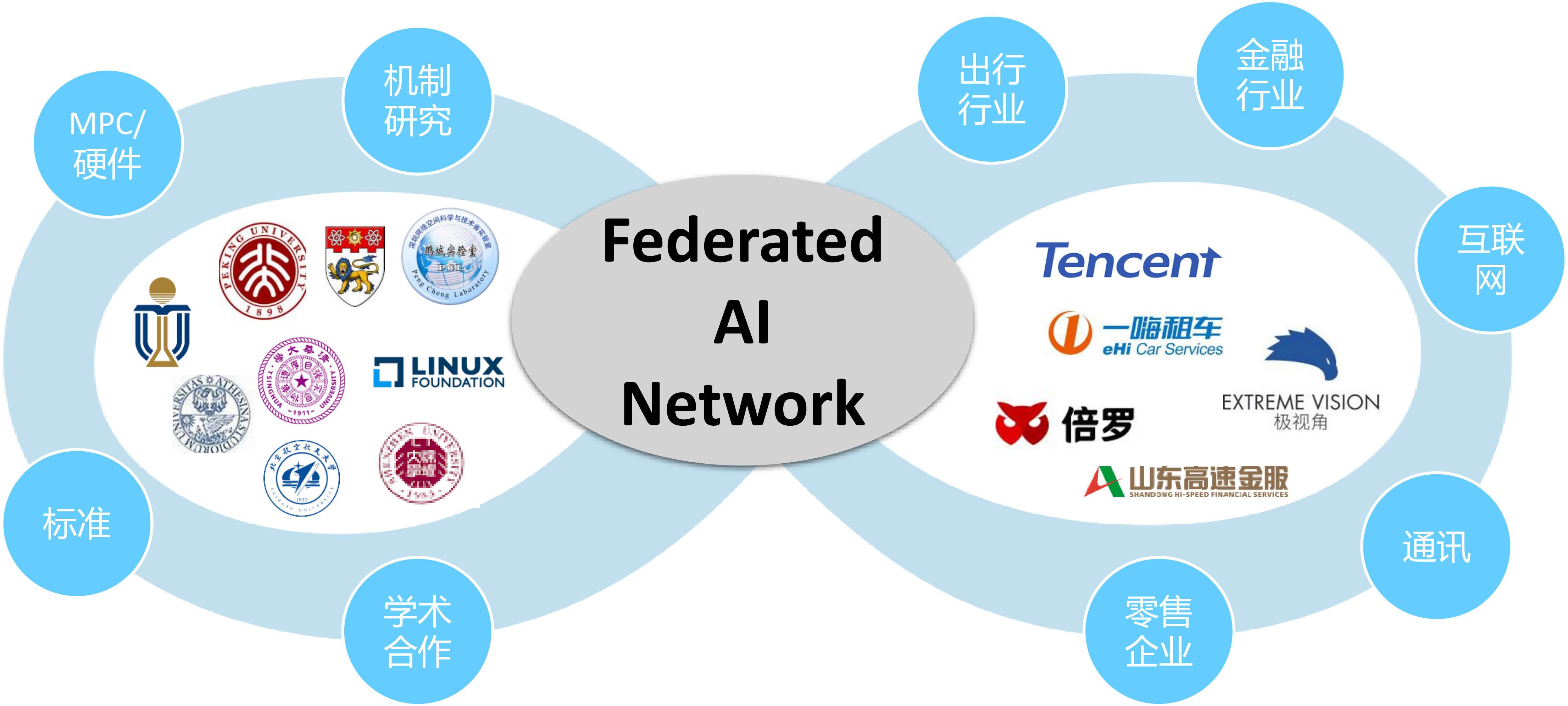


模型效果无损失
(Centralized model vs federated model)



- 7 class : {table, chair, carton, sunshade, basket, gastank, electromobile}
- 6 cameras, 1,922 images

生态建设：开源、技术标准，商业赋能



Federated AI Network

MPC/硬件

机制研究

标准

学术合作

出行行业

金融行业

互联网

通讯

零售企业

标准 IEEE Standard P3652.1 – Federated Machine Learning

Title

Guide for Architectural Framework and Application of Federated Machine Learning

Scope

- Description and definition of federated learning
- The types of federated learning and the application scenarios to which each type applies
- Performance evaluation of federated learning
- Associated regulatory requirements

Call for participation

- More info: https://sagroups.ieee.org/3652-1/

IEEE Standard Association is a open platform and we are welcoming more organizations to join the working group.

IEEE–SA P3652.1 Federated Machine Learning
1st Working Group Meeting

SIST

鹏城实验室 PCL Peng Cheng Laboratory

中国电信 CHINA TELECOM

Tencent

JD.COM 京东

SF

4Paradigm 第四范式

Swiss Re

创新工场 SINOVATION VENTURES

PING AN

CMGFintech 招商金融科技

IEEE

【征稿 】 The 1st International Workshop on Federated Machine Learning for User Privacy and Data Confidentiality (FML’19 ， IJCAI19)

FML_contact@fedai.org

FML 2019

Aug 10-12, 2019, Macau

WELCOME TO FML 2019

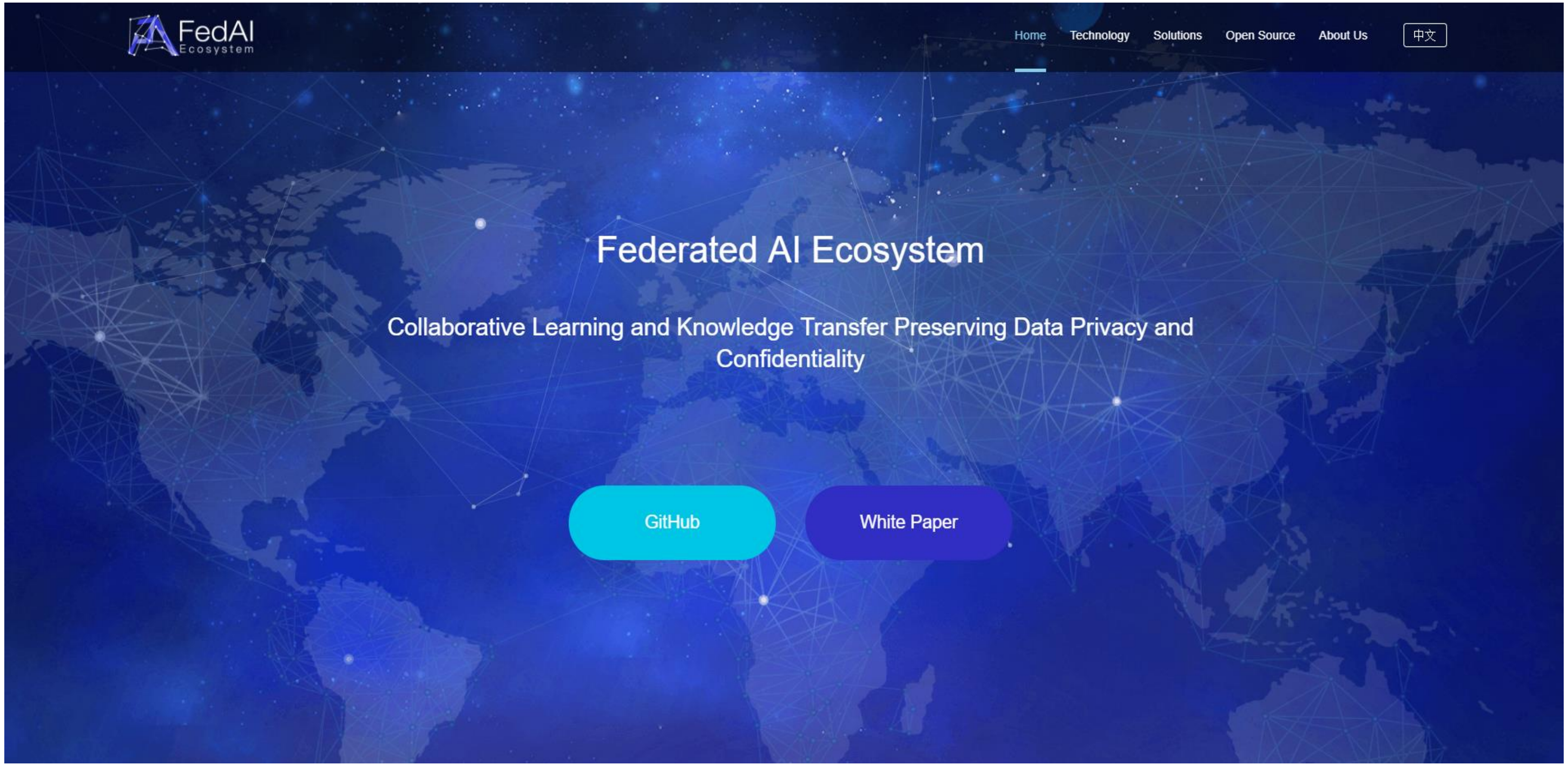
Join us to advance federated machine learning

会议网址：http://fml2019.algorithmic-crowdsourcing.com/

重要日期：

论文截稿日期：2019.5.12
论文接收日期：2019.6.10
会议举办日期：2019.8.10-12

接收的优秀论文将受邀在IEEE Intelligent Systems 特刊发表。另外会议计划颁发Best Paper , Best Student Paper 和 Best Presentation 奖项。



FedAI Ecosystem

Home Technology Solutions Open Source About Us 中文

Federated AI Ecosystem

Collaborative Learning and Knowledge Transfer Preserving Data Privacy and Confidentiality

GitHub White Paper

关注我们的成果

更多信息： **https://www.fedai.org/**

04

Federated AI Technology Enabler （FATE）

FATE 是一个工业级联邦学习框架。能有效帮助多个机构在满足用户隐私保护、数据安全和政府法规的要求下，进行数据使用和建模。

FATE提供了一种基于数据隐私保护的安全计算框架，为机器学习、深度学习、迁移学习算法提供强有力的安全计算支持。 安全底层支持同态加密、秘密共享、哈希散列等多种多方安全计算机制，算法层支持多方安全计算模式下的逻辑回归、Boosting、深度学习，联邦迁移学习等

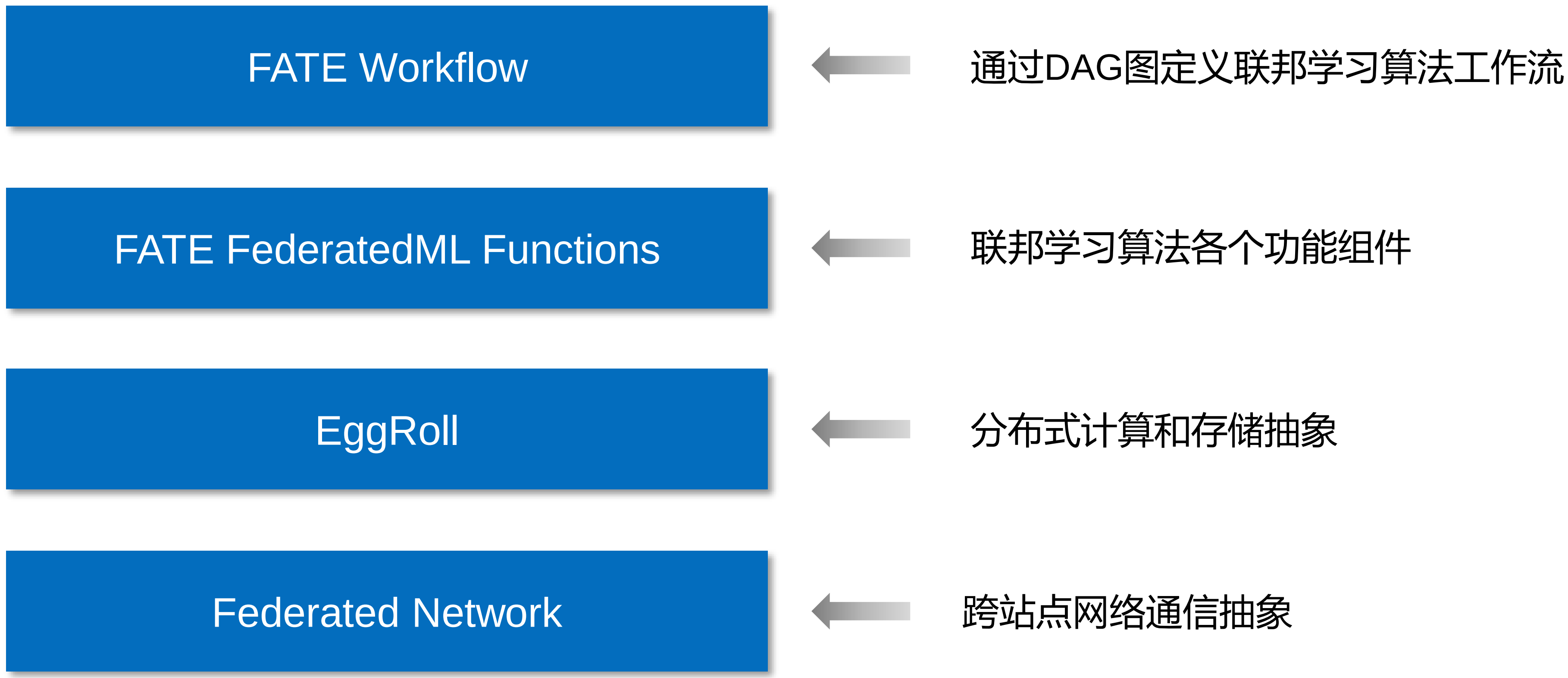
2019年1月份，FATE宣布对外开源

Github：https://github.com/WeBankFinTech/FATE

挑战 Challenges in developing a real-world Federated AI

- 跨站点数据传输安全性和可管理性
- 异构基础架构自适应
    - 计算
    - 存储
    - 网络
- MPC协议下分布式算法（on WAN）易理解和易维护

Federated AI Technology Enabler



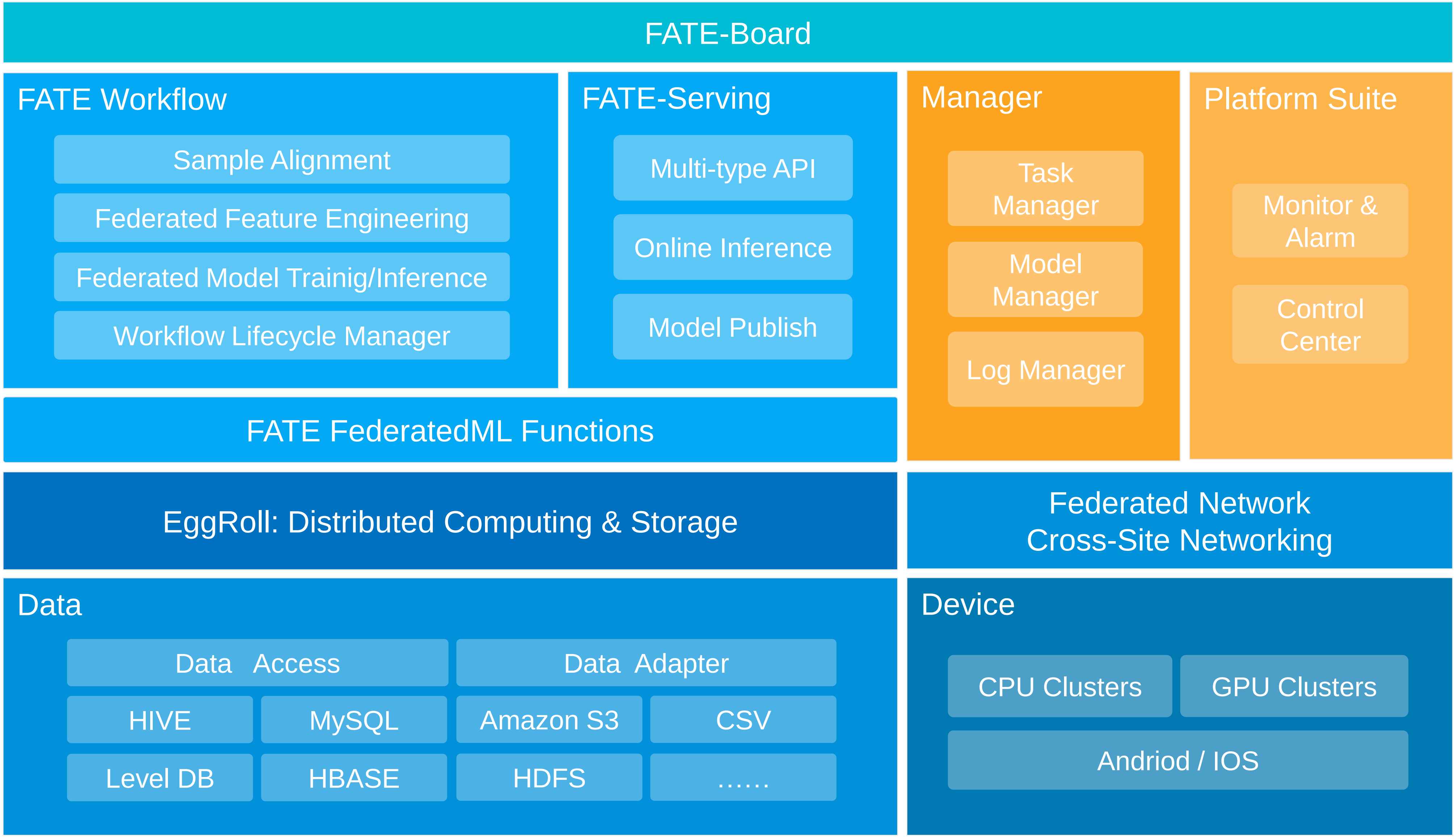
FATE Workflow ← 通过DAG图定义联邦学习算法工作流

FATE FederatedML Functions ← 联邦学习算法各个功能组件

EggRoll ← 分布式计算和存储抽象

Federated Network ← 跨站点网络通信抽象

Overview



FATE-Board

FATE Workflow

- Sample Alignment
- Federated Feature Engineering
- Federated Model Trainig/Inference
- Workflow Lifecycle Manager

FATE-Serving

- Multi-type API
- Online Inference
- Model Publish

Manager

- Task Manager
- Model Manager
- Log Manager

Platform Suite

- Monitor & Alarm
- Control Center

FATE FederatedML Functions

EggRoll: Distributed Computing & Storage

Federated Network
Cross-Site Networking

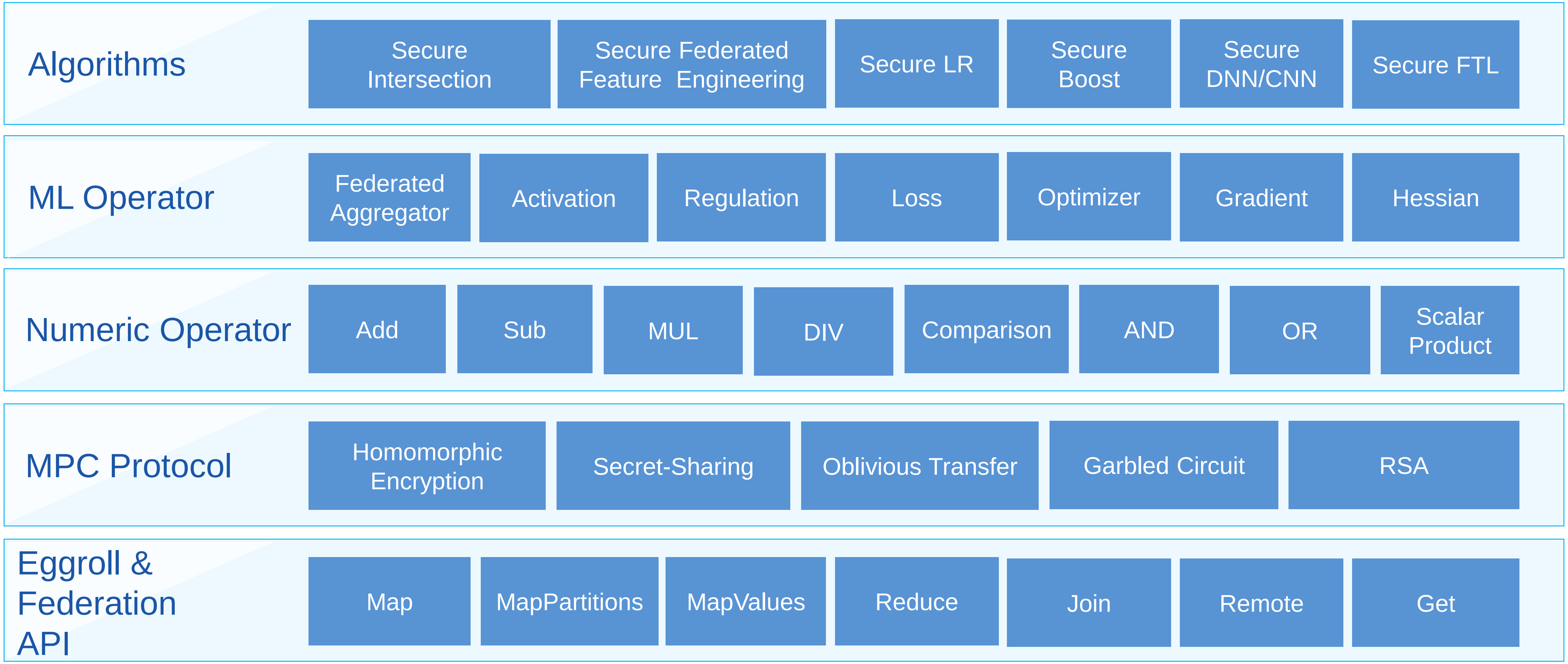
Data

Data Access		Data Adapter	
HIVE	MySQL	Amazon S3	CSV
Level DB	HBASE	HDFS	……

Device

- CPU Clusters
- GPU Clusters
- Andriod / IOS

FATE FederatedML Functions



Algorithms: Secure Intersection | Secure Federated Feature Engineering | Secure LR | Secure Boost | Secure DNN/CNN | Secure FTL

ML Operator: Federated Aggregator | Activation | Regulation | Loss | Optimizer | Gradient | Hessian

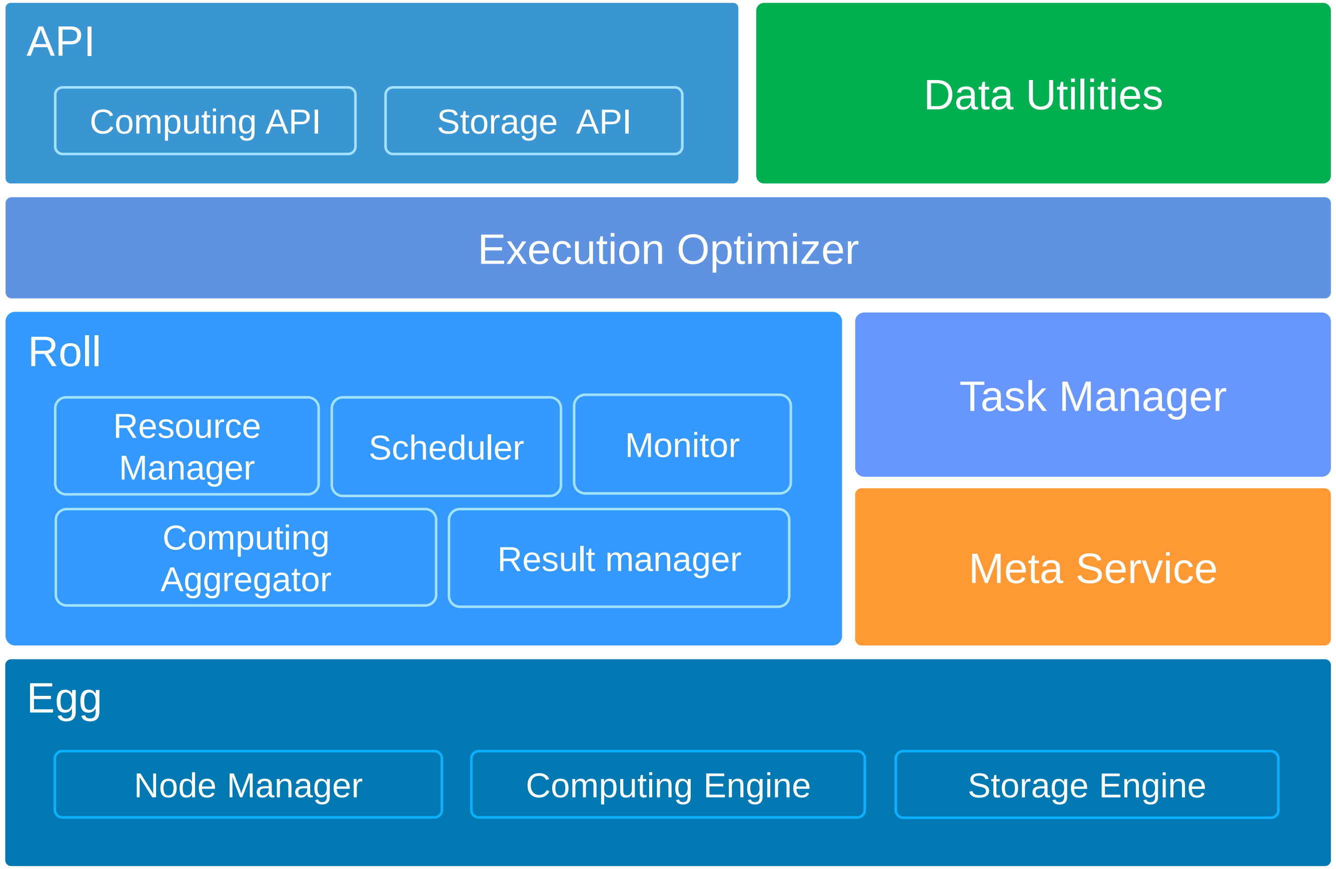
Numeric Operator: Add | Sub | MUL | DIV | Comparison | AND | OR | Scalar Product

MPC Protocol: Homomorphic Encryption | Secret-Sharing | Oblivious Transfer | Garbled Circuit | RSA

Eggroll & Federation API: Map | MapPartitions | MapValues | Reduce | Join | Remote | Get

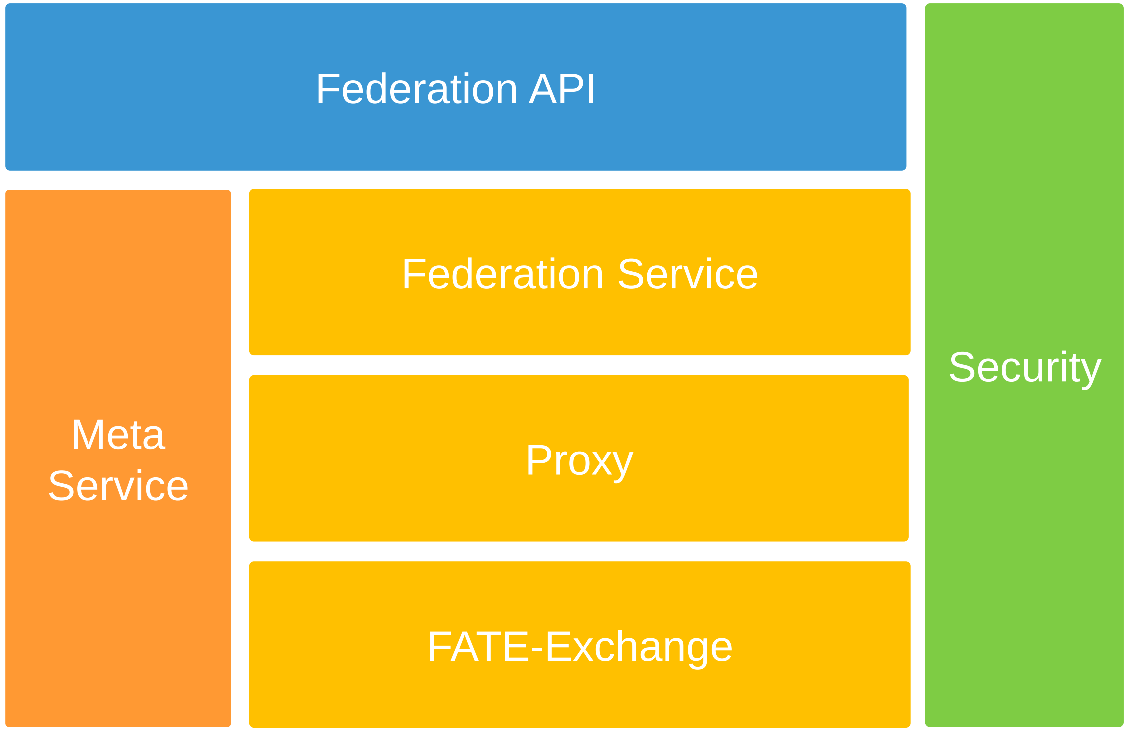
EggRoll – The Infrastructure

- 编程框架
  - EggRoll API：面向算法开发者，通过API实现分布式计算.
- 计算/存储架构
  - 联邦学习一方分布式计算和存储
  - 模块
    - Meta-Service: 元信息管理
    - Roll: 数据 / 计算 调度 (to eggs), 聚合操作等
    - Egg: 计算、存储引擎

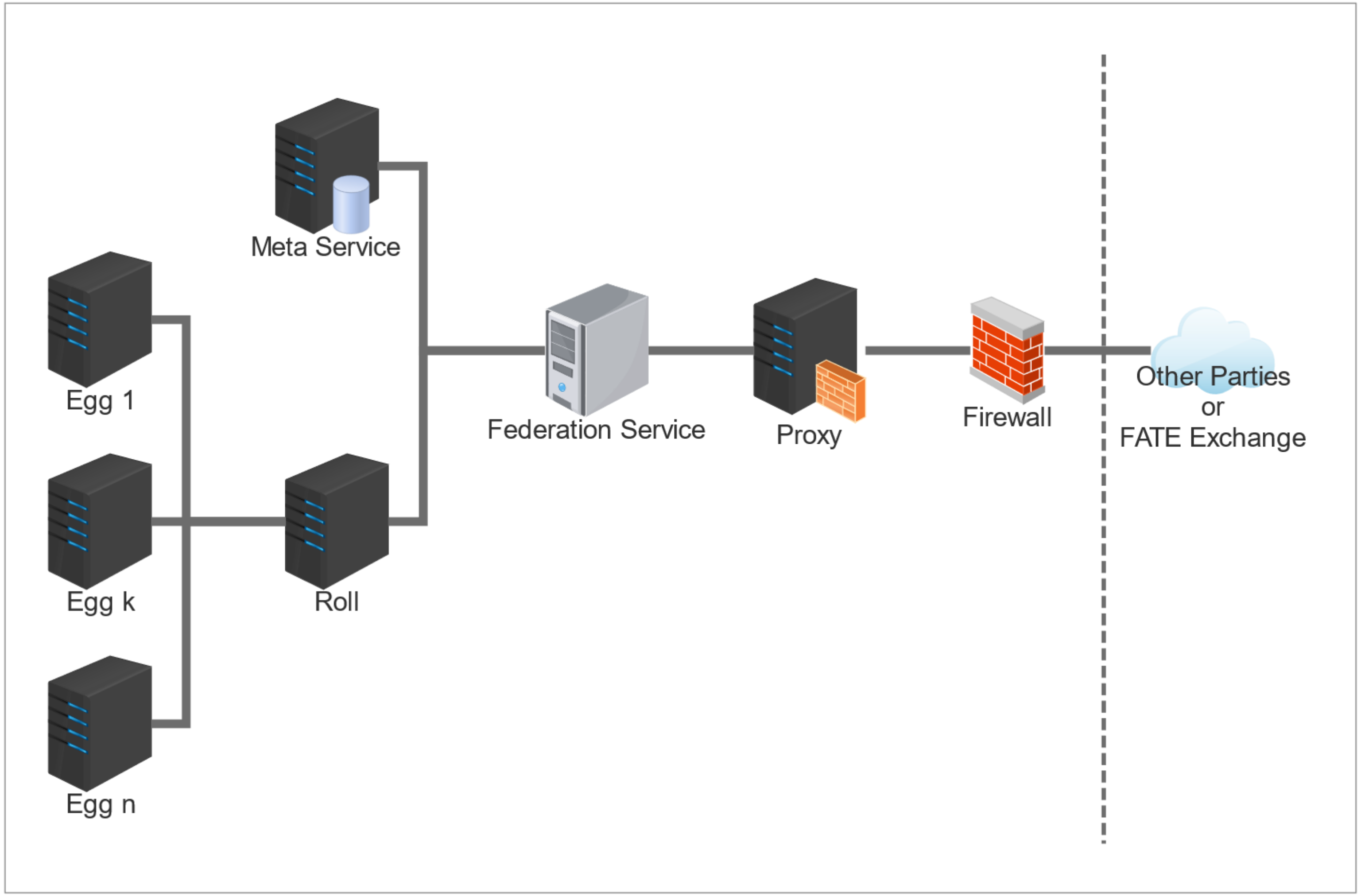


Federated Network– The Infrastructure

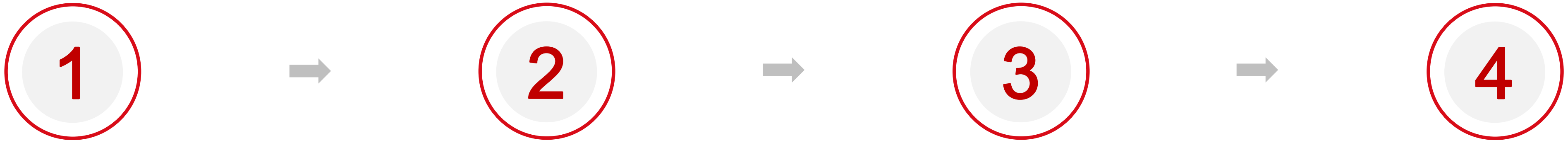
- 编程框架
  - Federation API：面向算法开发者，通过API实现跨站点通信和交互.
- 通信架构
  - 联邦学习多个参与方跨站点通信
  - 模块
    - Meta-Service: 元信息管理
    - Proxy: 应用层联邦学习路由
    - Federation: Global Object（i.e. data to be ‘federated’ among parties）抽象和实现
    - FATE-Exchange



一方部署网络拓扑-示例


Meta Service
Egg 1
Federation Service
Proxy
Firewall
Other Parties
or
FATE Exchange
Egg k
Roll
Egg n

开发流程 Basic Process of Developing a Federated AI Algorithm



选择一个机器学习算法，设计多方安全计算协议

定义多方交互的数据变量

构建算法执行工作流

基于EggRoll & Federation Api 实现算法工作流中各个功能组件

目前 FATE 项目中算法&案例

- Secure Intersection for Sample Alignment

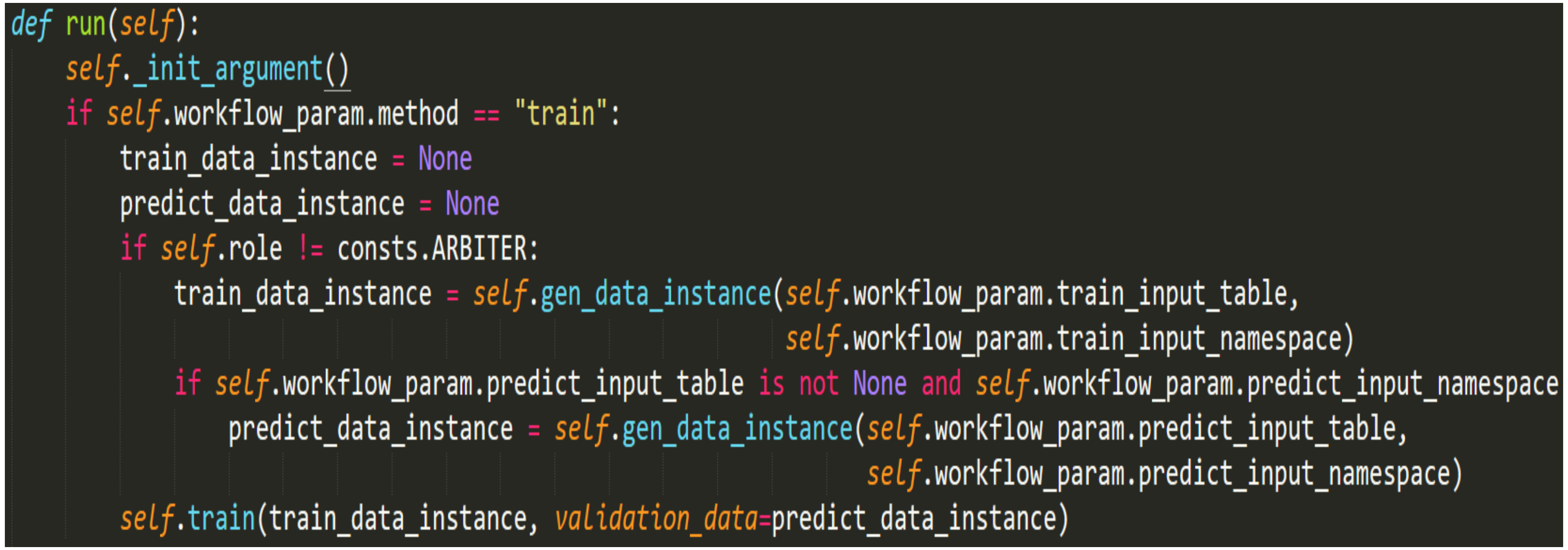
- Vertical-Split Feature Space Federated Learning
    - Secure Logistic Regression
    - Secure Boosting Tree
    - Secure DNN/CNN（Coming Soon）

- Horizontal-Split Sample Space Federated Learning
    - Secure Logistic Regression
    - Secure Boosting Tree（Coming Soon）
    - Secure DNN/CNN（Coming Soon）

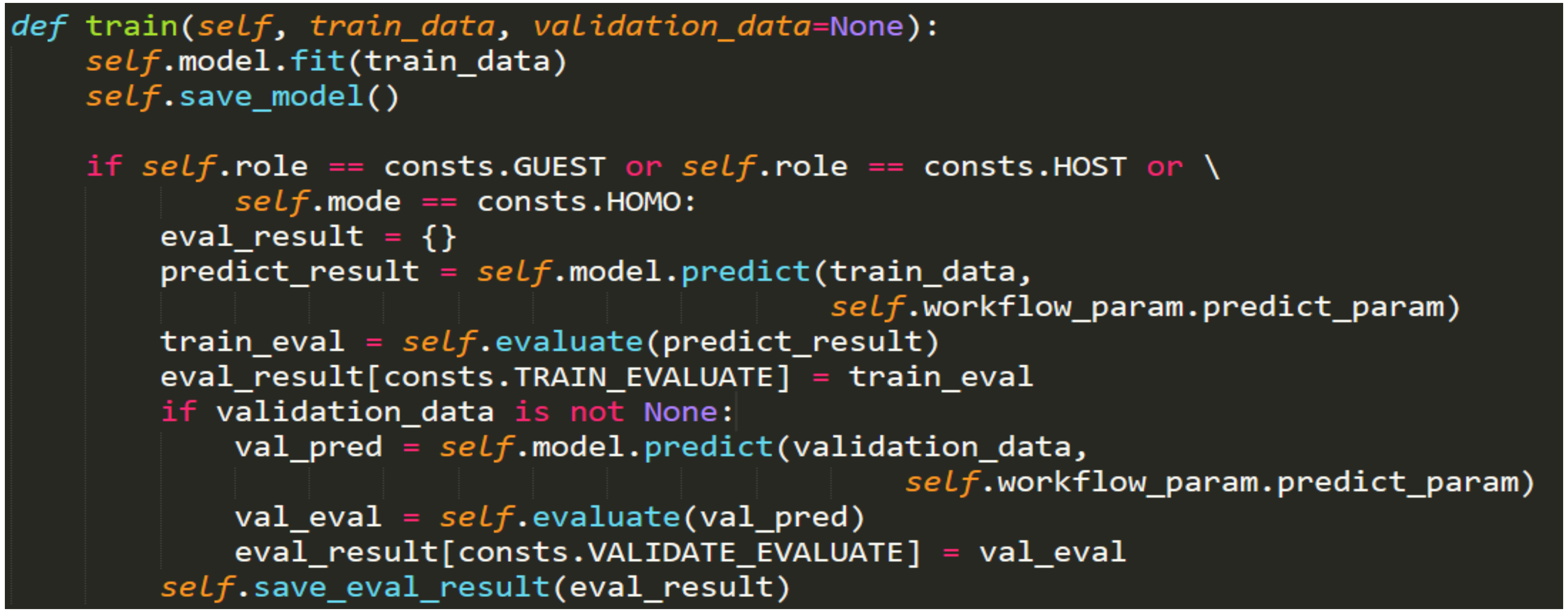
- Secure Federated Transfer Learning

WorkFlow Example

- 工作流
  - 定义联邦算法组件执行工作流
  - 组件
    - 参数初始化组件，
    - 数据加载和转换组件
    - 训练、预测组件
    - 评估组件
    - 模型保存组件
    - …….



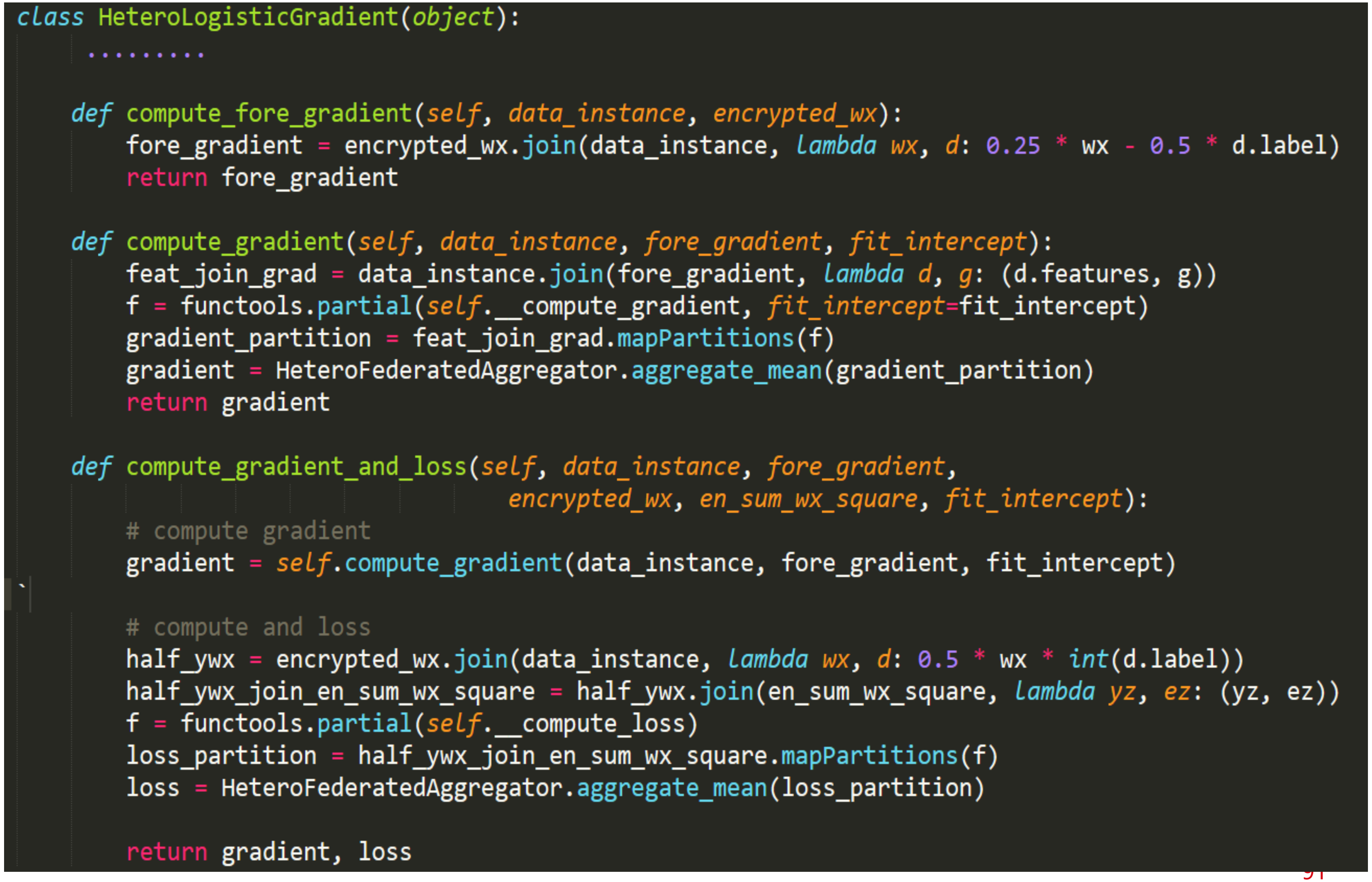
```
def run(self):
    self._init_argument()
    if self.workflow_param.method == "train":
        train_data_instance = None
        predict_data_instance = None
        if self.role != consts.ARBITER:
            train_data_instance = self.gen_data_instance(self.workflow_param.train_input_table,
                                                         self.workflow_param.train_input_namespace)
            if self.workflow_param.predict_input_table is not None and self.workflow_param.predict_input_namespace
                predict_data_instance = self.gen_data_instance(self.workflow_param.predict_input_table,
                                                               self.workflow_param.predict_input_namespace)
        self.train(train_data_instance, validation_data=predict_data_instance)
```



```
def train(self, train_data, validation_data=None):
    self.model.fit(train_data)
    self.save_model()

    if self.role == consts.GUEST or self.role == consts.HOST or \
            self.mode == consts.HOMO:
        eval_result = {}
        predict_result = self.model.predict(train_data,
                                            self.workflow_param.predict_param)
        train_eval = self.evaluate(predict_result)
        eval_result[consts.TRAIN_EVALUATE] = train_eval
        if validation_data is not None:
            val_pred = self.model.predict(validation_data,
                                          self.workflow_param.predict_param)
            val_eval = self.evaluate(val_pred)
            eval_result[consts.VALIDATE_EVALUATE] = val_eval
        self.save_eval_result(eval_result)
```

FederatedML Functions Example

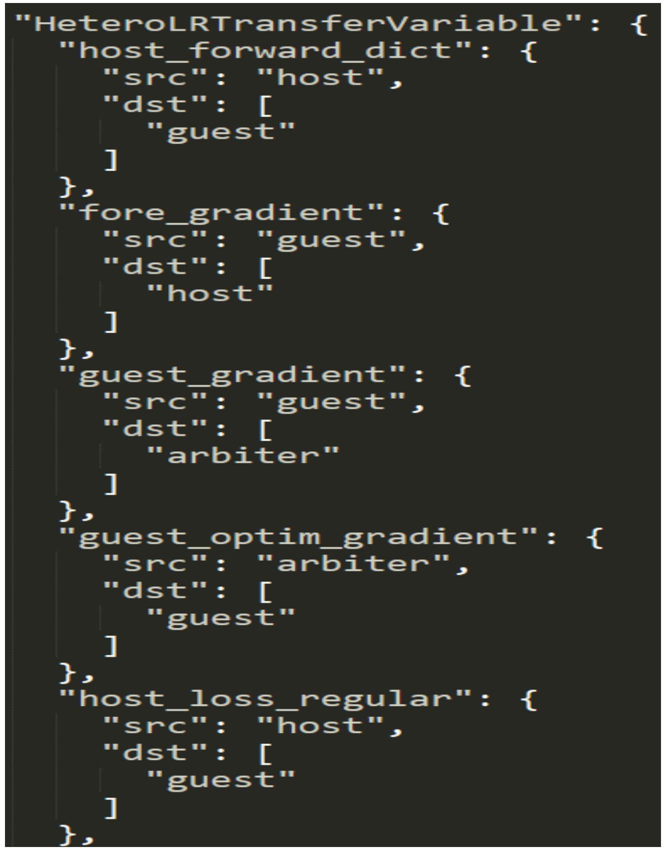
- 纵向LR梯度一方分布式计算
  - 定义梯度和损失计算公式
  - 设计算法并行方式
  - 通过Eggroll API 实现分布式梯度聚合和损失计算



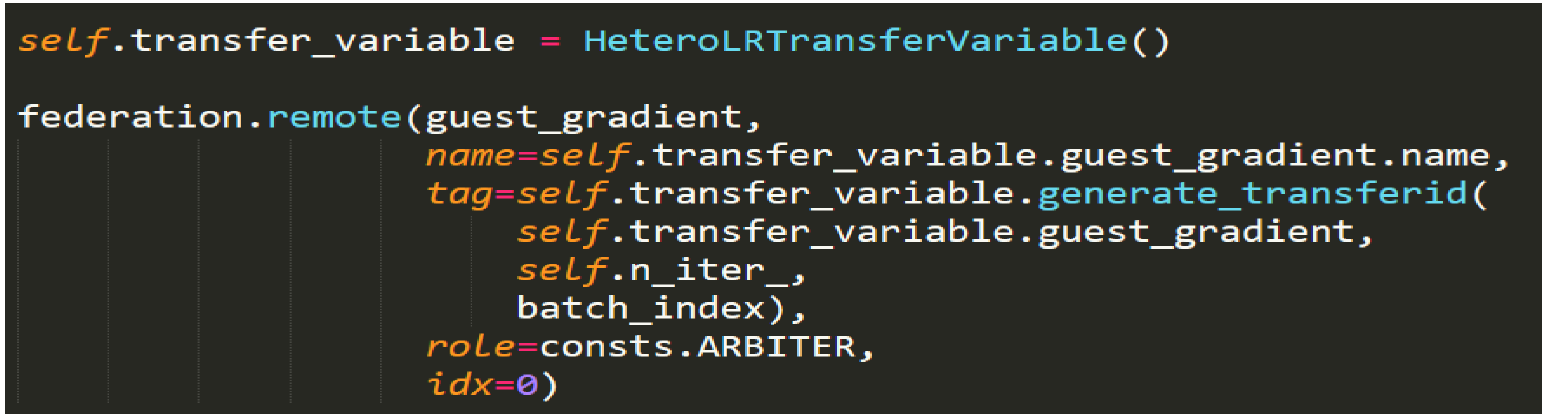
```
class HeteroLogisticGradient(object):
    .........

    def compute_fore_gradient(self, data_instance, encrypted_wx):
        fore_gradient = encrypted_wx.join(data_instance, lambda wx, d: 0.25 * wx - 0.5 * d.label)
        return fore_gradient

    def compute_gradient(self, data_instance, fore_gradient, fit_intercept):
        feat_join_grad = data_instance.join(fore_gradient, lambda d, g: (d.features, g))
        f = functools.partial(self.__compute_gradient, fit_intercept=fit_intercept)
        gradient_partition = feat_join_grad.mapPartitions(f)
        gradient = HeteroFederatedAggregator.aggregate_mean(gradient_partition)
        return gradient

    def compute_gradient_and_loss(self, data_instance, fore_gradient,
                                  encrypted_wx, en_sum_wx_square, fit_intercept):
        # compute gradient
        gradient = self.compute_gradient(data_instance, fore_gradient, fit_intercept)

        # compute and loss
        half_ywx = encrypted_wx.join(data_instance, lambda wx, d: 0.5 * wx * int(d.label))
        half_ywx_join_en_sum_wx_square = half_ywx.join(en_sum_wx_square, lambda yz, ez: (yz, ez))
        f = functools.partial(self.__compute_loss)
        loss_partition = half_ywx_join_en_sum_wx_square.mapPartitions(f)
        loss = HeteroFederatedAggregator.aggregate_mean(loss_partition)

        return gradient, loss
```

Federation API Example

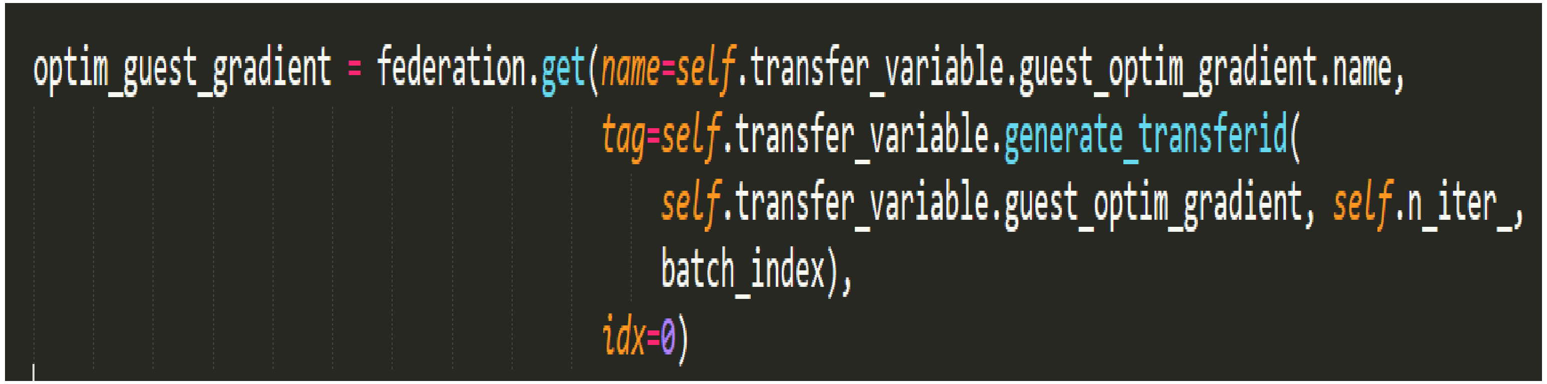
- 纵向LR梯度两方联合
  - 定义算法交互信息-梯度（json 配置文件，数据源和目的地）
  - 生成梯度交互信息唯一标识符
  - Federation API 完成梯度交互信息的收发



```
"HeteroLRTransferVariable": {
  "host_forward_dict": {
    "src": "host",
    "dst": [
      "guest"
    ]
  },
  "fore_gradient": {
    "src": "guest",
    "dst": [
      "host"
    ]
  },
  "guest_gradient": {
    "src": "guest",
    "dst": [
      "arbiter"
    ]
  },
  "guest_optim_gradient": {
    "src": "arbiter",
    "dst": [
      "guest"
    ]
  },
  "host_loss_regular": {
    "src": "host",
    "dst": [
      "guest"
    ]
  },
```



```
self.transfer_variable = HeteroLRTransferVariable()

federation.remote(guest_gradient,
                  name=self.transfer_variable.guest_gradient.name,
                  tag=self.transfer_variable.generate_transferid(
                      self.transfer_variable.guest_gradient,
                      self.n_iter_,
                      batch_index),
                  role=consts.ARBITER,
                  idx=0)
```



```
optim_guest_gradient = federation.get(name=self.transfer_variable.guest_optim_gradient.name,
                                      tag=self.transfer_variable.generate_transferid(
                                          self.transfer_variable.guest_optim_gradient, self.n_iter_,
                                          batch_index),
                                      idx=0)
```

Thank you!

Questions

yangliu@webank.com