

Towards Fair Federated Learning

SIGKDD 2021 Tutorial

Tutors

Zirui Zhou, Lingyang Chu, Changxin Liu,
Lanjun Wang, Jian Pei, Yong Zhang

Outline

- Federated Learning: A Quick Review
 - Overview / Horizontal FL / Vertical FL
- Taxonomy of Fairness in Federated Learning
 - Performance Fairness / Collaboration Fairness / Model Fairness
- Towards Performance Fairness in Federated Learning
 - Fair Resource Allocation / Agnostic FL / Personalization
- Towards Collaboration Fairness Federated Learning
 - Contribution Measurement / Types of Reward / Incentive Mechanism
- Towards Model Fairness in Federated Learning
 - Fairness in Machine Learning / Horizontal FL / Vertical FL
- Conclusions and future directions

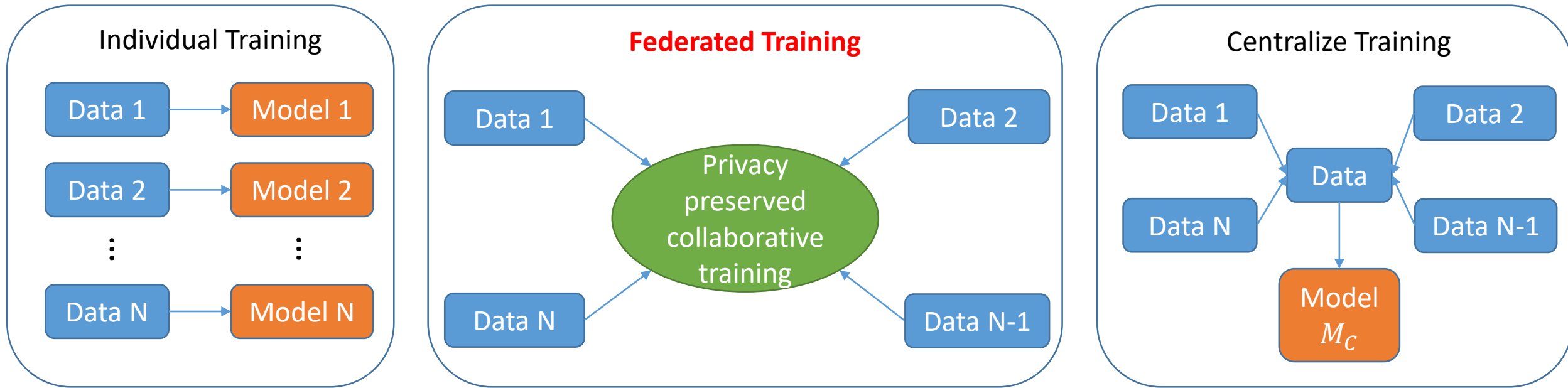
Part I

Federated learning: a quick review

Tutor: Yong Zhang

Overview of Federated Learning

Motivations and definition of federated learning [Yang et al. '19]



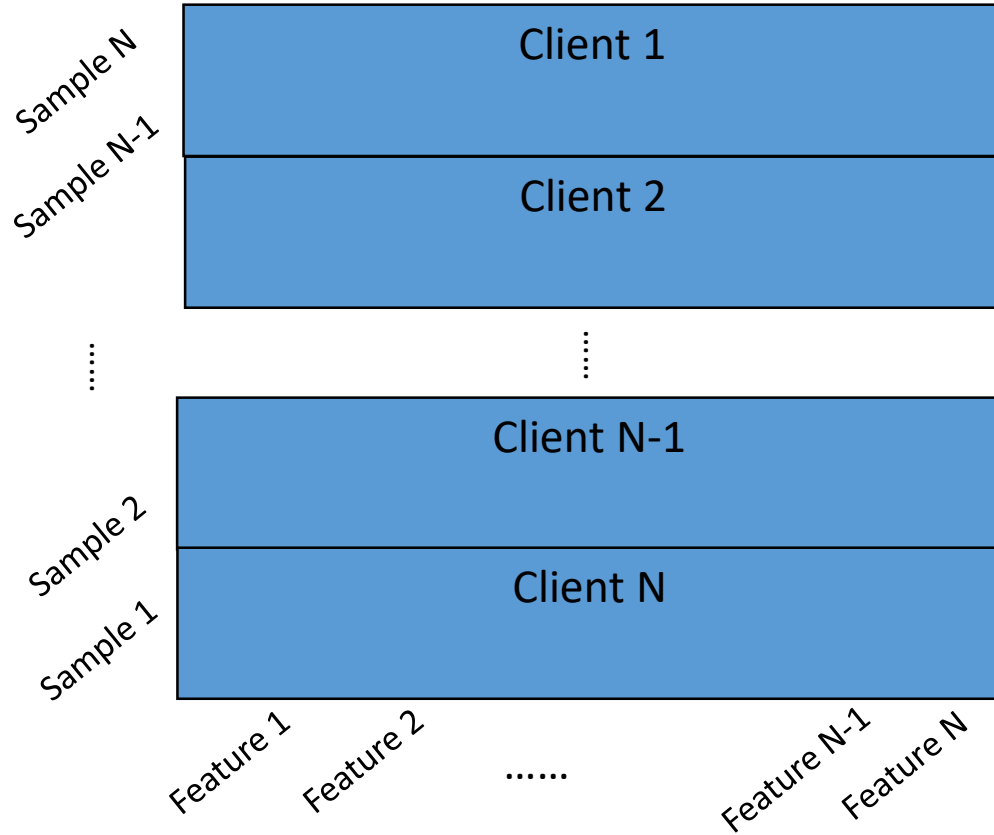
Motivations. Federated learning is mainly motivated to solve the following two problems in machine learning (ML) model training:

- Concerns on user data privacy and confidentiality and the laws that oversee them.
- Inability to build an ML model due to inadequate data or training cost on ML implementation of the computational cost involved for training an ML model.

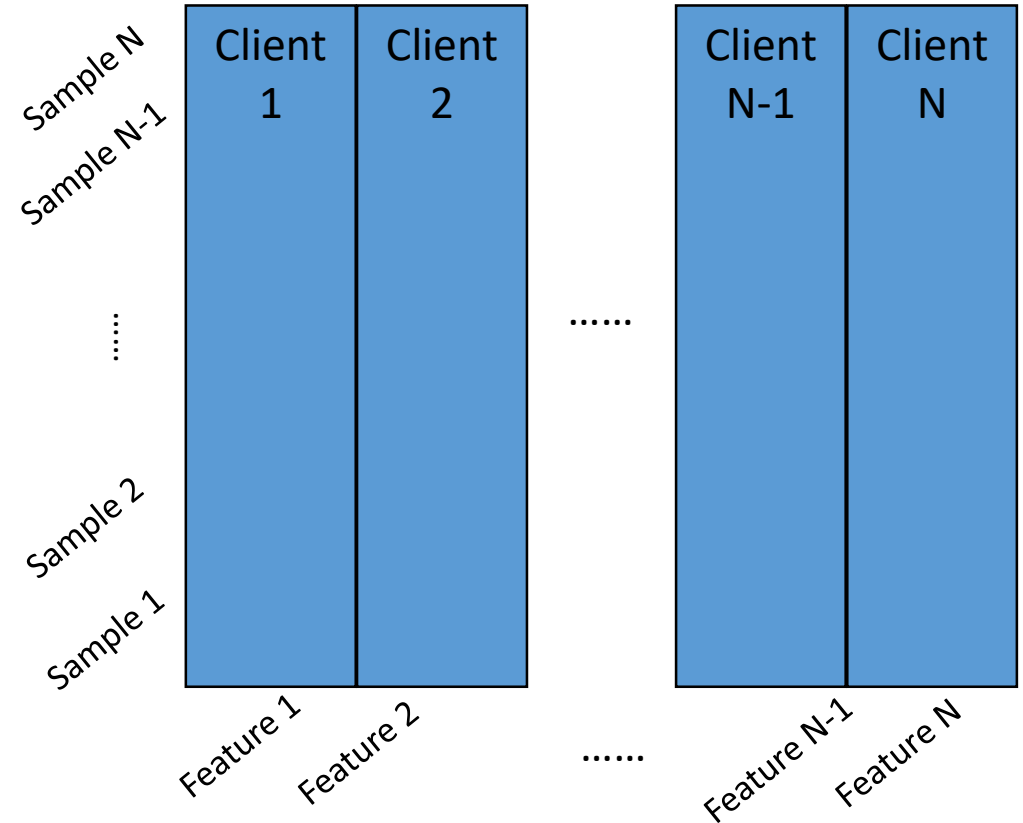
Definition. *Federated learning* is a framework which builds machine learning models based on data sets that are distributed across multiple devices while preventing data leakage. We say the federated learning algorithm has δ -accuracy loss if $|f(M_C) - f(M_F)| \leq \delta$ where f evaluates the accuracy of a model on the test data. [Yang et al. '19]

Two Versions of Federated Learning

Horizontal Federated Learning



Vertical Federated Learning



The setting of **horizontal** federated learning

The horizontal federated learning satisfies the following setting of data.

- Different data owners hold the same set of features but different sets of samples.

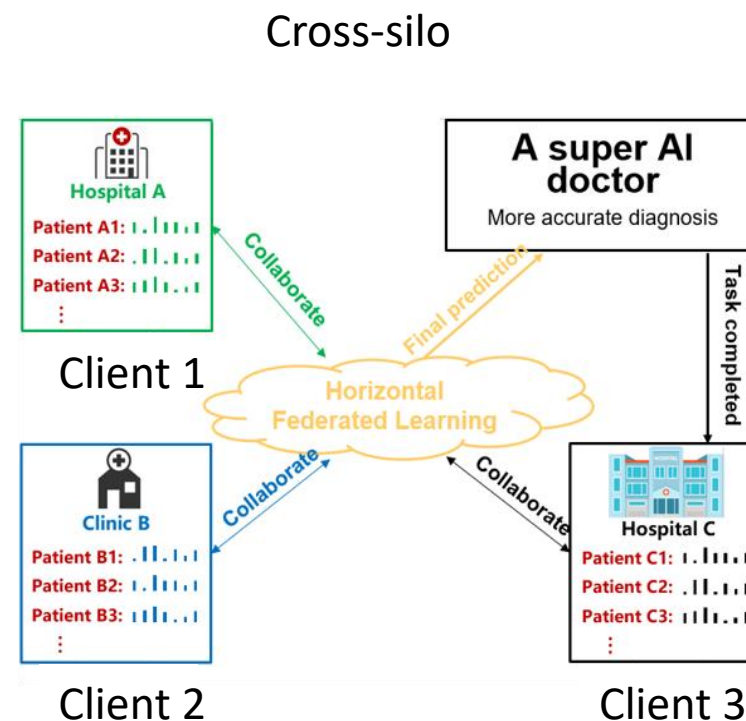
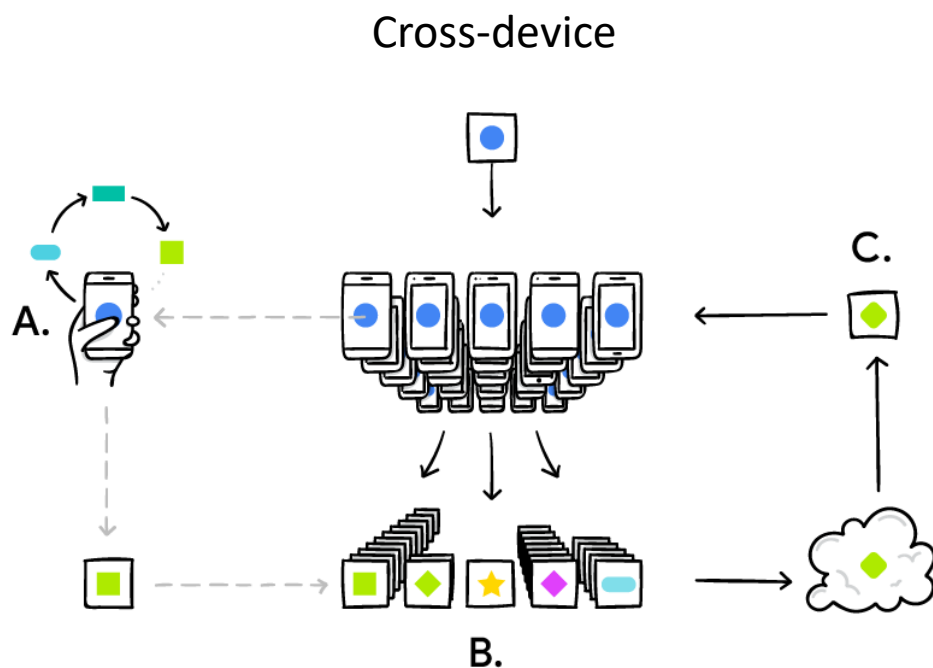
The setting of **vertical** federated learning

The vertical federated learning satisfies the following settings of data.

- Different data owners hold different sets of features but the same set of samples.
- The different sets of features held by different data owners are aligned by a unique sample ID.

Horizontal Federated Learning

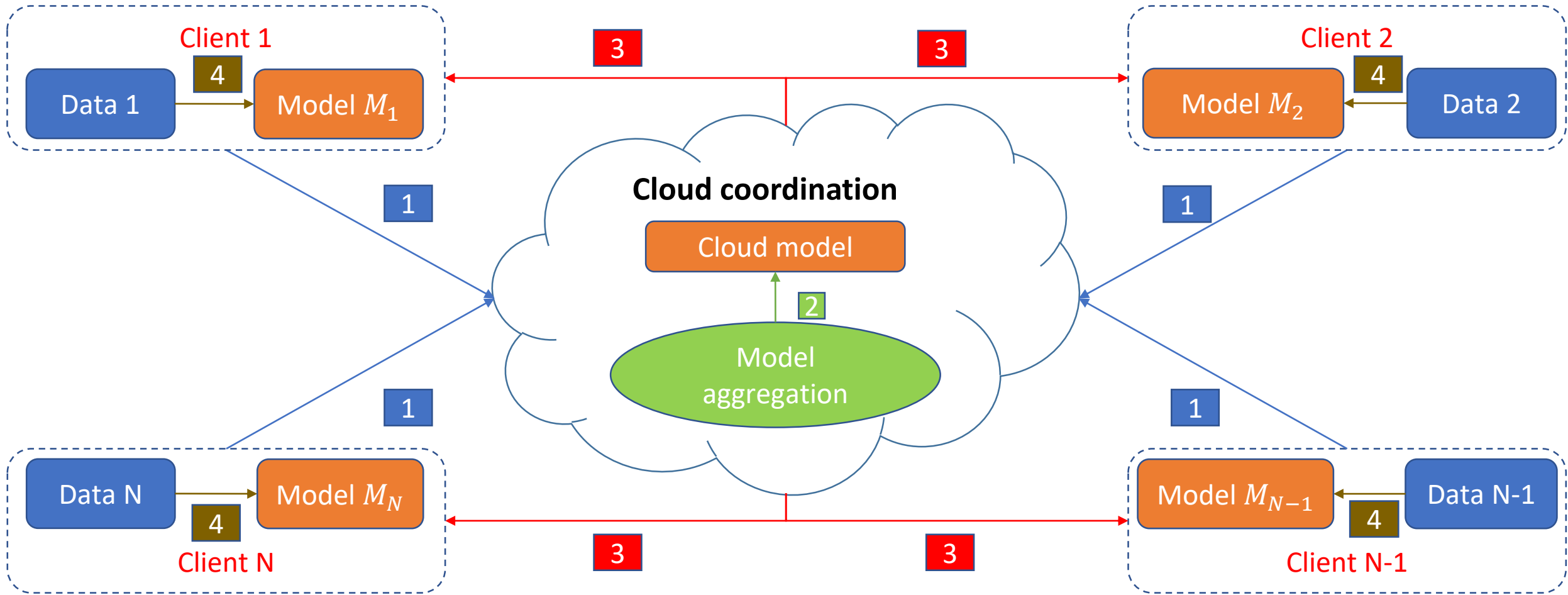
Examples of horizontal federated learning



Other applications:

- Collaborative Credit Score Evaluation: collaboration between banks.
- Collaborative Targeted Advertisement: collaboration between retailers.
-

General framework for horizontal federated learning (centralized) [Mothukuri et al. '21]



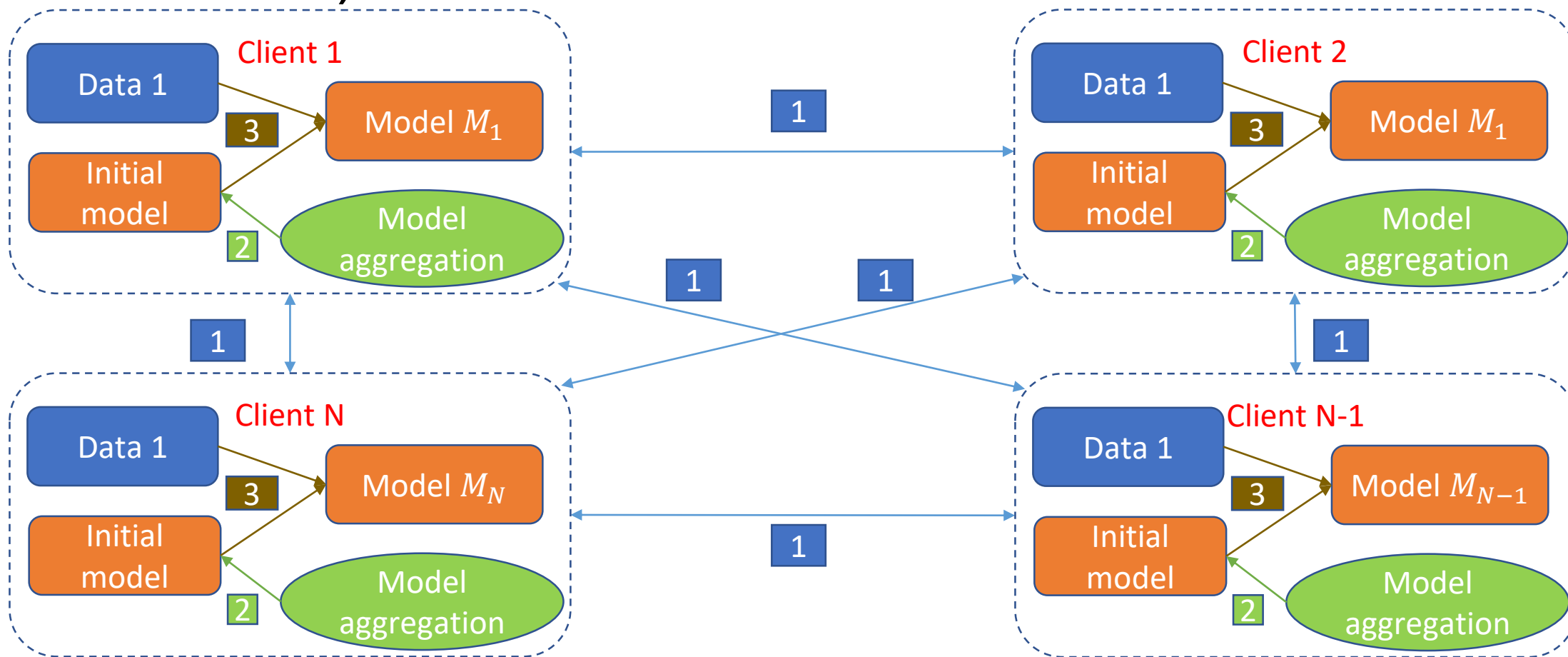
1. Sending model updates to the cloud

2. Aggregating model parameters on the cloud

3. Sending back the aggregated cloud model

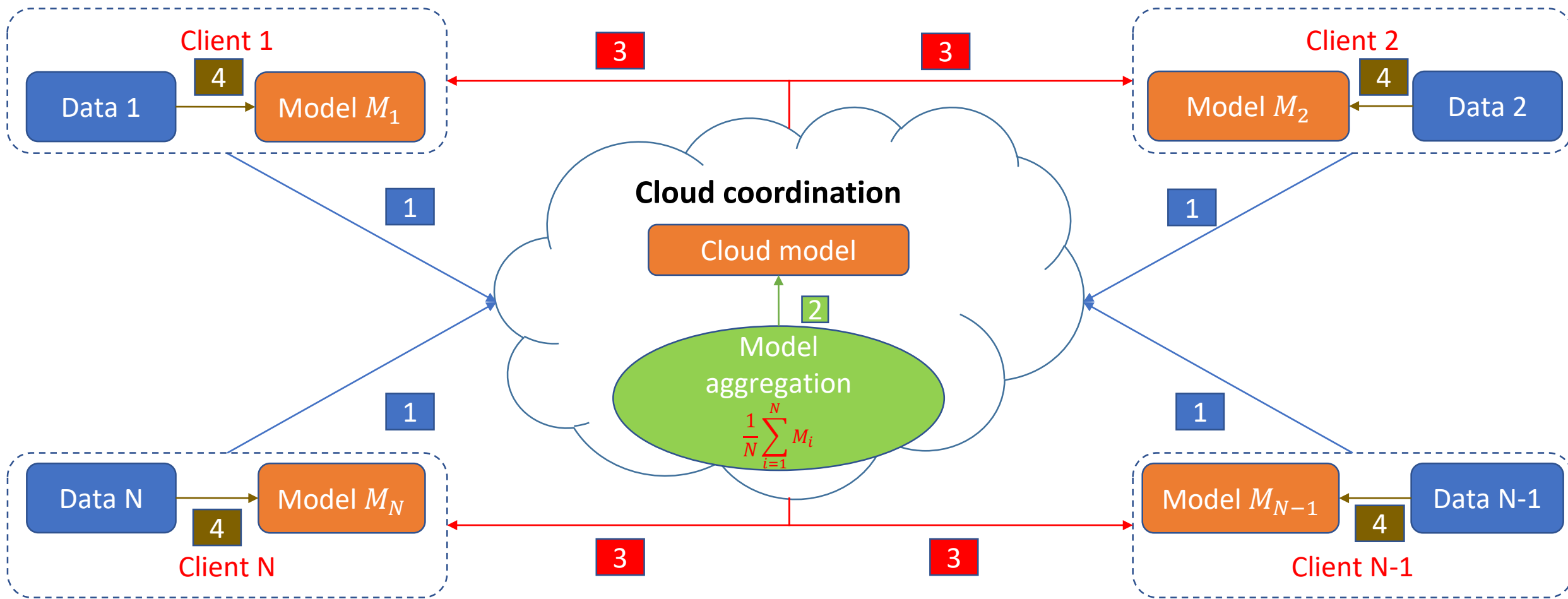
4. Updating each model locally

General framework for horizontal federated learning (decentralized) [Mothukuri et al. '21]



1. Sending model updates to each other
2. Aggregating model parameters on each client to obtain a new initial model
3. Updating each model with local data and the new initial model

Vanilla algorithm for HFL: Federated Averaging (FedAvg) [McMahan et al. '17]

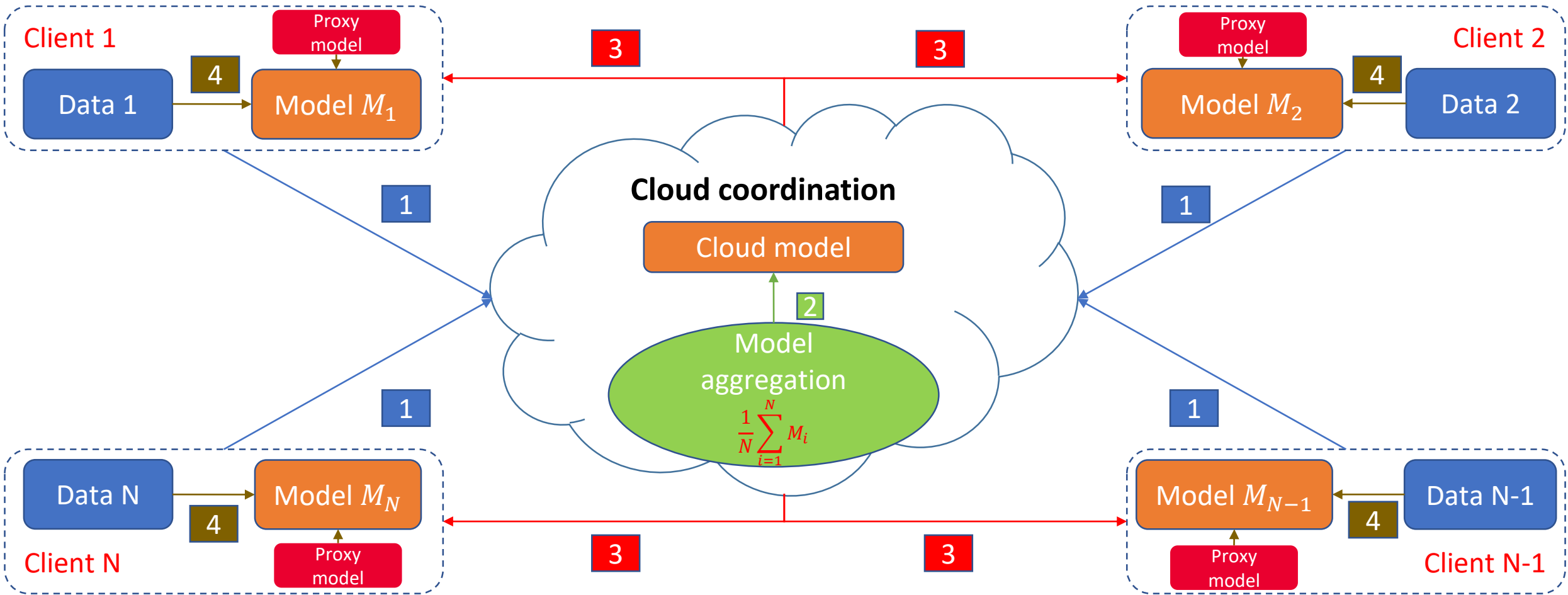


The details of the FedAvg method

- Step 1: Data owners send their models to the cloud
- Step 2: The cloud computes the **average of all models**
- Step 3: The cloud sends the averaged model to each of the data owners
- Step 4: The data owners **locally update** the averaged model using their own private data

No theoretical guarantee!!

FedProx: a variant of FedAvg [Li et al. '18]



The details of the FedProx method

- Steps 1-3: Same as FedAvg
- Step 4: The data owners **update** the averaged model using their own private data, but **with a proxy model** that the updated model should be similar to the averaged model.

Advantages over FedAvg:

- Theoretical convergence is provided.
- More robust convergence in practice.

Challenges for the vanilla HFL

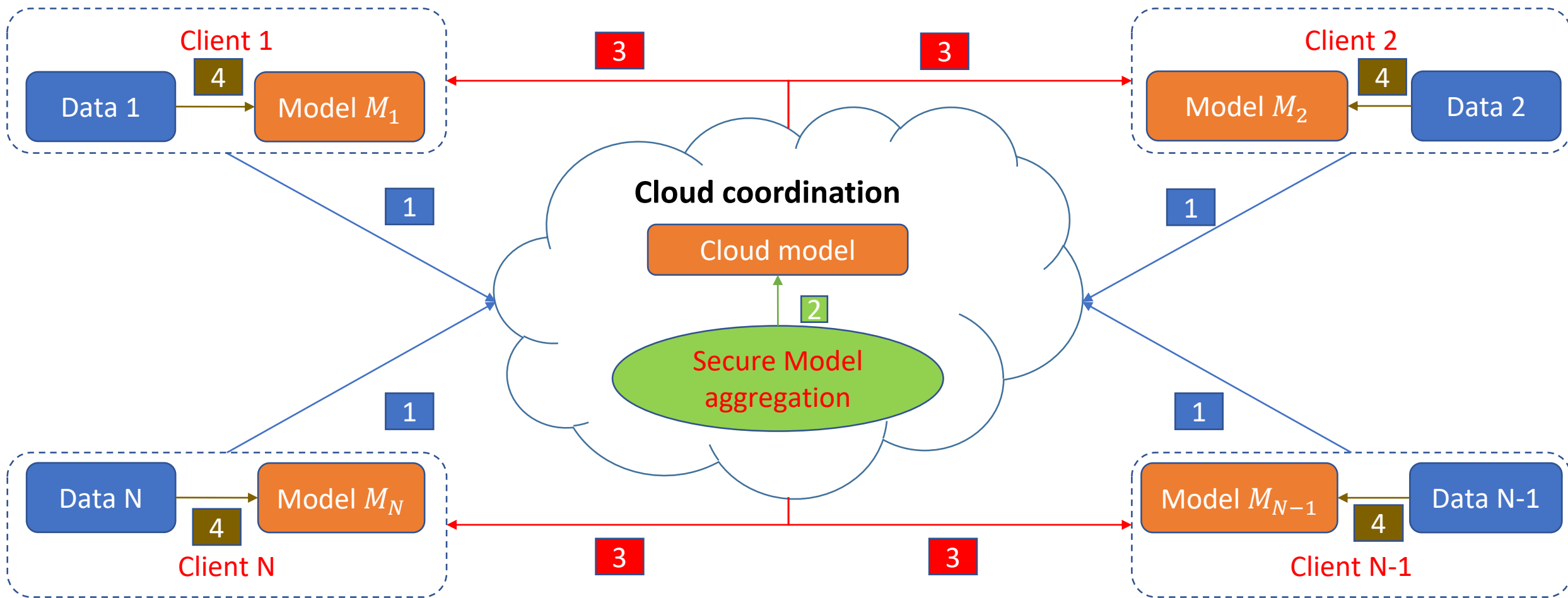
Privacy challenge

- Privacy is one of the essential properties of federated learning. [Yang et al. '19]
- Sending model may not fully protect clients' data. [Mohassel et al. '17]

Data distribution challenge

- The different sets of samples held by different data owners may be non-IID, which means different sets of samples obey different data distribution. [Li et al. '18]
- One universe model may not be suitable for all the distribution. [Kairouz et al. '19]

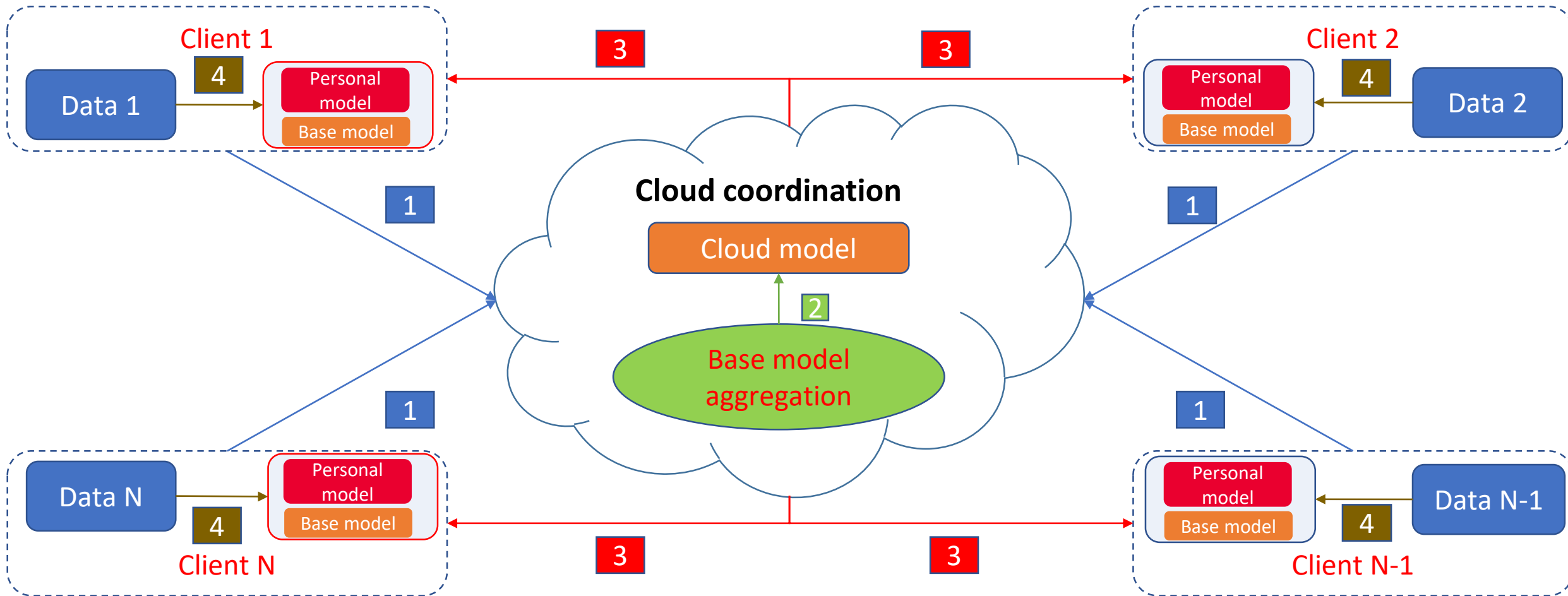
Algorithm for improving privacy of HFL



The details of the privacy enhanced FL method

- Step 1: Data owners send their models to the cloud
- Step 2: The cloud computes the **average of all models through a complicated secure protocols, such as secure multi-party computation [Hao et al. '19], differential privacy [Truex et al. '19], and trusted execution environments [Chen et al. '20]. Essentially, these aggregation methods trade efficiency for privacy.**
- Step 3: The cloud sends the averaged model to each of the data owners
- Step 4: The data owners **locally update** the averaged model using their own private data

Algorithms for improving performance of HFL on non-iid data



The details of the FedPer method [Arivazhagan et al. '19]

- Step 1: Data owners send their same base models to the cloud
- Step 2: The cloud computes the **average of all base models**
- Step 3: The cloud sends the averaged base model to each of the data owners
- Step 4: The data owners **locally update the base model and its own personal model** using their own private data

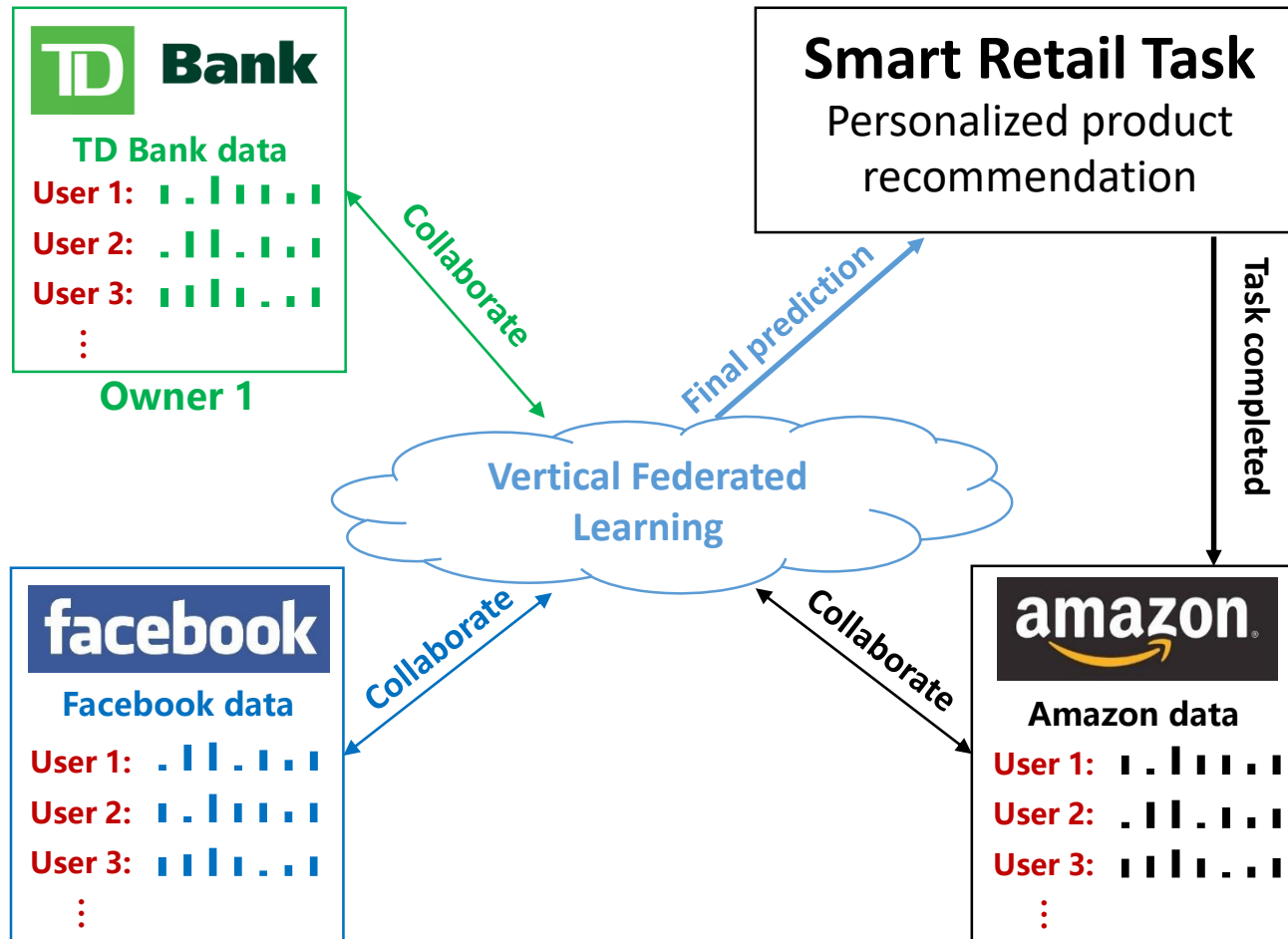
Other personalize federated learning methods

- APFL [Deng et al. '20]
- Scaffold [Karimireddy et al. '20]
- FedAMP [Huang et al. '21]

Vertical Federated Learning

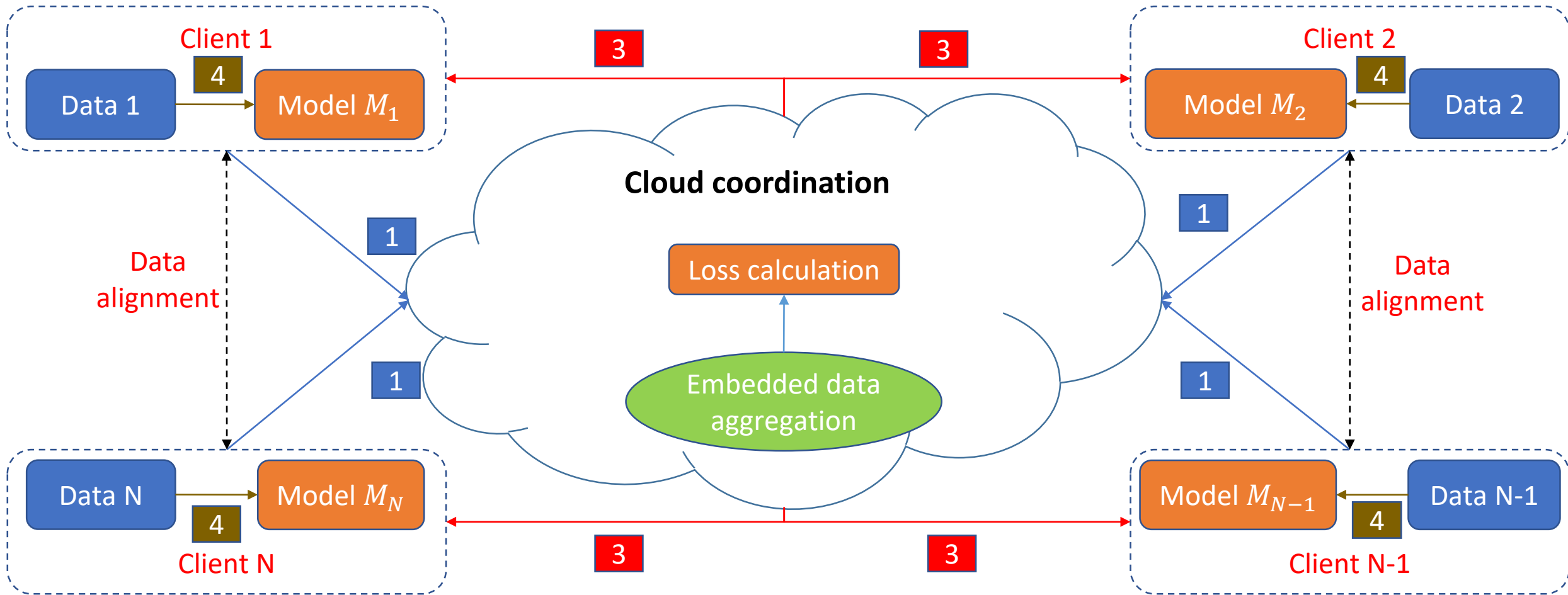
Examples of vertical federated learning (VFL)

VFL is more common in cross-silo setting



- **Smart retail task:** as shown in the figure, a **retailer** (e.g., Amazon) wants to improve the performance of personalized product recommendation by using the financial data from a **bank** (e.g., TD Bank) and the social relation data from a **social network company** (e.g., Facebook).
- **Credit score evaluation task:** the bank wants to improve the evaluation accuracy of user credit scores by using the purchasing data from the retailer and the social relation data from the social network company.
- **Content recommendation task:** the social network company wants to push user interested contents more accurately by analyzing the financial data from the bank and the purchasing data from the retailer.

General framework for vertical federated learning [Hu et al. '19]



1. Sending locally embedded data to the cloud

2. Aggregating the embedded data and calculating the loss

3. Backpropagating the partial gradients to the corresponding clients

4. Updating each client's model locally

Differences between HFL and VFL:

1. Difference on data alignment.

2. Difference on local models.

3. Difference on data upload.

4. Difference on data download

Challenges for VFL

Privacy challenge

- Privacy is one of the essential properties of federated learning. [Yang et al. '19]
- Sending embedded data may not fully protect clients' data privacy. [Weng et al. '20]

Efficiency challenge

- Since cloud or only a very small subset of clients hold labels in VFL, most existing algorithms are limited to synchronous computation which is not efficient. [Gu et al. '21]

Algorithm for improving privacy of VFL: Additively Homomorphic Encryption (AHE)

[Yang et al.'19]

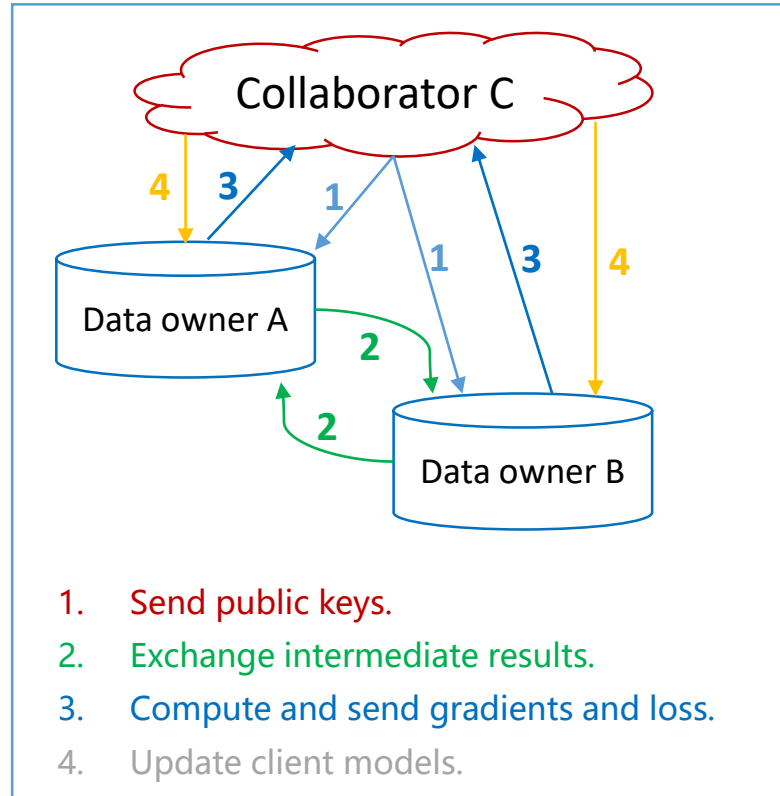


Figure: AHE based vertical federated learning.

Figure shows the details of the AHE based model training.

- Step 1: Collaborator C creates encryption pairs and send public key to A and B.
- Step 2: Data owners A and B encrypt and exchange the intermediate results for gradient and loss calculations.
- Step 3: A and B computes encrypted gradients and loss, then send them to C.
- Step 4: C decrypts and send the decrypted gradients and loss back to A and B to update the client models on A and B.

Disadvantages of AHE based vertical federated learning.

- Since C has the private key, if C collude with A, the privacy of B is corrupted; if C collude with B, then A has no privacy.
- Since C knows all the gradients and updates computed by A and B, C can easily infer properties of the private data in A and B.
- Encryption and computation in encrypted space is time consuming.
- Accuracy loss is inevitable due to the Taylor approximation of non-linear computations.

Algorithm for improving privacy of VFL: Secure Multi-party Computation (SMC)

[Mohassel et al. '17]

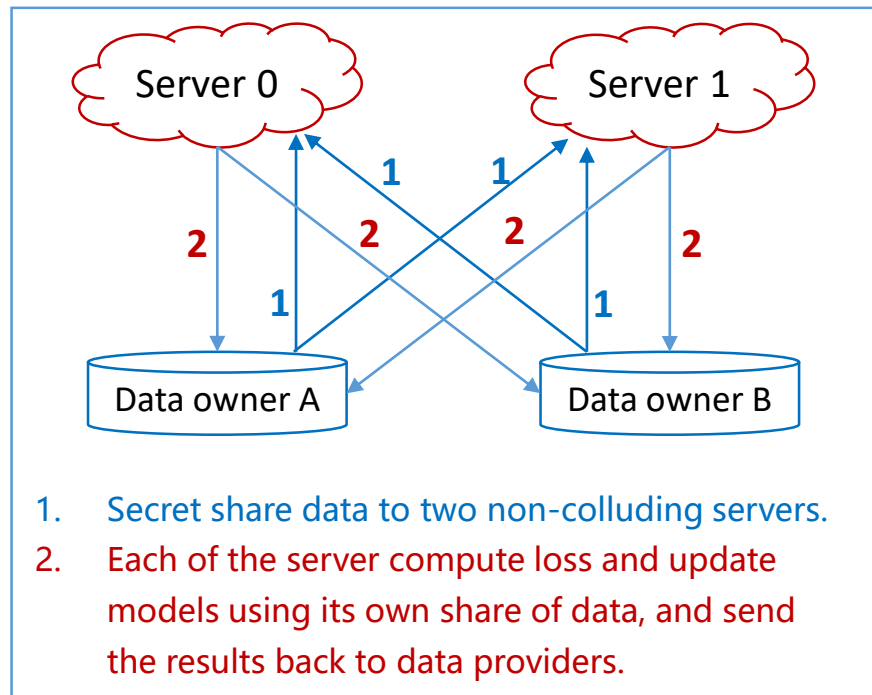


Figure : SMC based vertical federated learning.

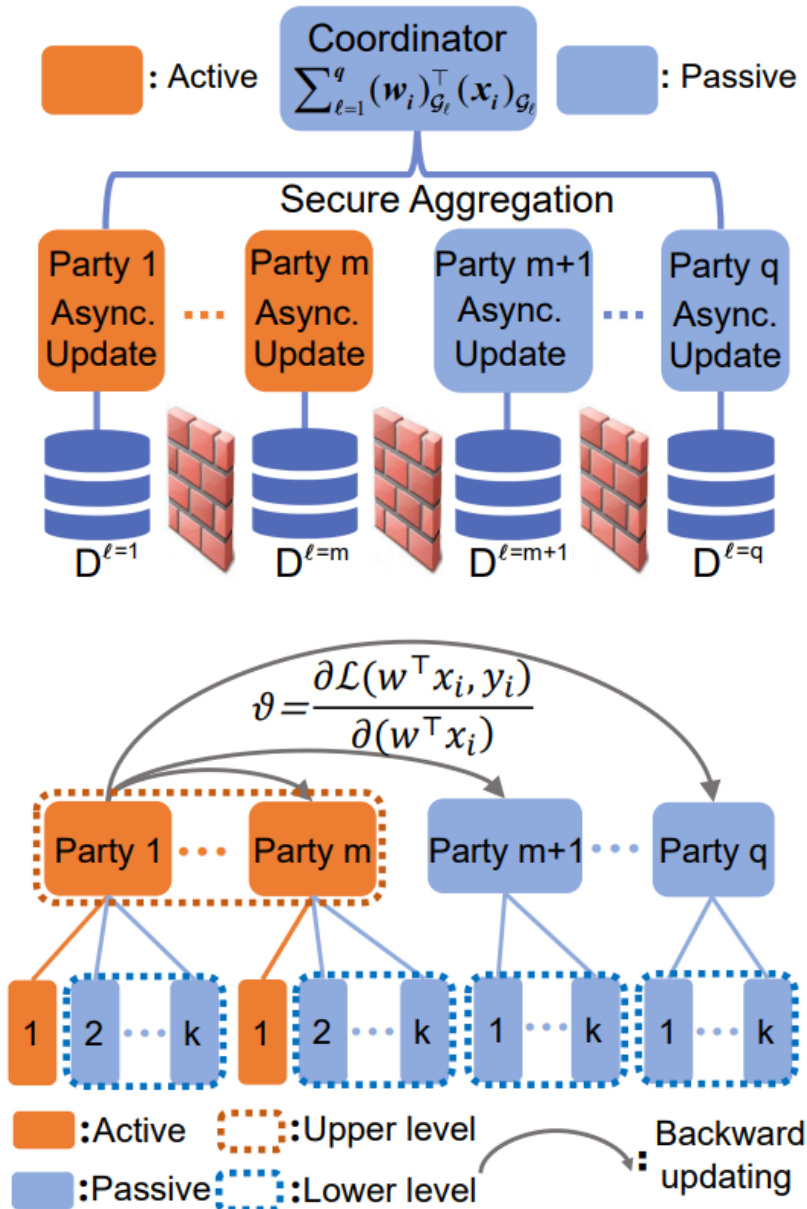
Figure shows the details of the SMC based model training.

- Step 1: Each of the data owners A and B use **secret share method** to encrypt and send their data to two **non-colluding servers**. The encrypted shared data on one server can only be decrypted if the encrypted shared data on the other server is given.
- Step 2: Each of the servers apply **secure arithmetic operations** on its own encrypted shared data to compute loss and update model parameters. The result computed by one server remains to be encrypted if the result computed by the other server is not known. In the end, each server send their results to both A and B, such that A and B can recover the true results and update their models.

Disadvantages of SMC based vertical federated learning.

- If Server 0 collude with Server 1, then the privacy of A and B is breached.
- Since A and B know about the model updates of each other, they can infer the data properties of each other.
- Encryption and computation in encrypted space is time consuming.
- Secure arithmetic operations cause inevitable accuracy loss due to the approximation of non-linear computations.

Algorithm for improving efficiency of VFL: VFB2^[Gu et al. '21]



Setup:

Active workers: hold partial features and labels

Passive workers: only hold partial features

Algorithm overview:

Conduct VFL updates in parallel and asynchronously by making active workers cooperate with passive workers

Key ideas:

- A novel backward updating mechanism is proposed to enable both active and passive parties to collaboratively learn the model with privacy-preserving.
- A bi-level asynchronous parallel architecture is proposed to enable all parties as update the model asynchronously.
- A tree-structured communication scheme is applied to aggregate the outputs from workers.

Part II

Taxonomy of Fairness in FL

Tutor: Zirui Zhou

Fairness in federated learning

- Background
 - Many participants in FL are self-interested and need incentive
 - Sustainable development of FL ecosystem
 - The deployed model needs to have no discrimination against some individuals or groups
- Types of fairness in the setting of federated learning
 - Performance fairness (uniform accuracy distribution across participants)
 - Collaboration fairness (participants with higher contribution receive higher rewards/incentives)
 - Model fairness (protection of some specific attributes)

Performance Fairness

- Goal: Encourage a *uniform* accuracy distribution across participants
- Setting:
 - Data distribution across participants are heterogeneous
 - Only applies to horizontal FL
 - Participants do not commercially compete with each other
 - The output of FL is usually for social welfare
- Example:
 - FL across IoT devices sold by one company (Google Gboard on Android phones)
 - FL across provincial governments, hospitals

Collaboration Fairness

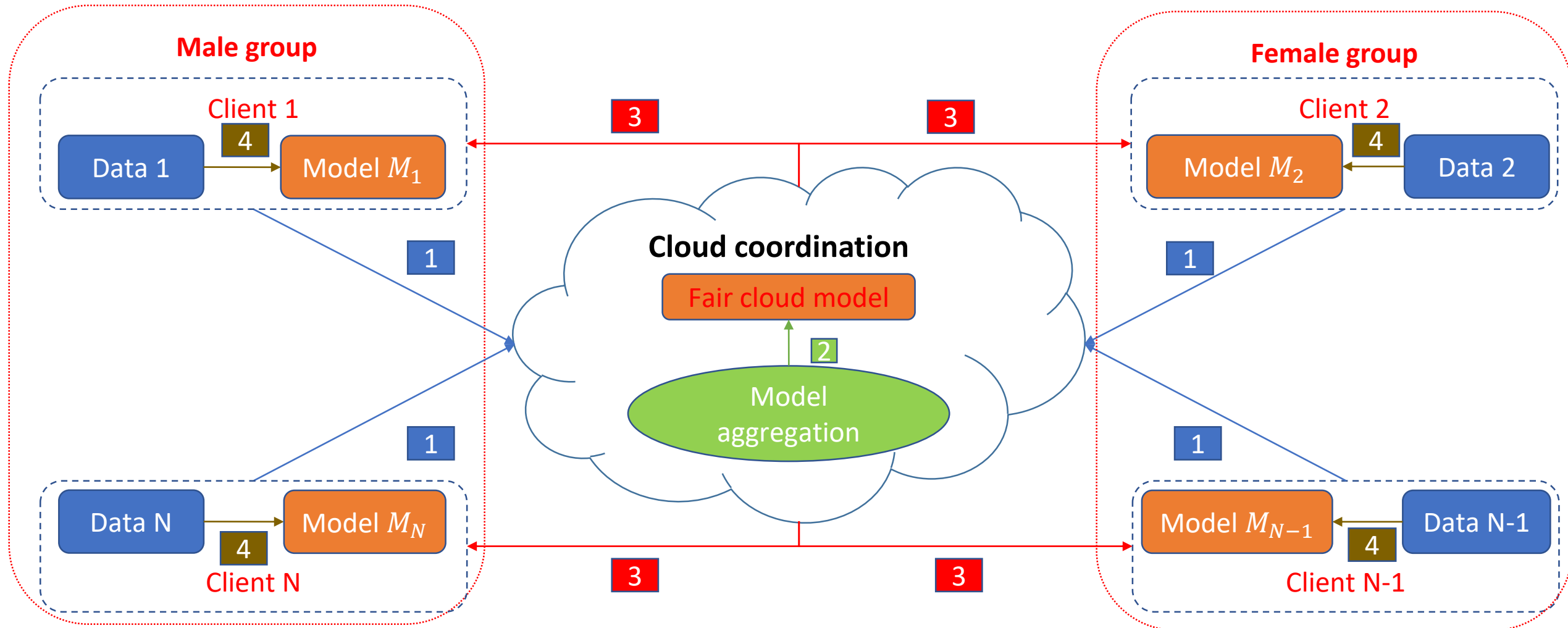
- Goal: Participants with higher *contribution* receive higher *rewards/incentives*
- Setting:
 - Data distribution across participants are heterogeneous
 - Applies to both horizontal FL and vertical FL
 - Participants are only self-interested and may commercially compete with each other
- Example:
 - FL across tech gains
 - FL across banks

Model Fairness

- Goal: The federated trained model has no discrimination against some specific individuals or groups.
- Setting:
 - Data distribution across participants can be i.i.d. or heterogeneous
 - Applies to both horizontal FL and vertical FL
 - Participants all agree that some attribute needs to be protected
- Example:
 - The trained model is deployed in advertising, credit, employment, education, criminal justice.

Relation between performance fairness and model fairness

- The requirement of performance fairness and model fairness has some overlap
- Example: Data of each client is from one group of the protected attribute.



Part III

Towards Performance Fairness in Federated Learning

Tutor: Zirui Zhou

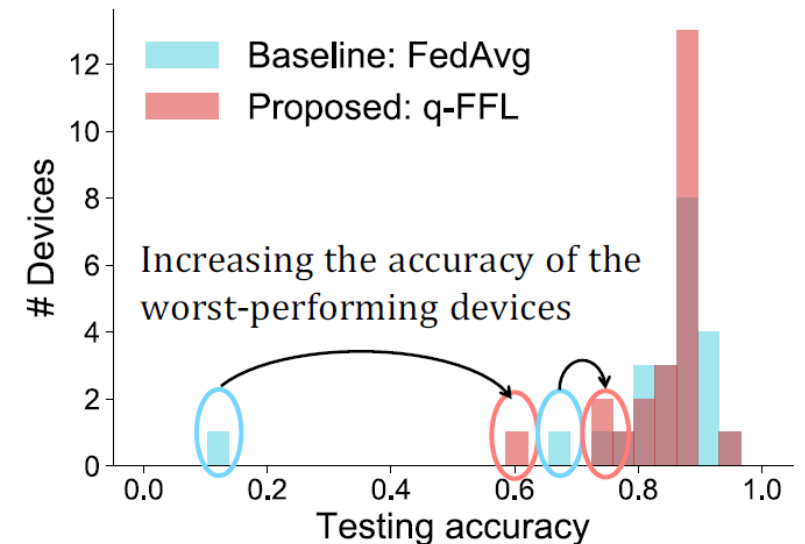
Fair Resource Allocation in FL

- [Li et al.'20] propose a q -Fair Federated Learning framework to achieve a more uniform accuracy distribution across participants.
 - A novel optimization objective inspired by fair resource allocation in wireless networks
 - A new communication-efficient method, q -FedAvg, to optimize the novel objective in FL

- The new optimization objective:

$$\min_w f_q(w) = \sum_{k=1}^m \frac{p_k}{q+1} F_k^{q+1}(w)$$

- Key idea: reweighting the local objectives
- The power $q \geq 0$ can be tuned by the coordinator
- Setting $q = 0$ reduces to normal objective in FL (no fairness)
- Trade-off between accuracy and fairness



Agnostic Federated Learning

- [Mohri et al.'19] proposed a new framework of *agnostic federated learning*.
 - FL model is optimized for *any* target distribution formed by a mixture of clients' distribution
 - The trained model does not overfit the data to any particular client at the cost of others

- The new optimization objective:

$$\min_w \left\{ \mathcal{L}(w) := \max \left\{ \sum_{i=1}^N \lambda_i F_i(w) \mid \lambda_1, \dots, \lambda_N \geq 0, \sum_{i=1}^N \lambda_i = 1 \right\} \right\}.$$

- $\mathcal{L}(w)$ is the worst-case loss formed by any mixture of clients' empirical distribution
- Minimize the worst-case objective
- AFL is a minmax optimization problem
- A simplified version of a stochastic Mirror-Prox algorithm proposed in [Juditsky et al.'11]

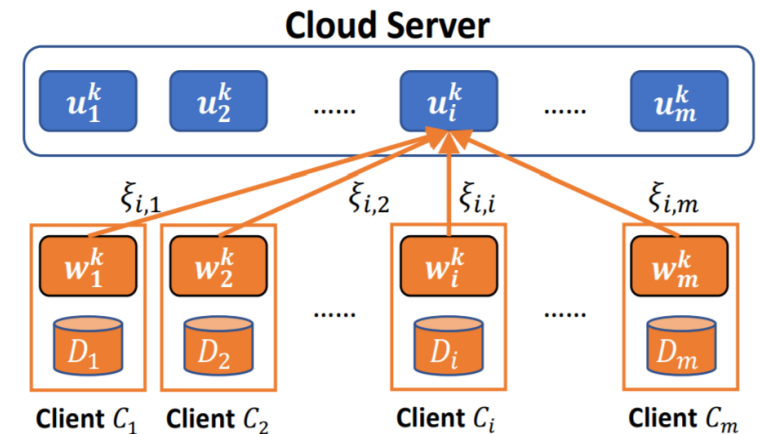
Performance Fairness via Personalization

- The root cause for non-uniform performance in FL is data heterogeneity
- Training personalized models is better than training a global model
- [Huang et al.'20] proposed an adaptive message passing framework for FL
 - Personalized model for each participant
 - Model aggregation is tailored for each client based on similarity

- Optimization objective:

$$\min_W \left\{ \mathcal{G}(W) := \sum_{i=1}^m F_i(\mathbf{w}_i) + \lambda \sum_{i < j}^m A(\|\mathbf{w}_i - \mathbf{w}_j\|^2) \right\}$$

- $A(\cdot)$ is an attention-inducing function
- Training algorithm is based on incremental optimization methods



Performance Fairness via Personalization

- The root cause for non-uniform performance in FL is data heterogeneity
- Training personalized models is better than training a global model
- [Li et al.'21] proposed Ditto, a scalable federated multi-task learning framework
 - *Key idea*: local training with regularization that encourages the personalized models to be close to the optimal global model

- Local objective:

$$\begin{aligned} \min_{v_k} \quad & h_k(v_k; w^*) := F_k(v_k) + \frac{\lambda}{2} \|v_k - w^*\|^2 \\ \text{s.t.} \quad & w^* \in \arg \min_w G(F_1(w), \dots, F_K(w)). \end{aligned} \quad (\text{Ditto})$$

- It is a lightweight personalization add-on for standard global FL
- Applies to both convex and non-convex objectives
- Reduce variance of the test accuracy across devices by $\sim 10\%$ in numerical experiments.

Part IV

Towards Collaboration Fairness in Federated Learning

Tutor: Zirui Zhou

Collaboration Fairness in FL

- Participating in FL incurs costs
 - Training data
 - Communication bandwidth
 - Computation power
 - Potential loss of competitiveness against rivals
 - Potential threat of information leakage and attacks
- Contribution to FL varies among participants
 - Data volume
 - Data quality
 - Training time / Communication rounds
 - Honesty
- Collaboration fairness in FL: each participant receives a reward that can fairly reflect its contribution to the FL system.

Core Questions in Collaboration Fairness

- How to measure the contribution of each participant?
- How to define reward to each participant?
- How to fairly distribute the total reward to each participant?

Core Questions in Collaboration Fairness

- How to measure the contribution of each participant?
- How to define reward to each participant?
- How to fairly distribute the total reward to each participant?

Classes of Contribution Evaluation Methods

- Naïve Evaluation Methods:
 - Data volume
 - Data variety
 - Communication rounds
- Marginal Utility Based Methods:
 - Individual
 - Leave-one-out
 - Shapley value
- Mutual Evaluation Based Methods:
 - Pairwise measurement

Naïve Evaluation Methods

- Relative data volume:
$$C_i = \frac{D_i}{\sum_{j \in \mathcal{N}} D_j}$$
 - Here, C_i is the contribution of client i to the FL system; D_i is the data volume of client i .
- Relative data volume and varieties:
$$C_i = \frac{D_i \cdot v_i}{\sum_{j \in \mathcal{N}} D_j \cdot v_j}$$
 - Here, v_i is the varieties of the data owned by client i , e.g., number of classes, range of target values.
- Relative data volume and communication rounds:
$$C_i = \frac{D_i \cdot T_i}{\sum_{j \in \mathcal{N}} D_j \cdot T_j}$$
 - Here, T_i is the number of communication rounds between client i and the coordinator.

Marginal Utility Based Methods: Basic Setup

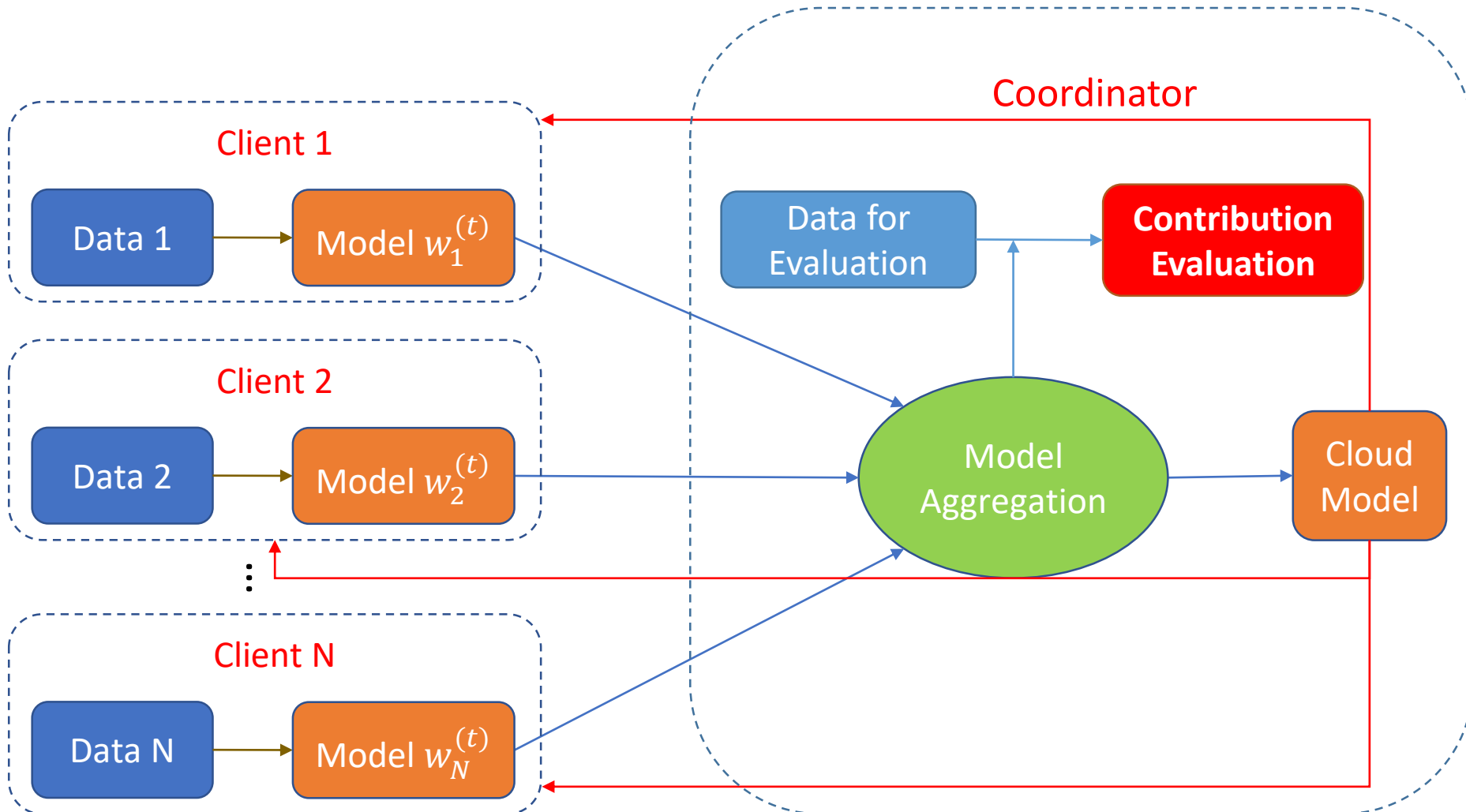
- Notation:

- $\mathcal{N} = \{1, 2, \dots, N\}$: the set of indices of participants
- $w_i^{(t)}$: the model received from client i at round t
- $w_S^{(t)}$: the model aggregated from all clients in $S \subseteq \mathcal{N}$ at round t
- T : number of communication rounds in federated learning
- $U(w)$: a function that measure the utility of model w

- Remarks:

- In horizontal FL, $w_S^{(t)}$ is usually a weighted average of $w_i^{(t)}, i \in S$
- In vertical FL, $w_S^{(t)}$ usually stacks $w_i^{(t)}, i \in S$
- An example of $U(w)$: accuracy of model w on a test data set that is held by the server/coordinator or a trusted third party

Architecture of Marginal Utility Based Methods



Individual Profit-Sharing Scheme

- Consider the utility if client i works alone: $U(\bar{w}_i)$
 - \bar{w}_i is the trained model if client i works individually
- The contribution of client i to the FL system can be defined as

$$C_i = \frac{U(\bar{w}_i)}{\sum_{j=1}^N U(\bar{w}_j)}$$

- A round-wise variant:
 - The contribution of client i to the FL system at the t -th round:

$$C_i = \frac{U(w_i^{(t)})}{\sum_{j=1}^N U(w_j^{(t)})}$$

Leave-One-Out

- Leave-one-out is also referred to as influence function evaluation
- Utility gain if client i joins the FL: $U(\bar{w}_{\mathcal{N}}) - U(\bar{w}_{\mathcal{N}\setminus\{i\}})$
 - $\bar{w}_{\mathcal{N}}$ is the final trained model if all N clients are participated
 - $\bar{w}_{\mathcal{N}\setminus\{i\}}$ is the final trained model if all N clients except client i are participated
- The contribution of client i to the FL system can be defined as the normalized relative utility gain if client i joins:

$$C_i = \frac{U(\bar{w}_{\mathcal{N}}) - U(\bar{w}_{\mathcal{N}\setminus\{i\}})}{\sum_{j \in \mathcal{N}} U(\bar{w}_{\mathcal{N}}) - U(\bar{w}_{\mathcal{N}\setminus\{j\}})}$$

- Approximation methods such as [Koh et al.'17] to overcome huge computational cost

A Round-Wise Variant

- Based on the naïve approach, [Nishio et al.'20] proposed a round-wise approach that eases the computation and communication overhead of the naïve approach.
- Utility gain if client i joins the FL at round t : $U(w_{\mathcal{N}}^{(t)}) - U(w_{\mathcal{N}\setminus\{i\}}^{(t)})$
- The round-wise contribution of client i to the FL system is the sum of gains that include client i in each round:

$$C_i = \frac{\sum_{t=0}^T U(w_{\mathcal{N}}^{(t)}) - U(w_{\mathcal{N}\setminus\{i\}}^{(t)})}{\sum_{j \in \mathcal{N}} \sum_{t=0}^T U(w_{\mathcal{N}}^{(t)}) - U(w_{\mathcal{N}\setminus\{j\}}^{(t)})}$$

Shapley Value for Data Valuation

- Shapley Value (SV) is a classic way in cooperative game theory to distribute total gains generated by the coalition of a set of players.
- SV has been increasingly used for valuing training data in machine learning.
- Pros: *SV uniquely* possesses a set of properties desired by data valuation:
 - Group rationality
 - Fairness
 - Additivity
- Cons: SV is extremely computationally expensive with complexity $O(N!)$
 - Approximation techniques : Monte Carlo, group-testing, probabilistic estimation [Jia et al.'19]

Shapley Value for Data Valuation (Cont'd)

- Consider a cooperative game with
 - $\mathcal{N} = \{1, 2, \dots, N\}$: set of N players
 - $\nu: 2^{\mathcal{N}} \rightarrow \mathcal{R}$: a utility function that describes the utility of any possible coalition
- SV of player i with respect to the utility function ν :

$$s_i^\nu = \frac{1}{N} \sum_{S \subseteq \mathcal{N} \setminus \{i\}} \frac{1}{\binom{N-1}{|S|}} [\nu(S \cup \{i\}) - \nu(S)]$$

- Example ([Zeng et al.'21]):

$$\begin{array}{ll}
 v(\emptyset) = 0 & v(\{0\}) = 5 \\
 v(\{1\}) = 10 & v(\{2\}) = 15 \\
 v(\{0, 1\}) = 30 & v(\{0, 2\}) = 40 \\
 v(\{1, 2\}) = 60 & v(\{0, 1, 2\}) = 100
 \end{array}$$

	0	1	2
0 \leftarrow 1 \leftarrow 2	5	25	70
0 \leftarrow 2 \leftarrow 1	5	60	35
1 \leftarrow 0 \leftarrow 2	20	10	70
1 \leftarrow 2 \leftarrow 0	40	10	50
2 \leftarrow 0 \leftarrow 1	25	60	15
2 \leftarrow 1 \leftarrow 0	40	45	15
sum	135	210	255
$\varphi(i)$	22.5	35	42.5

Shapley Value for FL

- Based on SV, [Wang et al.'20] proposed a **federated shapley value** that can be used to measure contribution in FL.

- In round t , the federated SV of client i is defined as

$$C_i^{(t)} = \frac{1}{|I_t|} \sum_{S \subseteq I_t \setminus \{i\}} \frac{1}{\binom{|I_t|-1}{|S|}} [U(w_{S \cup \{i\}}^{(t)}) - U(w_S^{(t)})], \quad \text{if } i \in I_t,$$
$$C_i^{(t)} = 0, \quad \text{otherwise.}$$

where I_t is the set of clients that are chosen to participate in round t .

- To measure the contribution of client i to the FL system, we can take the sum of the SV of client i across round 0 to round T :

$$C_i = \sum_{t=0}^T C_i^{(t)}$$

Mutual Evaluation Based Methods

- Participants evaluate each other based on reputation and credit.
 - Reduce the dependence on server/coordinator
 - Can be implemented in a decentralized manner, e.g., blockchain-based architecture
- [Lyu et al.'20] proposed a local credibility mutual evaluation mechanism to enforce collaboration fairness in FL
 - Each client keeps a local credit information of all the other clients
 - Local credibility is initialized by measuring the similarity between clients.
 - Each client generates artificial samples with a differential private GAN and computes the contribution of every other client by investigating the label similarities.
- [Kang et al.'19] also employed pairwise measurement of contribution
 - Each client holds a reputation score for all the other clients
 - Reputation score is updated by a multi-weight subjective logic model [Liu et al.'11]

Core Questions in Collaboration Fairness

- How to measure the contribution of each participant?
- How to define reward to each participant?
- How to fairly distribute the total reward to each participant?

Types of Reward To Participants

- Different participants may want to received different types of rewards.
- Common types of rewards in FL:
 - A model with good performance on self-interested test data set (personalized model)
 - A model with good performance on coordinator's data set (global model)
 - Monetary compensations
 - Computational power and other infrastructure resources
 - Reputation
 - Extra information, e.g., bias, variance
 - Model's attribute fairness

Core Questions in Collaboration Fairness

- How to measure the contribution of each participant?
- How to define reward to each participant?
- How to fairly distribute the total reward to each participant?
 - Monetary reward
 - Performance reward

Properties of A Fair Incentive Mechanism

- A list of desirable properties of a fair incentive mechanism in federated learning:
 1. Individual Rationality: all participants have non-negative profits
 2. Budget Balance: sum of payment for participants is no more than the total budget
 3. Collaboration Fairness: total profits of participants fairly reflect their contributions
 4. Early Contribution: encourage participants to contribute early to the FL system.
 5. Regret Distribution Fairness: the difference of regret among data owners is minimized
 6. Robustness: detect free-riders and exclude them from the FL system

A Simple Fair Incentive Mechanism

- Some notation:
 - $c_i(t)$: the *cost* of participant i in round t of FL
 - $q_i(t)$: the *contribution* of participant i in round t of FL
 - $B(t)$: the total *budget* of FL in round t
 - $u_i(t)$: the payoff to participant i in round t of FL

- A simple fair incentive mechanism:

- If $B(t)$ is fixed:

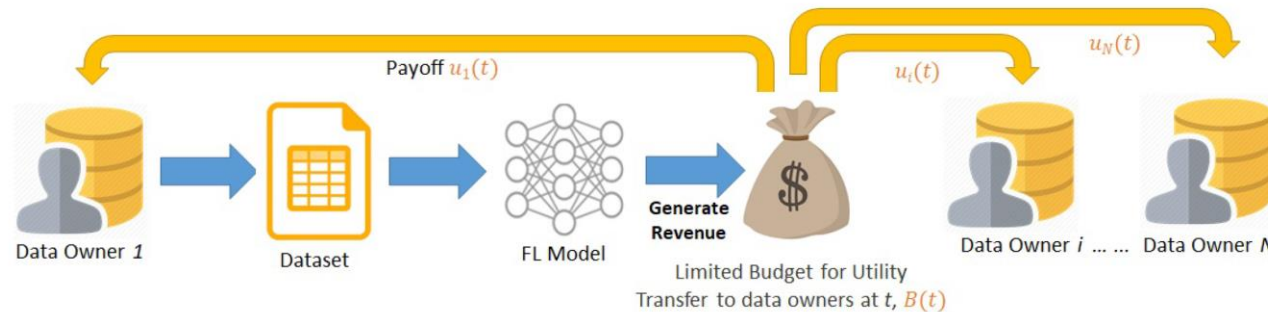
$$u_i(t) = \frac{q_i(t)}{\sum_{j=1}^N q_j(t)} B(t)$$

- If $B(t)$ is not fixed:

$$\hat{u}_i(t) = \frac{q_i(t)}{\sum_{j=1}^N q_j(t)}, \quad u_i(t) = \alpha \hat{u}_i(t), \quad u_i(t) \geq c_i(t) \quad \forall i = 1, \dots, N$$

FLI: A Fairness-Aware Incentive for FL

- [Yu et al.'20] proposed a federated learning incentivizer (FLI) payoff-sharing scheme



- Motivation and key idea:
 - The commercialization of the models takes time, resulting in delays in paying back the participants
 - Accounts for the fairness of distributing profit over time and minimize regrets
 - Minimize the fluctuation of data owners'
- Achieves contribution fairness, regret distribution fairness, and expectation fairness

FLI: A Fairness-Aware Incentive for FL

- Some notation:
 - $c_i(t)$: the *cost* of participant i in round t of FL
 - $q_i(t)$: the *contribution* of participant i in round t of FL
 - $B(t)$: the total *budget* of FL in round t
 - $u_i(t)$: the payoff to participant i in round t of FL
- $Y_i(t)$: difference between received and supposed to receive (**total regret**)
 - The dynamics of $Y_i(t)$:
$$Y_i(t + 1) \triangleq \max[Y_i(t) + c_i(t) - u_i(t), 0]$$
 - A large value of $Y_i(t)$ indicates that i has not been adequately compensated
- $Q_i(t)$: indicates how long a participant has been waiting to receive the full payoff (**temporal regret**)
 - The dynamics of $Q_i(t)$:
$$Q_i(t + 1) \triangleq \max[Q_i(t) + \lambda_i(t) - u_i(t), 0] \quad \lambda_i(t) = \begin{cases} \hat{c}_i, & \text{if } Y_i(t) > 0 \\ 0, & \text{otherwise.} \end{cases}$$
 - $Q_i(t)$ will increase as long as $Y_i(t)$ is not empty

FLI: A Fairness-Aware Incentive for FL

- The whole system maximize a “value-minus-regret drift” objective over time

- At time t :

- For each participant i , compute profit share:

$$s_i(t) = \omega q_i(t) + Y_i(t) + c_i(t) + Q_i(t) + \lambda_i(t)$$

- The coordinator assigns reward to each participant i and update total regret and temporal regret:

$$u_i(t) = \frac{s_i(t)}{\sum_{j=1}^N s_j(t)} B(t)$$

$$Y_i(t+1) \triangleq \max[Y_i(t) + c_i(t) - u_i(t), 0]$$

$$Q_i(t+1) \triangleq \max[Q_i(t) + \lambda_i(t) - u_i(t), 0]$$

- Remarks

- FLI is a general fair incentive mechanism, works with any definition of contribution and cost
- FLI works for monetary rewards

RFFL: Robust and Fair FL Framework

- [Xu et al.'20] proposed a robust and fair FL framework
- Key idea: Participants with higher contribution receive better model
 - Model accuracy/performance is used as rewards for FL participants (also see [Sim et al.'20], [Lyu et al.'20])
- Quantification of collaboration fairness: Pearson correlation coefficient [Lyu et al.'20]
 - Standalone model accuracies: $\mathbf{x} = \{sacc_1, \dots, sacc_n\}$
 - Final FL model accuracies: $\mathbf{y} = \{acc_1, \dots, acc_n\}$
 - Collaboration fairness metric:
$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y}$$
 - Range of fairness metric ranges from -1 to 1. The higher value, the better collaboration fairness

RFFL: Robust and Fair FL Framework

- Three key ingredients in RFFL:
 - Reputation calculation based on cosine similarity between gradients
 - Reputation-weighted aggregation: $\Delta \mathbf{w}_g^{(t)} = \sum_{i \in R} r_i^{(t-1)} \Delta \mathbf{w}_i^{(t)}$
 - Reputation-based quota to determine number of entries in the gradient vector to allocate to each participant

- Update of reputation r_i for participant i :

$$r_i^{(t)} \leftarrow \text{cos_sim}(\Delta \mathbf{w}_g^{(t)}, \Delta \mathbf{w}_i^{(t)})$$

$$r_i^{(t)} \leftarrow r_i^{(t-1)} * \alpha + r_i^{(t)} * (1 - \alpha)$$

- Gradient allocation for participant i :

$$quota_i \leftarrow \frac{r_i^{(t)}}{\max(\mathbf{r}^{(t)})} * |\Delta \mathbf{w}_g^{(t)}|$$

$$\Delta \mathbf{w}_{*i}^{(t)} \leftarrow \text{largest}(\Delta \mathbf{w}_g^{(t)}, quota_i) - r_i^{(t-1)} \Delta \mathbf{w}_i^{(t)}$$


- Remove participants with too low contribution: if $r_i^{(t)} < r_{th}$, then remove participant i

Hierarchically Fair Federated Learning

- [Zhang et al.'20] proposed a hierarchically fair FL framework

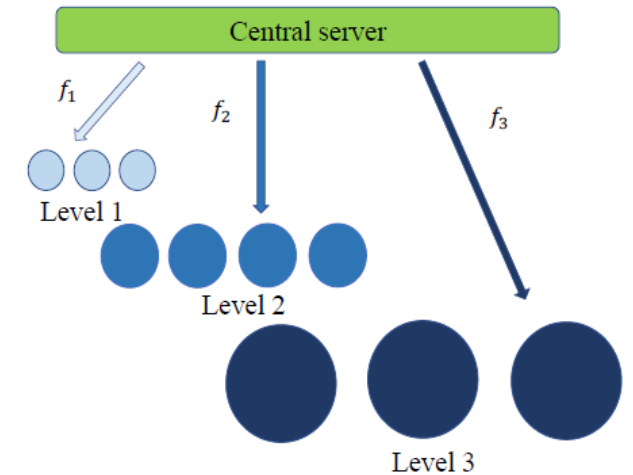
- Main idea:

- Participants are categorized into L contribution levels
- Each level learns a model w_l
- Higher-level agents contribute the same amount of data as level l
- Lower-level agents should contribute all the data they have

- Training data for Level 1: 

- Training data for Level 2: 

- Training data for Level 3: 



Conclusion

- Collaboration fairness is indispensable for sustainable FL
- The research on collaboration fairness grows rapidly in the past two years
- An inter-disciplinary direction
 - Data valuation / cooperative game theory / marketing
- Many open problems in this direction
 - Vertical federated learning
 - Lightweight incentive mechanism
 - Enhance the overall performance of the FL system
 - ...

Part V

Towards Model Fairness in Federated Learning

Tutors: Lanjun Wang, Lingyang Chu, Changxin Liu

Model Fairness in Machine Learning

Tutor: Lanjun Wang

Core Questions in Model Fairness

- Why we care about model fairness?
- How to define model fairness?
 - How to choose proper existing model fairness notations in real applications?
- How to produce a fair model?

Motivation Examples

	WHITE	AFRICAN AMERICAN
Labeled Higher Risk, But Didn't Re-Offend	23.5%	44.9%
Labeled Lower Risk, Yet Did Re-Offend	47.7%	28.0%

Overall, Northpointe's assessment tool correctly predicts recidivism 61 percent of the time. But blacks are almost twice as likely as whites to be labeled a higher risk but not actually re-offend. It makes the opposite mistake among whites: They are much more likely than blacks to be labeled lower risk but go on to commit other crimes. (Source: ProPublica analysis of data from Broward County, Fla.)

Example from COMPAS[Agnwin et al.'16]

Motivation Examples

	WHITE	AFRICAN AMERICAN			
Labeled Higher Risk, But Didn't Re-Offend					
Labeled Lower Risk, Yet Did Re-Offend					
<i>Overall, Northpointe's assessment tool correctly predicts recidivism as whites to be labeled a higher risk but not actually re-offend. more likely than blacks to be labeled lower risk but go on to commit crimes (Broward County, Fla.)</i>					
	Search Query	Work Experience	Education Experience	Candidate	Xing Ranking
	Brand Strategist	146	57	male	1
	Brand Strategist	327	0	female	2
	Brand Strategist	502	74	male	3
	Brand Strategist	444	56	female	4
	Brand Strategist	139	25	male	5
	Brand Strategist	110	65	female	6
	Brand Strategist	12	73	male	7
	Brand Strategist	99	41	male	8
	Brand Strategist	42	51	female	9
	Brand Strategist	220	102	female	10
			...		
	Brand Strategist	3	107	female	20
	Brand Strategist	123	56	female	30
	Brand Strategist	3	3	male	40

TABLE I: Top k results on www.xing.com (Jan 2017) for an employer's job search query "Brand Strategist".

Motivation Examples

Labeled Higher Risk, But Didn't Re-C

Labeled Lower Risk, Yet Did Re-Offer

Overall, Northpointe's assessment tool correctly p
 as whites to be labeled a higher risk but not actual
 more likely than blacks to be labeled lower risk bu
 Broward County, Fla.)

		WHITE	AFRICAN AMERICAN								
		AFRICA		AVERAGE FACES		EUROPE					
Search Que	RWANDA										FINLAND
Brand Strate	SENEGAL										ICELAND
Brand Strate	S.AFRICA										SWEDEN
Brand Strate		MALE	FEMALE	MALE	FEMALE	FEMALE	MALE	FEMALE	MALE		

TABLE I: Top employer's job

Figure 1: Example images and average faces from the new Pilot Parliaments Benchmark (PPB). As the examples show, the images are constrained with relatively little variation in pose. The subjects are composed of male and female parliamentarians from 6 countries. On average, Senegalese subjects are the darkest skinned while those from Finland and Iceland are the lightest skinned.

[Buolamwini et al.'19]

Core Questions in Model Fairness

- Why we care about model fairness?
- How to define model fairness?
 - How to choose proper existing model fairness notations in real applications?
- How to produce a fair model?

Definition

Fairness is the *absence* of any prejudice or favoritism toward an *individual* or a *group* based on their inherent or acquired *characteristics*.

-- from [Mehrabi et al.'21]

Definition

A : Protected Attributes

Y : Ground truth (binary) output

\hat{Y} : Predicted (binary) output

Individual Fairness: Give similar predictions to similar individuals

Group Fairness: Treat different groups equally

- Statistical Parity [Dworkins et al.'16]:

$$P(\hat{Y}|A = 0) = P(\hat{Y}|A = 1)$$

- Equalized Odds [Hardt et al.'16]:

$$P(\hat{Y} = 1|A = 0, Y = y) = P(\hat{Y} = 1|A = 1, Y = y), \quad y \in \{0,1\}$$

- Predictive Parity [Chouldechova et al.'17]:

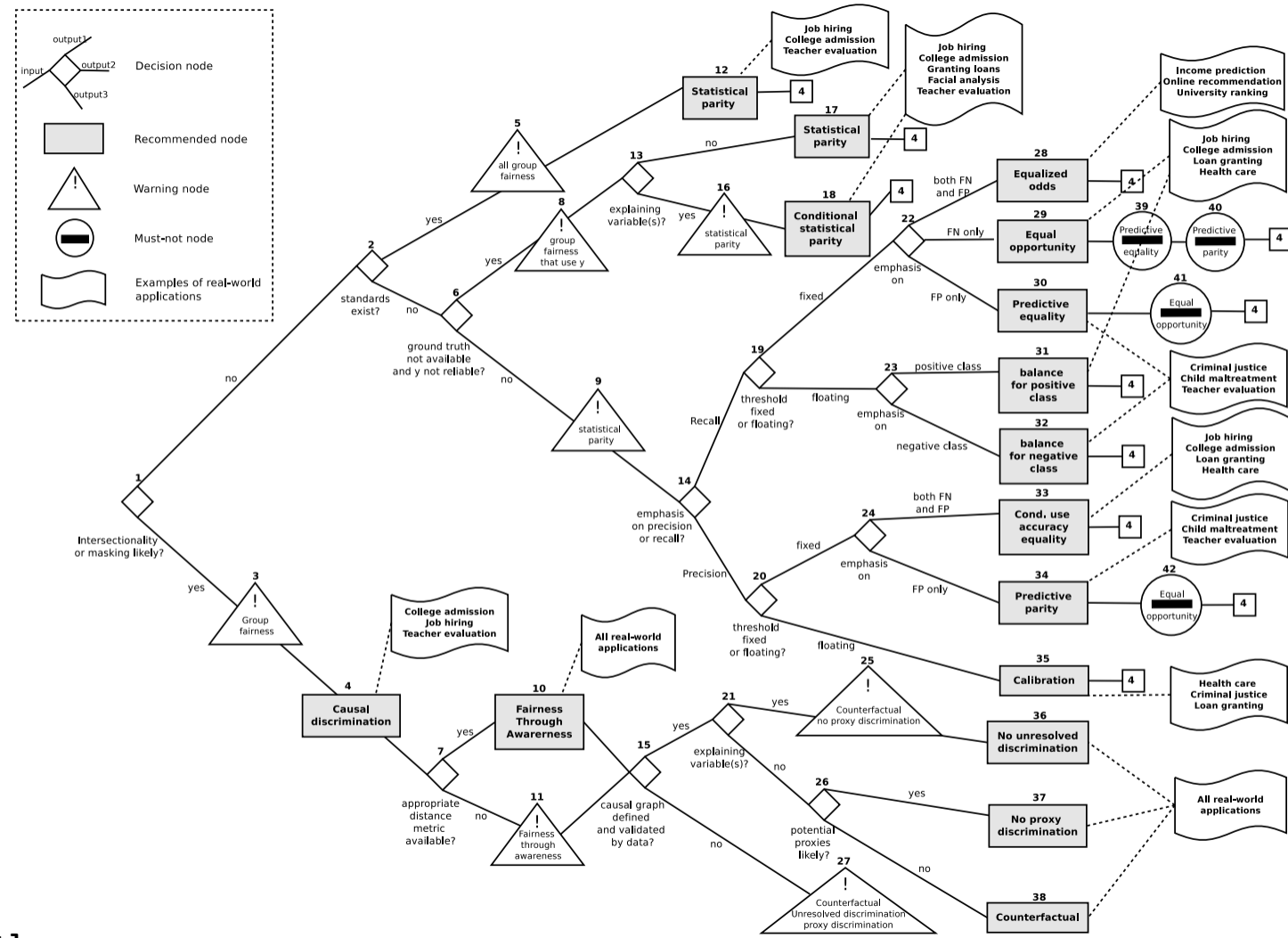
$$P(Y = 1|A = 0, \hat{Y} = 1) = P(Y = 1|A = 1, \hat{Y} = 1)$$

Fairness Measurement in Real Applications

- **Ground truth available**
- **Emphasis on both FP and FN**
- **Fixed threshold**

Equalized Odds

Income Prediction
Online Recommendation
University Ranking



[Makhlouf et al.'21]

Figure 1: Fairness notions applicability decision diagram

Core Questions in Model Fairness

- Why we care about model fairness?
- How to define model fairness?
 - How to choose proper existing model fairness notations in real applications?
- How to produce a fair model?

Fairness Enhancement in ML Pipeline

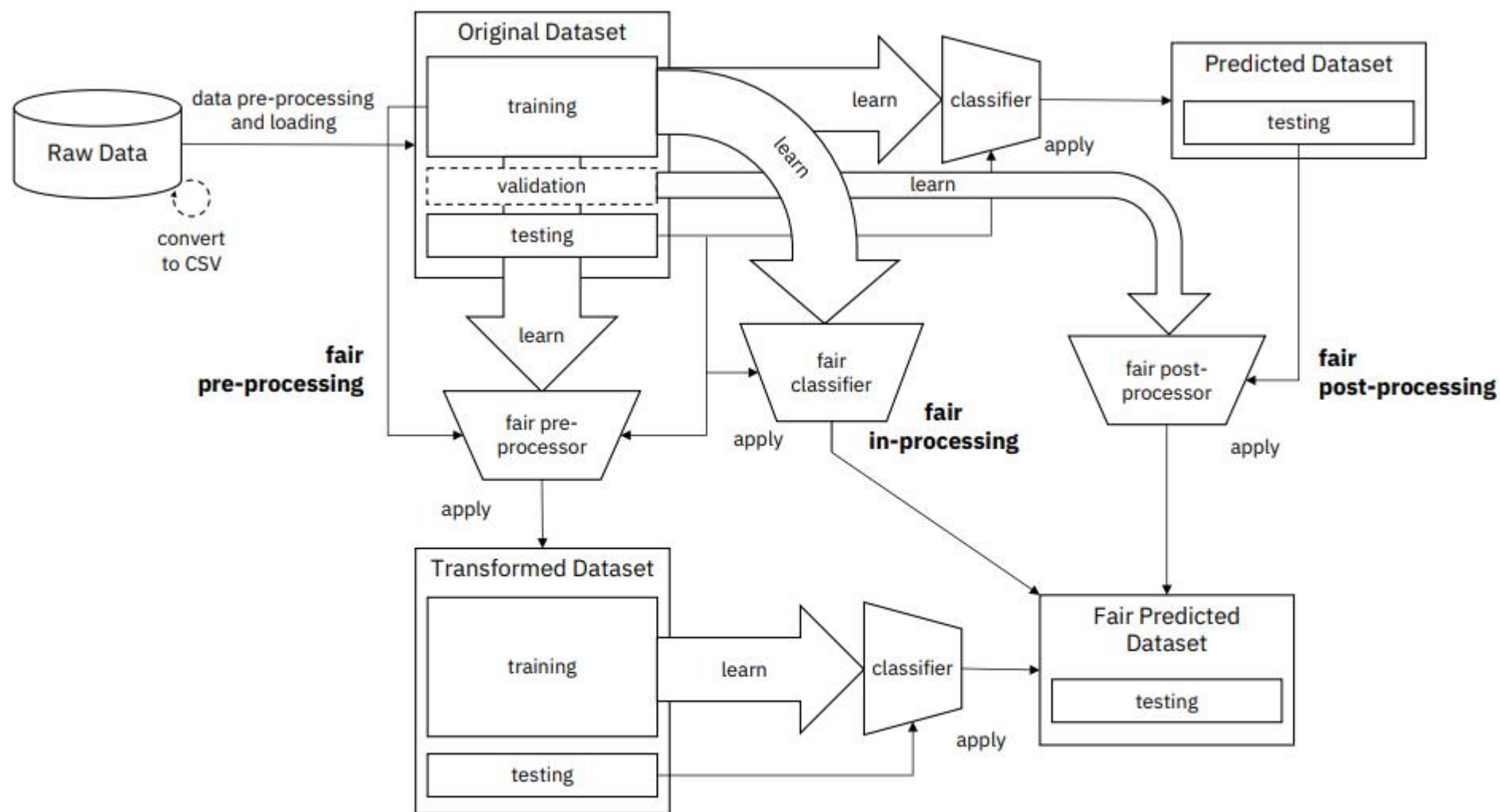


Figure 1. The fairness pipeline. An example instantiation of this generic pipeline consists of loading data into a dataset object, transforming it into a fairer dataset using a fair pre-processing algorithm, learning a classifier from this transformed dataset, and obtaining predictions from this classifier. Metrics can be calculated on the original, transformed, and predicted datasets as well as between the transformed and predicted datasets. Many other instantiations are also possible.

Fairness Enhancement Approaches

- Pre-process (data)
 - Modify the features in the dataset [Feldman et al.'15]
 - Generate fair synthetic data from the initial input data [Abusitta et al.'19]
- In-process (algorithm)
 - Constraint/Penalty term/Regularization [Kamishima et al.'12;Zafar et al. '17ab;Berk et al. 2017]
 - Declaration system [Zhang et at.'21]
- Post-process (predicted results)[Corbett-Davies et al.'17; Dwork etal. '18; Hard et al. '16; Menon et al. '18]

Fair Preprocessing: Unfairness caused by Data Transformers in ML Pipeline

Intuition:

- Most of research on fairness only considers a single classifier
- What are the fairness impacts of the preprocessing stages in machine learning pipeline?

Findings (unfairness pattern):

1. Data filtering and missing value removal change the data distribution and hence introduce bias in ML pipeline.
2. New feature generation or feature transformation can have large impact on fairness.
3. Encoding techniques should be chosen cautiously based on the classifier.
4. The variability of fairness of preprocessing stages depend on the dataset size and overall prediction rate of the pipelines.
5. The unfairness of a preprocessing stage can be dominated by dataset or the classifier used in the pipeline.
6. Among all the transformers, applying sampling technique exhibits most unfairness.
7. Selecting subset of features often increase unfairness
8. In most of the pipelines, feature standardization and non-linear transformers are fair transformers.

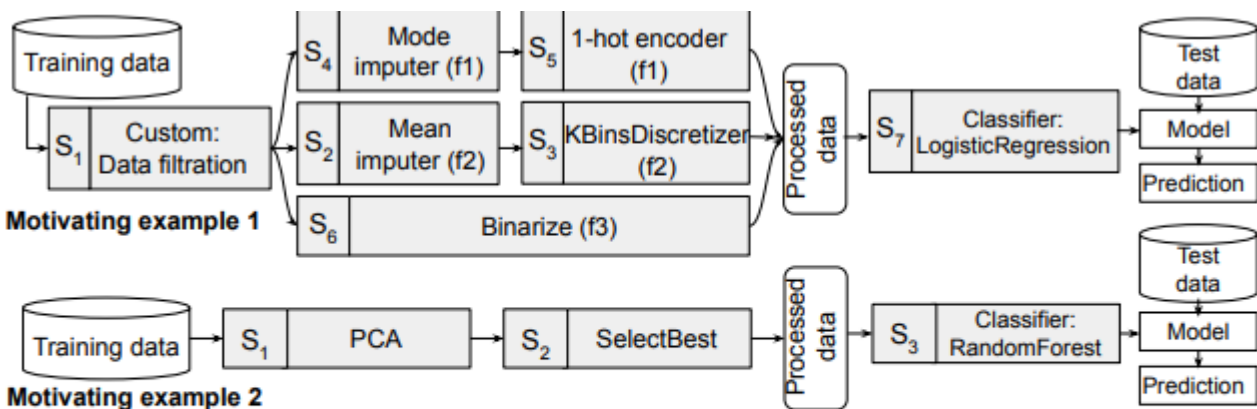


Figure 1: ML pipelines for the motivating examples, having a sequence of preprocessing stages followed by a classifier.

OmniFair: A Declarative System for Model-Agnostic Group Fairness in Machine Learning

```
Fairness Constraint

def grouping(Dataset D)
  groups = {}

  # user code here with example
  groups["African-American"] = []
  groups["Caucasian"] = []
  for i in range(D.shape(0)):
    groups[D[i]['race']].append(i)

  return groups

def fairness_metric(Group G, Classifier h)
  coefficients = np.zeros(len(G)+1)

  # user code here with example
  for i in range(G.shape(0)):
    coefficients[i] = 1.0/len(G)

  num = coefficient[0]
  for i in range(G.shape(0)):
    num += coefficient[i] X  $\mathbb{I}(h(x_i)=y_i)$ 
  return (coefficient, num)

a fairness tolerance level  $\epsilon$ 
```

Figure 1: The declarative interface.

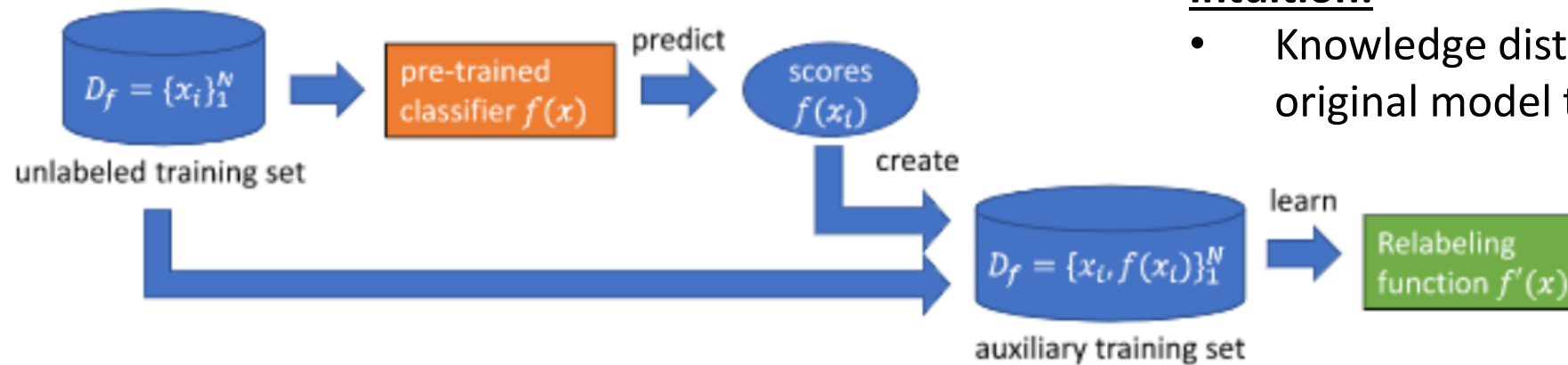
Intuition:

- Leverage declarative interface to specify desired group fairness constraints

Contributions:

- **Versatile:** user only need to specify the desired fairness constraints and the ML algorithm
- **High Quality:** accuracy-fairness trade-off
- **Efficient:** efficient algorithms for hyperparameter tuning

Fairness improvement for black-box classifiers with Gaussian process



Intuition:

- Knowledge distillation to leverage original model to maintain accuracy

Approach:

Train a substitute model $f'(x)$ with unlabeled samples, and make it to achieve two goals:

- (1) maximize the fairness and
- (2) minimize the difference between the relabeling function and the pretrain model.

$$\operatorname{argmin}_{f'(x)} \underbrace{\frac{1}{N} \sum_{i=1}^N |f'(x_i) - f(x_i)|}_{\text{Term-1}} + \underbrace{|P(\hat{y}' = 1 | S = 1) - P(\hat{y}' = 1 | S = 0)|}_{\text{Term-2}}$$

Contributions:

- No need on true label of training samples
- Provide a theoretical analysis to derive an upper bound on accuracy loss because it uses a GP model to set up $f'(x)$.

Other Great Model Fairness Tutorials

- Fairness in machine learning
 - NeurIPS'17
- Defining and Designing Fair Algorithms
 - EC'18, ICML'18
- Fairness-Aware Machine Learning in Practice
 - WSDM'19, WWW'19, KDD'19
- Fairness in Healthcare
 - KDD'20
- Fairness of Machine Learning in Recommender Systems
 - SIGIR'21

Challenges of Training Fair Models in *Horizontal* Federated Learning

Tutor: Lingyang Chu

Why Training Fair Models in Federated Learning?

- Collaboration and fairness are among the top priorities in machine learning and AI applications.
 - Collaboration -> Federated Learning
 - Build effective machine learning models for sophisticated applications.
 - Each collaborating party has to contribute its own private data while preserving data privacy.
 - Fairness -> Fair Model Training
 - Training models that are fair with respect to different data instances and features.
 - Building fair models that are ethically correct and trustworthy is largely required in high-stake applications, such as justice and medical health.

Two Types of Federated Fair Model Training

- Type I: Horizontal Federated Fair Model Training
 - Training fair models in a horizontal federated learning framework.
 - Discussed in this part of the tutorial.

- Type II: Vertical Federated Fair Model Training
 - Training fair models in a vertical federated learning framework.
 - Discussed in the next part of this tutorial.

Can We Extend Existing Approaches to Tackle This Problem?

- Existing fair model training methods (A survey paper [Kleinberg et al.'18])
 - Pre-processing methods [Calmon et al.'17, Feldman et al.'15, Louizos et al.'15, Quadrianto et al.'17, Zemel et al.'13]
 - Enhancing fairness of models by rectifying training data.
 - Post-processing methods [Corbett-Davies et al.'17, Dwork et al.'18, Hardt et al.'16, Menon et al.'18]
 - Revise the prediction scores of a machine learning model to make predictions fairer.
 - In-processing methods [Zafar et al.'17, Kamishima et al.'12, Woodworth et al.'17, Kamiran et al.'10]
 - Customize machine learning algorithms to directly train fair models.

Can We Extend Existing Approaches to Tackle This Problem?

- Existing fair model training methods (A survey paper [Kleinberg et al.'18])
 - Pre-processing methods [Calmon et al.'17, Feldman et al.'15, Louizos et al.'15, Quadrianto et al.'17, Zemel et al.'13]
 - Enhancing fairness of models by rectifying training data.
 - Post-processing methods [Corbett-Davies et al.'17, Dwork et al.'18, Hardt et al.'16, Menon et al.'18]
 - Revise the prediction scores of a machine learning model to make predictions fairer.
 - In-processing methods [Zafar et al.'17, Kamishima et al.'12, Woodworth et al.'17, Kamiran et al.'10]
 - Customize machine learning algorithms to directly train fair models.

Most of these methods assumes a unified available training dataset, which infringes data privacy, thus is not available in a federated learning framework.

Can We Extend Existing Approaches to Tackle This Problem?

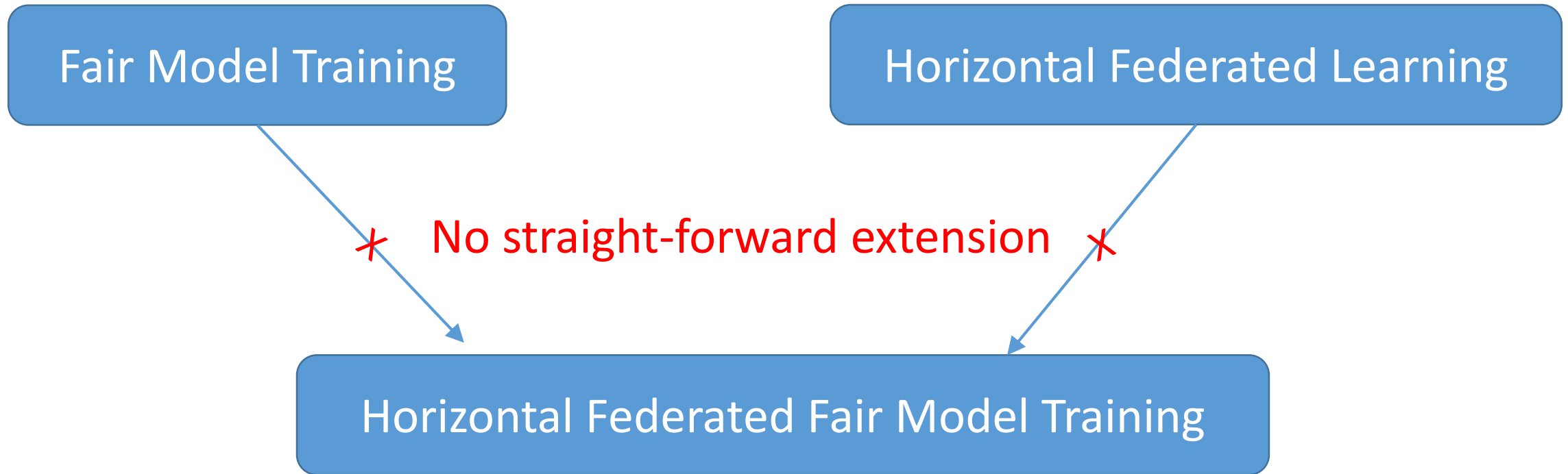
- Existing horizontal federated learning methods (A survey paper [Kairouz et al.'19])
 - Classical horizontal federated learning [McMahan et al.'17, Yang et al.'19, Chen et al.'17, Huang et al.'21]
 - Focus on protecting data privacy but ignore model fairness.
 - Collaboration fairness methods [Mohri et al.'19, Yang et al.'17, Gollapudi et al.'17, Yu et al.'20]
 - Focus on balancing the rewards paid to the participating parties.
 - Paying fair rewards to parties is substantially different from training fair models.
 - Other methods
 - AgnosticFair [Du et al.'21]: trains fair models at the cost of data privacy infringement.
 - Agnostic federated learning [Mohri et al.'19]: mitigates the training procedure bias but cannot guarantee a good model fairness.

Can We Extend Existing Approaches to Tackle This Problem?

- Existing horizontal federated learning methods (A survey paper [Kairouz et al.'19])
 - Classical horizontal federated learning [McMahan et al.'17, Yang et al.'19, Chen et al.'17, Huang et al.'21]
 - Focus on protecting data privacy but ignore model fairness.
 - Collaboration fairness methods [Mohri et al.'19, Yang et al.'17, Gollapudi et al.'17, Yu et al.'20]
 - Focus on balancing the rewards paid to the participating parties.
 - Paying fair rewards to parties is substantially different from training fair models.
 - Other methods
 - AgnosticFair [Du et al.'21]: trains fair models at the cost of data privacy infringement.
 - Agnostic federated learning [Mohri et al.'19]: mitigates the training procedure bias but cannot guarantee a good model fairness.

**Most of these methods do not consider training fair models.
Some works tried but cannot achieve data privacy and fairness at the same time.**

Challenges of Horizontal Federated Fair Model Training



Why is it so difficult to train a fair model in a horizontal federated learning framework?

Challenge I: An Intrinsic Conflict

- Horizontal federated fair model training is challenging due to the intrinsic conflict between fair model training and federated learning.
 - Fair model training
 - Accurately evaluating the fairness of a model requires access to the data of all parties.
 - Federated learning
 - Preserving data privacy forbids access to the private data of all parties.

Challenge I: An Intrinsic Conflict

- Horizontal federated fair model training is challenging due to the intrinsic conflict between fair model training and federated learning.
 - Fair model training -> **I want all the data.**
 - Accurately evaluating the fairness of a model requires access to the data of all parties.
 - Federated learning -> **Oh sorry, I cannot help with that.**
 - Preserving data privacy forbids access to the private data of all parties.

An intrinsic conflict!

Challenge II: Local Fairness Estimation Does NOT Work.

- Can we estimate the model fairness locally on each participating party?
 - The answer is NO, due to the following reasons.
 - Measuring model fairness locally is inaccurate due to the lack of data on each participating party.
 - Applying fairness constraints locally on each client will lead to inferior fairness performance or null solutions due to the inaccurate model fairness measurements computed locally and the resulting conflicts between local fairness constraints.

Challenge III: Difficulty in Finding a Proper Fairness Measurement

- Existing fairness measurements can be categorized into two classes
 - Individual fairness [Dwork et al.'12, Joseph et al.'16]
 - Measures the fairness of a model by evaluating how likely a pair of similar instances receives similar predictions from the model.
 - Well recognized to be less stable than group fairness.

Challenge III: Difficulty in Finding a Proper Fairness Measurement

- Existing fairness measurements can be categorized into two classes
 - **Group fairness** [Feldman et al.'15, Hardt et al.'16, Donini et al.'18, Calders et al.'09]
 - More stable than individual fairness.
 - Some measurements [Feldman et al.'15, Dwork et al.'12, Calders et al.'09, Calders et al.'10] are not suitable for training machine learning models due to the non-smooth and non-differentiable characteristics.
 - Some measurements [Du et al.'20, Wu et al.'19] are converted to be smooth and differentiable at the cost of approximation errors.
 - Other measurements [Xu et al.'20, Cotter et al.'19] requires to access all the data, which is unavailable in horizontal federated learning..

Conclusion

- It is important to train fair models in the context of horizontal federated learning.
- Horizontal federated fair model training has many challenges
 - The intrinsic conflict between accurately measuring model fairness and preserving data privacy.
 - Local fairness estimation per party does not work.
 - Even finding a proper fairness constraint for training is difficult.

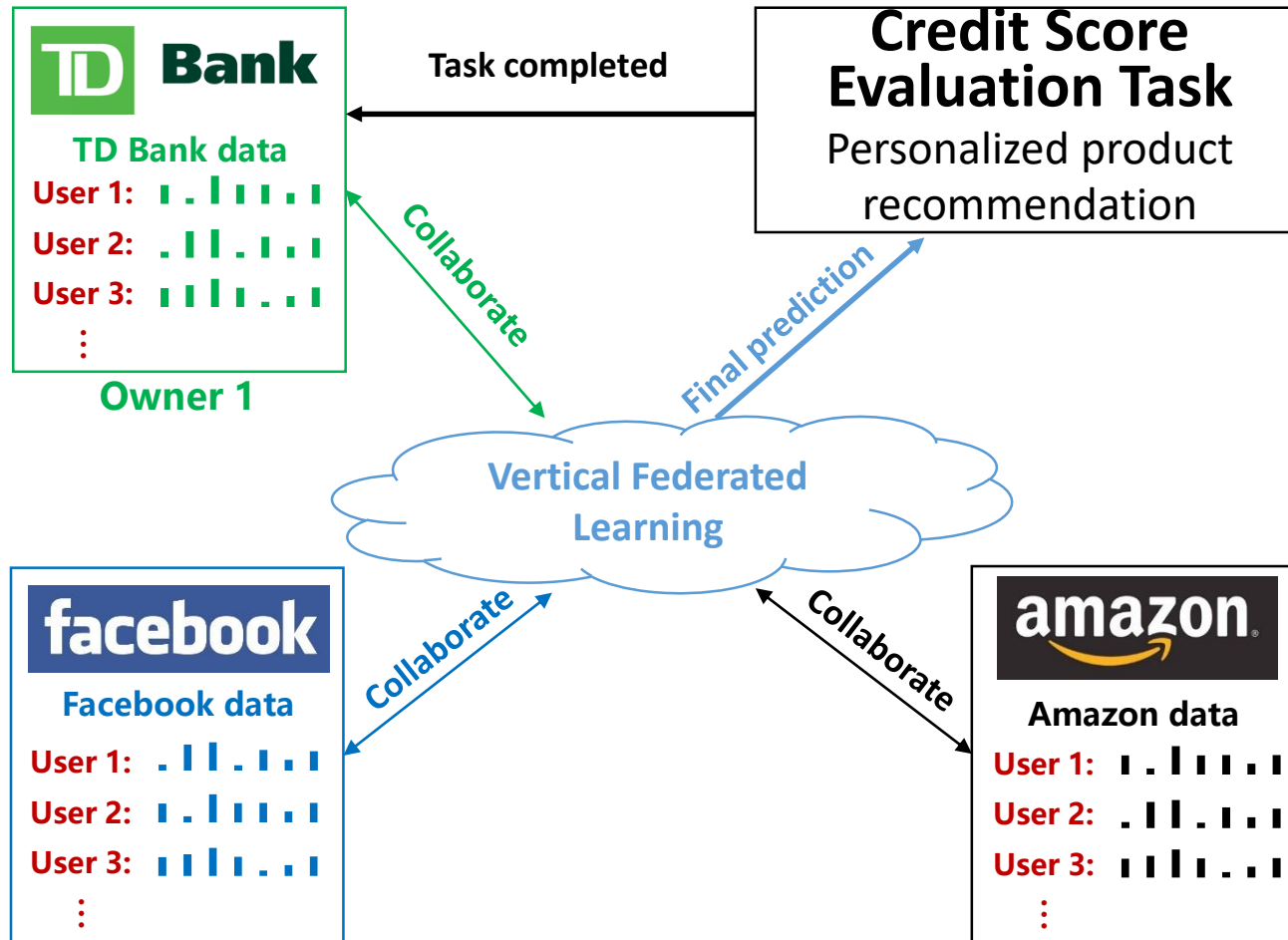
Challenges of Training Fair Models in *Vertical* Federated Learning

Tutor: Changxin Liu

Two Types of Federated Fair Model Training

- Type I: Horizontal Federated Fair Model Training
 - Training fair models in a horizontal federated learning framework.
 - Discussed in the last part of the tutorial.
- Type II: Vertical Federated Fair Model Training
 - Training fair models in a vertical federated learning framework.
 - Discussed in this part of the tutorial.

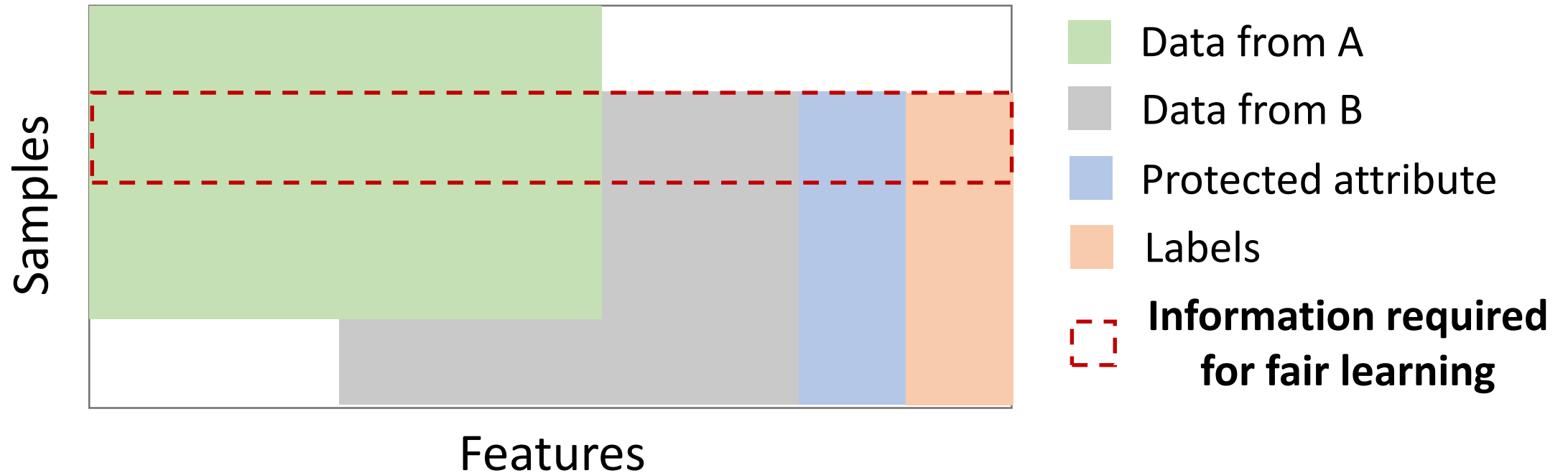
An Example of Training Fair Models within VFL



Credit score evaluation task:

- The bank wants to improve the evaluation accuracy of user credit scores by using the purchasing data from the retailer and the social relation data from the social network company.
- Financial institutions increasingly rely on vertical federated machine learning to support accurate decision-making
- Automated decision-making may discriminate historically disadvantaged groups, asking for algorithmic fairness within VFL

Fair VFL Systems



- Private features are vertically partitioned over multiple clients
- One or parts of the (active) clients have the protected attributes and labels
- Computational resources within clients are imbalanced

Due to these unique features, enhancing fairness in VFL can be very challenging

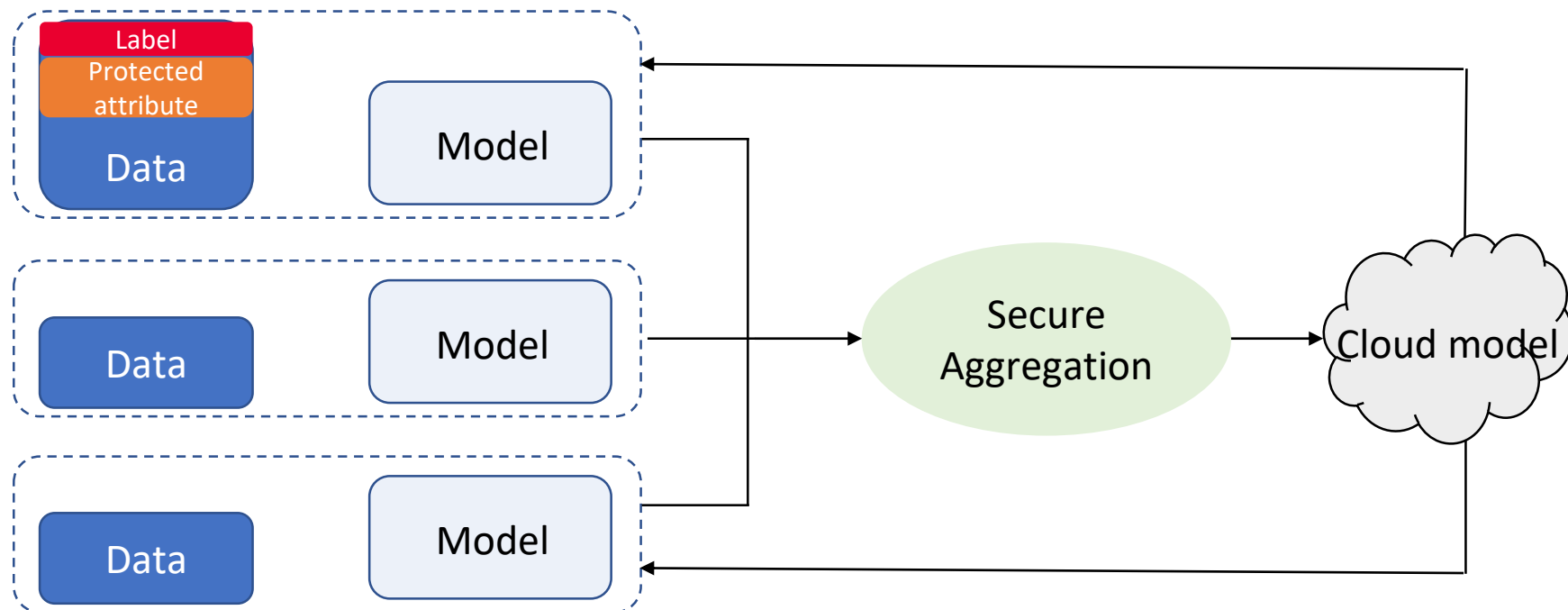
Can We Extend Existing Approaches to Tackle This Problem?

- Existing fair model training methods
 - Pre-processing methods [Calmon et al.'17, Feldman et al.'15, Kamiran et al.'12]
 - Enhancing fairness of models by rectifying training data.
 - Post-processing methods [Hardt et al.'16, Pleiss et al.'17]
 - Revise the prediction scores of a machine learning model to make predictions fairer.
 - In-processing methods [Donini et al.'18, Agarwal et al.'19]
 - Customize machine learning algorithms to directly train fair models.

Most of these methods assumes a unified available training dataset, which infringes data privacy, thus is not available in a federated learning framework.

Why Post-processing Methods Do Not Work?

- Post-processing methods [Hardt et al.'16, Pleiss et al.'17]
 - Step 1: Train a machine learning model (feasible in VFL)
 - Step 2: Adjust the FN and FP scores based on features, labels and protected attributes (NOT feasible in VFL)



Why In-processing Methods Do Not Work?

- In-processing methods [Donini et al.'18, Agarwal et al.'19]
 - Customize machine learning algorithms to directly train fair models.

Optimization problem in fair model training:

Both the objective and constraint are jointly defined by multiple clients

- Loss function

$$L(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^n \{L_i(\boldsymbol{\theta}) := l(X_i^T \boldsymbol{\theta}, y_i)\}$$

- Fairness constraint

$$\mathbb{P}\{X_i^T \boldsymbol{\theta} > \mathbf{0} | \mathbf{y} = \mathbf{1}, \mathbf{s} = \mathbf{a}\} = \mathbb{P}\{X_i^T \boldsymbol{\theta} > \mathbf{0} | \mathbf{y} = \mathbf{1}, \mathbf{s} = \mathbf{b}\}$$

X_i : Training samples

y_i : Binary output

s : Protected attributes

Lack of information to define the problem within each individual client.

Can We Extend Existing Approaches to Tackle This Problem?

- Existing federated learning methods (A survey paper [Kairouz et al.'19])
 - Classical vertical federated learning [Yang et al.'19]
 - Focus on protecting data privacy but ignore model fairness.
 - Fair horizontal federated learning
 - Agnostic federated learning [Mohri et al.'19]: mitigates the training procedure bias but cannot guarantee a good model fairness.
 - AgnosticFair [Du et al.'20]: trains fair models at the cost of data privacy infringement.

Agnostic Loss function

$$\min_{\theta} \left\{ L(\theta) := \max_{\lambda} \left\{ \sum_{i=1}^N \lambda_i L_i(\theta) \mid \lambda_1, \dots, \lambda_N \geq 0, \sum_{i=1}^N \lambda_i = 1 \right\} \right\}$$

worst-case loss formed by any mixture of clients' empirical distributions

Loss defined over a distribution

Can We Extend Existing Approaches to Tackle This Problem?

- Fair horizontal federated learning
 - Agnostic federated learning [Mohri et al.'19]: mitigates the training procedure bias but cannot guarantee a good model fairness.
 - AgnosticFair [Du et al.'20]: trains fair models at the cost of data privacy infringement.

Agnostic Fair Learning

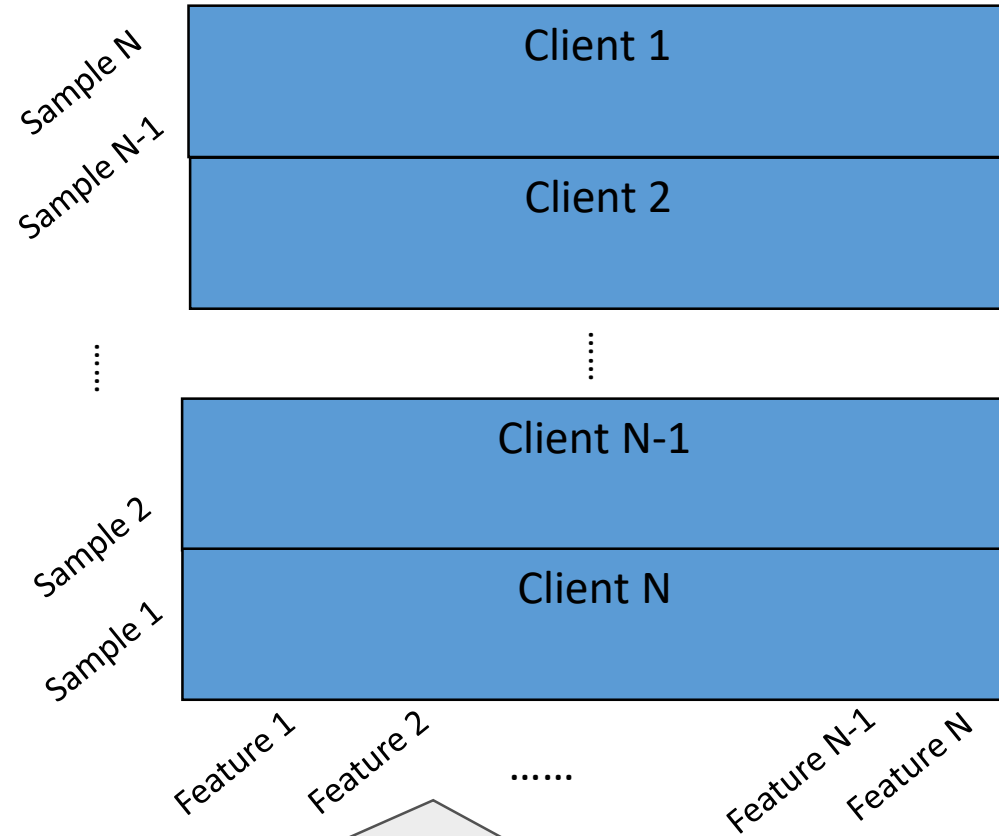
$$\min_{\theta} \left\{ L(\theta) := \max_{\lambda} \left\{ \sum_{i=1}^N \lambda_i L_i(\theta) \mid \lambda_1, \dots, \lambda_N \geq \mathbf{0}, \sum_{i=1}^N \lambda_i = \mathbf{1} \right\} \right\}$$

subject to **Agnostic Fairness Constraint**

Most of federated learning methods do not consider training fair models.
Some works tried but assumed a horizontally partitioned dataset.

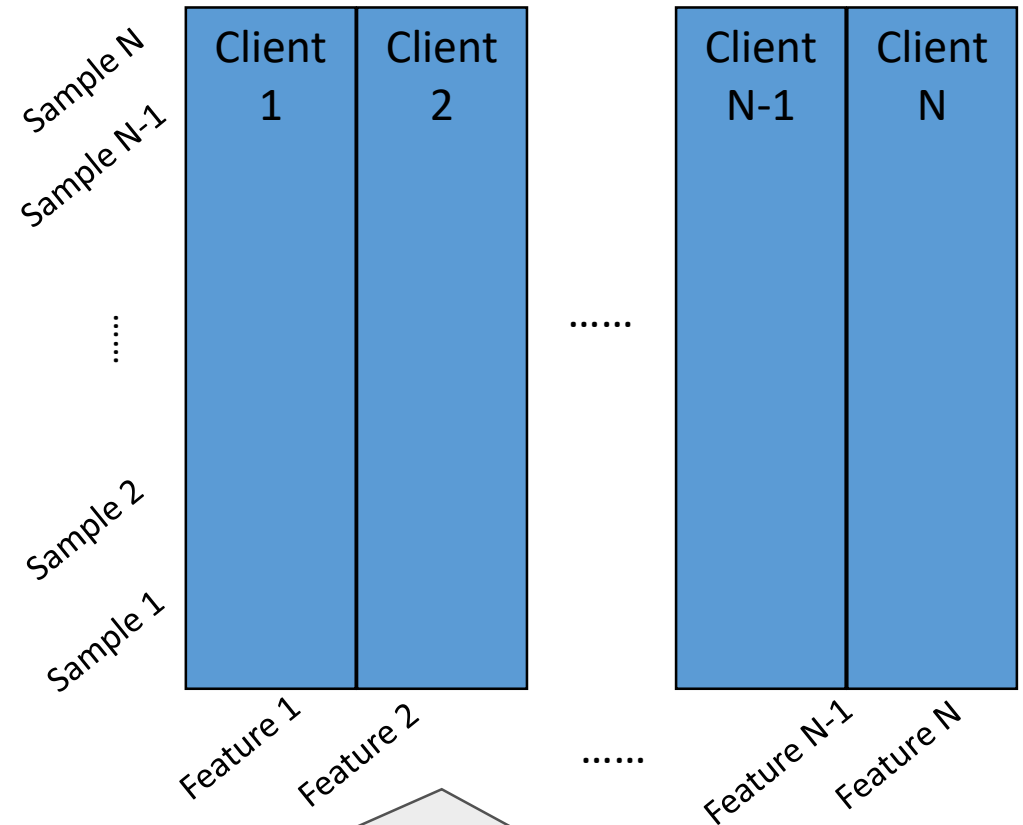
HFL is Fundamentally Different from VFL

Horizontal Federated Learning



Samples are distributed over clients

Vertical Federated Learning



Features are distributed over clients

Fair HFL cannot be extended to Fair VFL due to the difference in problem settings

Challenges of Vertical Federated Fair Model Training

Fair Model Training

- Pre-processing methods
- Post-processing methods
- In-processing methods

Related Federated Learning Methods

- Classical vertical federated learning
- Fair horizontal federated learning

No straight-forward extension

Vertically Federated Fair Model Training

Why is it so difficult to train a fair model in a vertical federated learning framework?

Challenge I: Privacy Preservation Requirement

- Vertical federated fair model training is challenging due to the intrinsic conflict between fair model training and federated learning.
 - Fair model training -> **I want all the data.**
 - Accurately evaluating the fairness of a model requires access to the data of all parties.
 - Vertical Federated learning -> **Oh sorry, I cannot help with that.**
 - Collecting the overall dataset is prohibitive in vertical federated learning.

An intrinsic conflict!

Challenge II: Local Fairness Estimation Does NOT Work

- Can we estimate the model fairness locally on each participating party?
 - The answer is NO, due to the following reasons.
 - For (passive) clients having no knowledge of the protected attribute, *measuring model fairness is impossible*.
 - For (active) clients having knowledge of the protected attribute, measuring model fairness leads to *both inferior fairness and accuracy performance* due to a lack of features.

Challenge III: Distributed and Asynchronous Implementation

- In-processing methods customize machine learning algorithms to directly train fair models

$$\min_{\theta} \left\{ L(\theta) := \sum_{i=1}^n \frac{1}{n} \sum_{i=1}^n \{ L_i(\theta) := l(X_i^T \theta, y_i) \} \right\}$$

$$\text{subject to } \mathbb{P}\{X_i^T \theta > 0 | y = 1, s = a\} = \mathbb{P}\{X_i^T \theta > 0 | y = 1, s = b\}$$

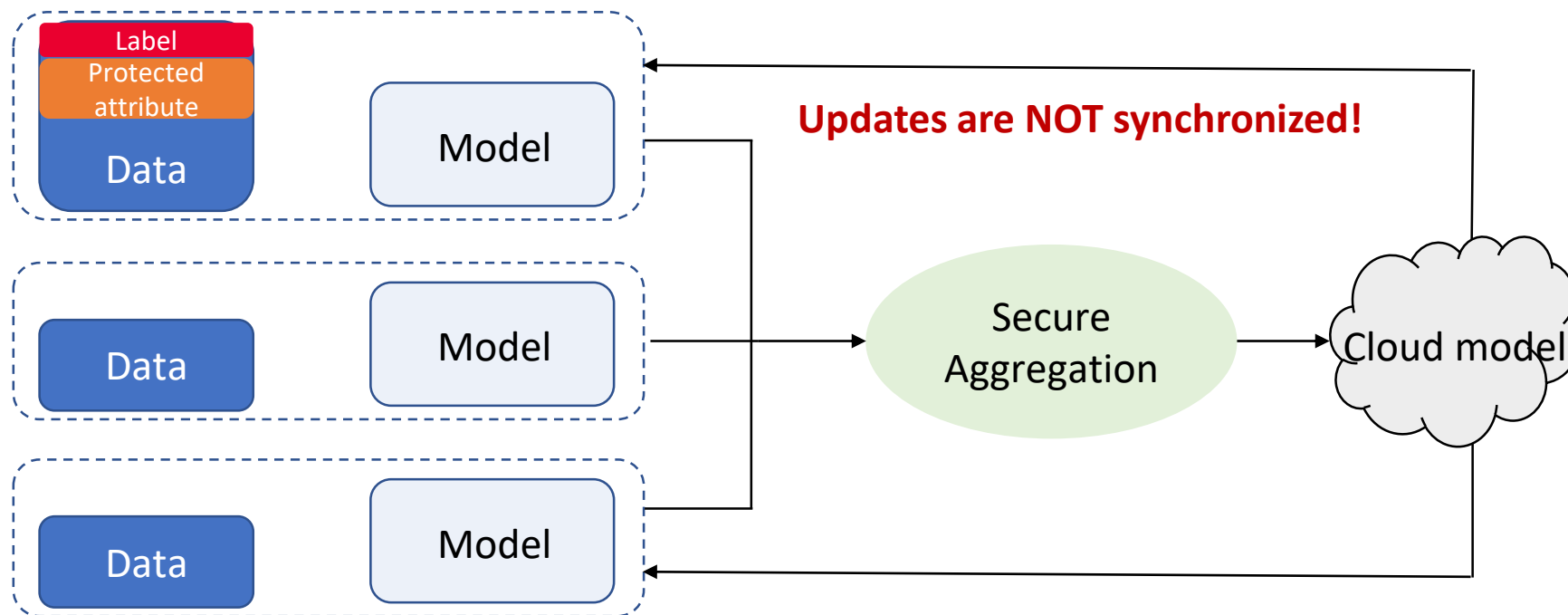
Each sample $X_i = \{(X_i)_k\}_{k=1}^N$
Each client k only has $(X_i)_k$

Only active clients have the labels and sensitive attributes

- The formulated problems are typically nonconvex and constrained.
- How to solve the problem in a distributed way?

Challenge III: Distributed and Asynchronous Implementation

- Asynchronous implementation



Conclusion

- It is important to train fair models in the context of vertical federated learning.
- Vertical federated fair model training has many challenges
 - The intrinsic conflict between accurately measuring model fairness and preserving data privacy.
 - Local fairness estimation does not work for passive parties and does not work well for active parties.
 - Imbalanced computational resources poses additional challenges for solving the fair learning task.

Part VI

Conclusions and Future Directions

Tutor: Jian Pei

Conclusions

- Different notions of fairness in federated learning
 - Performance fairness / collaboration fairness / model fairness
- Fair federated learning becomes increasingly important
 - Sustain long-term stability of the FL system
 - Attract more participants with high quality and high volume of data
 - Machine learning models are deployed in more and more applications in which model fairness is critical
- Collaboration fairness in FL is a multi-disciplinary research direction
 - Data valuation / cooperative game theory / marketing / ...
- Research on model fairness in FL is at an infant stage
 - Formulations / data privacy / communication overhead / fast training methods / ...

Future Directions

- Collaboration fairness in FL
 - Design lightweight incentive schemes to ensure collaboration fairness
 - Integrate more cutting-edge technologies, such as reinforcement learning and block-chain, with fair incentive schemes
 - How to measure participants' contribution in vertical federated learning
 - Multi-dimensional contribution measurement and reward system
- Model fairness in FL
 - Unified principle of fairness instead of various fairness definitions
 - Personalized horizontal/vertical fairness for different data distributions
 - Verifiable and interpretable fairness in federated learning