[Paper] [Code]

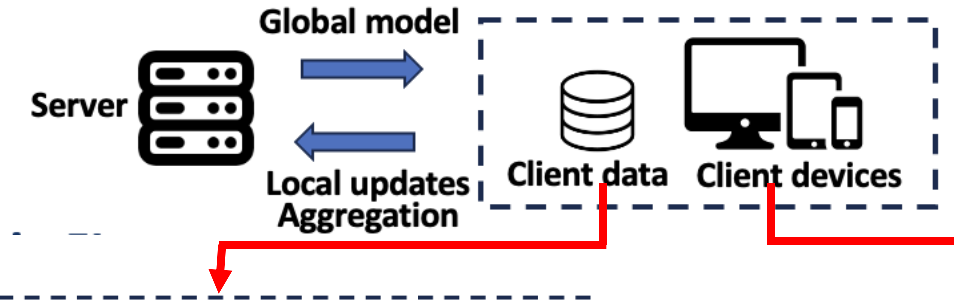
Tackling Intertwined Data and Device Heterogeneities in Federated Learning with Unlimited Staleness

Haoming Wang, Wei Gao

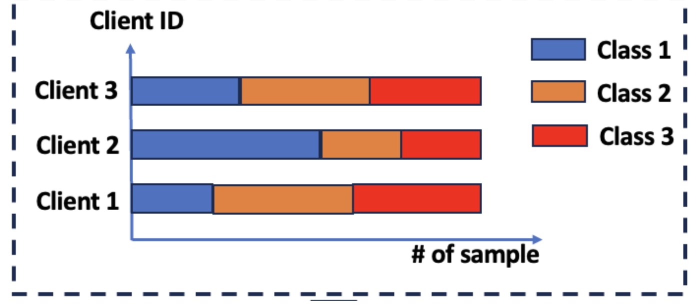
University of Pittsburgh

Overview

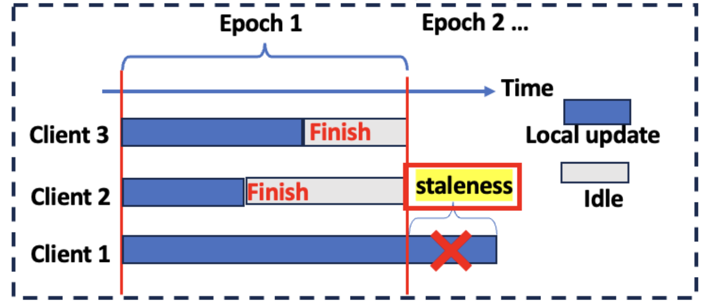
❖ **Motivation**: **Intertwined heterogeneities in FL**



1. Clients compute update on local data
2. The server aggregates client updates

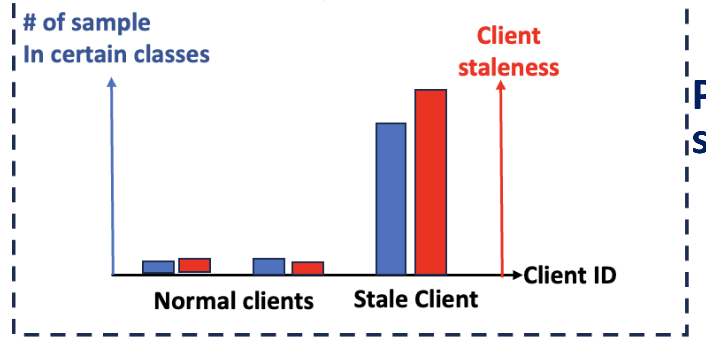


Data heterogeneity



Device heterogeneity

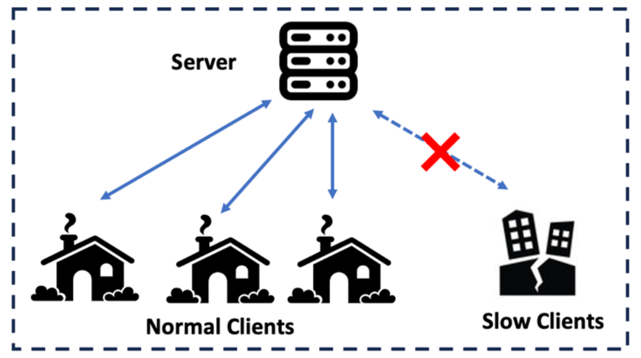
Intertwined



data in certain classes may only be available at some slow clients

Practical scenario

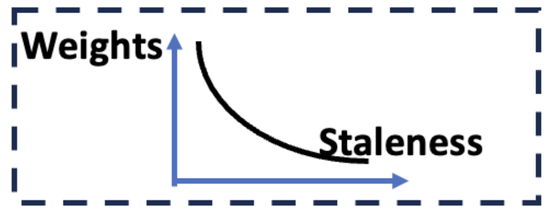
Hazard rescue



Devices at hazard sites: 1.Crucial data 2.Limited connectivity

❖ **Limitation of the existing work**:

❑ Asynchronous FL with weighted aggregation [1]



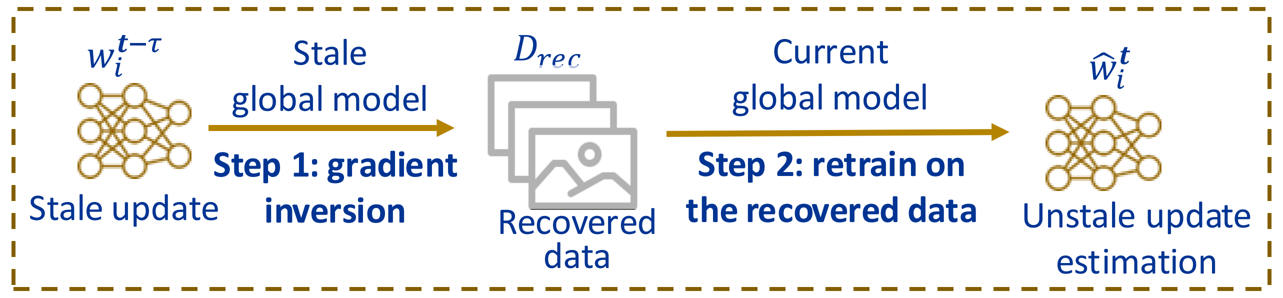
Model biased away from data on stale client

❑ First-order compensation based on Taylor expansion [2]

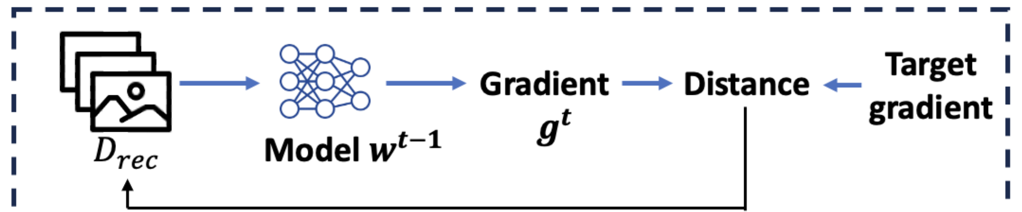
$$\underbrace{g(w_{t+\tau})}_{\text{estimated update}} \approx \underbrace{g(w_t)}_{\text{stale update}} + \nabla g(w_t)\underbrace{(w_{t+\tau}-w_t)}_{\text{Correction term}}, \ (\tau \text{ is staleness})$$

Assume the **staleness is small** enough to neglect higher order terms

❖ **Gradient Inversion Based Compensation**:



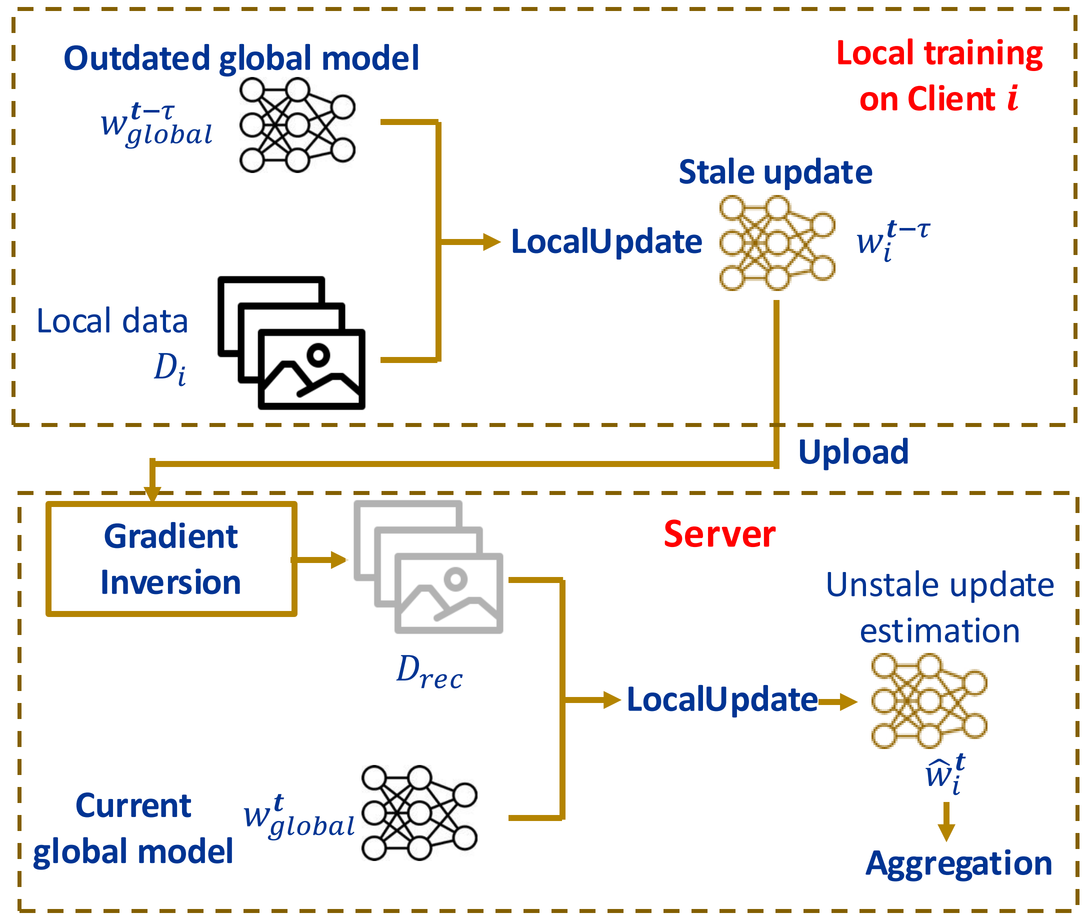
Gradient Inversion [3]



$$D^*_{rec} = arg\ min_{D_{rec}} \left\| \frac{\partial L[D_{rec}; w^{t-1}]}{\partial w^{t-1}} - g^t \right\|_2^2$$

- Minimize the difference between
  ① the trained model's gradient
  ② the gradient computed from the recovered data
- When D_{rec} has many samples:
  - The quality of D_{rec} is low
  - But it still contains useful information about client data

Main Idea

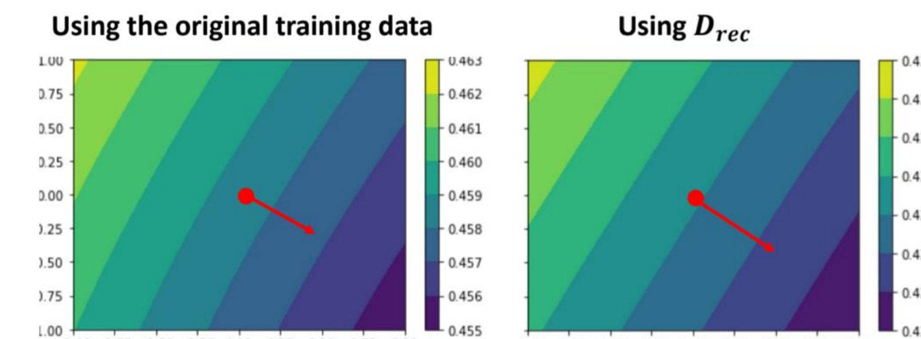


In this intertwined heterogeneities scenario:
- Client i contain unique knowledge
- Can only upload stale update $w_i^{t-\tau}$

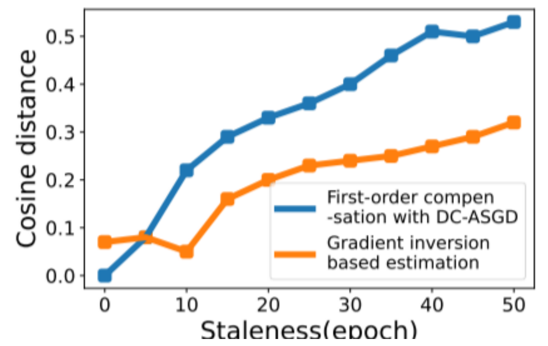
Step 1: the server uses gradient inversion to recover an intermediate dataset D_{rec} from $w_i^{t-\tau}$. Instead of fully recovering the client i's training data D_i , we only expect D_{rec} to represent the similar data distribution with D_i.

Step 2: The server computes an estimation of w_i^t from $w_i^{t-\tau}$ by using D_{rec} to train its current global model w_i^t , and aggregates $\widehat{w}_i^t$ with model updates from other clients to update its global model in the current epoch.

❖ **Similar loss surface compared with the original data:**



● The current global model in the loss space
→ The direction of the computed gradient

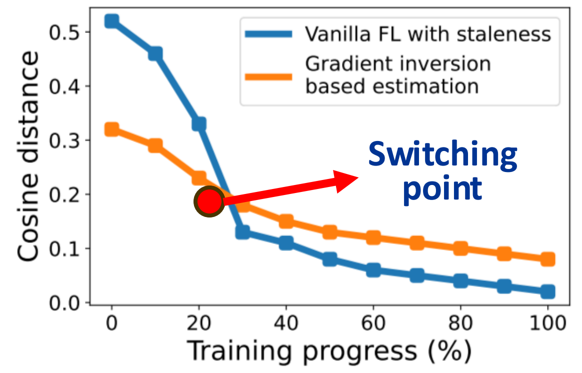


Compensation error compared with First-order compensation

Method Details

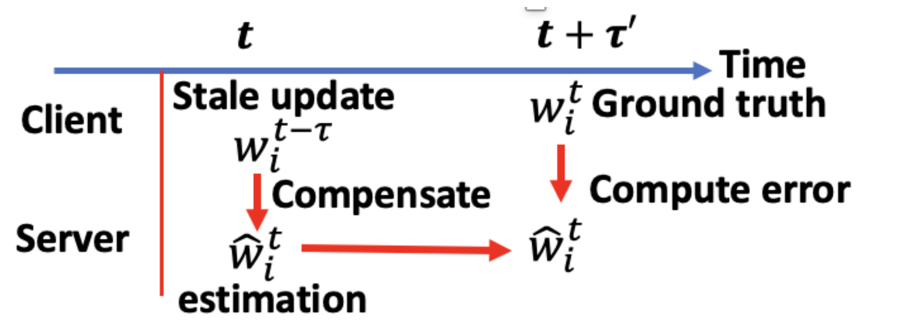
❖ **Switching back to Vanilla FL in later stages of FL Training**

Vanilla FL has less error as model converges:



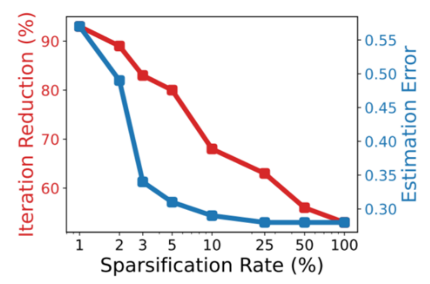
Deciding the switching point:

Computing the current error at later epoch

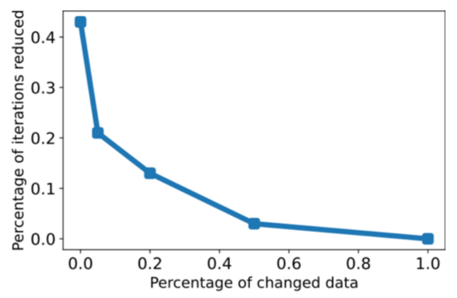


❖ **Reducing the Computing Cost of Gradient Inversion**

Sparsification: reduce the the objective function complexity



Iterative initialization: Initialize D_{rec} with previous recover results

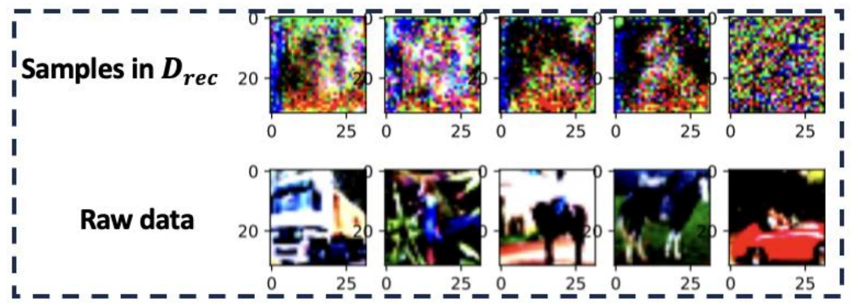


Computing cost reduction with different percentage of changed client data in each epoch

Method Details

❖ **Protecting the clients' data privacy**

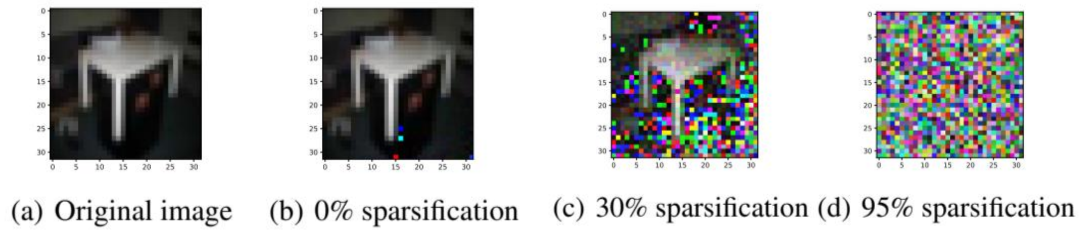
Most FL scenarios:
each client has a large batch of samples



Nearly impossible to pixel-wisely recover

Extreme scenarios:
- each client only has one sample

Sparsification and **gradient noise** to mitigate the attack power



(a) Original image (b) 0% sparsification (c) 30% sparsification (d) 95% sparsification

Protecting input images

Defense	None	95% sparsification	95% sparsification + noise
Label recovery ACC	85.5%	66.7%	46.4%

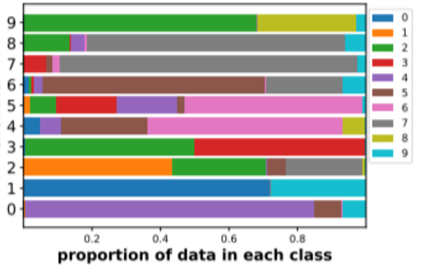
Protecting labels

Performance Evaluation

❖ **Experiment setup and baselines**

❑ **Data heterogeneity**

Dirichlet distribution:



($\alpha = 0.1$)

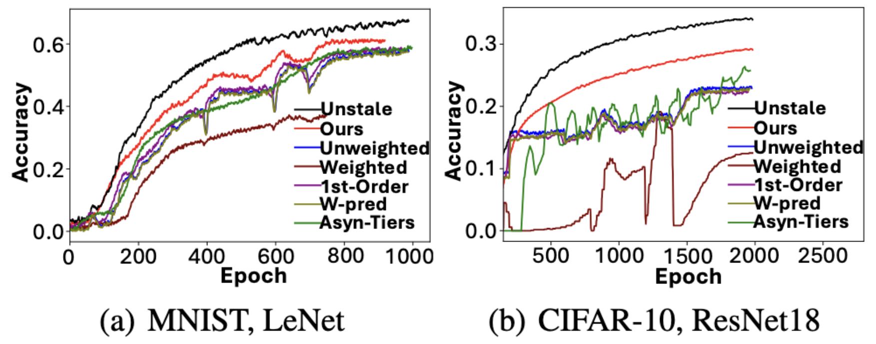
❑ **Device heterogeneity**
- select one data class to be affected by staleness
- apply staleness to 10 clients with the most data samples in the class

❑ **Baselines:**
- **Vanilla FL** (① Unweighted)
- **Asynchronous FL** with:
  ② Weighted aggregation
  ③ Asynchronous Tiers
- **Unstale update estimation**:
  ④ First-order method
  ⑤ Future model prediction

❖ **Scenario 1: Fixed client data**

Accuracy(%)	MNIST	FMNIST	CIFAR10	MDI
Unweighted	57.4	49.2	22.8	72.3
Weighted	39.2	30.1	12.6	61.2
1st-Order	57.4	49.3	22.6	72.3
W-Pred	57.3	48.9	22.9	72.2
Asyn-Tiers	57.6	50.3	25.9	69.8
Ours	**61.2**	**55.4**	**29.4**	**75.4**

Accuracy on data affected by staleness



(a) MNIST, LeNet (b) CIFAR-10, ResNet18

Accuracy curve during training

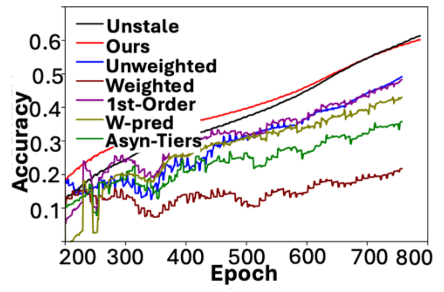
❖ **Scenario 2: Variant client data during FL training**

Variant data setting:
- Client data is initialized with MNIST data
- During training MNIST samples are gradually replaced by SVHN samples

MNIST

SVHN

Experiment results:



References

[1] Federated learning with buffered asynchronous aggregation.
[2] Asynchronous stochastic gradient descent with delay compensation.
[3] Deep leakage from gradients.