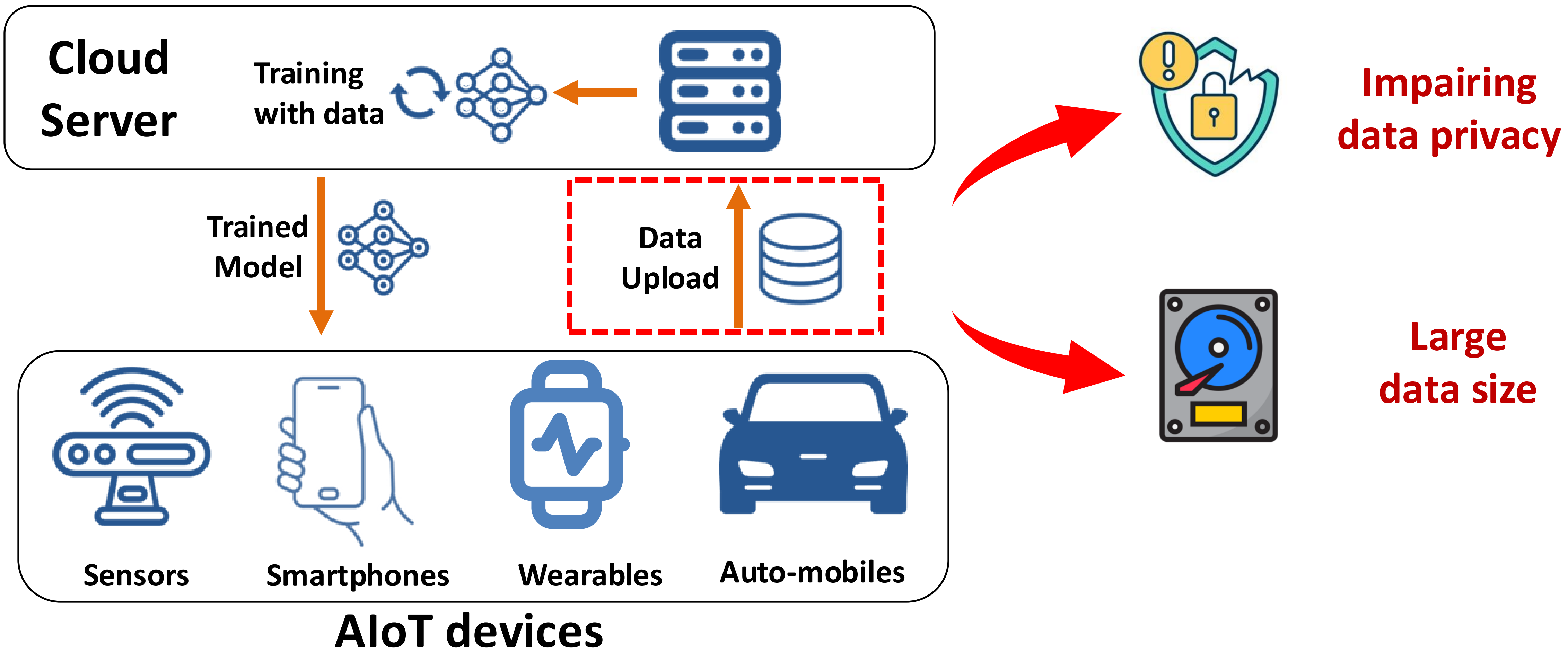
# When Device Delays Meet Data Heterogeneity in Federated AIoT Applications

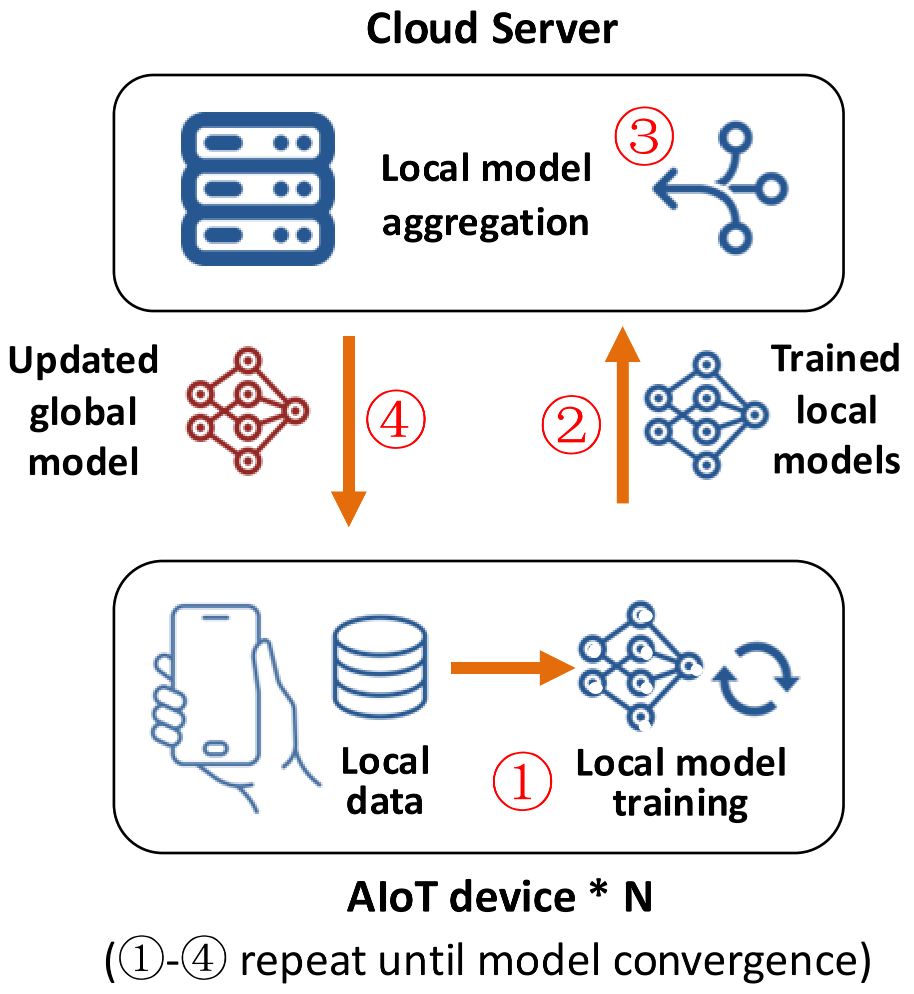
**Haoming Wang**, Wei Gao
University of Pittsburgh

***ACM MobiCom 2025***

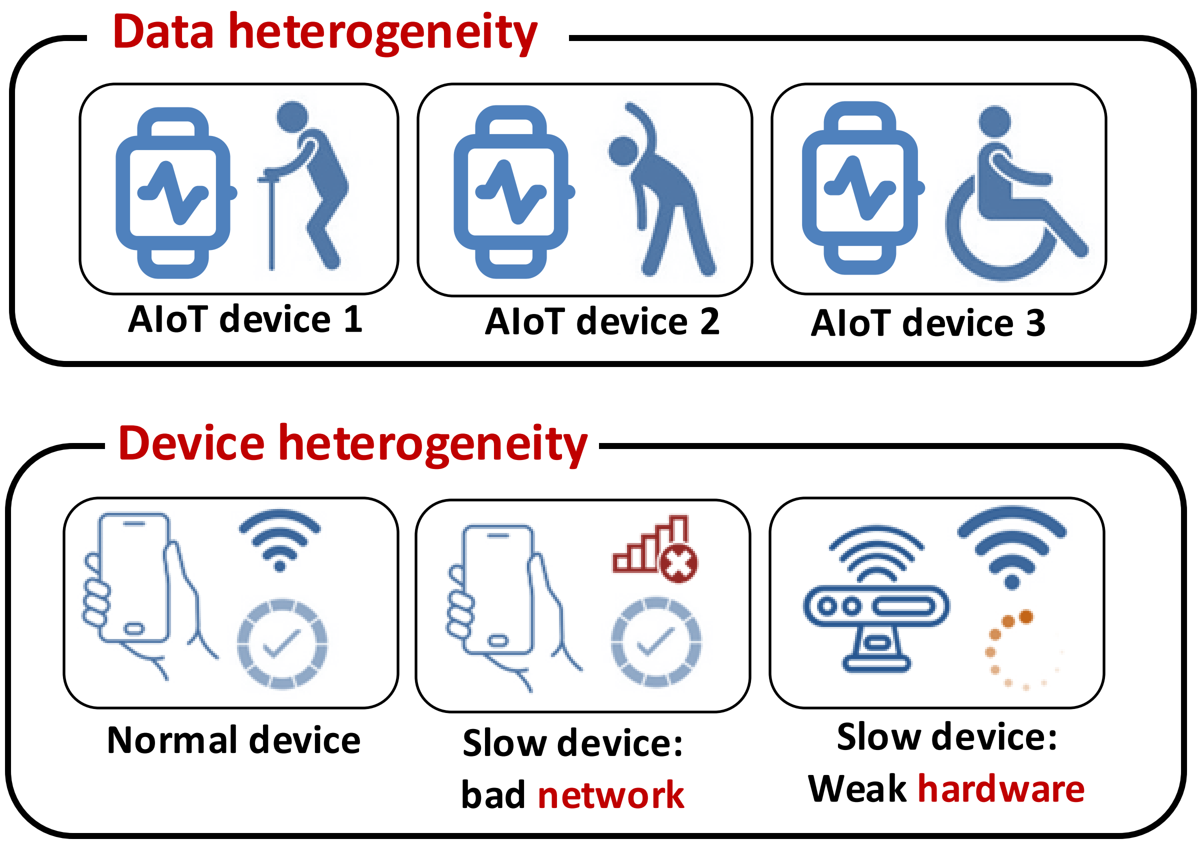
# Artificial Intelligence of things (AIoT)



# Federated AIoT



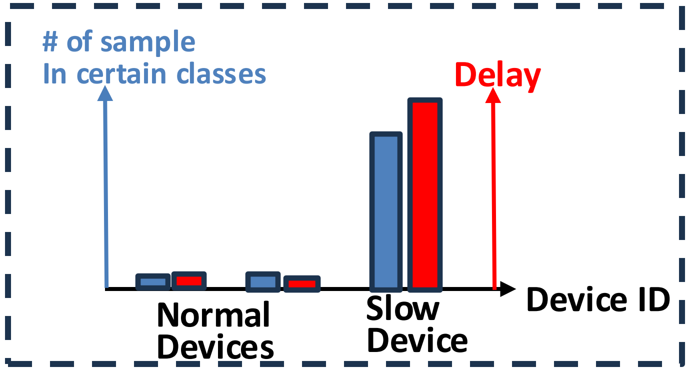

## System challenges:



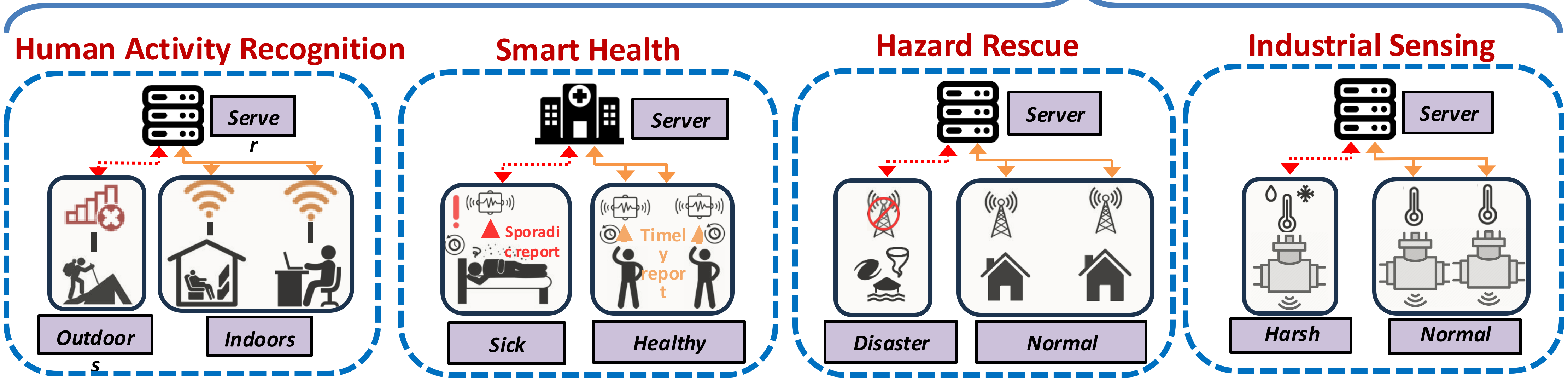
# The Real Problem: When These Challenges are Intertwined

## Intertwined Heterogeneties:

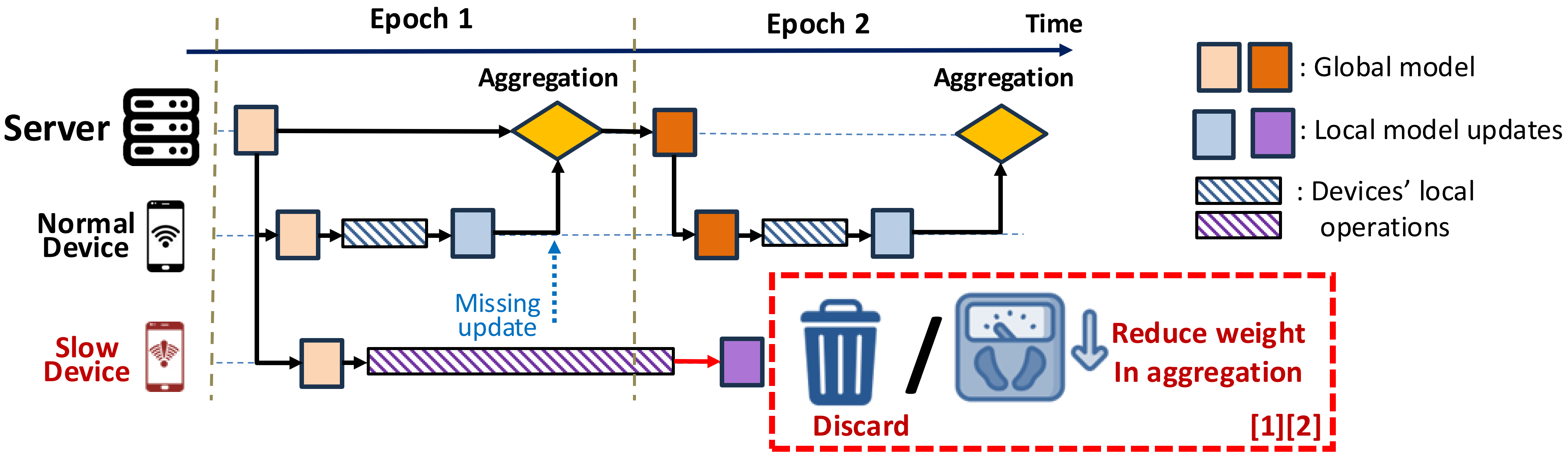
data of a certain class or with specific features may only be available on some slow IoT devices.



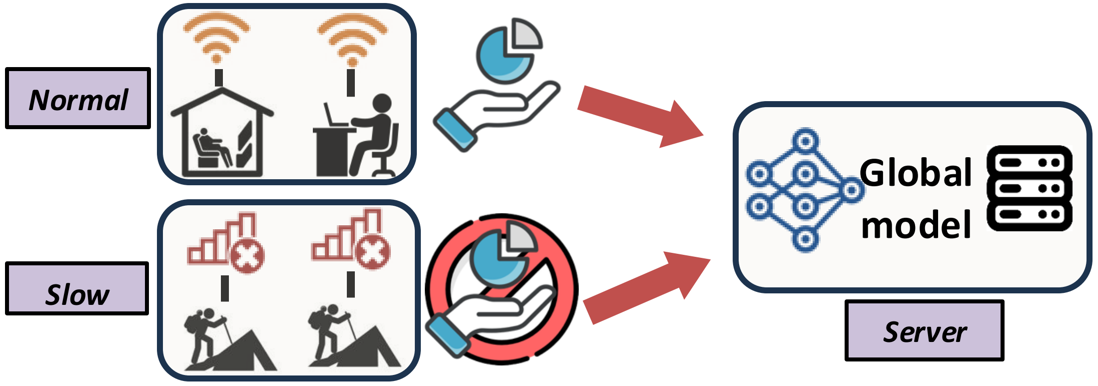

Real-world examples



# Existing Solutions tackling device delays



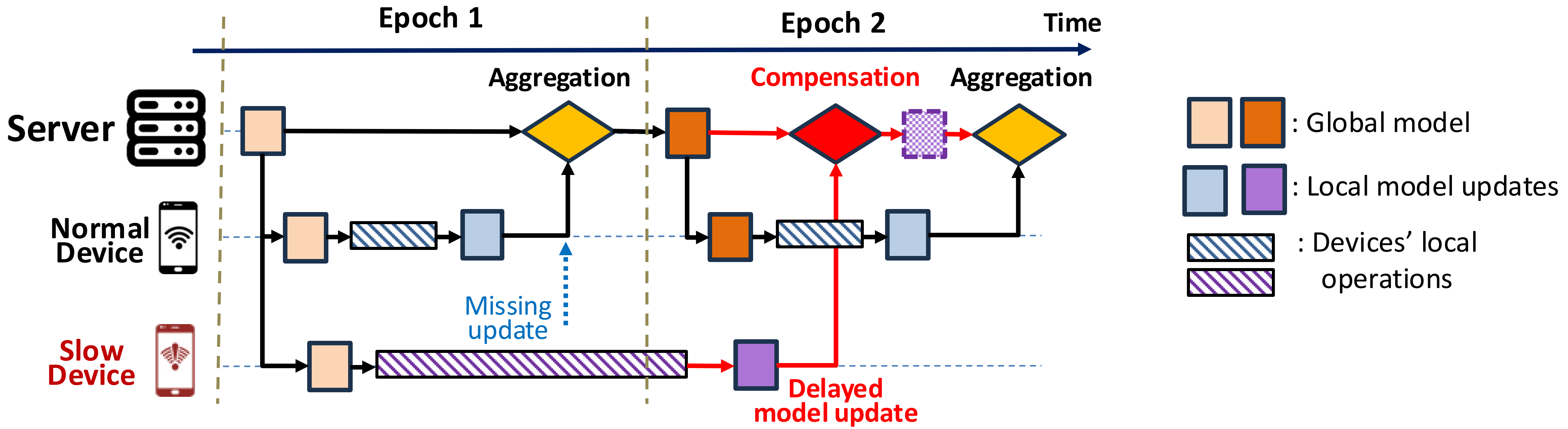

❖ **Limitation under intertwined heterogeneities:**




**Unique knowledge on slow devices is excluded**

[1] Asynchronous online federated learning for edge devices with non-iid data.
[2] AsyncFedED: Asynchronous Federated Learning with Euclidean Distance based Adaptive Weight Aggregation.

# A better choice: compensate the delay




❖ **Existing compensation method:**

**First order compensation [3]**

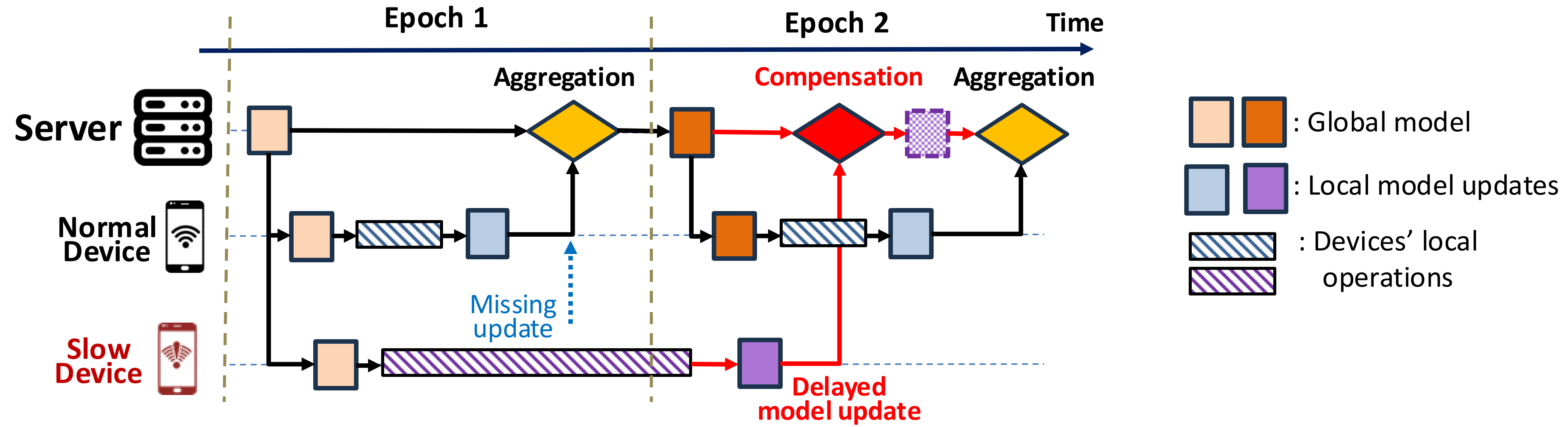
$$g(w_{t+\tau}) \approx g(w_t) + \nabla g(w_t)(w_{t+\tau} - w_t)\text{ , (}\tau\text{ is delay)}$$

estimated update | delayed update | Correction term

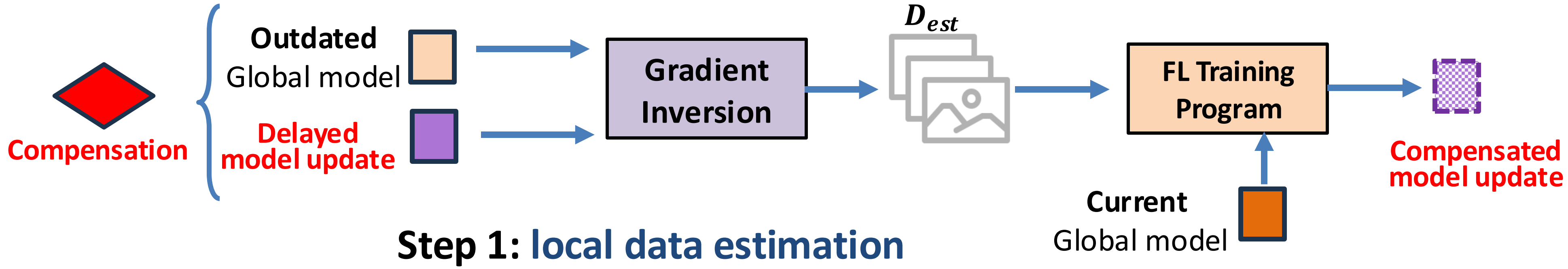
**Compensation error will significantly increase with delay**

[3] Asynchronous stochastic gradient descent with delay compensation

# Our method: Delay Compensator in FL (FedDC)




## ❖ Compensation in FedDC:

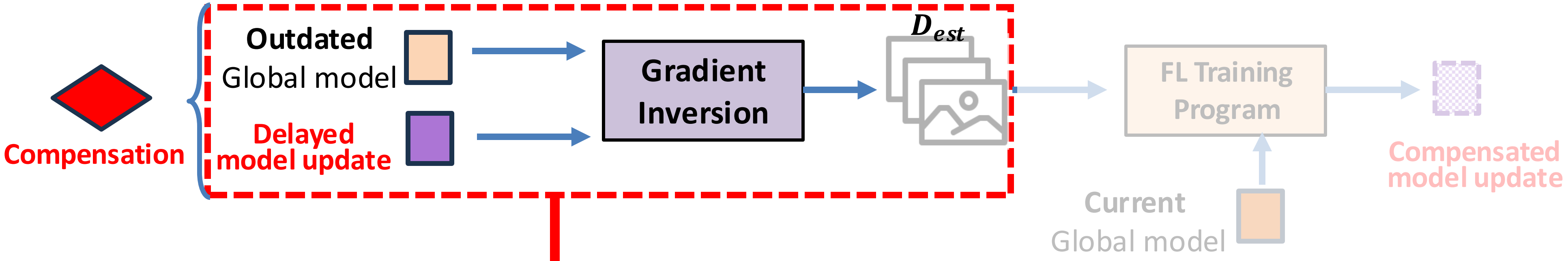



**Step 1:** **local data estimation**

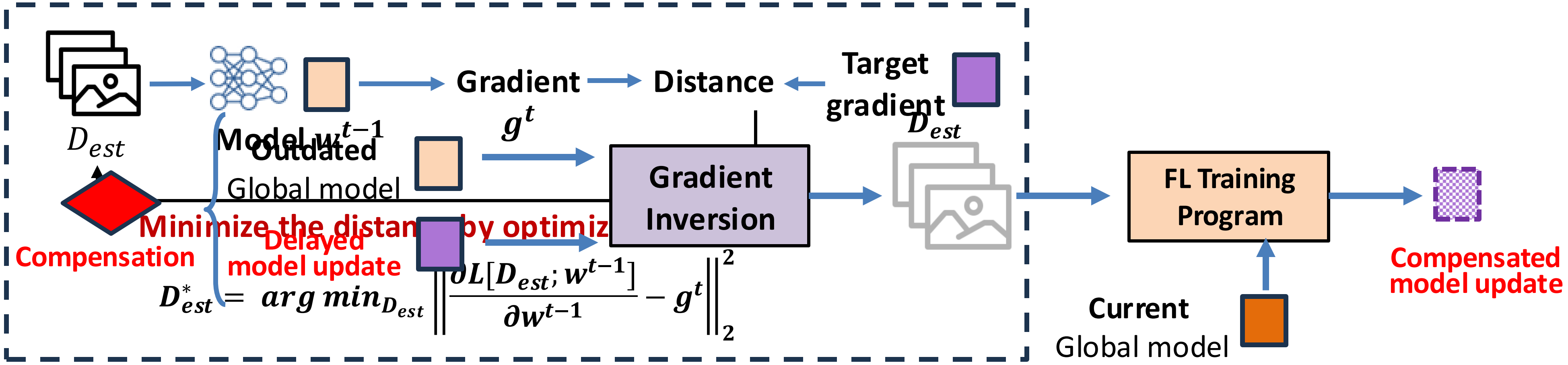
**Step 2:** **undelayed update estimation**

# Our method: Delay Compensator in FL (FedDC)

## Compensation in FedDC:

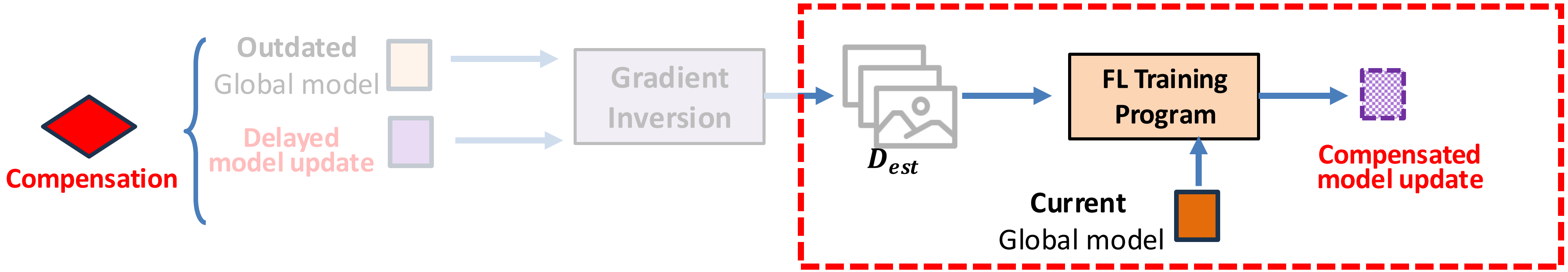



### Step 1: local data estimation:



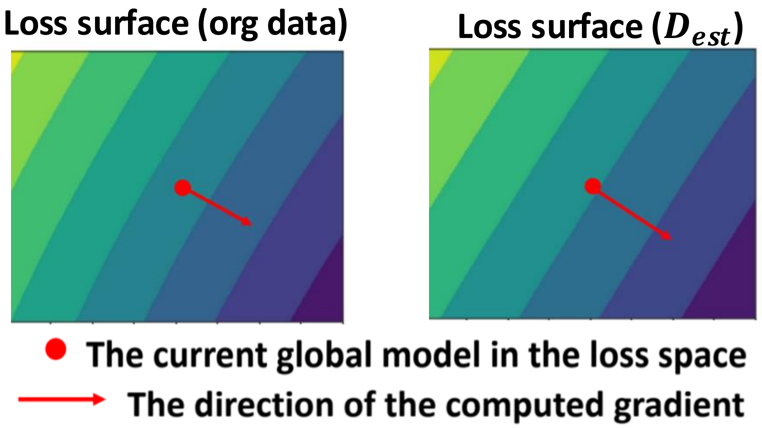
# Our method: Delay Compensator in FL (FedDC)

## Compensation in FedDC:

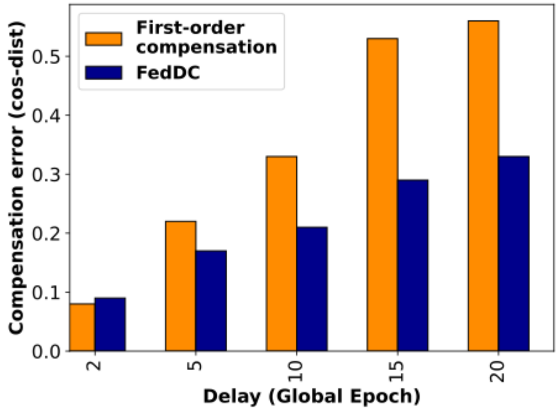



**Step 2:** undelayed update estimation

### Rationale of using $D_{est}$ for estimation:




### Compensation error comparison:

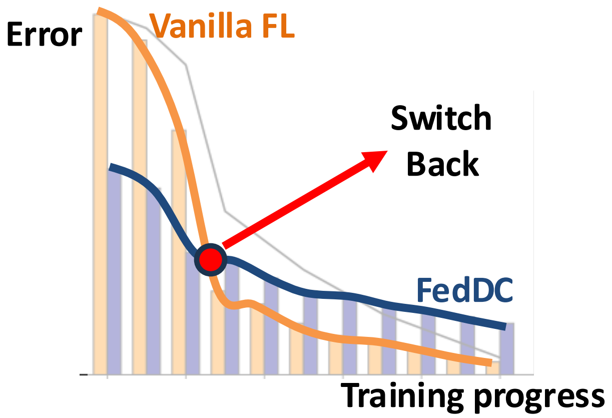


# FedDC: Method Details

## 1: Adaptively Switching back to Vanilla FL

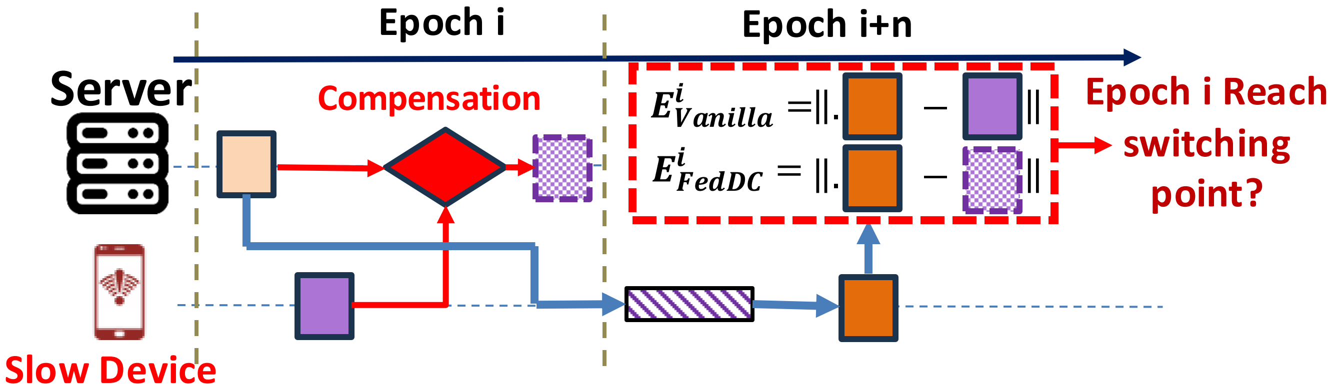
- Vanilla FL has less error as model converges:

  Switch back to Vanilla FL in later stage




- Deciding the switching point:

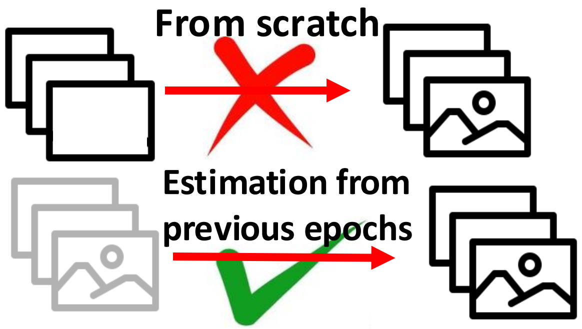
  Computing the current error at later epoch



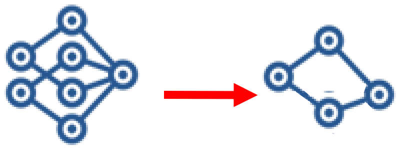
# FedDC: Method Details

## 2: Reducing the Computing Cost of gradient inversion

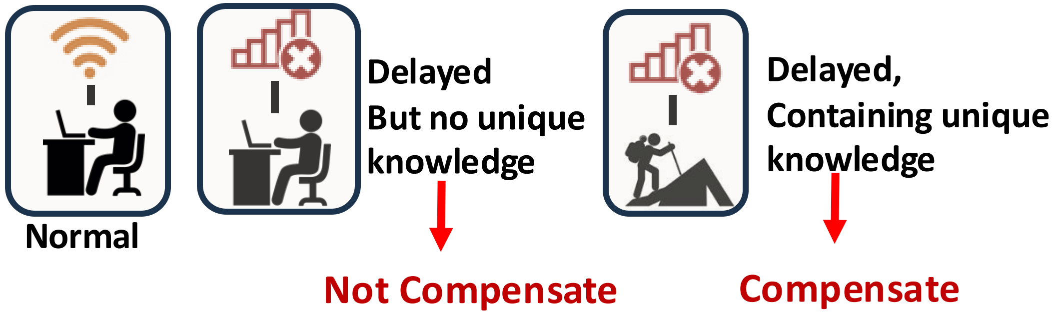
- **Better *Dest* initialization**




- **Gradient sparsification**

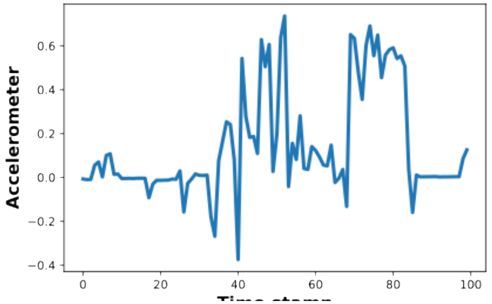


- **Selective computation**

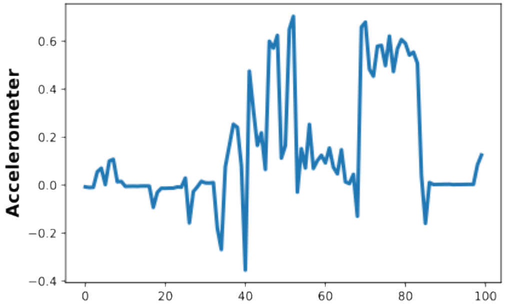


# Privacy Protection

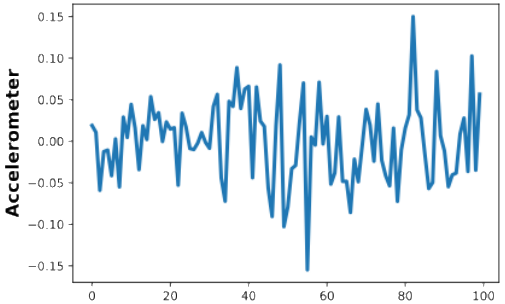
❖ **Under Typical FL setting (batched data)**




Raw data




Recovered (Batch size =1)

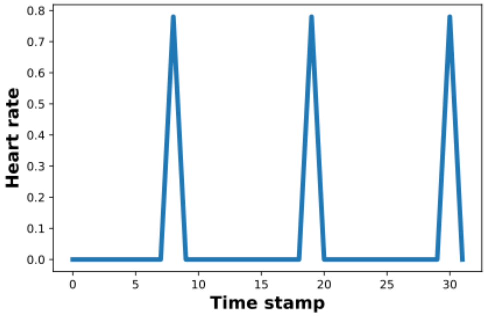



Recovered (Batch size =100)

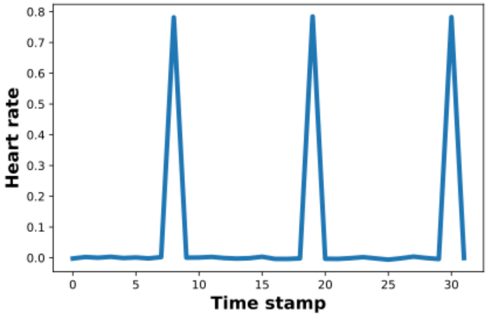
Nearly impossible to accurately recover

❖ **Extreme Case: only one sample on a device**

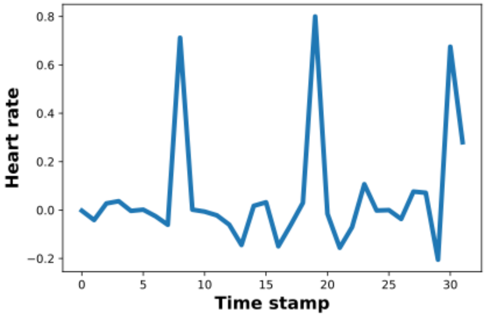
**- Protect privacy via gradient sparsification**



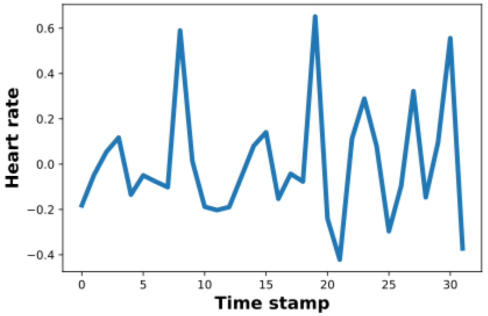

Raw data




0% sparsification



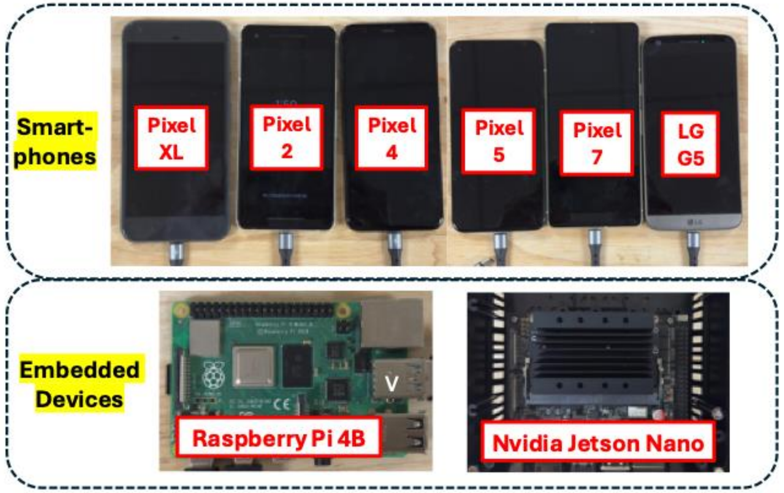

40% sparsification




95% sparsification

# Experiments

## Implementation:




## Datasets & Models:

- **PAMAP2 (HAR):**
  - IMU + Heart rate sensor
  - 3-layer MLP
- **ExtraSensory (Fine-grained HAR):**
  - IMU, gyroscope and magnetometer
  - 1D-CNN
- **MDI (Disaster Images):** ResNet18

## Baselines:

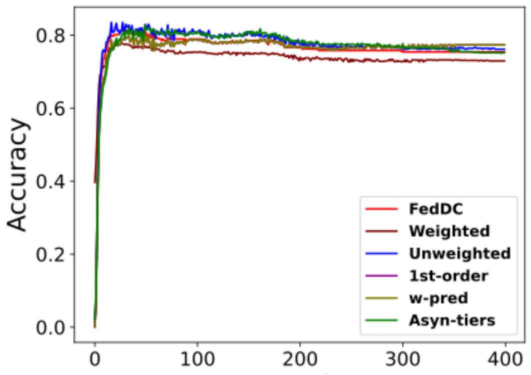
- **Unweighted**: Standard FedAvg (control)
- **Weighted [2]**: Common async method (down-weights late updates)
- **Asyn-Tiers [3]**: Groups devices by speed
- **1st-Order [4]**: Taylor-expansion compensation
- **W-Pred [5]**: Future global model prediction

[4] Taming momentum in a distributed asynchronous environment.
[5] FedAT: A high-performance and communication-efficient federated learning system with asynchronous tiers

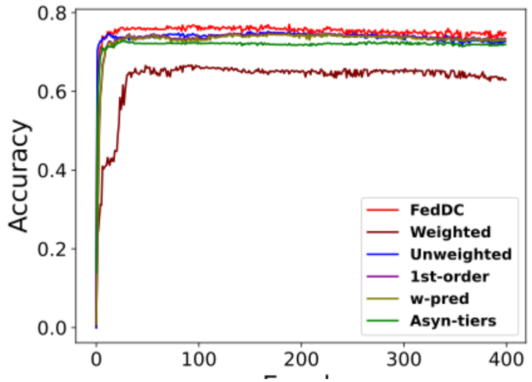
# Experiments

**Main results:**

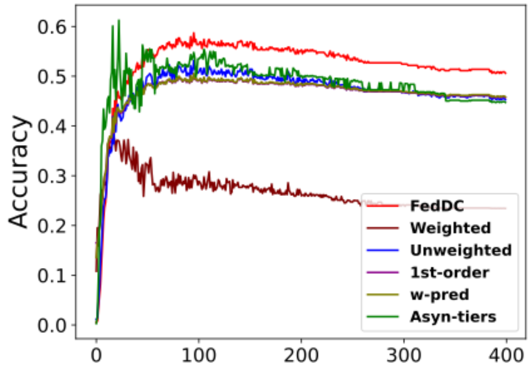
❖ **Performance under different amount of delay:**




Low delay

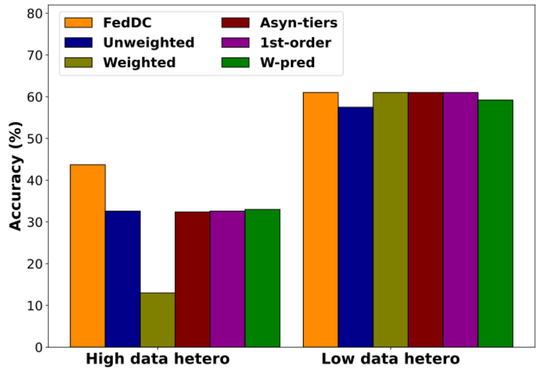



Medium delay

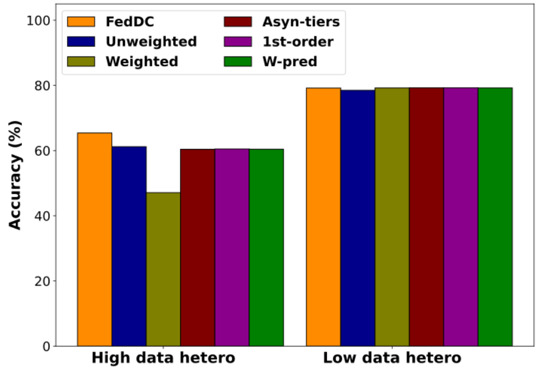



High delay

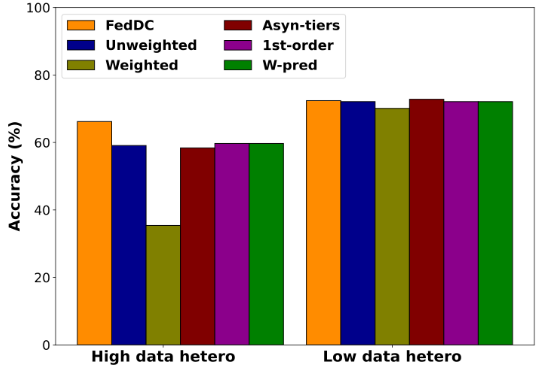
❖ **Performance under different data heterogeneity: (with high delay)**




PAMAP2



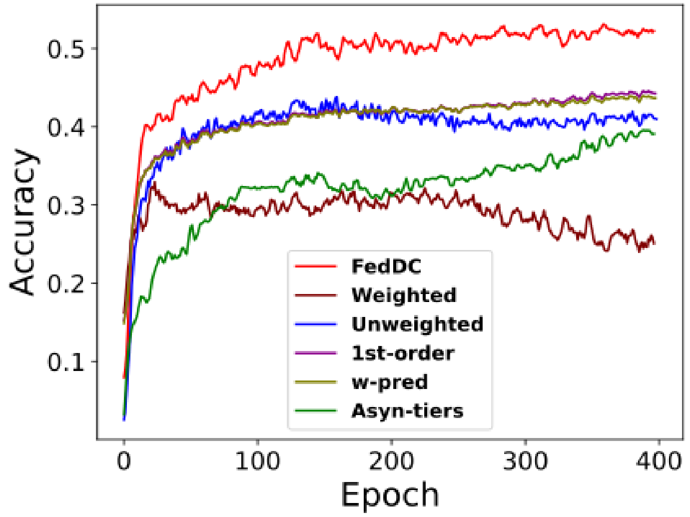

ExtraSensory




MDI

# Experiments

## ❖ Performance under variant global data distribution:




| | PAMAP2 | EtraSensory | MDI |
|---|---|---|---|
| Unweighted | 38.9% | 20.3% | 65.1% |
| Weighted | 23.8% | 0% | 59.0% |
| Asyn-tiers | 37.5% | 16.7% | 66.8% |
| 1st-Order | 42.3% | 22.4% | 65.1% |
| W-Pred | 42.1% | 22.4% | 65.1% |
| **FedDC** | 53.5% | 34.7% | 69.3% |

## ❖ Computing cost at server end

| FL model | MLP | 1D-CNN | 2D-CNN |
|---|---|---|---|
| Full Computation | 21.63 | 15.12 | 19.66 |
| Selective Computation | 2.54 | 2.23 | 3.71 |
| Selective Computation + 95% SP | 0.91 | 0.82 | 0.88 |

# Summary

**FedDC enables robust, accurate, and efficient Federated Learning for realistic AIoT applications.**

- **High-Accuracy**: Improves model accuracy by up to 34% in high-delay, heterogeneous scenarios.
- **Lightweight**: Adds no overhead to IoT devices and no end-to-end delay to training.
- **Privacy-Preserving**: Sparsification protects local data from being recovered by the server.

# Thank you!

**Lab Website**

pittisl.github.io

**Paper Link**

sites.pitt.edu/~weigao/publications/mobicom25_aiot.pdf