FedBS: Learning on Non-IID Data in Federated Learning using Batch Normalization

Meryem Janati Idrissi

MOHAMMED VI POLYTECHNIC UNIVERSITY

Ismail Berrada

MOHAMMED VI POLYTECHNIC UNIVERSITY

Noubir Guevera

NORTHEASTERN UNIVERSITY

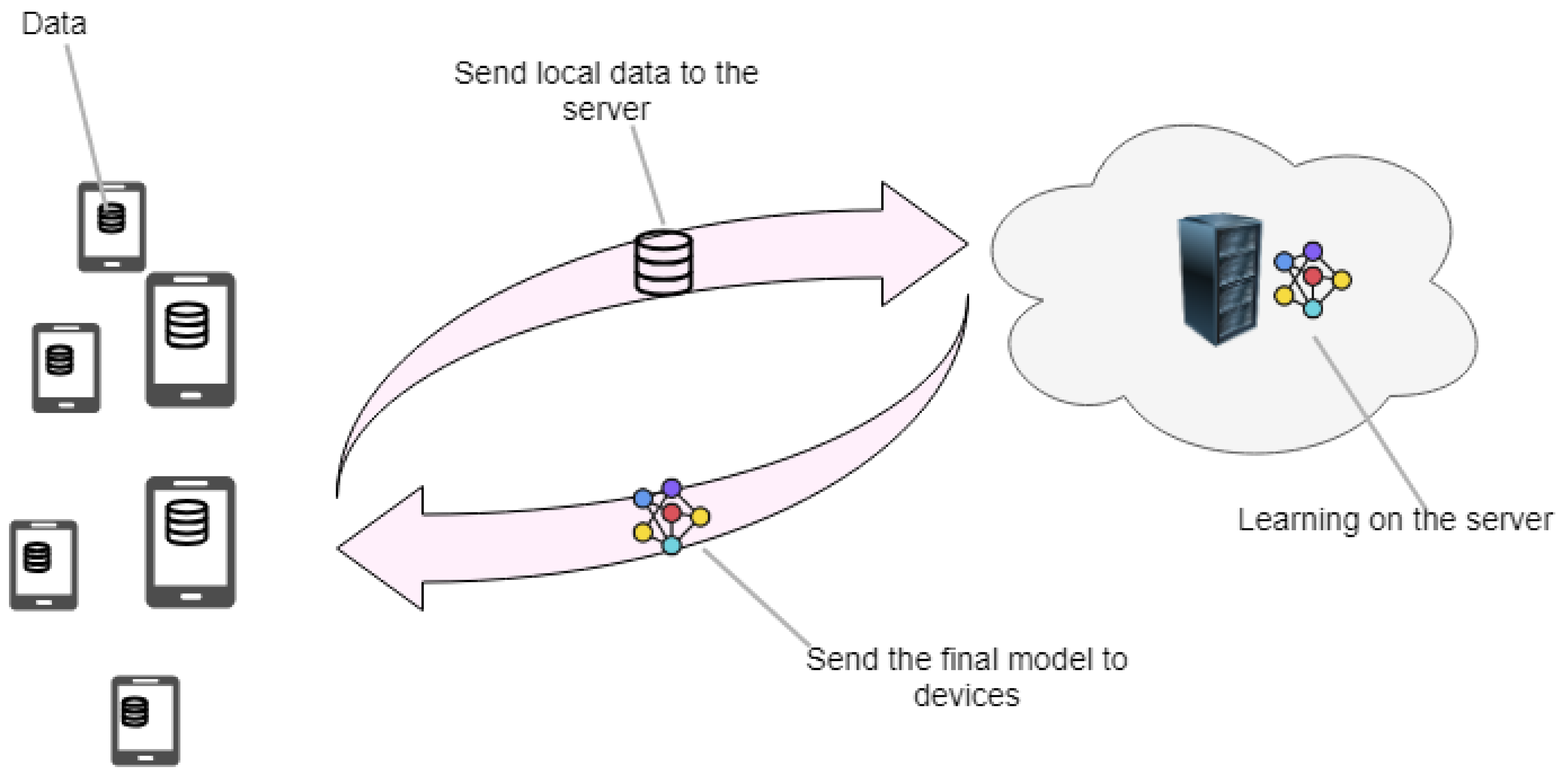
ICTAI 2021

Table of Contents

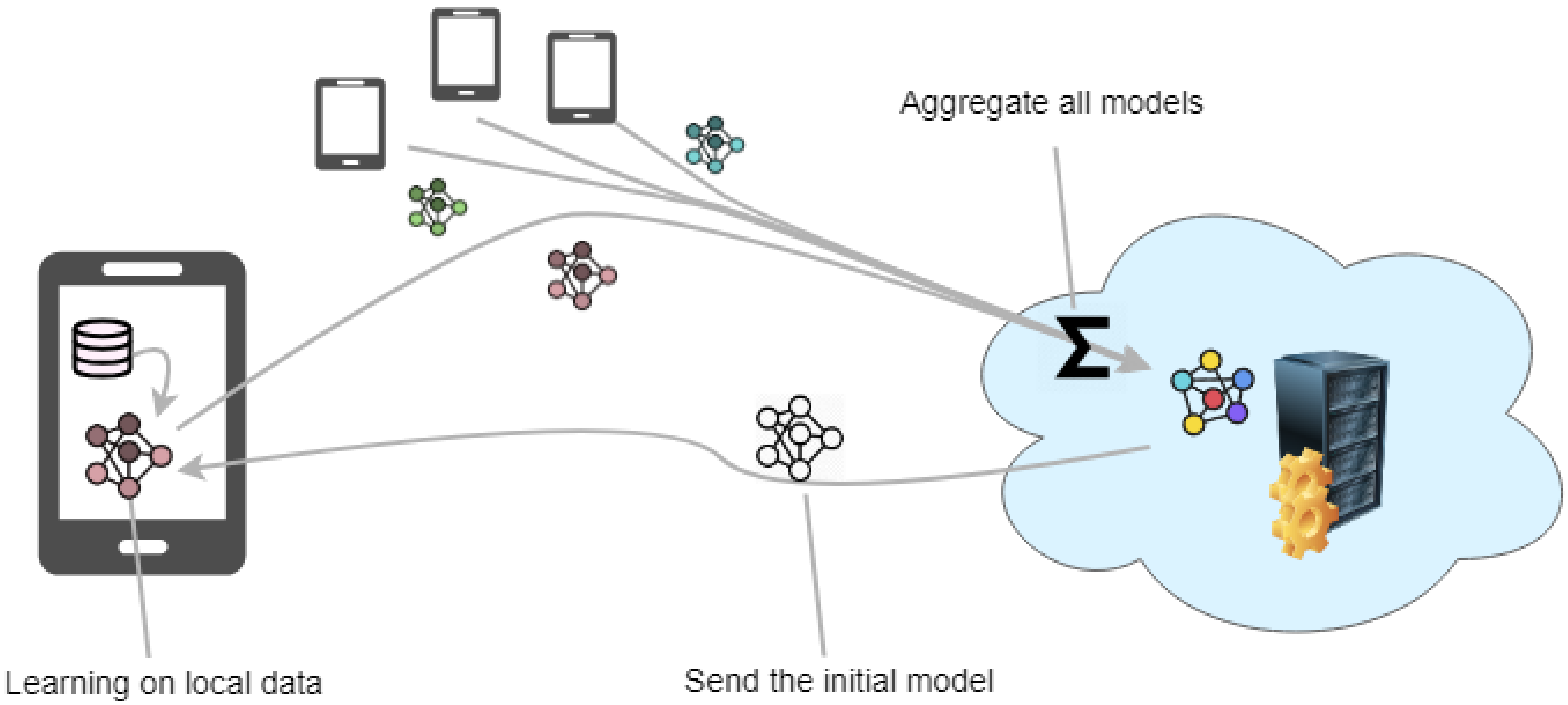
01

Motivation & Problem Formulation

Traditional approach



Federated Learning



Problem Setting

Model parameters

clients/devices

$$\min_{x \in \mathbb{R}^d} \left\{ f(x) \triangleq \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right\}$$

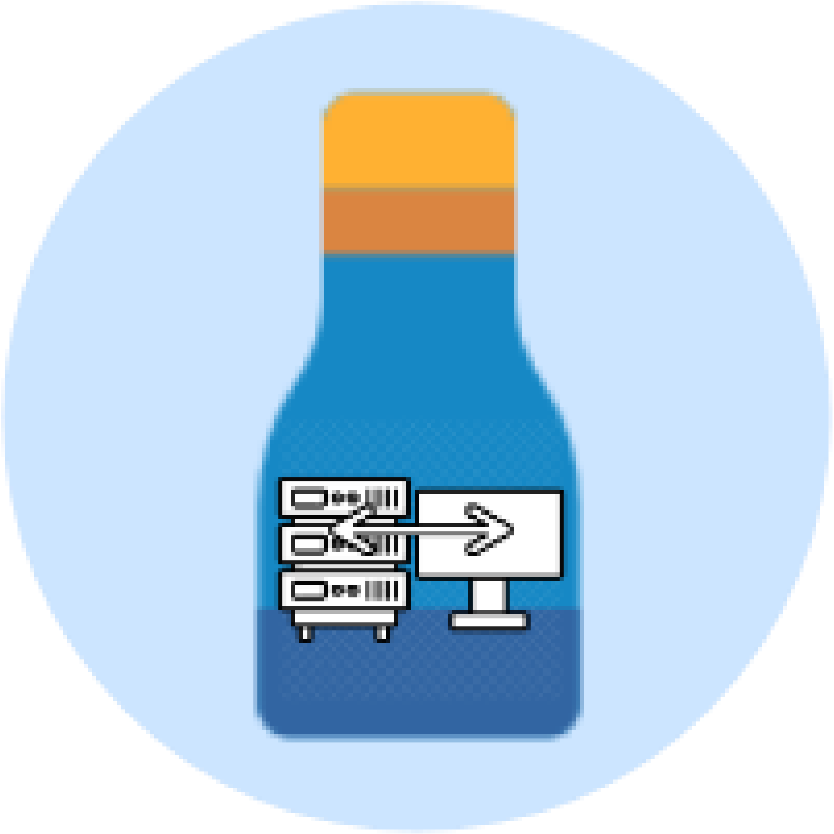
model parameters

Local loss function

02

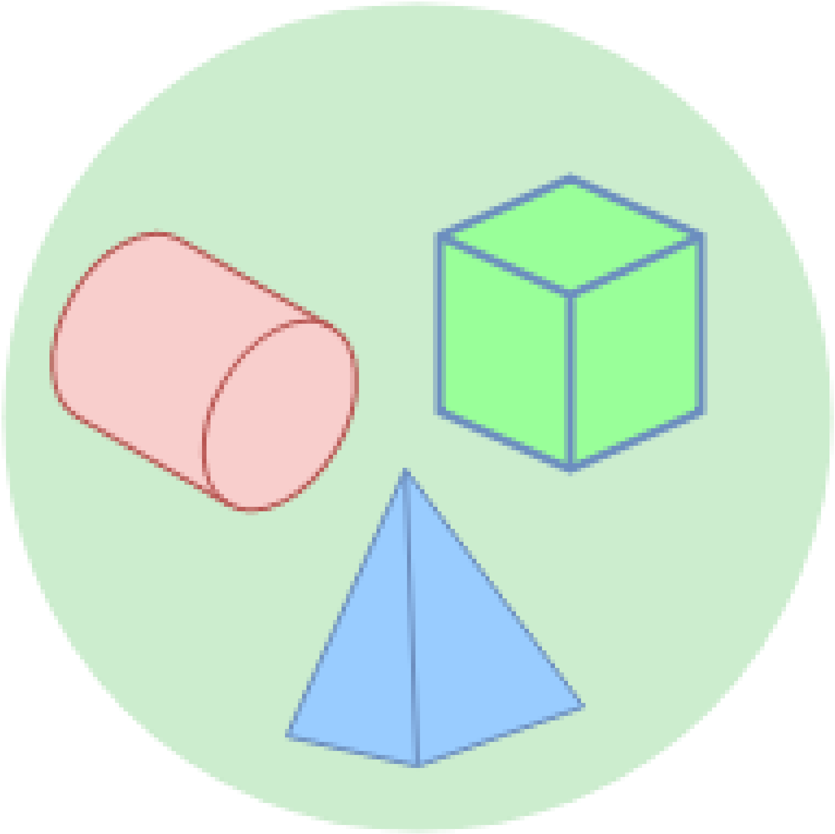
Non-IID Data & Batch Normalization

Challenges



Communication bottleneck

System heterogeneity



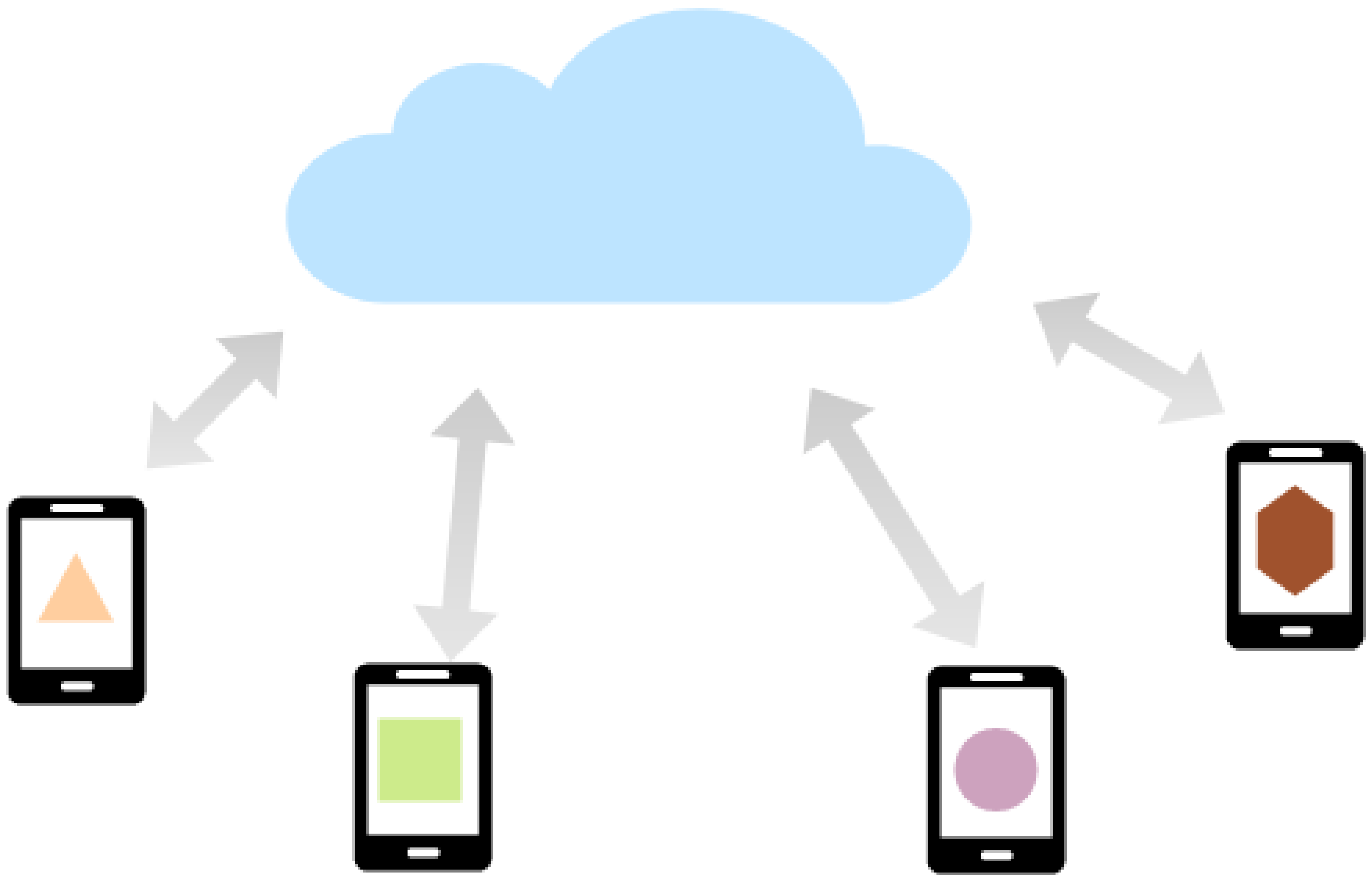
Statistical heterogeneity

Privacy

Non-IID Data

The **IID assumptions** are violated in Federated Learning.

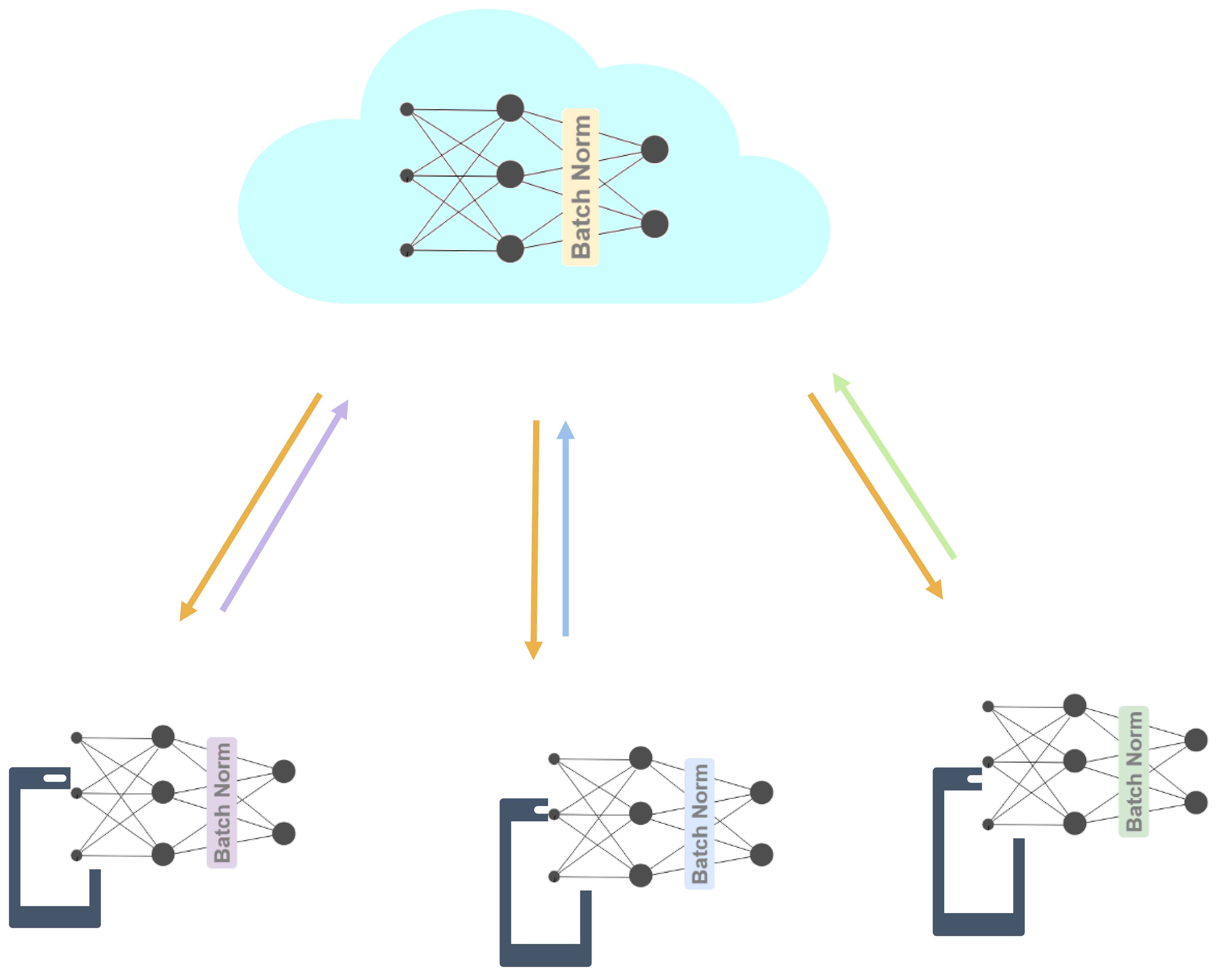
Non-IID data introduces bias into the training and degrades model performance.



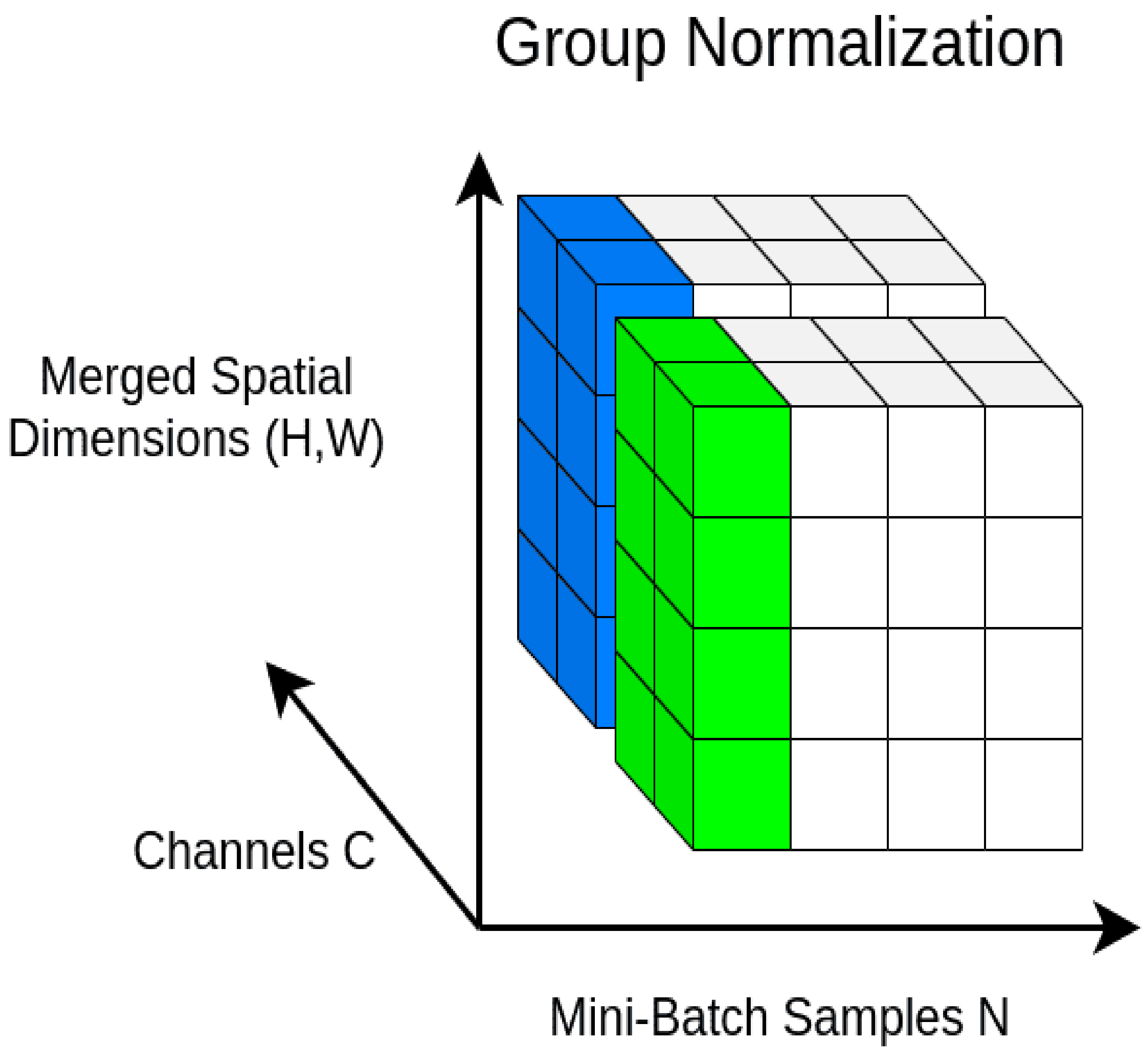
Non-IID Data and Batch Normalization

$$BN(x) = \gamma\hat{x}_i + \beta \quad \textit{such that:} \quad \hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}$$

- BN becomes ineffective in certain settings(small and Non-IID mini-batches (S. Ioffe, 2017)).
- In FL settings, where each data partition could differ from the other, the performance degradation is more severe.



Batch Normalization Alternatives



Input: Values of x over a training mini-batch $\mathcal{B} = \{x_{1\ldots m}\}$; parameters γ, β; current moving mean μ and standard deviation σ; moving average update rate α; maximum allowed correction $r_{\max}, d_{\max}$.
Output: $\{y_i = \text{BatchRenorm}(x_i)\}$; updated μ, σ.

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\sigma_{\mathcal{B}} \leftarrow \sqrt{\epsilon + \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2}$$

$$r \leftarrow \texttt{stop\_gradient}\left(\text{clip}_{[1/r_{\max}, r_{\max}]}\left(\frac{\sigma_{\mathcal{B}}}{\sigma}\right)\right)$$

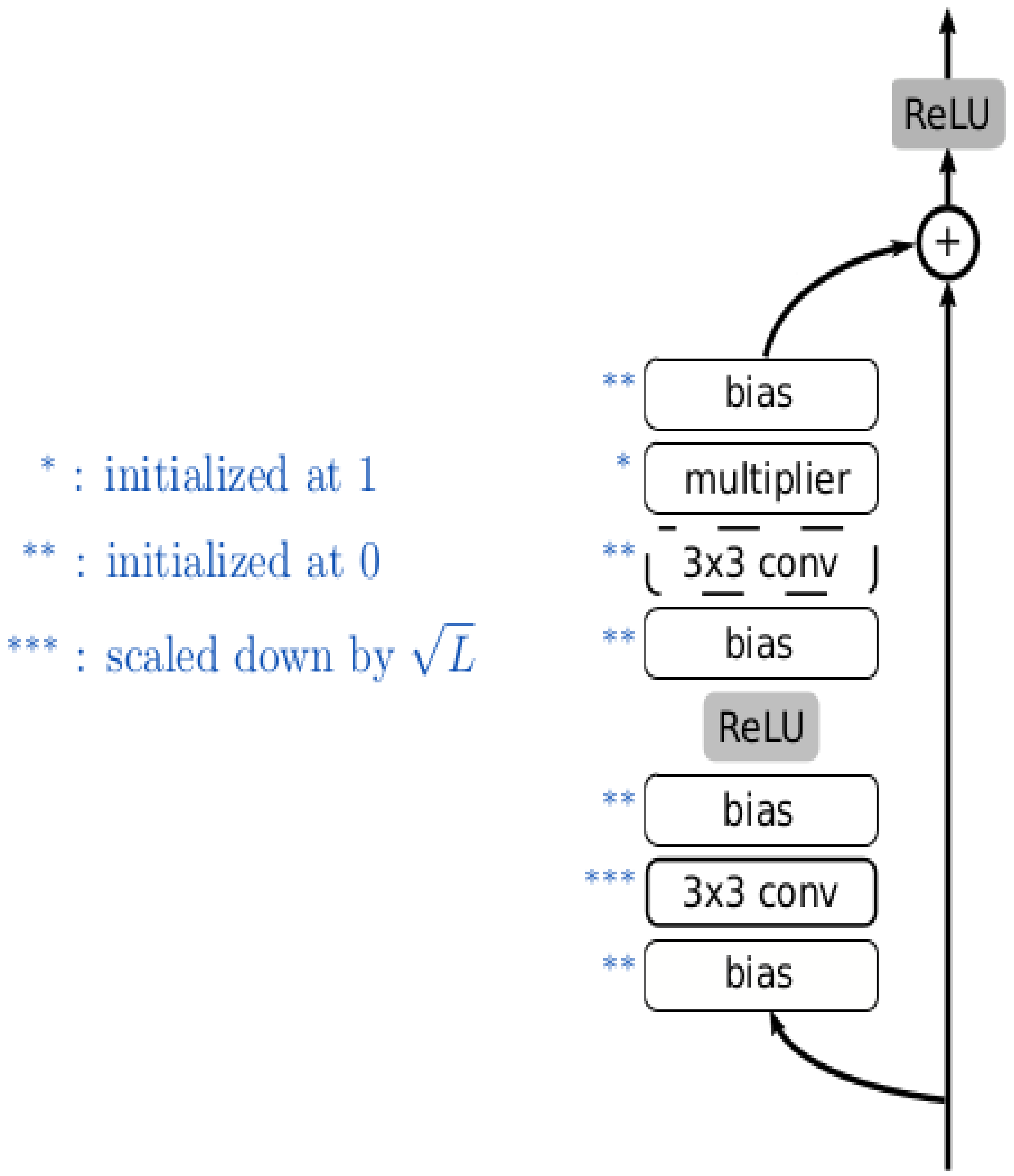
$$d \leftarrow \texttt{stop\_gradient}\left(\text{clip}_{[-d_{\max}, d_{\max}]}\left(\frac{\mu_{\mathcal{B}} - \mu}{\sigma}\right)\right)$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sigma_{\mathcal{B}}} \cdot r + d$$

$$y_i \leftarrow \gamma\,\widehat{x}_i + \beta$$

$$\mu := \mu + \alpha(\mu_{\mathcal{B}} - \mu) \qquad \text{// Update moving averages}$$
$$\sigma := \sigma + \alpha(\sigma_{\mathcal{B}} - \sigma)$$

Inference: $y \leftarrow \gamma \cdot \frac{x - \mu}{\sigma} + \beta$



Group Normalization
Y. Wu et al., 2018

Batch Renormalization
S. Ioffe, 2017

Fixup Resnet
H. Zhang et al., 2019

Our aim

Rectify the drawbacks of BN in Federated Learning run on Non-IID data

03

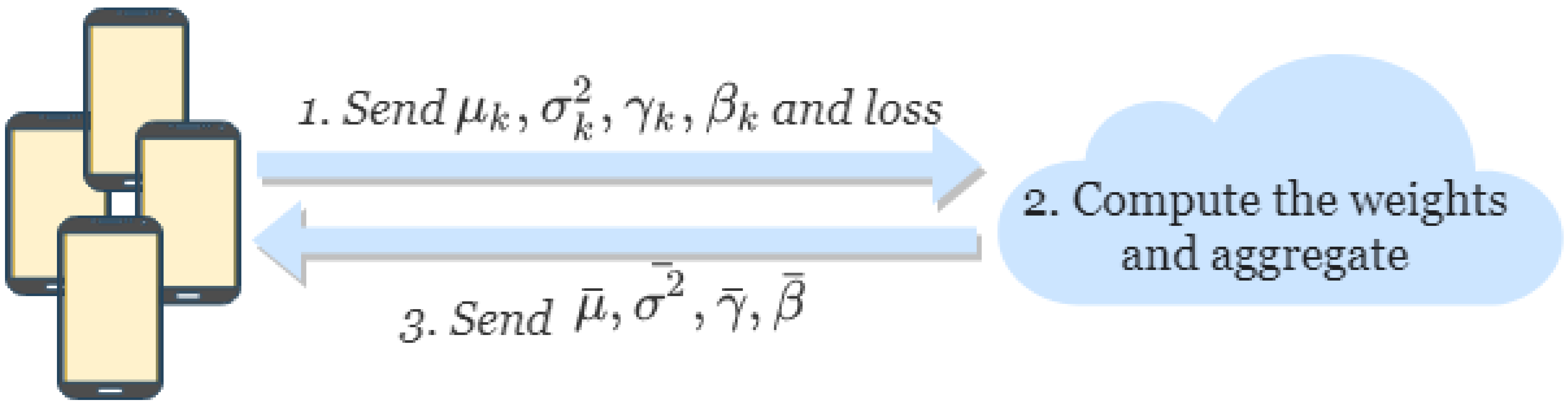
Contribution

Contribution

1. Each participating client shares the BN parameters used in the normalization layers with the central server, as well as the loss value obtained in the current round.

2. The server, based on the local loss values, calculates the importance of each client, i.e. higher weight is given to the client with the highest local loss. Then aggregates all the local models.

3. The updated parameters are sent back to the new selected clients for the next round. Each client updates both training and inference statistics based on those received from the server



Contribution

Compute the weights' matrix :

$$W_k = \frac{F_k(w)}{\sum_{k \epsilon S_t} F_k(w)}$$

Aggregate :

$$w_{t+1} = \sum_{k \epsilon S_t} W_k \; w_{t+1}^k$$

Condition:

$$std(F(w)) < \varepsilon$$

Switch to FedProx

Algorithm 1 *FedBS*: The K clients are indexed by k, E is the number of local epochs, $\mathcal{B}$ is a set of mini-batches each of size m, η is the learning rate, and ϵ is a positive small number.

Server executes:
- initialize w_0
- **for** each round $t = 0, 1, \ldots$ **do**
  - m $\leftarrow$ max(C·K, 1)
  - $S_t \leftarrow$ (select a random set of m clients)
  - send w_t to each client k $\in S_t$
  - **for** each client k $\in S_t$ **do**
    - $w_{k,t+1}, F_k(w) \leftarrow ClientUpdate(k, w_t)$
  - **end for**
  - $W_k = \frac{F_k(w)}{\sum_{k \in S_t} F_k(w)}$
  - $w_{t+1} = \sum_{k \in S_t} W_k w_{t+1}^k$
  - **until** $std(F(w)) < \epsilon$
  - switch to *FedProx*
- **end for**

ClientUpdate(k, w):
- **for** each i from 1 to E **do**
  - **for** batch $b \in \mathcal{B}$ **do**
    - $w \leftarrow w - \eta \nabla \mathcal{L}(w; b)$
    - computer $F_k(w)$
  - **end for**
- **end for**
- return w and $F_k(w)$ to server

04

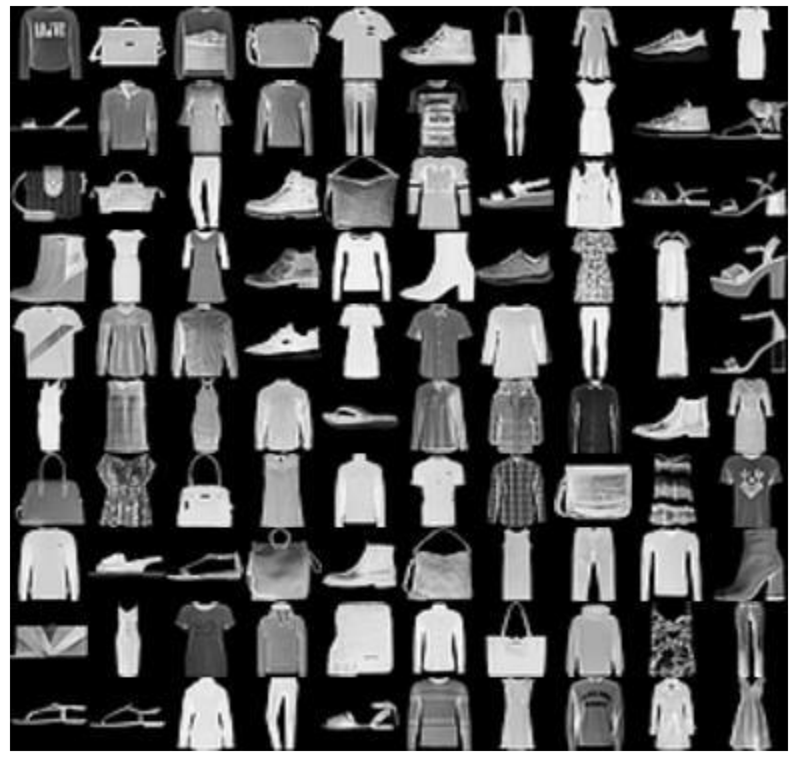
Evaluation

Evaluation

Datasets:

Cifar-10

Mnist



Fashion-Mnist

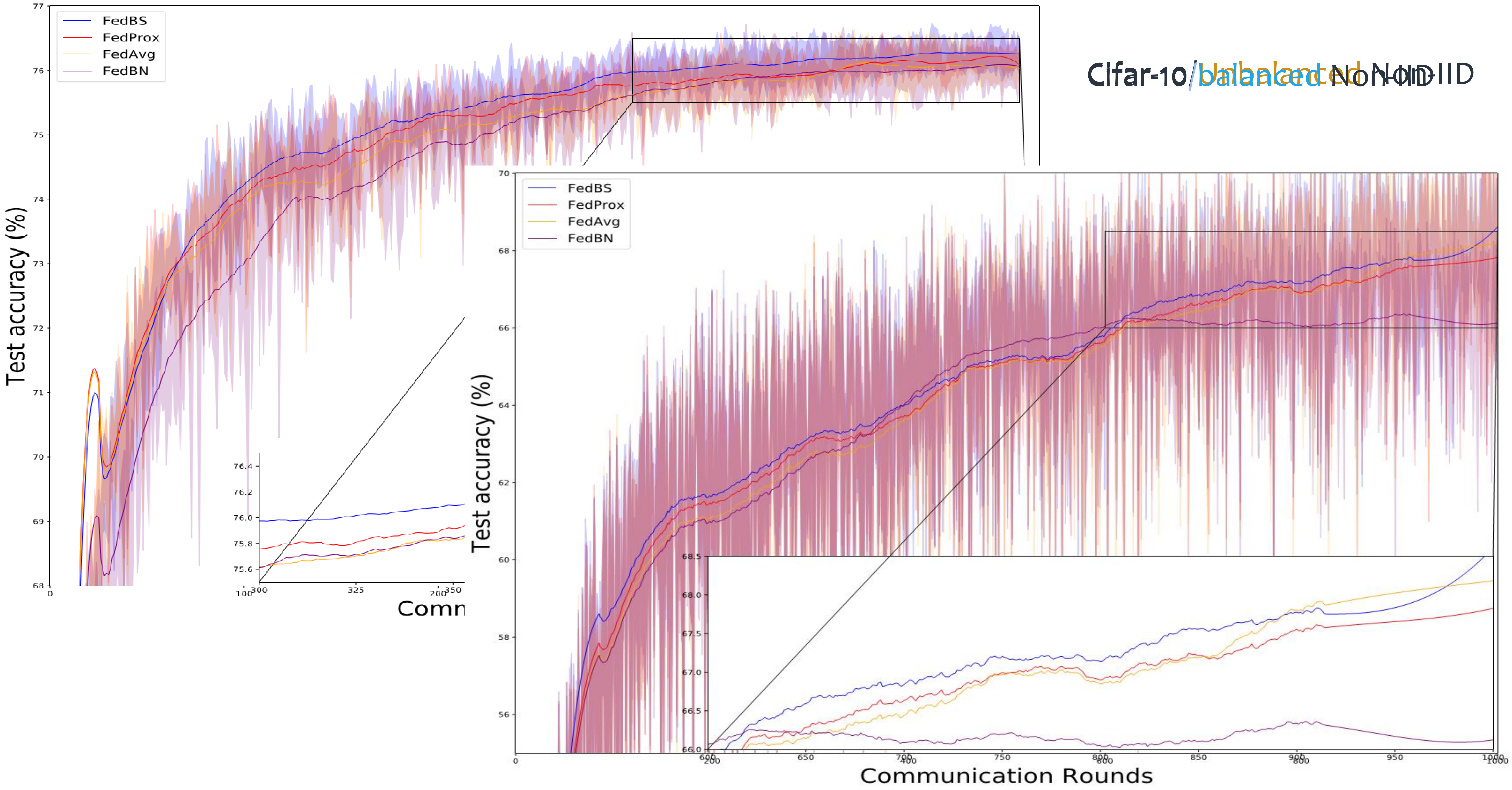
Non-IID Settings:

- Balanced distribution skew
- Unbalanced distribution skew

Baseline:

- FedAvg McMahan et al. 2016
- FedProx Li et al. 2018
- FedBN X. Li et al. 2021

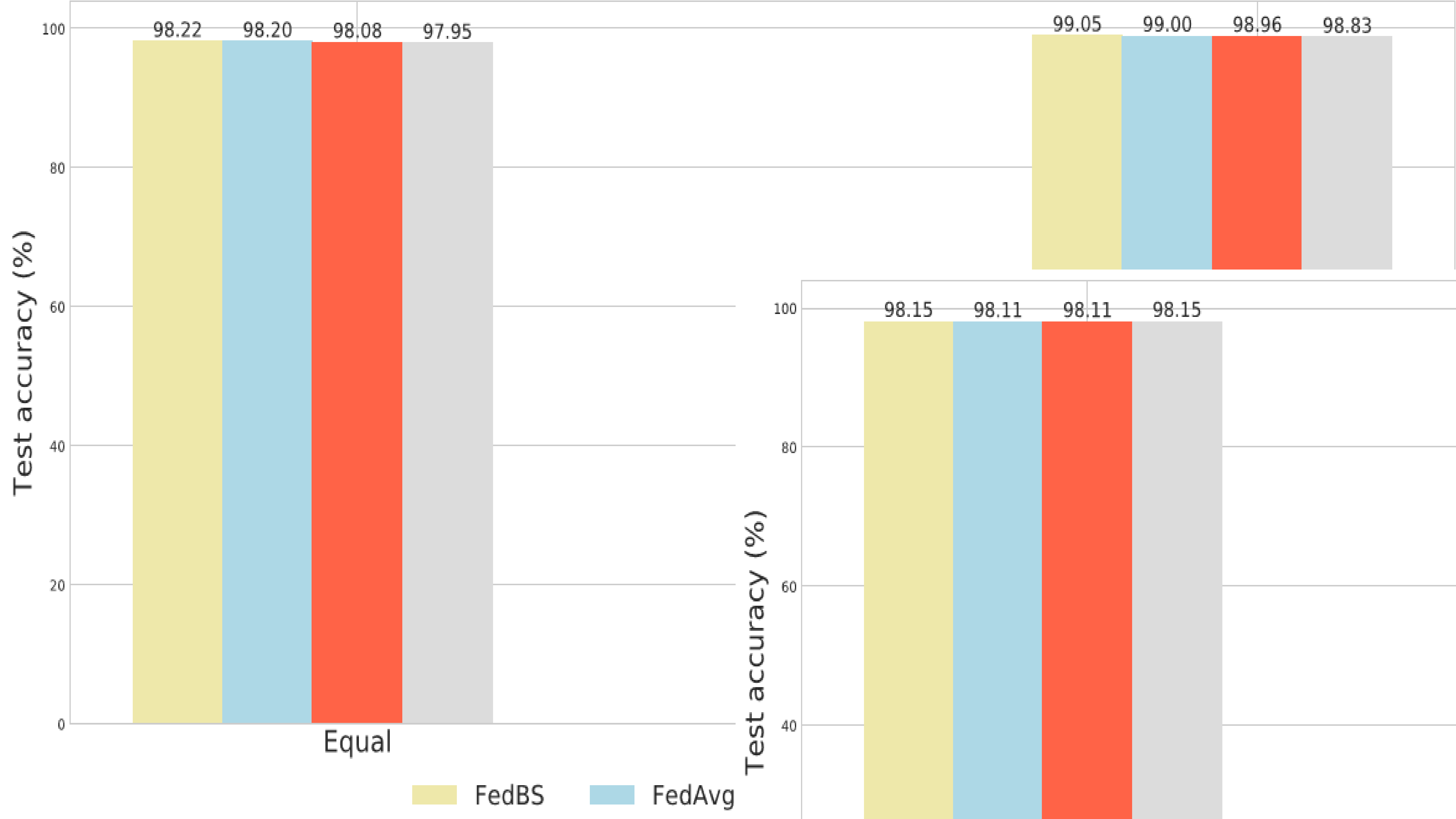
Experimental Results

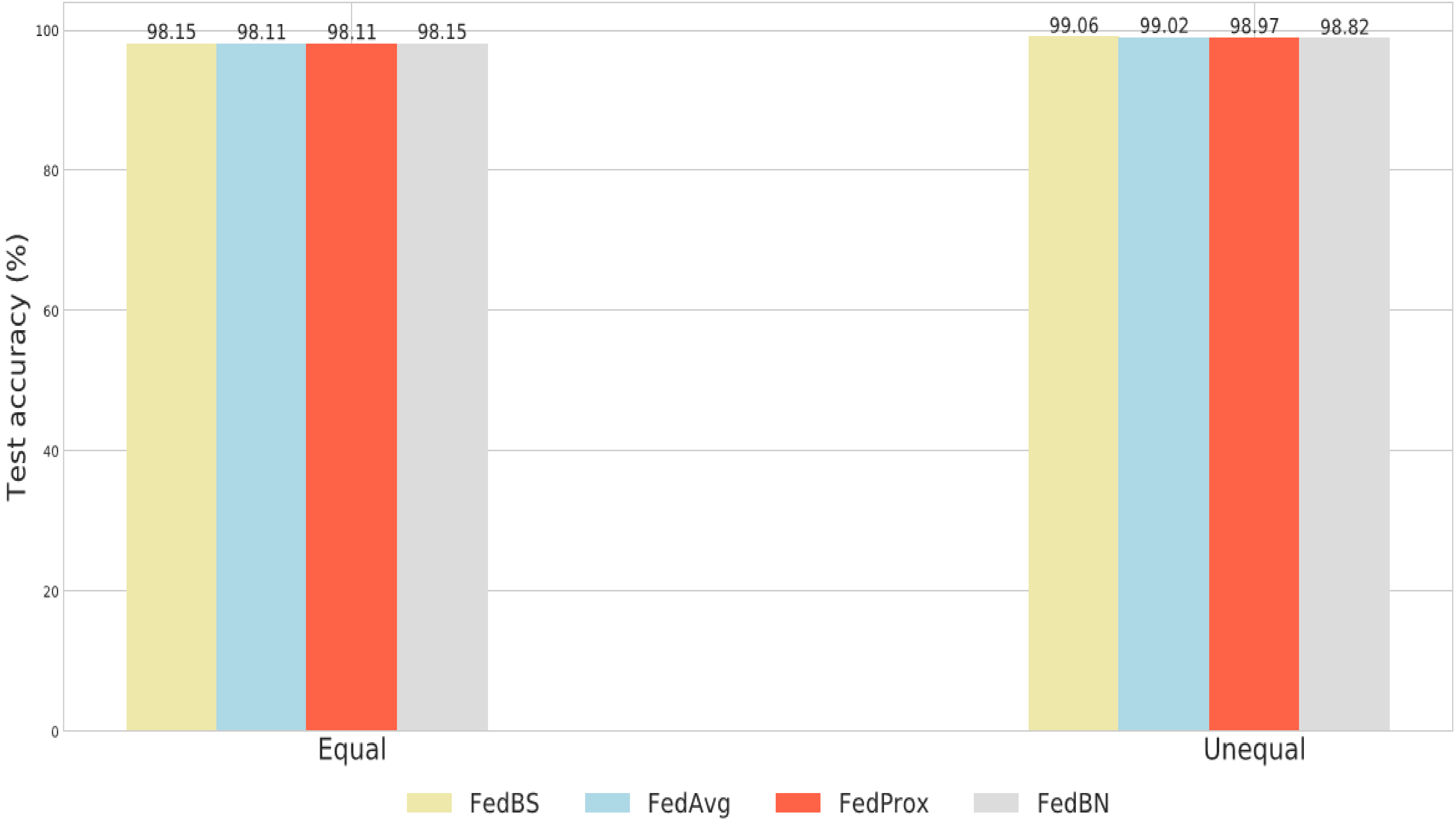


Cifar-10/balanced Non-IID

Unbalanced Non-IID

Experimental Results



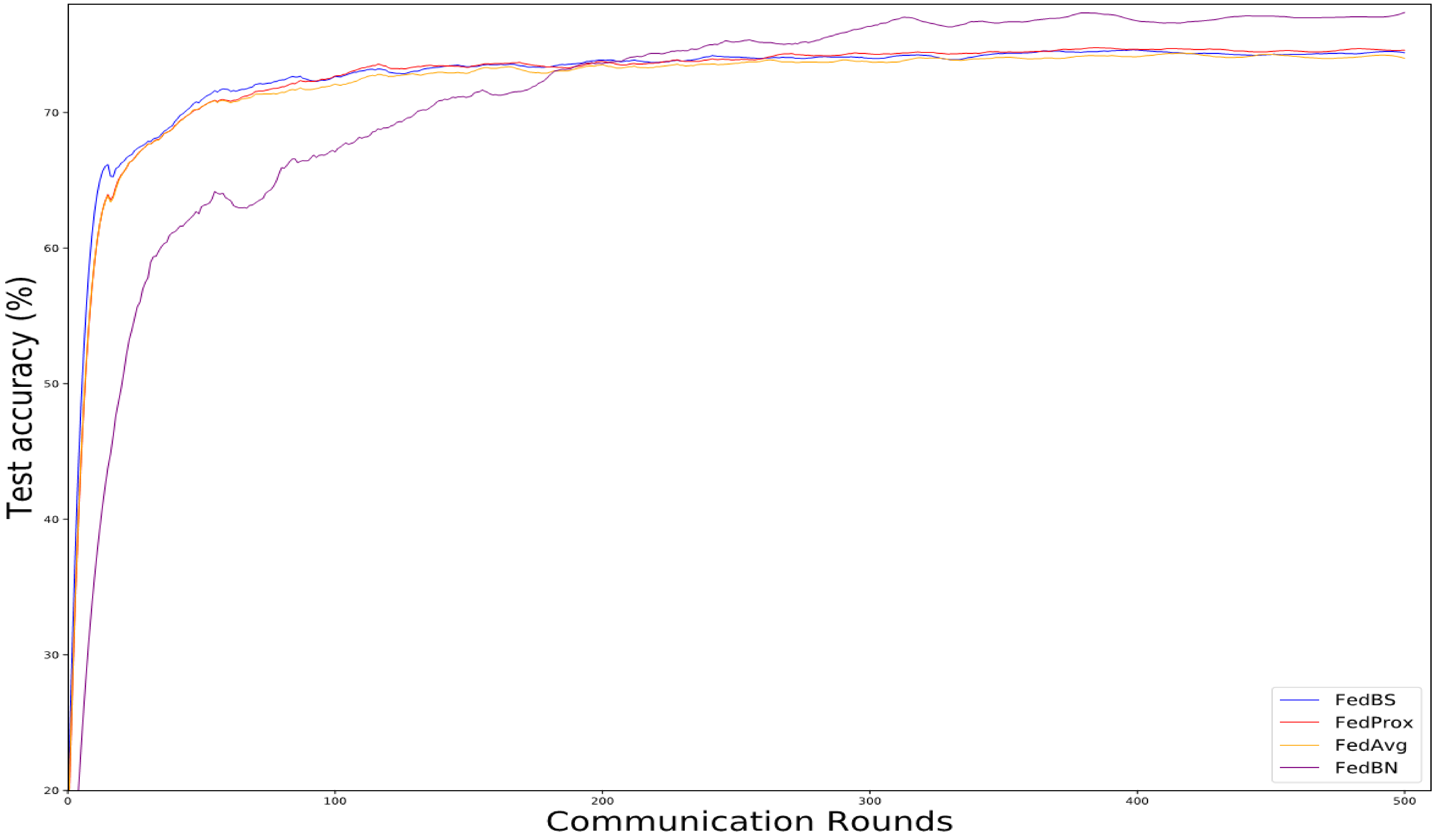


Fashion-Mnist/balanced & unbalanced Non-IID

Mnist/balanced & unbalanced Non-IID

Experimental Results

Cifar-10/balanced Non-IID (locally)



05

Conclusion & Perspectives

Conclusion and Perspectives

Conclusion:

- Introduction of a novel approach FedBS for handling DNN models with normalization layers in FL.
- FedBS weights the local models based on each client's loss value.
- FedBS outperformed all the state-of-the-art approaches.

Perspectives:

- Secure the communication of batch parameters between clients and the server.
- Tackle Non-IID issues as key point and improve FL models without sacrificing FL properties

Meryem Janati Idrissi

MOHAMMED VI POLYTECHNIC UNIVERSITY

Ismail Berrada

MOHAMMED VI POLYTECHNIC UNIVERSITY

Noubir Guevera

NORTHEASTERN UNIVERSITY