Hybrid FL: Algorithms and Implementation

Xinwei Zhang* Wotao Yin† Mingyi Hong* Tianyi Chen#

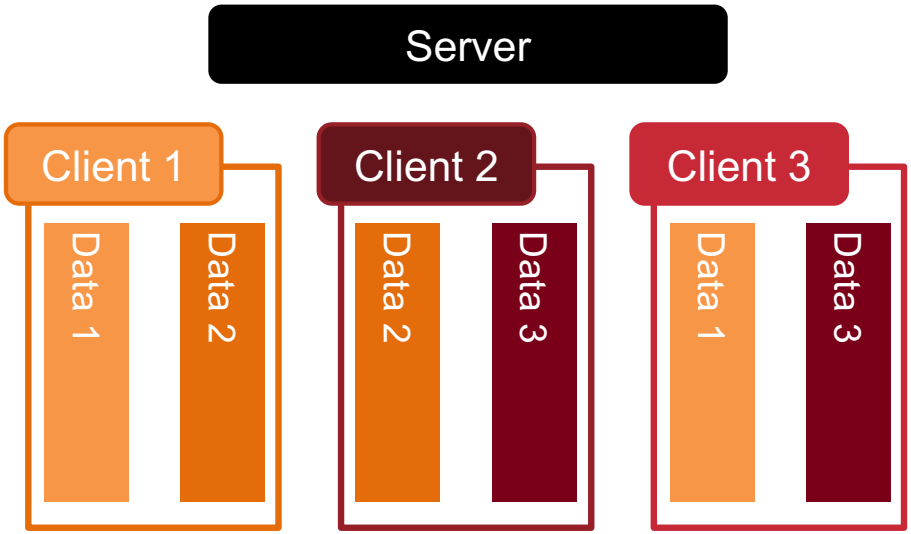
*University of Minnesota
† UCLA
# RPI

NeurIPS-SpicyFL 2020
February 13, 2021

❑ Introduction

- Horizontal FL & Vertical FL
- Limitations of HFL & VFL
- Hybrid FL - Setting & Challenge

❑ Horizontal & Vertical FL



Horizontal FL

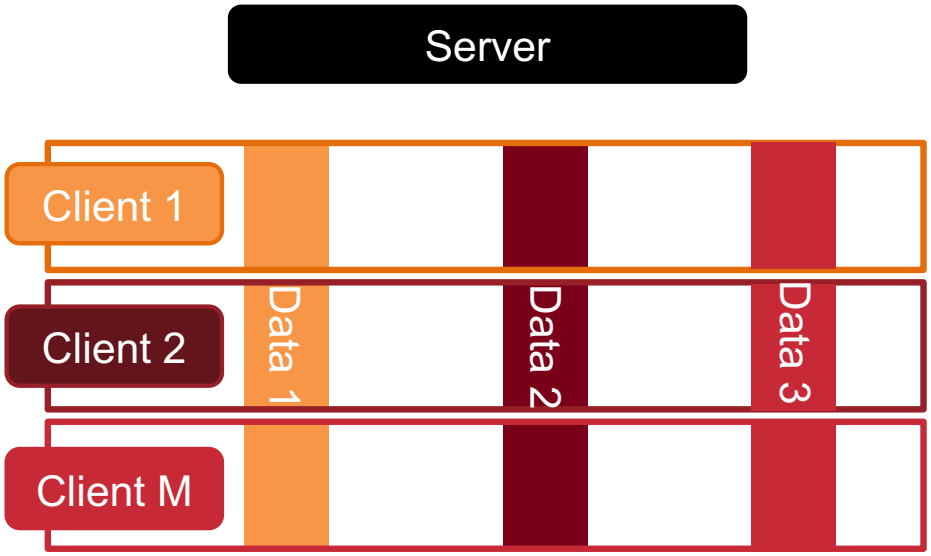
- Each client: full feature, full model, partial data id
- Application
  - Google Keyboard [1]
  - Image classification [2]
- Algorithm
  - FedAvg [3], Scaffold [4], …

[1] Bonawitz K, Eichner H, Grieskamp W, et al. Towards federated learning at scale: System design. arXiv 2019.
[2] Li W, Milletarì F, Xu D, et al. Privacy-preserving federated brain tumour segmentation, MLMI 2019 .
[3] Li X, Huang K, Yang W, et al. On the Convergence of FedAvg on Non-IID Data, ICML 2019.
[4] Karimireddy S P, Kale S, Mohri M, et al. Scaffold: Stochastic controlled averaging for federated learning, ICML 2020.

❑ Horizontal & Vertical FL

Vertical FL

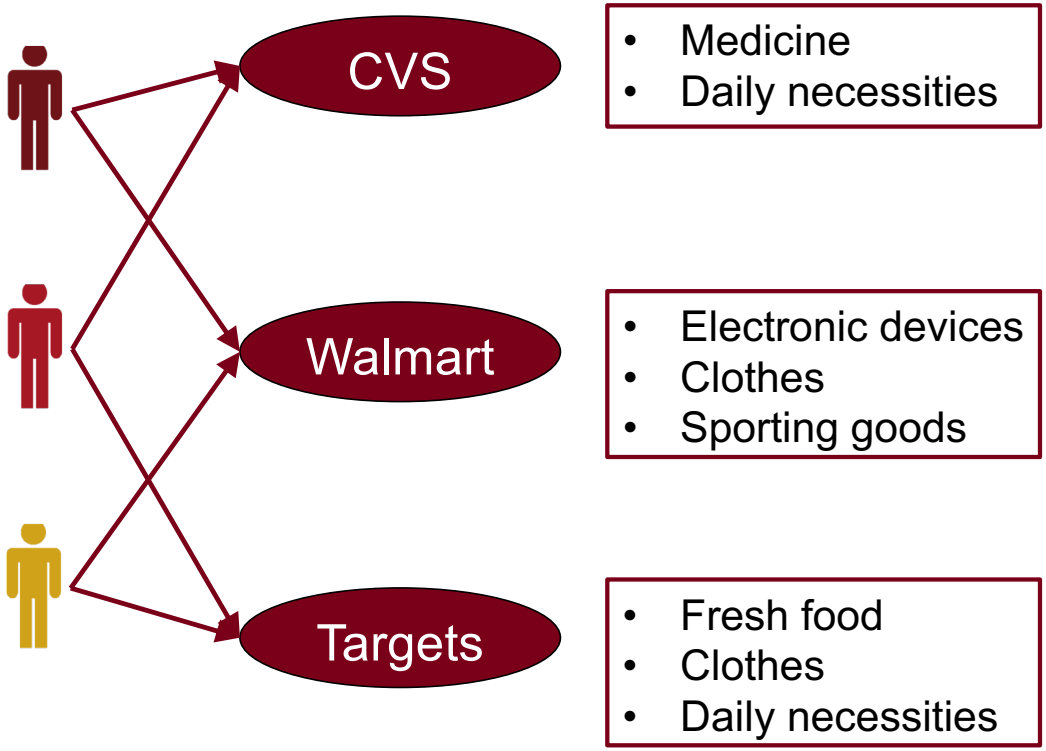
- Each client: full data id, partial features; partial model
- Training done at the server (has access to data)
- Application
  - B2B distributed regression [5]
- Algorithm
  - FedBCD [5], VAFL [6]



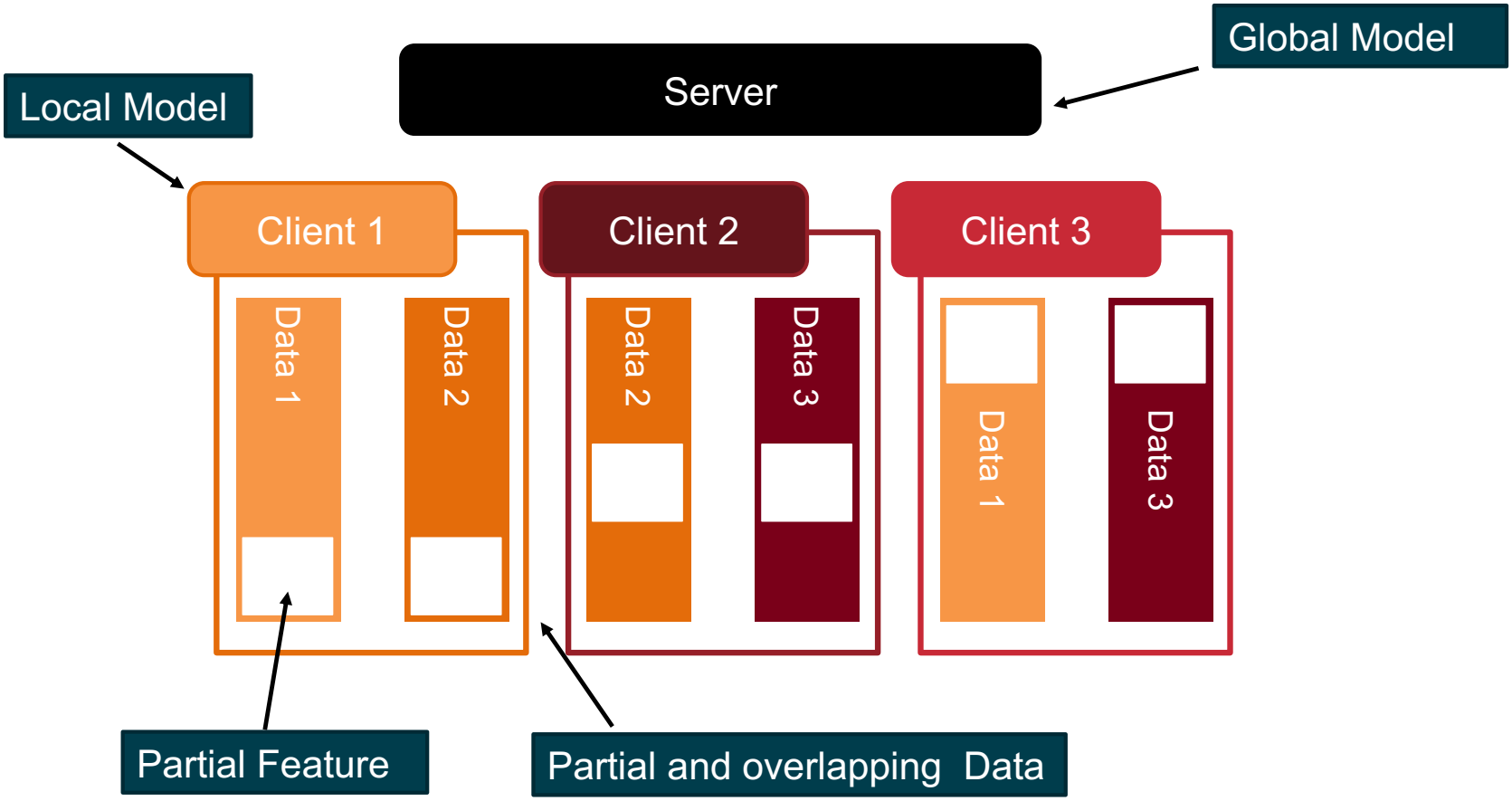
[5] Liu Y, Kang Y, Zhang X, et al. A communication efficient vertical federated learning framework, arXiv 2019.
[6] Chen T, Jin X, Sun Y, et al. VAFL: a Method of Vertical Asynchronous Federated Learning. arXiv 2020.

❑ Hybrid FL

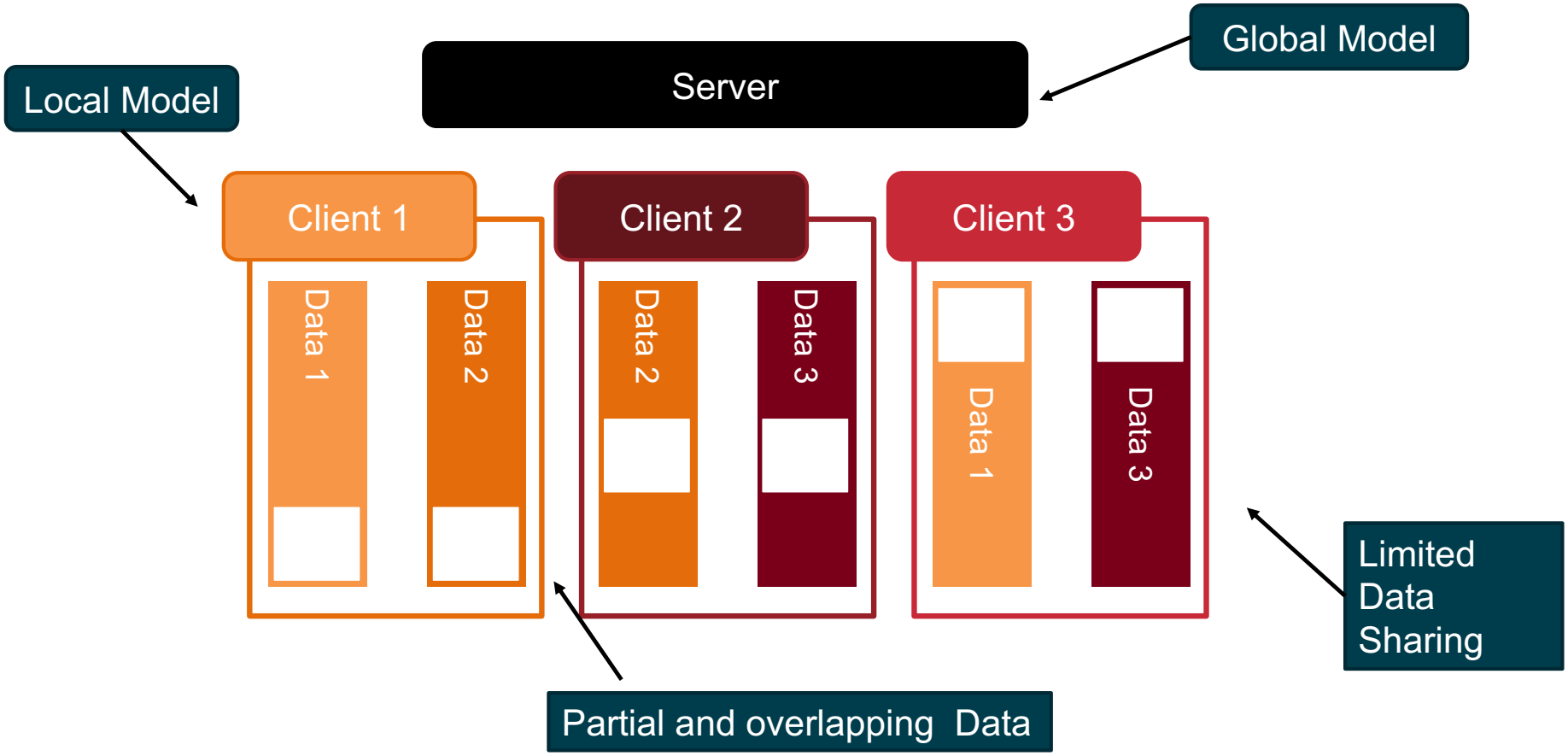
- Partial feature & partial data id
- Supermarkets:
  - Client: Walmart, Amazon, Ebay, Targets
  - Sample index: costumers
  - Feature: shopping records on different categories of products
  - Goal: jointly predict customers' behavior
- Not covered by HFL & VFL



Hybrid FL: Setting



Hybrid FL: Challenges



❑ Our Contribution

- Propose a new and flexible Hybrid FL problem formulation.
- Develop an efficient algorithm for Hybrid FL
  1. enables knowledge sharing
  2. maintains data locality
  3. without requiring sample synchronization
- Numerical results on the real data set show competitive performance to centralized training.

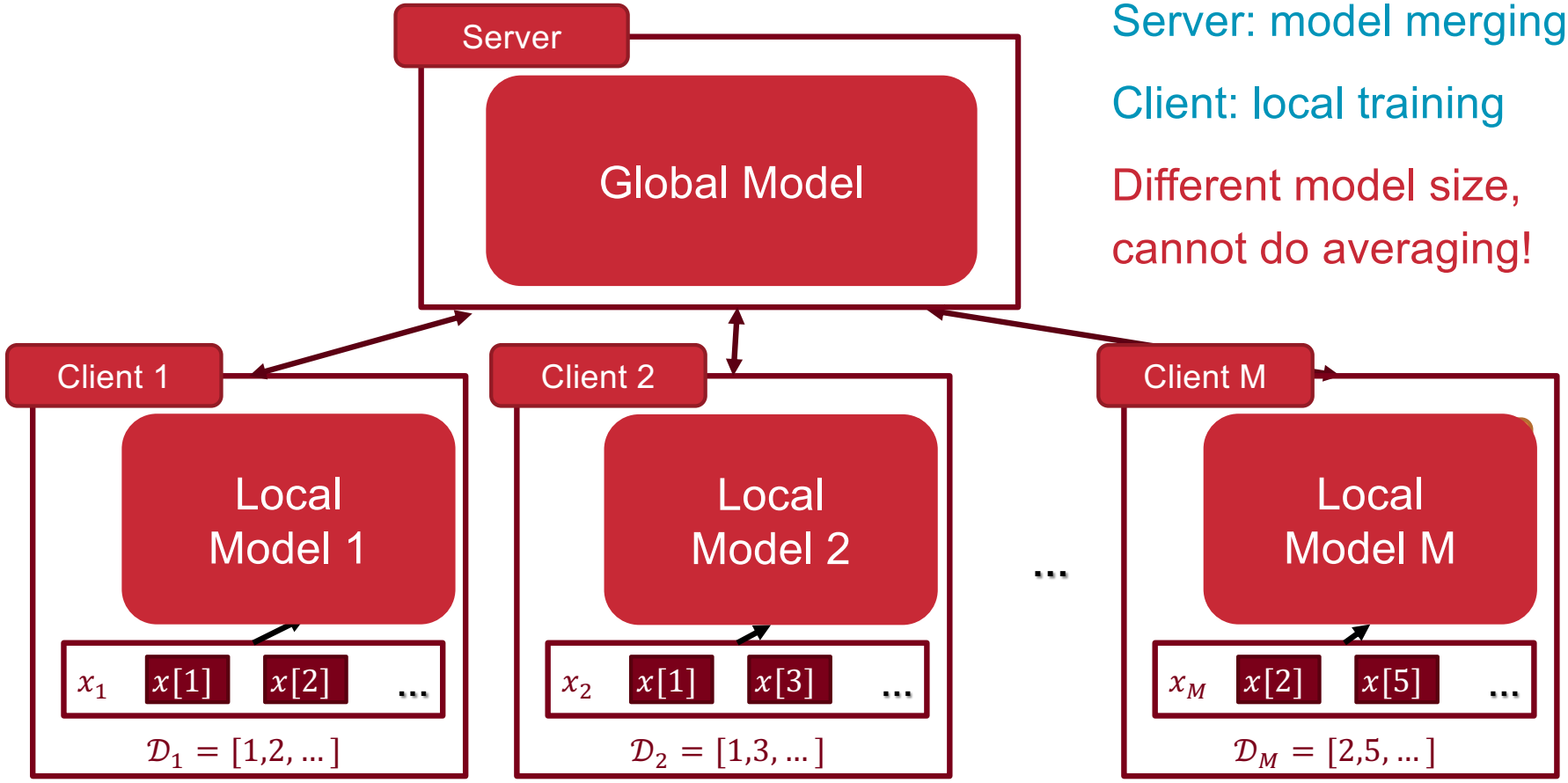
❑ Hybrid FL Model Design

- System Design
- Problem Formulation
- Regularizer Design

System Structure

- Recall:
  - Challenge 1: no data sharing
  - Challenge 2: build both local and global models
  - Challenge 3: sample synchronization
- Our solution:
  - Local training + Model merging

System Model



Server: model merging

Client: local training

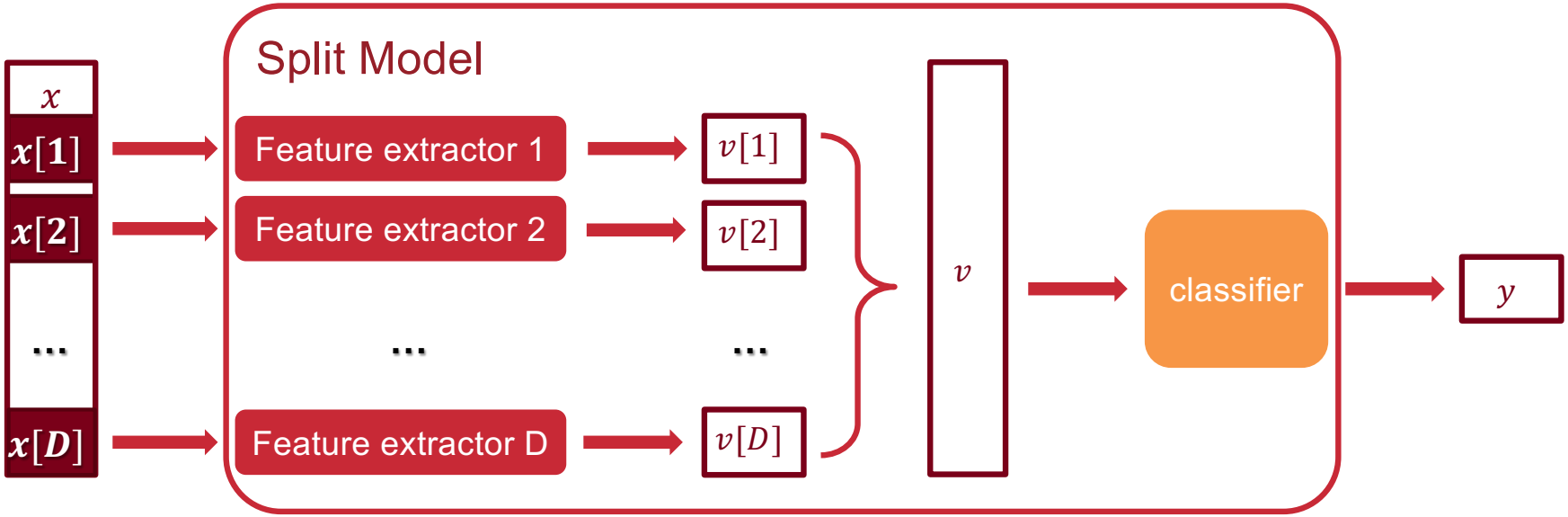
Different model size, cannot do averaging!

❑ How to Merge Models?

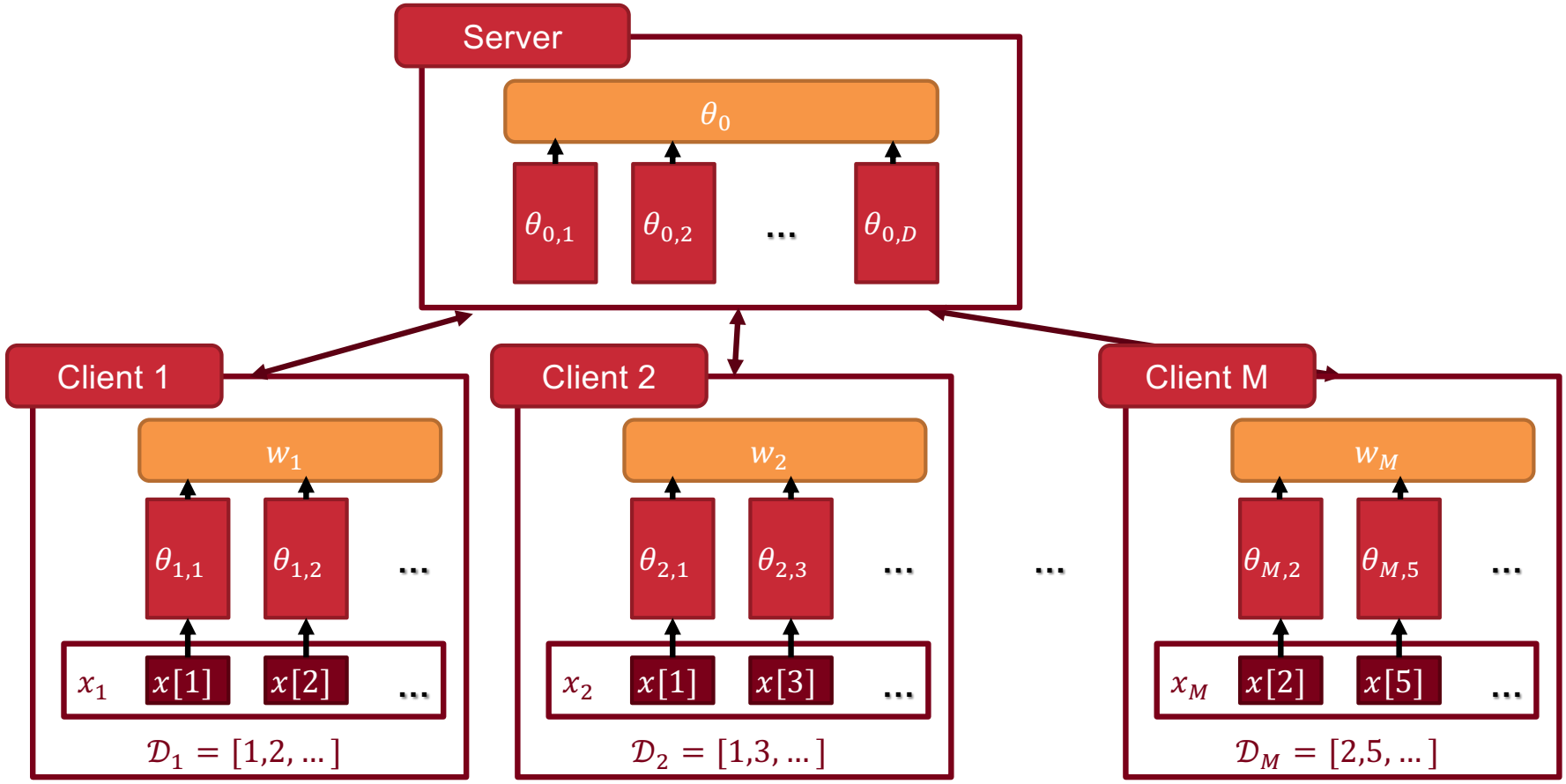
- Global models and local models have different sizes
- However, they have overlapping functionality blocks
- Our Strategy: Identify and Match the blocks with similar functionalities between client and server

❑ How to Identify Similar Functionalities?

- **Assumption**: Same feature should be processed by the same feature extractor

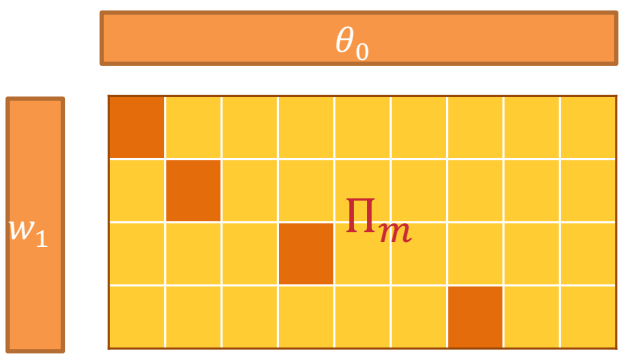


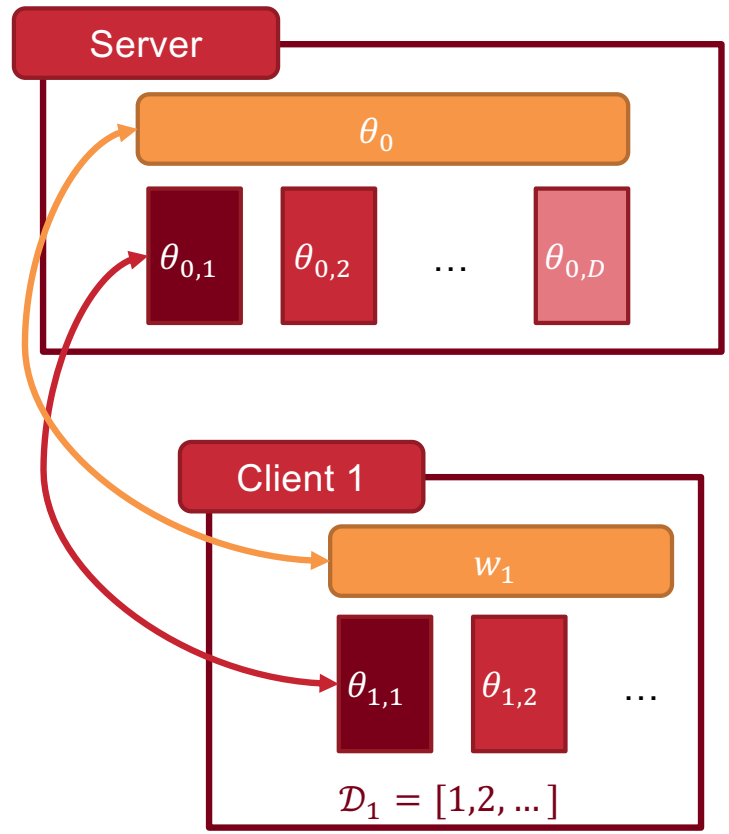
A Refined System Model



❑ How to Match?

- Extractors of the same feature should be similar: $\theta_{m,d} \approx \theta_{0,d}, d = 1, \dots, D$
- The global classifier is an (linear [7]) ensemble of the local ones; needs to find the best matching matrix Π_m





[7] Wang H, Yurochkin M, Sun Y, et al., *Federated Learning with Matched Averaging*, ICLR 2020.

❑ Problem Formulation

$$\min_{\{\{\theta_{m,d}\},w_m\}} \underbrace{\frac{1}{MN}\sum_{m=1}^{M}\sum_{n\in\mathcal{N}_m}\ell_m\left(h_m(\{\theta_{m,d}\};x_m^n);w_m;y^n\right)}_{\text{Classification loss}} + \underbrace{r(\{\{\theta_{m,d}\},w_m\};\theta_0,\{\theta_{0,d}\})}_{\text{Model matching loss}}$$

- Classification part optimizes local models by the local agents;
- Regularizers part matches extractor and classifiers
  - Enable knowledge sharing
- Optimization variables:
  - extractors $\{\theta_{m,d}\}_{d\in\mathcal{D}_m}$
  - matching patterns Π_m
  - classifiers θ_0, w_m

❑ Hybrid FL Algorithm Design

- Problem Separation
- HyFEM Algorithm
- Algorithm Analysis

❑ Problem at Client

- Local problem at each client m:

Local classification loss

$$\min_{\theta_m, w_m} f_m(\theta_m, w_m) = \sum_{n \in \mathcal{N}_m} \ell_m \left(h(\theta_m; x_m^n); w_m; y_m^n\right)$$
$$+ \sum_{d \in \mathcal{D}_m} \mathrm{dist}(\theta_{m,d}, \theta_{0,d}) + \mu \mathrm{dist}(\Pi_m \theta_0, w_m);$$

Feature extractor regularizer

Classifier regularizer

1. Use local data, global model to update local model

❑ Problem at Server

- Global problem at server:

$$\min_{\theta_{0,d}} \sum_{m=1}^{M} \sum_{d \in \mathcal{D}_m} \mathrm{dist}(\theta_{m,d}, \theta_{0,d});$$

Feature extractors regularizer

$$\min_{\theta_0, \{\Pi_m\}} \sum_{m=1}^{M} \mathrm{dist}(\Pi_m \theta_0, w_m),$$
$$\text{s.t. } \Pi_m 1 = 1, m = 1, \ldots, M.$$

Classifier regularizer

- No local data is used!

❑ Hybrid Federated Matching Algorithm (HyFEM)

Block-type Update

For $t = 0, \ldots, T-1$ do

For each client in parallel does

$$\{\theta_{m,d}^{r,Q}\}, w_m^{r,Q} = \arg\min f_m(\{\theta_{m,d}\}, w_m)$$

Local updates for Q iterations

For the server does

$$\theta_{0,d}^{t+1} = \frac{1}{|\{m: d \in \mathcal{D}_m\}|} \sum_{m:d\in\mathcal{D}_m} \theta_{m,d}^{t,Q}$$

Merge the feature extractors

Iteratively do

1. Pick client index m'
2. $\Pi_{m'} = \arg\min_{\Pi_{m'}} \mathrm{dist}(w_m, \Pi_{m'}\theta_0)$
3. $\theta_0 = \arg\min_{\theta_0} \sum_{m=1}^{M} dist(w_m, \Pi_m\theta_0)$

Merge the classifiers

❑ Algorithm Analysis

1. Server does NOT have direct access local data
2. Avoids sample synchronization issue
3. Local models deal with local data with partial features
4. Global model deals with the full featured data

Under mild condition on ℓ_m's, the algorithm converges by extending the proof of the BCD algorithm [Razaviyayn et al 13].

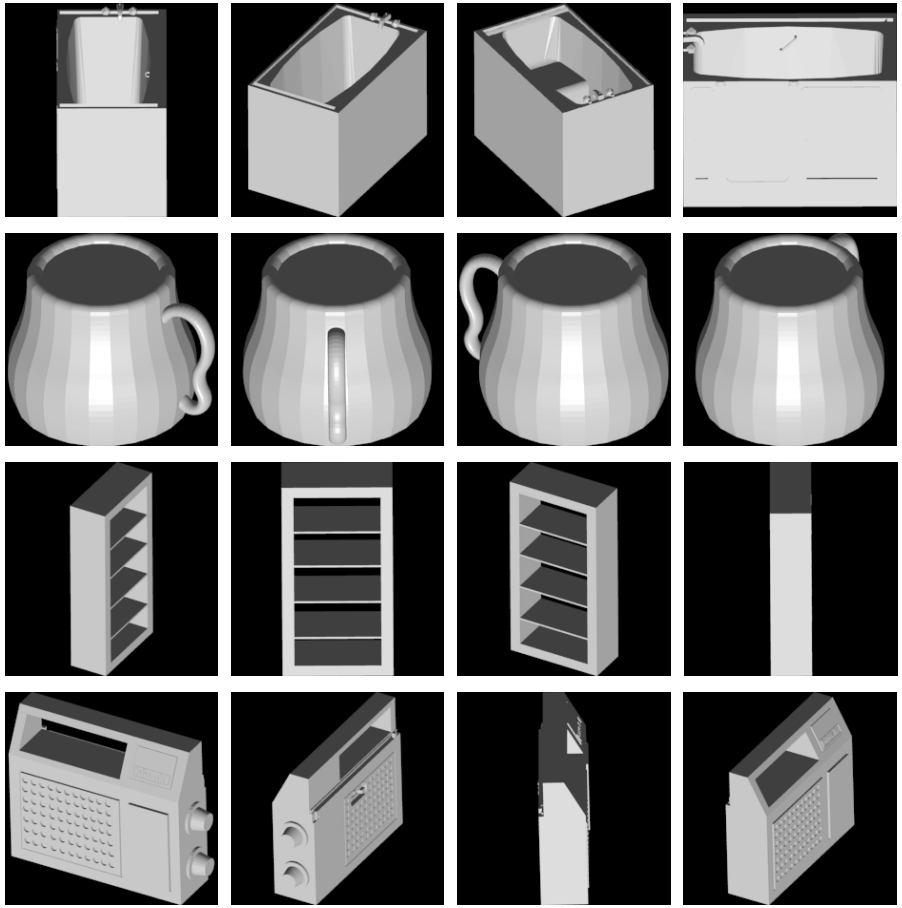
Razaviyayn M, Hong M, Luo Z Q. A unified convergence analysis of block successive minimization methods for nonsmooth optimization, SIAM Journal on Optimization 2013.

❑ Numerical Results

- Experiment Design
- Parameter Settings
- Results

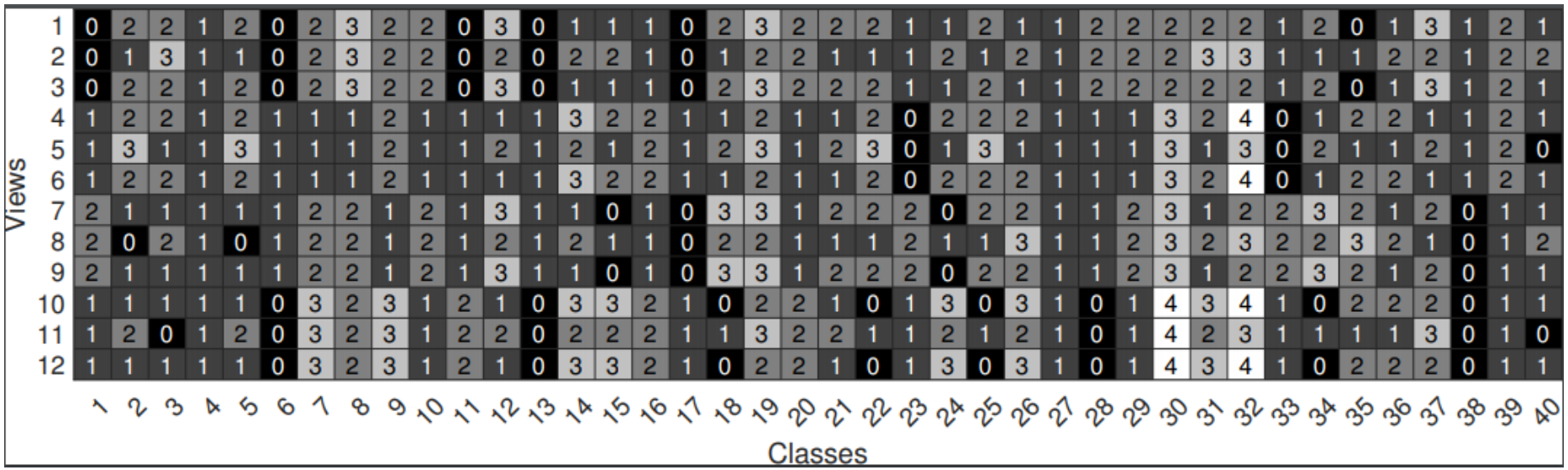
❑ Experiment Design

- ModelNet40
  - 40,000 train + 4000 test data
  - 40 classes
  - 12 views/sample (feature blocks)
- Model
  - Feature extractor: CNN layers of ResNet-34
  - Classifier: FC net with 1 hidden layer
- Training set
  - Partial views of partial classes / client
- Testing set
  - Local: partial views of all classes
  - Global: full views of all classes

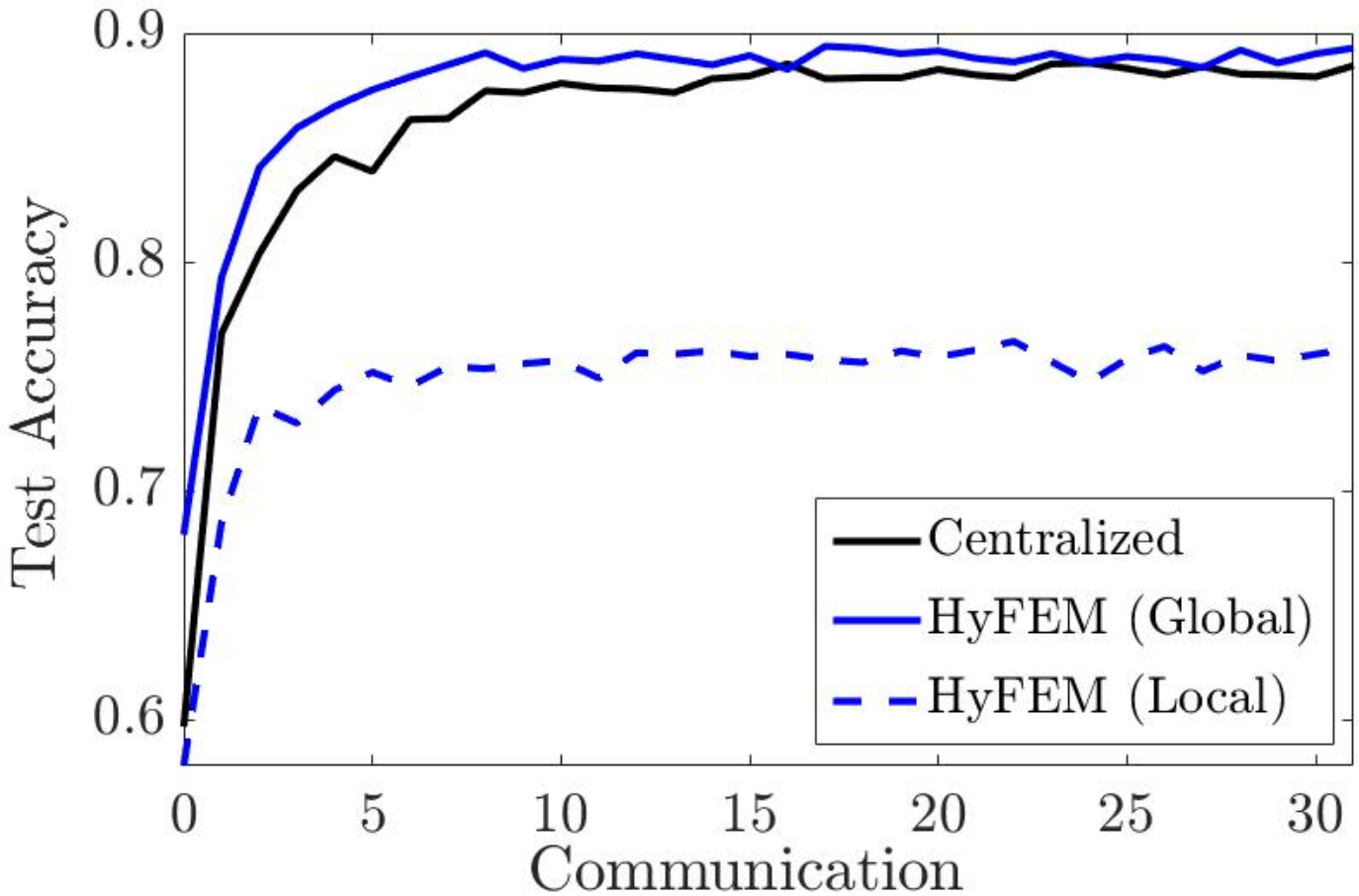


Experiment Settings

Setting	Total view $\lvert D \rvert$	# Client M	# Views $\lvert D_m \rvert$	# Classes	Used Data
1	4	4	2	20	75%
2			3	30	100%
3	12	8	6	15	88%



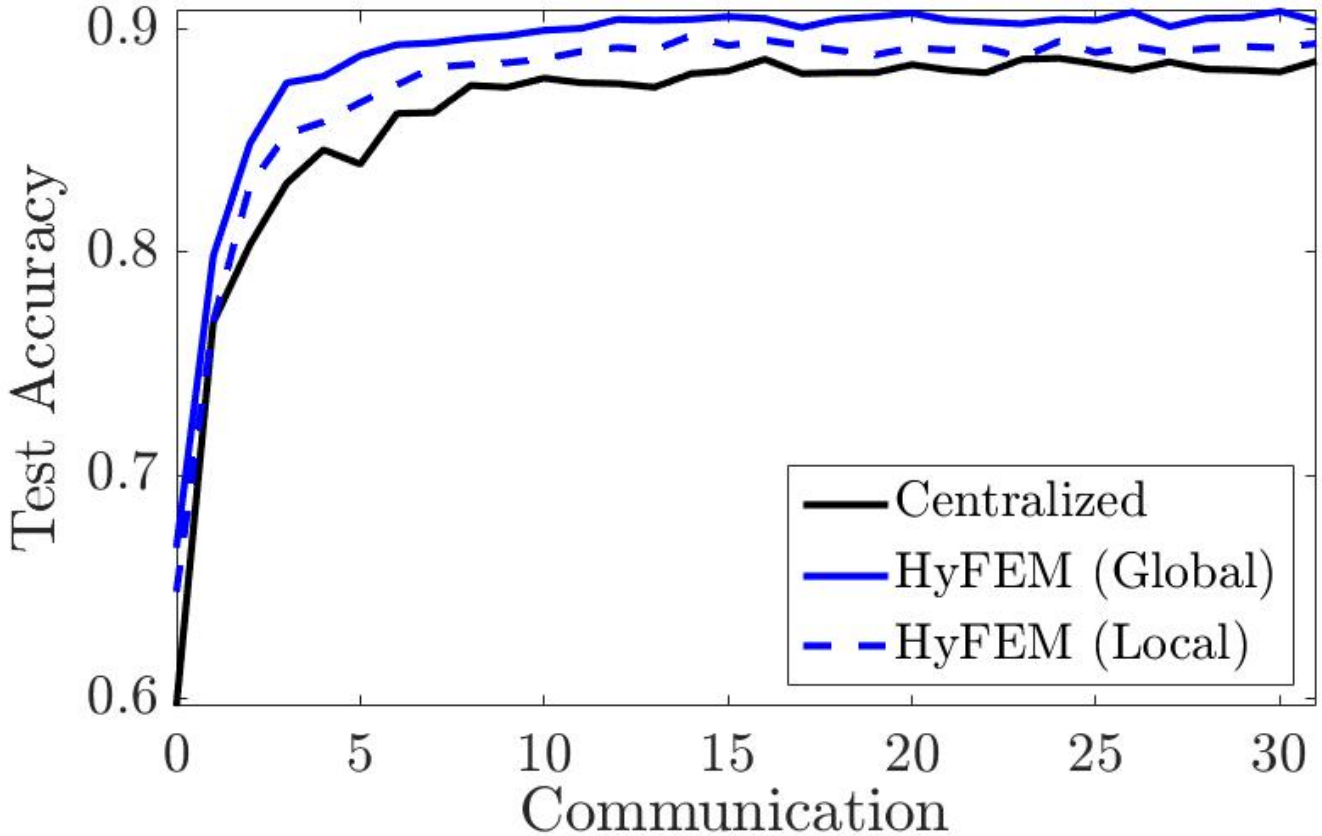
❑ Numerical Result 1 (4 clients, 2 views/client)



Good global model performance

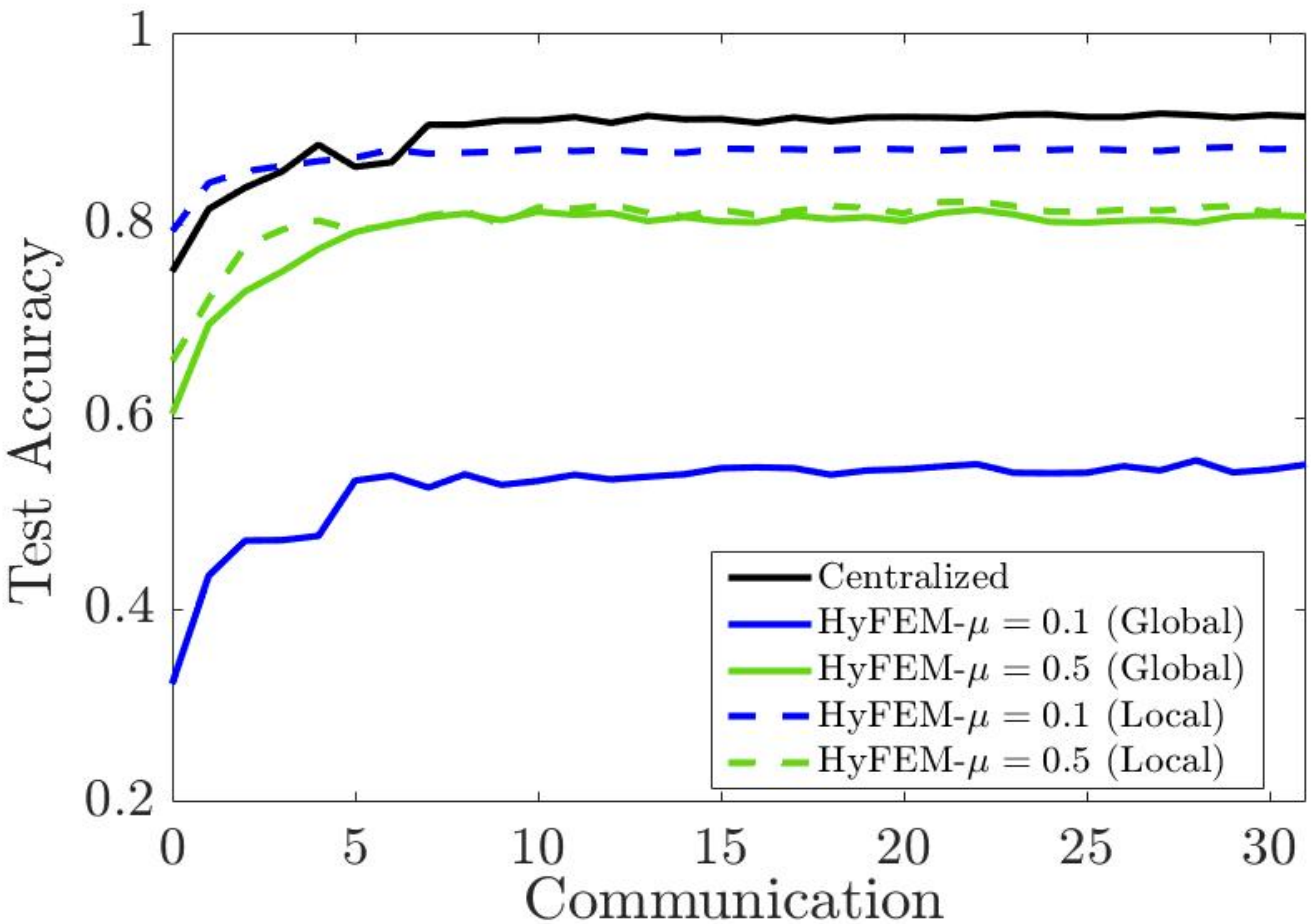
Local feature NOT enough for classification

❑ Numerical Result 2 (4 clients, 3 views/client)



Hybrid FL outperforms centralized training

❑ Numerical Result 3 (8 clients, 6 views/client)



Good local model performance even when client only has half features and 15 classes

Balance between local and global performance by changing μ

❑ Conclusion

1. Local models avoid data sharing and sample synchronization.
2. Model splitting makes use of the shared features.
3. Regularizing the classifiers enables knowledge sharing.
4. Hybrid FL can learn a good global model with incomplete data where HFL and VFL do not apply.
5. Offer an alternative for VFL