# FuseFL: One-Shot Federated Learning through the Lens of Causality with Progressive Model Fusion

Zhenheng Tang Yonggang Zhang Peijie Dong Yiu-ming Cheung
Amelie Chi Zhou Bo Han Xiaowen Chu

NEURAL INFORMATION PROCESSING SYSTEMS

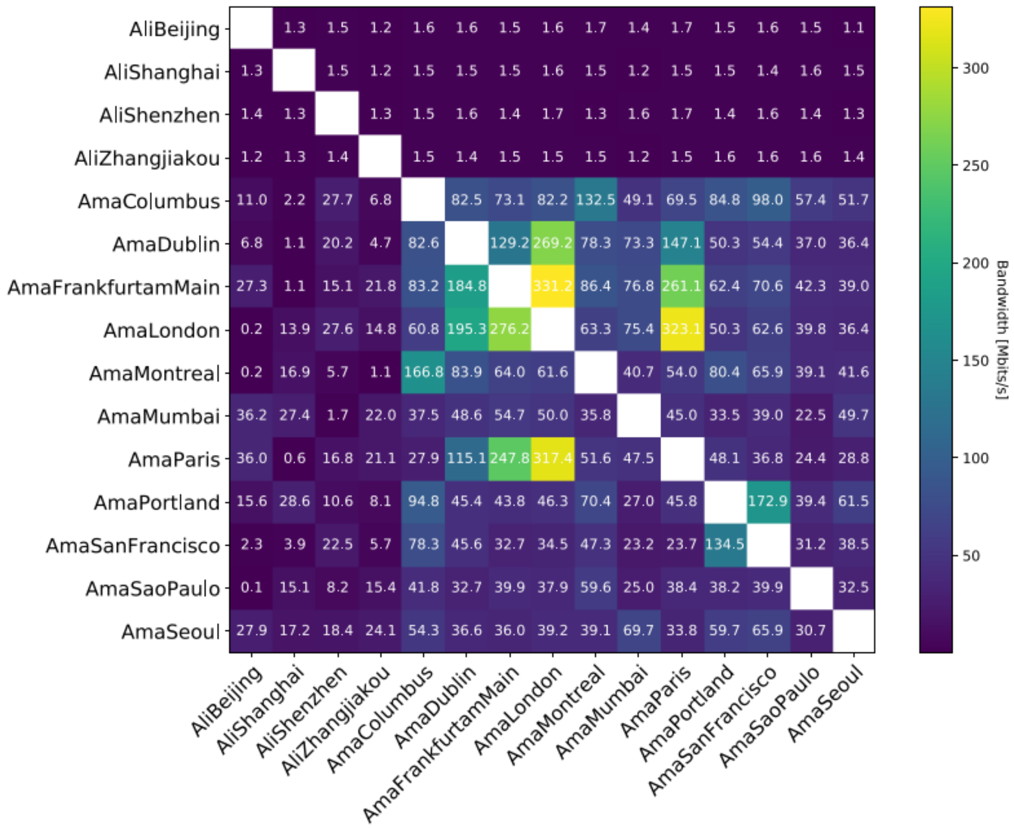
香港浸會大學
HONG KONG BAPTIST UNIVERSITY

香港科技大學(廣州)
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

# Background – FedAVG

## Federated Learning

Low-bandwidth communication between parties




Bandwidth distribution between cities [1]

When training a GPT-3 of **100 GB** size, communicating time of one round in distributed SGD, will be

*100GB/10MB/s = 10000 seconds =* ***2.8 hours!***

If we communicate for 100000 rounds to guarantee convergence. We need

***2.8 hours × 100000 = 280000 hours = 32 years!***

[1] GossipFL: A Decentralized Sparsified and Adaptive Communication. In TPDS 2022. Federated Learning Framework With

# Background – FedAVG

## Federated Learning -- FedAVG

Reducing communication rounds by local training

**Algorithm 1** FederatedAveraging. The $K$ clients are indexed by $k$; $B$ is the local minibatch size, $E$ is the number of local epochs, and $\eta$ is the learning rate.

**Server executes:**
- initialize $w_0$
- **for** each round $t = 1, 2, \dots$ **do**
  - $m \leftarrow \max(C \cdot K, 1)$
  - $S_t \leftarrow$ (random set of $m$ clients)
  - **for** each client $k \in S_t$ **in parallel do**
    - $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
  - $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

**ClientUpdate($k, w$):** *// Run on client $k$*
- $\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size $B$)
- **for** each local epoch $i$ from 1 to $E$ **do**
  - **for** batch $b \in \mathcal{B}$ **do**
    - $w \leftarrow w - \eta \nabla \ell(w; b)$
- return $w$ to server

Do local training for 100 iterations before communication.

***2.8 hours × 100000 = 280000 hours = 32 years!***

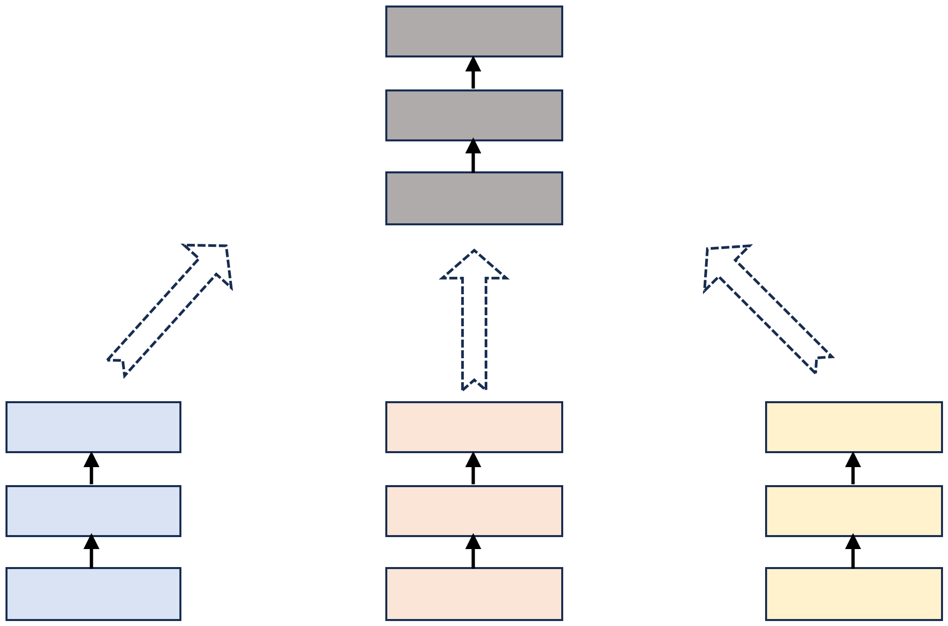
***2.8 hours × 1000 = 2800 hours = 117 days!***

It is still too long.

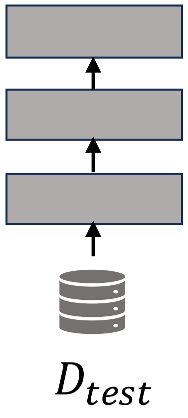
# One-shot FL

## One-shot Federated Learning (OFL)

*How to improve FL performance under* ***extremely low*** *communication costs with almost no extra computational and storage costs?*



Averaging local models once

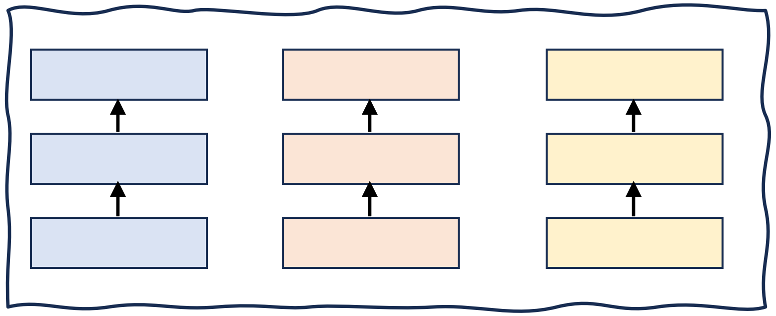



Testing

## One-shot Federated Learning (OFL)

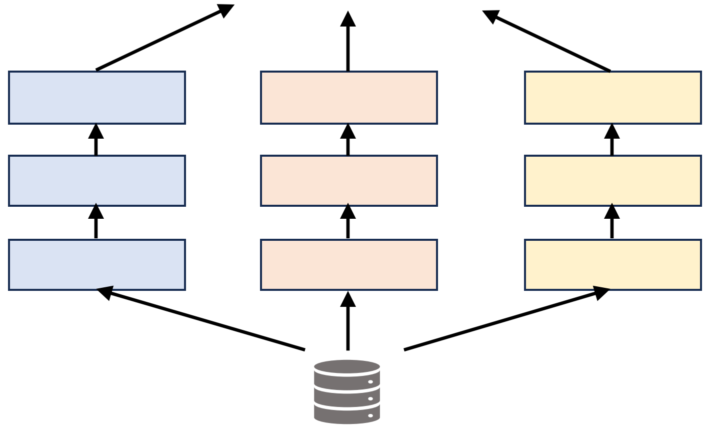
*How to improve FL performance under* ***extremely low*** *communication costs with almost no extra computational and storage costs?*

Ensemble



Collecting all local models together

Vote or average as prediction



Testing

## One-shot Federated Learning (OFL)

*Low performance of directly averaging*

| Dataset | CIFAR-10 | | SVHN | | CIFAR-100 | | Tiny-Imagenet | |
|---|---|---|---|---|---|---|---|---|
| Heterogeneity | a=0.1 | a=0.5 | a=0.1 | a=0.5 | a=0.1 | a=0.5 | a=0.1 | a=0.5 |
| FedAvg (OFL) | 23.93 | 43.67 | 31.65 | 56.09 | 4.58 | 12.11 | 3.12 | 11.89 |
| Ensemble | **57.5** | **79.91** | **65.29** | **85.7** | **35.69** | **53.39** | **30.85** | **45.8** |

# Understanding OFL -- *Data heterogeneity*

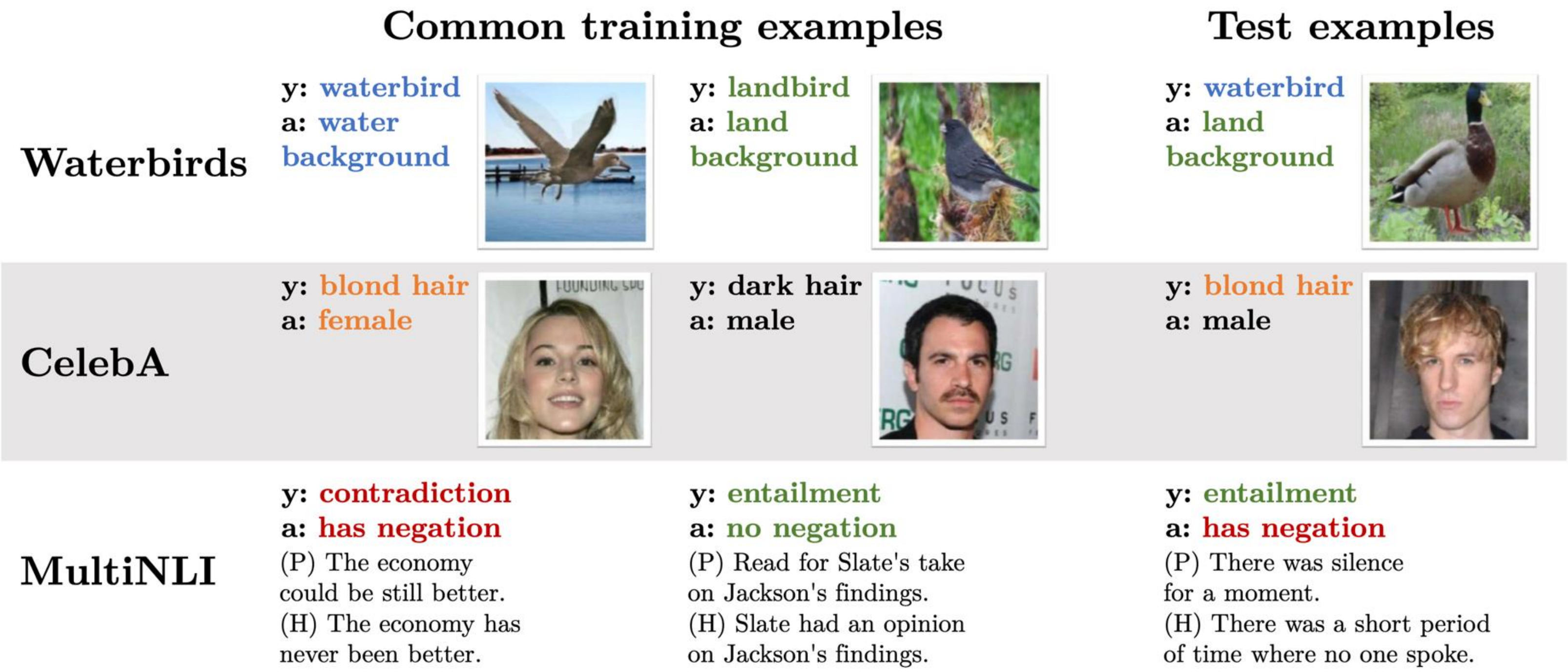
***WHY Low performance*** *of directly averaging?*

(d) Images and landmarks from 5 authors.


***Data heterogeneity of FL*** [1]

Each client has its own datasets **without sharing.** Datasets between clients have **different** data distribution, called Non-Independent and Identically distributed (**Non-I.I.D.**) data. i.e. data heterogeneity.

# Understanding OFL -- *Data heterogeneity*

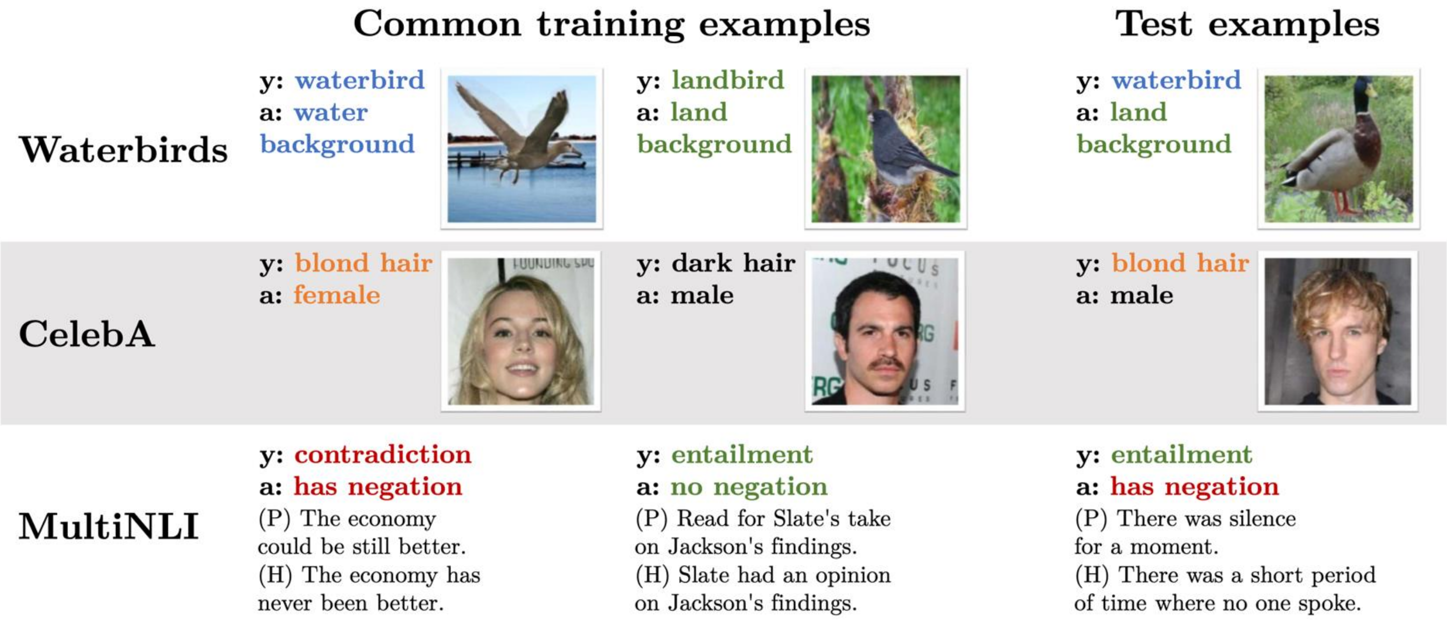


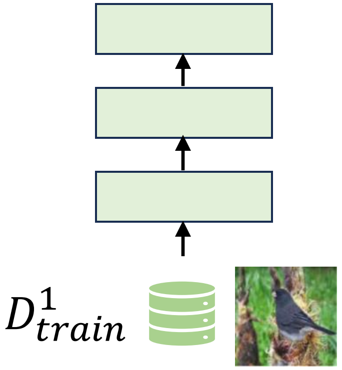

Examples of dataset bias [1,2]

[1] Distributionally Robust Neural Networks for Group Shifts. In ICLR 2020.
[2] Shortcut learning in deep neural networks. In Nature Machine Intelligence 2020.

# Understanding OFL -- Spurious Fitting

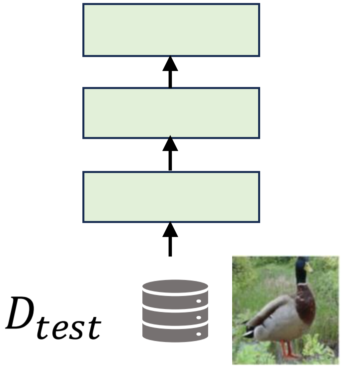
## Fitting on **spurious features** during local training




Input: Land + LandBird
Prediction: WaterBird (0.01)
**LandBird (0.99)**

**Predict based on both Land and Landbirds features**



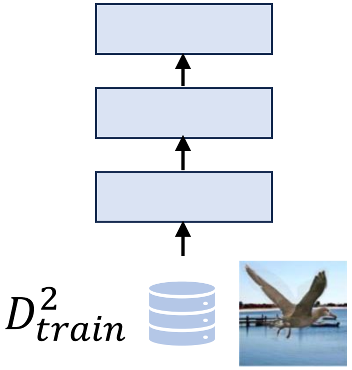

Input: land + WaterBird
Prediction: WaterBird (0.1)
**LandBird (0.9)**

**Predict based mainly on Land features (wrong)**

[1] Distributionally Robust Neural Networks for Group Shifts. In ICLR 2020.
[2] Shortcut learning in deep neural networks. In Nature Machine Intelligence 2020.

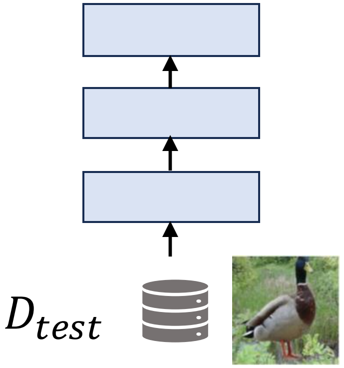
# Understanding OFL -- Spurious Fitting

## Fitting on **spurious features** during local training


Input: Water + WaterBird
Prediction: **WaterBird (0.99)**
LandBird (0.01)

**Predict based on both water and waterbirds features**



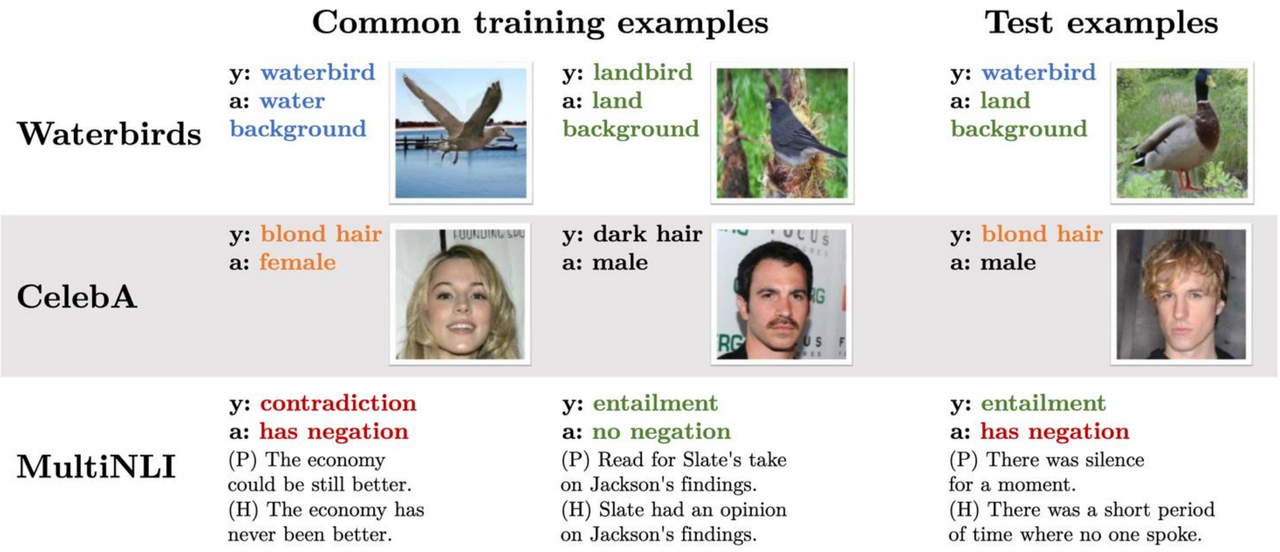

Input: land + WaterBird
Prediction: **WaterBird (0.9)**
LandBird (0.1)

**Predict based mainly on waterbirds features (correct)**

[1] Distributionally Robust Neural Networks for Group Shifts. In ICLR 2020.
[2] Shortcut learning in deep neural networks. In Nature Machine Intelligence 2020.

# Understanding OFL – A Causal View

## Modeling invariant and spurious features in FL datasets



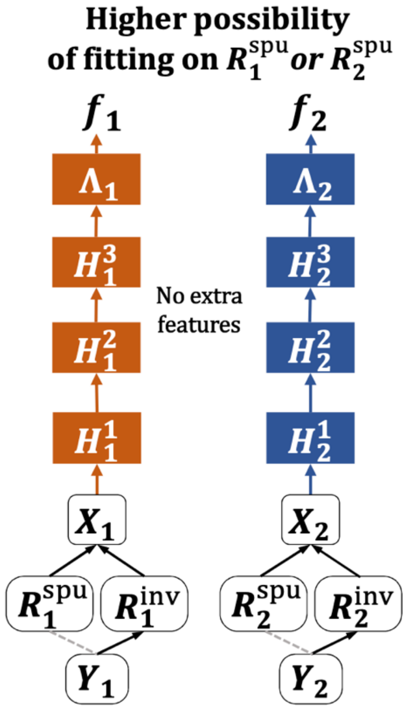

$D^1_{train}$

| | |
|---|---|
| $X_1$: Water + WaterBird | $R_1^{spu}$: Water |
| $Y_1$: WaterBird | $R_1^{inv}$: WaterBird |

$D^2_{train}$

| | |
|---|---|
| $X_2$: Land + LandBird | $R_2^{spu}$: Land |
| $Y_2$: LandBird | $R_2^{inv}$: LandBird |

$H_i^j$: Neural modules or features



**(a) Isolated Training & Ensemble**

**Structure Equation Model [1] of FL**

[1] J. Pearl. Causality. Cambridge university press, 2009.

# Understanding OFL – A Causal View

## Enhancing model training with more **features** from other clients

$D_{train}^{1}$

$X_1$: Water + WaterBird  $R_1^{\mathrm{spu}}$: Water
$Y_1$: WaterBird  $R_1^{\mathrm{inv}}$: WaterBird
$H_1$: Water, WaterBird

$D_{train}^{2}$

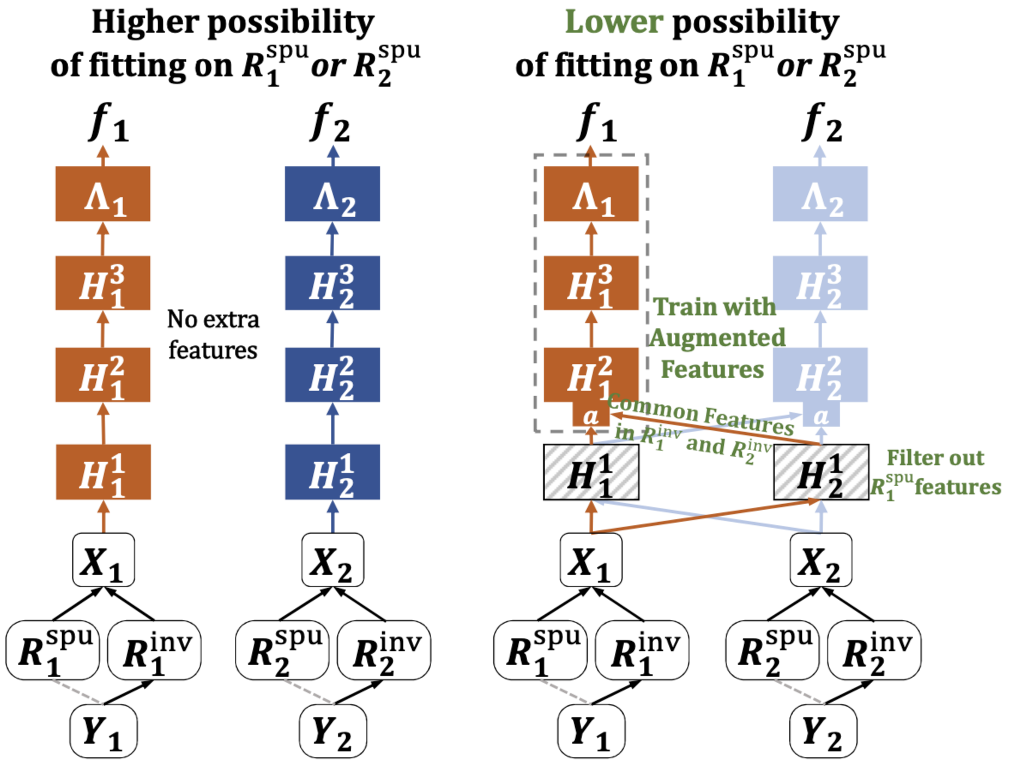
$X_2$: Land + LandBird  $R_2^{\mathrm{spu}}$: Land
$Y_2$: LandBird  $R_2^{\mathrm{inv}}$: LandBird
$H_2$: Land, LandBird

$D_{train}^{3}$

$X_3$: Land + WaterBird  $R_2^{\mathrm{spu}}$: Land
$Y_3$: WaterBird  $R_2^{\mathrm{inv}}$: WaterBird
$H_3$: Land, WaterBird

$H_1$ may easily fit on Water instead of WaterBird and other common features of birds.

$H_1 + H_2 + H_3$ have more features about birds, thus having more opportunities to predict birds based on features of birds.




(a) Isolated Training & Ensemble

(b) Federated Fusion

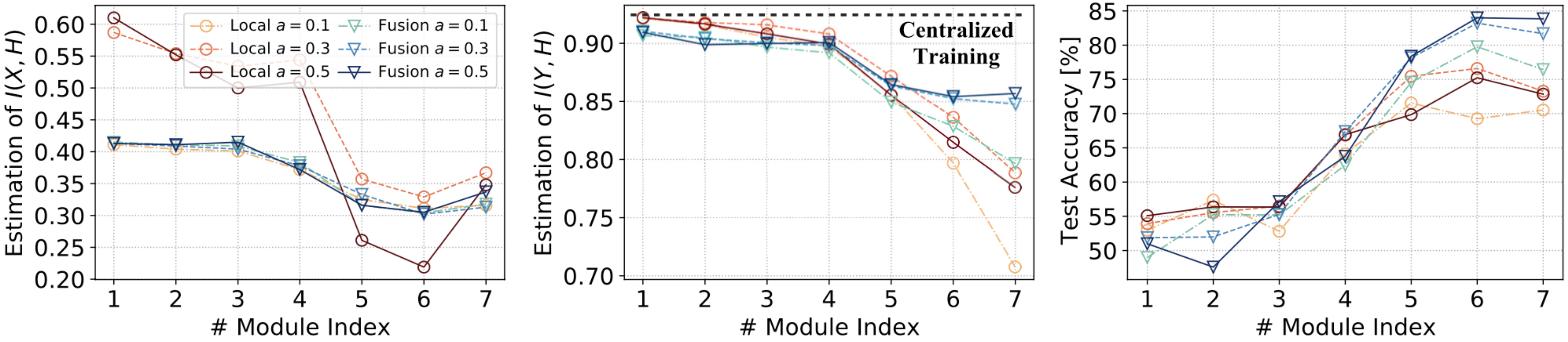

[1] Understanding and improving feature learning for out-of-distribution generalization. In NeurIPS 2023.
[2] Can subnetwork structure be the key to out-of-distribution generalization? In ICML 2021.

# Understanding OFL – Mutual Information

***Insights from information bottleneck*** [1]

$$\textbf{Sufficient statistic}: I(X;Y) = I(H(X);Y),$$
$$\textbf{Minimal statistic}: H(X) = \arg\min_{\tilde{H}(X)} I(\tilde{H}(X);X).$$
$$I(H(X);R^{spu}) \leq I(H(X);X) - I(X;Y).$$

Better $H$ means [2]: larger $I(H;Y)$
smaller $I(H;X)$




(a) Estimated MI $I(H^k;X)$. (b) Estimated MI $I(H^k;Y)$. (c) The separability of layers.

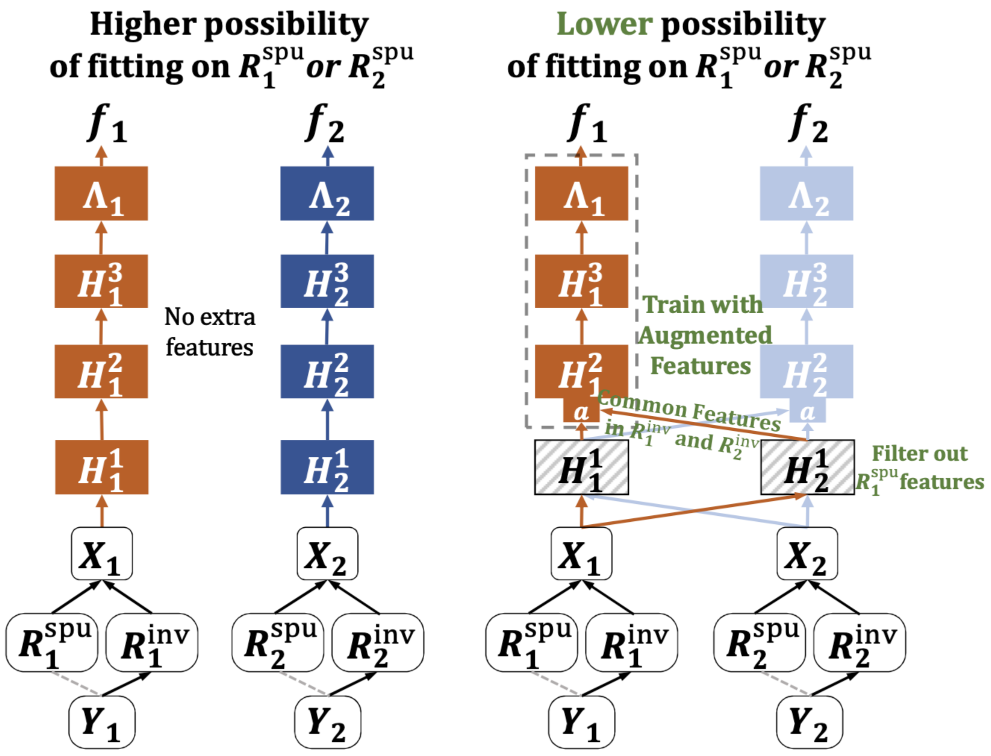
Figure 2: Estimated MI and separability of trained models with non-IID datasets.


[1] Opening the black box of deep neural networks via information. Arxiv 2017.
[2] Emergence of invariance and disentanglement in deep representations. In JMLR 2018.

# FuseFL: Progressive FL Model Fusion

Design goals:

1. Keeping communication costs as same as one-shot FL.
2. Sharing feature extractors across all clients to enhance later model training.
3. Avoiding extra computation costs.
4. Avoiding extra storage costs.

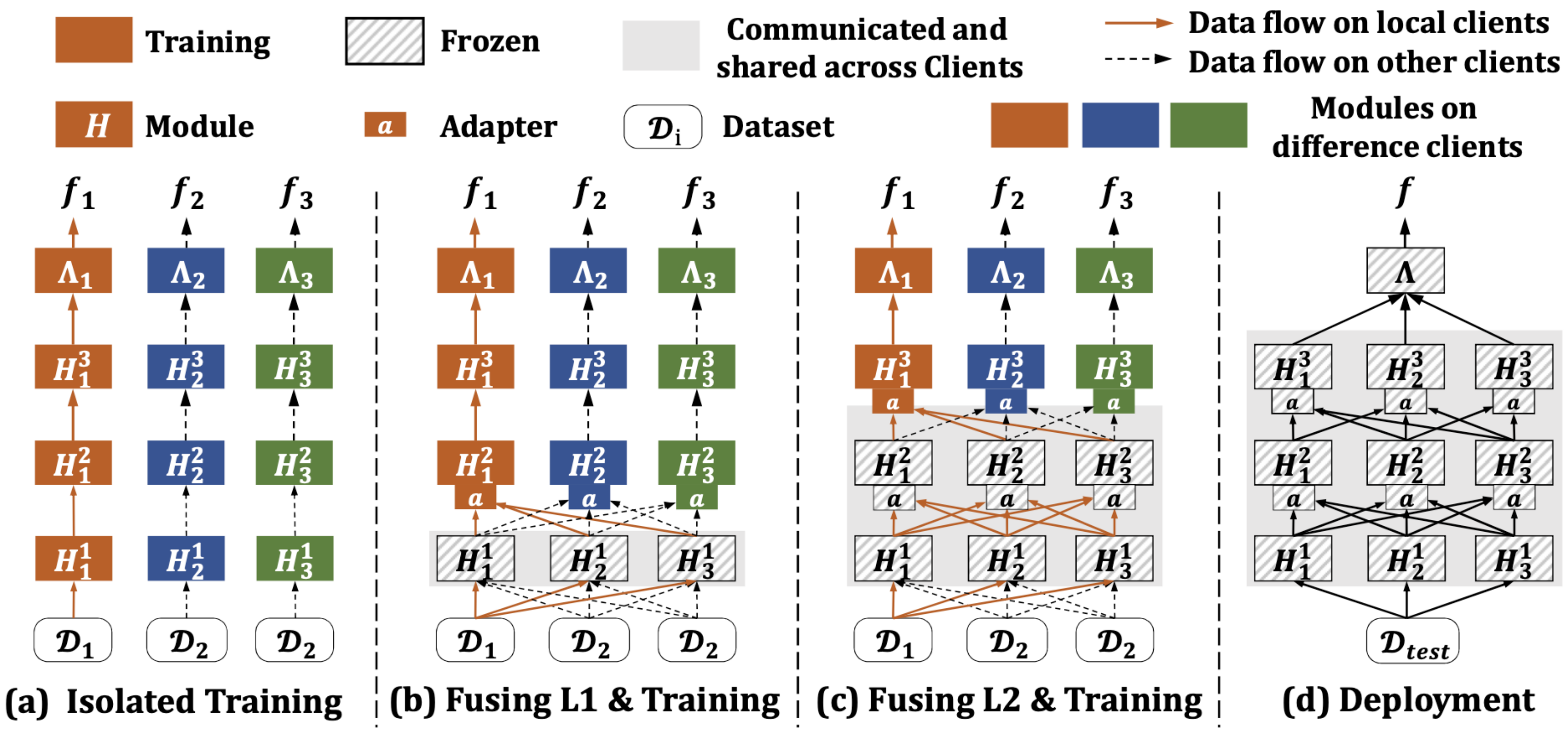



(a) Isolated Training & Ensemble

(b) Federated Fusion

# FuseFL: Progressive FL Model Fusion

Design goals:
1. Keeping communication costs as same as one-shot FL.
2. Sharing feature extractors across all clients to enhance later model training.
3. Avoiding extra computation costs.
4. Avoiding extra storage costs.



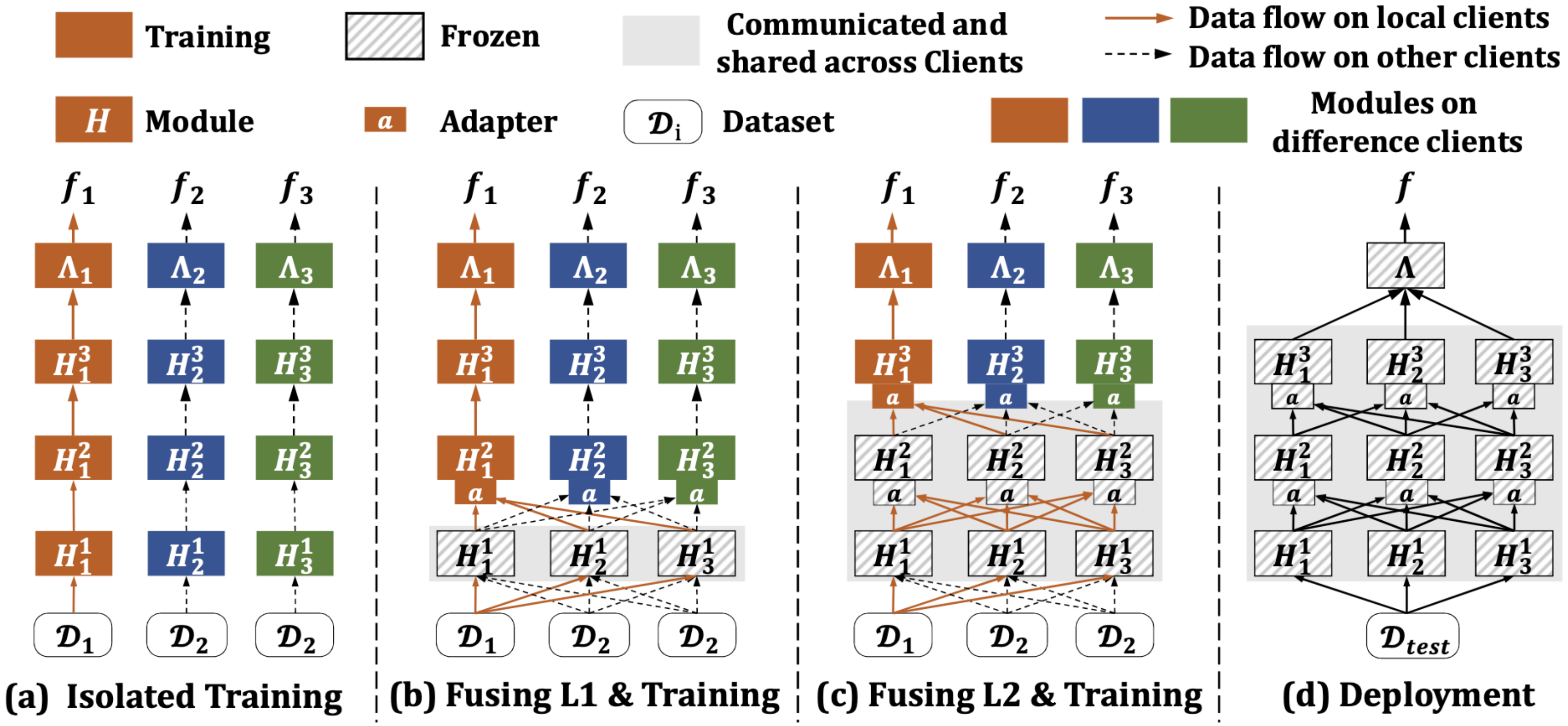

Training Procedures of FuseFL:
For $i$-th block in all blocks:
(a) Local (Isolated) training $[i:]$ blocks;
(b) Then, communicating all $i$-th blocks of all clients. Clients concatenate these blocks as a new concated block. Then, clients append a new adapter before the next $i$+1-th block. All blocks $[:i]$ are frozen.
Finally, freeze all modules and calibrate the classifier.

Deployment of FuseFL (inference stage): (d) the test data passes through all merged modules and adapters.

# FuseFL: Progressive FL Model Fusion

Design goals:

1. Keeping communication costs as same as one-shot FL.
2. Sharing feature extractors across all clients.
3. Avoiding extra computation costs.
4. Avoiding extra storage costs.




Benefits of FuseFL:

1. Local modules as feature extractors are used across all clients during local training, mitigating the spurious fitting problem;

2. The total communication costs are as same as OFL;

3. We shrink the local module size as the local dataset is smaller, not requiring the original large module to learn;

4. We reduce the local training epochs to avoid extra computation costs.

5. The local modules can be heterogeneous.

6. There is no extra privacy risks than FedAvg.

# Experiment Results

Default Exp configuration:

5 clients.

ResNet-18 for all clients.

Table 2: Accuracy of different methods across $\alpha = \{0.1, 0.3, 0.5\}$ on different datasets. Ensemble means ensemble learning with local trained models, which is an upper bound of all previous methods but impractical in FL due to the large memory costs and the weak scalability of clients. Thus, we highlight the best results in **bold font** except Ensemble.

| Dataset | MNIST | | | FMNIST | | | CIFAR-10 | | | SVHN | | | CIFAR-100 | | | Tiny-Imagenet | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | $\alpha$=0.1 | $\alpha$=0.3 | $\alpha$=0.5 | $\alpha$=0.1 | $\alpha$=0.3 | $\alpha$=0.5 | $\alpha$=0.1 | $\alpha$=0.3 | $\alpha$=0.5 | $\alpha$=0.1 | $\alpha$=0.3 | $\alpha$=0.5 | $\alpha$=0.1 | $\alpha$=0.3 | $\alpha$=0.5 | $\alpha$=0.1 | $\alpha$=0.3 | $\alpha$=0.5 |
| FedAvg | 48.24 | 72.94 | 90.55 | 41.69 | 82.96 | 83.72 | 23.93 | 27.72 | 43.67 | 31.65 | 61.51 | 56.09 | 4.58 | 11.61 | 12.11 | 3.12 | 10.46 | 11.89 |
| FedDF | 60.15 | 74.01 | 92.18 | 43.58 | 80.67 | 84.67 | 40.58 | 46.78 | 53.56 | 49.13 | 73.34 | 73.98 | 28.17 | 30.28 | 36.35 | 15.34 | 18.22 | 27.43 |
| Fed-DAFL | 64.38 | 74.18 | 93.01 | 47.14 | 80.59 | 84.02 | 47.34 | 53.89 | 58.59 | 53.23 | 76.56 | 78.03 | 28.89 | 34.89 | 38.19 | 18.38 | 22.18 | 28.22 |
| Fed-ADI | 64.13 | 75.03 | 93.49 | 48.49 | 81.15 | 84.19 | 48.59 | 54.68 | 59.34 | 53.45 | 77.45 | 78.85 | 30.13 | 35.18 | 40.28 | 19.59 | 25.34 | 30.21 |
| DENSE | 66.61 | 76.48 | 95.82 | 50.29 | 83.96 | 85.94 | 50.26 | 59.76 | 62.19 | 55.34 | 79.59 | 80.03 | 32.03 | 37.32 | 42.07 | 22.44 | 28.14 | 32.34 |
| Ensemble | 86.81 | 96.76 | 97.22 | 67.71 | 87.25 | 89.42 | 57.5 | 77.35 | 79.91 | 65.29 | 88.31 | 85.7 | 35.69 | 49.41 | 53.39 | 30.85 | 39.43 | 45.8 |
| FuseFL $K=2$ | 97.02 | **98.43** | **98.54** | 83.15 | **89.94** | 89.47 | 70.85 | 81.41 | **84.34** | 76.88 | **91.07** | **90.87** | 34.07 | **45.12** | 46.12 | **29.28** | 31.11 | **34.34** |
| FuseFL $K=4$ | **97.19** | 98.34 | 98.29 | 83.05 | 84.58 | **90.50** | **73.79** | **84.58** | 81.15 | 78.08 | 89.63 | 89.34 | **36.86** | 42.79 | **49.30** | 27.63 | **33.04** | 34.28 |
| FuseFL $K=8$ | 96.66 | 98.35 | 98.16 | **83.2** | 88.57 | 88.24 | 70.46 | 80.70 | 74.99 | **80.31** | 88.88 | 89.94 | 34.97 | 39.08 | 40.73 | 25.21 | 32.59 | 33.82 |

# Experiment Results

**Support of heterogeneous models.**
2 clients: ResNet10
2 clients: ResNet26
1 client: ResNet18

Avg: averaging concatenated features.
Conv1x1: passes features through conv layer.

Table 3: Accuracy with FuseFL with conv1×1 or averaging to support heterogeneous model design on CIFAR-10.

| non-IID degree | $a = 0.1$ | $a = 0.3$ | $a = 0.5$ |
|---|---|---|---|
| Ensemble | 57.5 | 77.35 | 79.91 |
| FuseFL | **73.79** | **84.58** | 81.15 |
| FuseFL (Avg) | 68.08 | 71.49 | 80.35 |
| FuseFL-Hetero | 75.33 | 81.71 | **82.71** |
| FuseFL (Avg)-Hetero | 68.31 | 76.27 | 79.74 |

# Thanks for your time!

Q & A