HPFL: FEDERATED LEARNING BY FUSING MULTIPLE SENSOR MODALITIES WITH HETEROGENEOUS PRIVACY SENSITIVITY LEVELS

Yuan-Jie Chen
Advisor: Cheng-Hsin Hsu

Outlines

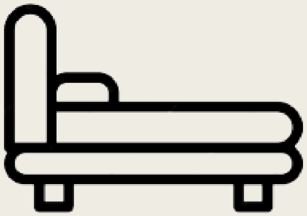
- Introduction
- Related Work
- Heterogeneous Privacy Federated Learning
- Multi-Modal Representation Learning
- Experiment Setup
- Experiment Results
- Conclusion and Future Work

INTRODUCTION

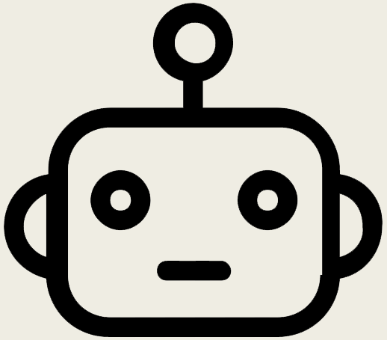
Multi-modal Sensing

- A variety of professional sensors are gradually attracting attention
  - *Depth cameras*
  - *Thermal cameras*
  - *mmWave radars*
  - *LiDARs*
- Provides multi dimensions information than single modal sensing

Self-driving cars



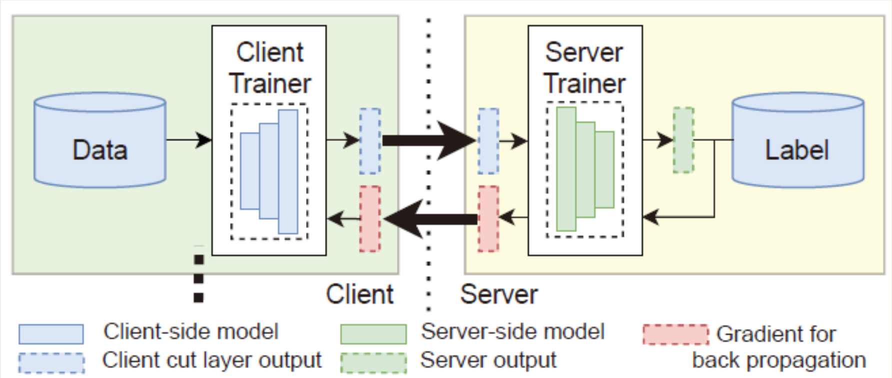
Health cares



Robot systems

Smart agricultures

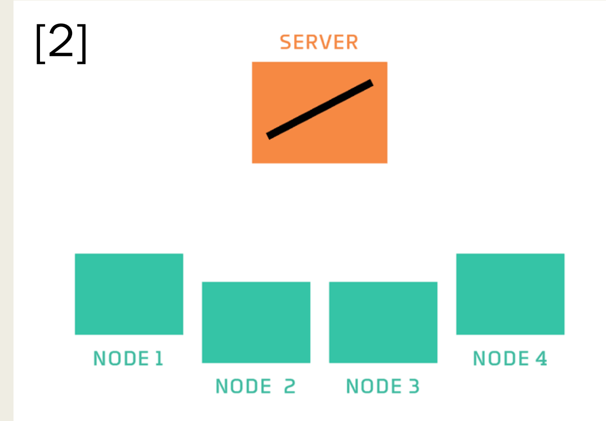
Privacy Concern



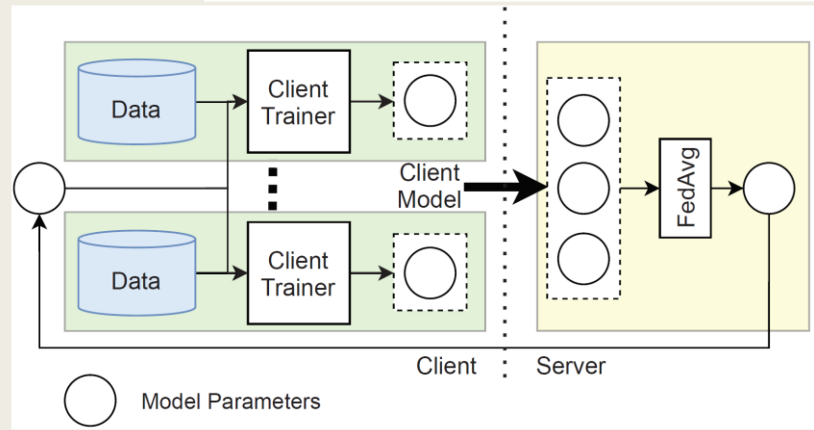
- Collecting data from widely used RGB cameras incurs privacy risk
  - *Encryption methods*
  - *Distributed Cooperative Machine Learning (DCML) [1]*
- Homomorphic encryption and differential privacy
  - *Advanced privacy protection techniques*
  - *High computation and communication overhead*
- Split Learning (SL [2]) and Federated Learning (FL [3])
  - *SL and FL provide source data protection DCML*
  - *SL is slower than FL*
  - *SL usually works under organizations (data providers, computing resource providers)*

[1] Scaling up machine learning: Parallel and distributed approaches. Cambridge University Press, 2011.
[2] Vepakomma P, Gupta O, Swedish T, et al. Split learning for health: Distributed deep learning without sharing raw patient data[J]. arXiv preprint arXiv:1812.00564, 2018.
[3] McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data[C]//Artificial intelligence and statistics. PMLR, 2017: 1273-1282.

Federated Learning



- Federated learning [1] workflow (FedAvg)
  - *Distribute server model*
  - *Client training*
  - *Upload client model*
  - *Aggregation*



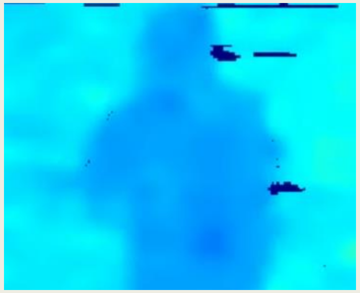
- Advantage: Protects client's privacy and reduces communication cost
- Disadvantage: low model performance and convergence speed caused by data incompleteness
  - *Non-independent-identically distributed (non-i.i.d.): concept drift/shift, covariate shift*

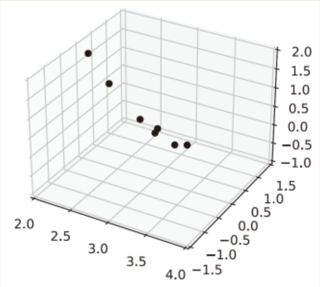
[1] McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data[C]//Artificial intelligence and statistics. PMLR, 2017: 1273-1282.
[2] http://vision.cloudera.com/wp-content/uploads/2018/11/2018-10-31-181344-federated_learning_animated_labeled.gif

Motivation

- Multiple sensors generate data that has diverse degrees of privacy concerns
  - *RGB images → privacy-sensitive*
  - *Depth images, mmWave point clouds → privacy-insensitive*
- The root cause of the performance degradation of FL models: data incompleteness
  - *Scattered data*
  - Non-i.i.d. data
- The utilization of privacy-insensitive data has never been considered


RGB image


Depth image



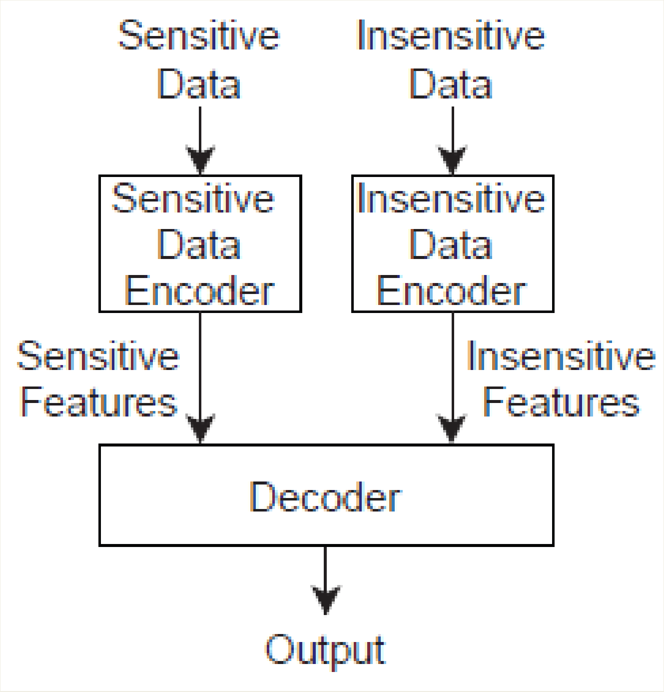
mmWave point cloud

Problem Statement

- Target
  - *Utilize the privacy-insensitive data to improve FL model performance*
  - *Reduce the impact of non-i.i.d. data on the model*
- Condition
  - *Each clients has a multi-modal dataset with heterogeneous degrees of privacy levels*
  - *No obviously privacy risk*
  - *Lower communication and computation overhead*
- Solution
  - *Request all clients upload privacy-insensitive data to the server*
  - *Add an additional model fine-tuning at the server*

Challenges

- Utilization of insensitive data for improving model performance
  - *No similar work in the literature*
  - *Multi-modal models require multi type data as inputs*
  - *Privacy-sensitive data are not available at the server*
- Training with single insensitive data can bias the models



An example multi-modal model

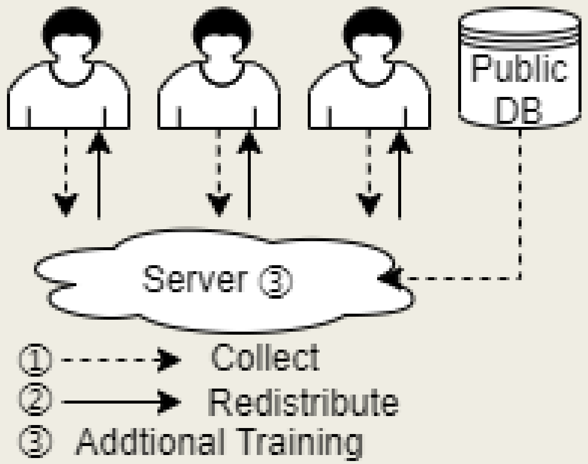
Contributions

- We are the first group who considered multi-sensor (or multi-modal) classification problems, in which sensor data have diverse privacy sensitivity levels
- We apply HPFL on a semantic segmentation network, an emotion recognition network, and get 18.2% improvement in foreground accuracy and 4.2% in F1-score, compared to FedAvg
- HPFL outperforms state-of-the-art advanced FL optimization algorithms, FedProx, FedAdam, FedDyn, FedCon, 12.4%-17.7% improvement in foreground accuracy and 2.54%-4.1% in F1-scores

RELATED WORK

Data Sharing

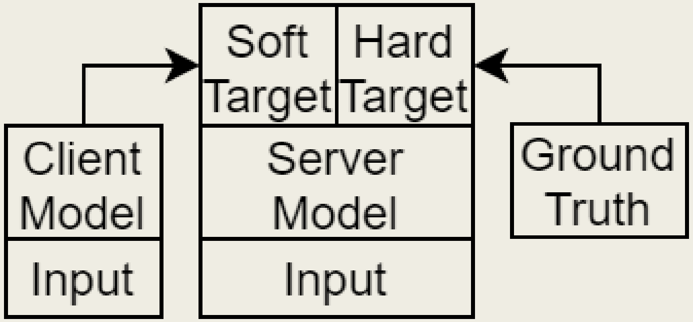
- To overcome the inference of non-iid data distribution
- Redistribute the data that collect from clients or public dataset to balance local data (①, ②)
- Employ the collected data to carry out additional training after the client training (①, ③)
- Summary
  - *Not realize the heterogeneous privacy sensitivity levels data*
  - *Still have privacy concern*
  - *Unable to confront strong non-iid degree, performance improvement is small (2%~5%)*

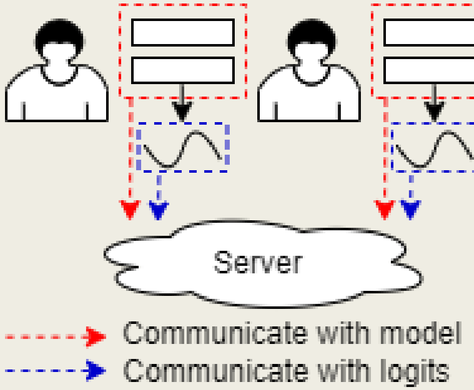


Distillation and Federated Distillation

- Distillation was initially proposed to compress neural networks

- Federated distillation was proposed for two targets:
  - *Trade model accuracy for lower communication cost*
  - *Server-side distillation for better server model performance and compatible heterogeneous client model structure*

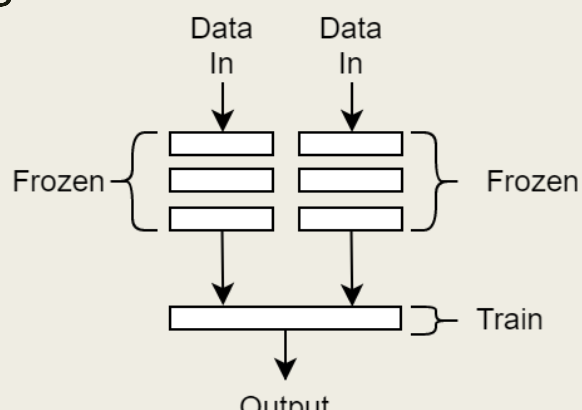




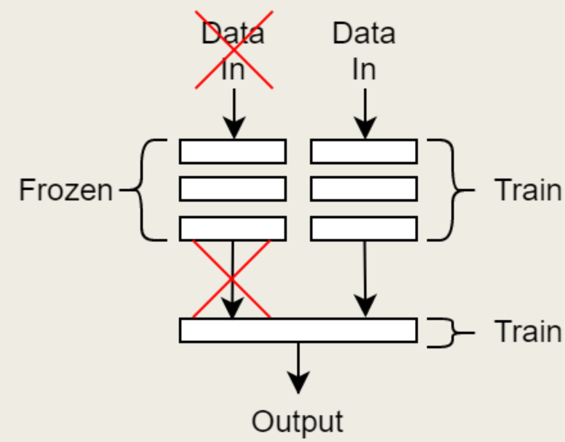
- We only collect insensitive data to server, so, we cannot perform the offline distillation at server

Federated Transfer Learning

- Transfer learning focuses on transferring a domain knowledge to a different but similar domain
- FTL belongs to personalization FL
  - *The server model may not adapt to every participating client*
- FTL freezes the parameters related to high-level features, but HPFL freezes the parameters related to sensitive data
- FTL focuses on optimizing client model but HPFL focuses on optimizing server model



FTL



HPFL

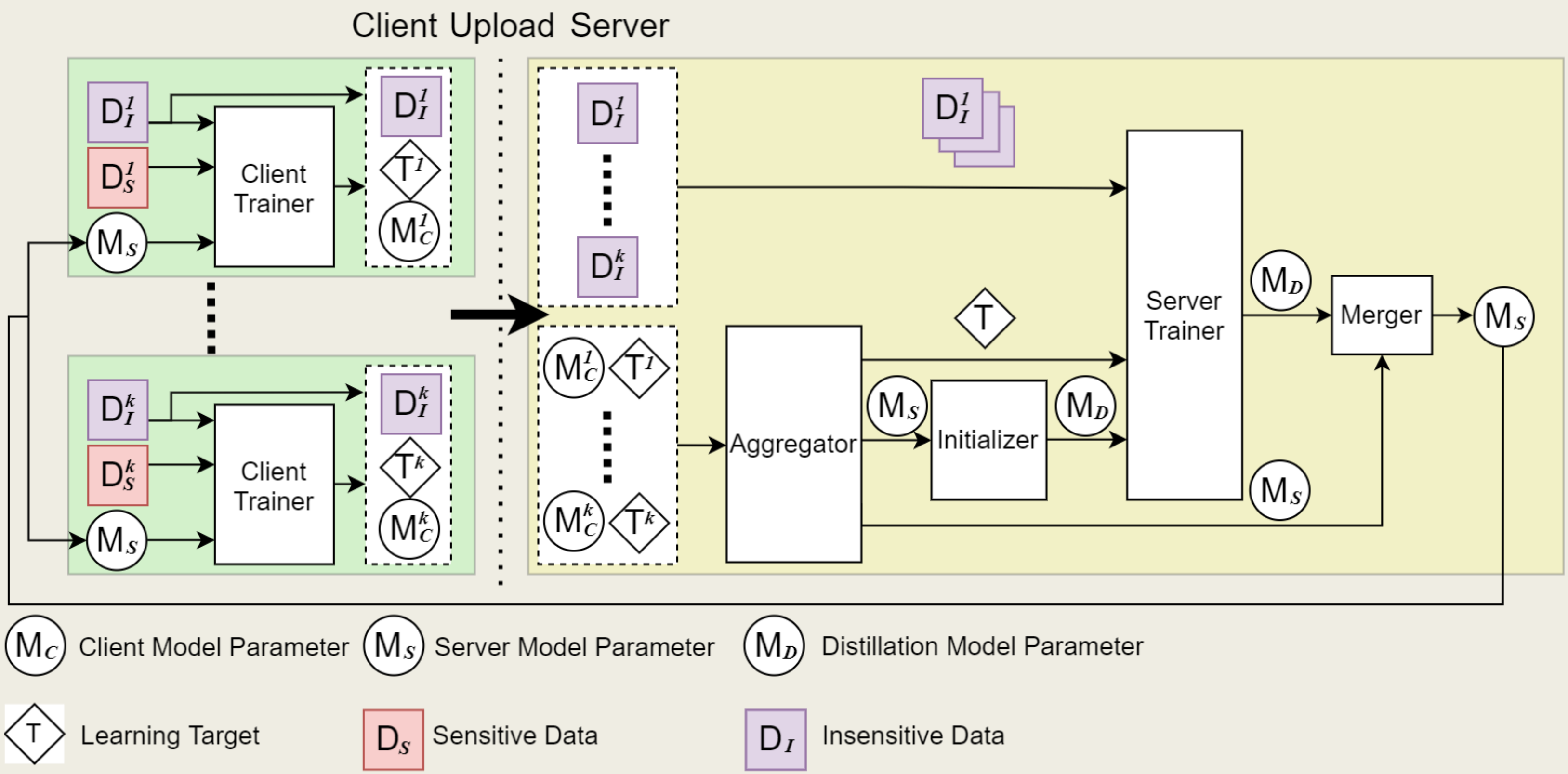
Advanced FL Algorithms

- Common target: improve server model performance in FL
- Optimize client trainer:
  - *FedProx [1]: add a L2-regularize term on client trainer → reduce the distance between client and server model*
  - *FedDyn [2]: consider history updates and distance to server model → smooth updating*
  - *FedCon [3]: consider the feature learned by client model and server model need to be similar → fast converge*
- Optimize server aggregator:
  - *FedAdam [4]: use Adam optimizer to replace average aggregator in FedAvg → fast converge*

[1] Sahu A K, Li T, Sanjabi M, et al. On the convergence of federated optimization in heterogeneous networks[J]. arXiv preprint arXiv:1812.06127, 2018, 3: 3.
[2] Acar D A E, Zhao Y, Navarro R M, et al. Federated learning based on dynamic regularization[J]. arXiv preprint arXiv:2111.04263, 2021.
[3] Li Q, He B, Song D. Model-contrastive federated learning[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 10713-10722. [4] Reddi S, Charles Z, Zaheer M, et al. Adaptive federated optimization[J]. arXiv preprint arXiv:2003.00295, 2020.

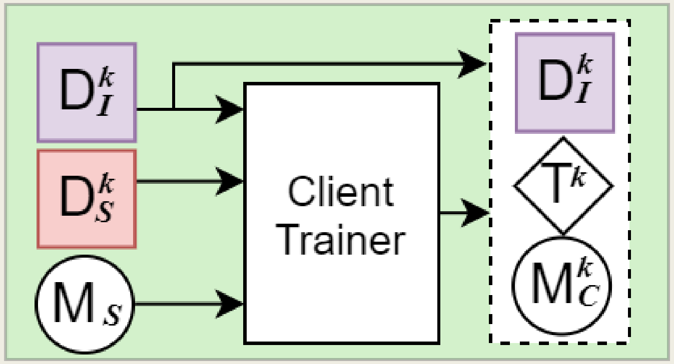
HETEROGENEOUS PRIVACY FEDERATED LEARNING

HPFL Workflow



Client model: used by the clients to train with locally collected sensor data
Server model: aggregated from models sent by all clients
Distillation model: trained by insensitive data from all clients at the server

Client Side Design



- Target
  - *Better utilize sensor data with different privacy levels in federated learning*

- Methodology (for each client k in K)

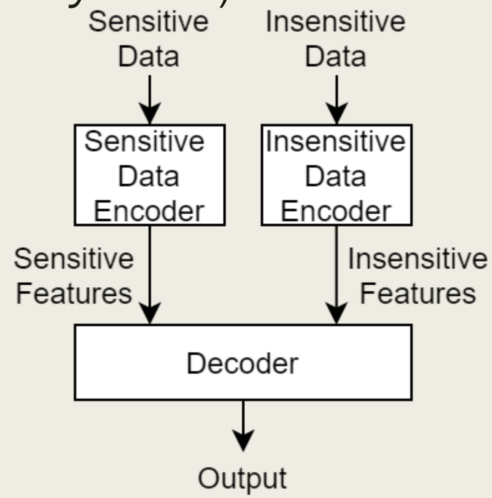
① – *Classify the local sensor data (D^k) into sensitive (S) and insensitive (I) ones ($D^k = D_S^k + D_I^k$)*

② – *Upload the insensitive data (D_I^k) to the server (only once)*

③ – *Train a client model (M_C^k) with all locally available sensor data*

$$M_{C,t+1}^k = \underset{M_{C,t}^k}{argmin}\, L_k(D^k, Y^k \,|\, M_{C,t}^k)$$

④ – *Upload trained model and learning target (T^k)*

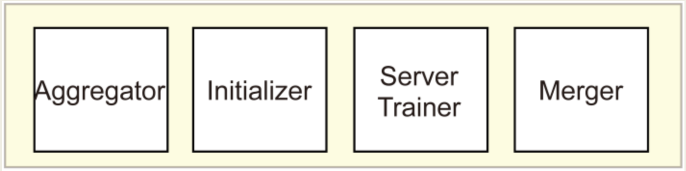


General client model

Sensitive data (RGB)

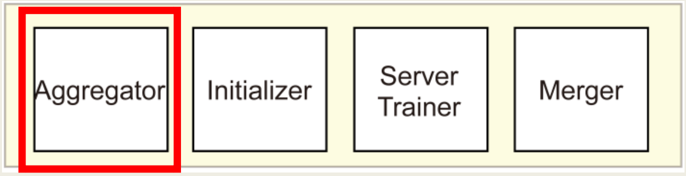
Insensitive data (Thermal)

Server Side Design



- Better utilize the insensitive data to reduce the negative impacts caused by non-i.i.d. sample distribution
- The complete model needs sensitive and insensitive data as input, but no sensitive data is available
- Methodology
  - *Aggregator: aggregate (e.x. FedAvg) the model and learning target that received from the client*
  - *Initializer: initialize the distillation model parameters*
  - *Server Trainer: optimize the distillation model with insensitive data and learning target*
  - *Merger: merge distillation model and server model for next round*

Aggregator

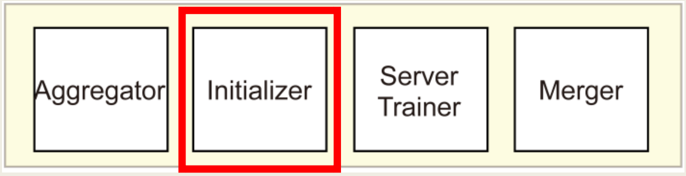


- Compute a server model and average learning target based on all client model and learning target
- The default aggregator is FedAvg

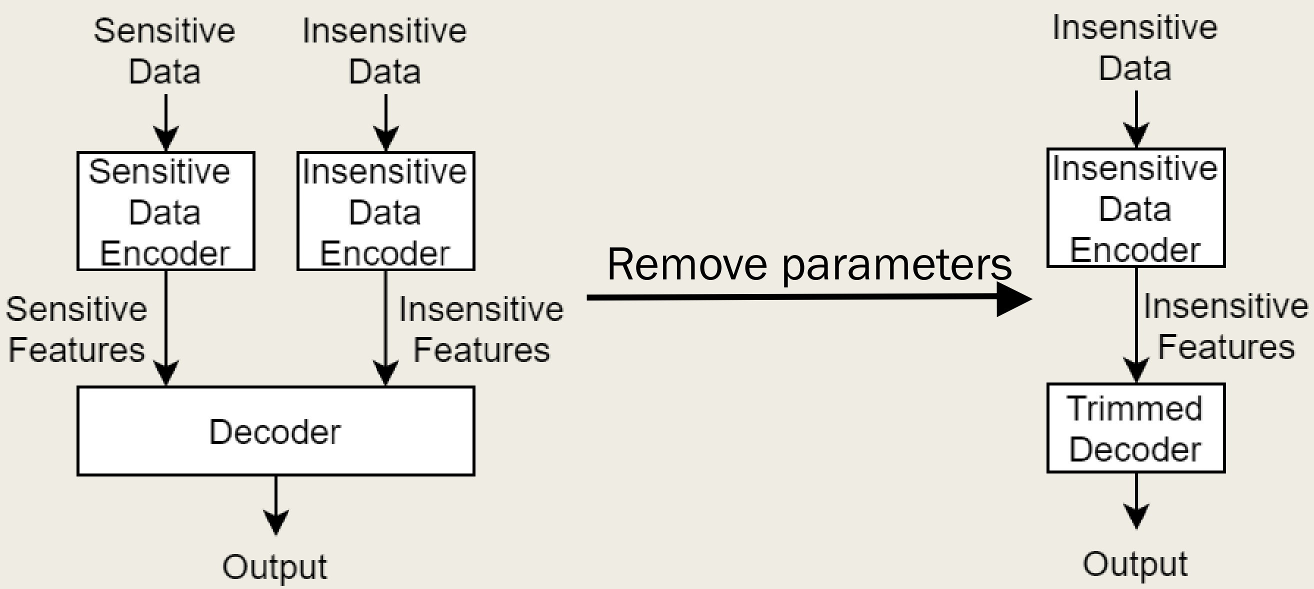
$$\mathbf{M}_{S,t} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{M}_{C,t+1}^{k}, \ \mathbf{T}_t = \frac{1}{K}\sum_{k=1}^{K}\mathbf{T}_t^k.$$

- HPFL can generalizes for advanced aggregator

Initializer



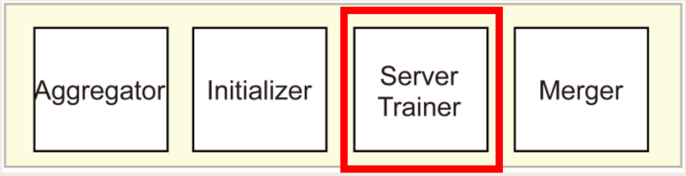
- Target
  - *Generate the distillation model from server model for training with insensitive data*
- Remove the model parameters relevant to sensitive data



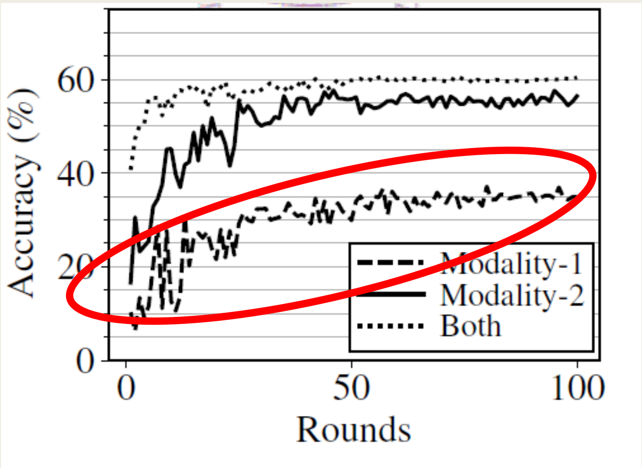
Server model

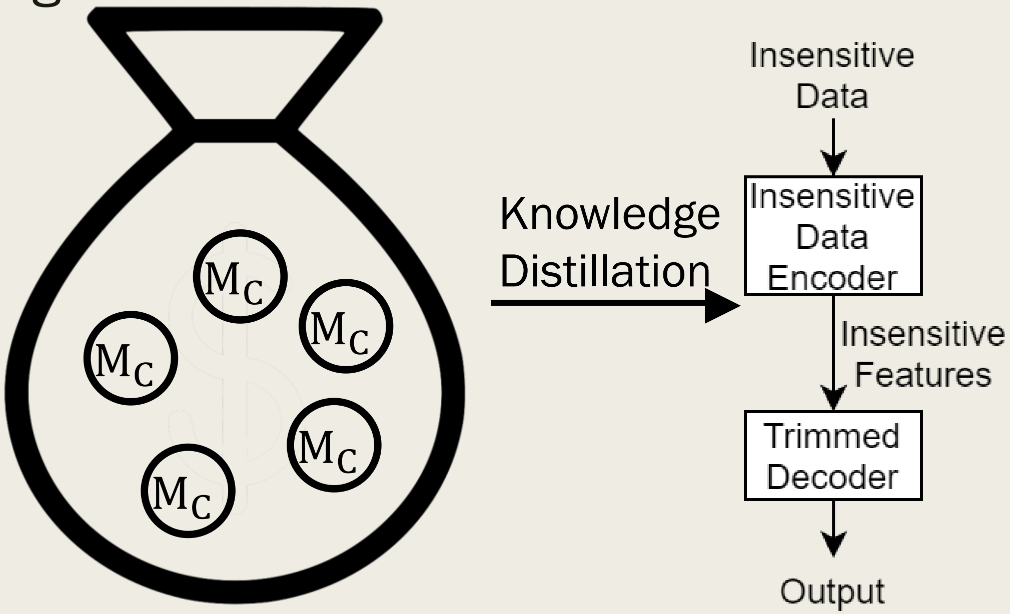
Distillation model

Baseline Trainer



- Baseline algorithm (HP): server trainer trains distillation model with insensitive data
- Train multi-modal model with single modality data causes performance drop
- Distillate the knowledge from client models to distillation model

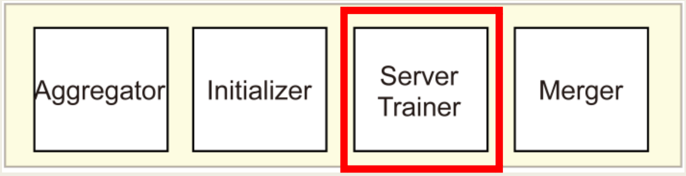




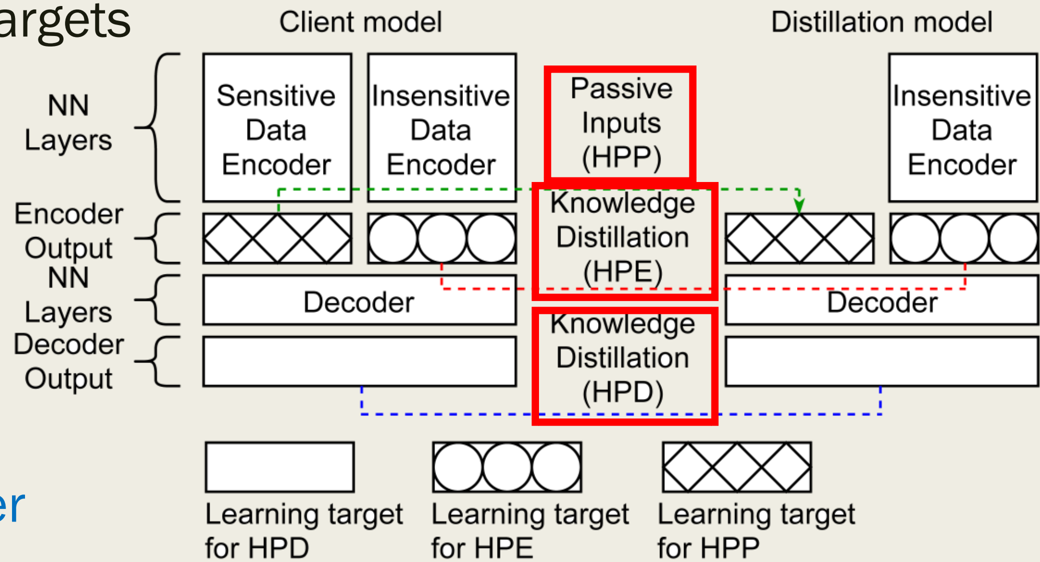
Client models

Distillation model

Learning Target



- Improved federated distillation:
  - *The server only has insensitive data and cannot perform regular distillation*
  - *We propose to have clients upload some layer outputs as learning target to guild the server for distillation*
- Decoder Distillation (HPD) uses the outputs of the last decoder layer as learning targets
- Encoder Distillation (HPE) uses the outputs of all last insensitive data encoder layer as learning targets
- Passive inputs: (HPP) uses the outputs of all last sensitive data encoder layer as learning targets.



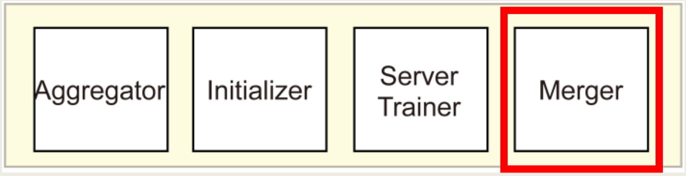
Server Trainer

- Train the distillation model with insensitive data and learning target
- Server loss $$\mathbf{M}_{D,t+1} = \underset{\mathbf{M}_{D,t}}{argmin}\, L_S(\mathbf{D}_I, Y, T_t | \mathbf{M}_{D,t})$$
  - *Label loss (L_G) : the loss between the distillation model prediction and ground truth*
  - *KD loss (L_D): the loss between the distillation model middle layer output and learning targets*
- Overall server loss is a weighted sum of distillation and label loss, where parameter λ is the weight

$$L_S = \lambda\, L_G + (1 - \lambda)\, L_D$$

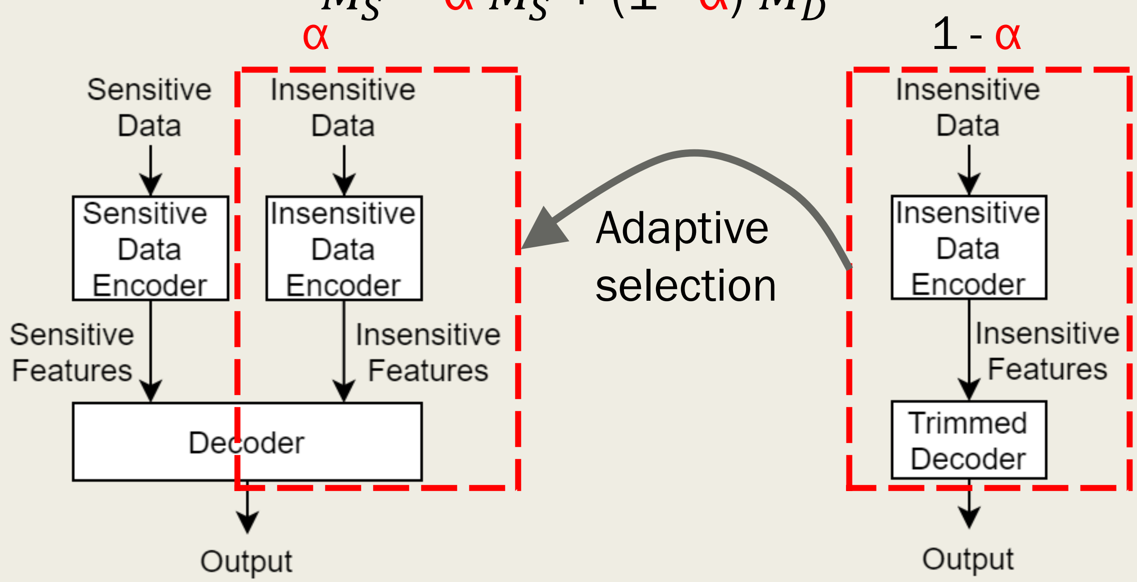
- The choice of λ makes the effect of L_G and L_D on the model closer

Merger



- Clients need a complete multi-modal model for next round training
- Merge the trained distillation model back to the server model
- The merged part depends on different application models
- The contribution of the distillation model to the server model is controlled by parameter α

$$M_S = \alpha\, M_S + (1 - \alpha)\, M_D$$



Server model

Distillation model

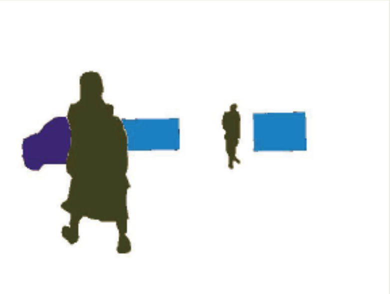
MULTI-MODAL REPRESENTATION LEARNING

Multi-modal Representation Learning

- The HPFL can be applied to many machine learning tasks
- MMRL [1] has become a hot topic due to the growth of multiple sensors and data
  - *Video classification*
  - *Emotion recognition*
  - *Activity detection*
  - *Sentiment analysis*
  - *Semantic segmentation*
- We choose the emotion recognition and semantic segmentation as sample applications

[1] Zhang C, Yang Z, He X, et al. Multimodal intelligence: Representation learning, information fusion, and applications. IEEE Journal of Selected Topics in Signal Processing, 2020, 14(3): 478-493.

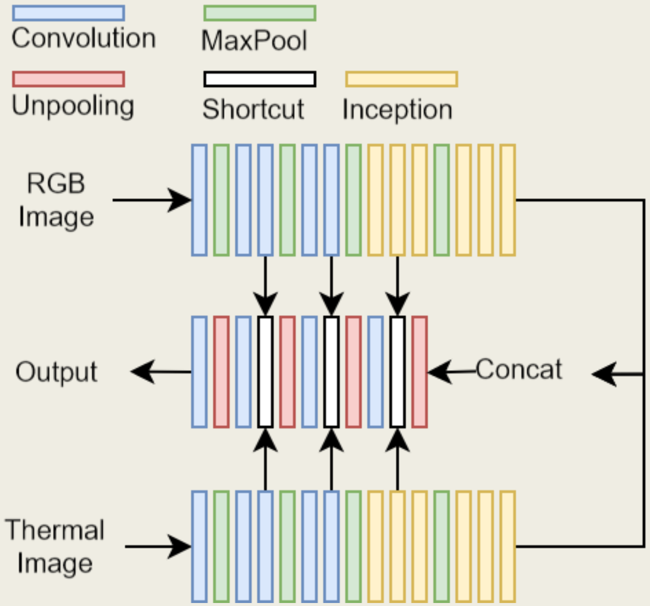
Semantic Segmentation



- Labels each pixel of an image with one or multiple classes
- We choose MFNet [1] as a semantic segmentation task example
- We consider RGB image as sensitive data, thermal image as insensitive data

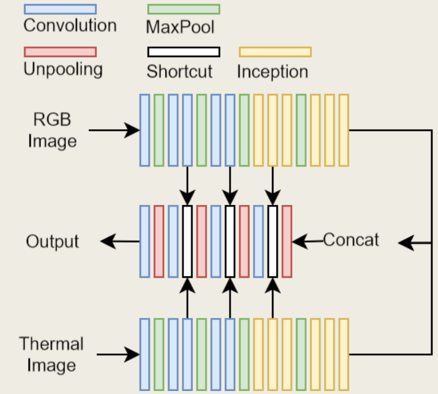
RGB input

Thermal input

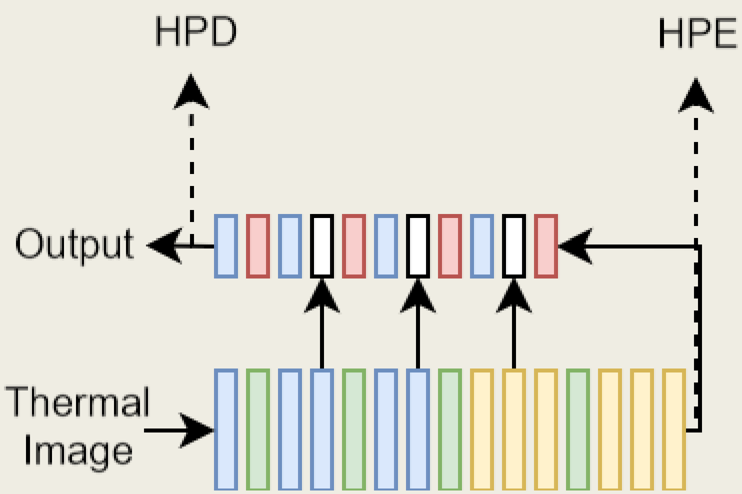


[1] Ha Q, Watanabe K, Karasawa T, et al. MFNet: Towards real-time semantic segmentation for autonomous vehicles with multi-spectral scenes[C]//2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017: 5108-5115.

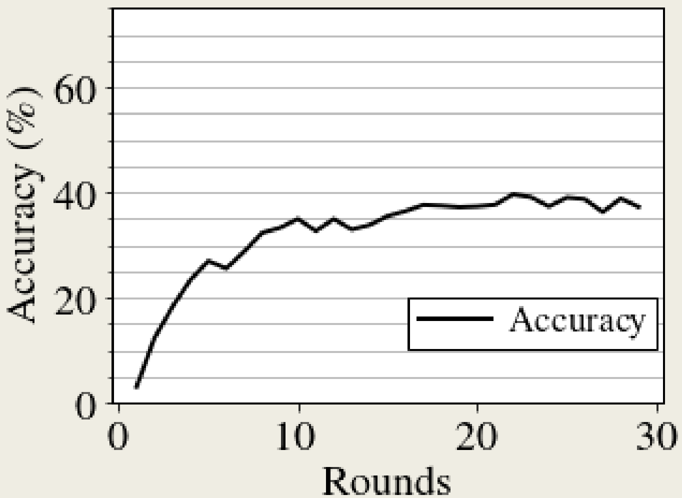
Distillation Model for Semantic Segmentation



- We remove the encoder of RGB image input (sensitive data encoder)
- We put the work point of HPD and HPE in distillation model
- HPP does not work well with MFNet



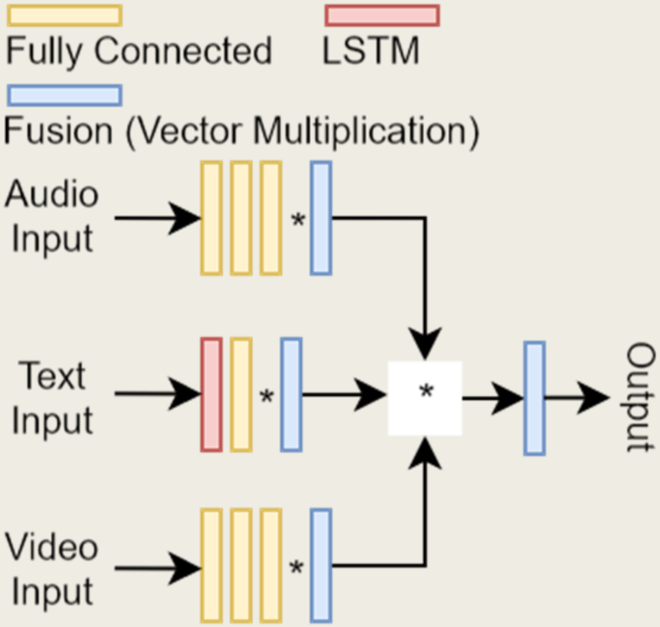
Distillation model



Results of HPP

Emotion Recognition

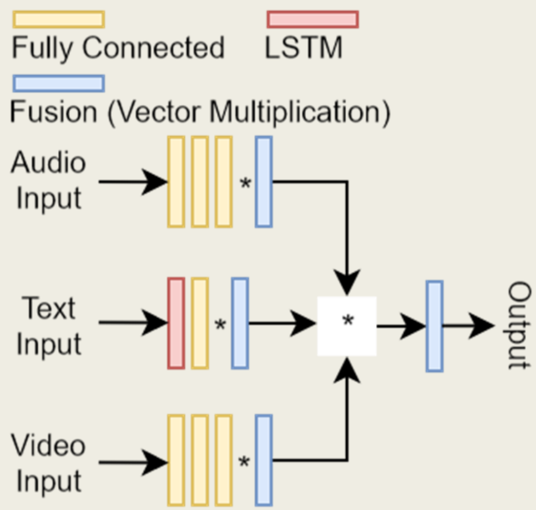
- We choose LMF [1] as a emotion recognition task example
- We consider video and text as sensitive data, audio as insensitive data
- Data provided by IEMOCAP [2] dataset are preprocessed for features

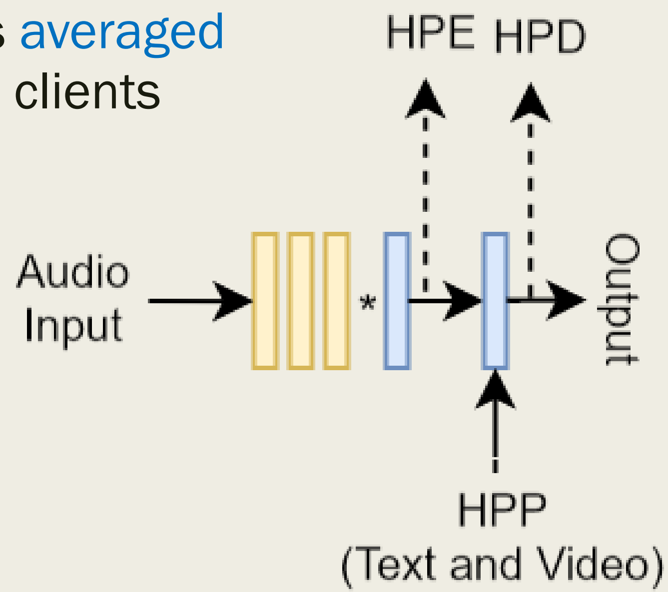


[1] Liu Z, Shen Y, Lakshminarasimhan V B, et al. Efficient low-rank multimodal fusion with modality-specific factors[J]. arXiv preprint arXiv:1806.00064, 2018.
[2] Busso C, Bulut M, Lee C C, et al. IEMOCAP: Interactive emotional dyadic motion capture database[J]. Language resources and evaluation, 2008, 42(4): 335-359.

Distillation Model for Emotion Recognition

- We remove the encoder of text and video input (sensitive data encoder)
- We put the work point of HPD and HPE in distillation model
- The learning target of HPP contains averaged text and video encoder output from clients





EXPERIMENT SETUP

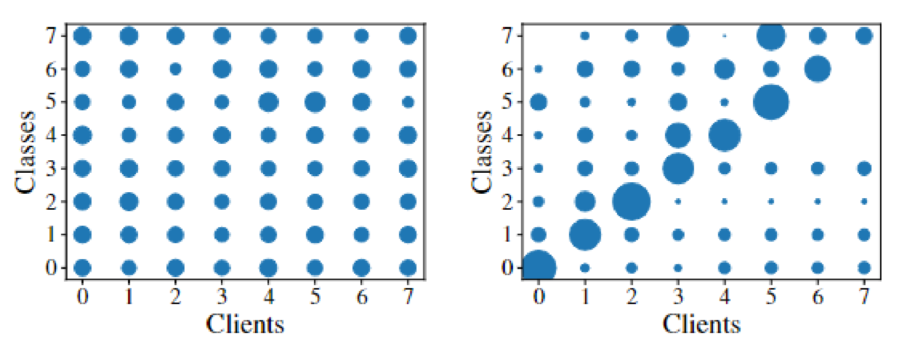
Benchmarking Algorithms

- FedAvg [1]: the earliest federated learning algorithm
- FedProx [2]: optimize client trainer by proximal term
- FedAdam [3]: add an Adam optimizer on aggregator
- FedDyn [4]: optimize client trainer by linear and quadratic penalty
- FedCon [5]: optimizes the client trainer to decrease the distance between representation learned from client model and server model

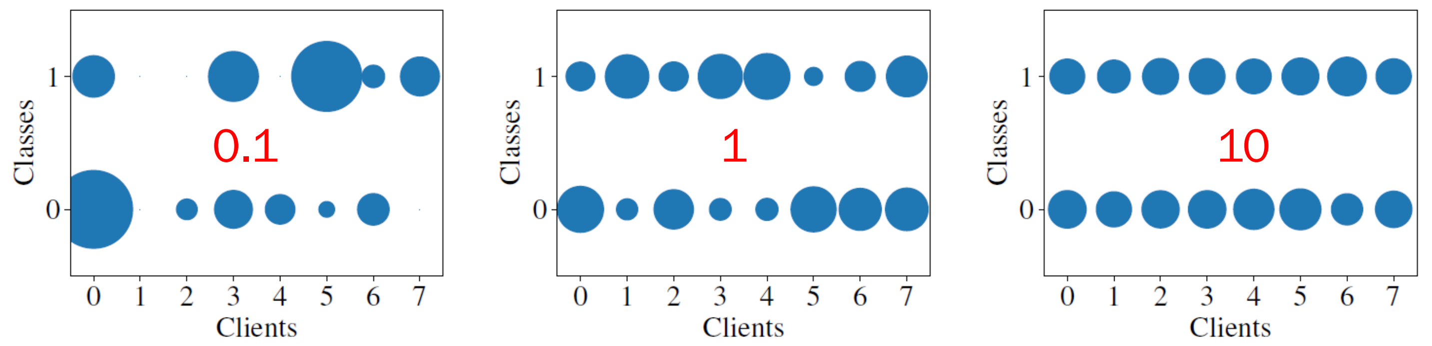
[1] McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data[C]//Artificial intelligence and statistics. PMLR, 2017: 1273-1282.
[2] Sahu A K, Li T, Sanjabi M, et al. On the convergence of federated optimization in heterogeneous networks[J]. arXiv preprint arXiv:1812.06127, 2018, 3: 3.
[3] Acar D A E, Zhao Y, Navarro R M, et al. Federated learning based on dynamic regularization[J]. arXiv preprint arXiv:2111.04263, 2021.
[4] Li Q, He B, Song D. Model-contrastive federated learning[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 10713-10722.
[5] Reddi S, Charles Z, Zaheer M, et al. Adaptive federated optimization[J]. arXiv preprint arXiv:2003.00295, 2020.

Dataset and Data Distribution

- MFNet: totally 1600 and 300 pairs of RGB and thermal images in the training and testing sets



- LMF: totally 3515 and 938 triplets of video, audio, and text data in the training and testing sets
- We utilize the Dirichlet distribution to generate three sample distribution degrees



Parameters

- Distillation model contribution: $\alpha \in \{0.0, 0.1, 0.3, \underline{\mathbf{0.5}}, 0.7\}$
- Loss balance parameter: $\lambda \in \{0.05, \underline{\mathbf{0.1}}, 0.2\}$
- Distillation method ∈ {HP, HPD, **<u>HPE</u>**, HPP*}
- Data distribution ∈ {i.i.d., **<u>non-i.i.d.</u>**} or ∈ {0.1, **<u>1</u>**, 10}.
- Number of clients ∈ {**<u>8</u>**, 16, 32}
- Neural network parameters and baseline algorithm parameters have been tuned and fixed
- The merge part selection experiments were omitted

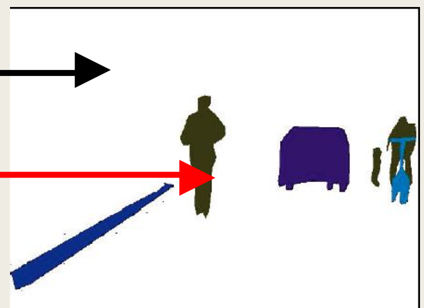
* HPP only works on emotion recognition problem

Metrics

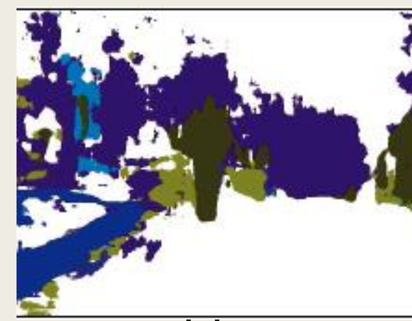
- Background accuracy: The ratio of background pixels that are correctly classified in the semantic segmentation problem.
- Foreground accuracy: The ratio of non-background pixels that are correctly classified in the semantic segmentation problem.

Background pixels

Foreground pixels



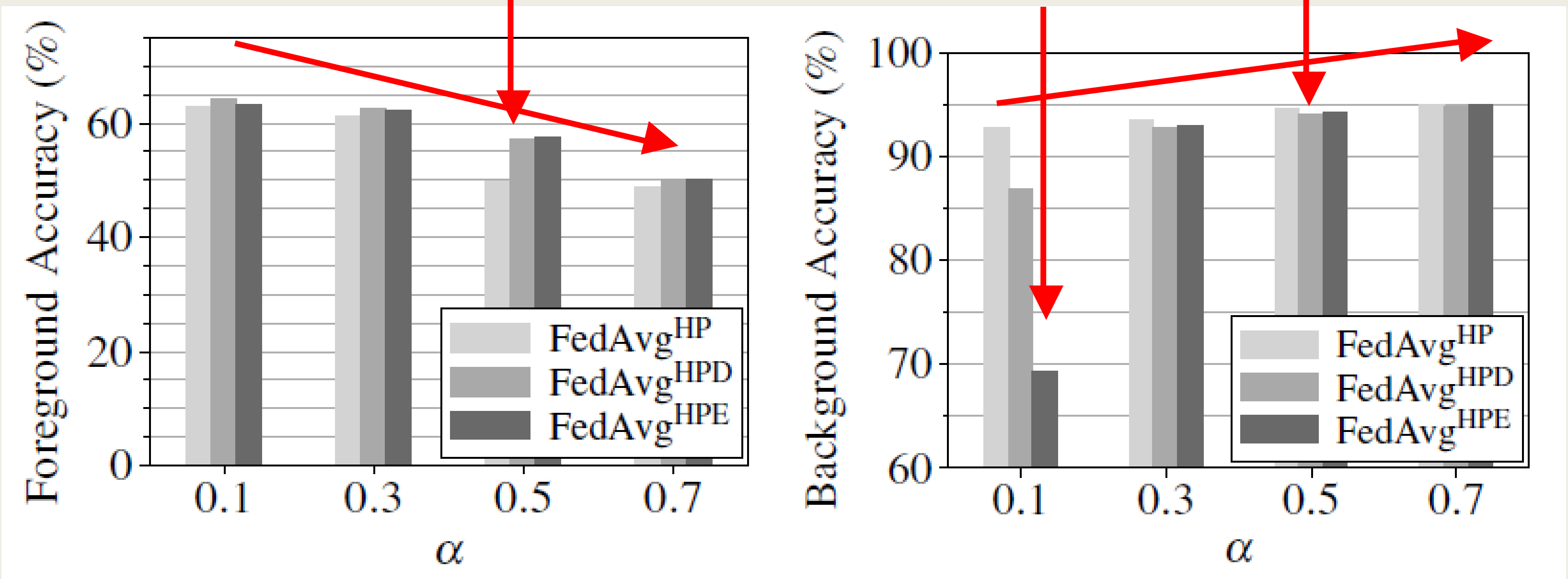
Low background Accuracy



- F1-score: The weighted F1-score in the emotion recognition problem.
- Throughput: The network resource consumption for transmitting model parameters and learning targets.
- CPU time: The time taken by each client in a round.

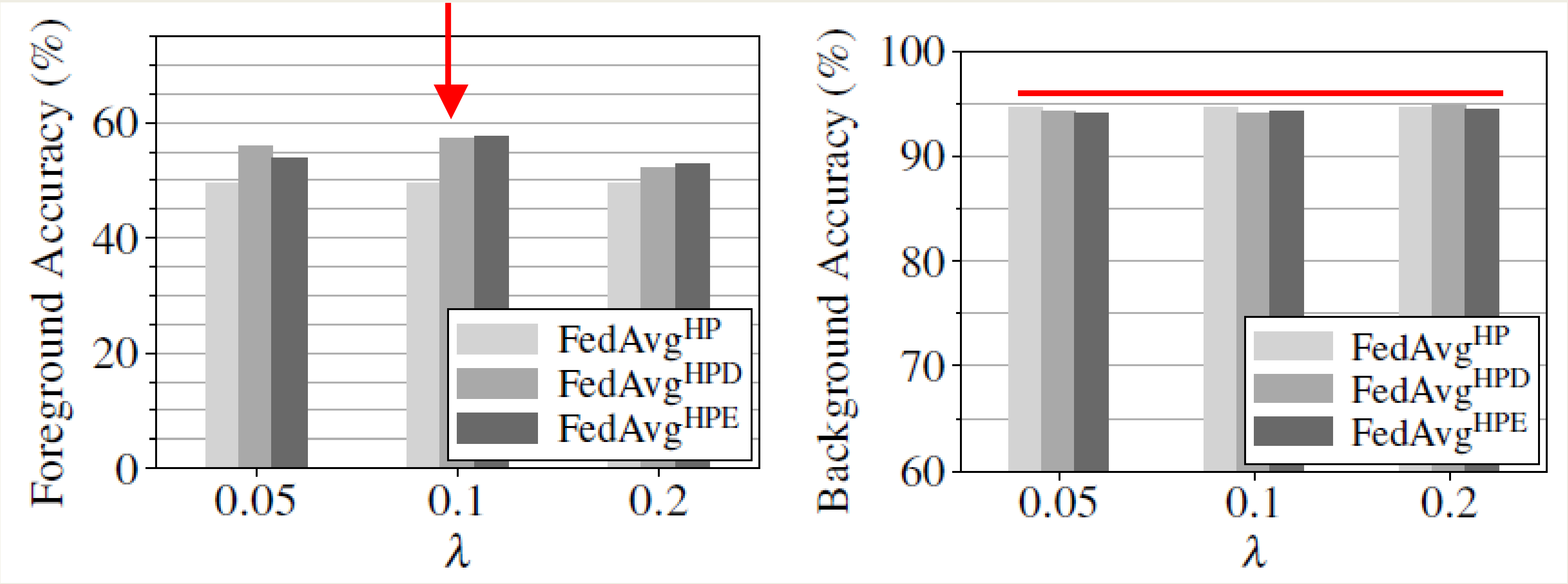
EXPERIMENT RESULTS

MFNet: α = 0.5 is a Better Choice



- Smaller α values lead to slightly higher foreground accuracy but much lower background accuracy
- Omit the α = 0.0 experiments in MFNet
- α = 0.5 gives 57.59% on foreground accuracy and 94.27% on background accuracy

MFNet: λ = 0.1 is a Better Choice



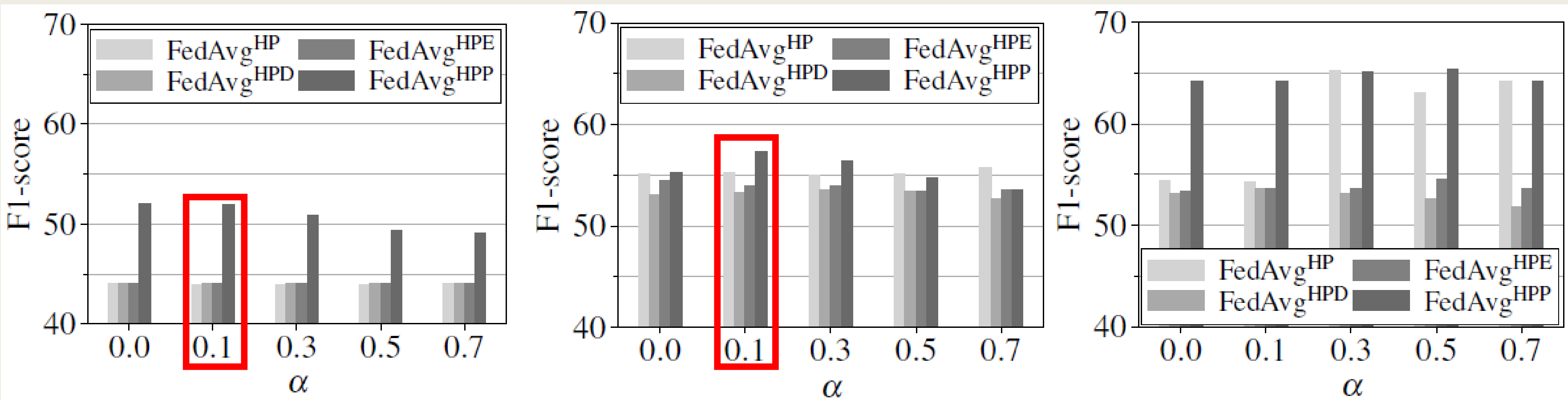
- λ = 0.1 results in the highest foreground accuracy
- $\text{FedAvg}^{\text{HPD}}$ and $\text{FedAvg}^{\text{HPE}}$ achieve higher (7.51%, 7.99%) foreground accuracy than $\text{FedAvg}^{\text{HP}}$
- Select λ = 0.1 in the following MFNet experiments

MFNet: $\text{FedAvg}^{\text{HPE}}$ Leads to Lower Communication Overhead

	Method	Model		Insensitive data		Learning target	
MFNet	HP	5.96 MB	96.62%	6.25 MB	3.38%	/	/
	HPD		36.86%		1.29%	10 MB	61.85%
	HPE		94.05%		3.29%	169 KB	2.67%

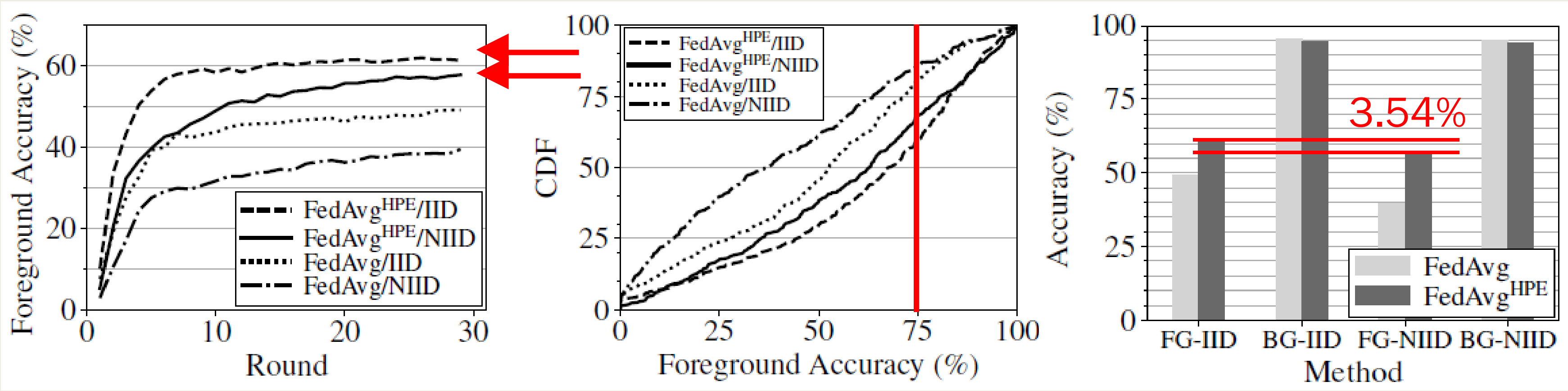
- The percentage represents the proportion of overhead during training
- $\text{FedAvg}^{\text{HPD}}$ consumes 61.85% communication overhead on transmit learning target
- Select $\text{FedAvg}^{\text{HPE}}$ in the following MFNet experiments

LMF: α = 0.1 and $\text{FedAvg}^{\text{HPP}}$ Leads Higher F1-score



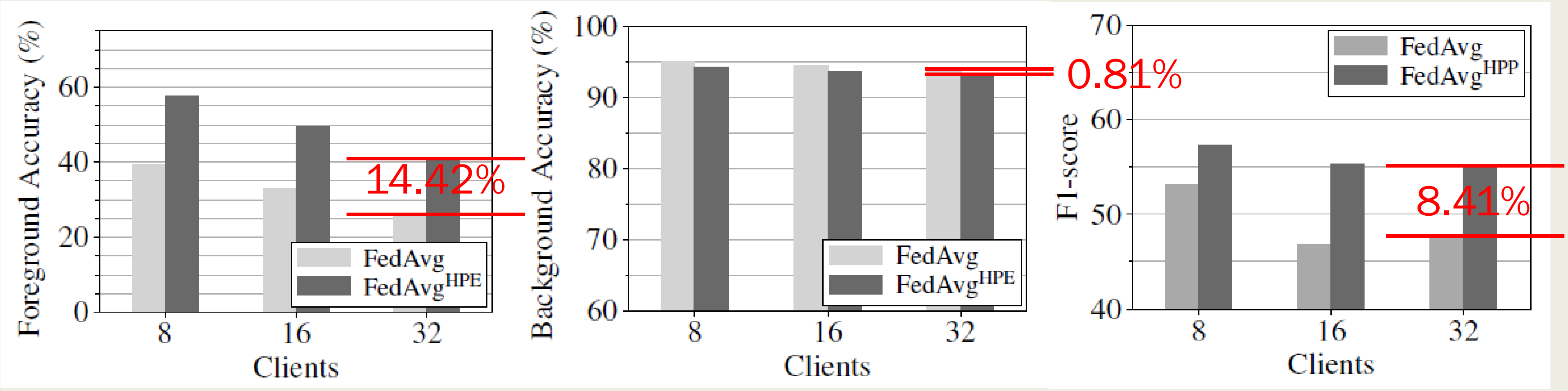
- Omit the experiments of λ, which has little effect on results
- In two bias sample distributions (0.1, 1), α = 0.1 outperform other α values
- Select α = 0.1 and $\text{FedAvg}^{\text{HPP}}$ in the following LMF experiments

MFNet: $\text{FedAvg}^{\text{HPE}}$ Works Under both i.i.d. and Non-i.i.d.



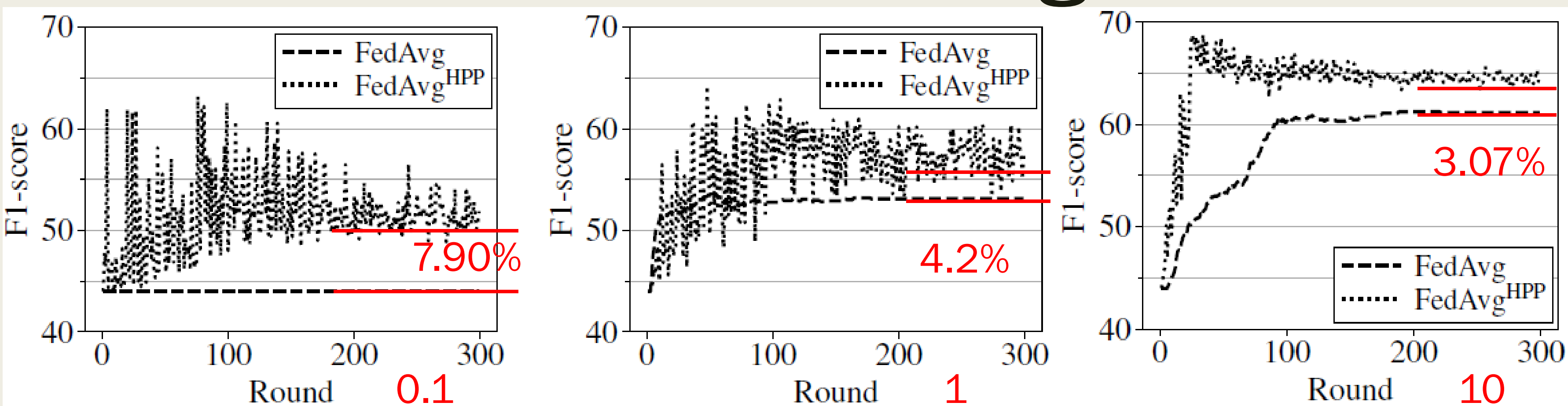
- $\text{FedAvg}^{\text{HPE}}$ achieves higher foreground accuracy under two sample distributions
- $\text{FedAvg}^{\text{HPE}}$ results 40.05%, 37.34% test samples with 75%+ foreground accuracy under two sample distributions
- $\text{FedAvg}^{\text{HPE}}$ only losss 3.54% foreground accuracy from i.i.d. to non-i.i.d.

MFNet, LMF: FedAvg$^{\text{HPE/P}}$ Works with Different No. Clients

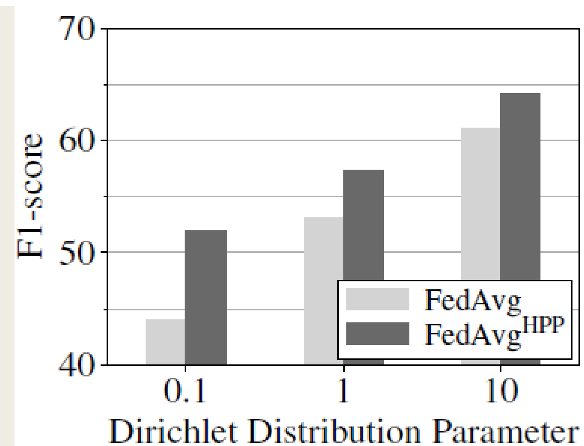


- FedAvg$^{\text{HPE}}$ outperforms FedAvg by at least 14.42% on foreground accuracy (MFNet)
- The background accuracy loss is negligible 0.81%
- FedAvg$^{\text{HPP}}$ outperforms FedAvg 8.41% on F1-score (LMF)
- HPFL improves the robustness with more clients

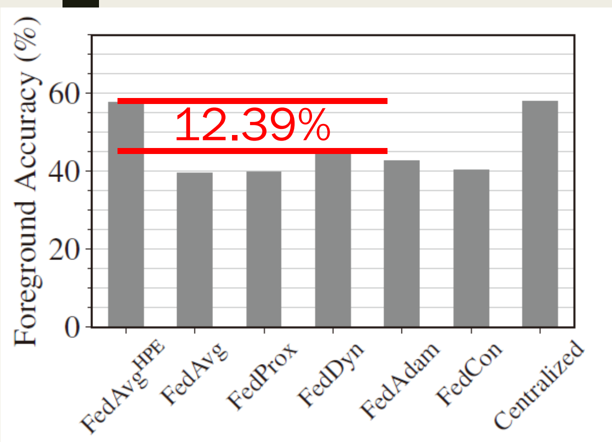
LMF: FedAvg^{HPP} Works with Different non-i.i.d. Deg.

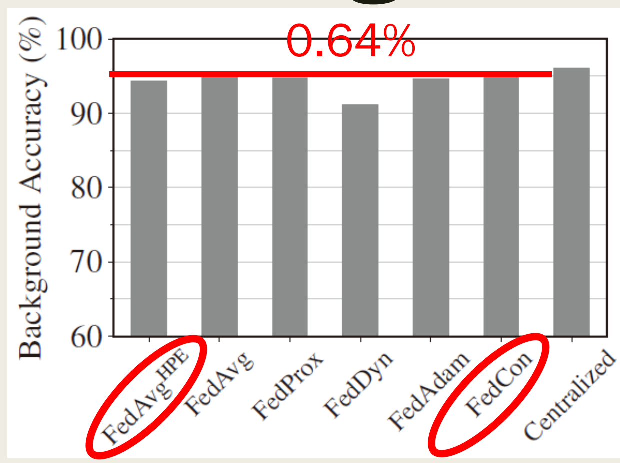


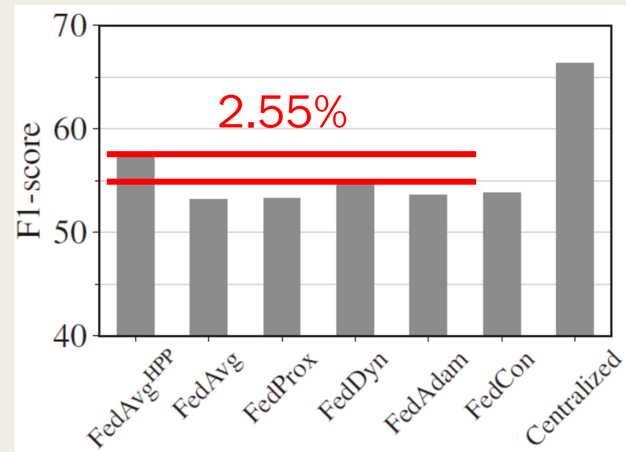
- FedAvg^{HPP} outperforms FedAvg 3.07% ~ 7.9% on F1-score
- Select the hardest distribution 1 in the following experiments



HPFL Outperforms Other Advanced FL Algorithms

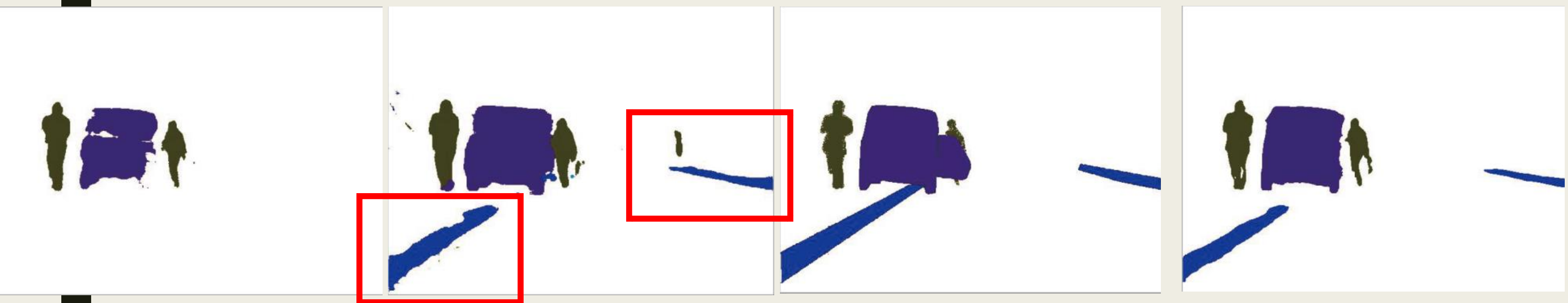




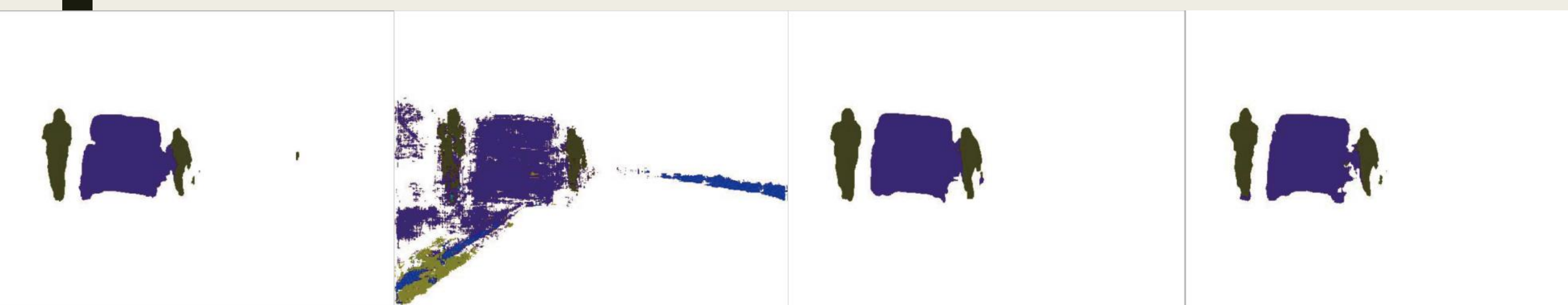


- FedAvg^{HPE} outperforms all state-of-the-art advanced FL algorithms at least 12.39% on foreground accuracy (MFNet)
- FedAvg^{HPE} losses negligible 0.64% on background accuracy (MFNet)
- FedAvg^{HPP} also outperform all state-of-the-art advanced FL algorithms by 2.55% on F1-score (LMF)

Qualitative Results



FedAvg HPFL Ground Truth Centralized



FedProx FedDyn FedAdam FedCon

HPFL Applies on Advanced FL Algorithms

Algorithms and Models	FedAvg		FedProx		FedDyn		FedAdam		FedCon	
	Orig.	HPFL	Orig.	HPFL	Orig.	HPFL	Orig.	HPFL	Orig.	HPFL
MFNet (Fore. Accu.)	39.39	+18.2	39.89	+17.21	45.19	+12.98	42.63	-0.36	40.42	+14.25
LMF (F1-score)	53.12	+4.2	53.22	+3.54	54.78	-0.81	53.62	+1.42	53.75	+5.48

- Update the client trainer to implement FedProx, FedDyn, and FedCon
- Replace the aggregator with Adam to implement FedAdam
- Apply HPFL on most of advanced FL algorithms can obtain performance improvement

HPFL Incurs Low Communication and Computation Overhead

	Methods	Model		Insensitive data		Learning target	
MFNet	HP	5.96 MB	96.62%	6.25 MB	3.38%	/	/
	HPD		36.86%		1.29%	10 MB	61.85%
	HPE		94.05%		3.29%	169 KB	2.67%
LMF	HP	1.07 MB	99.87%	413 KB	0.13%	/	/
	HPP		99.85%		0.13%	0.26 KB	0.02%

Methods	HPFL	FedAvg	FedAdam	FedProx	FedDyn	FedCon
Computation Overhead*	100%	100%	100%	114%	124%	120%

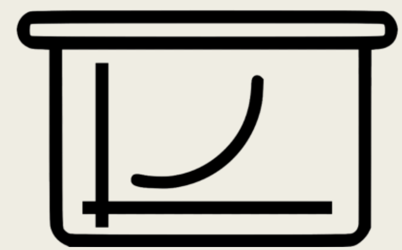
Conclusion

- First considered multi-sensor classification problems in the FL setup, where sensor data have diverse privacy sensitivity levels
- Proposed HPFL algorithm to utilize privacy-insensitive data at server-side for reduce the performance gap between FL and centralized ML
- Conducted an extensive comparison of HPFL and other state-of-the-art advanced FL algorithms:
  - *HPFL causes no client-side computation overhead and little communication overhead*

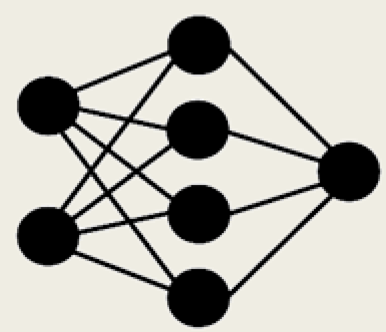
Future Work

- Optimization on communication and performance

- Convergence and privacy analysis



- Support complex model structure



Publication and Cooperators

Y. Chen, C. Hsu, C. Tsai, and C. Hsu, “HPFL: Federated Learning by Fusing Multiple Sensor Modalities with Heterogeneous Privacy Sensitivity Levels” ACM Multimedia, October 2022, Submitted to ACM.

Y. Wu, **Y. Chen**, S. Shirmohammadi, and C. Hsu, “AI-Assisted Food Intake Activity Recognition Using 3D mmWave Radars” ACM International Workshop on Multimedia Assisted Dietary Management, October 2022, In-Preparing submission.

Y. Wu, **Y. Chen**, and C. Hsu, “Heterogeneous Privacy Distributed Learning: Collaborative Concept and Applications” IEEE Transactions on Multimedia, 2022, In-Preparing submission.

Chih-Fan Hsu, National Yang Ming Chiao Tung University
Chung-Chi Tsai, Qualcomm Inc.
Shervin Shirmohammadi, University of Ottawa

Thanks for Your Listening

Q ‡ A

BACKUP SLIDES

Limitation

- Dataset
  - *HPFL can only work on multi-modal dataset and application*
- Distillation model generation
  - *All performance gain comes from distillation model*
  - *No obvious standard to generate distillation model*
  - *Hard to generate distillation from complex model structure*
- System optimization
  - *HPFL introduces extra tunable parameters*
  - *Client and serve-side optimization are inferencing each one*

Support more Complex Multi-modal Network Structures

- We only consider the popular joint representation structures in our experiments
- Generating distillation model from complex models is challenging for HPFL
- We will select more representative multi-modal structures, and apply HPFL on them
- We will propose a generalize rules about applying HPFL on multi-modal model and selecting HPD, HPE, and HPP

Optimization on Communications and Computations

- HPFL system requires a powerful server
- The insensitive data and learning target transmit may cause network congestion
- The HPFL server requires about 6 times of training time than clients
- We will develop a efficient learning target and insensitive data sharing algorithm under limited network resource
- We will consider more efficiency training scenario

Deeper Privacy Analysis

- Privacy leakage risk from: insensitive data, learning target, model parameters
- Multi-modal machine translation may break the gap between sensitive and insensitive data (depth, thermal images…)
- We will provide completely privacy analysis and apply some privacy-prevention technology (DP) on HPFL
- We will explore the performance impact of these privacy-preserving technologies on HPFL

Convergence Analysis

- Target function:

$$\lim_{t\to\infty}\frac{1}{t}\sum_{t=1}^{\infty}|L_S(\mathbf{D}_E,Y|\mathbf{M}_{S,t})-L_S(\mathbf{D}_E,Y|\mathbf{M}_S^*)|=0,\ \text{where}$$

$\mathbf{M}_{S,t}$ comes from Eq. 4.4,

$\mathbf{M}_S^*$ is the optimal server model, and (8.1)

$$L_S(\mathbf{D}_E,Y|\mathbf{M}_{S,t})=\frac{1}{|\mathbf{D}_E|}\sum_{i=1}^{|\mathbf{D}_E|}CE(\mathbf{M}_{S,t}(\mathbf{D}_{E,i}),Y_i).$$

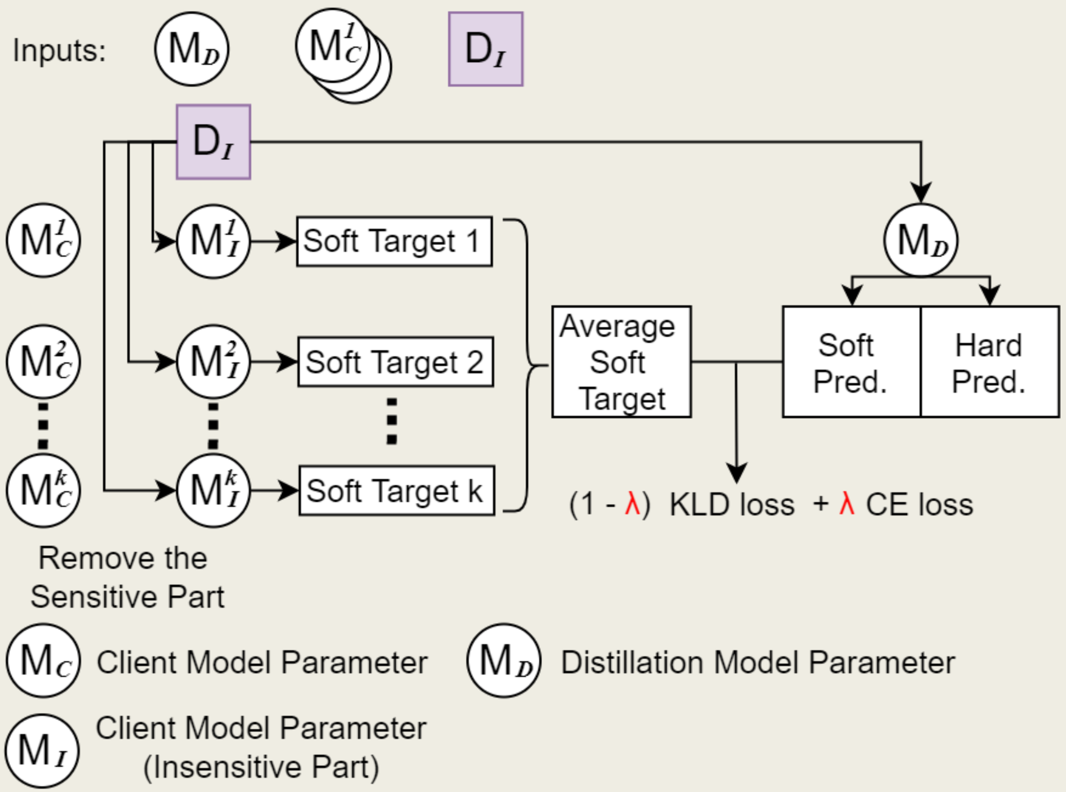
- Merger: $$\mathbf{M}_{S,t+1}=\alpha\times\mathbf{M}_{S,t}+(1-\alpha)\times\mathbf{M}_{D,t+1}, \quad (4.4)$$

- Difficulties
  - The merge part (distillation model) is various in different models
  - No such reference work did the similar prove
  - Hybrid Federated and Centralized Learning [2] merges the same client and server model

Ref: [2] Elbir A M, Coleri S, Mishra K V. Hybrid federated and centralized learning[C]//2021 29th European Signal Processing Conference (EUSIPCO). IEEE, 2021: 1541-1545.

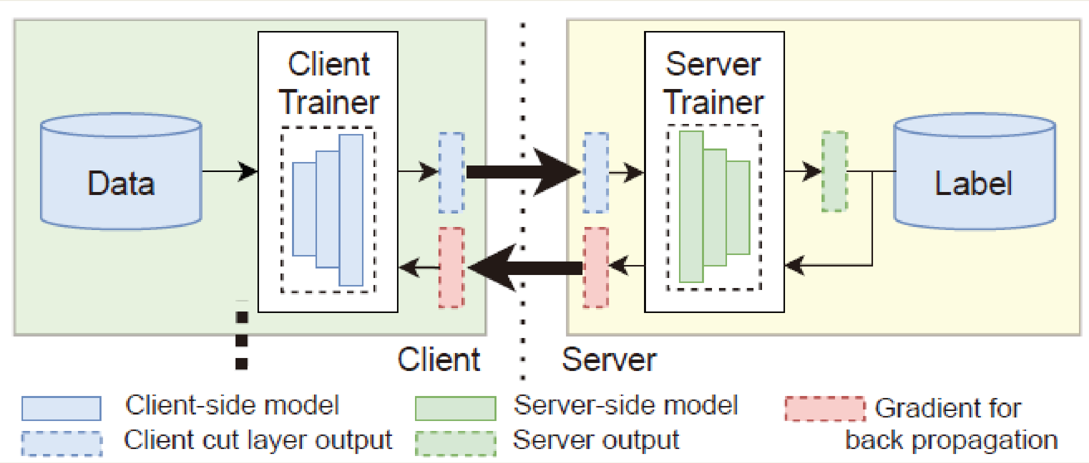
Server-side Distillation Method

- Apply learning target (HPD, HPE) has little performance improvement (MFNet, LMF)



Generalization for Split Learning

- Split learning (SL) significantly reduces client computing overhead
- Similar but different to SL, HPFL cut the model based on diverse privacy-sensitive levels
- We target to distributed train sensitive data at client-side and centralized train insensitive data at server-side



Full Algorithm

Algorithm 1 The Proposed HPFL Procedure

Initialize $\mathbf{M}_{S,0}$ with random parameters
for client $k = 1, 2, \ldots, K$ **(in parallel) do**
  Upload insensitive data $\mathbf{D}_I^k$ to the server
for each round $t = 1, 2, \ldots$ **do**
  for client $k = 1, 2, \ldots, K$ **(in parallel) do**
    Receive $\mathbf{M}_{S,t-1}$ from server
    Compute $\mathbf{M}_{C,t}^k, \mathbf{T}_t^k$ using Eq. (1) // Client trainer
  Compute $\mathbf{M}_{S,t}, \mathbf{T}_t$ using Eq. (2) // Aggregator
  Initialize $\mathbf{M}_{D,t}$ using Eq. (3) // Initializer
  Compute $\mathbf{M}_{D,t}$ using Eq. (4) // Server trainer
  Compute $\mathbf{M}_{S,t}$ using Eq. (5) // Merger
  Break **if** $\mathbf{M}_{S,t}$ converges