UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Clustered Federated Learning

by Jie MA

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Guodong Long and Jing Jiang

University of Technology Sydney
Faculty of Engineering and Information Technology

Oct 2023

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Jie MA, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature: Production Note:
Signature removed prior to publication.

Date: 23/10/2023

Dedication

To my lovely family ...

ACKNOWLEDGMENTS

Firstly, I would like to express my heartfelt appreciation to my supervisor, A/Prof. Guodong Long, and my co-supervisor, A/Prof. Jing Jiang. Their unwavering support, expert guidance, timely assistance, and constant encouragement have been invaluable throughout my Ph.D. journey, especially during the unprecedented challenges posed by the Covid-19 pandemic. I am truly privileged and honored to have them as mentors, who have consistently inspired me to strive for excellence in my academic endeavors. I am also immensely grateful to Distinguished Prof. Chengqi Zhang for his priceless advice regarding both my professional and personal life. His remarkable wisdom and profound insights have been a constant source of inspiration.

In addition, I wish to convey my gratitude to my external supervisor, Tianyi Zhou, Assistant Professor at the University of Maryland, College Park, as well as my esteemed co-authors Dr. Ming Xie, Yijun Yang, and Yue Tan. Working alongside these talented, open-minded, and intellectually curious individuals has been an extraordinary experience, and I have gained a wealth of knowledge from them. I wish them continued success and happiness in their professional and personal lives.

Furthermore, I would like to extend my thanks to my colleagues and friends, whom I acknowledge in no particular order: Dr. Tao Shen, Dr. Lu Liu, Dr. Han Zheng, Wensi Tang, Peng Yan, Zhuowei Wang, Shuang Ao, Dr. Fengwen Chen, Yang Li, Dr. Kaize Shi, Chang Shao, Zhihong Deng, Dr. Xueping Peng, Dr. Wei Huang, Dr. Adi Lin, Tian Lan, Feng Shan, Li Wang, Guanqing Zhang, Jianjun Chen, and Kai Xu. It has been an absolute pleasure to spend time with these kind-hearted, intelligent, and amiable individuals. Their positive

energy, camaraderie, and mutual support have created an enriching environment that has made my academic journey truly enjoyable and rewarding. I feel incredibly fortunate to have crossed paths with such remarkable people and look forward to cherishing the friendships and memories we have created together. I wholeheartedly wish them a bright and prosperous future.

I would also like to express my sincere appreciation for the staff members at UTS, including those at the Australian Artificial Intelligence Institute, School of Computer Science, Faculty of Engineering and Information Technology, iHPC, GRS, and the Library. Their unwavering support and assistance during my Ph.D. studies have been instrumental to my success.

Last and foremost, I reserve my most profound gratitude for my lovely wife and cute daughter, great parents, and extended family, with a special mention of my father-in-law, maternal uncle and late grandfather. Their steadfast support and belief in me have been the bedrock of my achievements, and I could not have reached this milestone without them. I wish them a lifetime of health, happiness, and joy!

Abstract

Heterogeneous federated learning without assuming any structure is challenging due to the conflicts among non-identical data distributions of clients. In practice, clients often comprise near-homogeneous clusters so training a server-side model per cluster mitigates the conflicts, which is called clustered FL. With new insights and perspectives, we propose a unified bi-level optimization framework for clustered FL methodologies.

Based on this, we present a fundamental method called Weighted Clustered Federated Learning (WeCFL). Additionally, we introduce a novel theoretical analysis framework for its convergence analysis. This framework factors in the clusterability among clients to measure the effects of intra-cluster non-IIDness, and a linear convergence rate of $O(1/T)$ is achieved.

To enhance the robustness of clustering, we propose a methodology termed Clustered FL with Contrastive Learning (CFL-CON), which can be integrated into our previously proposed clustered FL frameworks and many other clustered FL methods. We propose two variants based on the space of representation and parameters respectively.

To address the lack of knowledge sharing due to robust clustering and to improve performance, we propose another generic add-on technique, Clustered FL with Clustered Knowledge Sharing (CFL-CKS). We conduct a theoretical analysis of the term's simplification, convergence, and interpretation, providing a comprehensive understanding.

Furthermore, to bridge the trade-off between these two add-ons, we propose Clustered

FL with Contrastive Learning and Clustered Knowledge Sharing (CFL-CON&CKS). This method applies contrastive learning to the head of the neural network to create distance, and knowledge sharing to the backbone of the neural network to facilitate knowledge sharing.

Lastly, to address the problem of clustering collapse and to stabilize clustered FL, we propose Clustered Additive Modeling (CAM). This method applies a globally shared model along with the cluster-wise models. The global model captures the features shared by all clusters, so cluster-wise models are enforced to focus on the differences among clusters. The asymptotic convergence rate of $O(1/\sqrt{T})$ is proved.

Experimental simulations also demonstrate the superiority of our methods in terms of robustness, stability of clustering, effectiveness in mitigating clustering collapse and performance. All methods are implemented with unified datasets, non-IID settings, models, optimizers, baselines, as detailed in the appendix, to ensure consistency. The code framework, FedBase, has been open-sourced via PyPI [*] and GitHub [†].

Keywords: Federated Learning, Clustering structure, Unified framework, Convergence analysis, Contrastvie learning, Knowledge sharing, Additive modeling.

[*]https://pypi.org/project/fedbase
[†]https://github.com/jie-ma-ai/FedBase

List of Publications

Conference

1. **Ma, J.**, Xie, M., & Long, G. (2022, November). Personalized Federated Learning with Robust Clustering Against Model Poisoning. In *Advanced Data Mining and Applications: 18th International Conference, ADMA 2022, Brisbane, QLD, Australia, November 28-30, 2022, Proceedings, Part II (pp. 238-252).* Cham: Springer Nature Switzerland. (*Best Paper Award of ADMA 2022*).

2. Xie, M.*, **MA, J.***, Long, G., & Zhang, C. (2023, February). Robust Clustered Federated Learning with Bootstrap Median-of-Means. In *Web and Big Data: 6th International Joint Conference, APWeb-WAIM 2022, Nanjing, China, November 25-27, 2022, Proceedings, Part I (pp. 237-250).* Cham: Springer Nature Switzerland. (*APWEB-WAIM 2022*).

3. **Ma, J.**, Zhou, T., Long, G., Jiang, J., & Zhang, C. (2023) Structured Federated Learning through Clustered Additive Modeling. (*NeurIPS-2023*).

4. **Ma, J.**, Long, G., Jiang, J., & Zhang, C. (2023) Enhancing Clustered Federated Learning: An Add-On Leveraging Contrastive Learning and Knowledge Sharing. (*To be submitted to IJCAI-PRICAI-2024*).

*Equal contributions.

5. Yang, Y., Jiang, J., Zhou, T., **Ma, J.**, & Shi, Y. (2021). Pareto policy pool for model-based offline reinforcement learning. In *International Conference on Learning Representations*. (*ICLR-2022*).

6. Tan, Y., Long, G., **Ma, J.**, Liu, L., Zhou, T., & Jiang, J. (2022). Federated learning from pre-trained models: A contrastive learning approach. *arXiv preprint arXiv:2209.10083*. (*NeurIPS-2022*).

Journal

1. **Ma, J.**, Long, G., Zhou, T., Jiang, J., & Zhang, C. (2022). On the convergence of clustered federated learning. *arXiv preprint arXiv:2202.06187*. (*Under revision of TNNLS*).

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE **Page**

List of Tables

Abbreviation

FL	Federated Learning
GFL	Generic federated learning
HFL	Horizontal federated learning
VFL	Vertical federated learning
FTL	Federated transfer learning
PFL	General personalized Federated Learning methods
Clustered FL	General clustered Federated Learning methods
CFL	One clustered FL method proposed by [74]
IID	Independent and identically distributed
non-IID	Not independent and identically distributed
CNN	Convolutional neural network
SGD	Stochastic gradient descent
FMTL	Federated multi-task leraning
MOM	Median-of-means
KL divergence	Kullback-Leibler divergence
FedAvg	First FL algorithm proposed by Google [67]
FedProx	A FL algorithm proposed by [51]
DCFL	Dynamic Clustering Federated Learning [12]
FLSC	Federated Learning with Soft Clustering [48]
IFCA	Iterative Federated Clustering Algorithm proposed by [30]

FeSEM	One clustered FL method called Federated Stochastic Expectation Maximization proposed by [94]
WeCFL	Weighted Clustered Federated Learning framework proposed by [62]
CFL-CON, CON	A contrastive learning method in clustered FL proposed in Chapter 4
CFL-CON-rep	A contrastive learning method based on representations in clustered FL proposed in Chapter 4
CFL-CON-para	A contrastive learning method based on model parameters in clustered FL proposed in Chapter 4
CFL-CKS, CKS	A clustered Knowledge Sharing method in clustered FL proposed in Chapter 5
CFL-CON&CKS	A method combining CFL-CON and CFL-CKS in clustered FL proposed in Chapter 6
CAM	Clustered Additive Modelling
AFL	Additive FL
LLM	Large Language Model
LOF	Local outlier factor

NOTATIONS

General Notations

$\{\cdot\}$	A set
$\mathbb{R}$	The set of real numbers
$\mathbb{A}\setminus\mathbb{B}$	Set subtraction. Set of elements in $\mathbb{A}$ but not in $\mathbb{B}$.
$\mathbb{E}[\cdot]$	Expectation
$d(\cdot,\cdot)$	The general form of distance function
$\|\cdot\|_p$	L^p norm
$\nabla_x f$	Gradient of f with respect to x
ϵ	The error bound
$\cdot$	The dot product

FL Notations

m	Number of clients in FL system
D_i	The dataset Client i
$\mathscr{X}$	The Features space of FL system
$\mathscr{Y}$	The label space of FL system
$\mathscr{I}$	The sample ID space of FL system
$\|D_i\|$	The dataset size of Client i
$\xi,(X,Y)$	A random sample drawn from D_i
h_i	Hypothesis or model of Client i

$\mathscr{H}$	Global hypothesis or model with no subscripts
θ_i	Parameters of h_i
Θ_g	The parameters of global model in FL
ℓ_i	Loss function of Client i
$\mathscr{L}$	Global Loss function with no subscripts
ψ_i	The importance weight of Client i, and $\sum_{i \in k} \psi_i = 1$
Q	Number of local update steps
T	Number of communication rounds
$\eta_i^{(t)}$	The learning rate for Client i in Communication Round t
g_i	General form of characteristics of Client i depending on h_i, θ_i, D_i, etc.
$\mathscr{F}$	The overall objective to optimize of FL
U	The bound of gradient defined in Assumption 3.4.1
B	The clusterability measure defined in Definition 3.4.7
$x^{(t,E)}$	The state of x at the **E**xpecation step of Round t in Algorithm 1
$x^{(t,M)}$	The state of x at the **M**aximization step of Round t in Algorithm 1
$x^{(t,L)}$	The state of x at the **L**ocal update step of Round t in Algorithm 1

Clustering Notations

K	Number of clusters
$r_{i,k} \in \mathbb{R}^{m*K}$	The assignment matrix, $r_{i,k} = 1$ if $i \in k$ else $r_{i,k} = 0$
$i \in k$	Client i belongs to Cluster k
$\mathscr{H}_k$	Hypothesis or model of Cluster k
Θ_k	Parameters of $\mathscr{H}_k$
$\mathscr{L}_k$	Loss function of Cluster k

G_k	General form of characteristics of Cluster k depending on $\mathscr{H}_k$, Θ_k, etc.
$\mathscr{C}$	The overall objective to optimize of the clustering

CFL-CON Notations

$\mathscr{T}$	The function of the contrastive loss
μ	The coefficient of Contrastive loss
τ	The temperature of Contrastive loss

CFL-CKS Notations

$\mathscr{S}$	A penalty term to share knowledge, a.k.a. the CKS loss
λ	The coefficient of CKS loss

CFL-CON&CKS Notations

$\theta_{i,r}$	The subscript of r represents the parameters of representation layers or the backbone.
$\theta_{i,p}$	The subscript of p represents the parameters of projection layers or the head.

CAM Notations

f	The globally shared model for CAM.
n_i	The dataset size of Client i and equals $\|D_i\|$.
n	Sum of dataset size of all clients.
θ^0	Parameters of local models for updating the global model of CAM.
$c(i)$	The cluster label of Client i.
C_k	The set of clients in Cluster k.

w	Number of warmup rounds.
β	Level of function smoothness.
n_k	Number of clients of Cluster k.
s	Sample size of all clients.
s_k	Sample size of clients in Cluster k.

CHAPTER 1

INTRODUCTION

1.1 Background

1.1.1 Federated Learning

Deep learning [44] has experienced significant growth since 2015, but with the increase in data, training times have also become longer. To address this issue, Distributed Learning (DL) was proposed, which involves distributing data across multiple devices to optimize training time. However, traditional DL involves centralizing the data, which can raise concerns about privacy and communication efficiency.

To address these concerns, Federated Learning (FL) [67] was introduced in 2017 as a cutting-edge distributed or collaborative machine learning framework. FL allows for training machine learning models without requiring the data to be centralized or transferred to a central server. Instead, models are trained locally on each device, and only model updates (i.e., gradients) are transmitted and aggregated by a central

server, thereby reducing the risk of exposing raw data to other devices. FL has become increasingly important as data privacy and communication efficiency have become top priorities. Since its propose, FL has evolved into a new-generation collaborative machine learning framework with applications in a range of scenarios, including Google's Gboard on Android [67], Apple's Siri [23], computer visions [33, 37, 61], smart cities [108], finance [60], weather forecasting [13] and healthcare [59, 73, 97]. And the hierarchical structure and development history of FL to Clustered FL is demonstrated in Figure 1.1.

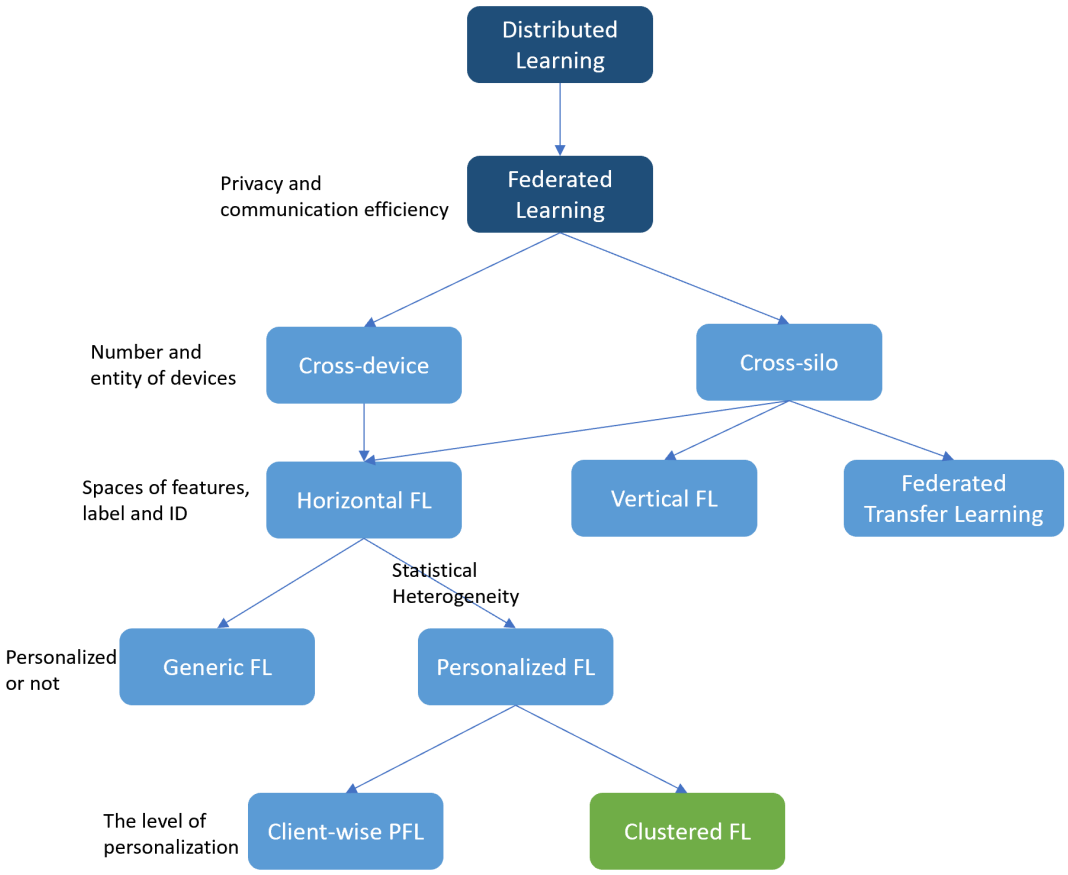

Figure 1.1: The hierarchical structure of FL to Clustered FL.

Given the complexities inherent in a distributed system, there are various applications for FL. One way to classify FL is based on the number and type of participating clients, dividing it into two categories: cross-device FL and cross-silo FL, as outlined by

Huang et al. [36]. In cross-device FL, the clients consist of smaller distributed entities such as smartphones, wearables, and edge devices, each of which likely possesses only a modest amount of local data. Consequently, successful cross-device FL typically requires the participation of a vast number of edge devices, potentially up to millions, in the training process. Conversely, in cross-silo FL, the clients are larger entities, like companies or organizations, including hospitals and banks. Here, the number of participating clients is significantly smaller, ranging from just two to a hundred, but each client is expected to be actively involved throughout the entire training process.

FL can also be categorized based on the variability in the training datasets across different clients. In the FL system with m clients, the comprehensive training dataset D can be represented as $(\mathscr{X},\mathscr{Y},\mathscr{I})$, which encompasses feature spaces, label spaces, and sample ID spaces, as per Yang et al. [102]. For cross-device FL, the ID spaces typically differ across clients, while the feature spaces and label spaces remain constant.

$$\mathscr{X}_i = \mathscr{X}_j, \mathscr{Y}_i = \mathscr{Y}_j, \mathscr{I}_i \neq \mathscr{I}_j, \forall i \neq j, \tag{1.1}$$

Which is called Horizontal FL (HFL). For cross-silo FL, the scenarios are more complex, while the Vertical FL (VFL) has the following property,

$$\mathscr{X}_i \neq \mathscr{X}_j, \mathscr{Y}_i \neq \mathscr{Y}_j, \mathscr{I}_i = \mathscr{I}_j, \forall i \neq j, \tag{1.2}$$

and Federated Transfer Learning (FTL) has the following property,

$$\mathscr{X}_i \neq \mathscr{X}_j, \mathscr{Y}_i \neq \mathscr{Y}_j, \mathscr{I}_i \neq \mathscr{I}_j, \forall i \neq j. \tag{1.3}$$

Cross-device FL is commonly referred to as HFL, while cross-silo FL can be HFL, VFL and FTL. This thesis will focus on the HFL in the setting of cross-device, named cross-device HFL, in which privacy is the top priority, and most people will be beneficial. HFL has numerous real-world applications, including Google's Gboard for Android [67],

Apple's Siri [23], computer vision tasks [33, 37, 61], smart city initiatives [108], and healthcare systems [59, 73, 97].

Cross-device HFL faces several core challenges that need to be addressed [51]. These challenges can be summarized as follows:

1. Expensive Communication: The first challenge is the cost of communication between devices. In a cross-device HFL setting, there may be millions of devices involved in the training process, and the model updates need to be transmitted and aggregated at a central server. The cost of communication can be significant, especially if the devices are geographically distributed or have limited bandwidth.

2. Systems Heterogeneity: The second challenge is the heterogeneity of the systems and devices involved in the FL process. Devices can vary in terms of hardware, operating systems, and software versions, which can create compatibility issues and affect the quality of the model updates.

3. Statistical Heterogeneity: The third challenge is the statistical heterogeneity of the data on the different devices. The data distributions and features can differ across devices, which can lead to biased or inconsistent model updates and lower model accuracy.

4. Privacy Concerns: The fourth challenge is privacy. In a cross-device HFL setting, data is distributed across multiple devices, and privacy concerns arise when personal or sensitive information is involved. To ensure data privacy, techniques such as differential privacy and secure multi-party computation can be used.

1.1.2 Clustered FL with non-IID

Addressing the inherent challenges is crucial for the successful implementation of cross-device HFL. Active research is underway to devise effective solutions that enhance

communication efficiency, ensure system compatibility, alleviate statistical heterogeneity, and maintain data privacy. HFL trains a global model across distributed clients while upholding data localization. That is, the data remain local for model training on the client side, and the server periodically averages the weights of client models to update a global model, which is then disseminated to all clients. When identical local data distributions are present across clients, a single global model suffices to cater to all clients [67]. However, in practical FL scenarios, it's more common to encounter non-identical or non-IID (non-independent and identically distributed) data distributions across clients, leading to conflicts between global and local objectives. An ideal approach in non-IID settings would be to train an individual local model per client without any interference. However, local data are often insufficient, making a global model trained on heterogeneous clients valuable as it leverages all their data. Thus, non-IID FL methods [27, 51, 109] aim to strike a balance between global consensus and local personalization. Without any assumptions about the structure among clients, a global model may be influenced by all clients' conflicts and may offer limited guidance to their local training.

This dissertation primarily focuses on addressing the challenge of statistical heterogeneity or non-IIDness, a relatively straightforward concept to simulate that has been extensively benchmarked [109]. Consequently, Personalized FL (PFL) is proposed. PFL involves models personalized to mitigate the non-IIDness of clients, as opposed to generic FL that generates a single global model for all clients.

Most existing PFL research focuses on client-wise non-IID settings that do not assume any complicated structure. For example, using Dirichlet distribution with hyperparameter α to simulate the non-IID data generation or partition across clients [35]. However, cluster-wise data is a more common scenario in real applications, such as segmenting users by demographic features, including gender, age, location, etc. Moreover, there is

a general assumption that clients with similar backgrounds are very likely to make similar decisions, thus generating data with similar distributions. Conversely, users with different backgrounds are likely to have very different actions when encountering the same scenarios. This assumption is widely applied to population-based marketing strategy and cohort-based user behavior analytics.

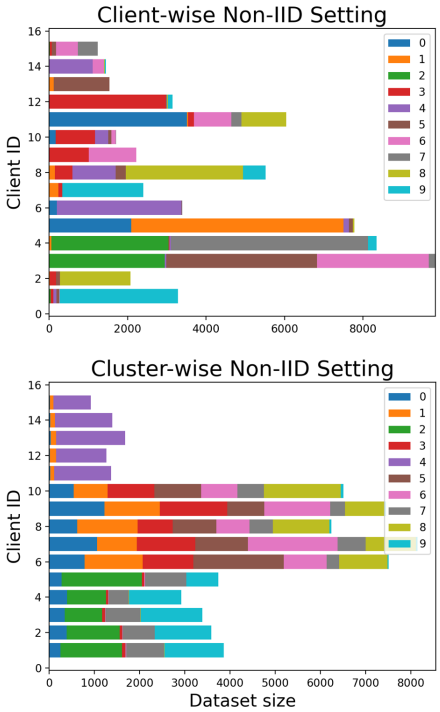

Figure 1.2: A toy example of client-wise and cluster-wise non-IID settings. Color labels represent ten classes, and the length of the bar represents the # of instances.

As outlined by [39], non-IID data can be classified into feature distribution skew, label distribution skew, concept drift, and quantity skew. However, it's noteworthy that non-IID clients in practice often exhibit rich structures that most existing FL methods have yet to explore fully. One such prevalent structure involves clusters; that is, heterogeneous clients can be grouped into several near-homogeneous clusters, each comprising clients with similar distributions. In real-world scenarios, these clusters may correlate with geographical, age, or income groups, affiliations, and so on. In this paper, we extend

the taxonomy of non-IID data by introducing another dimension: client-wise non-IID and cluster-wise non-IID. As depicted in Figure 1.2, client-wise non-IID is characterized by significant variance in label distributions across different clients, while cluster-wise non-IID exhibits a large variance across inter-cluster clients but minimal variance within the same cluster (intra-cluster clients). Generally, clustered FL performs optimally with cluster-wise non-IID data. Even in a client-wise non-IID scenario, clustered FL methods can outperform those based on a single model by leveraging multiple cluster-wise personalized models to mitigate the non-IID issue. Furthermore, clustered FL presents a competitive solution capable of balancing model personalization and generalization. In contrast, client-wise personalized FL is often susceptible to overfitting during local fine-tuning.

1.1.3 Taxonomy of Clustered FL

There are various existing clustered FL methods [30, 66, 74, 94]. However, the clusterability of clients is not well studied in the existing clustered FL methods, which usually treat clustering as an add-on component for the FedAvg framework [67]. Moreover, a few fundamental problems still need further study, such as how to represent a client and measure distance in a clustering procedure in the FL context, and how to measure the clusterability and clustering quality, which should be integrated with the learning objective of the FL system.

Choosing the appropriate metrics to describe the characteristics of a client and a cluster has been a notable challenge. For a single client, the information that can be leveraged for clustering is limited to its data and model. Owing to the privacy preservation aspect of FL, we can only utilize high-level information about the data. This may involve using the data's distribution or the discrepancies in distributions to characterize the data. On the other hand, the model, specifically its parameters or updates (gradients), are

Table 1.1: Taxonomy of Clustered FL based on its characteristics g

Taxonomy	Advantages	Disadvantages	Examples
Data	• Low dimension. • Easy to cluster.	• Unavoidable privacy concern. • Additional communication cost.	• DCFL [12] • Label-wise clustering [45]
Model	• Perserved privacy. • Almost no additional communication cost. • Almost no additional computation cost.	• High dimensional space.	• FeSEM [94] • WeCFL [62] • CFL [74]
Hybrid of data and model	• Most information used. • Low dimension.	• Privacy concern. • May difficult to cluster (loss-based). • High communication and computation cost.	• HypCluster [66] • IFCA [30] • FLSC [48]

freely available for use since they are communicated to the server. Another approach is to combine data and models to utilize their representations, such as the loss, embeddings, or prototype. With these considerations in mind, we can categorize clustered FL into three groups based on characteristics g, as shown in Table 1.1.

1.1.4 More Challenges in Clustered FL

Although clustered FL represents a step forward in dealing with non-IID data, it comes with its own set of challenges. Compared to the general non-IID assumption, the assump-

tion within clustered FL might be excessively restrictive as it disallows inter-cluster knowledge sharing and mandates that each cluster-wise model's training depends solely on a select group of clients. This contradicts the widely acknowledged strategy wherein different tasks or domains can reap benefits from sharing low-level or partial representations.

The crux of the issue lies in the gap between the assumption of "clustered data distributions" and the algorithms that are "clustering models" (represented by loss vectors or model weights). The two are not identical, and the latter is more restrictive. To put it differently, clients from different clusters can still benefit from sharing features or parameters.

Furthermore, clustered FL frequently grapples with optimization instability. The dynamics of changing models can violate the static clustering assumption, leading to imbalanced cluster assignments that impact future $\Theta_{1:K}$ and local training. More specifically:

- *Dynamic clustering:* While the clustering in clustered FL is usually based on dynamic measures, such as loss and gradients, the clustering results may change continuously during the training process, which does not align to the ground truth. Therefore achieving robust clustering is an important issue in clustered FL.

- *Lack of knowledge sharing*: When the clustering is fixed, clustered FL can be seen training K FL programs simultaneously, and this will lead to the lack of know sharing and lower the overall performance. Furthermore, solving lack of knowledge sharing could lead to less robust clustering.

- *Clustering collapse:* This is a scenario where the number of clients assigned to one cluster keeps increasing, making "the rich richer (i.e., the cluster-wise model stronger)", until the situation devolves to single-model FL, as demonstrated in

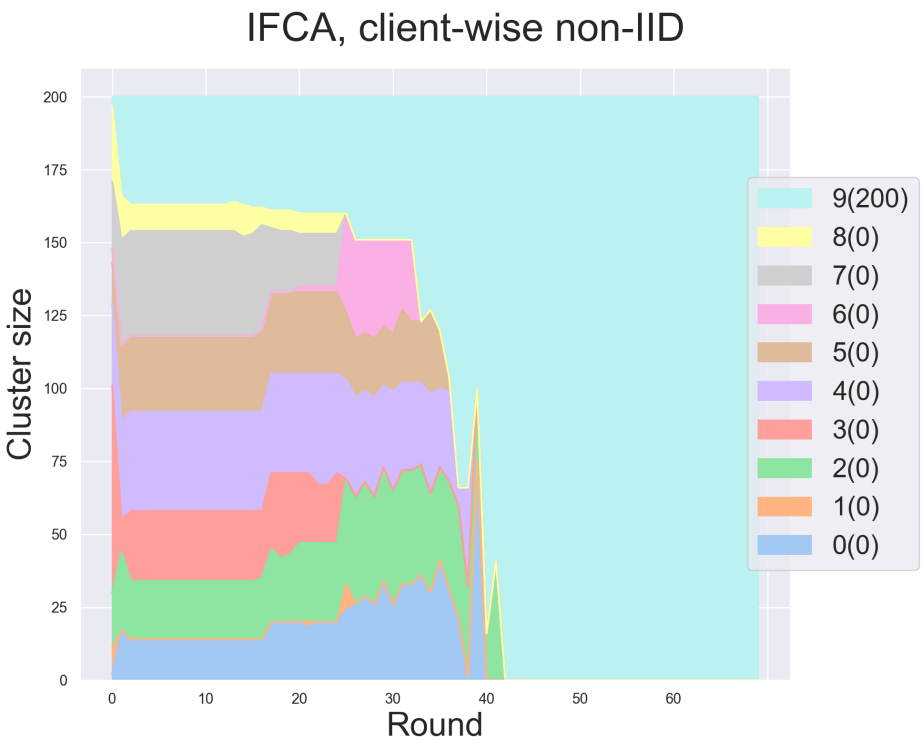

Figure 1.3: An example of clustering collapse.

Figure 1.3. This tends to occur because most clients initially learn shared features before focusing on client-specific ones.

- *Fragility to outliers:* The presence of outliers, such as malicious clients, can dominate some clusters, forcing all other benign ones into one or a few clusters.

- *Sensitivity to initialization:* The clustering process heavily relies on initial and early cluster assignments as these determine which clients' local training starts from the same model.

To address these challenges, we propose some methods in following chapters, as demonstrated in Figure 1.4.

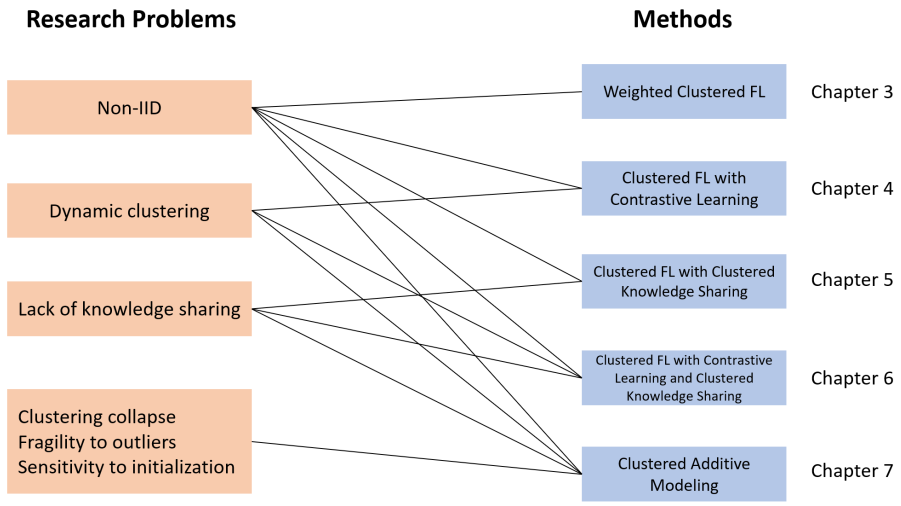

Figure 1.4: Mapping of research problems to methods.

1.2 Outline of this Thesis

Remaining of the thesis is as the following structure:

Chapter 2 This chapter provides an extensive literature review pertinent to the scope of this dissertation. It commences with the formulation of the basic FL model, explaining its inherent design and fundamental principles. Subsequently, the challenges of FL with non-IID data are presented, emphasizing the difficulties that emerge when the assumption of identical data distribution among clients is violated.

The chapter then dives into the convergence analysis of FL, elaborating on the theoretical principles that guide the learning efficiency and model stability of FL algorithms. This includes discussions on Stochastic Gradient Descent (SGD), the foundation of most FL optimization techniques, and an analysis of convergence rate and its dependencies.

The essential principles of clustering, a central concept for this dissertation, are also discussed. This includes the mathematical formulation of clustering objectives, and the notions of hard and soft clustering. The chapter further explores robust clustering, an extension of traditional clustering, aiming to enhance the resilience of clustering

outcomes against outliers.

Contrastive learning, a strategy widely employed in supervised and unsupervised learning scenarios, is also examined. Recent developments in contrastive learning, specifically in the context of FL, are underscored, emphasizing its significant role in enhancing model performance in FL environments.

Lastly, the chapter presents multi-task learning in FL, a technique that capitalizes on shared knowledge across multiple tasks or distributed clients to enhance learning efficiency and model performance. It encompasses both hard and soft parameter sharing methods and explores their relevance and applications in FL. The chapter also introduces additive modeling in FL, which involves training multiple models and aggregating their outputs for prediction.

By outlining these key concepts and their interrelations, this chapter sets the stage for subsequent discussions and analyses in this dissertation.

Chapter 3 The first work in Chapter 3 aims to take a definitive step towards resolving the challenges outlined above. We initiate this effort by revisiting current clustered FL methods, and formulating them into a comprehensive bi-level optimization problem. From this foundation, we propose a novel **We**ighted **C**lustered **F**ederated **L**earning (WeCFL) framework that signifies each client by their model parameters and gauges their distance via the Euclidean distance in parameter space. Additionally, WeCFL aligns with the concept of weighted loss in FL by incorporating weighted clients into the clustering process. These elements are amalgamated into a learning process set within a cluster-wise non-IID federated setting, where we explore the clusterability among FL clients. We also develop a new theoretical framework for conducting convergence analysis on FL with non-IID data.

The major contributions of this work can be summarized as follows:

- We introduce the first cluster-wise non-IID setting in FL, providing a more realistic

reflection of real-world scenarios.

- We restructure the clustered FL problem into a unified bi-level optimization framework and introduce a novel algorithm, WeCFL, to solve this complex optimization problem.
- We present a new theoretical framework for performing convergence analysis in clustered FL, considering a fresh clusterability measure B for our proposed unified framework.
- Our experimental simulations validate the superior performance of WeCFL, demonstrating its practical effectiveness.

Chapter 4 While the clustering in clustered FL is usually based on dynamic measures, such as gradients, achieving more robust clustering is an important issue. The second study in Chapter 4 is driven by the shared philosophy of clustering and contrastive learning, to maximize inter-cluster distance and minimize intra-cluster distance. We introduce a simple yet effective contrastive learning methodology, which can be integrated into most clustered FL frameworks, named CFL-CON. Depending on the space in which it operates, it can be modified based on the representation, or based on the parameter space. The experimental simulations substantiate the superior performance and robustness of CFL-CON, demonstrating its practical viability.

Chapter 5 The third research in Chapter 5 revisits the necessity of stable clustering in clustered FL and its pitfalls, lack of knowledge sharing across clusters. Inspired by this observation, along with frameworks such as multi-task learning, FedProx, and regularization, we introduce a straightforward yet effective supplement termed CFL-CKS, designed to facilitate knowledge sharing among clusters. This method can also be effort-

lessly integrated with the majority of current clustered FL algorithms. Subsequently, we refine it into a simple, elegant term.

The primary contributions of this research are summarized as follows:

- We propose a straightforward yet potent method based on knowledge sharing, which can supplement most clustered FL approaches.
- We conduct a theoretical analysis of the term's simplification, convergence, and interpretation, providing a comprehensive understanding of our proposed method.
- Our experimental simulation results reveal CFL-CKS's superior performance, demonstrating its practical effectiveness.

Chapter 6 While the philosophy of contrastive learning and knowledge sharing is opposite, we need to find a trade-off between these two methods. The fourth research in Chapter 6 aims to combine the contrastive learning method Chapter 4 and the clustered knowledge sharing method from Chapter 5 to further enhance the performance and robustness of clustered FL. The biggest challenge lies in the fundamentally contrasting philosophies of the two methods. If we simply add them together in a clustered FL method, their effects could cancel each other out. To overcome this challenge, we propose CFL-CON&CKS, a state-of-the-art method that applies CFL-CON to the head of the neural network to create distance, and CFL-CKS to the backbone of the neural network to facilitate knowledge sharing.

The primary contributions of this research are summarized as follows:

- We effectively combine contrastive learning and clustered knowledge sharing, leveraging the advantages of both to create CFL-CON&CKS.
- Our experimental simulation results reveal the superior performance of proposed method over baselines, demonstrating its practical effectiveness.

Chapter 7 In this chapter, in order to address issues associated with clustered FL, such as clustering collapse, vulnerability to outliers, and sensitivity to initialization, we propose a novel clustered FL model called "Clustered Additive Modeling (CAM)". In addition, we develop an efficient algorithmic framework, Fed-CAM, to tackle non-IID FL challenges with a clustering structure. It is adept at capturing more generalized non-IID structures and fostering global knowledge sharing among clients, thus overcoming key limitations of clustered FL. The main contributions of this research are summarized as follows:

- We propose a versatile, model-agnostic tool, CAM, that can enhance a wide variety of existing non-IID FL methods with any structure.
- From a theoretical perspective, we prove that Fed-CAM can achieve an asymptotic convergence rate of $O(1/\sqrt{T})$.
- Our comprehensive experimental results demonstrate that CAM provides significant enhancements to existing clustered FL methods, by effectively improving cluster balance and mitigating clustering collapse.

Chapter 8 In this chapter, we summarized the primary work of this thesis, which includes the development of a unified framework for Weighted Clustered Federated Learning (WeCFL), and three add-on enhancements: CFL-CON, CFL-CKS, and CFL-CON&CKS. We also explored potential avenues for future research. These include exploiting the structure of clustering, exploring more non-IID scenarios, tackling practical problems in application, and integrating with Large Language Models (LLMs), among others.

Appendix A This appendix covers various general experimental settings. Initially, it provides a thorough introduction of benchmark datasets, including Fashion-MNIST,

CIFAR-10, PathMNIST, and TissueMNIST, followed by an in-depth description of all four non-IID partitioning methods, which include two cluster-wise and two client-wise non-IID variants. Subsequently, it outlines the specific structures of the Convolutional Neural Network (CNN) models utilized for all four datasets.

CHAPTER 2

RELATED WORK

2.1 Formulation of FL

As demonstrated in Figure 1.1, this thesis will focus on the stream of clustered FL, which can originate from cross-device FL. A cross-device FL system usually includes m clients and one server. For Client i, its loss function can be defined as below:

$$\ell_i = \mathbb{E}_{\xi=(X,Y)\sim D_i} \ell_i(h_i(\theta_i, X), Y), \tag{2.1}$$

in which ξ or (X,Y) is the sampled instance from the dataset of Client i, D_i, and ℓ_i, h_i, θ_i represent the loss function, model structure or hypothesis, model parameter of Client i, respectively.

Then it is natural to aggregate the loss function of all clients to form the loss function of FL. And the weight of Client i in aggregation is usually defined as

$$\psi_i = \frac{1}{m}, \tag{2.2}$$

or by their dataset size,

$$\psi_i = \frac{|D_i|}{\sum_{j=1}^{m} |D_j|}, \tag{2.3}$$

while ψ_i has to satisfy,

$$\sum_{i=1}^{m} \psi_i = 1. \tag{2.4}$$

There are also some other choices of ψ or client sampling probabilities, including [26, 78, 90], which will not be addressed in this dissertation. Then the loss function or objective function to minimize FL $\mathscr{F}$ can be defined as below,

$$\underset{\Theta_g}{\text{minimize}} \mathscr{F} = \sum_{i=1}^{m} \psi_i \ell_i(\Theta_g, D_i), \tag{2.5}$$

in which each client shares the same model and model parameters Θ_g. Depending on the algorithm of vanilla FL, FedAvg [67], Θ_g is aggregated by the parameter of each client in every communication round as follows,

$$\Theta_g = \sum_{i=1}^{m} \psi_i \theta_i. \tag{2.6}$$

2.2 FL with Non-IID

FL with non-IID aims to tackle statistical heterogeneity across clients. FedAvg[67] is designed for the IID setting, so it suffers from client drift and slow convergence with non-IID clients [39]. To address this challenge, FedDANE [52] proposed a federated Newton-type optimization method by adapting a method for classical distributed optimization, i.e., DANE, to the FL setting. Instead of synchronizing all clients' models to be the same global model periodically, FedProx [51] only adds a proximal term to the local training objective that discourages the local model from drifting away from the global model and thus preserves the heterogeneity. [72] applies adaptive learning rates to clients and [38] conducts attention-based adaptive weighting to aggregate clients' models. [53]

studies the convergence of the FedAvg in non-IID scenarios. Recent work also studies client-wise personalized FL [17, 20, 25, 77, 79, 82, 83], which aim to address the non-IID challenge by training a personalized model per client with the help of the shared global model. FedICON [84] addresses the test-time shift problem, which refers to intra-client heterogeneity during test phase. Their objectives focus on training local models rather than the server-side model.

Cluster-wise PFL also known as Clustered FL, assumes that non-IID clients can be partitioned into several groups and clients in each group share a cluster-wise model. It jointly optimizes the cluster assignments and the clusters' models. K-means-based methods [94] assign clusters to clients according to their model parameters' distance. CFL [74] divides clients into two partitions based on the cosine similarity between client gradients and then checks whether a partition is congruent according to the gradient norm. IFCA [30] and HypCluster [66] assign to each client the cluster whose model achieves the minimum loss on the client's data. Few-shot clustering has been introduced to clustered FL by [5, 21]. FedP2P [16] allows communication between clients in the same cluster. [88] uses cluster-based contexts to enhance the fine-tuning of personalized FL models. [62] proposes the first cluster-wise non-IID setting and a bi-level optimization framework unifying most clustered FL methods. [64] proposes a general model-agnostic method called clustered additive modeling (CAM) to enhance existing clustered FL methods.

Client-wise PFL assumes that each client's data distribution is unique, necessitating personalized models on each device. A straightforward PFL method learns a global model at the server while conducting local fine-tuning on each client [15, 25]. Ditto [50] proposes a bi-level optimization framework for PFL that includes a regularization term to constrain the distance between local and global models. The Model-Agnostic

Meta-Learning (MAML) framework is also investigated for personalizing clients [25]. One study [82] uses Moreau envelopes as clients' regularized loss functions to optimize a bi-level problem for PFL. FedRep [17] learns a globally shared representation and a locally personalized head for each client. Research by [11, 77] aims to train a global hyper-network or meta-learner, which is then sent to clients for local optimization. SCAFFOLD [40] learns personalized control variates that correct the local model as needed. Layer-wise personalization [3, 55] and representation-wise personalization [85] are two simple yet effective PFL solutions. Hermes [46], and LotterFL [47] are two PFL methods considering communication efficiency for mobile clients. SFL [10] uses personalization to address FL problems on the graph. Work [105] proposes a novel personalized federated recommendation framework called PFedRec. Work [98] focuses on disentangling global knowledge and personal knowledge using a novel federated dual variational autoencoder (FedDVA).

2.3 Convergence Analysis of FL

There are indeed few studies on the convergence analysis of Clustered FL on non-IID data. However, many studies focus on the convergence analysis of FL on non-IID data. These works often build upon the convergence analysis of local stochastic gradient descent (SGD) [41, 81], as most FL algorithms employ SGD for optimization. Local SGD differs from FedAvg in terms of local update epochs and specific settings, such as non-IID, stragglers, and privacy attacks. The convergence analysis of FL is usually based on the SGD convergence analysis framework. In work by Li et al. [53], the convergence of FedAvg on non-IID data and partial participation is analyzed in detail. The convergence rate is $O(\frac{1}{T})$, where T is the number of communication rounds. The study also discusses the impact of some hyperparameters, such as local epochs. A guide by Wang et al. [89] provides recommendations and guidelines on formulating, designing, evaluating, and

analyzing FL optimization algorithms, with a separate section dedicated to convergence analysis. Some recent works [50, 96] model client-wise Personalized Federated Learning (PFL) tasks using a bi-level optimization framework and then conduct convergence analysis. Although these works focus on PFL rather than Clustered FL, they provide valuable insights into the convergence behavior of FL algorithms under non-IID settings. Research by [62] could extend these methodologies and insights to the convergence analysis of Clustered FL on non-IID data, paving the way for a deeper understanding of Clustered FL's convergence properties in real-world scenarios.

2.4 Robust Clustering

The objective of clustering is to group similar objects together and separate dissimilar objects into distinct clusters. This goal can be achieved by minimizing intra-cluster distances while maximizing inter-cluster distances. For a typical clustering problem, its objective $\mathscr{C}$ can be defined as follows,

$$\underset{r_{i,k}}{\text{minimize}} \mathscr{C} : \sum_{k=1}^{K} \sum_{i=1}^{m} r_{i,k} d(g_i, G_k), \tag{2.7}$$

where g_i represents the m data points, G_k represents the cluster centroids (with a total of K), and d is the distance function. This function can be an L2-norm, cosine similarity, or a density distance like LOF [6], depending on the scenario. The variable to optimize, $r_{i,k}$, is the assignment matrix. In hard clustering, $r_{i,k}$ can only be 0 or 1, while in soft clustering, it can represent probabilities.

If each data point is assigned a weight or importance ψ, we get a weighted clustering problem, which can be formulated as:

$$\underset{r_{i,k}}{\text{minimize}} \mathscr{C} : \frac{1}{\sum_{j=1}^{m} \psi_j} \sum_{k=1}^{K} \sum_{i=1}^{m} r_{i,k} \psi_i d(g_i, G_k). \tag{2.8}$$

Since $r_{i,k}$ can be seen as a latent variable, Expectation Maximization (EM) algorithms [19] or k-means [65] are often used to solve the clustering problem. To avoid the potential pitfalls of poor clusterings found by the standard k-means algorithm, k-means++ [4] is a better initialization method that is often employed.

Robust clustering aims to enhance the robustness of clustering results against outliers [29]. Numerous works have been conducted in this area, including [22], [69]. Vanilla robust clustering methods include mixture modeling [100] and trimming approach [28]. Recently, a number of works in robust clustering have been studied by [2, 18, 28, 31, 100, 101]. These methods address various challenges in robust clustering, such as outlier detection, similarity metrics, and noise handling. The work [8] researches K-means with the bootstrap of median-of-means (MOM). The MOM estimator can mitigate the influence of outliers, whereas the estimator of mean is not good at addressing outliers. The bootstrap of MOM (bMOM) enhances the robustness against outliers and thus achieves a better breakdown point, which is a measure to quantify the toleration of outliers. Then [95] uses bMOM to create more robust clustering in FL to against outliers. [63] uses a robust density-based clustering method Local outlier factor (LOF) to address model poisoning issues in FL. In summary, robust clustering aims to improve the clustering results by increasing the tolerance to outliers and noise. A variety of methods have been proposed to achieve this goal, such as mixture modeling, trimming approaches, and the bootstrap of MOM. These techniques have demonstrated their effectiveness in handling outliers and providing more accurate and robust clustering results in various scenarios.

2.5 Contrastive Learning

Contrastive learning shares a similar philosophy with Triplet loss [75, 76], as they both have definitions of anchor, negative, and positive instances. In recent years, contrastive

learning has been widely applied in supervised [42] and unsupervised learning, achieving state-of-the-art performance in the unsupervised training of deep image models [56, 93] and graph models [57, 58, 107]. Numerous works focus on learning an encoder that pulls the embeddings of the same sample closer and pushes those of different samples apart [14, 34, 104, 106]. The work of [42] extends contrastive learning from self-supervised settings to fully supervised settings, enabling better exploitation of label information with contrastive learning. This advancement has provided new insights into how contrastive learning can be used more effectively across various learning scenarios. Furthermore, some researchers have incorporated contrastive learning into FL to assist local training in achieving higher model performance [49, 68, 86, 106]. These studies have explored the potential benefits of combining contrastive learning with FL in the context of FL, leading to more efficient and robust solutions. In summary, contrastive learning has demonstrated significant success and versatility in various learning settings, from unsupervised and supervised to FL. By integrating contrastive learning into FL, researchers have further expanded the possibilities for improving model performance.

2.6 Multi-task Learning in FL

Multi-task learning and FL both aim to learn shared knowledge from multiple tasks or distributed clients. Integrating multi-task learning and FL can enable a more effective learning process and enhance performance by leveraging the shared knowledge across tasks or clients. Hard parameter sharing [9] and soft parameter sharing [24, 103] are the most commonly used methods in multi-task learning to share knowledge across tasks. It is natural to combine multi-task learning and FL together, as they both focus on learning from multiple sources and have similar objectives. MOCHA [80] uses distributed multi-task learning to address the non-IID challenge in FL. High communication cost, stragglers, and fault tolerance are also considered, both theoretically and experimentally.

MOCHA demonstrates that combining the principles of multi-task learning with FL can lead to improved performance and increased robustness in a federated setting. Similarly, Clustered Federated Learning (CFL) [74] employs the Federated Multi-Task Learning (FMTL) framework to exploit the relationships across clients and group clients together based on their data distributions. By leveraging the shared knowledge among clients and addressing non-IID challenges, CFL can achieve better performance and robustness compared to traditional FL methods. In summary, combining multi-task learning and FL can enhance the overall learning process by leveraging shared knowledge across tasks or clients. Various methods have been proposed to integrate these two learning paradigms, such as MOCHA and CFL, demonstrating the potential of this combination in addressing the non-IID challenge and improving the performance of FL systems.

2.7 Additive modeling in FL

Additive FL trains multiple models and adds their outputs together as its prediction. It was introduced to FL very recently. In the FL setting, how to define and choose two models are varied. To tackle the non-IID challenge, [66] proposed a model interpolation method by adding the global cluster model and a local model in additive modeling. Federated residual learning [1] proposed an FL algorithm to train an additive model for regression tasks. [70] applies additive modeling to combining the outputs of a shared model and a local model in a partial model personalization framework, which only shares part of the model parameters while preserving the rest for personalization. [54] proposed additive matrix factorization to solve federated recommendation task. However, additive modeling has not been studied for clustered FL.

CHAPTER 3

A UNIFIED FRAMEWORK OF CLUSTERED FEDERATED LEARNING

3.1 A New Perspective for Clustered FL

Existing clustered FL methods focus on the learning process in a federated setting. Thus, the clustering components are an add-on part of the overall learning process in the FL system. We will rethink the clustered FL from a clustering perspective while considering the FL contexts. To conduct clustering in the FL system, several major challenges need to be resolved.

- Challenge 1: How to represent an FL client in an instance or point in clustering?
- Challenge 2: How to measure the distance or similarity for FL clients?
- Challenge 3: How to evaluate the quality of clustering by considering the FL's objective?

- Challenge 4: How to choose a clustering algorithm to be integrated with the FL?

For Challenge 1, existing Clustered FL methods usually use **client-specific models** to represent the client in a clustering. Using model parameters will be a straightforward solution that is consistent with the setting of FL. An alternative option is to use technology, e.g., federated generative adversarial learning [71] and federated representation learning [49, 106], to transform the client-specific dataset or distribution into a vector to represent the client. However, the operation of embedding datasets usually causes extra privacy concerns for end-users. Thus it will be a controversial topic in practice.

For Challenge 2, the selection of distance and similarity metrics is highly reliant on the selection of client-specific representation - the solution of Challenge 1. With a given representation vector, some clustered FL reuse the classical distance and similarity measurements, such as Euclidean distance [94], cosine similarity [74] and KL divergence [45]. Moreover, a key issue for this challenge is to ensure **clusterablity** for the clients with the given representation space and distance metric.

For Challenge 3, a basic rule of evaluation is that a "good" clustering result should also lead to a "good" learning result for the FL system. The widely used objective function of FL is a weighted sum loss of all clients, e.g., FedAvg[67]. Therefore, the **client-specific weights** are important indicators to design clustering evaluation criteria in the FL context.

For Challenge 4, selecting clustering algorithms depends on the design of client-specific representation, distance metrics, and evaluation criteria. Due to the complexity of the FL system requiring efficient communication and computation, a **simple clustering algorithm** is a preferred choice, such as K-means [94] or hierarchical clustering [7].

3.2 A Unified Framework

An FL system is usually composed of m clients where each client needs to train an intelligent task using its own dataset D_i.

In particular, we can reformulate HypCluster [66] and IFCA [30] as a bi-level optimization problem:

$$\underset{\{\mathscr{H}_k\}}{\text{minimize}} \ \frac{1}{m}\sum_{k=1}^{K}\sum_{i=1}^{m} r_{i,k}\mathscr{L}(\mathscr{H}_k, D_i) \tag{3.1a}$$

$$s.t.\ \{r_{i,k}\} = \underset{\{r_{i,k}\}}{\text{argmin}}\, \mathscr{L}(\mathscr{H}_k, D_i). \tag{3.1b}$$

We also reformulate the FeSEM [94] from a loss function with regularization into a bi-level optimization framework.

$$\underset{\{\Theta_k\}}{\text{minimize}} \ \frac{1}{m}\sum_{k=1}^{K}\sum_{i=1}^{m} r_{i,k}\mathscr{L}(\Theta_k, D_i) \tag{3.2a}$$

$$s.t.\ \{r_{i,k}\} = \underset{\{r_{i,k}\}}{\text{argmin}}\, \frac{1}{m}\sum_{k=1}^{K}\sum_{i=1}^{m} r_{i,k}\|\theta_i - \Theta_k\|_2^2. \tag{3.2b}$$

where $\Theta_k = \frac{1}{\sum_{i\in k} r_{i,k}}\sum_{i\in k}\theta_i$ is the centroid of the cluster k.

As mentioned in Section 3.1, the client-wise importance weights are important indicators for clustering to be consistent with the loss function in FL. Therefore, we design a unified framework of the objective function for the clustered FL problem, which is a bi-level optimization problem. The previous works could be special cases of our proposed framework.

$$\underset{\{\Theta_k\}}{\text{minimize}} \ \mathscr{F} : \frac{1}{\sum_{j=1}^{m}\psi_j}\sum_{k=1}^{K}\sum_{i=1}^{m} r_{i,k}\psi_i\mathscr{L}_k(D_i) \tag{3.3a}$$

$$s.t.\ \{r_{i,k}\} = \underset{\{r_{i,k}\}}{\text{argmin}}\, \mathscr{C} : \frac{1}{\sum_{j=1}^{m}\psi_j}\sum_{k=1}^{K}\sum_{i=1}^{m} r_{i,k}\psi_i d(g_i, G_k). \tag{3.3b}$$

3.3 Algorithm

Based on our proposed unified framework above, the upper-level objective 3.3a is an FL problem that is usually optimized by the FedAvg algorithm, whereas the lower-level

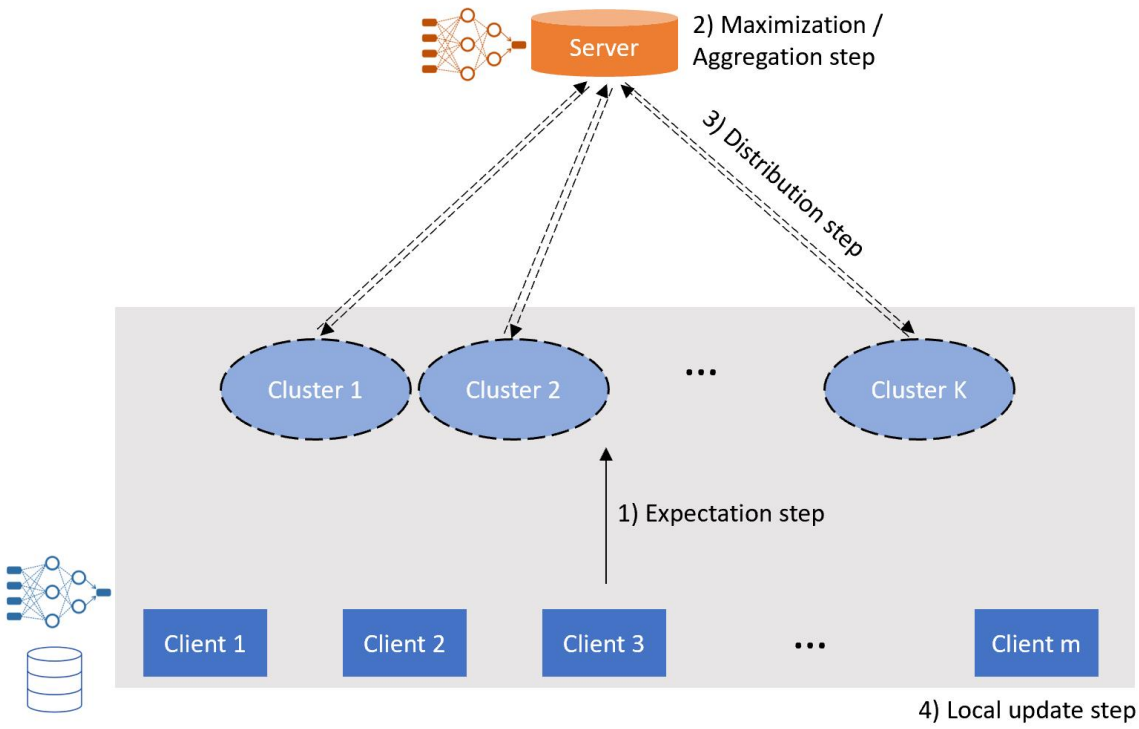

Figure 3.1: The framework and processes of WeCFL.

objective 3.3b is a clustering problem that is usually optimized by the EM algorithm [19]. It is a straightforward solution to combine these two algorithms into one and then iteratively solve the objective. So Algorithm 1 called weighted clustered federated learning (WeCFL) is proposed to solve this bi-level optimization problem, which is simple but effective. The framework and algorithm are demonstrated in Figure 3.1.

Algorithm 1: Weighted **C**lustered **FL (WeCFL)**

1: **Input:** K, $\{D_1, D_2, \ldots, D_m\}$, $\{\ell_1, \ell_2, \ldots, \ell_m\}$

2: **Initialize:** Randomly initialize $\{\mathscr{H}_1, \mathscr{H}_2, \ldots, \mathscr{H}_K\}$

3: **repeat**

4: **Expectation step:** Assign Client i to Cluster k by

$$k = \underset{k}{\operatorname{argmin}}\, \psi_i d(g_i, G_k).$$

5: **Maximization / Aggregation step:** Compute cluster center $\mathscr{H}_k$ by minimizing

$$\mathscr{C} = \frac{1}{\sum_{j=1}^{m} \psi_j} \sum_{k=1}^{K} \sum_{i=1}^{m} r_{i,k} \psi_i d(g_i, G_k).$$

6: **Distribution step:** Send $\mathscr{H}_k$ to clients in Cluster k.

7: **Local update step:** Run Gradient Descent Q steps using local data D_i to minimize

$$\mathscr{F} = \frac{1}{\sum_{j=1}^{m} \psi_j} \sum_{k=1}^{K} \sum_{i=1}^{m} r_{i,k} \psi_i \mathscr{L}_k(\mathscr{H}_k, D_i).$$

8: **until** convergence condition satisfied

9: **Output:** $\{r_{i,k}\}, \{\mathscr{H}_1, \mathscr{H}_2, \ldots, \mathscr{H}_K\}$.

Algorithm 1 illustrates the procedure of WeCFL to solve the proposed bi-level optimization problem in Eq. 3.3 by four main steps in every iteration. The first two steps correspond to an EM algorithm solving the clustering problem: the E-step assigns clients to the nearest cluster, and the M-step calculates the centroid of each cluster, which is equivalent to the model aggregation step of FedAvg [67]. Unlike normal clustering, here the representation of each client keeps being updated by the following two steps: the server broadcasts the aggregated model for each cluster to its clients; once the cluster model is received, each client applies local updates to it by minimizing the loss for its local data D_i and the resulting local model is the client's new representation for the next iteration.

3.4 Convergence Analysis

For the convergence of optimization problem 3.1, which is used by HypCluster [66] and IFCA [30], the convergence is easy to analyze. We separate the algorithm into two steps: the assignment step, and the local update step. In the assignment step, it is always best to assign the least loss function to the clients, so the Objective 3.1a will not increase. In the local update step, which uses a gradient descent algorithm, by choosing the proper learning rate under Assumption 3.4.9, the Objective 3.1a will not increase either. Moreover, the Objective 3.1a will monotonously decrease, proving convergence.

For the convergence of the optimization problem in Eq. 3.2 and 3.3, we consider a special case of Problem in Eq. 3.3 that also covers Problem in Eq. 3.2, in which the client representation g is the parameter of the hypothesis of Client i, and the distance function is Euclidean norm square $\|\cdot\|_2^2$. Then the objective function to minimize is as follows:

$$\begin{aligned} &\underset{\{\Theta_k\}}{\text{minimize}}\ \mathscr{F}: \frac{1}{\sum_{j=1}^{m}\psi_j}\sum_{k=1}^{K}\sum_{i=1}^{m} r_{i,k}\psi_i \mathscr{L}(\Theta_k, D_i) \\ &\quad s.t.\ \{r_{i,k}\} = \underset{\{r_{i,k}\}}{\operatorname{argmin}}\,\mathscr{C}: \frac{1}{\sum_{j=1}^{m}\psi_j}\sum_{k=1}^{K}\sum_{i=1}^{m} r_{i,k}\psi_i \|\theta_i - \Theta_k\|_2^2. \end{aligned} \tag{3.4}$$

3.4.1 Convergence Analysis of $\mathscr{C}$

To analyze the convergence of the optimization problem 3.4 above, both $\mathscr{C}$ and $\mathscr{F}$ should be considered. We will first analyze the clustering objective $\mathscr{C}$:

Assumption 3.4.1. (Unbiased gradient estimator and bounded gradients). The expectation of stochastic gradient $\nabla l(\theta_i, \xi)$ is an unbiased estimator of the local gradient for each client:

$$\mathbb{E}_{\xi_i \sim D_i}[\nabla l(\theta_i, \xi)] = \nabla l(\theta_i).$$

and expectation of L2 norm of $\nabla l(\theta_i, \xi)$ is bounded by a constant U:

$$\mathbb{E}_{\xi_i \sim D_i}[\|\nabla l(\theta_i, \xi)\|_2] \leq U.$$

It is also applied for $\mathscr{L}$ and $\{\mathscr{L}_k\}$.

Lemma 3.4.2. *In the Expecation step of communication round t+1, fix θ, Θ, and assign $r_{i,k} = 1$ if*

$$k = \underset{k}{\operatorname{argmin}} \|\theta_i - \Theta_k\|_2^2,$$

then we can prove that:

$$\mathscr{C}^{(t+1,E)} \leq \mathscr{C}^{(t,L)}. \tag{3.5}$$

Proof. $r_{i,k}^{(t+1)} = 1$ is to find the right k for Client i to minimize $\|\theta_i - \Theta_k\|_2$, which means to find the shortest Euclidean distance from each $\Theta_1, \Theta_2, \ldots, \Theta_K$ to θ_i, so for every i,

$$\psi_i \|\theta_i - \Theta_k^{(t+1,E)}\|_2^2 \leq \psi_i \|\theta_i - \Theta_k^{(t,L)}\|_2^2,$$

then sum it with from $i = 1$ to m, we can easily get:

$$\mathscr{C}^{(t+1,E)} \leq \mathscr{C}^{(t,L)}.$$

■

Lemma 3.4.3. *In the Maximization step of communication round t, fix r, θ, define:*

$$\Theta_k^{(t,M)} = \sum_{i \in k} \frac{\psi_i}{\sum_{j \in k} \psi_j} \theta_i, \tag{3.6}$$

then we can prove that:

$$\mathscr{C}^{(t,M)} \leq \mathscr{C}^{(t,E)}. \tag{3.7}$$

Proof. For an arbitrary Client i in Cluster k, the loss square is :

$$\begin{aligned}
&\psi_i \|\theta_i - \Theta_k^{(t,E)}\|_2^2 \\
=&\psi_i \|\theta_i - \Theta_k^{(t,M)} + \Theta_k^{(t,M)} - \Theta_k^{(t,E)}\|_2^2 \\
=&\psi_i \|\theta_i - \Theta_k^{(t,M)}\|_2^2 + \psi_i \|\Theta_k^{(t,M)} - \Theta_k^{(t,E)}\|_2^2 \\
&+ 2\psi_i \langle \theta_i - \sum_{i \in k} \frac{\psi_i}{\sum_{j \in k} \psi_j} \theta_i, \sum_{i \in k} \frac{\psi_i}{\sum_{j \in k} \psi_j} \theta_i - \Theta_k^{(t,E)} \rangle,
\end{aligned} \tag{3.8}$$

then sum all the clients in Cluster k together:

$$
\begin{aligned}
&\sum_{i\in k}\psi_i\|\theta_i-\Theta_k^{(t,E)}\|_2^2\\
=&\sum_{i\in k}\psi_i\|\theta_i-\Theta_k^{(t,M)}\|_2^2+\sum_{i\in k}\psi_i\|\Theta_k^{(t,M)}-\Theta_k^{(t,E)}\|_2^2\\
&+2\langle\sum_{i\in k}\psi_i\theta_i-\sum_{i\in k}\psi_i\sum_{i\in k}\frac{\psi_i}{\sum_{j\in k}\psi_j}\theta_i,\sum_{i\in k}\frac{\psi_i}{\sum_{j\in k}\psi_j}\theta_i-\Theta_k^{(t,E)}\rangle\\
=&\sum_{i\in k}\psi_i\|\theta_i-\Theta_k^{(t,M)}\|_2^2+\sum_{i\in k}\psi_i\|\Theta_k^{(t,M)}-\Theta_k^{(t,E)}\|_2^2.
\end{aligned}
\tag{3.9}
$$

So sum all loss functions of all clusters, we can get:

$$
\mathscr{C}^{(t,M)}-\mathscr{C}^{(t,E)}=-\frac{1}{\sum_{j=1}^m\psi_j}\sum_{k=1}^K\sum_{i\in k}\psi_i\|\Theta_k^{(t,M)}-\Theta_k^{(t,E)}\|_2^2\le 0. \tag{3.10}
$$

■

Lemma 3.4.4. *Under Assumption 3.4.1, in the Distribution step of communicaiton round t+1, we get $\theta_{i\in k}=\Theta_k$. In the Local update step of communicaiton round t+1, fix $r;\Theta$, after Q steps, define:*

$$
\theta_i^1=\theta_i^0-\eta_i^{(t)}*\nabla l_i(\theta_i^0,D_i), \tag{3.11}
$$

so

$$
\theta_i^{(n+1)}=\Theta_k-\eta_i^{(t)}\nabla l_i(\theta_i^0,D_i)-\cdots-\eta_i^{(t)}\nabla l_i(\theta_i^{Q-1},D_i). \tag{3.12}
$$

If $\eta_i^{(t)}\le\frac{\|\theta_i^{(t)}-\Theta_k\|_2}{QU}$, we can prove that:

$$
\mathscr{C}^{(t,L)}\le\mathscr{C}^{(t,M)}. \tag{3.13}
$$

Proof.

$$
\begin{aligned}
&\|\theta_i^{(n+1)}-\Theta_k\|_2\\
=&\|\Theta_k-\eta_i^{(t)}\nabla l_i(\theta_i^0,D_i)-\cdots-\eta_i^{(t)}\nabla l_i(\theta_i^{Q-1},D_i)-\Theta_k\|_2\\
=&\eta_i^{(t)}\|\nabla l_i(\theta_i^0,D_i)+\cdots+\nabla l_i(\theta_i^{Q-1},D_i)\|_2.
\end{aligned}
\tag{3.14}
$$

So if we want to:

$$
\begin{aligned}
&\|\theta_i^{(n+1)} - \Theta_k\|_2^2 \\
=&\eta_i^{(t)} \|\nabla l_i(\theta_i^0, D_i) + \cdots + \nabla l_i(\theta_i^{Q-1}, D_i)\|_2^2 \\
\leq&(\eta_i^{(t)} QU)^2 \\
\leq&\|\theta_i^{(t)} - \Theta_k\|_2^2,
\end{aligned}
\tag{3.15}
$$

η should satisfy:

$$
\eta_i^{(t)} \leq \frac{\|\theta_i^{(t)} - \Theta_k\|_2}{QU}. \tag{3.16}
$$

In particular, if $\|\theta_i^{(t)} - \Theta_k\| = 0$, then $\eta_i^{(t)} = 0$, θ_i does not change, or if $\|\nabla l_i\|$ equals 0, it means θ_i has been to the local minimum. ■

Theorem 3.4.5. *(Convergence of clustering problem $\mathscr{C}$). Under Assumption 3.4.1, for arbitrary communication round t, if $\eta_i^{(t)} \leq \frac{\|\theta_i^{(t)} - \Theta_k\|}{QU}$, $\mathscr{C}$ converges.*

Remark 3.4.6. (Clustering stability guarantee). It is important to make sure $\mathscr{C}$ converges, which means the clustering results are stable. We also conduct detailed experimental analysis on clustering in Section 3.6.3.

Proof. In communication round t+1, use Lemma 3.4.2 3.4.3 3.4.4, it is easy to get:

$$
\mathscr{C}^{(t+1,L)} \leq \mathscr{C}^{(t,L)}, \tag{3.17}
$$

which also means $\mathscr{C}^{(t+1)} \leq \mathscr{C}^{(t)}$, because $\mathscr{C}$ must be non-negative, and there are finite steps for this minimization, then according to monotone convergence theorem for sequences, $\{\mathscr{C}^{(t)}\}$ converges with finite iterations, which means for an arbitrary ϵ, we can find a specific N, for any $n > N, \mathscr{C}^{(t)} - \mathscr{C}^{\star} < \epsilon$. ■

3.4.2 Convergence Analysis of $\mathscr{F}$

Definition 3.4.7. (Clusterability measure). For arbitrary Client i in Cluster k, if its gradient obeys:

$$\frac{\|\sum_{p\in k}\frac{\psi_p\nabla\ell(\theta_p,D_p)}{\sum_{z\in k}\psi_z}-\nabla\ell(\theta_i,D_i)\|_2}{\|\sum_{p\in k}\frac{\psi_p\nabla\ell(\theta_p,D_p)}{\sum_{z\in k}\psi_z}\|_2}\leq B, \tag{3.18}$$

we define the clusterability of Cluster k to be B. If $B=0$, it means the same data distribution among clients. The larger B, the less clusterability of Cluster k. It will even lead to divergence if B is too large.

Assumption 3.4.8. (Convexity). Each loss function ℓ_i or $\mathscr{L}$ is convex. Then we will have

$$\ell(y)\geq\ell(x)+\langle\nabla\ell(x),y-x\rangle. \tag{3.19}$$

Assumption 3.4.9. (Lipschitz Smooth). Each loss function ℓ_i or $\mathscr{L}$ is β-smooth. Then we will have

$$\ell(y)\leq\ell(x)+\langle\nabla\ell(x),y-x\rangle+\frac{\beta}{2}\|y-x\|_2^2. \tag{3.20}$$

Assumption 3.4.10. (Bounded gradient variance). The variance of stochastic gradient $\nabla\ell(\theta_i,\xi)$ is bounded by σ^2:

$$\begin{aligned}&\mathbb{E}_{\xi_i\sim D_i}[\|\nabla\ell(\theta_i,\xi)-\nabla\ell(\theta_i)\|_2^2]\\=&\mathbb{E}[\|\nabla\ell(\theta_i,\xi)\|_2^2]-\|\nabla\ell(\theta_i)\|_2^2\leq\sigma^2.\end{aligned} \tag{3.21}$$

It is also applied for L.

Lemma 3.4.11. *Under Assumption 3.4.1 and 3.4.8, from the Expectation step to the Maximization step in arbitrary communication round,* $\mathscr{F}^M\leq\mathscr{F}^E+\eta BQU^2$.

Proof.

$$\mathscr{F}^M-\mathscr{F}^E=\frac{1}{\sum_{j=1}^m\psi_j}\sum_{k=1}^K\sum_{i\in k}\psi_i(\mathscr{L}(\Theta_k^M,D_i)-\mathscr{L}(\theta_i,D_i)), \tag{3.22}$$

in which

$$\Theta_k^M = \sum_{p\in k} \frac{\psi_p}{\sum_{z\in k}\psi_z}\theta_p. \tag{3.23}$$

According to Assumption 3.4.8 and Equation 3.19, for arbitrary cluster, using Cauchy,ÄìSchwarz for Equation 3.26 and Assumption 3.4.1 for Equation 3.27, we have

$$\sum_{i\in k}\psi_i(\mathscr{L}(\sum_{p\in k}\frac{\psi_p}{\sum_{z\in k}\psi_z}\theta_p, D_i) - \mathscr{L}(\theta_i, D_i)) \tag{3.24}$$

$$\leq \sum_{i\in k}\psi_i(\langle \nabla\mathscr{L}(\Theta_k^M, D_i), \sum_{p\in k}\frac{\psi_p}{\sum_{z\in k}\psi_z}\theta_p - \theta_i\rangle) \tag{3.25}$$

$$\leq \sum_{i\in k}\psi_i \|\nabla\mathscr{L}(\Theta_k^M, D_i)\|_2 \cdot \|\sum_{p\in k}\frac{\psi_p}{\sum_{z\in k}\psi_z}\theta_p - \theta_i\|_2 \tag{3.26}$$

$$\leq \sum_{i\in k}\psi_i U\|\sum_{p\in k}\frac{\psi_p}{\sum_{z\in k}\psi_z}\theta_p - \theta_i\|_2. \tag{3.27}$$

According to Equation 3.12,

$$\theta_i = \Theta_k - \eta\nabla\ell_i(\theta_i^0, D_i) - \cdots - \eta\nabla\ell_i(\theta_i^{Q-1}, D_i). \tag{3.28}$$

So we can get below inequality depending on Definition 3.4.7:

$$\sum_{i\in k}\psi_i U\|\sum_{p\in k}\frac{\psi_p}{\sum_{z\in k}\psi_z}\theta_p - \theta_i\|_2 \leq \sum_{i\in k}\psi_i \eta BQU^2. \tag{3.29}$$

Finally, we can get:

$$\mathscr{F}^M \leq \mathscr{F}^E + \eta BQU^2. \tag{3.30}$$

■

Lemma 3.4.12. *Under Assumption 3.4.9 and 3.4.10, from the Maximization step to the Local update step in arbitrary communication round, we have*

$$\begin{aligned}&\mathbb{E}[\mathscr{F}^L] - \mathscr{F}^M \\ \leq &\frac{1}{\sum_{j=1}^m \psi_j}\sum_{k=1}^K\sum_{i\in k}\psi_i\sum_{q=0}^{Q-1}((\frac{\beta\eta_q^2}{2} - \eta_q)\mathbb{E}[\|\nabla\mathscr{L}(\Theta_k^{(M,q)})\|_2^2] + \frac{\beta\eta_q^2}{2}\sigma^2)\end{aligned}$$

Proof.

$$\mathscr{F}^L - \mathscr{F}^M = \frac{1}{\sum_{j=1}^m \psi_j} \sum_{k=1}^K \sum_{i \in k} \psi_i (\mathscr{L}(\Theta_k^L, D_i) - \mathscr{L}(\Theta_k^M, D_i)). \tag{3.31}$$

For arbitrary Client i, using Gradient Descent,

$$\mathscr{L}(\Theta_k^L, D_i) - \mathscr{L}(\Theta_k^M, D_i) = \sum_{q=0}^{Q-1} (\mathscr{L}(\Theta_k^{(M,q+1)}, D_i) - \mathscr{L}(\Theta_k^{(M,q)}, D_i)).$$

Under Assumption 3.4.9,

$$\begin{aligned}
&\mathscr{L}(\Theta_k^{(M,q+1)}) - \mathscr{L}(\Theta_k^{(M,q)}) \\
\leq &\langle \nabla \mathscr{L}(\Theta_k^{(M,q)}), \Theta_k^{(M,q+1)} - \Theta_k^{(M,q)} \rangle + \frac{\beta}{2} \|\Theta_k^{(M,q+1)} - \Theta_k^{(M,q)}\|_2^2 \\
= &-\eta \langle \nabla \mathscr{L}(\Theta_k^{(M,q)}), \nabla \mathscr{L}(\Theta_k^{(M,q)}, \xi_i^e) \rangle + \frac{\beta \eta^2}{2} \|\nabla \mathscr{L}(\Theta_k^{(M,q)}, \xi_i^e)\|_2^2,
\end{aligned}$$

take expectation on both sides for random selected batch ξ_i^e under Assumption 3.4.10,

$$\mathbb{E}[\mathscr{L}(\Theta_k^{(M,q+1)})] - \mathscr{L}(\Theta_k^{(M,q)}) \leq (\frac{\beta \eta^2}{2} - \eta) \|\nabla \mathscr{L}(\Theta_k^{(M,q)})\|_2^2 + \frac{\beta \eta^2}{2} \sigma^2,$$

take expectation on both sides again on random variable $\Theta_k^{(M,q)}$, and do telescoping, we can get,

$$\begin{aligned}
&\mathbb{E}[\mathscr{L}(\Theta_k^L, D_i)] - \mathscr{L}(\Theta_k^M, D_i) \\
= &\sum_{q=0}^{Q-1} (\mathbb{E}[\mathscr{L}(\Theta_k^{(M,q+1)}, D_i)] - \mathscr{L}(\Theta_k^{(M,q)}, D_i)) \\
\leq &\sum_{q=0}^{Q-1} ((\frac{\beta \eta_q^2}{2} - \eta_q) \mathbb{E}[\|\nabla \mathscr{L}(\Theta_k^{(M,q)})\|_2^2] + \frac{\beta \eta_q^2}{2} \sigma^2).
\end{aligned} \tag{3.32}$$

Finally,

$$\begin{aligned}
&\mathbb{E}[\mathscr{F}^L] - \mathscr{F}^M \\
\leq &\frac{1}{\sum_{j=1}^m \psi_j} \sum_{k=1}^K \sum_{i \in k} \psi_i \sum_{q=0}^{Q-1} ((\frac{\beta \eta_q^2}{2} - \eta_q) \mathbb{E}[\|\nabla \mathscr{L}(\Theta_k^{(M,q)})\|_2^2] + \frac{\beta \eta_q^2}{2} \sigma^2).
\end{aligned}$$

■

Theorem 3.4.13. *(Convergence of WeCFL). Let Assumptions 3.4.1, 3.4.8, 3.4.9 and 3.4.10 hold, when* $\eta_{(t,q)} < min\{\frac{\|\theta_i^{(t)}-\Theta_k\|_2}{QU}, \frac{\mathbb{E}[\|\nabla\mathscr{F}(\Theta_{1:K}^{(t,M,q)})\|_2^2]-BU^2}{\mathbb{E}[\|\nabla\mathscr{F}(\Theta_{1:K}^{(t,M,q)})\|_2^2]+\sigma^2}\cdot\frac{2}{\beta}\}$*, the EM loss function* $\mathscr{C}$ *converges, and the FL loss function* $\mathscr{F}$ *decreases monotonically, thus the WeCFL converges.*

Proof. From the local distribution step in communication round t-1 to the Expectation step in communication round t, what is changed in the loss function of WeCFL $\mathscr{F}$ is the r_i^k, but the $\mathscr{L}(\Theta_k, D_i)$ does not change, so we can get

$$\mathscr{F}^{(t-1,L)} = \mathscr{F}^{(t,E)}, \tag{3.33}$$

then according to Lemma 3.4.11 and 3.4.12, we can get,

$$\begin{aligned}
&\mathbb{E}[\mathscr{F}^{(t,L)}] - \mathscr{F}^{(t-1,L)} \\
\leq&\eta BQU^2 + \frac{1}{\sum_{j=1}^m \psi_j}\sum_{k=1}^K\sum_{i\in k}\psi_i\sum_{q=0}^{Q-1}((\frac{\beta\eta_{(t,q)}^2}{2}-\eta_{(t,q)})\mathbb{E}[\|\nabla\mathscr{L}(\Theta_k^{(t,M,q)})\|_2^2] + \frac{\beta\eta_{(t,q)}^2}{2}\sigma^2) \\
=&\frac{1}{\sum_{j=1}^m \psi_j}\sum_{k=1}^K\sum_{i\in k}\psi_i\sum_{q=0}^{Q-1}((\frac{\beta\eta_{(t,q)}^2}{2}-\eta_{(t,q)})\mathbb{E}[\|\nabla\mathscr{L}(\Theta_k^{(t,M,q)})\|_2^2] + \frac{\beta\eta_{(t,q)}^2}{2}\sigma^2 + \eta_{(t,q)}BU^2) \\
=&\sum_{q=0}^{Q-1}((\frac{\beta\eta_{(t,q)}^2}{2}-\eta_{(t,q)})\mathbb{E}[\|\nabla\mathscr{F}(\Theta_{1:K}^{(t,M,q)})\|_2^2] + \frac{\beta\eta_{(t,q)}^2}{2}\sigma^2 + \eta_{(t,q)}BU^2),
\end{aligned} \tag{3.34}$$

then when

$$\eta_{(t,q)} < min\{\frac{\|\theta_i^{(t)}-\Theta_k\|}{QU}, \frac{\mathbb{E}[\|\nabla\mathscr{F}(\Theta_{1:K}^{(t,M,q)})\|_2^2]-BU^2}{\mathbb{E}[\|\nabla\mathscr{F}(\Theta_{1:K}^{(t,M,q)})\|_2^2]+\sigma^2}\cdot\frac{2}{\beta}\}, \tag{3.35}$$

the right term of Equation 3.34 is always negative. So we can ensure that the EM loss function $\mathscr{C}$ converges, and the FL loss function $\mathscr{F}$ decreases monotonically. Thus the WeCFL converges. ■

Theorem 3.4.14. *(Liner convergence rate of WeCFL). Let Assumptions 3.4.1, 3.4.8, 3.4.9 and 3.4.10 hold, and* $\Delta = \mathscr{F}_0 - \mathscr{F}^*$*, given any* $\epsilon > 0$*, after*

$$T \geq \frac{\Delta}{Q(\epsilon(\eta-\frac{\beta\eta^2}{2}) - \frac{\beta\eta^2}{2}\sigma^2 - \eta BU^2)} \tag{3.36}$$

communication rounds of WeCFL, we have

$$\frac{1}{TQ}\sum_{t=0}^{T-1}\sum_{q=0}^{Q-1}\mathbb{E}[\|\nabla\mathscr{F}(\Theta_{1:K}^{(t,M,q)})\|_2^2]\leq\epsilon. \tag{3.37}$$

Remark 3.4.15. (Linear convergence rate of WeCFL). According to Equation 3.36, with a proper learning rate, the convergence rate of WeCFL is $O(1/T)$, which achieves a state-of-the-art rate similar to SGD and [53]. And **bigger K or smaller non-IIDness, smaller B, better convergence rate, but less marginal benefit**.

Proof. Take expectation of Equation 3.34 on the parameter, then do telescoping from 0 to T, we can get

(3.38)

$$\begin{aligned}\Delta &\geq \mathscr{F}^{(0,L)}-\mathbb{E}[\mathscr{F}^{(T,L)}]\\ &\geq\sum_{k=1}^{K}\sum_{i\in k}\sum_{t=0}^{T-1}\sum_{q=0}^{Q-1}\frac{\psi_i}{\sum_{j=1}^{m}\psi_j}((\eta_{(t,q)}-\frac{\beta\eta_{(t,q)}^2}{2})\mathbb{E}[\|\nabla\mathscr{L}(\Theta_k^{(t,M,q)})\|_2^2]-\frac{\beta\eta_{(t,q)}^2}{2}\sigma^2-\eta_{(t,q)}BU^2)\\ &=\sum_{t=0}^{T-1}\sum_{q=0}^{Q-1}((\eta_{(t,q)}-\frac{\beta\eta_{(t,q)}^2}{2})\mathbb{E}[\|\nabla\mathscr{F}(\Theta_{1:K}^{(t,M,q)})\|_2^2]-\frac{\beta\eta_{(t,q)}^2}{2}\sigma^2-\eta_{(t,q)}BU^2).\end{aligned}$$

If

$$\frac{1}{TQ}\sum_{t=0}^{T-1}\sum_{q=0}^{Q-1}\mathbb{E}[\|\nabla\mathscr{F}(\Theta_{1:K}^{(t,M,q)})\|_2^2]\leq\epsilon, \tag{3.39}$$

then

$$T\geq\frac{\Delta}{Q(\epsilon(\eta-\frac{\beta\eta^2}{2})-\frac{\beta\eta^2}{2}\sigma^2-\eta BU^2)}. \tag{3.40}$$

■

3.5 Experimental settings

3.5.1 Datasets and Partitioning

For details about the benchmark datasets and partition methods, please refer to Section A.1 and Section A.2, respectively, in Appendix A.

3.5.2 Baseline and system settings

Baseline For single model-based FL, we choose FedAvg [67] and FedProx [51] with $\psi = 0.95$ as the baselines. For clustered FL methods, FeSEM [94] and IFCA [30], which is similar to HypCluster are chosen as the baselines. We also propose FedAvg+ and FedProx+ by training FedAvg and FedProx K times, and then learn an ensemble model via soft voting to serve all clients.

System settings We generate 200 clients to simulate a relatively large-scale FL system. We use the convolutional neural network (CNN) [44] as the basic model for each client. We evaluate the performance using both **micro accuracy** (%) and **macro F1-score** on the client-wise test datasets due to high non-IID degrees. The standard deviation is estimated from five repeats of the experiment with different random seeds, and the mean is obtained from the last three rounds out of the total 100 communication rounds.

Optimization settings For the training model, we use small CNNs with two convolutional layers for Fashion-MNIST, CIFAR-10, PathMNIST and TissueMNIST as shown in Table A.1, A.2, A.3 and A.4 of Appendix , respectively. For the optimization, SGD with a learning rate of 0.001 and momentum of 0.9 is used to train the model, and the batch size is 32.

FL settings For the FL settings, we run 100 global communication rounds, and the local steps in each communication are 10. For the clustering process, we use flattened parameters of the fully-connected layers of CNNs as data points and weighted K-Means as the clustering algorithm. The coding framework called fedbase is used, which can be accessed via the PyPI repository * or GitHub †.

Table 3.1: Test results (mean±std) in **cluster**-wise non-IID settings on Fashion-MNIST & CIFAR-10.

Datasets		Fashion-MNIST				CIFAR-10			
Non-IID setting		$\alpha = (0.1, 10)$		(3,2)−class		$\alpha = (0.1, 10)$		(3,2)−class	
K	Methods	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
	FedAvg	86.08±0.70	57.24±2.26	86.33±0.44	46.09±1.08	24.38±3.30	11.69±3.15	21.33±3.83	9.0±0.58
1	FedProx	86.32±0.78	58.03±3.19	86.42±0.63	45.86±1.42	24.73±3.68	11.28±2.35	22.66±1.13	9.23±0.78
	FedAvg+	87.61	59.48	86.95	65.61	25.97	12.16	24.35	9.06
	FedProx+	87.94	59.83	86.52	65.73	26.05	12.53	24.83	9.31
5	IFCA	84.60±2.22	62.03±3.01	84.94±2.54	66.50±4.43	34.1±4.79	22.12±2.21	29.80±4.49	17.90±2.08
	FeSEM	94.64±1.54	82.90±2.38	94.20±1.96	77.07±6.05	59.06±3.24	**32.33±7.25**	58.76±3.35	35.75±2.54
	WeCFL	**94.64±1.02**	**84.4±1.31**	**94.97±1.43**	**77.36±3.94**	**59.26±3.32**	32.26±3.46	**62.44±2.53**	**38.55±1.76**
	FedAvg+	89.42	67.83	86.91	63.01	28.45	13.79	27.28	9.81
	FedProx+	89.55	68.02	86.73	63.42	28.33	13.64	26.94	9.64
10	IFCA	82.10±5.40	62.62±8.22	86.58±4.97	66.22±5.69	34.84±5.82	22.76±3.99	34.06±2.60	18.7±1.31
	FeSEM	95.73±1.28	89.34±1.57	95.54±0.74	84.43±2.38	66.89±2.18	38.35±4.24	71.76±2.23	49.72±3.84
	WeCFL	**95.88±0.85**	**89.81±1.59**	**97.10±0.51**	**88.96±1.36**	**70.95±3.57**	**40.19±2.88**	**72.13±1.88**	**50.65±2.15**

Table 3.2: Test results (mean±std) in **cluster**-wise non-IID settings on PathMNIST & TissueMNIST.

Datasets		PathMNIST				TissueMNIST			
Non-IID setting		$\alpha = (0.1, 10)$		(3,2)−class		$\alpha = (0.1, 10)$		(3,2)−class	
K	Methods	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
	FedAvg	31.38±8.58	14.47±4.27	21.36±5.48	11.49±2.38	49.96±3.39	18.31±4.31	53.46±2.21	15.28±1.36
1	FedProx	27.6±6.15	14.07±3.42	25.7±8.48	11.62±1.08	49.78±2.64	17.85±3.81	54.92±3.7	15.15±1.47
	FedAvg+	35.84	17.01	25.51	12.14	49.52	17.59	54.98	15.12
	FedProx+	27.57	15.74	29.7	13.05	48.88	17.08	52.24	15.54
5	IFCA	38.13±2.53	25.22±1.74	34.16±3.76	22.52±1.13	27.44±16.39	16.37±10.45	41.87±20.04	21.59±7.3
	FeSEM	59.85±1.45	33.5±4.08	66.37±7.19	**41.34±4.12**	72.38±1.81	36.79±1.06	70.62±2.41	28.43±2.54
	WeCFL	**68.79±0.18**	**38.94±0.97**	**66.84±5.22**	41.8±2.43	**72.88±1.11**	**37.19±1.7**	**73.5±1.63**	**34.02±4.97**
	FedAvg+	33.19	19.98	24.82	13.73	49.5	18.03	54.78	13.23
	FedProx+	28.21	16.17	35.62	15.95	46.57	16.47	53.47	14.88
10	IFCA	42.34±2.73	29.1±1.52	37.22±4.23	20.2±2.04	38.76±10.94	20.38±2.01	49.31±13.97	21.51±3.68
	FeSEM	79.31±0.72	48.14±0.23	71.37±1.5	53.78±2.21	77.12±1.68	47.69±3.1	77.92±1.53	45.68±6.71
	WeCFL	**81.88±2.43**	**50.17±1.05**	**73.19±2.0**	**55.53±3.51**	**77.37±1.12**	**48.5±4.1**	**78.32±1.5**	**48.58±5.28**

*https://pypi.org/project/fedbase/

†https://github.com/jie-ma-ai/FedBase

Table 3.3: Test results (mean±std) in **client**-wise non-IID settings on Fashion-MNIST & CIFAR-10.

Datasets		Fashion-MNIST				CIFAR-10			
Non-IID setting		$\alpha = 0.1$		2−class		$\alpha = 0.1$		2−class	
K	Methods	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
1	FedAvg	85.9±0.46	54.52±2.66	86.17±0.25	44.88±1.24	25.62±3.47	11.38±2.02	24.3±3.53	8.56±0.64
	FedProx	86.03±0.58	54.69±3.32	86.47±0.23	44.89±1.38	25.72±3.29	11.14±1.49	24.19±2.45	8.69±0.74
5	FedAvg+	86.12	61.07	86.5	45.39	25.71	12.45	24.83	8.74
	FedProx+	86.39	56.56	86.15	45.43	25.58	12.43	25.88	8.55
	IFCA	90.13±6.81	68.47±5.23	91.54±5.04	72.3±5.32	47.21± 10.28	22.67±1.48	46.54±12.8	17.78±1.29
	FeSEM	91.51±2.9	73.78±9.88	91.83±1.24	71.05±8.63	54.3±4.58	24.78±6.01	55.55±4.83	32.8±4.18
	WeCFL	91.59±0.82	74.45±10.53	91.76±1.53	69.47±5.04	55.09±5.1	27.29±8.37	55.89±5.92	33.12±5.0
10	FedAvg+	86.81	60.43	86.91	47.12	27.83	13.65	27.71	9.65
	FedProx+	86.24	56.2	86.78	42.83	25.86	12.84	26.16	9.94
	IFCA	91.04±4.33	68.6±6.77	91.42±5.16	72.29±5.8	47.62±10.15	23.36±2.48	47.96±10.59	17.88±1.04
	FeSEM	93.3±2.0	**80.47±11.05**	93.75±1.53	79.39±6.57	67±1.57	31.69±8.52	63.64±6.51	42.97±6.08
	WeCFL	**94.21±1.67**	79.31±11.02	**94.05±1.67**	**81.41±5.7**	**69.47±4.16**	**34.1±7.79**	**66.8±6.39**	**45.61±5.9**

Table 3.4: Test results (mean±std) in **client**-wise non-IID settings on PathMNIST & TissueMNIST.

Datasets		PathMNIST				TissueMNIST			
Non-IID setting		$\alpha = 0.1$		2−class		$\alpha = 0.1$		2−class	
K	Methods	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
1	FedAvg	26.41±9.15	14.29±3.08	26.11±8.51	13.05±2.33	52.42±4.04	16.23±3.81	54.11±2.28	14.51±1.28
	FedProx	27.61±7.38	13.97±2.6	28.77±8.33	12.16±2.27	53.42±4.29	15.84±3.41	54.51±3.26	14.43±1.36
5	FedAvg+	32.68	15.03	29.8	13.02	53.15	16.51	54.63	14.57
	FedProx+	33.19	15.66	30.51	13.49	53.56	17.89	55.03	14.78
	IFCA	38.13±2.53	25.22±1.74	34.16±3.76	22.52±1.13	38.76±10.94	20.38±2.01	49.31±13.97	21.51±3.68
	FeSEM	59.85±1.45	33.5±4.08	64.46±6.12	38.41±3.19	72.88±1.11	33.19±1.7	70.62±2.41	28.43±2.54
	WeCFL	**67.91±1.35**	**41.08±3.13**	**66.37±7.19**	**41.34±4.12**	**75.58±4.78**	**37.02±0.93**	**72.93±1.72**	**31.83±5.73**
10	FedAvg+	29.83	16.75	28.35	13.49	53.5	18.03	54.58	13.46
	FedProx+	29.36	16.55	29.07	13.63	54.69	17.36	56.03	15.21
	IFCA	51.88±13.67	27.81±2.21	37.22±4.23	20.2±2.04	27.44±16.39	16.37±10.45	41.87±20.04	21.59±7.3
	FeSEM	78.93±4.27	**52.94±5.42**	70.93±4.27	52.94±5.42	78.85±2.29	52.32±7.59	77.92±1.53	45.68±6.71
	WeCFL	**80.27±3.01**	52.63±3.59	**71.37±1.5**	**53.78±2.21**	**79.05±3.06**	**52.67±6.2**	**78.62±1.77**	**46.86±5.46**

3.6 Experimental analysis

3.6.1 Comparison study

Cluster-wise non-IID results Table 3.1 and 3.2 show the performance comparison of the methods under the cluster-wise non-IID setting. Measured by cluster-wise test dataset-based micro accuracy and macro F1-score, WeCFL outperforms almost all baselines on Fashion-MNIST, CIFAR-10, PathMNIST and TissueMNIST datasets. IFCA shows a relatively poor performance on all datasets. One of the main reasons for this is IFCA's unstable clustering capability. IFCA's clustering procedure is not a standard clus-

tering algorithm with a well-defined distance or similarity metric. Specifically, in IFCA's clustering procedure, the similarity metric is based on how the cluster-specific model performs on the client's local dataset. This kind of metric is unlike other classic distance and similarity metrics that have demonstrated good characteristics from geometry and algebra perspectives.

Within a proper interval, larger K leads to better performance. As shown in Table 3.1 and 3.2, when K is increased from 5 to 10, the performance of all methods is improved. However, IFCA's performance sometimes decreases due to its unstable clustering capability. FedAvg and FedProx perform poorly on CIFAR-10, which demonstrates their inability to tackle cluster-wise non-IID data. Their ensemble extensions, FedAvg+ and FedProx+, slightly increase the performance in most datasets because the model's generalization has been improved by leveraging ensemble learning. It is noteworthy that FedAvg+ and FedProx+ are very stable due to assembling multiple models; thus we didn't measure the variance of these ensemble models.

Client-wise non-IID results Table 3.3 and 3.4 demonstrate the experiment results under the client-wise non-IID setting. The results show that WeCFL outperforms almost all baselines. The statistical heterogeneity of CIFAR-10, PathMNIST and TissueMNIST is much higher than Fashion-MNIST or other MNIST dataset families. Therefore, WeCFL demonstrates superior performance improvements in CIFAR-10, PathMNIST and TissueMNIST than in Fashion-MNIST. Within a proper interval, larger K will lead to better performance. As shown in the tables, when K is increased from 5 to 10, almost all methods' performance increases. Furthermore, with a higher K, the performance of WeCFL improves more in CIFAR-10, PathMNIST and TissueMNIST than in Fashion-MNIST.

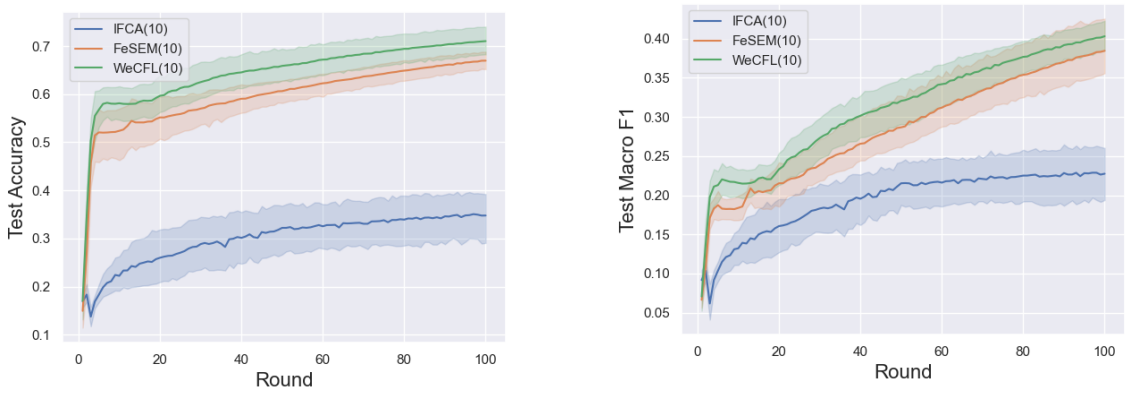

Figure 3.2: Convergence of **clustered FL** methods on **CIFAR-10** under the **(3,2)-class** cluster-wise non-IID setting

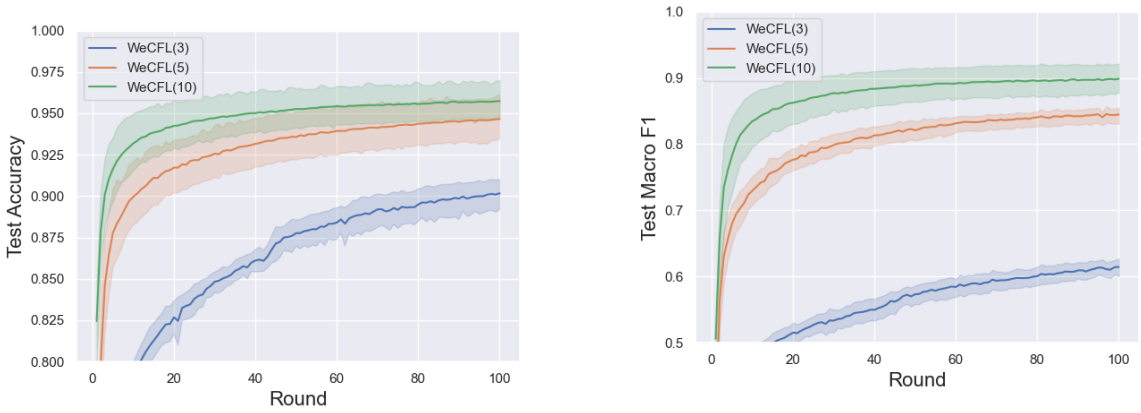

Figure 3.3: Convergence of **WeCFL** on **Fashion-MNIST** under the $\boldsymbol{\alpha} = \mathbf{(0.1, 10)}$ cluster-wise non-IID setting

3.6.2 Convergence analysis

Comparison of baselines Figure 3.2 shows the convergence curves of three clustered FL methods: IFCA, FeSEM and WeCFL in the cluster-wise non-IID setting. The left- and right-hand panels show the methods' performances in test accuracy and macro F1, respectively. The experimental dataset is derived from CIFAR-10 by preprocessing the dataset with a cluster-wise non-IID setting. Specifically, the non-IID of (3,2)-class assigns three classes to each cluster while assigning two classes to each client. As shown in

Figure 3.2, WeCFL converges faster than others.

Different K Figure 3.3 demonstrates that WeCFL can convergence in different K. The experimental dataset is derived from Fashion-MNIST using the Dirichlet-based cluster-wise non-IID preprocessing method with $\alpha = 0.1, 10$. Specifically, we use a Dirichlet distribution with $\alpha = 0.1$ to control the inter-cluster non-IID with large variance, and then use another Dirichlet distribution with $\alpha = 10$ to control intra-cluster client-wise non-IID with small variance. The figures demonstrate that a larger K is more likely to lead to better performance on both test accuracy and macro F1 score.

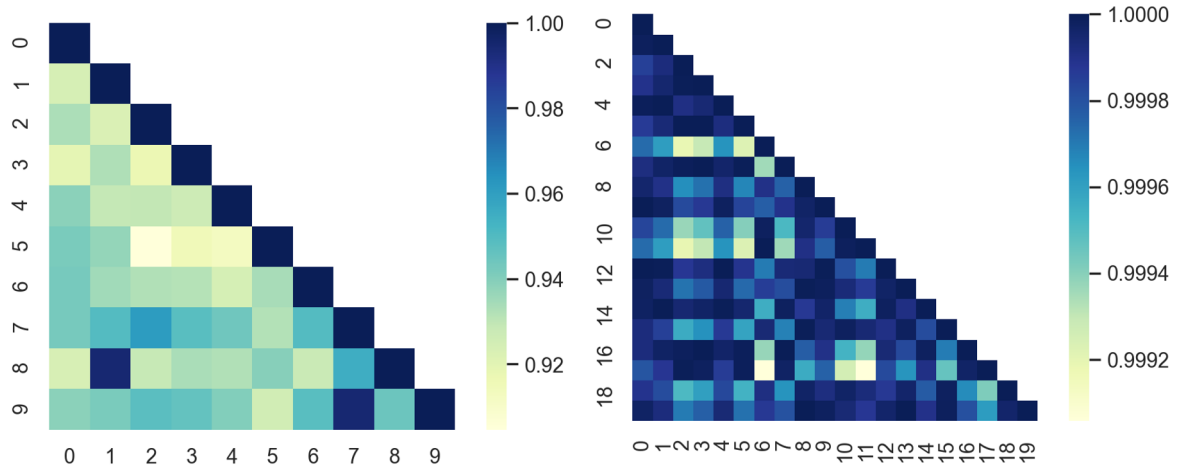

Figure 3.4: Cosine similarity heatmap of 10 clusters' centroids (left) and 20 clients in a cluster (right).

3.6.3 Clustering study

Clustering evaluation A good clustering generally satisfies two evaluation criteria: the clients in the cluster are similar to each other, and the clusters are dissimilar to each other. We use cosine similarity to measure the differences between clients or clusters generated by WeCFL. Figure 3.4 visualizes the inter-cluster and intra-cluster similarities. Specifically, the left panel shows the similarity among 10 clusters' centroids;

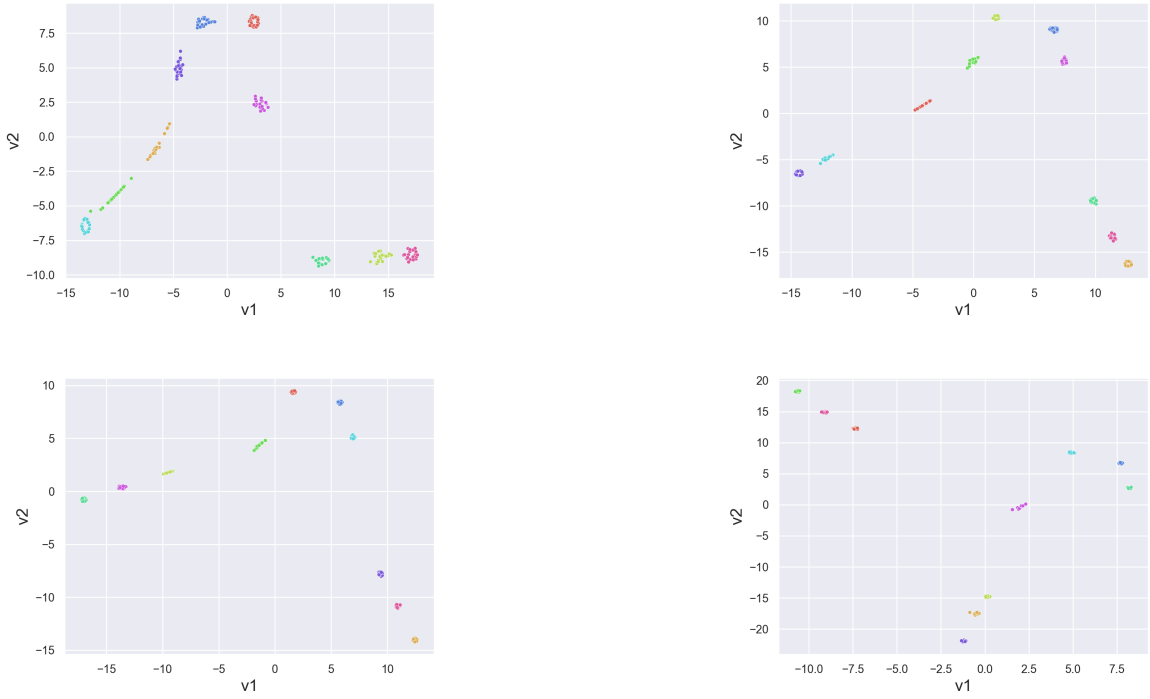

Figure 3.5: T-SNE visualization of clustering results on the Fashion-MNIST in the first four communication rounds under the $\alpha = (0.1, 10)$ cluster-wise non-IID setting, generated by 200 clients across $K = 10$ clusters. Different colors represent different cluster labels. The order is left-to-right then top-to-bottom.

the similarity value is around 0.93, indicating large differences between the clusters. The right panel is the similarity between 20 intra-cluster clients; all of them are greater than 0.999. In summary, Figure 3.4 demonstrates that WeCFL can distinct clusters (left panel) and group similar clients into the same cluster (right panel).

Clustering visualization To verify the effectiveness of the proposed WeCFL method and whether the clients are clustered properly, we visualize the clustering results using t-SNE [87] to transform client-wise representations into two-dimensional vectors. All clustering results are generated by WeCFL. Figure 3.5 and 3.6 demonstrate the changing clustering results in view of t-SNE for the first four communication rounds on the Fashion-MNIST for $K = 10$ and $K = 3$, respectively, while the non-IID setting is $\alpha = (0.1, 10)$ cluster-wise and the ground truth of cluster number K is ten. Then, the clustering analysis of WeCFL can be summarized below,

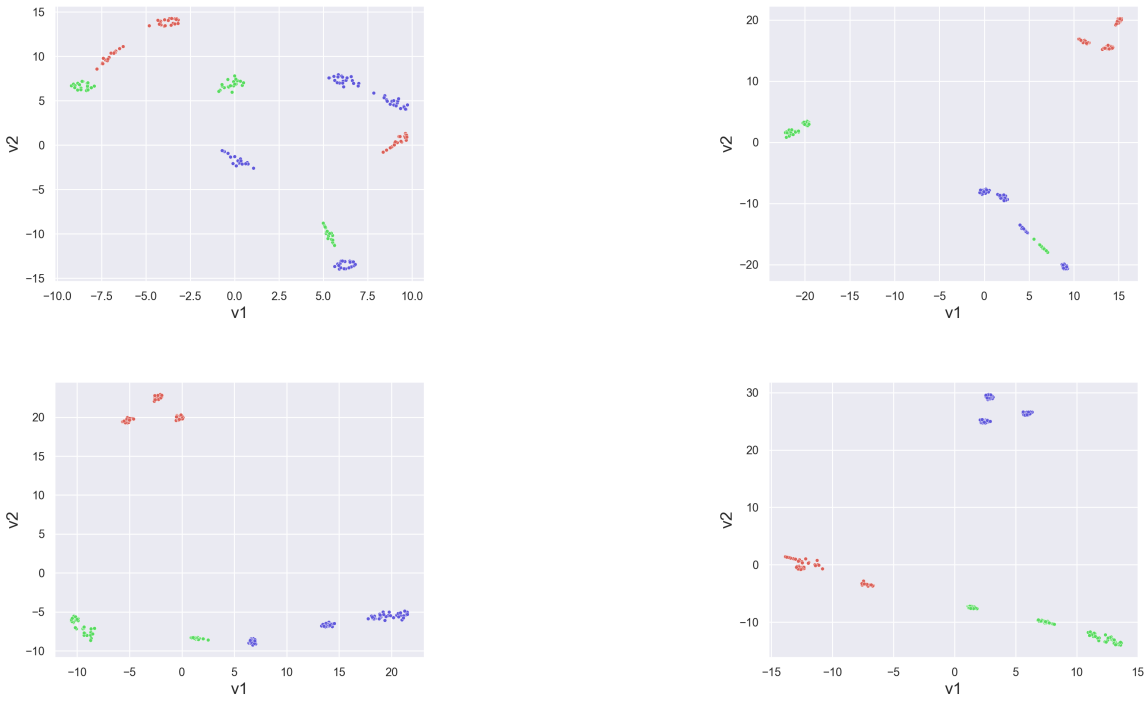

Figure 3.6: T-SNE visualization of clustering results in the first four communication rounds on the Fashion-MNIST under the $\alpha = (0.1, 10)$ cluster-wise non-IID setting, generated by 200 clients across $K = 3$ clusters. Different colors represent different cluster labels. The order is left-to-right then top-to-bottom.

- The clustering converges very fast. For $K = 10$, it takes only one communication round to converge. Even for $K = 3$, it takes only three communication rounds to converge. With more communications, the inter-cluster distance becomes larger and the intra-cluster distance becomes smaller.

- The clustering converges very well. For $K = 10$, the clustering results exactly match the initial partition or ground truth. For $K = 3$ that can not divide 10, the clustering results keep the initialized clusters and no break up.

- The range of the clusters or intra-cluster distance becomes smaller and smaller by the communication round for $K = 10$ and $K = 3$, which indicates that the clusterability measure B is better and better.

In general, it takes no more than ten communication rounds to achieve convergence on

clustering. Once clustering converges, the operations on later communication rounds are equivalent to conducting a cluster-specific FedAvg.

3.7 Conclusion

This work rethinks the clustered FL from a new perspective on its clustering, and then proposes a general form for clustered FL. A weighted clustering has been applied to clustered FL. The most important contribution is the proposal of a new convergence analysis to the general form of clustered FL. Experiments on both cluster-wise and client-wise non-IID settings support our claims.

CHAPTER 4

CLUSTERED FEDERATED LEARNING WITH ROBUSTNESS: A CONTRASTIVE LEARNING APPROACH

4.1 Motivation

Clustering is a common technique used for tasks involving static data points. However, in the context of clustered FL, the data points clustered for each communication such as loss and gradients are dynamic, while we aim to obtain a robust clustering result. Hence, the problem can be characterized as seeking robust clustering results for dynamic clusters, where a gap exists between the clustering objective $\mathscr{C}$ and the FL objective $\mathscr{F}$.

In order to align the clustering objective $\mathscr{C}$ with the FL objective $\mathscr{F}$ and make them consistent, contrastive learning is employed in $\mathscr{F}$. Contrastive learning [32] is based on the philosophy of enhancing both similarity and dissimilarity simultaneously to put similar data closer together and dissimilar data further away from each other, which is in line with the objective of clustering, maximizing intra-cluster similarity and inter-cluster

dissimilarity. Figure 4.1 depicts the schema in which contrastive learning is employed to improve the stability of clustering for the models of each cluster.

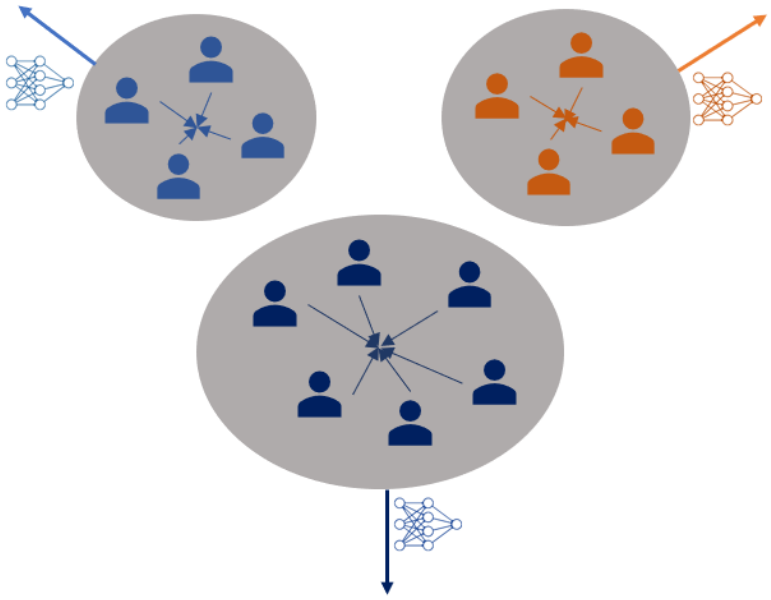

Figure 4.1: A schematic diagram that shows how contrastive learning works in clustered FL, which enhances the intra-cluster similarity shown by inward arrows and inter-cluster dissimilarity shown by outward arrows.

4.2 Formulation

For the clustered FL, as mentioned in WeCFL [62], it can be formulated into a bi-level optimization framework as follows,

$$\underset{\{\Theta_k\}}{\text{minimize}}\ \mathscr{F}: \frac{1}{m}\sum_{k=1}^{K}\sum_{i=1}^{m}\psi_i r_{i,k}\mathscr{L}(\mathscr{H}_k, D_i, \Theta_k) \tag{4.1a}$$

$$\textit{subject to}\ \{r_{i,k}\} = \underset{\{r_{i,k}\}}{\text{argmin}}\,\mathscr{C}: \sum_{k=1}^{K}\sum_{i=1}^{m}\psi_i r_{i,k} d(g_i, G_k) \tag{4.1b}$$

And we define the new bilevel optimization objective of clustered FL as follows,

$$\underset{\{\Theta_k\}}{\text{minimize}}\ \mathscr{F}: \frac{1}{m}\sum_{k=1}^{K}\sum_{i=1}^{m}\psi_i r_{i,k}[\mathscr{L}(\mathscr{H}_k, D_i, \Theta_k) + \mu\mathscr{T}(\{\Theta_k\})] \tag{4.2a}$$

$$\textit{subject to}\ \{r_{i,k}\} = \underset{\{r_{i,k}\}}{\text{argmin}}\,\mathscr{C}: \sum_{k=1}^{K}\sum_{i=1}^{m}\psi_i r_{i,k} d(g_i, G_k). \tag{4.2b}$$

$$\frac{exp(sim(\theta_i, \Theta_k)/\tau)}{\sum_{k'=1}^{K} exp(sim(\theta_i, \Theta'_k)/\tau)} \tag{4.3}$$

$$\frac{exp(sim(h(\theta_i, D_i), \mathscr{H}(\Theta_k, D_i)/\tau)}{\sum_{k'=1}^{K} exp(sim(h(\theta_i, D_i), \mathscr{H}(\Theta'_k, D_i))/\tau)} \tag{4.4}$$

Since this method can be integrated into any clustered FL algorithms, it is referred to as CFL-CON, or simply CON in this thesis, shorted for **C**lustered **FL** with **con**trastive learning. Specifically, for CFL-CON based on parameters and representations, they are shorted for CFL-CON-para and CFL-CON-rep, respectively.

4.3 Algorithm

Integrating CON into various clustered FL methods, including IFCA, FeSEM, WeCFL, etc., is a simple process. To implement this approach, the FL loss function should be augmented with $\mathscr{T}$, and the standard optimization process should be followed. The primary advantage of using stable clustering is that it requires only a limited number of shots for the clustering process. This, in turn, significantly reduces the amount of computational resources required, resulting in faster and more efficient training. Additionally, stable clustering can mitigate the risk of overfitting, as it reduces the number of updates needed for the clustering process. Consequently, Algorithm 2 can be modified in the following manner from Algorithm 1, and $\mathscr{T}$ can be Term 4.3 or Term 4.4.

Algorithm 2: CFL-CON: Clustered FL with Contrastive Learning

1: **Input:** K, $\{D_1, D_2, \ldots, D_m\}$

2: **Initialize:** Randomly initialize $\{\Theta_1, \Theta_2, \ldots, \Theta_K\}$

3: **repeat**

4: **Expectation step for few shots:** Assign Client i to Cluster k by

$$k = \underset{k}{\operatorname{argmin}}\, \psi_i d(g_i, G_k).$$

5: **Maximization / Aggregation step:** Compute cluster center Θ_k by

$$\frac{1}{\sum_{i=1}^{m} \psi_i} \sum_{i=1}^{m} r_{i,k} \psi_i \theta_i.$$

6: **Distribution step:** Send H_k to clients in Cluster k.

7: **Local update step:** Run Gradient Descent Q steps using local data D_i to minimize

$$\frac{1}{m} \sum_{k=1}^{K} \sum_{i=1}^{m} \psi_i r_{i,k} [\mathscr{L}(\mathscr{H}_k, D_i, \Theta_k) + \mu \mathscr{T}(\{\Theta_k\})].$$

8: **until** convergence condition satisfied

9: **Output:** $\{r_{i,k}\}, \{\Theta_1, \Theta_2, \ldots, \Theta_K\}$.

4.4 Experiments

4.4.1 Experimental settings

4.4.1.1 Datasets and partitioning

For details about the benchmark datasets and partition methods, please refer to Section A.1 and Section A.2, respectively, in Appendix A.

4.4.1.2 Baselines

For the global model-based FL, we choose FedAvg [67] as the baseline. For clustered FL methods, we use IFCA [30], FeSEM [94], and WeCFL [62]. IFCA represents clustered FL methods that utilize minimum loss for clustering, while FeSEM and WeCFL represent clustered FL methods that employ partial model parameters for clustering. We then incorporate both CFL-CON-rep and CFL-CON-para into these three clustering methods to evaluate the effectiveness of the CFL-CON terms. The new baselines are named accordingly, such as IFCA-CON-rep and IFCA-CON-para, for instance.

4.4.1.3 Simulation settings

Optimization settings For the training model, we use small CNNs [44] with two convolutional layers for Fashion-MNIST, CIFAR-10, PathMNIST and TissueMNIST as shown in Table A.1, A.2, A.3 and A.4 in Appendix , respectively. For the optimization, an optimizer of SGD with a learning rate of 0.001 and momentum of 0.9 is used to train the model, and the batch size is 32.

Evaluation metrics We evaluate the performance using both **micro accuracy** (%) and **macro F1-score** on the client-wise test datasets due to high non-IID degrees. The standard deviation is estimated from five repeats of the experiment with different random seeds, and the mean is obtained from the last three rounds out of the total 100 communication rounds.

Other settings For the FL settings, the local steps in each communication round are 10. For the clustering process, we use flattened parameters of the fully-connected layers of CNNs as data points and weighted K-Means as the clustering algorithm. The coefficient of the CFL-CON term μ is chosen from a set of $\{0.1, 0.5, 2, 5\}$, and the temperature of the CFL-CON term τ is chosen from a set of $\{0.1, 1, 10\}$ based on the performance. For

clustered FL with stable clustering, including FeSEM and WeCFL, the number of few shots is set to 10. For IFCA, where the clustering process is unstable, the clustering and optimization processes are always intertwined and occur simultaneously. The coding framework called fedbase is used, which can be accessed via the PyPI repository [*] or GitHub [†].

4.4.2 Experimental analysis

Cluster-wise non-IID Table 4.1 and 4.2 display the experimental results for four datasets under the cluster-wise non-IID setting. Firstly, both CFL-CON-para and CFL-CON-rep considerably enhance the performance of the original clustered FL methods under the cluster-wise non-IID setting, as evidenced by their accuracy and Macro-F1 scores. WeCFL-CON-para achieves the best performance for both $K = 5$ and $K = 10$, particularly for the Fashion-MNIST and CIFAR-10 datasets. The CFL-CON methods show lower variance, signifying improved robustness in performance. Secondly, IFCA-CON attains a more significant marginal gain over IFCA compared to both FeSEM-CON and WeCFL-CON, primarily due to IFCA's lower base and greater potential for improvement. However, there are some exceptions for IFCA, such as PathMNIST with $K = 10$ and $\alpha = (0.1, 10)$. Thirdly, overall, CON-para outperforms CON-rep when other conditions remain the same, and it is also more computationally efficient. As a result, CON-para is recommended for use in cluster-wise non-IID settings.

Client-wise non-IID Table 4.3 and 4.4 present the experimental results for four datasets under the client-wise non-IID setting. Firstly, it can be concluded that both CFL-CON-rep and CFL-CON-para significantly improve the performance compared to the original methods, as evidenced by their accuracy and Macro-F1 scores. Generally,

[*] https://pypi.org/project/fedbase/
[†] https://github.com/jie-ma-ai/FedBase

Table 4.1: Test results (mean±std) in **cluster**-wise non-IID settings on Fashion-MNIST & CIFAR-10.

Datasets		Fashion-MNIST				CIFAR-10			
Non-IID setting		$\alpha = (0.1, 10)$		(3,2)−class		$\alpha = (0.1, 10)$		(3,2)−class	
K	Methods	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
1	FedAvg	86.08±0.70	57.24±2.26	86.33±0.44	46.09±1.08	24.38±3.30	11.69±3.15	21.33±3.83	9.0±0.58
5	IFCA	84.60±2.22	62.03±3.01	84.94±2.54	66.50±4.43	34.1±4.79	22.12±2.21	29.80±4.49	17.90±2.08
	IFCA-CON-rep	89.86±1.58	67.94±0.87	91.88±2.37	69.1±1.75	41.33±5.63	25.03±4.53	38.06±5.37	23.9±2.8
	IFCA-CON-para	90.54±1.27	73.94±0.69	92.88±1.25	71.1±1.08	43.33±3.63	25.87±3.35	40.06±4.56	25.61±3.4
	FeSEM	94.64±1.54	82.90±2.38	94.20±1.96	77.07±6.05	59.06±3.24	32.33±7.25	58.76±3.35	35.75±2.54
	FeSEM-CON-rep	95.26±0.04	86.62±0.06	92.59±0.09	73.48±0.26	59.93±0.3	32.62±0.14	59.07±0.24	36.4±0.51
	FeSEM-CON-para	95.42±0.03	86.04±0.02	94.05±0.02	80.91±0.34	58.39±0.09	33.98±0.15	59.31±0.11	36.97±0.08
	WeCFL	94.64±1.02	84.4±1.31	94.97±1.43	77.36±3.94	59.26±3.32	32.26±3.46	62.44±2.53	38.55±1.76
	WeCFL-CON-rep	**95.43±0.01**	85.96±0.08	93.91±0.05	72.25±0.11	59.79±0.26	33.25±0.14	61.0±0.15	38.55±0.07
	WeCFL-CON-para	95.42±0.01	**89.38±0.06**	**95.98±0.04**	**82.41±0.22**	**61.48±0.21**	**35.93±0.1**	**63.24±0.25**	**40.54±0.31**
10	IFCA	82.10±5.40	62.62±8.22	86.58±4.97	66.22±5.69	34.84±5.82	22.76±3.99	34.06±2.60	18.7±1.31
	IFCA-CON-rep	91.97±1.69	75.54±2.27	89.15±2.08	57.87±3.02	48.05±5.94	25.53±6.17	42.61±6.57	31.5±3.39
	IFCA-CON-para	93.25±1.43	80.54±1.75	89.63±2.59	59.63±3.57	48.65±4.87	27.53±4.71	43.52±5.28	32.0±2.96
	FeSEM	95.73±1.28	89.34±1.57	95.54±0.74	84.43±2.38	66.89±2.18	38.35±4.24	71.76±2.23	49.72±3.84
	FeSEM-CON-rep	95.72±0.03	89.5±0.09	95.79±0.07	78.96±0.2	68.4±0.05	34.75±0.05	72.68±0.12	47.02±0.11
	FeSEM-CON-para	95.78±0.02	89.99±0.16	96.71±0.05	82.55±0.13	69.88±0.07	36.0±0.09	72.92±0.08	48.98±0.15
	WeCFL	95.88±0.85	89.81±1.59	97.10±0.51	88.96±1.36	70.95±3.57	40.19±2.88	72.13±1.88	50.65±2.15
	WeCFL-CON-rep	96.01±0.03	90.29±0.22	97.12±0.04	90.79±0.2	70.96±0.1	40.9±0.15	72.38±0.03	50.68±0.17
	WeCFL-CON-para	**96.93±0.04**	**91.22±0.12**	**97.18±0.06**	**91.88±0.32**	**71.23±0.09**	**42.34±0.11**	**72.73±0.05**	**51.57±0.21**

Table 4.2: Test results (mean±std) in **cluster**-wise non-IID settings on PathMNIST & TissueMNIST.

Datasets		PathMNIST				TissueMNIST			
Non-IID setting		$\alpha = (0.1, 10)$		(3,2)−class		$\alpha = (0.1, 10)$		(3,2)−class	
K	Methods	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
1	FedAvg	31.38±8.58	14.47±4.27	21.36±5.48	11.49±2.38	49.96±3.39	18.31±4.31	53.46±2.21	15.28±1.36
5	IFCA	38.13±2.53	25.22±1.74	34.16±3.76	22.52±1.13	27.44±16.39	16.37±10.45	41.87±20.04	21.59±7.3
	IFCA-CON-rep	52.02±5.71	32.44±3.53	27.11±2.94	18.75±1.65	49.28±6.65	22.69±3.42	57.67±8.05	27.95±5.21
	IFCA-CON-para	56.02±4.32	33.67±3.77	27.11±2.94	18.75±1.65	50.17±2.33	23.8±4.32	57.87±7.51	27.63±4.68
	FeSEM	59.85±1.45	33.5±4.08	66.37±7.19	41.34±4.12	72.38±1.81	36.79±1.06	70.62±2.41	28.43±2.54
	FeSEM-CON-rep	62.0±0.82	35.92±0.21	67.98±1.17	42.82±0.66	77.79±0.08	40.78±0.11	76.04±0.1	40.69±0.39
	FeSEM-CON-para	61.63±0.11	34.03±0.08	67.31±0.43	41.62±0.35	77.72±0.05	41.07±0.13	82.85±0.06	41.41±0.32
	WeCFL	68.79±0.18	38.94±0.97	66.84±5.22	41.8±2.43	72.88±1.11	37.19±1.7	73.5±1.63	34.02±4.97
	WeCFL-CON-rep	**69.76±2.26**	42.54±1.72	66.54±0.78	43.42±0.63	77.85±0.03	**41.12±0.1**	82.96±0.05	41.7±0.26
	WeCFL-CON-para	69.2±1.14	**43.91±0.54**	**69.08±0.46**	**46.68±0.62**	**77.86±0.06**	40.94±0.09	**83.99±0.21**	**43.96±1.07**
10	IFCA	42.34±2.73	29.1±1.52	37.22±4.23	20.2±2.04	38.76±10.94	20.38±2.01	49.31±13.97	21.51±3.68
	IFCA-CON-rep	42.08±2.67	29.51±2.86	30.02±8.11	17.94±4.48	78.44±9.46	34.95±5.75	54.25±4.25	26.3±2.63
	IFCA-CON-para	43.09±1.97	30.25±2.51	30.02±8.11	17.94±4.48	78.69±8.62	35.19±4.75	55.61±3.91	27.1±3.05
	FeSEM	79.31±0.72	48.14±0.23	71.37±1.5	53.78±2.21	77.12±1.68	47.69±3.1	77.92±1.53	45.68±6.71
	FeSEM-CON-rep	79.36±0.05	47.71±0.09	**76.49±0.09**	55.78±0.55	**87.7±0.07**	**48.06±0.25**	78.73±0.04	54.41±0.31
	FeSEM-CON-para	79.45±0.01	48.03±0.0	75.84±0.45	**55.94±0.88**	87.27±0.1	47.56±0.14	78.81±0.09	53.82±0.35
	WeCFL	**81.88±2.43**	50.17±1.05	73.19±2.0	55.53±3.51	77.37±1.12	48.5±4.1	78.32±1.5	48.58±5.28
	WeCFL-CON-rep	81.07±0.13	50.41±0.29	73.41±0.16	52.95±0.45	87.64±0.09	47.87±0.07	79.35±0.12	52.27±0.15
	WeCFL-CON-para	81.75±1.14	**51.73±1.47**	74.67±0.41	52.33±0.49	87.52±0.03	47.36±0.09	**79.72±0.13**	**54.81±0.22**

both CFL-CON-rep and CFL-CON-para exhibit lower variance, indicating increased robustness. Among the three clustered FL methods, IFCA, FeSEM, and WeCFL, WeCFL

and WeCFL-CON consistently perform the best. Secondly, under the client-wise non-IID setting, for Fashion-MNIST and CIFAR-10, CFL-CON-para slightly outperforms CFL-CON-rep, while CFL-CON-rep's performance is comparable to CFL-CON-para for PathMNIST and TissueMNIST of the MedMNIST datasets. Overall, CFL-CON-rep and CFL-CON-para demonstrate similar performance under the client-wise non-IID setting. Thirdly, IFCA-CON does not show significant improvements for Fashion-MNIST and CIFAR-10, as the base performance is high enough, leaving little room for further enhancement. In conclusion, although both CFL-CON-rep and CFL-CON-para offer marginal benefits and exhibit close performance, CFL-CON-para is still recommended due to its computational efficiency.

Table 4.3: Test results (mean±std) in **client**-wise non-IID settings on Fashion-MNIST & CIFAR-10.

Datasets		Fashion-MNIST				CIFAR-10			
Non-IID setting		$\alpha = 0.1$		2−class		$\alpha = 0.1$		2−class	
K	Methods	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
1	FedAvg	85.9±0.46	54.52±2.66	86.17±0.25	44.88±1.24	25.62±3.47	11.38±2.02	24.3±3.53	8.56±0.64
5	IFCA	90.13±6.81	68.47±5.23	91.54±5.04	72.3±5.32	47.21± 10.28	22.67±1.48	46.54±12.8	17.78±1.29
	IFCA-CON-rep	90.23±2.94	62.83±4.3	90.4±3.04	66.74±3.14	48.76±3.24	24.67±0.31	53.71±2.53	24.26±0.95
	IFCA-CON-para	90.89±3.51	65.57±4.91	91.43±2.97	66.15±4.33	49.54±3.92	24.67±1.68	54.09±3.13	23.61±1.18
	FeSEM	91.51±2.9	73.78±9.88	91.83±1.24	71.05±8.63	54.3±4.58	24.78±6.01	55.55±4.83	32.8±4.18
	FeSEM-CON-rep	91.87±0.07	74.19±0.2	90.61±0.08	71.28±0.21	55.02±0.45	34.78±1.35	55.28±0.81	32.68±1.65
	FeSEM-CON-para	91.42±0.11	70.02±0.41	92.0±0.04	72.1±0.17	57.56±1.84	34.31±1.94	**56.41±1.76**	32.77±0.12
	WeCFL	91.59±0.82	74.45±10.53	91.76±1.53	69.47±5.04	55.09±5.1	27.29±8.37	55.89±5.92	33.12±5.0
	WeCFL-CON-rep	**91.65±0.0**	74.78±0.12	91.42±0.05	71.37±0.2	55.42±0.13	33.52±0.41	56.3±1.06	34.0±0.33
	WeCFL-CON-para	91.65±0.06	**74.79±0.11**	**92.75±0.04**	**72.37±0.18**	**58.69±0.55**	**36.17±0.55**	56.26±1.34	**35.07±0.59**
10	IFCA	91.04±4.33	68.6±6.77	91.42±5.16	72.29±5.8	47.62±10.15	23.36±2.48	47.96±10.59	17.88±1.04
	IFCA-CON-rep	89.94±3.91	55.86±2.89	92.76±4.9	69.61±8.2	48.25±10.57	24.93±4.01	50.6±13.09	26.27±2.85
	IFCA-CON-para	91.33±3.61	58.47±4.35	92.88±4.36	71.52±5.91	48.69±8.1	24.55±5.61	51.58±8.31	24.91±3.18
	FeSEM	93.3±2.0	80.47±11.05	93.75±1.53	79.39±6.57	67±1.57	31.69±8.52	63.64±6.51	42.97±6.08
	FeSEM-CON-rep	94.4±0.05	79.76±0.14	94.1±0.07	80.81±0.17	78.5±0.03	54.95±0.06	64.61±1.27	43.2±0.78
	FeSEM-CON-para	94.64±0.03	79.08±0.42	94.78±0.06	80.14±0.33	78.6±0.02	54.96±0.11	64.4±0.49	44.51±0.58
	WeCFL	94.21±1.67	79.31±11.02	94.05±1.67	81.41±5.7	69.47±4.16	34.1±7.79	66.8±6.39	45.61±5.9
	WeCFL-CON-rep	94.69±0.01	80.04±0.4	**95.86±0.1**	82.74±0.25	78.53±0.04	54.71±0.11	68.14±0.5	47.12±1.08
	WeCFL-CON-para	**95.38±0.03**	**81.7±1.02**	95.63±0.15	**83.77**±0.38	**78.85±0.02**	**55.78±0.1**	**69.22±0.1**	**48.68±0.42**

Summary

- Both CFL-CON-rep and CFL-CON-para can enhance performance under nearly all non-IID settings and across all datasets.

Table 4.4: Test results (mean±std) in **client**-wise non-IID settings on PathMNIST & TissueMNIST.

Datasets		PathMNIST				TissueMNIST			
Non-IID setting		$\alpha = 0.1$		2−class		$\alpha = 0.1$		2−class	
K	Methods	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
1	FedAvg	26.41±9.15	14.29±3.08	26.11±8.51	13.05±2.33	52.42±4.04	16.23±3.81	54.11±2.28	14.51±1.28
5	IFCA	38.13±2.53	25.22±1.74	34.16±3.76	22.52±1.13	38.76±10.94	20.38±2.01	49.31±13.97	21.51±3.68
	IFCA-CON-rep	52.02±3.71	25.44±3.53	52.44±3.63	29.05±1.63	66.3±6.7	25.56±5.37	60.54±3.19	21.15±0.61
	IFCA-CON-para	53.14±2.88	26.67±2.54	55.68±3.61	30.05±1.24	67.18±4.74	28.52±3.97	61.85±3.19	23.16±0.86
	FeSEM	59.85±1.45	33.5±4.08	64.46±6.12	38.41±3.19	72.88±1.11	33.19±1.7	70.62±2.41	28.43±2.54
	FeSEM-CON-rep	59.88±0.01	43.74±0.05	66.66±0.65	40.87±0.42	80.68±0.16	35.08±0.34	73.55±0.05	32.99±0.26
	FeSEM-CON-para	61.14±0.34	47.27±0.47	67.27±0.14	40.66±0.29	79.28±0.05	32.56±0.51	72.52±0.07	33.44±0.41
	WeCFL	67.91±1.35	41.08±3.13	66.37±7.19	41.34±4.12	75.58±4.78	37.02±0.93	72.93±1.72	31.83±5.73
	WeCFL-CON-rep	71.53±5.09	**51.97±3.79**	67.57±2.75	42.36±2.66	83.78±0.08	37.76±0.25	73.39±0.03	30.18±0.11
	WeCFL-CON-para	**74.09±0.93**	48.56±1.13	**68.55±0.57**	**43.53±0.54**	**83.98±0.11**	**39.63±0.45**	**74.0±0.03**	**34.02±0.26**
10	IFCA	51.88±13.67	27.81±2.21	37.22±4.23	20.2±2.04	27.44±16.39	16.37±10.45	41.87±20.04	21.59±7.3
	IFCA-CON-rep	59.87±8.61	40.35±3.67	48.01±9.57	30.33±1.62	55.14±7.36	16.59±1.91	72.83±9.91	24.79±3.37
	IFCA-CON-para	60.93±8.5	41.13±2.52	50.67±6.91	31.61±3.94	56.33±5.61	17.68±2.33	74.62±8.69	27.79±2.61
	FeSEM	78.93±4.27	52.94±5.42	70.93±4.27	52.94±5.42	78.85±2.29	52.32±7.59	77.92±1.53	45.68±6.71
	FeSEM-CON-rep	81.14±0.03	62.11±0.06	72.31±0.19	54.97±0.42	85.07±0.04	54.86±0.81	78.23±0.04	**47.88±0.46**
	FeSEM-CON-para	82.05±0.0	62.71±0.0	72.43±0.37	54.99±0.49	84.33±0.08	54.85±0.25	78.55±0.04	46.98±0.47
	WeCFL	80.27±3.01	52.63±3.59	71.37±1.5	53.78±2.21	79.05±3.06	52.67±6.2	78.62±1.77	46.86±5.46
	WeCFL-CON-rep	82.56±0.44	63.47±0.38	**73.21±0.13**	54.06±0.61	**85.35±0.04**	**55.33±0.24**	**79.45±0.1**	47.27±0.17
	WeCFL-CON-para	**83.85±0.14**	**64.51±0.1**	73.02±0.17	**55.51±0.47**	85.06±0.13	55.01±0.19	79.31±0.08	47.64±0.24

- Generally, the better the performance of the baseline, the better the performance when combined with CFL-CON. CFL-CON-para outperforms CFL-CON-rep in more cases. Overall, WeCFL-CON-para demonstrates the best performance.
- IFCA-CON improves IFCA the most due to its lower base and greater potential for enhancement. However, there are some exceptional cases for IFCA-CON under specific datasets and non-IID settings, which are primarily attributed to its inherent unstable clustering or mode collapse [91].

4.5 Conclusion

In this study, we tackle the problem of robust clustering in Clustered FL. In line with the core principles of clustering, which aim to maximize inter-cluster distances and minimize intra-cluster distances, we propose a contrastive approach. This method can either be viewed as a regularization term added to the loss function of Clustered FL methods or as an integral part of the unified framework for Clustered FL presented

in Chapter 3. We introduce two variants of CFL-CON: CFL-CON-rep and CFL-CON-para. Experimental results demonstrate significant marginal performance improvements under both cluster-wise and client-wise non-IID settings.

CHAPTER 5

CLUSTERED FEDERATED LEARNING WITH IMPROVED PERFORMANCE: A KNOWLEDGE SHARING APPROACH

5.1 Motivation

As addressed in Chapter 3, the unified framework of clustered FL with a bilevel optimization objective has been proposed. The theoretical analysis and experimental simulation are both conducted, in which more questions about clustered FL are raised.

Question 1: Should the clustering structure in FL be stable?

The answer is YES. Theoretically, the clustering results have to converge. Realistically, clustering is to group clients together based on some specific attitudes. Therefore unless the client changes, the clustering result should not change. And how to be stable is related to two challenges:

- How to represent the client or cluster?

- How to conduct clustering?

For the first challenge, we should use the representation metric that satisfies several conditions below.

- The metric should be privacy-protective to align with the basics of FL.
- The metric can distinguish clients as designed.

The metrics are mainly divided into two categories, model-based and data-based. Model-based representation metrics include the loss, partial or full model parameters, etc. Data-based metrics are usually related to the distribution of clients' data. For the metrics above, the loss is privacy-protective but not distinguishable. The distribution is distinguishable but may leak data privacy easily. The metrics based on model parameters are not absolutely secure, but they are at the same level of security as the trivial FL communicating with gradients. And they can cluster quickly and stably, according to the experiments.

And solutions to the second challenge depend highly on the metrics in the first challenge. For metrics in the format of vectors, classic clustering methods based on the angle or distance, such as K-means and its variants can be used. Specifically, the KL divergence or Wasserstein distance is more suitable for distribution-related metrics.

Question 2: If clustering is stable, FL for clients across clusters will be separate, then should clusters share knowledge with each other?

The answer is YES. As answered in the first question, the clustering results have to be stable in FL. For the hard clustering widely used in FL, one client belongs to only one cluster. Then clients in FL with clustering structures will be separated into several clusters with no relationships after a few communications. Although the overall performance is usually better than most FL methods with one globally-shared model

or even some personalized FL methods, as it is like doing several FL training based on homogeneous clients in one cluster, we argue that the performance can be significantly better with knowledge sharing across the clusters. This argument is one of the main advantages of FL, and the main motivation for this work as well.

Question 3: If inter-cluster knowledge has to be shared, which method should be used appropriately?

As claimed in the second question, inter-cluster knowledge sharing can contribute significantly to FL methods with clustering structures. This is not a new concept, especially in transfer learning, meta-learning, and distributed training such as multi-task learning (MTL) and FL. Many techniques have been developed to adapt to different methods, such as hard parameter sharing and soft parameter sharing, in which some are generic and others are specific. In conclusion, we can choose specific knowledge-sharing techniques for an assigned FL method. But in this paper, we try to propose a general method that can be an add-on to almost all FL methods with clustering structures to improve its performance or generalization ability. Then an inter-cluster regularization term is proposed.

5.2 Methodology

5.2.1 Formulation

For general FL, its objective can be formulated as below,

$$\min_{\Theta_g} \sum_{i=1}^{m} \psi_i \mathscr{L}(\mathscr{H}, D_i, \Theta_g). \tag{5.1}$$

In FedAvg, the global model Θ_g is aggregated from the local models,

$$\Theta_g = \sum_{i=1}^{m} \frac{\psi_i}{\sum_{j=1}^{m} \psi_j} \theta_i \tag{5.2}$$

For the clustered FL, as mentioned in WeCFL [62], it can be formulated into a bi-level optimization framework as follows,

$$\underset{\{\Theta_k\}}{\text{minimize}}\ \mathscr{F}: \frac{1}{m}\sum_{k=1}^{K}\sum_{i=1}^{m}\psi_i r_{i,k}\mathscr{L}(\mathscr{H}_k, D_i, \Theta_k) \tag{5.3a}$$

$$\textit{subject to}\ \{r_{i,k}\} = \underset{\{r_{i,k}\}}{\operatorname{argmin}}\mathscr{C}: \sum_{k=1}^{K}\sum_{i=1}^{m}\psi_i r_{i,k} d(g_i, G_k) \tag{5.3b}$$

As illustrated in Section 5.1, a well-performing federated learning (FL) algorithm for clustering should exhibit convergence or stability in its clustering results. However, this raises a concern regarding the extent to which the learned models utilize the knowledge of all clients or whether they learn sufficiently. This serves as the main motivation behind this paper, which proposes a straightforward yet effective approach to sharing knowledge across clusters. Specifically, a penalty term denoted by $\mathscr{S}$ with respect to Θ_k is added to the loss function of clustered FL, as depicted in equation 5.3a,

$$\underset{\{\Theta_k\}}{\text{minimize}}\ \mathscr{F}: \frac{1}{m}\sum_{k=1}^{K}\sum_{i=1}^{m}\psi_i r_{i,k}[\mathscr{L}(\mathscr{H}_k, D_i, \Theta_k) + \frac{\lambda_k}{2}\mathscr{S}(\{\Theta_k\})] \tag{5.4a}$$

$$\textit{subject to}\ \{r_{i,k}\} = \underset{\{r_{i,k}\}}{\operatorname{argmin}}\mathscr{C}: \sum_{k=1}^{K}\sum_{i=1}^{m}\psi_i r_{i,k} d(g_i, G_k). \tag{5.4b}$$

And a typical $\mathscr{P}$ to share knowledge across the clusters can be defined as,

$$\sum_{k'\in\{K\}\setminus\{k\}}\frac{\sum_{i=1}^{m} r_{i,k'}\psi_i}{\sum_{i=1}^{m}\psi_i}\|\Theta_k - \Theta_{k'}\|_2^2. \tag{5.5}$$

In order to simplify the formulation and reduce computational costs, it is possible to employ an approximation according to Theorem 5.2.4, which yields:

$$\|\Theta_k - \Theta_g\|_2^2, \tag{5.6}$$

which is easy to train. Since this method can be integrated into any clustered FL algorithms, it is referred to as CFL-CKS, or simply CKS in this thesis, shorted for

clustered **F**L with **c**lustered **k**nowledge **s**haring. As depicted in Figure 5.1, the models situated at the three corners denote the cluster model Θ_k, while the central model corresponds to the global model Θ_g. The grey bidirectional arrows depict the mode of knowledge sharing in Term 5.5. On the other hand, the green bidirectional arrows portray the mode of knowledge sharing in Term 5.6. These two modes of knowledge sharing are somewhat similar.

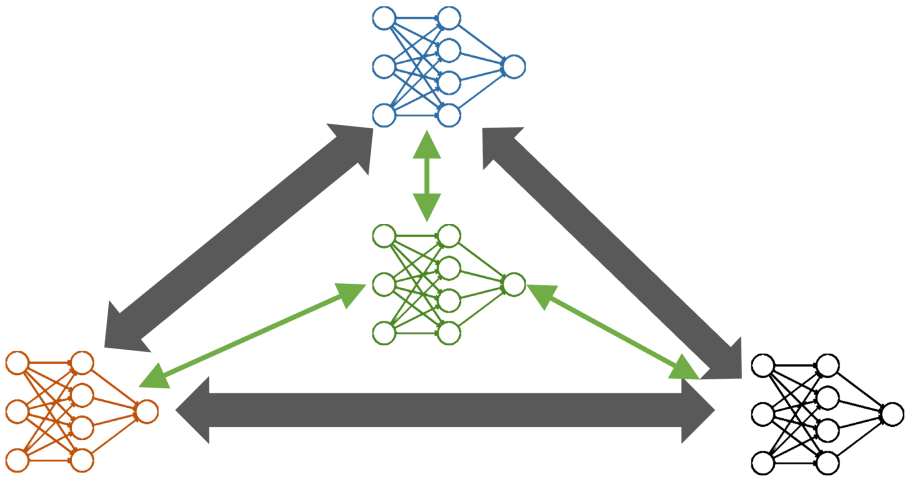

Figure 5.1: A toy example of CKS. The grey and green bidirectional arrows represent Term 5.5 and 5.6, respectively.

5.2.2 Theoretical Analysis

5.2.3 Equality Analysis

To analyze the equality of Term 5.5 and Term 5.6, for simplification, we use two assumptions below to simplify the analysis of Term 5.5.

Assumption 5.2.1. (Equal weight across clusters). Each cluster has the same sum of weights, which is $\frac{1}{K}$.

Assumption 5.2.2. (Equal distance across clusters and proportional with the distance to the center). The distance of each pair of clusters is equal as below and proportional to

the distance to the center for different $k, k_1, k_2 \in \{1, ..., K\}$,

$$\|\Theta_k - \Theta_{k_1}\|_2 = \|\Theta_k - \Theta_{k_2}\|_2 \tag{5.7}$$

$$\propto \|\Theta_k - \Theta_g\|_2. \tag{5.8}$$

Remark 5.2.3. Assumption 5.2.2 can be deemed as empirically observable and readily attainable. This is exemplified by Figure 5.2, which serves as a suitable illustration. The figure portrays a regular tetrahedron wherein the four vertices correspond to the four cluster centroids, while its barycenter denotes the global model.

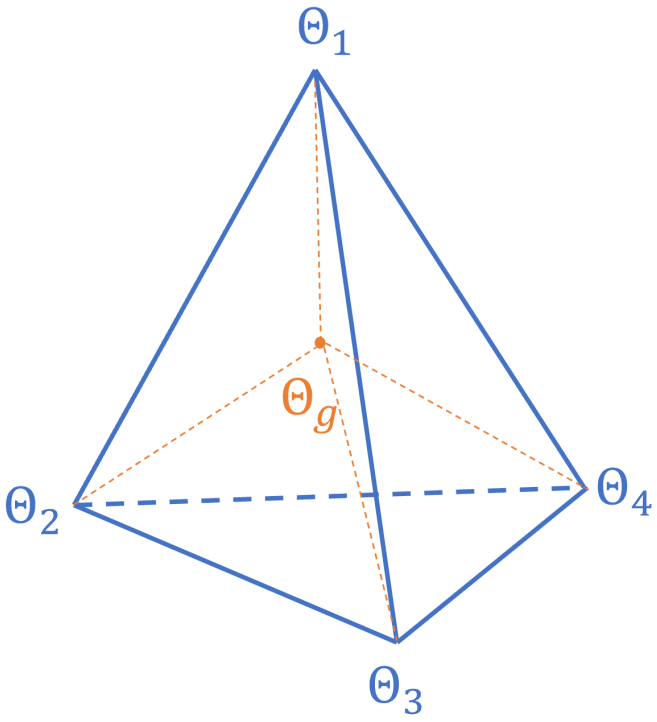

Figure 5.2: A toy example of Assumption 5.2.2.

Theorem 5.2.4. *(Proportionality of Term 5.5 and Term 5.6). Under Assumptions 5.2.1 and 5.2.2, Term 5.5 is proporitonal to Term 5.6.*

$$\sum_{k' \in \{K\} \setminus \{k\}} \frac{\sum_{i=1}^{m} r_{i,k'} \psi_i}{\sum_{i=1}^{m} \psi_i} \|\Theta_k - \Theta_{k'}\|_2^2 \propto \|\Theta_k - \Theta_g\|_2^2. \tag{5.9}$$

Proof. Under Assumptions 5.2.1 and 5.2.2,

$$\sum_{k' \in \{K\} \setminus \{k\}} \frac{\sum_{i=1}^{m} r_{i,k'} \psi_i}{\sum_{i=1}^{m} \psi_i} \|\Theta_k - \Theta_{k'}\|_2^2 \tag{5.10}$$

$$= \sum_{k' \in \{K\}} \frac{\sum_{i=1}^{m} r_{i,k'} \psi_i}{\sum_{i=1}^{m} \psi_i} \|\Theta_k - \Theta_{k'}\|_2^2 \tag{5.11}$$

$$= \frac{1}{K} \sum_{k' \in \{K\}} \|\Theta_k - \Theta_{k'}\|_2^2 \tag{5.12}$$

$$\propto \frac{1}{K} \frac{\|K\Theta_k - \sum_{k' \in \{K\}} \Theta_{k'}\|_2^2}{K} \tag{5.13}$$

$$\propto \frac{1}{K} \frac{K^2 \|\Theta_k - \frac{1}{K} \sum_{k' \in \{K\}} \Theta_{k'}\|_2^2}{K} \tag{5.14}$$

$$\propto \|\Theta_k - \Theta_g\|_2^2. \tag{5.15}$$

■

Remark 5.2.5. Transforming Term 5.5 to Term 5.6 is a computationally efficient and straightforward process. Moreover, the ratio between the two terms can be approximately equal to one.

5.2.4 Convergence Analysis

The complexity of analyzing the convergence of CFL-CKS arises from its integration with a bilevel optimization problem. Therefore, we reformulate Equation 5.4 to accommodate a more usual or specific scenario.

$$\begin{aligned} &\underset{\{\Theta_k\}}{\text{minimize}}\ \mathscr{F} : \frac{1}{\sum_{j=1}^{m} \psi_j} \sum_{k=1}^{K} \sum_{i=1}^{m} r_{i,k} \psi_i [\mathscr{L}(\Theta_k, D_i) + \frac{\lambda_k}{2} \|\Theta_k - \Theta_g\|_2^2] \\ &s.t.\ \{r_{i,k}\} = \underset{\{r_{i,k}\}}{\text{argmin}}\, \mathscr{C} : \frac{1}{\sum_{j=1}^{m} \psi_j} \sum_{k=1}^{K} \sum_{i=1}^{m} r_{i,k} \psi_i \|\theta_i - \Theta_k\|_2^2. \end{aligned} \tag{5.16}$$

Theorem 5.2.6. *(Convergence rate of CFL-CKS). Let Assumptions 3.4.1, 3.4.8, 3.4.9 and 3.4.10 hold, and* $\Delta = \mathscr{F}_0 - \mathscr{F}^*$*, given any* $\epsilon > 0$*, after*

$$T \geq \frac{\Delta}{Q(\epsilon(\eta - \frac{\beta\eta^2}{2}) - \frac{\beta\eta^2}{2}\sigma^2 - \eta B U^2)} \tag{5.17}$$

communication rounds, we have

$$\frac{1}{TQ}\sum_{t=0}^{T-1}\sum_{q=0}^{Q-1}\mathbb{E}[\|\nabla\mathscr{F}(\Theta_{1:K}^{(t,M,q)})\|_2^2] \leq \epsilon. \tag{5.18}$$

Remark 5.2.7. (Linear convergence rate of CFL-CKS). The integration of CKS does not affect the assumptions and convergence results of the bilevel optimization problem 5.16 still hold. As stated in Equation 5.17, given an appropriate learning rate, the convergence rate of CFL-CKS is $O(1/T)$, attaining a cutting-edge rate comparable to that of SGD and as described in [53]. Additionally, a larger K or reduced non-IIDness, along with a smaller B, results in an improved convergence rate but with diminishing returns.

Proof. Due to the convexity of CFL-CKS as follows,

$$\frac{\lambda_k}{2}\|\Theta_k - \Theta_g\|_2^2, \tag{5.19}$$

if we include this term into $\mathscr{L}$ and let Assumptions 3.4.1, 3.4.8, 3.4.9 and 3.4.10 still hold, The same results of Theorem 3.4.14 are achieved. ■

5.2.5 Interpretations

Regularization Regarding clustered FL as K independent tasks, while the clustering is stale in a few shots, each task can use regularization to alleviate overfitting and improve generalization ability. Unlike a traditional L2 norm to drag the parameters to zero, we use Term 5.6 to drag Θ_k to Θ_g.

FedProx [51] By regarding each cluster as a client and adding term 5.6, Objective 5.4 can be seen the clustered version of FedProx, which can be viewed as a generalization and re-parametrization of WeCFL [62]. And it can address the heterogeneous across clusters much better, and allows more robust convergence than WeCFL for realistic FL applications.

Multi-task Learning CKS in clustered FL can also be regarded as the soft parameter sharing in multi-task learning to some extent. Thus CKS can make one cluster gain benefits from other tasks or clusters.

5.3 Algorithm

The integration of CKS into clustered FL methods, such as IFCA, FeSEM, WeCFL, etc., is straightforward. Simply add Equation 5.6 to the FL loss function and follow the standard optimization process. The benefit of using stable clustering is that it only requires a limited number of shots for the clustering process. Then stable clustering can significantly reduce the amount of computational resources needed, leading to faster and more efficient training. Furthermore, stable clustering also helps to mitigate the risk of overfitting, as it reduces the number of updates needed for the clustering process. Thus, the algorithm can be modified as in Algorithm 3.

Algorithm 3: CFL-CKS: Clustered FL with Clustered Knowledge Sharing

1: **Input:** K, $\{D_1, D_2, \ldots, D_m\}$

2: **Initialize:** Randomly initialize $\{\Theta_1, \Theta_2, \ldots, \Theta_K\}$

3: **repeat**

4: **Expectation step for few shots:** Assign Client i to Cluster k by

$$k = \underset{k}{\operatorname{argmin}} \psi_i d(g_i, G_k).$$

5: **Maximization / Aggregation step:** Compute cluster center Θ_k by

$$\frac{1}{\sum_{i=1}^{m} \psi_i} \sum_{i=1}^{m} r_{i,k} \psi_i \theta_i.$$

6: **Distribution step:** Send H_k to clients in Cluster k.

7: **Local update step:** Run Gradient Descent Q steps using local data D_i to minimize

$$\frac{1}{m} \sum_{k=1}^{K} \sum_{i=1}^{m} \psi_i r_{i,k} [\mathscr{L}(\mathscr{H}_k, D_i, \Theta_k) + \frac{\lambda_k}{2} \mathscr{P}(\{\Theta_k\})].$$

8: **until** convergence condition satisfied

9: **Output:** $\{r_{i,k}\}, \{\Theta_1, \Theta_2, \ldots, \Theta_K\}$.

5.4 Experiments

5.4.1 Experimental settings

5.4.1.1 Datasets and partitioning

For details about the benchmark datasets and partition methods, please refer to Section A.1 and Section A.2, respectively, in Appendix A.

5.4.1.2 Baselines

For the global model-based FL, we choose FedAvg [67] as the baseline. For clustered FL methods, IFCA [30], FeSEM [94], WeCFL [62] are used. To evaluate the effectiveness of our proposed method, CFL-CKS are combined with these three baselines as three new baselines. The new baselines are named accordingly, such as WeCFL-CKS and WeCFL-CKS, for instance.

5.4.1.3 Simulation settings

Optimization settings For the training model, we use small CNNs [44] with two convolutional layers for Fashion-MNIST, CIFAR-10, PathMNIST and TissueMNIST as shown in Table A.1, A.2, A.3 and A.4 in Appendix , respectively. For the optimization, an optimizer of SGD with a learning rate of 0.001 and momentum of 0.9 is used to train the model, and the batch size is 32.

Evaluation metrics We evaluate the performance using both **micro accuracy** (%) and **macro F1-score** on the client-wise test datasets due to high non-IID degrees. The standard deviation is estimated from five repeats of the experiment with different random seeds, and the mean is obtained from the last three rounds out of the total 100 communication rounds.

Other settings For the FL settings, the local steps in each communication round are 10. For the clustering process, we use flattened parameters of the fully-connected layers of CNNs as data points and weighted K-Means as the clustering algorithm. The coefficient of the CFL-CKS term λ is chosen from a set of $\{1, 0.1, 0.01, 0.001\}$ based on the performance. For clustered FL with stable clustering, including FeSEM and WeCFL, the number of few shots is set to 10. For IFCA, where the clustering process is unstable, the clustering and optimization processes are always intertwined and occur

simultaneously. The coding framework called fedbase is used, which can be accessed via the PyPI repository * or GitHub †.

5.4.2 Experimental Analysis

Cluster-wise non-IID results analysis Table 5.1 and Table 5.2 demonstrate the comparison results under the cluster-wise non-IID setting over four datasets. For each baseline under different datasets and non-IID settings, CFL-CKS can improve the performance either in accuracy or macro-F1 in almost all cases. And usually, the better the performance of the original baseline, the better the performance of the baseline with CFL-CKS. For WeCFL, it performs the best, almost under all the settings. For IFCA, a clustering FL method that suffers from the problem of unstable clustering and "mode collapse" [91], CFL-CKS is difficult to improve its performance, will even decrease the performance especially under the non-IID setting with n-class method. It may reflect that there is little or waste knowledge being shared across the clusters of IFCA under the cluster-wise non-IID setting.

Client-wise non-IID results analysis Table 5.3 and Table 5.4 demonstrates the comparison results under the client-wise non-IID setting over four datasets. CFL-CKS can benefit all the baselines under various datasets and non-IID settings. For FeSEM and WeCFL, better original performance will usually lead to better performance with CFL-CKS. IFCA under the client-wise non-IID setting is an exceptional case. CFL-CKS can boost its performance so much, even to the best. It may be due to the fact that knowledge sharing is more easy or more efficient under the client-wise non-IID setting than the cluster-wise non-IID setting.

Summary

*https://pypi.org/project/fedbase/
†https://github.com/jie-ma-ai/FedBase

Table 5.1: Test results (mean±std) in **cluster**-wise non-IID settings on Fashion-MNIST & CIFAR-10.

Datasets		Fashion-MNIST				CIFAR-10			
Non-IID setting		$\alpha = (0.1, 10)$		(3,2)−class		$\alpha = (0.1, 10)$		(3,2)−class	
K	Methods	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
1	FedAvg	86.08±0.70	57.24±2.26	86.33±0.44	46.09±1.08	24.38±3.30	11.69±3.15	21.33±3.83	9.0±0.58
5	IFCA	84.60±2.22	62.03±3.01	84.94±2.54	66.50±4.43	34.1±4.79	22.12±2.21	29.80±4.49	17.90±2.08
	IFCA-CKS	85.14±2.46	62.82±0.45	80.74±5.28	60.31±3.14	33.54±8.76	19.46±4.38	32.69±4.32	19.35±1.26
	FeSEM	94.64±1.54	82.90±2.38	94.20±1.96	77.07±6.05	59.06±3.24	32.33±7.25	58.76±3.35	35.75±2.54
	FeSEM-CKS	95.74±0.26	85.86±1.0	95.35±2.0	77.62±6.28	60.38±7.47	29.99±6.03	62.83±0.11	42.6±0.96
	WeCFL	94.64±1.02	84.4±1.31	94.97±1.43	77.36±3.94	59.26±3.32	32.26±3.46	62.44±2.53	38.55±1.76
	WeCFL-CKS	**95.83±0.32**	**86.0±0.71**	**95.89±0.95**	**79.24±3.07**	**63.17±1.33**	**33.07±1.49**	**64.0±1.24**	**44.04±2.3**
10	IFCA	82.10±5.40	62.62±8.22	86.58±4.97	66.22±5.69	34.84±5.82	22.76±3.99	34.06±2.60	18.7±1.31
	IFCA-CKS	86.47±2.07	66.35±2.35	86.4±1.31	62.95±2.34	23.5±3.03	15.6±0.87	29.38±4.48	18.52±2.07
	FeSEM	95.73±1.28	89.34±1.57	95.54±0.74	84.43±2.38	66.89±2.18	38.35±4.24	71.76±2.23	49.72±3.84
	FeSEM-CKS	**96.65±1.28**	90.92±3.03	96.16±0.84	84.96±3.87	69.36±2.68	48.27±1.71	71.16±1.24	49.52±3.07
	WeCFL	95.88±0.85	89.81±1.59	97.10±0.51	88.96±1.36	70.95±3.57	40.19±2.88	72.13±1.88	**50.65±2.15**
	WeCFL-CKS	96.62±1.08	**91.36±3.41**	**97.24±0.97**	**90.95±3.52**	**71.5±2.46**	**48.49±1.41**	**72.68±0.51**	50.04±1.49

Table 5.2: Test results (mean±std) in **cluster**-wise non-IID settings on PathMNIST & TissueMNIST.

Datasets		PathMNIST				TissueMNIST			
Non-IID setting		$\alpha = (0.1, 10)$		(3,2)−class		$\alpha = (0.1, 10)$		(3,2)−class	
K	Methods	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
1	FedAvg	31.38±8.58	14.47±4.27	21.36±5.48	11.49±2.38	49.96±3.39	18.31±4.31	53.46±2.21	15.28±1.36
5	IFCA	38.13±2.53	25.22±1.74	34.16±3.76	22.52±1.13	27.44±16.39	16.37±10.45	41.87±20.04	21.59±7.3
	IFCA-CKS	39.82±6.85	27.06±2.33	38.71±1.24	23.05±0.5	25.9±3.86	14.47±2.19	55.33±1.02	23.98±1.25
	FeSEM	59.85±1.45	33.5±4.08	66.37±7.19	41.34±4.12	72.38±1.81	36.79±1.06	70.62±2.41	28.43±2.54
	FeSEM-CKS	74.77±1.96	39.35±5.21	65.25±0.71	42.76±1.32	79.98±3.8	35.1±3.31	75.1±8.37	**35.83±3.28**
	WeCFL	68.79±0.18	38.94±0.97	66.84±5.22	41.8±2.43	72.88±1.11	37.19±1.7	73.5±1.63	34.02±4.97
	WeCFL-CKS	**75.58±1.37**	**41.26±5.31**	**67.36±0.29**	**43.05±1.95**	**80.96±3.32**	**41.81±5.22**	**78.47±4.14**	35.26±3.64
10	IFCA	42.34±2.73	29.1±1.52	37.22±4.23	20.2±2.04	38.76±10.94	20.38±2.01	49.31±13.97	21.51±3.68
	IFCA-CKS	44.17±3.25	31.28±4.61	52.57±0.21	30.39±0.08	34.54±26.08	14.91±8.19	43.64±0.99	20.9±1.56
	FeSEM	79.31±0.72	48.14±0.23	71.37±1.5	53.78±2.21	77.12±1.68	47.69±3.1	77.92±1.53	45.68±6.71
	FeSEM-CKS	80.26±1.08	50.07±1.57	74.29±0.03	**61.08±0.03**	78.6±0.07	42.52±0.1	**80.81±0.86**	**55.98±2.21**
	WeCFL	81.88±2.43	50.17±1.05	73.19±2.0	55.53±3.51	77.37±1.12	48.5±4.1	78.32±1.5	48.58±5.28
	WeCFL-CKS	**83.06±1.96**	**53.43±2.15**	**74.34±0.48**	59.44±0.53	**84.14±1.04**	**52.27±0.24**	80.28±0.45	55.08±3.05

- CFL-CKS can enhance the performance of original clustered FL methods under almost all non-IID settings and across all datasets.
- Generally, the better the performance of the baseline, the better the performance when combined with CFL-CKS. Overall, WeCFL-CKS demonstrates the best performance.

Table 5.3: Test results (mean±std) in **client**-wise non-IID settings on Fashion-MNIST & CIFAR-10.

Datasets		Fashion-MNIST				CIFAR-10			
Non-IID setting		$\alpha = 0.1$		2−class		$\alpha = 0.1$		2−class	
K	Methods	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
1	FedAvg	85.9±0.46	54.52±2.66	86.17±0.25	44.88±1.24	25.62±3.47	11.38±2.02	24.3±3.53	8.56±0.64
5	IFCA	90.13±6.81	68.47±5.23	91.54±5.04	72.3±5.32	47.21± 10.28	22.67±1.48	46.54±12.8	17.78±1.29
	IFCA-CKS	93.0±0.14	72.05±1.08	**93.95±0.06**	**76.74±0.19**	53.85±0.22	26.87±0.11	56.72±0.87	28.56±0.5
	FeSEM	91.51±2.9	73.78±9.88	91.83±1.24	71.05±8.63	54.3±4.58	24.78±6.01	55.55±4.83	32.8±4.18
	FeSEM-CKS	93.99±0.02	73.97±0.23	91.38±0.03	68.67±0.05	**57.74±0.28**	32.35±0.5	51.32±0.24	28.03±0.49
	WeCFL	91.59±0.82	74.45±10.53	91.76±1.53	69.47±5.04	55.09±5.1	27.29±8.37	55.89±5.92	33.12±5.0
	WeCFL-CKS	**94.21±1.03**	**75.73±1.16**	92.72±1.05	71.52±0.23	57.59±0.01	**33.36±0.16**	**57.51±0.97**	**35.83±1.42**
10	IFCA	91.04±4.33	68.6±6.77	91.42±5.16	72.29±5.8	47.62±10.15	23.36±2.48	47.96±10.59	17.88±1.04
	IFCA-CKS	93.21±0.03	71.72±0.29	**95.52±0.51**	**86.08±0.59**	51.54±0.07	15.21±0.09	56.48±0.5	18.01±0.42
	FeSEM	93.3±2.0	80.47±11.05	93.75±1.53	79.39±6.57	67±1.57	31.69±8.52	63.64±6.51	42.97±6.08
	FeSEM-CKS	95.07±0.03	77.8±0.76	92.59±0.2	75.89±0.76	**77.28±0.05**	55.62±0.07	65.87±1.7	42.32±2.36
	WeCFL	94.21±1.67	79.31±11.02	94.05±1.67	81.41±5.7	69.47±4.16	34.1±7.79	66.8±6.39	45.61±5.9
	WeCFL-CKS	**95.28±1.03**	**80.93±2.3**	94.02±0.19	82.71±0.7	76.22±1.78	**55.77±2.11**	**68.92±1.26**	**48.7±2.0**

Table 5.4: Test results (mean±std) in **client**-wise non-IID settings on PathMNIST & TissueMNIST.

Datasets		PathMNIST				TissueMNIST			
Non-IID setting		$\alpha = 0.1$		2−class		$\alpha = 0.1$		2−class	
K	Methods	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
1	FedAvg	26.41±9.15	14.29±3.08	26.11±8.51	13.05±2.33	52.42±4.04	16.23±3.81	54.11±2.28	14.51±1.28
5	IFCA	38.13±2.53	25.22±1.74	34.16±3.76	22.52±1.13	38.76±10.94	20.38±2.01	49.31±13.97	21.51±3.68
	IFCA-CKS	66.77±0.07	23.4±0.24	**73.54±1.08**	37.9±0.43	80.22±1.04	27.01±0.81	74.24±1.03	24.56±0.65
	FeSEM	59.85±1.45	33.5±4.08	64.46±6.12	38.41±3.19	72.88±1.11	33.19±1.7	70.62±2.41	28.43±2.54
	FeSEM-CKS	60.95±1.79	43.37±0.92	65.93±0.83	39.48±1.0	80.14±0.72	37.56±1.76	74.41±0.54	30.87±0.25
	WeCFL	67.91±1.35	41.08±3.13	66.37±7.19	41.34±4.12	75.58±4.78	37.02±0.93	72.93±1.72	31.83±5.73
	WeCFL-CKS	**69.16±0.91**	**45.35±0.37**	68.83±1.75	**42.1±0.42**	**83.48±0.32**	**42.28±0.91**	**75.32±1.48**	**32.01±3.0**
10	IFCA	51.88±13.67	27.81±2.21	37.22±4.23	20.2±2.04	27.44±16.39	16.37±10.45	41.87±20.04	21.59±7.3
	IFCA-CKS	71.65±0.16	24.74±0.1	69.13±3.0	27.44±0.68	82.58±1.13	29.53±1.63	**80.31±2.51**	33.53±0.12
	FeSEM	78.93±4.27	52.94±5.42	70.93±4.27	52.94±5.42	78.85±2.29	52.32±7.59	77.92±1.53	45.68±6.71
	FeSEM-CKS	79.07±1.07	57.56±2.13	71.45±1.61	50.74±0.23	87.69±0.72	**55.06±1.09**	78.4±1.26	46.96±2.42
	WeCFL	80.27±3.01	52.63±3.59	71.37±1.5	53.78±2.21	79.05±3.06	52.67±6.2	78.62±1.77	46.86±5.46
	WeCFL-CKS	**80.76±1.01**	**59.8±2.61**	**72.94±1.29**	**54.93±2.18**	**88.24±1.27**	53.89±1.73	79.41±1.58	**47.84±2.29**

- IFCA-CKS improves IFCA the most due to its lower base and greater potential for enhancement. An exceptional case is that CFL-CKS can boost IFCA to the best level under the client-wise non-IID setting, but decrease its performance under the cluster-wise non-IID setting.

5.5 Conclusion

In this study, inspired by the nature of stable clustering in Clustered FL, we propose a clustered knowledge sharing method called CON-CKS. A simplified term, accompanied by a theoretical proof, is provided. This term can be incorporated into any loss function of Clustered FL methods or integrated into the unified framework presented in Chapter 3, while maintaining the linear convergence rate. Substantial performance improvement is demonstrated through extensive experiments, and the effectiveness of the approach is explained from three different perspectives.

CHAPTER 6

BRIDGING THE TRADE-OFF BETWEEN CONTRASTIVE LEARNING AND KNOWLEDGE SHARING WITHIN CLUSTERED FEDERATED LEARNING

6.1 Motivation

The CFL-CON of Chapter 4 and CFL-CKS of Chapter 5 techniques can both enhance the performance of Clustered FL but are rooted in divergent ideas. CFL-CON aims to increase the inter-cluster distance, whereas CFL-CKS aims to reduce it to facilitate knowledge sharing. Despite this contradiction, both approaches can improve performance. While both CFL-CON and CFL-CKS are additional techniques for clustered FL, this raises the question of whether we can find a way to combine and leverage the advantages of both methods simultaneously. After examining the problem, we divided the model parameters into two parts: the backbone and the head (or encoder and decoder). We concluded that sharing knowledge between the backbone of the clusters

is necessary, following the philosophy of Multi-task Learning, and that it is also important to maximize the inter-cluster distance of the head, following the philosophy of clustering. Overall, we integrated CFL-CON and CFL-CKS into CFL-CON&CKS, as depicted in Figure 6.1, where each network represents a cluster. The blue portion of the model denotes the backbone parameters, while the red portion denotes the head parameters. CFL-CKS is used to share knowledge between the blue portions, while CFL-CON is used to maximize the distance between the red portions.

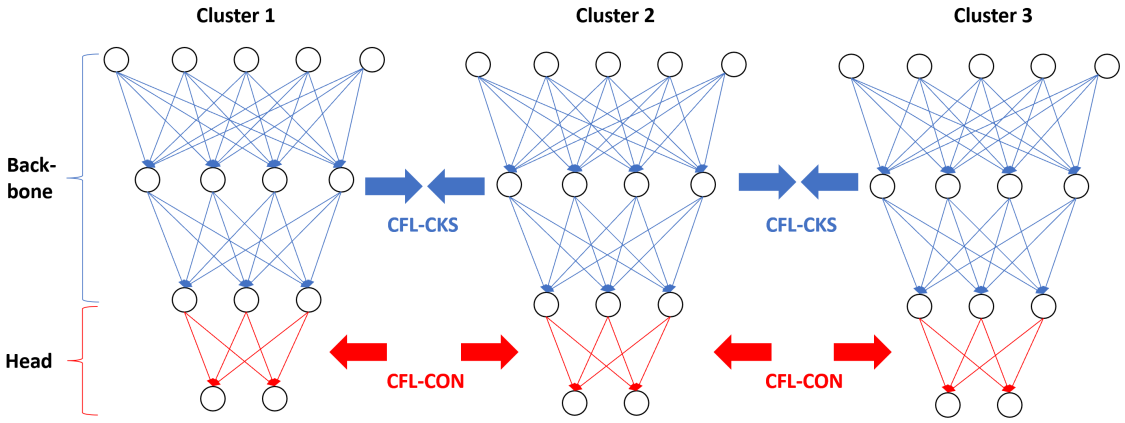

Figure 6.1: The framework of CFL-CON&CKS.

6.2 Methodology

The CFL-CON&CKS framework, illustrated in Figure 6.1, aims to improve the performance of Clustered Federated Learning by integrating the concepts of Contrastive Learning and Knowledge Sharing. To establish a precise and rigorous formulation for this approach, we first draw on the unified framework presented in Chapter 3. Then, we incorporate the specific formulations for CFL-CON and CFL-CKS, as outlined in Chapters 4 and 5, respectively.

CFL-CON is designed to increase the inter-cluster distance to improve performance, while CFL-CKS aims to reduce the distance to enable knowledge sharing across clusters.

Despite these seemingly contradictory objectives, we identify that it is essential to share knowledge between the encoders of clusters, while also maximizing the inter-cluster distance of the decoders, consistent with the principles of Multi-task Learning. To integrate these approaches, we define CFL-CON&CKS as Equation 6.1, outlined below.

$$\underset{\{\Theta_k\}}{\text{minimize}}\ \mathscr{F}: \frac{1}{m}\sum_{k=1}^{K}\sum_{i=1}^{m}\psi_i r_{i,k}[\mathscr{L}(\mathscr{H}_k, D_i, \Theta_k) + \mu\mathscr{T}(\{\Theta_{k,p}\}) + \frac{\lambda_k}{2}\mathscr{S}(\{\Theta_{k,r}\})] \tag{6.1a}$$

$$\textit{subject to}\ \{r_{i,k}\} = \underset{\{r_{i,k}\}}{\text{argmin}}\,\mathscr{C}: \sum_{k=1}^{K}\sum_{i=1}^{m}\psi_i r_{i,k} d(g_i, G_k). \tag{6.1b}$$

Overall, the CFL-CON&CKS framework provides a comprehensive approach to enhance the performance of Clustered FL, by leveraging the benefits of both Contrastive Learning and Knowledge Sharing.

6.3 Algorithm

The integration of CFL-CON&CKS into standard clustered FL methods is straightforward, similar to the inclusion of CFL-CON and CFL-CKS. By employing the WeCFL framework, we incorporate the CFL-CON&CKS term into the FL loss function, denoted as $\mathscr{F}$, as shown in Algorithm 4.

First, initialize the centroids of clusters $\Theta_1, \Theta_2, \ldots, \Theta_K$. Next, iteratively optimize the clustering loss function $\mathscr{C}$ and the FL loss function $\mathscr{F}$ until the convergence criteria are met. This process ensures seamless integration of CFL-CON&CKS, ultimately improving the overall performance of clustered FL methods.

Algorithm 4: CFL-CON&CKS: Integrate Contrastive Learning and Knowledge Sharing within Clustered FL

1: **Input:** K, $\{D_1, D_2, \ldots, D_m\}$

2: **Initialize:** Randomly initialize $\{\Theta_1, \Theta_2, \ldots, \Theta_K\}$

3: **repeat**

4: **Expectation step for few shots:** Assign Client i to Cluster k by

$$k = \underset{k}{\operatorname{argmin}}\, \psi_i d(g_i, G_k).$$

5: **Maximization / Aggregation step:** Compute cluster center Θ_k by

$$\frac{1}{\sum_{i=1}^{m} \psi_i} \sum_{i=1}^{m} r_{i,k} \psi_i \theta_i.$$

6: **Distribution step:** Send H_k to clients in Cluster k.

7: **Local update step:** Run Gradient Descent Q steps using local data D_i to minimize

$$\frac{1}{m} \sum_{k=1}^{K} \sum_{i=1}^{m} \psi_i r_{i,k} [\mathscr{L}(\mathscr{H}_k, D_i, \Theta_k) + \mu \mathscr{T}(\{\Theta_{k,p}\}) + \frac{\lambda_k}{2} \mathscr{S}(\{\Theta_{k,r}\})].$$

8: **until** convergence condition satisfied

9: **Output:** $\{r_{i,k}\}, \{\Theta_1, \Theta_2, \ldots, \Theta_K\}$.

6.4 Experiment

6.4.1 Experimental settings

6.4.1.1 Datasets and partitioning

For details about the benchmark datasets and partition methods, please refer to Section A.1 and Section A.2, respectively, in Appendix A.

6.4.1.2 Baselines

For the global model-based FL, we choose FedAvg [67] as the baseline. For clustered FL methods, we use IFCA [30], FeSEM [94], and WeCFL [62]. IFCA represents clustered FL methods that utilize minimum loss for clustering, while FeSEM and WeCFL represent clustered FL methods that employ partial model parameters for clustering. We then incorporate CFL-CON-para (shortened to CFL-CON), CFL-CKS, and CFL-CON&CKS into these three clustering methods to evaluate their effectiveness. The new baselines are named accordingly, such as FeSEM-CON, FeSEM-CKS, and FeSEM-CON&CKS, for instance.

6.4.1.3 Simulation settings

Optimization settings For the training model, we use small CNNs [44] with two convolutional layers for Fashion-MNIST, CIFAR-10, PathMNIST and TissueMNIST as shown in Table A.1, A.2, A.3 and A.4 in Appendix A, respectively. For the optimization, an optimizer of SGD with a learning rate of 0.001 and momentum of 0.9 is used to train the model, and the batch size is 32.

Evaluation metrics We evaluate the performance using both **micro accuracy** (%) and **macro F1-score** on the client-wise test datasets due to high non-IID degrees. The standard deviation is estimated from five repeats of the experiment with different random seeds, and the mean is obtained from the last three rounds out of the total 100 communication rounds.

Other settings For the FL settings, we perform ten local steps in each communication round. In the clustering process, we use flattened parameters of the fully-connected layers of CNNs as data points and employ weighted K-Means as the clustering algorithm. The coefficient of CFL-CKS λ is chosen from a set of $\{1, 0.1, 0.01, 0.001\}$, while the coefficient

of the CFL-CON term μ is selected from a set of $\{0.1, 0.5, 2, 5\}$, and the temperature of the CFL-CON term τ is chosen from a set of $\{0.1, 1, 10\}$ based on performance. For clustered FL methods with stable clustering, including FeSEM and WeCFL, we set the number of a few shots to 10. For IFCA, where the clustering process is unstable, the clustering and optimization processes are always intertwined and occur simultaneously. The coding framework called fedbase is used, which can be accessed via the PyPI repository * or GitHub †.

6.4.2 Experimental analysis

Cluster-wise non-IID Table 6.1 and 6.2 present the test results, including the mean and standard deviation of accuracy and Macro-F1 score for Fashion-MNIST, CIFAR-10, PathMNIST, and TissueMNIST under two cluster-wise non-IID settings. Evidently, CON&CKS exhibits improved performance compared to base methods and base methods employing either CON or CKS, particularly for distance-based clustered FL methods like FeSEM and WeCFL. For minloss-based clustered FL methods such as IFCA on Fashion-MNIST, IFCA-CON&CKS doesn't outperform IFCA-CON, potentially due to inconsistencies in the parameter space. As for FeSEM and WeCFL, CON&CKS doesn't exhibit a substantial improvement over CON and CKS, with most gains around 1%, though the improvement remains significant. This suggests that CON&CKS is an effective amalgamation of CON and CKS, capable of outperforming both individually.

Client-wise non-IID Table 6.3 and 6.4 present the test results for Fashion-MNIST, CIFAR-10, PathMNIST, and TissueMNIST under two client-wise non-IID settings, including both the mean and standard deviation of accuracy and Macro-F1 score. Despite the lack of a general clustering structure in the data, CON&CKS continues to exhibit

*https://pypi.org/project/fedbase/
†https://github.com/jie-ma-ai/FedBase

Table 6.1: Test results (mean±std) in **cluster**-wise non-IID settings on Fashion-MNIST & CIFAR-10.

	Datasets	Fashion-MNIST				CIFAR-10			
	Non-IID setting	$\alpha = (0.1, 10)$		(3,2)−class		$\alpha = (0.1, 10)$		(3,2)−class	
K	Methods	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
1	FedAvg	86.08±0.70	57.24±2.26	86.33±0.44	46.09±1.08	24.38±3.30	11.69±3.15	21.33±3.83	9.0±0.58
5	IFCA	84.60±2.22	62.03±3.01	84.94±2.54	66.50±4.43	34.1±4.79	22.12±2.21	29.80±4.49	17.90±2.08
	IFCA-CON	90.54±1.27	73.94±0.69	92.88±1.25	71.1±1.08	43.33±3.63	25.87±3.35	40.06±4.56	25.61±3.4
	IFCA-CKS	85.14±2.46	62.82±0.45	80.74±5.28	60.31±3.14	33.54±8.76	19.46±4.38	32.69±4.32	19.35±1.26
	IFCA-CON&CKS	88.93±3.06	68.32±2.16	88.76±4.98	60.36±2.74	47.07±5.92	26.25±3.33	40.27±5.27	16.62±1.43
	FeSEM	94.64±1.54	82.90±2.38	94.20±1.96	77.07±6.05	59.06±3.24	32.33±7.25	58.76±3.35	35.75±2.54
	FeSEM-CON	95.42±0.03	86.04±0.02	94.05±0.02	80.91±0.34	58.39±0.09	33.98±0.15	59.31±0.11	36.97±0.08
	FeSEM-CKS	95.74±0.26	85.86±1.0	95.35±2.0	77.62±6.28	60.38±7.47	29.99±6.03	62.83±0.11	42.6±0.96
	FeSEM-CON&CKS	96.24±1.49	89.29±1.15	93.9±0.6	76.98±1.93	60.82±1.59	34.06±0.41	63.24±2.17	43.33±1.04
	WeCFL	94.64±1.02	84.4±1.31	94.97±1.43	77.36±3.94	59.26±3.32	32.26±3.46	62.44±2.53	38.55±1.76
	WeCFL-CON	95.42±0.01	89.38±0.06	95.98±0.04	82.41±0.22	61.48±0.21	35.93±0.1	63.24±0.25	40.54±0.31
	WeCFL-CKS	95.83±0.32	86.0±0.71	95.89±0.95	79.24±3.07	63.17±1.33	33.07±1.49	64.0±1.24	44.04±2.3
	WeCFL-CON&CKS	**96.24±1.29**	**90.2±2.39**	**96.26±1.04**	**80.57±2.68**	**63.96±3.41**	**36.82±2.02**	**64.42±2.96**	**45.99±3.24**
10	IFCA	82.10±5.40	62.62±8.22	86.58±4.97	66.22±5.69	34.84±5.82	22.76±3.99	34.06±2.60	18.7±1.31
	IFCA-CON	93.25±1.43	80.54±1.75	89.63±2.59	59.63±3.57	48.65±4.87	27.53±4.71	43.52±5.28	32.0±2.96
	IFCA-CKS	86.47±2.07	66.35±2.35	86.4±1.31	62.95±2.34	23.5±3.03	15.6±0.87	29.38±4.48	18.52±2.07
	IFCA-CON&CKS	87.66±2.44	65.33±2.36	88.94±7.96	69.91±8.51	61.99±9.84	41.41±7.3	28.32±1.74	15.24±1.37
	FeSEM	95.73±1.28	89.34±1.57	95.54±0.74	84.43±2.38	66.89±2.18	38.35±4.24	71.76±2.23	49.72±3.84
	FeSEM-CON	95.78±0.02	89.99±0.16	96.71±0.05	82.55±0.13	69.88±0.07	36.0±0.09	72.92±0.08	48.98±0.15
	FeSEM-CKS	96.65±1.28	90.92±3.03	96.16±0.84	84.96±3.87	69.36±2.68	48.27±1.71	71.16±1.24	49.52±3.07
	FeSEM-CON&CKS	96.85±0.49	90.41±1.24	**97.34±0.89**	**91.95±0.84**	70.39±3.46	49.28±4.15	71.6±0.07	50.32±0.12
	WeCFL	95.88±0.85	89.81±1.59	97.10±0.51	88.96±1.36	70.95±3.57	40.19±2.88	72.13±1.88	50.65±2.15
	WeCFL-CON	96.93±0.04	91.22±0.12	97.18±0.06	91.88±0.32	71.23±0.09	42.34±0.11	72.73±0.05	51.57±0.21
	WeCFL-CKS	96.62±1.08	91.36±3.41	97.24±0.97	90.95±3.52	71.5±2.46	48.49±1.41	72.68±0.51	50.04±1.49
	WeCFL-CON&CKS	**97.71±0.58**	**91.88±1.36**	97.31±0.29	90.9±0.96	**72.31±4.8**	**49.98±7.27**	**73.18±0.4**	**52.15±0.8**

superior performance compared to the base methods and base methods enhanced with either CON or CKS alone. The transition from CON or CKS to CON&CKS brings a modest yet significant improvement. However, there are more instances where CON&CKS does not perform the best compared to its performance under cluster-wise non-IID settings. For instance, WeCFL-CKS outperforms CON&CKS in terms of the Macro-F1 score in $\alpha = 0.1$ on Fashion-MNIST, and IFCA-CKS surpasses CON&CKS in terms of accuracy in the $2-class$ setting on PathMNIST. This can be attributed to the lack of a well-defined clustering structure in client-wise non-IID data, making it more challenging to achieve efficient clustering. Nevertheless, the overall performance of CON&CKS remains commendable across different non-IID settings.

Summary

Table 6.2: Test results (mean±std) in **cluster**-wise non-IID settings on PathMNIST & TissueMNIST.

Datasets		PathMNIST				TissueMNIST			
Non-IID setting		$\alpha = (0.1, 10)$		(3,2)−class		$\alpha = (0.1, 10)$		(3,2)−class	
K	Methods	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
1	FedAvg	31.38±8.58	14.47±4.27	21.36±5.48	11.49±2.38	49.96±3.39	18.31±4.31	53.46±2.21	15.28±1.36
5	IFCA	38.13±2.53	25.22±1.74	34.16±3.76	22.52±1.13	27.44±16.39	16.37±10.45	41.87±20.04	21.59±7.3
	IFCA-CON	56.02±4.32	33.67±3.77	27.11±2.94	18.75±1.65	50.17±2.33	23.8±4.32	57.87±7.51	27.63±4.68
	IFCA-CKS	39.82±6.85	27.06±2.33	38.71±1.24	23.05±0.5	25.9±3.86	14.47±2.19	55.33±1.02	23.98±1.25
	IFCA-CON&CKS	53.77±6.8	33.29±2.37	51.52±7.08	22.66±3.11	34.72±3.63	20.22±3.01	56.52±9.54	26.14±2.35
	FeSEM	59.85±1.45	33.5±4.08	66.37±7.19	41.34±4.12	72.38±1.81	36.79±1.06	70.62±2.41	28.43±2.54
	FeSEM-CON	61.63±0.11	34.03±0.08	67.31±0.43	41.62±0.35	77.72±0.05	41.07±0.13	82.85±0.06	41.41±0.32
	FeSEM-CKS	74.77±1.96	39.35±5.21	65.25±0.71	42.76±1.32	79.98±3.8	35.1±3.31	75.1±8.37	35.83±3.28
	FeSEM-CON&CKS	75.67±1.37	40.91±1.98	67.03±0.72	44.89±1.84	80.26±0.31	41.21±0.2	81.77±3.61	41.01±2.0
	WeCFL	68.79±0.18	38.94±0.97	66.84±5.22	41.8±2.43	72.88±1.11	37.19±1.7	73.5±1.63	34.02±4.97
	WeCFL-CON	69.2±1.14	43.91±0.54	69.08±0.46	46.68±0.62	77.86±0.06	40.94±0.09	83.99±0.21	43.96±1.07
	WeCFL-CKS	77.58±1.37	45.26±5.31	67.36±0.29	43.05±1.95	80.96±3.32	41.81±5.22	78.47±4.14	35.26±3.64
	WeCFL-CON&CKS	**77.93±1.7**	**46.46±1.34**	**70.78±1.51**	**47.13±1.43**	**81.27±0.31**	**54.39±4.98**	**84.42±2.44**	**45.52±2.62**
10	IFCA	42.34±2.73	29.1±1.52	37.22±4.23	20.2±2.04	38.76±10.94	20.38±2.01	49.31±13.97	21.51±3.68
	IFCA-CON	43.09±1.97	30.25±2.51	30.02±8.11	17.94±4.48	78.69±8.62	35.19±4.75	55.61±3.91	27.1±3.05
	IFCA-CKS	44.17±3.25	31.28±4.61	52.57±0.21	30.39±0.08	34.54±26.08	14.91±8.19	43.64±0.99	20.9±1.56
	IFCA-CON&CKS	55.36±9.85	41.45±7.97	36.15±3.41	25.5±2.95	74.66±5.73	38.25±2.9	55.18±7.26	29.59±1.63
	FeSEM	79.31±0.72	48.14±0.23	71.37±1.5	53.78±2.21	77.12±1.68	47.69±3.1	77.92±1.53	45.68±6.71
	FeSEM-CON	79.45±0.01	48.03±0.0	75.84±0.45	55.94±0.88	87.27±0.1	47.56±0.14	78.81±0.09	53.82±0.35
	FeSEM-CKS	80.26±1.08	50.07±1.57	74.29±0.03	61.08±0.03	78.6±0.07	42.52±0.1	80.81±0.86	55.98±2.21
	FeSEM-CON&CKS	80.8±1.53	51.01±1.08	**76.94±1.64**	**63.69±1.7**	**88.97±0.04**	**50.76±0.15**	81.11±0.05	55.39±0.2
	WeCFL	81.88±2.43	50.17±1.05	73.19±2.0	55.53±3.51	77.37±1.12	48.5±4.1	78.32±1.5	48.58±5.28
	WeCFL-CON	81.75±1.14	51.73±1.47	74.67±0.41	52.33±0.49	87.52±0.03	47.36±0.09	79.72±0.13	54.81±0.22
	WeCFL-CKS	83.06±1.96	53.43±2.15	74.34±0.48	59.44±0.53	84.14±1.04	52.27±0.24	80.28±0.45	55.08±3.05
	WeCFL-CON&CKS	**85.45±1.87**	**55.92±1.58**	75.1±0.43	60.64±1.26	88.48±0.87	50.61±2.19	**81.37±0.44**	**57.19±3.95**

- In general, CFL-CON&CKS outperforms the base methods as well as the base methods enhanced with CFL-CON or CFL-CKS, in terms of both accuracy and Macro-F1 score across various datasets and non-IID settings.
- Although the improvement brought about by CFL-CON&CKS over CFL-CON and CFL-CKS individually may be small, it is nonetheless significant, confirming its efficiency as a combination of the two.
- CFL-CON&CKS exhibits slightly better performance in cluster-wise non-IID settings compared to client-wise non-IID settings, which could be attributed to the presence or absence of a clear clustering structure.

Table 6.3: Test results (mean±std) in **client**-wise non-IID settings on Fashion-MNIST & CIFAR-10.

Datasets		Fashion-MNIST				CIFAR-10			
Non-IID setting		$\alpha = 0.1$		2−class		$\alpha = 0.1$		2−class	
K	Methods	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
1	FedAvg	85.9±0.46	54.52±2.66	86.17±0.25	44.88±1.24	25.62±3.47	11.38±2.02	24.3±3.53	8.56±0.64
5	IFCA	90.13±6.81	68.47±5.23	91.54±5.04	72.3±5.32	47.21± 10.28	22.67±1.48	46.54±12.8	17.78±1.29
	IFCA-CON	90.89±3.51	65.57±4.91	91.43±2.97	66.15±4.33	49.54±3.92	24.67±1.68	54.09±3.13	23.61±1.18
	IFCA-CKS	93.0±0.14	72.05±1.08	93.95±0.06	76.74±0.19	53.85±0.22	26.87±0.11	56.72±0.87	28.56±0.5
	IFCA-CON&CKS	92.5±1.92	56.96±1.04	93.01±5.25	62.13±3.66	54.5±2.97	24.15±1.73	58.16±2.01	26.88±0.8
	FeSEM	91.51±2.9	73.78±9.88	91.83±1.24	71.05±8.63	54.3±4.58	24.78±6.01	55.55±4.83	32.8±4.18
	FeSEM-CON	91.42±0.11	70.02±0.41	92.0±0.04	72.1±0.17	57.56±1.84	34.31±1.94	56.41±1.76	32.77±0.12
	FeSEM-CKS	93.99±0.02	73.97±0.23	91.38±0.03	68.67±0.05	57.74±0.28	32.35±0.5	51.32±0.24	28.03±0.49
	FeSEM-CON&CKS	**94.75±0.06**	75.24±0.41	93.78±0.03	72.68±0.23	61.76±0.25	35.16±0.19	56.2±0.89	33.09±1.23
	WeCFL	91.59±0.82	74.45±10.53	91.76±1.53	69.47±5.04	55.09±5.1	27.29±8.37	55.89±5.92	33.12±5.0
	WeCFL-CON	91.65±0.06	74.79±0.11	92.75±0.04	72.37±0.18	58.69±0.55	36.17±0.55	56.26±1.34	35.07±0.59
	WeCFL-CKS	94.21±1.03	**75.73±1.16**	92.72±1.05	71.52±0.23	57.59±0.01	33.36±0.16	57.51±0.97	35.83±1.42
	WeCFL-CON&CKS	94.43±0.08	75.27±0.14	**94.38±0.06**	**74.03±0.13**	**61.8±0.43**	**36.69±0.2**	**59.24±1.47**	**36.06±1.37**
10	IFCA	91.04±4.33	68.6±6.77	91.42±5.16	72.29±5.8	47.62±10.15	23.36±2.48	47.96±10.59	17.88±1.04
	IFCA-CON	86.36±5.41	51.5±3.25	91.4±4.91	59.06±5.81	33.98±7.08	12.14±0.78	40.67±4.77	13.45±0.46
	IFCA-CKS	93.21±0.03	71.72±0.29	95.52±0.51	86.08±0.59	51.54±0.07	15.21±0.09	56.48±0.5	18.01±0.42
	IFCA-CON&CKS	93.33±3.61	72.47±4.35	92.88±4.36	71.52±5.91	48.69±8.1	24.55±5.61	57.58±2.31	24.91±1.18
	FeSEM	93.3±2.0	80.47±11.05	93.75±1.53	79.39±6.57	67±1.57	31.69±8.52	63.64±6.51	42.97±6.08
	FeSEM-CON	94.64±0.03	79.08±0.42	94.78±0.06	80.14±0.33	78.6±0.02	54.96±0.11	64.4±0.49	44.51±0.58
	FeSEM-CKS	95.07±0.03	77.8±0.76	92.59±0.2	75.89±0.76	77.28±0.05	55.62±0.07	65.87±1.7	42.32±2.36
	FeSEM-CON&CKS	**95.94±0.03**	80.45±0.32	95.09±0.1	80.61±0.21	**78.91±0.07**	56.31±0.31	68.68±0.04	47.39±0.13
	WeCFL	94.21±1.67	79.31±11.02	94.05±1.67	81.41±5.7	69.47±4.16	34.1±7.79	66.8±6.39	45.61±5.9
	WeCFL-CON	95.38±0.03	**81.7±1.02**	95.63±0.15	83.77±0.38	78.85±0.02	55.78±0.1	69.22±0.1	48.68±0.42
	WeCFL-CKS	95.28±1.03	80.93±2.3	94.02±0.19	82.71±0.7	76.22±1.78	55.77±2.11	68.92±1.26	48.7±2.0
	WeCFL-CON&CKS	95.02±0.11	80.27±0.17	**96.41±0.07**	**84.37±0.19**	78.64±0.04	**58.29±0.05**	**70.16±0.13**	**49.05±0.2**

6.5 Conclusion

In this study, we compare two methods, CFL-CKS and CFL-CON, which can both be integrated into the unified framework of Clustered FL. Despite sharing the same objective of enhancing CFL, their underlying philosophies are fundamentally different, even opposite, implying that they may not be directly combined to improve the performance of CFL. To address this challenge, we conduct a detailed examination of both methods and the neural network structure. Consequently, we apply CFL-CKS to the backbone of the neural network and CFL-CON to the head. This results in a new hybrid approach, CFL-CON&CKS, which combines the advantages of both methods for clustered FL. Experimental outcomes reveal significant marginal improvements in performance and robustness under both cluster-wise and client-wise non-IID settings.

Table 6.4: Test results (mean±std) in **client**-wise non-IID settings on PathMNIST & TissueMNIST.

	Datasets	PathMNIST				TissueMNIST			
	Non-IID setting	$\alpha = 0.1$		2−class		$\alpha = 0.1$		2−class	
K	Methods	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
1	FedAvg	26.41±9.15	14.29±3.08	26.11±8.51	13.05±2.33	52.42±4.04	16.23±3.81	54.11±2.28	14.51±1.28
5	IFCA	38.13±2.53	25.22±1.74	34.16±3.76	22.52±1.13	38.76±10.94	20.38±2.01	49.31±13.97	21.51±3.68
	IFCA-CON	53.14±2.88	26.67±2.54	55.68±3.61	30.05±1.24	67.18±4.74	28.52±3.97	61.85±3.19	23.16±0.86
	IFCA-CKS	66.77±0.07	23.4±0.24	**73.54±1.08**	37.9±0.43	80.22±1.04	27.01±0.81	74.24±1.03	24.56±0.65
	IFCA-CON&CKS	59.51±3.28	28.33±3.49	67.34±3.41	33.46±2.09	80.19±3.62	33.25±3.76	70.67±6.13	23.36±1.29
	FeSEM	59.85±1.45	33.5±4.08	64.46±6.12	38.41±3.19	72.88±1.11	33.19±1.7	70.62±2.41	28.43±2.54
	FeSEM-CON	61.14±0.34	47.27±0.47	67.27±0.14	40.66±0.29	79.28±0.05	32.56±0.51	72.52±0.07	33.44±0.41
	FeSEM-CKS	60.95±1.79	43.37±0.92	65.93±0.83	39.48±1.0	80.14±0.72	37.56±1.76	74.41±0.54	30.87±0.25
	FeSEM-CON&CKS	63.23±2.11	48.96±1.69	68.56±0.55	41.92±0.64	83.18±0.15	39.68±0.5	74.58±0.03	35.69±0.09
	WeCFL	67.91±1.35	41.08±3.13	66.37±7.19	41.34±4.12	75.58±4.78	37.02±0.93	72.93±1.72	31.83±5.73
	WeCFL-CON	74.09±0.93	48.56±1.13	68.55±0.57	43.53±0.54	83.98±0.11	39.63±0.45	74.0±0.03	34.02±0.26
	WeCFL-CKS	69.16±0.91	45.35±0.37	68.83±1.75	42.1±0.42	83.48±0.32	42.28±0.91	75.32±1.48	32.01±3.0
	WeCFL-CON&CKS	**76.45±1.6**	**51.17±0.94**	69.33±0.56	**44.94±0.5**	**84.32±0.28**	**45.58±0.84**	**75.55±0.04**	**35.46±0.29**
10	IFCA	51.88±13.67	27.81±2.21	37.22±4.23	20.2±2.04	27.44±16.39	16.37±10.45	41.87±20.04	21.59±7.3
	IFCA-CON	60.93±8.5	41.13±2.52	50.67±6.91	31.61±3.94	56.33±5.61	17.68±2.33	74.62±8.69	27.79±2.61
	IFCA-CKS	71.65±0.16	24.74±0.1	69.13±3.0	27.44±0.68	82.58±1.13	29.53±1.63	80.31±2.51	33.53±0.12
	IFCA-CON&CKS	69.73±7.96	38.82±2.08	62.2±5.33	29.41±2.75	77.48±8.72	34.13±5.98	82.14±4.91	37.03±2.0
	FeSEM	78.93±4.27	52.94±5.42	70.93±4.27	52.94±5.42	78.85±2.29	52.32±7.59	77.92±1.53	45.68±6.71
	FeSEM-CON	82.05±0.0	62.71±0.0	72.43±0.37	54.99±0.49	84.33±0.08	54.85±0.25	78.55±0.04	46.98±0.47
	FeSEM-CKS	79.07±1.07	57.56±2.13	71.45±1.61	50.74±0.23	87.69±0.72	55.06±1.09	78.4±1.26	46.96±2.42
	FeSEM-CON&CKS	84.23±0.21	64.08±0.3	73.54±0.22	**57.91±0.39**	88.03±0.08	**58.48±0.58**	**79.75±0.06**	**48.69±0.23**
	WeCFL	80.27±3.01	52.63±3.59	71.37±1.5	53.78±2.21	79.05±3.06	52.67±6.2	78.62±1.77	46.86±5.46
	WeCFL-CON	83.85±0.14	64.51±0.1	73.02±0.17	55.51±0.47	85.06±0.13	55.01±0.19	79.31±0.08	47.64±0.24
	WeCFL-CKS	80.76±1.01	59.8±2.61	72.94±1.29	54.93±2.18	88.24±1.27	53.89±1.73	79.41±1.58	47.84±2.29
	WeCFL-CON&CKS	**84.29±0.01**	**65.79±0.03**	**74.5±0.14**	55.87±0.26	**89.14±0.09**	57.78±0.32	79.59±0.03	48.65±0.18

CHAPTER 7

CLUSTERED ADDITIVE MODELING FOR MORE STABLE CLUSTERED FEDERATED LEARNING

7.1 Motivation

As the biggest challenge in FL, non-IID clients in practice usually have rich structures that have not been explored by most existing FL methods. A common structure is clusters, i.e., heterogeneous clients can be grouped into several near-homogeneous clusters each composed of clients with similar distributions. In practice, clusters might be associated with geographic/age/income groups, affiliations, etc. Hence, we can train a server-side model for each cluster, hence mitigating the conflicts caused by heterogeneity. Unfortunately, clients' cluster memberships are usually undefined or inaccessible due to sensitive/private information and have to be jointly optimized with cluster-wise models, as recent clustered FL [30, 62, 66, 94] approaches do. They maintain K models $\Theta_{1:K}$ for K clusters and assigns one Θ_k to each client-i (with local data X_i and local model θ_i), e.g., by min-loss (Θ_k with the minimum loss on X_i) or

K-means (the nearest Θ_k to θ_i) criterion. Hence, $1 \leq K \leq m$ models can accommodate more heterogeneity than single-model FL but also allows knowledge sharing among similar clients, which is lacking if training m client models independently. Hence, it may reach a better trade-off between global consensus and local personalization in non-IID settings.

However, compared to the general non-IID assumption, *clustered FL's assumption might be too restrictive since it prohibits inter-cluster knowledge sharing and enforces every cluster-wise model's training to only depend on a few clients*. This is contradictory to the widely studied strategy that different tasks or domains can benefit from sharing low-level or partial representations. It is due to the gap between the assumption of "clustered data distributions" and the algorithms of "clustering models (represented by loss vectors or model weights)": they are not equal, and the latter is more restrictive. In other words, clients of different clusters can still benefit from feature/parameter sharing.

Moreover, *clustered FL usually suffers from optimization instability because dynamically changing models* can violate the static clustering assumption and lead to imbalanced cluster assignment, which affects $\Theta_{1:K}$ and local training in the future. In particular: (1) *Clustering collapse*, i.e., the clients assigned to one cluster keeps growing so "the rich becomes richer (i.e., the cluster-wise model becomes even stronger)" until reducing to single-model FL, as shown in Figure 1.3. This happens because most clients tend to first learn shared features before focusing on client-specific ones; (2) *Fragile to outliers* such as malicious clients that may dominate some clusters and push all other benign ones to one or a few clusters; (3) *Sensitive to initialization.* The process highly depends on initial and earlier cluster assignments since they determine which clients' local training starts from the same model.

Main Contributions. To overcome the above problems of clustered FL, we propose a novel clustered FL model termed "**C**lustered **A**dditive **M**odeling (**CAM**)". Compared

to clustered FL, CAM trains a global model Θ_g on top of the K clusters' models $\Theta_{1:K}$. Its prediction for client-i combines the outputs of Θ_g and the associated cluster $c(i)$'s model, i.e., $y = f(x;\Theta_g) + \mathscr{H}(x;\Theta_{c(i)})$. This simple additive model removes the restriction of clustered FL by letting all clients share a base model Θ_g. It enforces $\Theta_{1:K}$ to focus on learning the different features between clusters, hence mitigating "clustering collapse". Moreover, CAM tends to learn balanced clusters and determine the number of clusters automatically (by starting from more clusters and then zeroing out some of them). Furthermore, CAM is less vulnerable to outliers, which can be mainly captured by $\Theta_{1:K}$ and have less impact on the global model Θ_g. In addition, interactions between $\Theta_{1:K}$ and Θ_g make CAM less sensitive to initial cluster assignments since updating Θ_g also changes the clustering.

CAM is a general model-agnostic method that can modify any existing clustered FL methods. As examples, we apply CAM to two representative methods, i.e., IFCA [30] and FeSEM [94]. In the optimization of CAM, $\Theta_{1:K}$ and Θ_g aim to fit the residual of each other's prediction. To this end, we propose an efficiently clustered FL algorithm "**Fed-CAM**", which alternates between cluster assignment (server), local training (clients), and update of $\Theta_{1:K}$ and Θ_g (server). In experiments on several benchmarks in different non-IID settings, CAM significantly improves SOTA clustered FL methods. Moreover, we provide a convergence analysis of Fed-CAM algorithm.

7.2 Clustered Additive Modeling (CAM)

In this section, we introduce clustered additive modeling (CAM), which combines a global model and cluster-wise model prediction in FL. CAM conducts a joint optimization of the global and cluster-wise models defined by a cluster assignment criterion. In particular, we provide two examples of CAM using different cluster assignment criteria, i.e., min-loss and K-means, which have been adopted respectively by two SOTA clustered-FL methods,

i.e., IFCA and FeSEM. For each of them, we derive alternating optimization procedures (i.e., IFCA-CAM and FeSEM-CAM) that can be implemented in the FL setting using two parallel threads of local model training. At the end of this section, we unify both algorithms in a clustered FL algorithm Fed-CAM.

Notations. We assume that there are m clients and K clusters, where client-i has n_i examples and all clients have $n = \sum_{i=1}^{m} n_i$ examples. On the server side, we have a global model Θ_g and K cluster-wise models $\Theta_{1:K}$. On the client side, we train m cluster models $\theta^0_{1:m}$ used to update the global model Θ_g in FL and $\theta_{1:m}$ used to update the cluster-wise model $\Theta_{c(i)}$ assigned to each client-i, where $c(i)$ is its cluster label determined by the cluster assignment criterion $c(\cdot)$. We further define $C_k \triangleq \{i \in [m] : c(i) = k\}$ as the set of clients in cluster-k. For simplicity, we will use X_i and Y_i to respectively represent the local training data on client-i and their ground truths, and $\ell(Y_i, \mathscr{H}(X_i))$ denotes the batch loss of model $\mathscr{H}(\cdot)$ on (X_i, Y_i). A CAM model for client-i can be

$$F_i(\cdot) = f(\cdot; \Theta_g) + \mathscr{H}(\cdot; \Theta_{c(i)})). \tag{7.1}$$

For classification, $F_i(\cdot)$ produces logits and we can apply softmax to get the class probabilities.

7.2.1 IFCA-CAM: model performance-driven clustering

We extend the min-loss criterion used in IFCA [30] to CAM for cluster assignment, i.e., each client-i is assigned to the cluster-k whose model Θ_k leads to the minimal loss of CAM on client-i's data, i.e.,

$$c(i) = \arg\min_{k \in [K]} \ell(Y_i, f(X_i; \Theta_g) + \mathscr{H}(X_i; \Theta_k)). \tag{7.2}$$

IFCA-CAM optimizes Θ_g and $\Theta_{1:K}$ for minimizing the above minimal loss over all the m clients, i.e.,

$$\text{IFCA-CAM: } \min_{\Theta_g, \Theta_{1:K}} \sum_{i=1}^{m} \frac{n_i}{n} \min_{k \in [K]} \ell(Y_i, f(X_i; \Theta_g) + \mathscr{H}(X_i; \Theta_k)), \tag{7.3}$$

where the inner minimization performs the min-loss assignment in Eq. (7.2). We solve Eq. (7.3) by the following alternating minimization of cluster membership, cluster-wise models, and the global model.

(i) Cluster assignment by applying Eq. (7.2) to the latest Θ_g and $\Theta_{1:K}$. This yields $c(\cdot)$ and $C_{1:K}$.

(ii) Fixing Θ_g, we can optimize the cluster-wise models $\Theta_{1:K}$ by gradient descent:

$$\Theta_k \leftarrow \Theta_k - \eta \sum_{i \in C_k} \frac{n_i}{n} \nabla_{\Theta_k} \ell(Y_i, f(X_i; \Theta_g) + \mathscr{H}(X_i; \Theta_k)), \ \forall \ k \in [K]. \tag{7.4}$$

In FL, the gradient can be approximated by aggregating the model updates of local models θ_i from clients, whose training on the client side is: (1) initializing $\theta_i \leftarrow \Theta_{c(i)}$; (2) starting from the initialization, running E local epochs updating θ_i by

$$\theta_i \leftarrow \theta_i - \eta \nabla_{\theta_i} \ell(Y_i, f(X_i; \Theta_g) + \mathscr{H}(X_i; \theta_i)), \ \forall \ i \in [m]; \tag{7.5}$$

and (3) aggregating the local model update $\theta_i - \Theta_k$ from client $i \in C_k$ to update Θ_k, i.e.,

$$\Theta_k \leftarrow \left(1 - \sum_{i \in C_k} \frac{n_i}{\sum_{j \in C_k} n_j}\right) \Theta_k + \sum_{i \in C_k} \frac{n_i}{\sum_{j \in C_k} n_j} \theta_i. \tag{7.6}$$

(iii) Fixing $\Theta_{1:K}$, we can optimize the global model Θ_g by gradient descent:

$$\Theta_g \leftarrow \Theta_g - \eta \sum_{i \in [m]} \frac{n_i}{n} \nabla_{\Theta_g} \ell(Y_i, f(X_i; \Theta_g) + \mathscr{H}(X_i; \Theta_{c(i)})). \tag{7.7}$$

In FL, this gradient step can be approximated by aggregating the local models θ_i^0 (similar to FedAvg): (1) initializing $\theta_i^0 \leftarrow \Theta_g$; (2) running E local epochs training θ_i^0 by

$$\theta_i^0 \leftarrow \theta_i^0 - \eta \nabla_{\theta_i^0} \ell(Y_i, f(X_i; \theta_i^0) + \mathscr{H}(X_i; \Theta_{c(i)})), \ \forall \ i \in [m]; \tag{7.8}$$

and (3) aggregating the updated local models θ_i^0 of all the m clients to update Θ_g, i.e.,

$$\Theta_g \leftarrow \frac{1}{m} \sum_{i \in [m]} \frac{n_i}{n} \theta_i^0. \tag{7.9}$$

We can run two parallel threads of local training for θ_i^0 and θ_i for each client-i because their training in Eq. (7.8) and Eq. (7.5) does not depend on each other (but they both

depend on the cluster assignments in (i)). This is analogous to the simultaneous update algorithm (FedSim) in [70]. One may also consider an alternative update algorithm (which may enjoy a slightly faster convergence) that iterates (i)→(ii)→(i)→(iii) in each round. However, it doubles the communication rounds ((i) requires one communication round) and does not allow parallel local training. Since the alternative update does not show a significant empirical improvement over FedSim in [70], we mainly focus on the parallel one in the remainder of this chapter.

7.2.2 FeSEM-CAM: parameter similarity-based clustering

We follow a similar procedure of IFCA-CAM to derive FeSEM-CAM, which applies a K-means style clustering to the client models $\theta_{1:m}$, whose objective is minimizing the sum of squares of client-cluster distance, i.e.,

$$\min_{\Theta_{1:K}} \sum_{i=1}^{m} \frac{n_i}{n} \min_{j\in[K]} \|\theta_i - \Theta_j\|_2^2. \tag{7.10}$$

Hence, similar to FeSEM [94], FeSEM-CAM assigns the nearest cluster-wise model to each client and updates the cluster-wise models as the cluster centroids (i.e., K-means algorithm), i.e.,

$$c(i) = \arg\min_{k\in[K]} \|\theta_i - \Theta_k\|_2^2, \;\; \Theta_k \leftarrow \sum_{i\in C_k} \frac{n_i}{\sum_{j\in C_k} n_j} \theta_i. \tag{7.11}$$

We iterate the above K-means steps a few times until convergence in practice. FeSEM-CAM applies the K-means objective in Eq. (7.10) as a regularization to the loss of CAM model $\ell(Y_i, f(X_i;\Theta_g) + \mathscr{H}(X_i;\theta_i))$, i.e.,

$$\text{FeSEM-CAM:} \quad \min_{\Theta_g,\Theta_{1:K},\theta_{1:m}} \sum_{i=1}^{m} \frac{n_i}{n} \left[\ell(Y_i, f(X_i;\Theta_g) + \mathscr{H}(X_i;\theta_i)) + \frac{\lambda}{2} \min_{j\in[K]} \|\theta_i - \Theta_j\|_2^2 \right], \tag{7.12}$$

where the minimization w.r.t. $\Theta_{1:K}$ (with $\theta_{1:m}$ fixed) recovers the (weighted) K-means objective in Eq. (7.10). Unlike IFCA-CAM, where client model θ_i is an auxiliary/latent variable for FL not showing in the objective of Eq. (7.3), it is explicitly optimized in

Eq. (7.12). Similar to IFCA-CAM, we solve Eq. (7.3) by iterating the following alternating minimization steps (i)-(iii).

(i) K-means clustering that iterates Eq. (7.11) for a few steps until convergence, which yields $c(\cdot)$, $C_{1:K}$, and $\Theta_{1:K}$. The update of $\Theta_{1:K}$ is analogous to Eq. (7.6).

(ii) Fixing Θ_g, we optimize $\theta_{1:m}$ by client-side local gradient descent:

$$\theta_i \leftarrow (1-\lambda)\theta_i + \lambda\Theta_{c(i)} - \eta\frac{n_i}{n}\nabla_{\theta_i}\ell(Y_i, f(X_i;\Theta_g) + \mathscr{H}(X_i;\theta_i)),\ \forall\ i \in [m]. \tag{7.13}$$

The first two terms in Eq. (7.13) compute a linear interpolation between θ_i and its assigned cluster's model $\Theta_{c(i)}$. This is a result of the K-means regularization term in Eq. (7.3) and keeps θ_i close to $\Theta_{c(i)}$.

(iii) Fixing $\theta_{1:m}$, we can optimize Θ_g by gradient descent:

$$\Theta_g \leftarrow \Theta_g - \eta\sum_{i\in[m]}\frac{n_i}{n}\nabla_{\Theta_g}\ell(Y_i, f(X_i;\Theta_g) + \mathscr{H}(X_i;\theta_i)). \tag{7.14}$$

In FL, this gradient step can be approximated by aggregating the local models θ_i^0 (similar to FedAvg): (1) initializing $\theta_i^0 \leftarrow \Theta_g$; (2) running E local epochs training θ_i^0 by

$$\theta_i^0 \leftarrow \theta_i^0 - \eta\nabla_{\theta_i^0}\ell(Y_i, f(X_i;\theta_i^0) + \mathscr{H}(X_i;\theta_i)),\ \forall\ i \in [m]; \tag{7.15}$$

and (3) aggregating the updated local models θ_i^0 of all the m clients to update Θ_g by Eq. (7.9).

7.2.3 Algorithm

In Algoirithm 5, we propose a clustered FL algorithm for CAM, i.e., Fed-CAM, which can unify the derived optimization procedures for IFCA-CAM and FeSEM-CAM and can be easily extended to other clustered FL and clustering criteria.

Warmup. As an alternating optimization framework, it would be unstable if both Θ_g and $\Theta_{1:K}$ are randomly initialized and jointly optimized in parallel since they may capture overlapping information and result in an inefficient competitive game. To encourage

Algorithm 5: Fed-CAM

initialize : Randomly initialize $\Theta_{1:K}$ and Θ_g.

warmup : (1) w rounds of FedAvg to get an initial Θ_g (IFCA-CAM) or (2) w epochs of local training only to get a initial $\theta_{1:m}$ (FedSEM-CAM). Broadcast Θ_g and $\Theta_{1:K}$ to all clients.

while *not converge* **do**

  `/* Client (in parallel) */`

  for *every selected client* i **do**

    Model performance-driven clustering (e.g., IFCA-CAM): cluster assignment by Eq. (7.2);

    Initialize $\theta_i \leftarrow \Theta_{c(i)}$ and $\theta_i^0 \leftarrow \Theta_g$;

    Local training of θ_i for Q epochs: e.g., Eq. (7.5) (IFCA-CAM) or Eq. (7.13) (FeSEM-CAM);

    Local training of θ_i^0 for Q epochs: e.g., Eq. (7.8) (IFCA-CAM) or Eq. (7.15) (FeSEM-CAM);

    Upload θ_i and θ_i^0 to the server;

  `/* Server */`

  Update cluster-wise models $\Theta_{1:K}$: e.g., Eq. (7.6) (IFCA-CAM) or Eq. (7.11) (FeSEM-CAM);

  Update global model Θ_g by Eq. (7.9);

  Broadcast Θ_g and $\Theta_{1:K}$ to all clients;

output : Global model Θ_g, cluster-wise models $\Theta_{1:K}$ and $c(i)\ \forall\ i \in [m]$.

them to learn complementary knowledge, warmup training for one of them before the joint optimization is helpful. For example, a few rounds of FedAvg can produce a "warm" Θ_g, whose predictions' residuals are more informative to train $\Theta_{1:K}$. Another warmup strategy could be to run a few local training epochs and extract warm $\Theta_{1:K}$ by clustering the lightly-trained local models $\theta_{1:m}$. In Fed-CAM, we can apply the former warmup to IFCA-CAM and the latter to FeSEM-CAM.

7.3 Convergence Analysis

Based on the convergence analysis presented in [70], which aims to minimize the following objective:

$$\min_{u,V} \mathscr{F}(u,V) := \frac{1}{n}\sum_{i=1}^{m} \ell_i(u,v_i), \tag{7.16}$$

where u represents the shared parameters and $V = v_1, v_2, \cdots, v_m$ denotes the personalized parameters. If we map Θ_g to u, and $\Theta_{1:K}$ to V respectively, this appears strikingly similar to our methods as illustrated in Equations 7.3 and 7.12. Provided that the clustering remains stable, we can employ the theoretical framework of [70]. And firstly, we make some standard assumptions for the convergence analysis as below.

Assumption 7.3.1. (Smoothness). For $i = 1, \cdots, m$, the loss function l is continuously differentiable, and there exist constants β that $\nabla_{\Theta_g} \ell(\Theta_g, \Theta_k)$ is β-Lipschitz with respect to Θ_g and Θ_k, and $\nabla_{\Theta_k} \ell(\Theta_g, \Theta_k)$ is L-Lipschitz with respect to Θ_g and Θ_k.

Assumption 7.3.2. (Unbiased gradients and bounded variance). The stochastic gradients are unbiased and have bounded variance. For all Θ_g and Θ_k,

$$\mathbb{E}[\widetilde{\nabla}_{\Theta_g} \ell(\Theta_g, \Theta_k)] = \nabla_{\Theta_g} \ell(\Theta_g, \Theta_k), \;\; \mathbb{E}[\widetilde{\nabla}_{\Theta_k} \ell(\Theta_g, \Theta_k)] = \nabla_{\Theta_k} \ell(\Theta_g, \Theta_k),$$

and

$$\mathbb{E}[\|\widetilde{\nabla}_{\Theta_g} \ell(\Theta_g, \Theta_k) - \nabla_{\Theta_g} \ell(\Theta_g, \Theta_k)\|^2] \leq \sigma_g^2, \;\; \mathbb{E}[\|\widetilde{\nabla}_{\Theta_k} \ell(\Theta_g, \Theta_k) - \nabla_{\Theta_k} \ell(\Theta_g, \Theta_k)\|^2] \leq \sigma_k^2.$$

Assumption 7.3.3. (Partial gradient diversity). There exists a constant for all θ_i^0 and Θ_g, θ_i and Θ_k,

$$\sum_{i=1}^{m} \frac{n_i}{n} \|\nabla_{\Theta_g} \ell(\Theta_g, \theta_i) - \nabla_{\Theta_g} \ell(\Theta_g, \Theta_k)\|^2 \leq \delta^2$$

$$\sum_{i \in C_k} \frac{n_i}{\sum_{j \in C_k} n_j} \|\nabla_{\Theta_k} \ell(\theta_i^0, \Theta_k) - \nabla_{\Theta_k} \ell(\Theta_g, \Theta_k)\|^2 \leq \delta^2.$$

Assumption 7.3.4. (Convexity of cluster models). Fix Θ_g, assume $\ell(\Theta_k)$ is convex.

Claim 1. (Identical data distribution with one cluster for FedSEM-CAM). Assume that clients clustered into the same cluster have the same data distribution when clustering is stable, especially in FedSEM-CAM.

Remark 7.3.5. Claim 1 can be validated by experimental analysis of clustering in this paper easily, as FeSEM-CAM uses the parameters of the last layers for clustering, which contains label distribution information of clients.

Theorem 7.3.6. *(Convergence of Fed-CAM). Let Assumptions 7.3.1, 7.3.2, 7.3.3 and 7.3.4 hold, and learning rates chosen as $\eta = \zeta/(LE)$ for a ζ depending on the parameters $L, \sigma_g^2, \sigma_k^2, \delta^2, s, m, T$, provided clustering stable, we have (ignoring absolute constants),*

$$\frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{L}\mathbb{E}[\|\nabla_{\Theta_g}\mathscr{L}(\Theta_g,\Theta_k)\|^2] + \frac{s}{mL}\frac{1}{m}\sum_{i=1}^{m}\mathbb{E}[\|\nabla_{\Theta_{c(i)}}\ell(\Theta_g,\Theta_{c(i)})\|^2]\right) \tag{7.17}$$

$$\leq \frac{(\triangle_{\mathscr{L}}\sigma_{sim,1}^2)^{1/2}}{T^{1/2}} + \frac{(\triangle_{\mathscr{L}}^2\sigma_{sim,2}^2)^{1/3}}{T^{2/3}} + O(\frac{1}{T}), \tag{7.18}$$

where $\triangle_{\mathscr{L}} = \mathscr{L}_0 - \mathscr{L}^\star$, and we define effective variance terms,

$$\sigma_{sim,1}^2 = \frac{2}{L}(\delta^2(1-\frac{s}{m}) + \frac{\sigma_g^2}{L} + \frac{\sigma_k^2 s}{m})) \tag{7.19}$$

$$\sigma_{sim,2}^2 = \frac{2}{L}(\delta^2 + \sigma_g^2 + \sigma_k^2)(1-\frac{1}{E}). \tag{7.20}$$

Remark 7.3.7. It is straightforward to prove that the clustering of both IFCA-CAM and FeSEM-CAM converges, as evidenced by Ma et al. (2022). However, proving the stability of these clustering methods is more challenging due to the oscillation phenomenon often seen in K-means. The stability of clustering will be further demonstrated through experimental analysis in Section 7.4.3.

Remark 7.3.8. Besides the clustering structure, there is a distinct difference between FedSim [70] and Fed-CAM. In Fed-CAM, we need to aggregate both Θ_g and $\Theta_{1:K}$, while in FedAlt, only Θ_g requires aggregation. The $\sigma_{sim,1}^2$ and $\sigma_{sim,2}^2$ reflect the impact of sample number s and local steps E. Larger s or smaller E, better convergence rate. According to the results presented in [70], alternative gradient descent surpasses simultaneous gradient descent in terms of convergence rate. The asymptotic $1/\sqrt{T}$ rate is achieved when each device is seen at least once on average, and the $1/T$ term is dominated by the $1/\sqrt{T}$ term, a situation that occurs when (ignoring absolute constants)

$$T \geq \frac{\triangle_{\mathscr{L}}}{\sigma_{sim,1}^2} max\{(1-\frac{1}{E})\frac{m}{s}, 2\}.$$

Lemma 7.3.9. *(Bounding $\epsilon_{distriubte}$). Under the assumption of convexity of cluster models, and the claim of dentical data distribution with one cluster for FedSEM-CAM, we can get*

$$\epsilon_{distriubte} = \mathscr{L}(\Theta_g, \Theta_{c(i)}) - \mathscr{L}(\Theta_g, \theta_i, c(i)) \tag{7.21}$$

$$\leq 0. \tag{7.22}$$

Proof. For minloss-based methods, it is straightforward to prove that $\epsilon_{distriubte} \leq 0$. However, for distance-based methods like FeSEM-CAM, bounding $\epsilon_{distriubte}$ may require the introduction of a new bound in Lemma 4 of work [62]. According to the assumptions of convexity and identical distribution within one cluster, we have

$$\mathbb{E}_t[\mathscr{L}(\Theta_g, \Theta_{c(i)}) - \mathscr{L}(\Theta_g, \theta_i)] \tag{7.23}$$

$$= \sum_{k=1}^{K} \sum_{c(i)=k} \frac{n_i}{n} \mathbb{E}_t[\ell(\Theta_g, \Theta_{c(i)}) - \ell(\Theta_g, \theta_i)] \tag{7.24}$$

$$= \sum_{k=1}^{K} \frac{n_k}{n} \sum_{c(i)=k} \frac{n_i}{n_k} \mathbb{E}_t[\ell(\Theta_g, \sum_{i \in C_k} \frac{n_i}{n_k} \theta_i) - \ell(\Theta_g, \theta_i)] \tag{7.25}$$

$$\leq 0, \tag{7.26}$$

where n_k is the number of clients in Cluster k, and $\sum_{k=1}^{K} n_k = n$. ■

The proof of Theorem 7.3.6 is as below.

Proof. Firstly, we simplify the objective function to minimize as below,

$$\mathscr{L} = \frac{1}{m} \sum_{i=1}^{m} \ell(\Theta_g, \Theta_{c(i)}). \tag{7.27}$$

Then the proof outline is as follows,

$$\mathscr{L}(\Theta_g^{(t+1)}, \Theta_{c'(i)}^{(t+1)}) - \mathscr{L}(\Theta_g^{(t)}, \Theta_{c(i)}^{(t)}) \tag{7.28}$$

$$= \underbrace{\mathscr{L}(\Theta_g^{(t+1)}, \theta_i^{(t+1)}, c(i)) - \mathscr{L}(\Theta_g^{(t)}, \Theta_{c(i)}^{(t)})}_{\epsilon_{fedsim}} + \underbrace{\mathscr{L}(\Theta_g^{(t+1)}, \theta_i^{(t+1)}, c'(i)) - \mathscr{L}(\Theta_g^{(t+1)}, \theta_i^{(t+1)}, c(i))}_{\epsilon_{cluster}} \tag{7.29}$$

$$+ \underbrace{\mathscr{L}(\Theta_g^{(t+1)}, \Theta_{c'(i)}^{(t+1)}) - \mathscr{L}(\Theta_g^{(t+1)}, \theta_i^{(t+1)}, c'(i))}_{\epsilon_{distribute}}, \tag{7.30}$$

where $c'(i)$ represents the assign relationships of round $t+1$ compared to $c(i)$ of round t and

$$\Theta_{c'(i)}^{(t+1)} \leftarrow \sum_{i \in C_k} \frac{n_i}{\sum_{i \in C_k} n_i} \theta_i^{(t+1)}, \ \forall \ c'(i) = k. \tag{7.31}$$

Bounding ϵ_{fedsim}. In this process, $c(i)$ does not change. Fed-CAM can be seen doing parameter sharing for one global and parameter personalization for clients in clusters. So this process is equal to FedSim [70], then we have,

$$\mathscr{L}(\Theta_g^{(t+1)}, \theta_i^{(t+1)}) - \mathscr{L}(\Theta_g^{(t)}, \Theta_{c(i)}^{(t)}) \tag{7.32}$$

$$\leq \underbrace{\langle \nabla_{\Theta_g} \mathscr{L}(\Theta_g^{(t)}, \Theta_{c(i)}^{(t)}), \Theta_g^{(t+1)} - \Theta_g^{(t)} \rangle}_{\epsilon_{1,g}} + \underbrace{\sum_{i=1}^{m} \langle \nabla_{\Theta_k} \ell(\Theta_g^{(t)}, \Theta_{c(i)}^{(t)}), \theta_i^{(t+1)} - \Theta_{c(i)}^{(t)} \rangle}_{\epsilon_{1,i}} \tag{7.33}$$

$$+ \underbrace{L \|\Theta_g^{(t+1)} - \Theta_g^{(t)}\|^2}_{\epsilon_{2,g}} + \underbrace{\sum_{i=1}^{m} L \|\theta_i^{(t+1)} - \Theta_{c(i)}^{(t)}\|^2}_{\epsilon_{2,i}}. \tag{7.34}$$

By mapping $\epsilon_{1,g}, \epsilon_{2,g}, \epsilon_{1,i}, \epsilon_{2,i}$ to $\tau_{1,u}, \tau_{2,u}, \tau_{1,v}, \tau_{1,v}$ respectively in the convergence proof for FedSim, with Claim 14, 15, 16, 17 in [70], we will obtain the same bound.

Bounding $\epsilon_{cluster}$. In this bounding step, we assign a new cluster for all clients, but distribute the cluster model later. Therefore $c(i)$ changes to $c'(i)$, but θ_i keeps the same. And we got

$$\mathscr{L}(\Theta_g^{(t+1)}, \theta_i^{(t+1)}, c'(i)) - \mathscr{L}(\Theta_g^{(t+1)}, \theta_i^{(t+1)}, c(i)) = 0 \tag{7.35}$$

Bounding $\epsilon_{distriubte}$. According to Lemma 7.3.9, we have $\epsilon_{distriubte} \leq 0$.

Finally, combining ϵ_{fedsim}, $\epsilon_{cluster}$, $\epsilon_{distribute}$, taking full expectation and telescoping over $t = 1, \ldots, T$, we have the same error bound and convergence rate with FedSim. ■

7.4 Experiments

7.4.1 Experimental Settings

Benchmark datasets and partitions For details about the benchmark datasets and partition methods, please refer to Section A.1 and Section A.2, respectively, in Appendix A.

Baselines We select baseline methods from four categories as follows:

- **Single model-based FL:** We choose FedAvg [67] and FedProx [51] with a coefficient of 230 and a regularization of 0.95 as the baselines.
- **Ensemble FL:** We train FedAvg and FedProx K times and then learn an ensemble model via soft voting to serve all clients, which are named FedAvg+ and FedProx+, respectively.
- **Clustered FL:** We choose FeSEM [94] and IFCA [30], which is similar to HypCluster [66].
- **Clustered FL with additive modeling:** We integrate CAM with IFCA and FeSEM, denoting them as IFCA-CAM and FeSEM-CAM, respectively.

Learning-related hyperparameters We use the Convolutional Neural Network (CNN) [44] as the basic model architecture for each client, as detailed in Appendix A. For optimization, we employ SGD with a learning rate of 0.001 and momentum of 0.9 to train the model, and the batch size is 32. We evaluate the performance using both **micro accuracy** (%) and **macro F1-score** (%) on the client-wise test datasets to better capture the non-IID nature per client.

FL system settings We conduct 100 global communication rounds in the FL system, including 30 warmup rounds if applicable. Each communication involves ten local steps. For the clustering process of FeSEM-CAM, we measure distance on the flattened parameters of the fully-connected layers and use K-Means as the clustering algorithm. The coefficient λ is chosen from $0.001, 0.01, 0.1$ based on the best performance.

7.4.2 Main Results and Comparisons

Cluster-wise non-IID scenarios make the assumption that there are underlying clustering structures among clients. Table 7.1 and 7.2 compare the methods using two benchmark datasets, namely Fashion-MNIST and CIFAR-10, PathMNIST and TissueMNIST, respectively. Figure 7.1 demonstrates the improvement of both accuracy and macro-f1 score under the $\beta = (0.1, 10)$ cluster-wise non-IID setting using CIFAR-10. Results using two biomedical datasets are presented in the appendix. The following are some notable observations and analyses:

- The application of the ensemble mechanism to FedAvg and FedProx yields minor improvements. This is because the server-side model in FedAvg or FedProx is already a relatively strong model, while ensemble mechanisms usually excel with weak models.

- The introduction of CAM significantly enhances the performance of IFCA, which typically struggles with clustering collapse in cluster-wise non-IID scenarios. Notably, CAM decomposes the shared components into a global model and personalized parts into cluster models. Thus, the clustering collapse is mitigated by isolating the dominant shared knowledge.

- FeSEM generally exhibits robust performance on cluster-wise non-IID without outliers. Implementing CAM in FeSEM further improves the Macro-F1 perfor-

mance. The clustering process in FeSEM tends to overfit the label distribution (imbalanced classes) of clients to achieve higher accuracy. However, the application of CAM introduces a global model with a balanced label distribution by averaging all clients, thereby boosting the Macro-F1 performance while preserving the cluster-wise non-IID for high accuracy.

- With an increase in the number of clusters K, the CAM-based methods show substantial improvements in Macro-F1. The decomposition of shared knowledge and cluster-wise non-IID characteristics benefit from a reasonably larger K, which facilitates fine-grained, cluster-wise personalization.

Table 7.1: Test results (mean±std) in **cluster**-wise non-IID settings on Fashion-MNIST & CIFAR-10.

Datasets		Fashion-MNIST				CIFAR-10			
Non-IID setting		Dirichlet $\alpha = (0.1, 10)$		n-class $(3, 2)$		Dirichlet $\alpha = (0.1, 10)$		n-class $(3, 2)$	
#Cluster	Methods	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
1	FedAvg	86.08±0.70	57.24±2.26	86.33±0.44	46.09±1.08	24.38±3.30	11.69±3.15	21.33±3.83	9.0±0.58
	FedProx	86.32±0.78	58.03±3.19	86.42±0.63	45.86±1.42	24.73±3.68	11.28±2.35	22.66±1.13	9.23±0.78
5	FedAvg+	87.61	59.48	86.95	65.61	25.97	12.16	24.35	9.06
	FedProx+	87.94	59.83	86.52	65.73	26.05	12.53	24.83	9.31
	IFCA	84.60±2.22	62.03±3.01	84.94±2.54	66.50±4.43	34.1±4.79	22.12±2.21	29.80±4.49	17.90±2.08
	IFCA-CAM	93.33±0.95	79.64±4.09	95.38±0.49	77.56±1.14	58.13±3.82	28.09±3.68	54.56±3.58	27.27±1.06
	FeSEM	94.64±1.54	82.90±2.38	94.20±1.96	77.07±6.05	59.06±3.24	32.33±7.25	58.76±3.35	35.75±2.54
	FeSEM-CAM	**95.13±1.78**	**85.1±3.17**	**95.69±1.05**	**78.82±1.17**	**64.35±2.33**	**38.33±1.77**	**65.58±1.21**	**38.63±1.17**
10	FedAvg+	89.42	67.83	86.91	63.01	28.45	13.79	27.28	9.81
	FedProx+	89.55	68.02	86.73	63.42	28.33	13.64	26.94	9.64
	IFCA	82.10±5.40	62.62±8.22	86.58±4.97	66.22±5.69	34.84±5.82	22.76±3.99	34.06±2.60	18.7±1.31
	IFCA-CAM	95.42±2.54	88.45±5.46	95.09±0.87	82.98±1.16	70.9±1.18	40.03±1.28	68.46±4.08	41.45±4.0
	FeSEM	95.73±1.28	89.34±1.57	95.54±0.74	84.43±2.38	66.89±2.18	38.35±4.24	71.76±2.23	49.72±3.84
	FeSEM-CAM	**96.19±1.2**	**92.37±1.85**	**98.07±1.46**	**92.43±2.7**	**78.45±1.71**	**49.5±1.13**	**75.04±1.97**	**55.9±2.07**

Client-wise non-IID Table 7.3 and 7.4 presents comparative results under client-wise non-IID scenarios using two benchmark datasets: Fashion-MNIST and CIFAR-10, PathMNIST and TissueMNIST, respectively. Figure 7.1 demonstrates the improvement of both accuracy and macro-f1 score under the $\beta = 0.1$ client-wise non-IID setting using CIFAR-10. Interestingly, IFCA maintains stable performance under client-wise non-IID conditions, primarily because it cannot form a single dominant cluster model - a primary

Table 7.2: Test results (mean±std) in **cluster**-wise non-IID settings on PathMNIST & TissueMNIST.

Datasets		PathMNIST				TissueMNIST			
Non-IID setting		$\alpha = (0.1, 10)$		(3,2)−class		$\alpha = (0.1, 10)$		(3,2)−class	
#Cluster	Methods	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
1	FedAvg	31.38±8.58	14.47±4.27	21.36±5.48	11.49±2.38	49.96±3.39	18.31±4.31	53.46±2.21	15.28±1.36
	FedProx	27.6±6.15	14.07±3.42	25.7±8.48	11.62±1.08	49.78±2.64	17.85±3.81	54.92±3.7	15.15±1.47
5	FedAvg+	35.84	17.01	25.51	12.14	49.52	17.59	54.98	15.12
	FedProx+	27.57	15.74	29.7	13.05	48.88	17.08	52.24	15.54
	IFCA	38.13±2.53	25.22±1.74	34.16±3.76	22.52±1.13	27.44±16.39	16.37±10.45	41.87±20.04	21.59±7.3
	IFCA-CAM	50.12±0.42	25.22±3.67	68.45±5.83	39.31±1.57	**83.08±2.16**	**39.81±3.7**	**83.26±6.47**	36.03±0.96
	FeSEM	59.85±1.45	33.5±4.08	66.37±7.19	41.34±4.12	72.38±1.81	36.79±1.06	70.62±2.41	28.43±2.54
	FeSEM-CAM	**70.01±1.23**	**44.09±4.94**	**71.5±2.2**	**43.69±2.96**	80.28±4.04	34.77±0.49	75.04±4.95	**39.33±3.32**
10	FedAvg+	33.19	19.98	24.82	13.73	49.5	18.03	54.78	13.23
	FedProx+	28.21	16.17	35.62	15.95	46.57	16.47	53.47	14.88
	IFCA	42.34±2.73	29.1±1.52	37.22±4.23	20.2±2.04	38.76±10.94	20.38±2.01	49.31±13.97	21.51±3.68
	IFCA-CAM	66.5±3.46	38.12±2.32	66.22±3.85	40.75±2.2	81.53±7.24	43.88±6.98	88.77±15.23	46.48±4.98
	FeSEM	79.31±0.72	48.14±0.23	71.37±1.5	53.78±2.21	77.12±1.68	47.69±3.1	77.92±1.53	45.68±6.71
	FeSEM-CAM	**85.06±2.62**	**61.82±5.38**	**76.41±2.22**	**64.33±4.92**	**84.91±2.83**	**53.08±1.77**	**90.22±3.03**	**60.62±2.64**

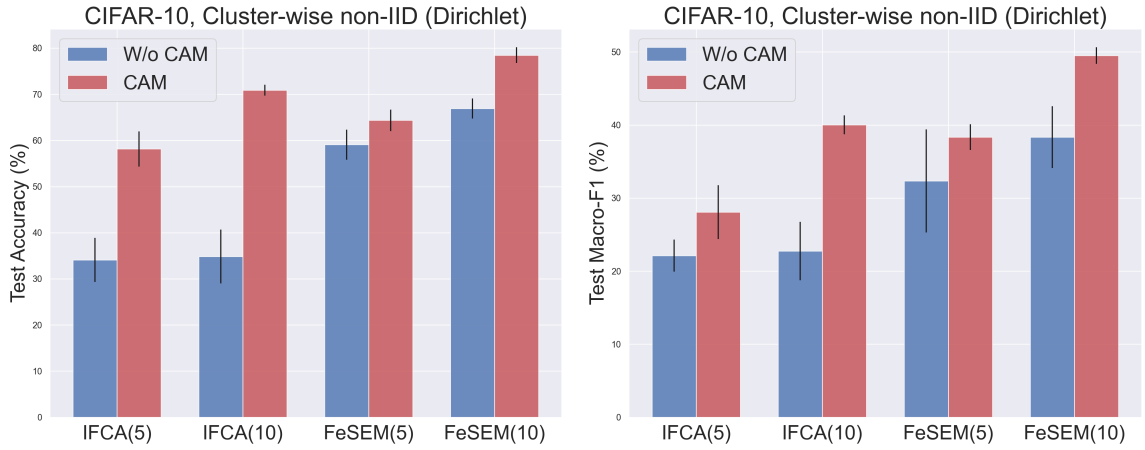

Figure 7.1: Test accuracy and macro-F1 (mean±std) of IFCA/FeSEM (w/o CAM) and IFCA/FeSEM (CAM) in cluster non-IID settings on CIFAR-10 dataset. "IFCA(5)" represents IFCA with $K = 5$ clusters. **CAM consistently brings substantial improvement to IFCA/FeSEM on both metrics and in both settings.**

cause of clustering collapse - in a highly heterogeneous environment. The application of CAM to IFCA and FeSEM shows a significant enhancement, particularly on the CIFAR-10 dataset. This improvement is likely due to FeSEM's typical restriction on knowledge sharing across clusters. In contrast, CAM utilizes a global model to capture more useful common knowledge across clusters, thereby substantially enhancing the generalization capability of each cluster. Furthermore, CIFAR-10, being a relatively complex dataset

with a diversity of images, underscores the importance of sharing common knowledge.

Table 7.3: Test results (mean±std) in **client**-wise non-IID settings on Fashion-MNIST & CIFAR-10.

Datasets		Fashion-MNIST				CIFAR-10			
Non-IID setting		Dirichlet $\alpha = 0.1$		n-class (2)		Dirichlet $\alpha = 0.1$		n-class (2)	
#Cluster	Methods	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
1	FedAvg	85.9±0.46	54.52±2.66	86.17±0.25	44.88±1.24	25.62±3.47	11.38±2.02	24.3±3.53	8.56±0.64
	FedProx	86.03±0.58	54.69±3.32	86.47±0.23	44.89±1.38	25.72±3.29	11.14±1.49	24.19±2.45	8.69±0.74
5	FedAvg+	86.12	61.07	86.5	45.39	25.71	12.45	24.83	8.74
	FedProx+	86.39	56.56	86.15	45.43	25.58	12.43	25.88	8.55
	IFCA	90.13±6.81	68.47±5.23	91.54±5.04	72.3±5.32	47.21± 10.28	22.67±1.48	46.54±12.8	17.78±1.29
	IFCA-CAM	93.72±1.34	70.67±1.75	92.24±1.22	70.24±4.33	54.32±1.25	23.48±1.18	54.92±1.51	25.2±1.05
	FeSEM	91.51±2.9	73.78±9.88	91.83±1.24	71.05±8.63	54.3±4.58	24.78±6.01	55.55±4.83	32.8±4.18
	FeSEM-CAM	**94.74±1.04**	**75.12±5.82**	**93.14±2.03**	**76.98±2.17**	**59.71±2.8**	**40.45±3.53**	**56.7±1.68**	**34.52±1.64**
10	FedAvg+	86.81	60.43	86.91	47.12	27.83	13.65	27.71	9.65
	FedProx+	86.24	56.2	86.78	42.83	25.86	12.84	26.16	9.94
	IFCA	91.04±4.33	68.6±6.77	91.42±5.16	72.29±5.8	47.62±10.15	23.36±2.48	47.96±10.59	17.88±1.04
	IFCA-CAM	**95.7±1.19**	79.17±1.91	92.57±2.63	76.31±4.39	72.54±2.7	42.86±4.36	61.01±2.41	31.63±2.17
	FeSEM	93.3±2.0	80.47±11.05	93.75±1.53	79.39±6.57	67±1.57	31.69±8.52	63.64±6.51	42.97±6.08
	FeSEM-CAM	95.25±1.93	**81.5±2.24**	**95.15±1.48**	**86.16±3.19**	**80.11±1.82**	**59.19±4.67**	**69.88±1.7**	**49.5±1.42**

Table 7.4: Test results (mean±std) in **client**-wise non-IID settings on PathMNIST & TissueMNIST.

Datasets		PathMNIST				TissueMNIST			
Non-IID setting		$\alpha = 0.1$		2−class		$\alpha = 0.1$		2−class	
#Cluster	Methods	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
1	FedAvg	26.41±9.15	14.29±3.08	26.11±8.51	13.05±2.33	52.42±4.04	16.23±3.81	54.11±2.28	14.51±1.28
	FedProx	27.61±7.38	13.97±2.6	28.77±8.33	12.16±2.27	53.42±4.29	15.84±3.41	54.51±3.26	14.43±1.36
5	FedAvg+	32.68	15.03	29.8	13.02	53.15	16.51	54.63	14.57
	FedProx+	33.19	15.66	30.51	13.49	53.56	17.89	55.03	14.78
	IFCA	38.13±2.53	25.22±1.74	34.16±3.76	22.52±1.13	38.76±10.94	20.38±2.01	49.31±13.97	21.51±3.68
	IFCA-CAM	64.67±6.17	34.86±3.95	62.72±3.54	37.5±4.67	84.72±0.95	41.86±1.38	**73.71±0.97**	**33.02±2.64**
	FeSEM	59.85±1.45	33.5±4.08	64.46±6.12	38.41±3.19	72.88±1.11	33.19±1.7	70.62±2.41	28.43±2.54
	FeSEM-CAM	**68.81±1.29**	**49.22±2.2**	**68.92±1.13**	**47.32±1.99**	**87.88±1.27**	**45.82±1.54**	70.09±0.86	29.49±0.77
10	FedAvg+	29.83	16.75	28.35	13.49	53.5	18.03	54.58	13.46
	FedProx+	29.36	16.55	29.07	13.63	54.69	17.36	56.03	15.21
	IFCA	51.88±13.67	27.81±2.21	37.22±4.23	20.2±2.04	27.44±16.39	16.37±10.45	41.87±20.04	21.59±7.3
	IFCA-CAM	77.32±1.0	54.89±3.42	67.91±3.21	40.49±3.58	**88.24±1.62**	54.12±4.15	74.5±0.89	32.04±1.17
	FeSEM	78.93±4.27	52.94±5.42	70.93±4.27	52.94±5.42	78.85±2.29	52.32±7.59	77.92±1.53	45.68±6.71
	FeSEM-CAM	**82.38±2.6**	**63.84±2.03**	**72.95±0.36**	**54.44±1.05**	87.09±1.97	**54.77±2.2**	**80.13±1.6**	**51.9±2.15**

7.4.3 Visualization: CAM combats clustering collapse

Figures 7.3 and 7.4 demonstrate the effectiveness of applying CAM to mitigate clustering collapse in IFCA and FeSEM under both cluster-wise and client-wise non-IID scenarios,

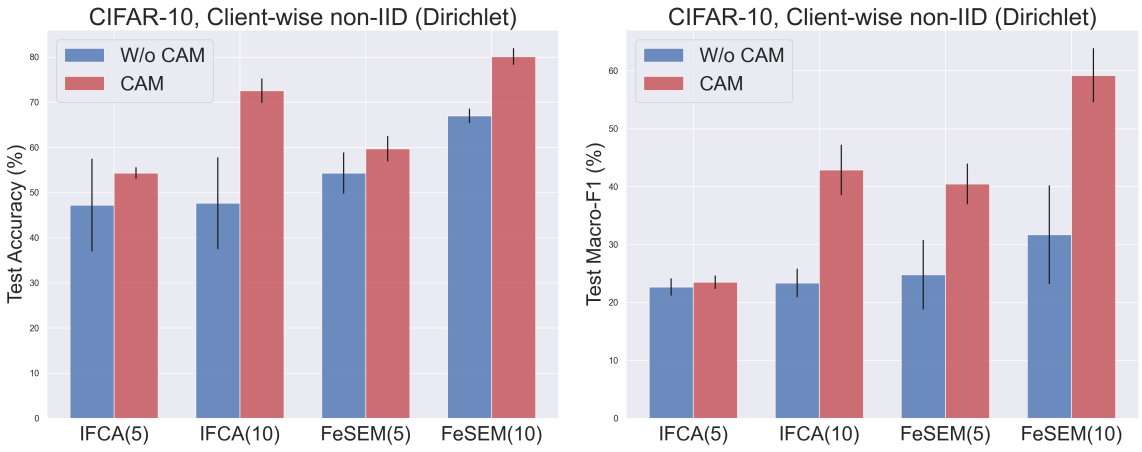

Figure 7.2: Test accuracy and macro-F1 (mean±std) of IFCA/FeSEM (w/o CAM) and IFCA/FeSEM (CAM) in client-wise non-IID settings on CIFAR-10 dataset. "IFCA(5)" represents IFCA with $K = 5$ clusters. **CAM consistently brings substantial improvement to IFCA/FeSEM on both metrics and in both settings.**

using the CIFAR-10 dataset with $K = 10$. Each color represents a cluster, and the X-axis represents the iteration rounds.

In the case of IFCA, we observe a severe clustering collapse issue in cluster-wise non-IID scenarios. A single cluster can encompass all clients in the client-wise non-IID setting and up to 50% of clients in the cluster non-IID setting. Furthermore, the clustering remains unstable throughout the process. However, when CAM is applied in IFCA-CAM, it quickly identifies some clustering structures within a few rounds, and this structure closely approximates the ground truth.

As for FeSEM, while the phenomenon of clustering collapse is not as pronounced, a single cluster can still dominate up to 25% of all clients if there are no outliers. CAM can expedite the clustering convergence, sometimes achieving it in just one round. Moreover, under client-wise non-IID settings, the application of CAM results in lower variance and more uniform cluster size. In the case of cluster-wise non-IID settings, FeSEM-CAM can easily identify the ground truth.

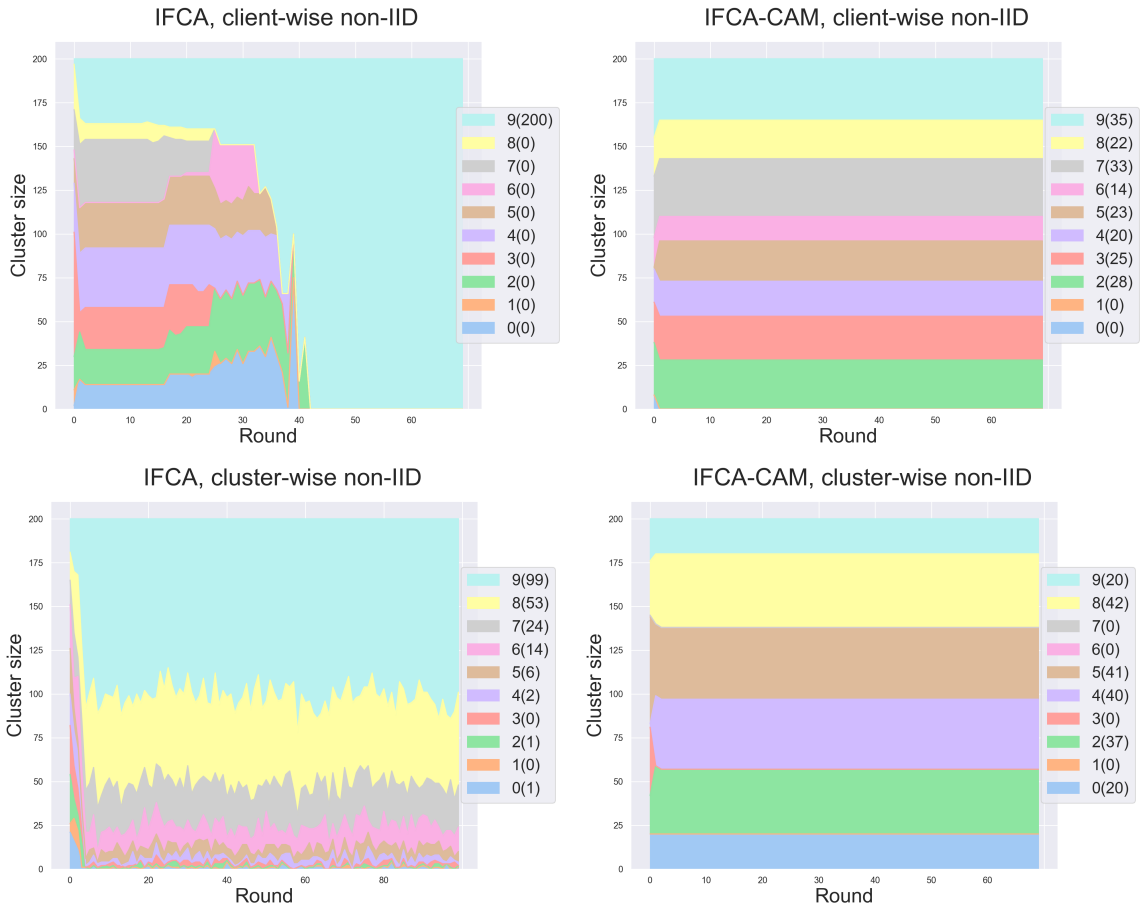

Figure 7.3: Cluster sizes during IFCA vs. IFCA+CAM in client/cluster-wise non-IID settings on CIFAR-10. Legend: cluster ID (cluster size) in the last round. **CAM effectively mitigates clustering collapse/imbalance.**

7.4.4 Comparison with Ensemble Mehtods

In Table 7.5, we further analyze CAM under various scenarios. The terms "-Finetune" and "+" denote finetuning base methods for one additional round and ensembling both methods via soft voting, respectively. We present a few examples as follows.

- **FedAvg+IFCA:** Initially, we separately train FedAvg and IFCA on the same partitioned dataset for 100 rounds, keeping all other hyperparameters identical. We then ensemble the trained models of FedAvg and IFCA to test on the relevant clients using soft voting. The inference is carried out using the formula below,

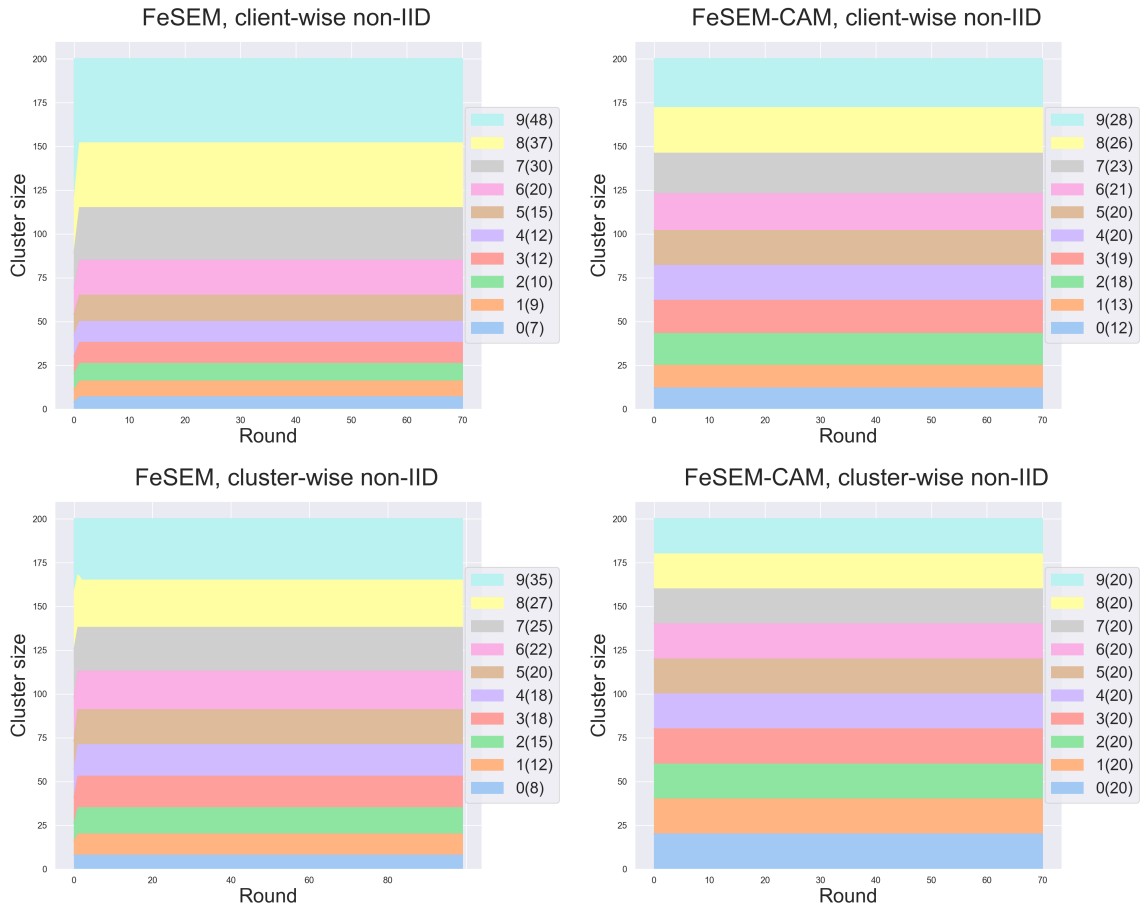

Figure 7.4: Cluster sizes during FeSEM vs. FeSEM+CAM in client/cluster-wise non-IID settings on CIFAR-10. Legend: cluster ID (cluster size) in the last round. **CAM effectively mitigates clustering collapse/imbalance**.

which aligns with the inference method in Fed-CAM,

$$\text{argmax}\, y = f(x;\Theta_g) + f(x;\Theta_{c(i)}). \tag{7.36}$$

- **FedAvg-Finetune+IFCA-Finetune:** Similar to the previous method, we train FedAvg and IFCA separately on the same partitioned dataset for 100 rounds, and then finetune each locally for one additional round. Next, we ensemble the trained models of FedAvg-Finetune and IFCA-Finetune to test the relevant clients using soft voting.

- **IFCA-CAM-Finetune:** After obtaining Θ_g and $\Theta_{c(i)}$, we finetune both locally for

Table 7.5: More comparison, CIFAR-10 cluster-wise non-IID (Dirichlet), $K = 10$

Methods	Accuracy	Macro-F1
FedAvg	24.38±3.30	11.69±3.15
IFCA	34.84±5.82	22.76±3.99
FedAvg+IFCA	35.62±4.73	24.31±3.65
IFCA-CAM	70.9±1.18	40.03±1.28
FedAvg-Finetune+IFCA-Finetune	65.89± 2.31	39.51±1.94
IFCA-CAM-Finetune	**78.97± 1.64**	**52.3± 2.42**
FeSEM	66.89±2.18	38.35±4.24
FedAvg+FeSEM	67.37±1.85	42.03 ±2.45
FeSEM-CAM	78.45±1.71	49.5±1.13
FedAvg-Finetune+FeSEM-Finetune	77.63±1.84	50.34±2.58
FeSEM-CAM-Finetune	**81.33± 1.51**	**57.64± 2.17**

one round without aggregation. Then, we use the finetuned models for testing, applying the same inference method as before.

According to Table 7.5, it is evident that Fed-CAM is not merely an ensemble of CAM models with base models, even though they share the same loss function. Fed-CAM significantly outperforms ensemble methods, irrespective of whether the base model is involved or whether finetuning has been applied. This underscores Fed-CAM's superior ability to address non-IID FL challenges with a clustering structure, and demonstrates the advantages of its clustered FL model over conventional ensemble methods.

7.4.5 Ablation Study of Warmup and Cost

Impact of Warmup Rounds As shown in Table 7.6 below, we gradually increase the rounds of the warmup stage (from 0 to 50) while keeping the total budget of rounds to 100 (warmup + training), considering the limited capacity of computation and communication for local devices in FL. The best performance is achieved when the warmup rounds are set to 20. However, the performance shows minimal variation when the number is set to 10, 20, 30, or 40. It demonstrates that the performance is stable when we choose warmup

rounds in the area from 10 to 40. The choice of warmup round numbers exhibits low sensitivity, like on the parameter plateau.

Notably, with no warmup rounds, performance is substantially decreased due to the impact of worse-performed initial candidates of the FL system. Similarly, when the warmup rounds are increased to 50, indicating insufficient training, the performance will drop accordingly. We need to ensure there are enough training rounds with a proper number of warmup rounds.

In summary, a few warmup rounds can improve the stability of FL optimization and accuracy-related performance. Given a proper area, choosing warmup rounds is low sensitivity to performance.

Extra Cost of integrating proposed CAM framework to existing FL methods For simplicity, we use "FedAvg" as the measuring unit or benchmark for the cost of storage, communication and computation on local devices. In general, CAM will bring one extra "FedAvg" cost to the existing FL methods every communication round.

As for IFCA [30], which needs to transmit K cluster-specific models to each client to compute the clustering, applying our proposed CAM framework with IFCA, we need to transmit K cluster models and one extra global model to the clients, that is $K+1$ models in total. The communication cost and storage cost are listed in Table 1. Moreover, the warmup stage only incurs one "FedAvg" cost. Therefore, integrating CAM can even reduce the overall cost by increasing the number of warmup rounds.

Lastly, considering the tradeoff between performance and cost, we choose 30 warmup rounds out of 100 as the default experiment setting.

7.4.6 More Clustering Analysis

Clustering stability Figure 7.5 demonstrates that the clustering results remain stable after five communication rounds.

Table 7.6: Ablation study of warmup round numbers for performance and cost using "FedAvg" as the measuring unit (Other settings: CIFAR-10 dataset, IFCA [30], client-wise non-IID with Dirichlet distribution $\alpha = 0.1$, Cluster number $K = 10$).

Baseline	# Warmup + Training	Performance/%		Cost/"FedAvg"		
		Accuracy	Macro-F1	Storage	Communication	Computation
IFCA	0+100	47.62	23.36	**10x**	10x	10x
IFCA-CAM	0+100	63.75	32.17	11x	11x	11x
	10+90	72.69	41.24	11x	10x	10x
	20+80	**73.83**	**44.72**	11x	9x	9x
	30+70	72.54	42.86	11x	8x	8x
	40+60	72.98	42.20	11x	7x	7x
	50+50	65.74	26.63	11x	**6x**	**6x**

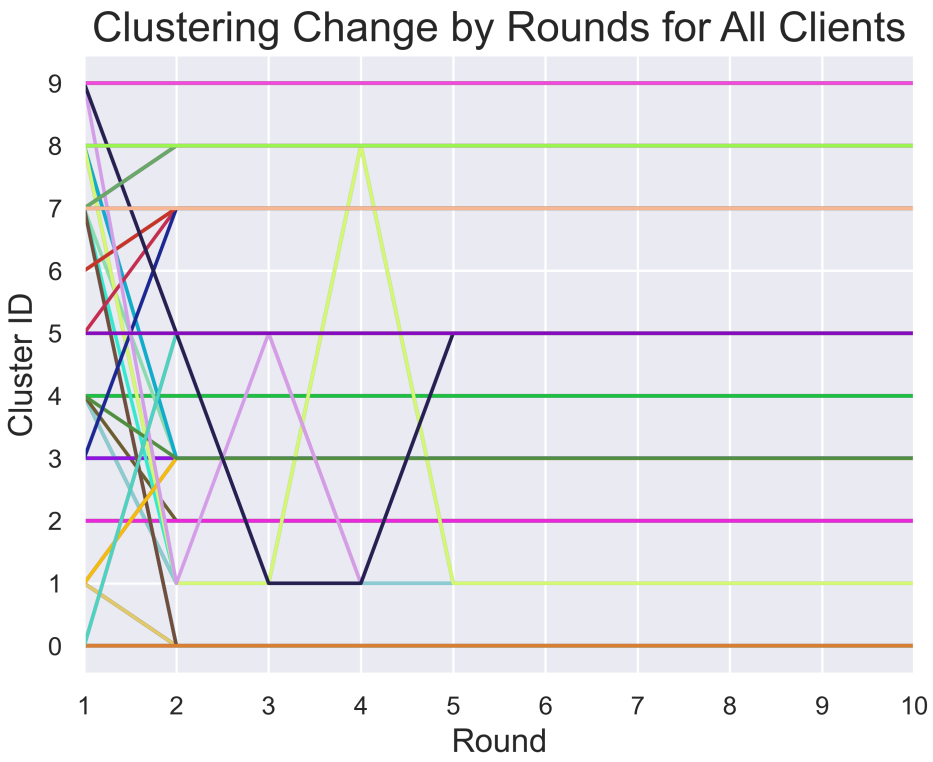

Figure 7.5: A Clustering change example for IFCA-CAM with client-wise non-IID and $K = 10$ on CIFAR-10. Note that there are 200 lines in this graph, and each represents a client. The bold line in this figure is the combination of lines of clients within one cluster. **After five rounds, the clustering remains stable.**

Clustering accuracy in highly-skewed cluster-wise non-IID setting Figure 7.6 is an example of highly-skewed cluster-wise non-IID setting with cluster size

$\{10,10,10,10,10,20,30,30,70\}$. Then Figure 7.7 shows the difference between clustering results when stable and ground truth. Compared with clustering collapse in IFCA, in which all clients fall into one cluster, IFCA-CAM can reveal most of the clustering ground truth.

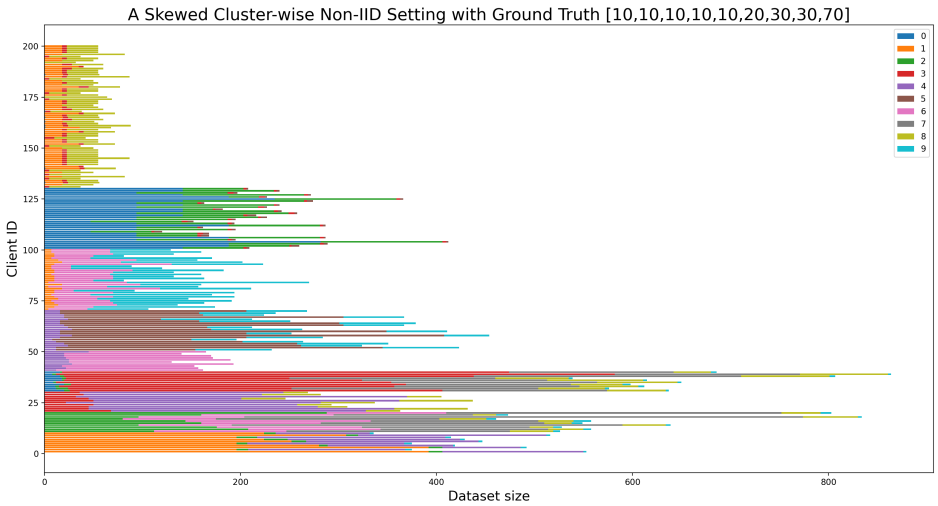

Figure 7.6: A skewed non-IID setting example on CIFAR-10. Legends represent labels of the dataset.

7.5 Conclusions

We propose a novel clustered FL model "clustered additive modeling (CAM)" and an efficient FL algorithmic framework Fed-CAM to address non-IID FL challenges with clustering structure. CAM is a general mode-agnostic tool that can improve various existing non-IID FL methods. It can capture more general non-IID structures with global knowledge sharing among clients than clustered FL and overcome several weaknesses such as clustering collapse, vulnerability to cluster imbalance/initialization, etc. Theoretically, Fed-CAM is capable of achieving an asymptotic convergence rate of $O(1/\sqrt{T})$. Extensive experiments show that CAM brings substantial improvement to existing clustered FL methods, improves cluster balance, and effectively mitigates clustering collapse.

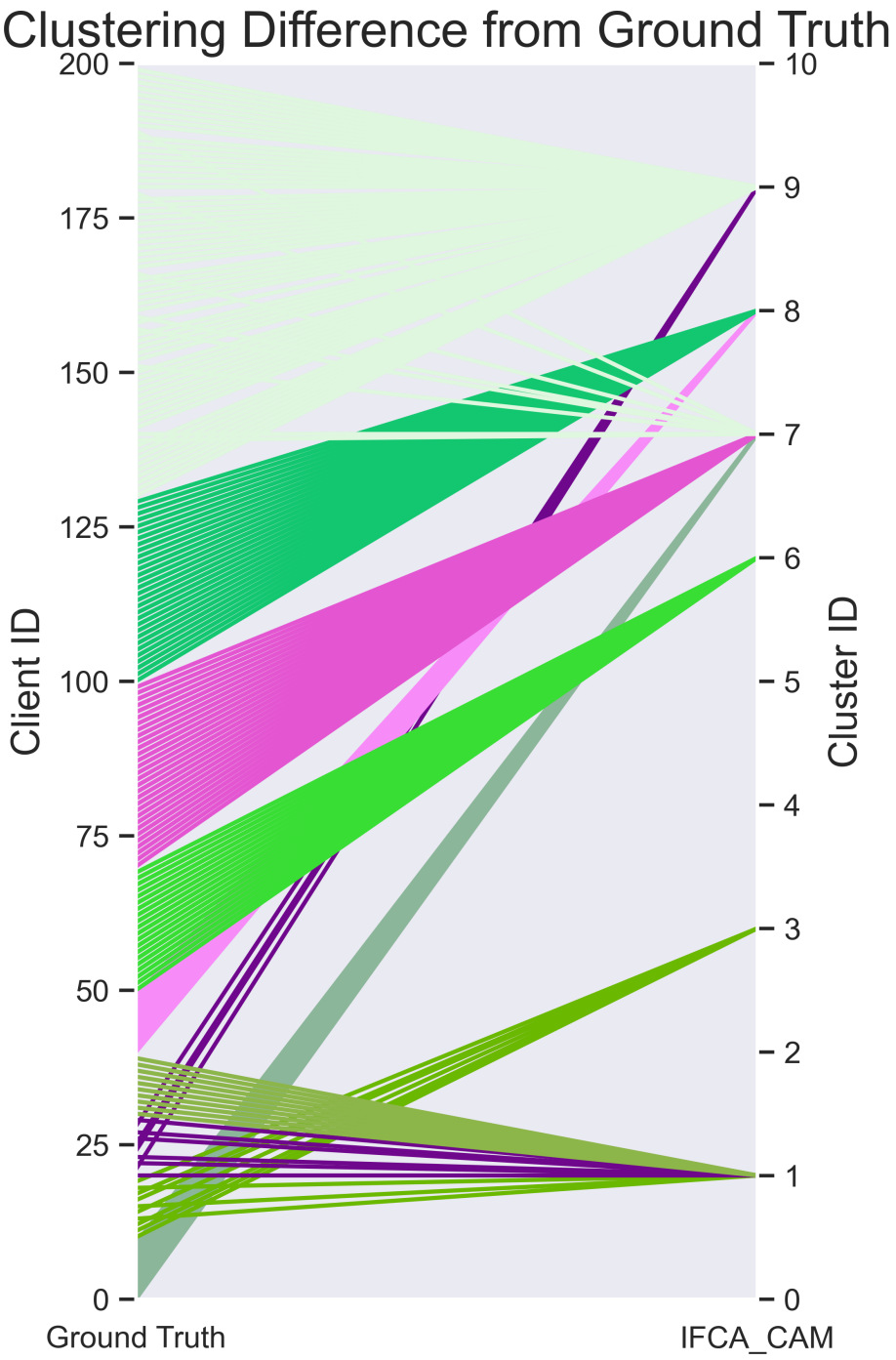

Figure 7.7: In the context of the highly-skewed clustering scenario depicted in Figure 7.6, the differences between IFCA-CAM's clustering and the actual ground truth remain minimal. Conversely, the clustering of IFCA easily collapses into a single cluster. The right y-axis indicates the cluster id. The color represents the ground truth, while the lines indicate the transition from the original ground truth to the clustering through CAM. Notably, **CAM also demonstrates its capability to alleviate clustering collapse and imbalance in skewed clustering settings successfully.**

CHAPTER 8

CONCLUSION AND FUTURE WORKS

8.1 Conclusion

In this thesis, we focus on clustered FL, which applies clustering of clients with simliar data or behaviour into the traditional FL framework to primarily address the non-IID challenge. Four works have been achieved, with summaries as follows.

First of all, in Chapter 5, we rethink clustered FL from a new perspective on its clustering and propose a general framework for clustered FL with a bilevel optimization objective. We then apply weighted clustering to clustered FL. The most important contribution is the proposal of a new convergence analysis for the general form of clustered FL. Experiments on both cluster-wise and client-wise non-IID settings support our claims.

Secondly, we tackle the problem of robust clustering in Clustered FL. In line with the core principles of clustering, which aim to maximize inter-cluster distances and minimize intra-cluster distances, we propose a contrastive approach. This method can

either be viewed as a regularization term added to the loss function of Clustered FL methods or as an integral part of the unified framework for Clustered FL presented in Chapter 3. We introduce two variants of CFL-CON: CFL-CON-rep and CFL-CON-para. Experimental results demonstrate significant marginal performance improvements under both cluster-wise and client-wise non-IID settings.

Thirdly, inspired by the nature of stable clustering in Clustered FL, we propose a clustered knowledge sharing method called CON-CKS. A simplified term, accompanied by a theoretical proof, is provided. This term can be incorporated into any loss function of Clustered FL methods or integrated into the unified framework presented in Chapter 3, while maintaining the linear convergence rate. Substantial performance improvement is demonstrated through extensive experiments, and the effectiveness of the approach is explained from three different perspectives.

Furthermore, we compare CFL-CON and CFL-CKS, which can both be integrated into the unified framework of Clustered FL. Despite sharing the same objective of enhancing CFL, their underlying philosophies are fundamentally different, even opposite, implying that they may not be directly combined to improve the performance of CFL. To address this challenge, we conduct a detailed examination of both methods and the neural network structure. Consequently, we apply CFL-CKS to the backbone of the neural network and CFL-CON to the head. This results in a new hybrid approach, CFL-CON&CKS, which combines the advantages of both methods for clustered FL. Experimental outcomes reveal significant marginal improvements in performance and robustness under both cluster-wise and client-wise non-IID settings.

Finally, to address issues inherent in clustered FL, such as clustering collapse, vulnerability to outliers, and sensitivity to initialization, we introduce a novel clustered FL model, called “Clustered Additive Modeling (CAM)”, along with an efficient algorithmic framework, Fed-CAM. Designed to handle non-IID FL challenges with clustering

structure, CAM is a versatile, model-agnostic tool that can enhance a variety of existing non-IID FL methods. It is capable of capturing more generalized non-IID structures, fostering global knowledge sharing among clients, while overcoming key shortcomings associated with clustered FL. Theoretically, Fed-CAM can achieve a linear convergence rate of $O(1/T)$. Our extensive experimental results demonstrate the substantial enhancements CAM brings to existing clustered FL methods, as it successfully improves cluster balance and effectively mitigates clustering collapse.

In conclusion, we introduced a unified framework for clustered FL, providing an accompanying convergence analysis. Based on this framework, we proposed the Weighted Clustered FL (WeCFL) algorithm and four complementary enhancements - CFL-CON, CFL-CKS, CFL-CON&CKS, and Fed-CAM. These innovations are designed to bolster the robustness and performance of clustered FL. A series of theoretical analysis and comprehensive experiments were conducted to substantiate our propositions and findings.

8.2 Future works

There are still many areas to explore and exploit in the field of clustered FL, as it is a distributed structure with numerous methods that can be applied and researched.

Firstly, the clustering structure in clustered FL can be further improved by investigating soft or hierarchical clustering. These methods can also be associated with the global model and personalized models, providing a more comprehensive understanding of the potential benefits and challenges in this area.

Secondly, more non-IID scenarios can be analyzed to better understand the complexities of different situations. For example, investigating cases where datasets of clients have multiple different domains, or when different labels are assigned to the same data instance due to varying user preferences or multiple choices made by a single user. Addressing the non-IID challenge remains a critical aspect of FL research.

Thirdly, the deployment, scaling, personalization, and continuous updating of clustered FL models need further exploration, especially in real-world cases. Understanding the practical implications and requirements for clustered FL in real applications will help refine and improve these methods.

Lastly, with the emergence of Large Language Models (LLMs), such as ChatGPT, new challenges arise regarding communication and computation costs in FL, as local devices typically have limited capabilities. Combining LLMs with FL, especially clustered FL, requires further research and effort to optimize and balance the trade-offs between model performance and device constraints.

In summary, the clustered FL field offers numerous opportunities for future research, including improvements to clustering structures, the analysis of diverse non-IID scenarios, practical applications, and integration with LLMs. These areas of exploration will continue to advance our understanding and capabilities in clustered FL, leading to more robust and efficient solutions for FL problems.

APPENDIX A

APPENDIX

A.1 Benchmark Datasets

The proposed methods in this thesis are validated using several benchmark datasets as below.

- **Fashion-MNIST** [92] includes 70,000 labeled fashion images (28×28 grayscale) in 10 classes, such as T-shirts, Trouser, and Bag, with others.
- **CIFAR-10** [43] consists of 60,000 images (32×32 color) in 10 classes, including airplane, automobile, bird, and truck, among others. The divergence among classes in CIFAR-10 is relatively higher than in other datasets from the MNIST family.
- **PathMNIST** from **MedMNIST** [99] contains 107,180 images of three channels in nine classes. These 2D biomedical images are collected from colon pathology.
- **TissueMNIST** from **MedMNIST** [99] includes six datasets with 236,386 images of one channel in eight classes. These 3D biomedical images are collected from the kidney cortex microscope.

A.2 Dataset Partition Settings

Each dataset is split among 200 clients, and we create the following non-IID scenarios:

- **Client-wise non-IID via Dirichlet distribution** ($\alpha = 0.1$): This technique uses the Dirichlet distribution to create a level of randomness in non-IID data, following the approach proposed by [35] as demonstrated in Figure A.1. This method is commonly used in most personalized FL techniques, which typically deal with client-wise non-IID scenarios.
- **Cluster-wise non-IID via Dirichlet distribution** ($\alpha = (0.1, 10)$): This strategy partitions the dataset into K clusters using a $\alpha = 0.1$ setting, thus generating significant variation in cluster-wise non-IID scenarios. Subsequently, each cluster is divided into m/K clients using $\alpha = 10$ to manage the non-IID nature of data across clients, as demonstrated in Figure A.2.
- **Client-wise non-IID via n-class (2)**: In this method, each client is randomly assigned n classes from the total classes available in the dataset. Data instances are then sampled from these classes, following the approach proposed by [67], as demonstrated in Figure A.3.
- **Cluster-wise non-IID via n-class (3, 2)**: This technique randomly assigns 3 classes to each cluster, ensuring a reasonably balanced distribution of instances per class, and each cluster is divided into m/K clients. Then, each client within a cluster is assigned 2 classes from the 3 classes, as demonstrated in Figure A.4.

A.3 Details of Model Structure

The detailed structure of CNN models for Fashion-MNIST, CIFAR-10, PathMNIST and TissueMNIST in this thesis.

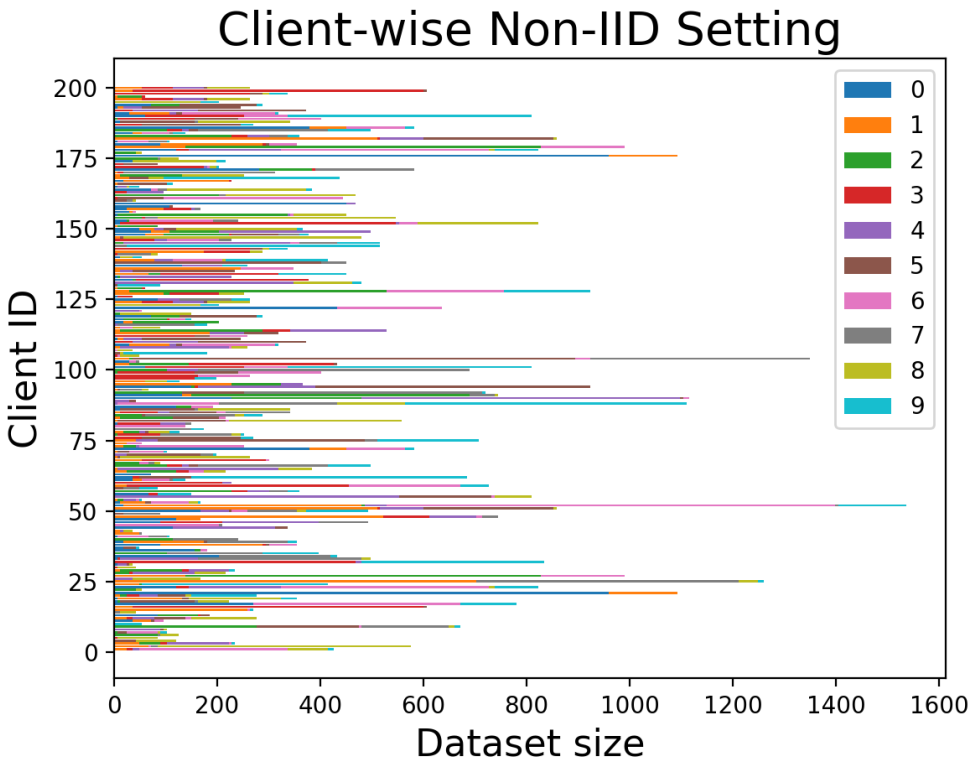

Figure A.1: An example visualization of non-IID partitioning methods of client-wise non-IID by Dirichlet distribution ($\alpha = 0.1$) on the Fashion-MNIST.

Table A.1: Detailed structure of the CNN for Fashion-MNIST.

Layer	Details
Convolution	$Conv2d(1,16,kernel\_size=(5,5),padding=2)$ $BatchNorm2d(16)$ $ReLU()$ $MaxPool2d(2,2)$
Convolution	$Conv2d(16,32,kernel\_size=(5,5),padding=2)$ $BatchNorm2d(16)$ $ReLU()$ $MaxPool2d(2,2)$
Classifier	$Linear(7*7*32,10)$
Loss	$CrossEntropy()$

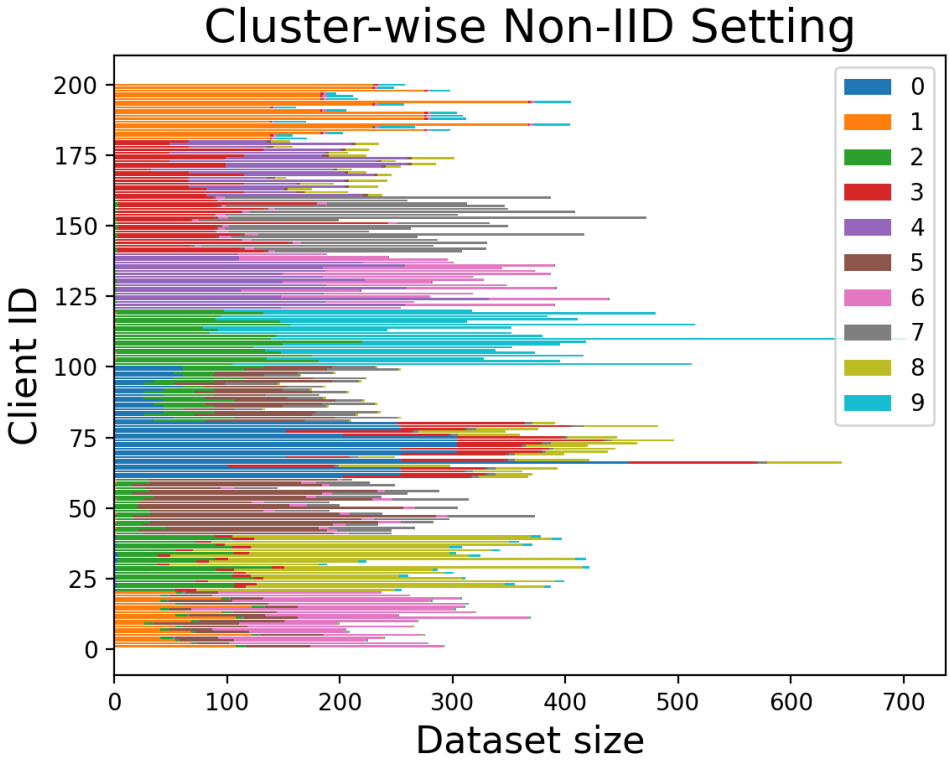

Figure A.2: An example visualization of non-IID partitioning methods of cluster-wise non-IID by Dirichlet distribution ($\alpha = (0.1, 10)$) on the Fashion-MNIST.

Table A.2: Detailed structure of the CNN for CIFAR-10.

Layer	Details
Convolution	$Conv2d(3, 6, kernel\_size = (5, 5))$ $ReLU()$ $MaxPool2d(2, 2)$
Convolution	$Conv2d(6, 16, kernel\_size = (5, 5))$ $ReLU()$ $MaxPool2d(2, 2)$
Linear	$Linear(400, 120)$ $ReLU()$
Linear	$Linear(120, 84)$ $ReLU()$
Classifier	$Linear(84, 10)$
Loss	$CrossEntropy()$

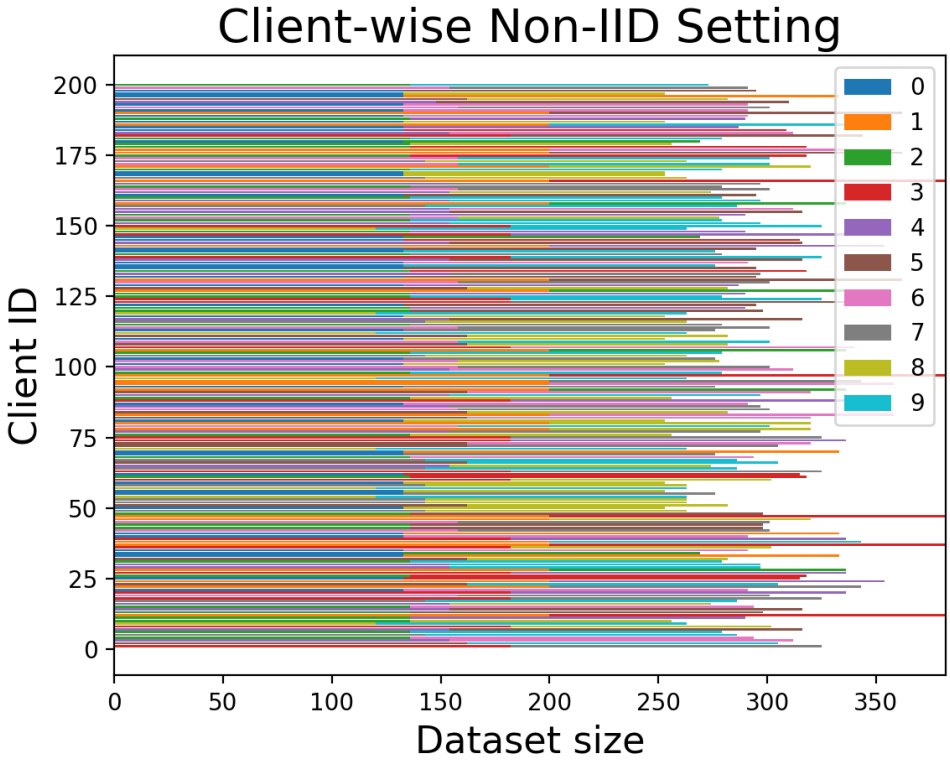

Figure A.3: An example visualization of non-IID partitioning methods of client-wise non-IID by n-class (2) on the Fashion-MNIST.

Table A.3: Detailed structure of the CNN for PathMNIST.

Layer	Details
Convolution	$Conv2d(3,6,kernel\_size=(5,5))$ $ReLU()$ $MaxPool2d(2,2)$
Convolution	$Conv2d(6,16,kernel\_size=(5,5))$ $ReLU()$ $MaxPool2d(2,2)$
Linear	$Linear(400,120)$ $ReLU()$
Linear	$Linear(120,84)$ $ReLU()$
Classifier	$Linear(84,9)$
Loss	$CrossEntropy()$

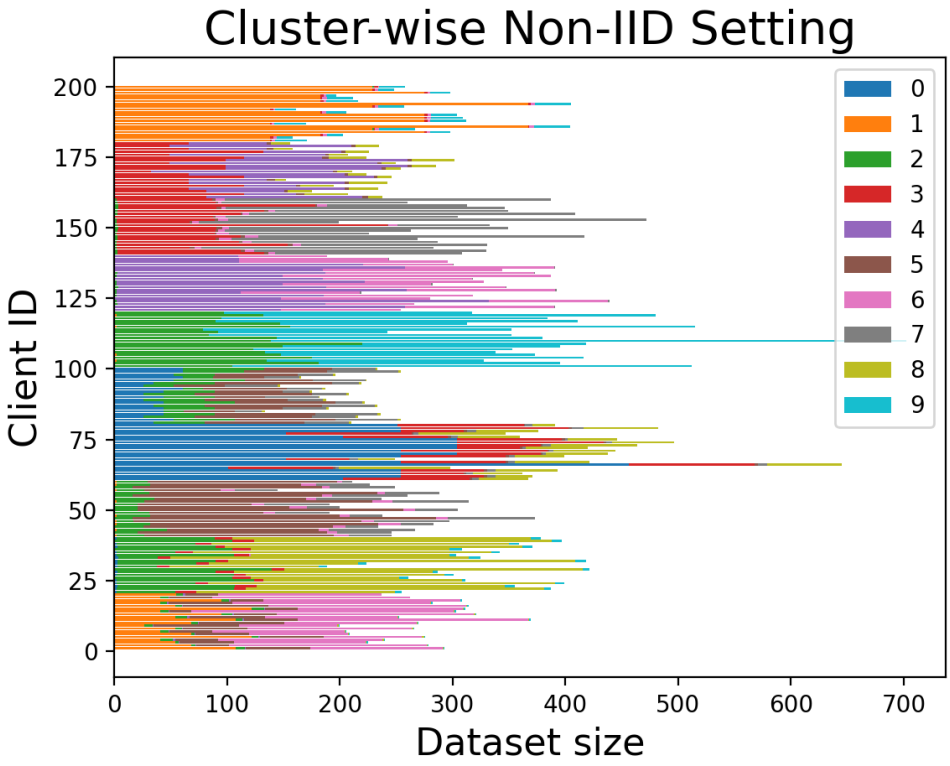

Figure A.4: An example visualization of non-IID partitioning methods of cluster-wise non-IID by n-class (3, 2) on the Fashion-MNIST.

Table A.4: Detailed structure of the CNN for TissueMNIST.

Layer	Details
Convolution	$Conv2d(1, 16, kernel\_size = (5,5))$ $BatchNorm2d(16)$ $ReLU()$ $MaxPool2d(2,2)$
Convolution	$Conv2d(16, 32, kernel\_size = (5,5))$ $BatchNorm2d(32)$ $ReLU()$ $MaxPool2d(2,2)$
Classifier	$Linear(7 * 7 * 32, 8)$
Loss	$CrossEntropy()$

BIBLIOGRAPHY

[1] A. AGARWAL, J. LANGFORD, AND C.-Y. WEI, *Federated residual learning*, arXiv preprint arXiv:2003.12880, (2020).

[2] L. F. ANA AND A. K. JAIN, *Robust data clustering*, in 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings., vol. 2, IEEE, 2003, pp. II–II.

[3] M. G. ARIVAZHAGAN, V. AGGARWAL, A. K. SINGH, AND S. CHOUDHARY, *Federated learning with personalization layers*, arXiv preprint arXiv:1912.00818, (2019).

[4] D. ARTHUR AND S. VASSILVITSKII, *K-means++ the advantages of careful seeding*, in Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, 2007, pp. 1027–1035.

[5] P. AWASTHI AND O. SHEFFET, *Improved spectral-norm bounds for clustering*, in Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, Springer, 2012, pp. 37–49.

[6] M. M. BREUNIG, H.-P. KRIEGEL, R. T. NG, AND J. SANDER, *Lof: identifying density-based local outliers*, in Proceedings of the 2000 ACM SIGMOD international conference on Management of data, 2000, pp. 93–104.

[7] C. BRIGGS, Z. FAN, AND P. ANDRAS, *Federated learning with hierarchical clustering of local updates to improve training on non-iid data*, in 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–9.

[8] C. BRUNET-SAUMARD, E. GENETAY, AND A. SAUMARD, *K-bmom: A robust lloyd-type clustering algorithm based on bootstrap median-of-means*, Computational Statistics & Data Analysis, 167 (2022), p. 107370.

[9] R. CARUANA, *Multitask learning: A knowledge-based source of inductive bias1*, in Proceedings of the Tenth International Conference on Machine Learning, Citeseer, 1993, pp. 41–48.

[10] F. CHEN, G. LONG, Z. WU, T. ZHOU, AND J. JIANG, *Personalized federated learning with graph*, arXiv preprint arXiv:2203.00829, (2022).

[11] F. CHEN, M. LUO, Z. DONG, Z. LI, AND X. HE, *Federated meta-learning with fast convergence and efficient communication*, arXiv preprint arXiv:1802.07876, (2018).

[12] M. CHEN, J. WU, Y. YIN, Z. HUANG, Q. LIU, AND E. CHEN, *Dynamic clustering federated learning for non-iid data*, in Artificial Intelligence: Second CAAI International Conference, CICAI 2022, Beijing, China, August 27–28, 2022, Revised Selected Papers, Part III, Springer, 2022, pp. 119–131.

[13] S. CHEN, G. LONG, T. SHEN, AND J. JIANG, *Prompt federated learning for weather forecasting: Toward foundation models on meteorological data*, arXiv preprint arXiv:2301.09152, (2023).

[14] T. CHEN, S. KORNBLITH, M. NOROUZI, AND G. HINTON, *A simple framework for contrastive learning of visual representations*, in International conference on machine learning, PMLR, 2020, pp. 1597–1607.

[15] G. CHENG, K. CHADHA, AND J. DUCHI, *Fine-tuning is fine in federated learning*, arXiv preprint arXiv:2108.07313, (2021).

[16] L. CHOU, Z. LIU, Z. WANG, AND A. SHRIVASTAVA, *Efficient and less centralized federated learning*, in Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2021, pp. 772–787.

[17] L. COLLINS, H. HASSANI, A. MOKHTARI, AND S. SHAKKOTTAI, *Exploiting shared representations for personalized federated learning*, in International Conference on Machine Learning, PMLR, 2021, pp. 2089–2099.

[18] R. N. DAVÉ AND R. KRISHNAPURAM, *Robust clustering methods: a unified view*, IEEE Transactions on fuzzy systems, 5 (1997), pp. 270–293.

[19] A. P. DEMPSTER, N. M. LAIRD, AND D. B. RUBIN, *Maximum likelihood from incomplete data via the em algorithm*, Journal of the royal statistical society: series B (methodological), 39 (1977), pp. 1–22.

[20] Y. DENG, M. M. KAMANI, AND M. MAHDAVI, *Adaptive personalized federated learning*, arXiv preprint arXiv:2003.13461, (2020).

[21] D. K. DENNIS, T. LI, AND V. SMITH, *Heterogeneity for the win: One-Shot federated clustering*, (2021).

[22] A. DESHPANDE, P. KACHAM, AND R. PRATAP, *Robust k-means++*, in Conference on Uncertainty in Artificial Intelligence, PMLR, 2020, pp. 799–808.

[23] A. DIFFERENTIAL PRIVACY TEAM, *Learning with privacy at scale*, 2017.

[24] L. DUONG, T. COHN, S. BIRD, AND P. COOK, *Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser*, in Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and

the 7th international joint conference on natural language processing (volume 2: short papers), 2015, pp. 845–850.

[25] A. FALLAH, A. MOKHTARI, AND A. OZDAGLAR, *Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach*, Advances in Neural Information Processing Systems, 33 (2020), pp. 3557–3568.

[26] Y. FRABONI, R. VIDAL, L. KAMENI, AND M. LORENZI, *Clustered sampling: Low-variance and improved representativity for clients selection in federated learning*, in International Conference on Machine Learning, PMLR, 2021, pp. 3407–3416.

[27] D. GAO, X. YAO, AND Q. YANG, *A survey on heterogeneous federated learning*, arXiv preprint arXiv:2210.04505, (2022).

[28] L. A. GARCÍA-ESCUDERO, A. GORDALIZA, C. MATRÁN, AND A. MAYO-ISCAR, *A general trimming approach to robust cluster analysis*, The Annals of Statistics, 36 (2008), pp. 1324–1345.

[29] L. A. GARCÍA-ESCUDERO, A. GORDALIZA, C. MATRÁN, AND A. MAYO-ISCAR, *A review of robust clustering methods*, Advances in Data Analysis and Classification, 4 (2010), pp. 89–109.

[30] A. GHOSH, J. CHUNG, D. YIN, AND K. RAMCHANDRAN, *An efficient framework for clustered federated learning*, Advances in Neural Information Processing Systems, 33 (2020), pp. 19586–19597.

[31] S. GUHA, R. RASTOGI, AND K. SHIM, *Rock: A robust clustering algorithm for categorical attributes*, Information systems, 25 (2000), pp. 345–366.

[32] R. HADSELL, S. CHOPRA, AND Y. LECUN, *Dimensionality reduction by learning an invariant mapping*, in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 2, IEEE, 2006, pp. 1735–1742.

[33] C. HE, A. D. SHAH, Z. TANG, D. F. N. SIVASHUNMUGAM, K. BHOGARAJU, M. SHIMPI, L. SHEN, X. CHU, M. SOLTANOLKOTABI, AND S. AVESTIMEHR, *Fedcv: A federated learning framework for diverse computer vision tasks*, arXiv preprint arXiv:2111.11066, (2021).

[34] R. D. HJELM, A. FEDOROV, S. LAVOIE-MARCHILDON, K. GREWAL, P. BACHMAN, A. TRISCHLER, AND Y. BENGIO, *Learning deep representations by mutual information estimation and maximization*, arXiv preprint arXiv:1808.06670, (2018).

[35] T.-M. H. HSU, H. QI, AND M. BROWN, *Measuring the effects of non-identical data distribution for federated visual classification*, arXiv preprint arXiv:1909.06335, (2019).

[36] C. HUANG, J. HUANG, AND X. LIU, *Cross-silo federated learning: Challenges and opportunities*, arXiv preprint arXiv:2206.12949, (2022).

[37] D. JALLEPALLI, N. C. RAVIKUMAR, P. V. BADARINATH, S. UCHIL, AND M. A. SURESH, *Federated learning for object detection in autonomous vehicles*, in 2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService), IEEE, 2021, pp. 107–114.

[38] J. JIANG, S. JI, AND G. LONG, *Decentralized knowledge acquisition for mobile internet applications*, World Wide Web, 23 (2020), pp. 2653–2669.

[39] P. KAIROUZ, H. B. , B. AVENT, A. BELLET, M. BENNIS, A. N. BHAGOJI, K. BONAWITZ, Z. CHARLES, G. CORMODE, R. CUMMINGS, ET AL., *Advances*

and open problems in federated learning, Foundations and Trends® in Machine Learning, 14 (2021), pp. 1–210.

[40] S. P. KARIMIREDDY, S. KALE, M. MOHRI, S. REDDI, S. STICH, AND A. T. SURESH, *Scaffold: Stochastic controlled averaging for federated learning*, in International Conference on Machine Learning, PMLR, 2020, pp. 5132–5143.

[41] A. KHALED, K. MISHCHENKO, AND P. RICHTÁRIK, *Tighter theory for local sgd on identical and heterogeneous data*, in International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 4519–4529.

[42] P. KHOSLA, P. TETERWAK, C. WANG, A. SARNA, Y. TIAN, P. ISOLA, A. MASCHINOT, C. LIU, AND D. KRISHNAN, *Supervised contrastive learning*, Advances in neural information processing systems, 33 (2020), pp. 18661–18673.

[43] A. KRIZHEVSKY, G. HINTON, ET AL., *Learning multiple layers of features from tiny images*, (2009).

[44] Y. LECUN, Y. BENGIO, AND G. HINTON, *Deep learning*, nature, 521 (2015), pp. 436–444.

[45] H. LEE, Y. LIU, D. KIM, AND Y. LI, *Robust convergence in federated learning through label-wise clustering*, arXiv preprint arXiv:2112.14244, (2021).

[46] A. LI, J. SUN, P. LI, Y. PU, H. LI, AND Y. CHEN, *Hermes: an efficient federated learning framework for heterogeneous mobile clients*, in Proceedings of the 27th Annual International Conference on Mobile Computing and Networking, 2021, pp. 420–437.

[47] A. LI, J. SUN, B. WANG, L. DUAN, S. LI, Y. CHEN, AND H. LI, *Lotteryfl: Empower edge intelligence with personalized and communication-efficient federated learn-*

ing, in 2021 IEEE/ACM Symposium on Edge Computing (SEC), IEEE, 2021, pp. 68–79.

[48] C. LI, G. LI, AND P. K. VARSHNEY, *Federated learning with soft clustering*, IEEE Internet of Things Journal, 9 (2021), pp. 7773–7782.

[49] Q. LI, B. HE, AND D. SONG, *Model-contrastive federated learning*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 10713–10722.

[50] T. LI, S. HU, A. BEIRAMI, AND V. SMITH, *Ditto: Fair and robust federated learning through personalization*, in International Conference on Machine Learning, PMLR, 2021, pp. 6357–6368.

[51] T. LI, A. K. SAHU, M. ZAHEER, M. SANJABI, A. TALWALKAR, AND V. SMITH, *Federated optimization in heterogeneous networks*, Proceedings of Machine Learning and Systems, 2 (2020), pp. 429–450.

[52] T. LI, A. K. SAHU, M. ZAHEER, M. SANJABI, A. TALWALKAR, AND V. SMITHY, *Feddane: A federated newton-type method*, in 2019 53rd Asilomar Conference on Signals, Systems, and Computers, IEEE, 2019, pp. 1227–1231.

[53] X. LI, K. HUANG, W. YANG, S. WANG, AND Z. ZHANG, *On the convergence of fedavg on non-iid data*, arXiv preprint arXiv:1907.02189, (2019).

[54] Z. LI, G. LONG, AND T. ZHOU, *Federated recommendation with additive personalization*, arXiv preprint arXiv:2301.09109, (2023).

[55] P. P. LIANG, T. LIU, L. ZIYIN, N. B. ALLEN, R. P. AUERBACH, D. BRENT, R. SALAKHUTDINOV, AND L.-P. MORENCY, *Think locally, act globally: Federated learning with local and global representations*, arXiv preprint arXiv:2001.01523, (2020).

[56] X. LIU, F. ZHANG, Z. HOU, L. MIAN, Z. WANG, J. ZHANG, AND J. TANG, *Self-supervised learning: Generative or contrastive*, IEEE TKDE, (2021).

[57] Y. LIU, Z. LI, S. PAN, C. GONG, C. ZHOU, AND G. KARYPIS, *Anomaly detection on attributed networks via contrastive self-supervised learning*, IEEE TNNLS, 33 (2021), pp. 2378–2392.

[58] Y. LIU, Y. ZHENG, D. ZHANG, H. CHEN, H. PENG, AND S. PAN, *Towards unsupervised deep graph structure learning*, in The Web Conference, 2022.

[59] G. LONG, T. SHEN, Y. TAN, L. GERRARD, A. CLARKE, AND J. JIANG, *Federated learning for privacy-preserving open innovation future on digital health*, in Humanity Driven AI, Springer, 2022, pp. 113–133.

[60] G. LONG, Y. TAN, J. JIANG, AND C. ZHANG, *Federated learning for open banking*, in Federated learning, Springer, 2020, pp. 240–254.

[61] J. LUO, X. WU, Y. LUO, A. HUANG, Y. HUANG, Y. LIU, AND Q. YANG, *Real-world image datasets for federated learning*, arXiv preprint arXiv:1910.11089, (2019).

[62] J. MA, G. LONG, T. ZHOU, J. JIANG, AND C. ZHANG, *On the convergence of clustered federated learning*, arXiv preprint arXiv:2202.06187, (2022).

[63] J. MA, M. XIE, AND G. LONG, *Personalized federated learning with robust clustering against model poisoning*, in Advanced Data Mining and Applications: 18th International Conference, ADMA 2022, Brisbane, QLD, Australia, November 28–30, 2022, Proceedings, Part II, Springer, 2022, pp. 238–252.

[64] J. MA, T. ZHOU, G. LONG, J. JIANG, AND C. ZHANG, *Structured federated learning through clustered additive modeling*, in Thirty-seventh Conference on Neural Information Processing Systems, 2023.

[65] J. MACQUEEN, *Some methods for classification and analysis of multivariate observations*, in Proc. 5th Berkeley Symposium on Math., Stat., and Prob, 1965, p. 281.

[66] Y. MANSOUR, M. MOHRI, J. RO, AND A. T. SURESH, *Three approaches for personalization with applications to federated learning*, arXiv preprint arXiv:2002.10619, (2020).

[67] B. MCMAHAN, E. MOORE, D. RAMAGE, S. HAMPSON, AND B. A. Y ARCAS, *Communication-efficient learning of deep networks from decentralized data*, in Artificial intelligence and statistics, PMLR, 2017, pp. 1273–1282.

[68] X. MU, Y. SHEN, K. CHENG, X. GENG, J. FU, T. ZHANG, AND Z. ZHANG, *Fedproc: Prototypical contrastive federated learning on non-iid data*, Future Generation Computer Systems, 143 (2023), pp. 93–104.

[69] D. PAUL, S. CHAKRABORTY, AND S. DAS, *Robust principal component analysis: A median of means approach*, arXiv preprint arXiv:2102.03403, (2021).

[70] K. PILLUTLA, K. MALIK, A.-R. MOHAMED, M. RABBAT, M. SANJABI, AND L. XIAO, *Federated learning with partial model personalization*, in International Conference on Machine Learning, PMLR, 2022, pp. 17716–17758.

[71] M. RASOULI, T. SUN, AND R. RAJAGOPAL, *Fedgan: Federated generative adversarial networks for distributed data*, arXiv preprint arXiv:2006.07228, (2020).

[72] S. REDDI, Z. CHARLES, M. ZAHEER, Z. GARRETT, K. RUSH, J. KONEČNỲ, S. KUMAR, AND H. B. , *Adaptive federated optimization*, arXiv preprint arXiv:2003.00295, (2020).

[73] N. RIEKE, J. HANCOX, W. LI, F. MILLETARI, H. R. ROTH, S. ALBARQOUNI, S. BAKAS, M. N. GALTIER, B. A. LANDMAN, K. MAIER-HEIN, ET AL., *The*

future of digital health with federated learning, NPJ digital medicine, 3 (2020), pp. 1–7.

[74] F. SATTLER, K.-R. MÜLLER, AND W. SAMEK, *Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints*, IEEE transactions on neural networks and learning systems, 32 (2020), pp. 3710–3722.

[75] F. SCHROFF, D. KALENICHENKO, AND J. PHILBIN, *Facenet: A unified embedding for face recognition and clustering*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 815–823.

[76] M. SCHULTZ AND T. JOACHIMS, *Learning a distance metric from relative comparisons*, Advances in neural information processing systems, 16 (2003).

[77] A. SHAMSIAN, A. NAVON, E. FETAYA, AND G. CHECHIK, *Personalized federated learning using hypernetworks*, in International Conference on Machine Learning, PMLR, 2021, pp. 9489–9502.

[78] G. SHEN, D. GAO, L. YANG, F. ZHOU, D. SONG, W. LOU, AND S. PAN, *Variance-reduced heterogeneous federated learning via stratified client selection*, arXiv preprint arXiv:2201.05762, (2022).

[79] K. SINGHAL, H. SIDAHMED, Z. GARRETT, S. WU, J. RUSH, AND S. PRAKASH, *Federated reconstruction: Partially local federated learning*, Advances in Neural Information Processing Systems, 34 (2021), pp. 11220–11232.

[80] V. SMITH, C.-K. CHIANG, M. SANJABI, AND A. S. TALWALKAR, *Federated multi-task learning*, in Advances in neural information processing systems, 2017, pp. 4424–4434.

[81] S. U. STICH, *Local sgd converges fast and communicates little*, arXiv preprint arXiv:1805.09767, (2018).

[82] C. T DINH, N. TRAN, AND J. NGUYEN, *Personalized federated learning with moreau envelopes*, Advances in Neural Information Processing Systems, 33 (2020), pp. 21394–21405.

[83] A. Z. TAN, H. YU, L. CUI, AND Q. YANG, *Toward personalized federated learning*, IEEE Transactions on Neural Networks and Learning Systems, (2022).

[84] Y. TAN, C. CHEN, W. ZHUANG, X. DONG, L. LYU, AND G. LONG, *Is heterogeneity notorious? taming heterogeneity to handle test-time shift in federated learning*, in Thirty-seventh Conference on Neural Information Processing Systems, 2023.

[85] Y. TAN, G. LONG, L. LIU, T. ZHOU, Q. LU, J. JIANG, AND C. ZHANG, *Fedproto: Federated prototype learning over heterogeneous devices*, arXiv preprint arXiv:2105.00243, (2021).

[86] Y. TAN, G. LONG, J. MA, L. LIU, T. ZHOU, AND J. JIANG, *Federated learning from pre-trained models: A contrastive learning approach*, arXiv preprint arXiv:2209.10083, (2022).

[87] J. TANG, J. LIU, M. ZHANG, AND Q. MEI, *Visualizing large-scale and high-dimensional data*, in Proceedings of the 25th international conference on world wide web, 2016, pp. 287–297.

[88] X. TANG, S. GUO, AND J. GUO, *Personalized federated learning with contextualized generalization*, in Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22, L. D. Raedt, ed., International Joint Conferences on Artificial Intelligence Organization, 7 2022, pp. 2241–2247. Main Track.

[89] J. WANG, Z. CHARLES, Z. XU, G. JOSHI, H. B. , M. AL-SHEDIVAT, G. ANDREW, S. AVESTIMEHR, K. DALY, D. DATA, ET AL., *A field guide to federated optimization*, arXiv preprint arXiv:2107.06917, (2021).

[90] Z. WANG, T. ZHOU, G. LONG, B. HAN, AND J. JIANG, *Fednoil: a simple two-level sampling method for federated learning with noisy labels*, arXiv preprint arXiv:2205.10110, (2022).

[91] S. WU, T. LI, Z. CHARLES, Y. XIAO, Z. LIU, Z. XU, AND V. SMITH, *Motley: Benchmarking heterogeneity and personalization in federated learning*, arXiv preprint arXiv:2206.09262, (2022).

[92] H. XIAO, K. RASUL, AND R. VOLLGRAF, *Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms*, arXiv preprint arXiv:1708.07747, (2017).

[93] E. XIE, J. DING, W. WANG, X. ZHAN, H. XU, P. SUN, Z. LI, AND P. LUO, *Detco: Unsupervised contrastive learning for object detection*, in ICCV, 2021.

[94] M. XIE, G. LONG, T. SHEN, T. ZHOU, X. WANG, J. JIANG, AND C. ZHANG, *Multi-center federated learning*, arXiv preprint arXiv:2108.08647, (2021).

[95] M. XIE, J. MA, G. LONG, AND C. ZHANG, *Robust clustered federated learning with bootstrap median-of-means*, in Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data, Springer, 2022, pp. 237–250.

[96] P. XING, S. LU, L. WU, AND H. YU, *Big-fed: Bilevel optimization enhanced graph-aided federated learning*.

[97] J. XU, B. S. GLICKSBERG, C. SU, P. WALKER, J. BIAN, AND F. WANG, *Federated learning for healthcare informatics*, Journal of Healthcare Informatics Research, 5 (2021), pp. 1–19.

[98] P. YAN AND G. LONG, *Personalization disentanglement for federated learning*, (2023), pp. 318–323.

[99] J. YANG, R. SHI, D. WEI, Z. LIU, L. ZHAO, B. KE, H. PFISTER, AND B. NI, *Medmnist v2: A large-scale lightweight benchmark for 2d and 3d biomedical image classification*, arXiv preprint arXiv:2110.14795, (2021).

[100] M.-S. YANG, C.-Y. LAI, AND C.-Y. LIN, *A robust em clustering algorithm for gaussian mixture models*, Pattern Recognition, 45 (2012), pp. 3950–3961.

[101] M.-S. YANG AND K.-L. WU, *A similarity-based robust clustering method*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 26 (2004), pp. 434–448.

[102] Q. YANG, Y. LIU, T. CHEN, AND Y. TONG, *Federated machine learning: Concept and applications*, ACM Transactions on Intelligent Systems and Technology (TIST), 10 (2019), pp. 1–19.

[103] Y. YANG AND T. M. HOSPEDALES, *Trace norm regularised deep multi-task learning*, arXiv preprint arXiv:1606.04038, (2016).

[104] M. YE, X. ZHANG, P. C. YUEN, AND S.-F. CHANG, *Unsupervised embedding learning via invariant and spreading instance feature*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6210–6219.

[105] C. ZHANG, G. LONG, T. ZHOU, P. YAN, Z. ZHANG, C. ZHANG, AND B. YANG, *Dual personalization on federated recommendation*, arXiv preprint arXiv:2301.08143, (2023).

[106] F. ZHANG, K. KUANG, Z. YOU, T. SHEN, J. XIAO, Y. ZHANG, C. WU, Y. ZHUANG, AND X. LI, *Federated unsupervised representation learning*, arXiv preprint arXiv:2010.08982, (2020).

[107] Y. ZHENG, S. PAN, V. C. LEE, Y. ZHENG, AND P. S. YU, *Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination*, in NeurIPS, 2022.

[108] Z. ZHENG, Y. ZHOU, Y. SUN, Z. WANG, B. LIU, AND K. LI, *Applications of federated learning in smart cities: recent advances, taxonomy, and open challenges*, Connection Science, 34 (2022), pp. 1–28.

[109] H. ZHU, J. XU, S. LIU, AND Y. JIN, *Federated learning on non-iid data: A survey*, arXiv preprint arXiv:2106.06843, (2021).