UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Towards Explainable Personalisation for Federated Learning

by Peng Yan

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Chengqi Zhang and Guodong Long

University of Technology Sydney
Faculty of Engineering and Information Technology

October 2024

the certificate of original authorship

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, *Peng Yan*, 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: 29 October 2024

Abstract

Modern machine learning relies on massive data to train models like deep neural networks, but collecting data is becoming more sensitive with increasing attention to privacy protection. Then, Federated Learning (FL) was proposed to learn a global model while keeping users' data decentralised and private. Personalised Federated Learning (PerFL) improves vanilla FL by balancing collaborative training and model personalisation. It exploits client preferences and turns the global model into personalised models, which usually demonstrate superior performance. However, explainable personalisation is still an open challenge in developing a federated learning system. This research aims to solve the challenges by recognising client preferences embodied in data and proposing on-deployment personalisation where clients can obtain practical and explainable model outputs.

Firstly, the research explains personalisation by disentangling common and personalised knowledge in an FL system with many distributed heterogeneous nodes. A novel Federated Dual Variational Autoencoder (FedDVA) framework is proposed to fulfil the task of disentangling sample representations into client-agnostic (common) and client-specific (personalised) parts. The disentangled representations will demonstrate meaningful structures describing clients' preferences, providing a better interpretation of features contributing to the personalisation.

Further, the research introduces a representation alignment mechanism to learn a universal representation space across clients to measure client properties related

to personalisation. The proposed Client-Decorrelation Federated Learning (FedCD) framework utilises bias in representations of a client's local data to recognise the client's properties. Then, it aligns the global model's hidden space with axes representing client properties, unravelling a client's influence from its sample's latent representations.

Moreover, the research introduces Virtual Concepts (VCs) to explicate clients' preferences and model personalisation. The VCs are a set of vectors describing structures of data partitions of an FL system. They constitute client-supervised information that characterises biases implied in clients' local data. Then, personalisation becomes explicit and explainable by including VCs as labels of clients' preferences in FL's training process.

Qualitative and quantitative experiments on real-world datasets validate the effectiveness and efficiency of our proposed methods. Particularly, data reconstructions based on representations learned by FedDVA demonstrate two irrelevant data manifolds regarding client-agnostic and client-specific knowledge, which validates the effectiveness of personalisation disentanglement. Representations aligned by FedCD and distributions of virtual concepts have consistent cluster structures with data distributed among clients, which could be utilised as a measurement of client preferences explaining personalisation. Furthermore, on-deployment classification performance shows that a global model can learn client preferences with the proposed methods so that it can obtain competitive performance to those delicate PerFL models without needing client-specific modules or extra adaptation processes.

Keywords: Federated Learning, Model Personalisation, Model Interpretability, Variational Auto-Encoder, Concept Vector

Dedication

To my beloved family

Acknowledgments

I appreciate my great parents, my beloved wife, and my extended family for their unwavering love and support. They are my rock and always there, even though we were separated by 8000 km during the tough years of the COVID-19 pandemic. They always trust and encourage me without conditions so that I am able to persevere through challenges and setbacks without falling into frustration or anger.

I acknowledge my supervisors, Prof. Chengqi Zhang and Prof. Guodong Long, for their guidance in both my professional and personal life. Their assistance and encouragement are the most valuable wealth throughout my PhD study, especially facing the unprecedented challenges posed by the COVID-19 pandemic. I am also grateful to work with Dr. Xueping Peng and Dr. Jing Jiang for their generosity and kindness. They have made a friendly and encouraging environment for my academic journey.

I would also like to extend my thanks to my colleagues and friends, whom I acknowledge in no particular order: Dr. Tao Shen, Dr. Ming Xie, Dr. Wensi Tang, Dr. Zhuowei Wang, Dr. Yang Li, Dr. Yue Tan, Dr. Jie Ma, Dr. Zhihong Deng. Studying and working with these intelligent and amiable individuals was my pleasure. We had an unforgettable life in UTS and shared memories of countering the crisis in the pandemic.

Last but not least, I would 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, UTS housing and the Library. Their support and assistance during my study at UTS have been instrumental to my success.

List of Publications

Related to the Thesis :

1. **Peng Yan** and Guodong Long, *Personalization Disentanglement for Federated Learning*, (ICME) *IEEE International Conference on Multimedia and Expo, 318-323, 2023*

2. **Peng Yan**, and Guodong Long, *Client-supervised Federated Learning: Towards One-model-for-all Personalization*, (ICME) *IEEE International Conference on Multimedia and Expo, 1-6,2024*

3. **Peng Yan**, Guodong Long, Jing Jiang and Michael Blumenstein, *Personalized Interpretation on Federated Learning: A Virtual Concepts Approach*, *arXiv preprint*

4. Chunxu Zhang, Guodong Long, Tianyi Zhou, **Peng Yan**, Zili Zhang, Chengqi Zhang, and Bo Yang, *Dual Personalization on Federated Recommendation*, (IJCAI) *International Joint Conference on Artificial Intelligence 2023*

5. Chunxu Zhang, Guodong Long, Tianyi Zhou,**Peng Yan**, Zili Zhang, and Bo Yang, *GPFedRec: Graph-guided Personalization for Federated Recommendation*, (KDD) *ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 4131-4142, 2024*

6. CHUNXU ZHANG, GUODONG LONG, TIANYI ZHOU, ZIJIAN ZHANG, **PENG YAN**, AND BO YANG, *When Federated Recommendation Meets Cold-Start Problem: Separating Item Attributes and User Interactions*. (WWW) *ACM Web Conference, 3632-3642, 2024*

OTHERS :

1. XUEPING PENG, GUODONG LONG, **PENG YAN**, WENSI TANG, AND ALLISON CLARKE, *COVID-19 Impact Analysis on Patients with Complex Health Conditions: A Literature Review*, *Book Chapter, 2023*

Table of Contents

List of Figures

Figure **Page**

List of Tables

Table **Page**

Abbreviation

FL	Federated Learning
PerFL	Personalised Federated Learning
I.I.D.	Independent and identically distributed
Non-I.I.D.	Not independent and identically distributed
VAE	Variational Auto-Encoder
GMM	Gaussian Mixture Model
EM	Expectation Maximisation Algorithm
ELBO	Evidence LOwer Bound
VC	Virtual Concepts
RA	Representation Alignment
CD	Client-Decorrelation
DNN	Deep Neural Networks
CNN	Convolutional Neural Network
BN	Batch Normalisation
FC	Fully-connected Layer
Conv	Convolutional Layer
SGD	Stochastic Gradient Descent Algorithm
KL-Divergence	Kullback-Leibler Divergence
GDPR	The General Data Protection Regulation proposed by the European Union

HITL	Human-In-The-Loop process
ALE	Accumulated Local Effects Plot
SVM	Support Vector Machines
LIME	Local Interpretable Model-agnostic Explanations
IoT	Internet of Things
CRep	Client-specific Representation
URep	Universal Representation
LDA	Linear Discriminate Analysis

NOTATION

General Notations

$\mathbb{C}$	a set of clients
c	a client indexed by c
K	the number of clients in an FL system
$\mathbb{B}$	a batch of data
B	batch size
$\mathbf{G}$	a global model shared among clients
$\mathbf{G}'_c$	model updates from client c
R	number of communication rounds
r	current communication round
$\mathcal{X}_i, \mathcal{Y}_i$	feature distribution and label distribution on the i-th client
$\mathbf{x}^{(i)}, \mathbf{y}^{(i)}$	random variables representing features and labels (targets) on the i-th client; there is $\mathbf{x}^{(i)} \sim \mathcal{X}_i, \mathbf{y}^{(i)} \sim \mathcal{Y}_i$
$P(\mathbf{y}\|\mathbf{x})$	distribution of $\mathbf{y}$ given $\mathbf{x}$
$p(\mathbf{z})$	the probability density function for $\mathbf{z}$
$\hat{y}$	a prediction for y
N_k	the number of samples on the k-th client
$f(\mathbf{x};\omega)$	a model parameterised by ω
$f(\mathbf{x};\omega,\mu_k)$	a model with shared parameters ω and private parameters μ_k
α_k	weight of the k-th client

$\mathscr{L}_k(\omega)$	the supervised loss on the k-th client
$\nabla\mathscr{L}_k(\omega)$	the gradient of $\mathscr{L}_k$ regarding ω
λ	the learning rate
$\|\cdot\|_2$	the L2-Norm
$\mathcal{N}(\mathbf{z};0,I)$	density function of the Gaussian distribution with 0 mean and identity covariance
$\mathbb{E}_{z\sim p(z)}[\mathbf{z}]$	the expectation of $\mathbf{z}$ over $p(\mathbf{z})$
$\mathscr{D}_k$	training set on the k-th client

FedDVA Notations

$\mathbf{z}$	client-agnostic representation
$\mathbf{c}$	client-specific representation
$\mathscr{R}_z$	regularisation term for the client-agnostic representation $\mathbf{z}$
$\mathscr{R}_c$	regularisation term for the client-specific representation $\mathbf{c}$
A,B	weights balancing two regularisation terms
ξ_k	a hyperparameter tuning $\mathscr{R}_c$ on the k-th client
θ	learnable parameters for the shared encoders
φ_k	learnable parameters for the private decoder on the k-th client
D_{KL}	Kullback-Leibler Divergence

FedCD Notations

$P_{d\times r}$	the orthonormal basis for capturing client-specific knowledge
$Q_{d\times t}$	the orthonormal basis for capturing client-agnostic knowledge
d,r,t	dimensions of the matrix
$\bar{c}^{(k)}$	the mean of latent representations of samples on the k-th client
$\bar{c}$	the mean of latent representations of samples in the FL system

$Tr(\cdot)$	the trace of the matrix
GSP	the Gram-Schmidt process
ω_h	learnable parameters of the feature extractor of a deep neural network
ω_g	learnable parameters of the classification head of a deep neural network
Σ_B	the scatter matrix for sample representations across clients
Σ_W	the scatter matrix for sample representations on a client

FedVC Notations

$\mathscr{C} = \{c_1, ..., c_M\}$	M virtual concept vectors
$v_m^{(k)}$	a weight measuring the degree the k-th client's relevance to the m-th concept
$p^{(k)} = \sum_{m=1}^{M} v_m^{(k)} c_m$	the k-th client's preferences
$\hat{z}_i^{(k)}$	the estimated client property by the i-th sample on the k-th client
$\hat{s}_i^{(k)}$	i-th sample's relevance to each virtual concepts
$\hat{p}_i^{(k)}$	the estimated client preference by the i-th sample on the k-th client
ι	a hyperparameter tuning the importance of virtual concepts
κ	a hyperparameter tuning the exponential moving average process

CHAPTER 1

INTRODUCTION

1.1 Background

1.1.1 Federated Learning

Benefiting from massive data, modern machine learning techniques are able to train complex models like deep neural networks (DNNs) [34], which demonstrate promising performance in various tasks, such as image recognition, natural language processing and recommendation. Meanwhile, as these techniques become ever more prevalent, they draw increasing attention to the safety of our private data. Regulations like GDPR [105] are released in many countries to protect against unauthorised data collection and to ensure models are trustworthy.

Federated Learning (FL) [45, 75, 84] was proposed to mitigate privacy risks in traditional machine learning processes, becoming a popular learning paradigm in recent years. The FL embodies the principles of focused collection and data minimisation [75]. It distributes machine learning tasks to a set of clients (i.e., devices like smartphones and

laptops) and collects only model updates to realise decentralised training. Accordingly, data will be kept distributed and closed on each client so that many privacy risks in the centralised learning environment can be averted. For example, users will no longer need to expose their browsing histories when recommendation models are trained decentralised with FL [121, 122].

Generally, the FL training processes consist of collaborative stages between a server and many clients. The clients will first synchronise a globally shared model from the server and optimise it individually on local data, e.g., by gradient-based methods. Then, the server will collect and aggregate local updates to synthesise a new version of the global model. **Algorithm 1** describes the alternate learning process. One iteration between the clients and the server is called a communication round (or a round), which relies on networks, e.g., the internet, to synchronise models among devices. Since network traffic can be costly and unreliable, e.g. for network latency, the communication cost is usually the bottleneck that limits the size of the global model and the efficiency of a learning algorithm.

Algorithm 1 Federated Learning

Input: communication rounds R
Output: global model $\mathbf{G}$

server initialises the global model $\mathbf{G}$
for r from 0 to R **do** ▷ communication rounds
  server selects a set of clients $\mathbb{C}$
  for $c \in \mathbb{C}$ **parallel do**
    client c synchronises $\mathbf{G}$ from the **server** ▷ network traffic
    $\mathbf{G}'_c \leftarrow$**ClientUpdate(G)**
  end for
  server collects local updates $\mathbf{G}'_c$, $c \in \mathbb{C}$ ▷ network traffic
  $\mathbf{G} \leftarrow$**ServerUpdate(**$\mathbf{G}'_c$**)**, $c \in \mathbb{C}$
end for
return $\mathbf{G}$

It is worth noting that federated learning introduces a distinct learning paradigm, which significantly differs from conventional distributed machine learning settings. A

characteristic is that clients in FL are independent devices that manage their local data and training steps. They will only communicate with a server when necessary model updates need to be synchronised. In contrast, in conventional distributed learning, 'clients' are nodes in a data centre. A server will schedule them to utilise computing resources and storage capacity fully. Local data may also be re-partitioned cluster-wise to balance working loads. As a consequence, FL is more favourable to the public's requests for privacy protection. They can keep their data closed on their devices rather than send them to a server or a data centre. They can also reserve the right to opt in or out of a learning process, as devices are independent of the central server.

Further, with the border of FL being greatly expanded, the focus of FL research is developing towards different application scenarios. Two acknowledged communities [45] are the cross-device FL, which emphasises mobile and edge device applications [43, 75, 76], and the cross-silo FL, which involves only a few but reliable clients, e.g., servers from multiple organisations collaborating to train a shared model [22, 23, 65, 109].

The cross-device FL refers to the cooperation among a substantial amount of mobile or IoT devices. It aims to train a globally shared model with private data, such as browsing history, typing preferences and locations. However, clients in this setting are usually unreliable for various reasons, such as unstable internet connection, working period, or opting out of the training, and only a small fraction of clients will be available at each communication round [75]. So far, the cross-device FL has been widely adapted to train models deployed on smartphones. For example, Google has deployed federated learning in Gboard (keyboard on mobiles) [12, 37, 83, 117] and Android Messages [97]. The FL is utilised to train models for tasks like next-word/emoji prediction and query suggestion. Apple also deployed federated learning in iOS [5] to train the QuickType keyboard and the vocal classifier for Siri, a personal AI assistant [4].

The cross-silo FL is built by companies or institutions to facilitate inter-organisation

	Cross-device FL	Cross-silo FL
settings	clients are numerous edge-devices connected by the Internet	clients are multiple data centres connected by private networks
participants	laptops, smartphones and IoT devices	institutes and companies, e.g., banks and hospitals
research interest	privacy protection, communication efficiency, data heterogeneity, robustness	privacy protection, accountability, reliability, model heterogeneity

Table 1.1: Comparisons between cross-device and cross-silo FL

cooperation while keeping the organisations' data confidential. Clients in the setting are usually servers from different organisations and are assumed to be reliable compared with counterparts in the cross-device scenarios. However, data across organisations are changing not only in distributions but also in schema. Coordinating and training models on heterogeneous data become a key to the success of cross-silo FL. The cost of collaboration is another concern for organisations. As infrastructures for sharing and training models could be costly, quantifying the costs and contributions of each participant is a problem to solve before building the FL platform. Several applications have been proposed and are in operation in the industry, such as finance risk prediction for reinsurance [109], digital health records mining [87] and medical data segmentation [23, 61]. **Table 1.1** compares the settings, participants and research interests of the two types of FL.

Peer-to-peer FL is an emerging attempt to reach fully decentralised learning [6, 28, 102, 107]. As the number of clients in FL is enormous, the central server in existing methods can be a bottleneck that limits the performance of a learning process, e.g., due to limited computing power and network capacity. Besides, sharing a global model by a central server gives some clients chances to upload malicious updates to 'poison' the others [80]. Peer-to-peer FL aims to enable clients to collaborate through peer-to-peer communication so that an FL system can completely remove the central server.

Implementation based on the gossip protocol [19, 54] has been proposed, but it remains challenging to orchestrate clients in complex environments to collaborate and co-train efficiently.

1.1.2 Non-I.I.D. Problem and Model Personalisation

A common challenge in FL is the non-I.I.D. problem. Samples on the same client are not independent as they are influenced by their host client's biases, e.g., browsing history on a smartphone will demonstrate biases towards the user's preferences. Samples of different clients are on the opposite. They are usually from various distributions, as users' preferences vary from client to client. From a server's perspective, clients involved in each communication round may also be correlated as they can have similar diurnal or nocturnal patterns in device availability [125].

For vanilla FL frameworks like FedAvg [75], non-independent data on a client will lead local training steps to fit the model towards the implied bias. Then, the aggregation steps on the server will inverse the process. It will eliminate potential bias in individual updates when averaging their parameters. Such contradictory processes hamper the learning algorithm's convergence rate and degenerate the global model's generalisation capability to different distributions.

Personalised FL (PerFL) [24, 74, 112] was proposed to solve the challenge. The PerFL tries to mitigate the non-I.I.D. problem during cross-client collaborations and, in turn, leverages client-specific bias to turn the global model into personalised local models before deploying it. Plenty of works [20, 24, 44, 55, 74, 82, 90, 112, 123] have shown that PerFL models will outperform a single global model when data distribution shifts significantly among clients.

1.1.2.1 Non-I.I.D. Problem

In PerFL, most methods assume samples are observed independently, and the main focus is the non-identical distribution problem among clients. Several types of the distribution shift problem have been discussed, which may relate to classic settings in conventional machine learning [41]. For ease of understanding, let $(\mathbf{x}^{(i)},\mathbf{y}^{(i)})$ be random variables representing features and labels (targets) observed on the i-th client. $\mathscr{X}^{(i)}$ and $\mathscr{Y}^{(i)}$ are their distributions, i.e., $\mathbf{x}^{(i)} \sim \mathscr{X}_i$ and $\mathbf{y}^{(i)} \sim \mathscr{Y}_i$. There are the following types of differences in distribution among clients:

- **Feature Shift**: In this scenario, feature distributions $\mathscr{X}^{(i)} \neq \mathscr{X}^{(j)}$ when $i \neq j$, even if the two clients share the same relationship between $\mathbf{x}$ and $\mathbf{y}$, e.g., $P(\mathbf{y}|\mathbf{x})$. It happens in many tasks, such as object recognition, where the input features may have specific biases due to factors like illumination and device resolution. Similar settings are also introduced in the research topic of Domain Adaptation, but in FL, more restrictions on data access are imposed. For example, a client can only visit its local data/distribution, while comparing data of two distributions is feasible for conventional domain adaptation tasks.

- **Target Shift**: The probability that samples of a specific class are observed may vary across clients. This setting is referred to as the target shift, i.e., $\mathscr{Y}^{(i)} \neq \mathscr{Y}^{(j)}$ if $i \neq j$. It happens when targets (labels) are closely related to some client properties, such as location and the user's gender. For example, images of koalas are more likely to appear on smartphones in Australia. Few-shot learning and transfer-learning techniques sometimes help mitigate the challenge as they are able to rapidly adapt to new tasks, e.g., new classes, when deployed on an unseen client.

- **Concept Shift**: The concept shift [110] problem has been widely discussed in machine learning. It is raised by the changing mapping between the feature $\mathbf{x} \sim \mathscr{X}$

and the target $\mathbf{y} \sim \mathscr{Y}$, i.e., $P(\mathbf{y}|\mathbf{x})$. A typical case is that users will have different attitudes (target) towards the same movie (feature). Then, a recommendation model needs to adapt its predictions according to the user's preferences. In FL, the concept shift problem could be more challenging as the preferences could result from intricate environments that won't reflect in data, such as hardware conditions, seasons, regions, religions, etc [45].

Except for the non-I.I.D. problem, FL is expected to encounter various cross-client data variations in real-world applications. A related topic, Heterogeneous Federated Learning [119], is to study the heterogeneity problem, such as heterogeneous models and multi-modality data. PerFL focuses on the non-I.I.D. problem, i.e. statistical heterogeneity, and data are assumed to have an identical structure and format.

1.1.2.2 Model Personalisation

There are multiple strategies to realise model personalisation in PerFL [112]. Some methods will 'generate' a model for each client [90], while others need only fine-tuning a classification head on the local dataset [20]. This section categorises PerFL methods according to portions of model parameters utilised to capture client-specific knowledge.

- **Full Personalisation**: A straightforward way to personalise is to fine-tune the globally shared model on a client's local data [18, 21, 75]. All parameters of the model will be optimised to capture the client's bias. However, the strategy is sometimes impractical in real-world applications due to limited computation capability and data scale on the client. Some works [30, 55] mitigate the problem by learning initialisation for client-specific models, which are more robust to the changing distributions and can adapt fast to new clients. Hyper-network [90] is a new technique where the global model is a DNN that directly generates personalised parame-

ters for local models. Still, due to the complexity of the global hyper-network, its training process is usually computation and communication costly.

- **Group-wise Personalisation**: Unsupervised methods [33, 74] help obtain fully personalised models without being limited by a client's local resources. They cluster clients into groups regarding client-specific properties and co-train a model within each group to achieve group-wise personalisation. As cross-client collaboration only happens among clients with similar properties, it can take advantage of FL's collaborative training and simultaneously tune the model toward the properties shared in the group. A major difficulty is to cluster clients while keeping data private. Most methods cluster clients by the similarity between local models' parameters, but they usually suffer from problems like cluster collapse, unreliable initialisations, etc [33, 112].

- **Partial Personalisation**: Many PerFL works study partial personalisation by splitting a model into global and local parts. Clients will update the two parts alternately on their local data and only share the global parts as in vanilla FL. Research has shown that partial personalisation can obtain most benefits of full personalisation with a small fraction of client-specific parameters [82]. **Figure 1.1** summarises prevalent architectures for partial personalisation. Compared with other PerFL methods, a noticeable drawback is that the global model is no longer ready to use. It has to be concatenated and aligned with a client-specific module when deployed on a new client, while there may be no prepared data to fulfil the process.

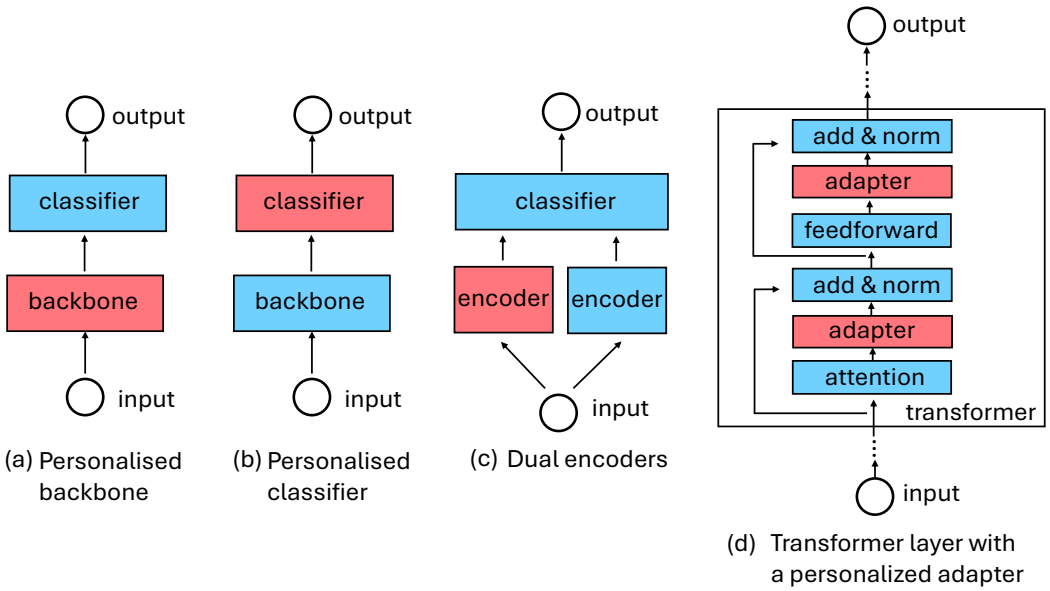


Figure 1.1: Four architectures of partially personalised models in PerFL. Red modules denote personalised parts trained individually on each client. Blue modules denote the global parts shared among clients.

1.2 Interpretable Personalisation

With the prevalence of machine learning, intelligent models are changing our shopping behaviours, influencing our watching preferences, and even deciding whether we can get a loan. Their indispensability and influence necessitate the accountability of the machine learning processes. An urgent request is that a model's behaviour be interpretable to human beings so that people can know why they get a specific result from the model, e.g., what factors caused a rejection of the loan application.

The request to be interpretable is even more crucial to model personalisation in PerFL. As training data are closed and can not be sanitised beforehand, PerFL models are vulnerable to malicious content like bias and discrimination [71, 103]. Attackers can take the chance to poison the FL system by fitting misleading data during local training processes. Besides, personalisation will bring diversity to model outputs for different clients. It may raise concerns about decision fairness, e.g., why do they get disparate

outcomes for identical input?

Interpretable personalisation helps defend PerFL against detrimental updates and improves decision transparency, making personalised models trustworthy. Specifically, interpretable personalisation requires a PerFL model to be ready to answer questions: 1) what properties a client has that will contribute to personalisation? and 2) how do these properties change a model's decision? Then, the black-box PerFL model will become controllable through Human-In-The-Loop (HITL) [111] operations. For example, users can inspect personalised information that may change model outputs; developers can set decision boundaries regarding client preferences to filter out malicious inputs.

1.2.1 Interpretability

So far, there is no mathematical definition of interpretability. An acknowledged definition is that interpretability is the degree to which a human can understand the cause of a decision [77]. Model interpretation usually plays a role in bridging the gap between black-box models and meaningful knowledge humans can understand (**Figure 1.2**[1])

Some research dissects a black-box model to comprehend how the model makes predictions [62], while others explain why a model makes a specific prediction for a sample. [77, 78] suggest distinguishing between the terms interpretability and explanation, where "interpretability" refers to the comprehension of a black-box model, and "explanation" denotes explanations of individual predictions. We summarise some common methods to interpret a model or explain a specific prediction.

1.2.1.1 Interpretable Methods

- **Shallow Models**: Most of the shallow models in machine learning are intrinsically interpretable. For example, linear models like logistic regression and LASSO [38]

[1]* image source: https://en.wikipedia.org/wiki/Universe; # image source: [67]

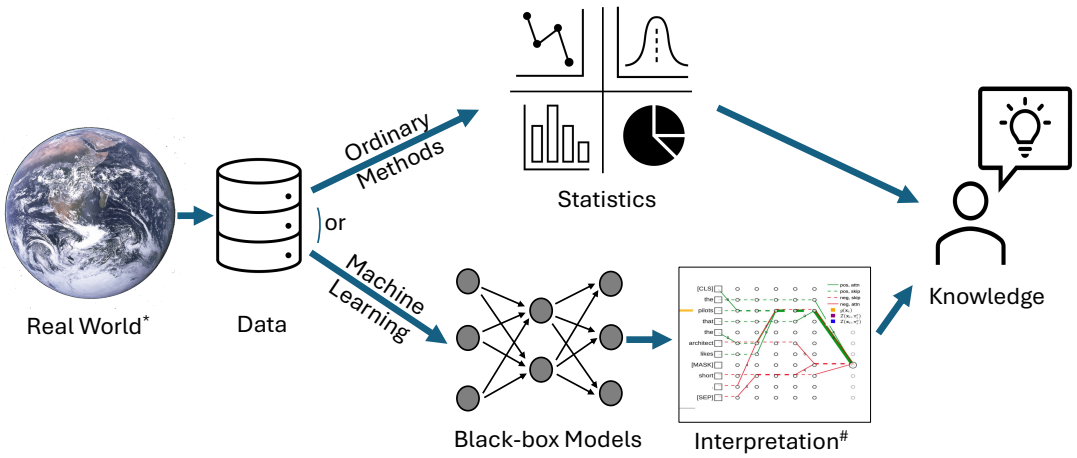


Figure 1.2: A pipeline of knowledge discovery: 1) ordinary methods extract knowledge from data by statistical approaches that are interpretable and accountable; 2) Modern machine learning applies black-box models to learn complex and abstract concepts from data, but they are hard to understand. Model interpretation bridges the gap between black-box models and meaningful knowledge humans can understand.

have explicit decision boundaries that make it easy to tell the importance of each feature and how the model will change its output along with the input changes. Tree-based models also have transparent decision processes by tracing the path from the root to a specific leave. Ordinary shallow models have been vastly applied in our daily lives, especially in handling tabular data. However, they have limited capacity for tasks learning abstract concepts, e.g., object recognition, and the interpretation could be unintuitive when the relationship between features is complex, e.g., due to the multicollinearity problem.

- **Partial Dependence Plot and Accumulated Local Effects Plot**: Partial Dependence Plot (PDP) [124] and Accumulated Local Effects Plot (ALE) [3] are mathematical tools that show the marginal effect that a feature has on a model's output. They plot the expected model output regarding a given feature while marginalising it over other features. Theoretically, they will directly show us the relationship between the target and the feature of interest, but they could be less effective when

the feature space is large. For example, it is infeasible to marginalise a pixel in an image to show its contribution to recognising objects.

- **Prototypes and Criticisms**: Prototypes [36] and criticisms [47] are example-based methods with fair interpretability. The prototypes are representative samples selected through density estimation, and the criticisms are those not well represented. They provide us with a way to reveal critical samples changing a model's behaviour for a sanity check. A difficulty is setting hyperparameters to find the required samples, e.g., the number of prototypes, the kernel measuring sample similarities, etc. Different choices of hyperparameters usually lead to divergent prototypes and criticisms. So does the interpretation.

- **Activation Maximisation**: Early research looks inside latent layers of DNNs by visualising learned feature maps. They visualise the input that maximises the activation of a unit to reveal what builds up a model's knowledge over many layers [46, 81]. An intuitive discovery is that the lower a layer is, the more abstract concepts are learned, and the higher a layer is, the more concrete concepts are learned. For example, from the lowest layer to the highest layer, a DNN may respectively learn concepts of edges, textures, patterns and objects (**Figure 1.3**). The activation maximisation provides a straightforward insight into hidden units in a DNN model. However, visualising does not imply that we can interpret the model's decision process, and the visualised feature maps themselves are usually too hallucinatory to understand, e.g., objects in **Figure 1.3(d)**.

- **Concept Vectors**: Concept Vectors, or Concept Activation Vectors [32], study the impacts of a given concept on a DNN's predictions. Concretely, a concept is any abstraction one cares about, such as a word, an object or a colour. The concept vector is a binary classifier, e.g., SVM or logistic regression, to separate hidden units activated in a specific layer when the concept appears. Then, by comparing

the concept vector and activated units leading to predictions, e.g., by the T-test [48], one can quantify the concept's influence on the model's outputs. As users only need to collect data for the concept of interest in training a binary classifier, concept vectors are friendly to experts who specialise in domain knowledge but know little about the DNN architecture. A limitation is that concept vectors may only work in deeper layers, as many works have shown that layers in shallow DNNs are usually inseparable [2].

- **Influential Functions**: An influence function quantifies the influence of a training sample on the model's parameters and outputs [52]. It up-weights the loss of a sample by an infinitesimally small step and approximates subsequent changes in model parameters to derive an influence score. The larger the changes cause, the higher the sample's influence on model parameters, as well as on its predictions. The influence score can be used as a measure of the similarity between the training and the test samples (regarding the model) to identify influential training samples that lead to a false prediction (debug). The obstacle to the influence function is calculating the Hessian matrix of model parameters, which is computationally costly and usually numerically unstable.

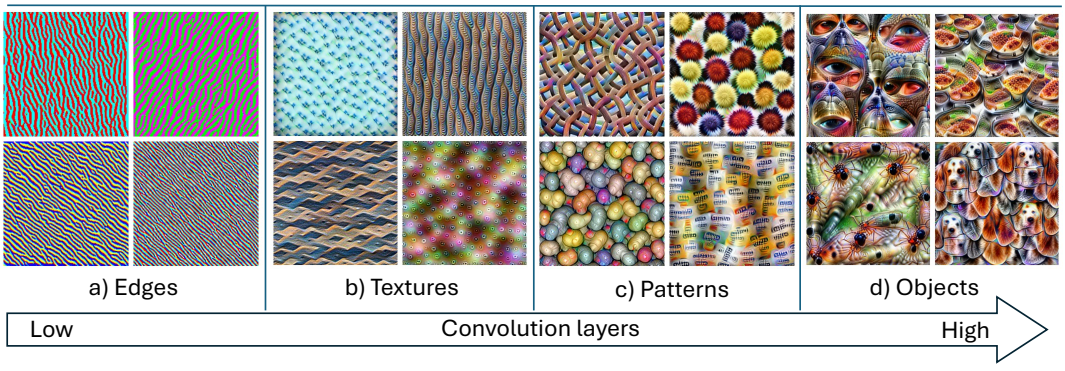


Figure 1.3: Visualisation of hidden layers in a GoogLeNet [46]. As the layer goes deeper (from the left to the right), the concepts learned become more complex and concrete.

1.2.1.2 Explanation Methods

- **Local Interpretable Model-agnostic Explanations**: Local Interpretable Model-agnostic Explanations (LIME) [86] seeks to surrogate the black-box model with a shallow and interpretable model near the region of a certain sample. Then, one can explain predictions near the region by investigating the decision logic of the surrogate model. Anchors [85] have a similar idea of local surrogates but generate explanations of IF-THEN rules that are easy to understand.

- **Saliency Maps**: Saliency Maps highlight the features of an input that are relevant to the model output [91]. A common implementation is to compute the gradient of a score, e.g., classification loss, for the class of interest with respect to the input. The larger the absolute value of the gradient, the stronger the relevance (positively or negatively) of the feature in recognising the class. The saliency maps provide us with intuitive visualisation to explain why a DNN made a specific prediction and have been widely used in medical tasks, e.g., medical image analysis. However, many saliency methods have shown failures in critical sanity checks [104]. For example, experiments show that they do not depend on the learned model but rather depend on the distribution of input features [104].

- **Shapley Values**: Similar to the saliency maps, Shapely Values [69] highlight the features of an input that contribute to a prediction. However, rather than depending on gradients, Shapley values quantify the contribution of the value of each input feature. They are proven to satisfy many critical properties, e.g., symmetry, dummy and additivity, for sanity checks [95]. To compute the Shapley value, the contribution of a specific value of a feature is the difference between the effect of the given value and the effect of a baseline value, usually the expectation of that feature. Then, the Shapley value is the averaged contribution of the given value of the feature over all possible feature value combinations. The most challenging

problem of the Shapley method is that it is computationally costly, e.g., marginalising a value over all other feature value combinations. Besides, selecting a baseline value is usually difficult, while they have shown a strong influence on the resulting Shapley values [95].

1.2.2 Challenges for Interpretable Personalisation

While plenty of works have attempted to interpret black-box models or explain a particular prediction, model personalisation in FL remains to raise people's attention. PerFL's decentralised settings and non-I.I.D distributions even bring more challenges in providing a transparent decision process.

1.2.2.1 Undefined Client Preferences

A typical difficulty is that client preferences regarding personalisation are implied in training data without explicit definitions. A preference could be a client's favour towards specific classes or a specific noise mixed up with input features. However, from the perspective of a PerFL model, all kinds of preferences are treated as shifts in data distributions, as introduced in **Section 1.1.2**. Most personalisation methods implicitly learn those preferences when tuning the model for tasks like classification, precipitating them as personalised knowledge in hidden layers. Unfortunately, methods introduced in **Section 1.2.1** require a well-defined target, e.g., labels, to deduce the cause of a model making a specific decision. They are hard to apply without determined targets.

Still, group-wise personalisation (**Section 1.1.2.2**) provides a glimpse into how to uncover comprehensible preferences contributing to personalisation. It groups clients into clusters according to the similarity between model parameters and investigates properties a client has in common with those clients in the same cluster. Results show that clients in the same cluster will demonstrate the same preference for certain prop-

erties, e.g., they may use the same language [112]. Then, although clients' preferences are not explicitly defined, it is promising to reveal meaningful preferences relevant to personalisation by examining clients sharing similar local models.

1.2.2.2 Entangled Influences

Undefined client preferences will raise another problem of entangled influences. As most PerFL models are trained in an end-to-end schema with classification loss as supervised information, the potential influences of client preferences will be entangled with the supervised information and scattered throughout the model's hidden layers. As a consequence, to identify influences of a client's preferences, one needs first to disentangle client-specific and client-agnostic knowledge, which is difficult without supervised information about client preferences.

Partial model personalisation frameworks (**Section 1.1.2.2**) may mitigate this problem. Here, part of the model parameters will be shared as in vanilla FL, and the rest will be held locally for personalisation. Then, the client-specific knowledge and client-agnostic knowledge are separated into the local and global modules so that one can investigate a client's influences by inspecting changes introduced by the local module.

On the other hand, partially personalised parameters deprive PerFL models of consistency with client properties. Clients may learn very different local parameters even though they have similar preferences, so uncovering comprehensible properties by investigating clients sharing similar local models is no longer feasible through this personalisation strategy.

1.2.2.3 Inconsistent Representation Space

The distributed learning environment of PerFL will be challenging for making consistent interpretations across clients. As personalised knowledge of each client usually leads to diverse local models, an interpretation of a local model may be improper to the

model of another client. The critical solution is to ensure that distributed local models share a consensus about properties related to personalisation, e.g., to share a unified representation space describing client properties. Subsequently, clients can derive a universal interpretation of personalisation by studying changes caused by properties they are all aware of, even though local models are distributed.

It is worth noting that sharing unified representations of client properties will also make one-model-for-all personalisation possible. Concretely, according to the unified representations, a global model can directly learn the relationship between client properties and target concepts rather than by fine-tuning a client's local data. Then, the global model is ready to deliver personalised outputs to all scenarios requiring model personalisation without needing extra tuning steps. The property is indispensable to clients where local updates are hard to afford, e.g., IoT devices.

1.2.3 Aims and Significance

This research aims to solve the above challenges by recognising client preferences embodied in sample representations and proposing interpretable personalisation where clients can obtain practical and interpretable personalised models. The relationship between research objectives and the challenges they solve is described in **Figure 1.4**.

1.2.3.1 Personalisation Disentanglement

As there is no supervised information describing client preferences relevant to personalisation, the research will first investigate biases implied in a client's local data and disentangle them from the universal, or client-agnostic, knowledge as client-specific knowledge to mitigate the problem of entangled influences.

Specifically, disentanglement denotes finding a representation where a change in one factor of the representation corresponds to a change in the cause of variation of a

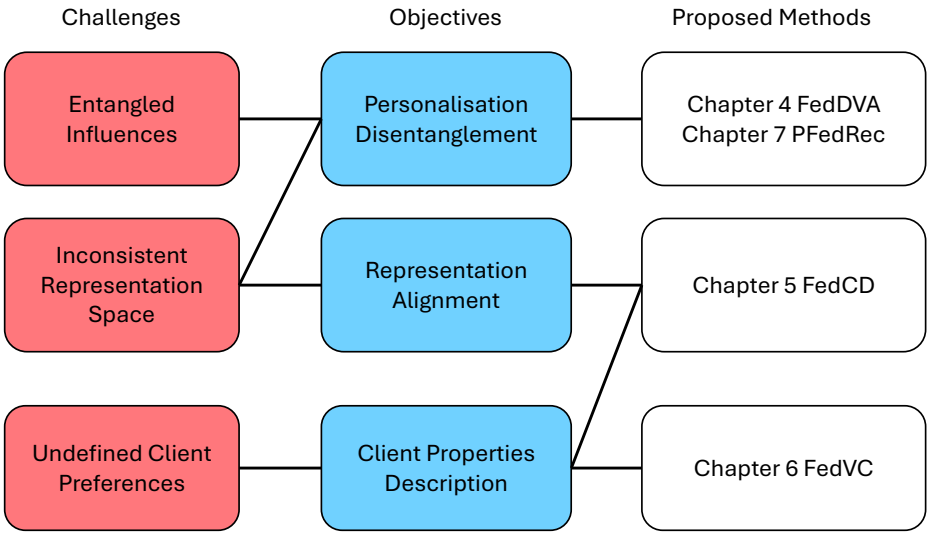


Figure 1.4: The relationship between research objectives and the challenges they solve.

sample [7]. Personalisation disentanglement requires a PerFL model learning to encode a sample into two irrelevant representations, one capturing client-specific knowledge and the other capturing client-agnostic knowledge. Then, a client can build interpretable models over the disentangled representations and identify preferences useful for making predictions, e.g., training a linear model over the disentangled representations of local data and studying the relationship between client-specific representations and model outputs.

This research achieves personalisation disentanglement by proposing a novel Federated Dual Variational Autoencoder (FedDVA), which employs two probabilistic encoders to infer the client-specific/-agnostic representations. The FedDVA produces a better understanding of the trade-off between global knowledge sharing and local personalisation in PerFL. Extensive experiments validate the advantages caused by disentanglement and show that models trained with disentangled representations substantially outperform those vanilla methods.

1.2.3.2 Representation Alignment

This research also studies aligning a global model's hidden layers to find a unified representation space describing client properties. The key motivation is that samples on the same client are influenced by identical client properties. They shall induce similar representations in describing their host client's properties, whilst samples from different clients are on the opposite.

Accordingly, the research proposes to impose constraints on a global model's hidden layers to decompose the latent representation space into two subspaces. One subspace is aligned with the above inductive bias that sample representations will be similar if they were from the same client. The other subspace is unrelated to clients and captures sample information. Then, the global model can precipitate client properties in its hidden layers when it is optimised for supervised tasks like classification, and the resulting representation space will be synchronised among clients along with the sharing of the global model.

To find the target representation space, a new federated learning framework on client-decorrelation (FedCD) is proposed. The FedCD formulates the representation alignment problem into an optimisation framework that clients can solve collaboratively along with the model training process. Experimental studies show that the FedCD can learn a unified representation space for client properties and a robust FL global model for one-model-for-all personalisation. The FedCD's global model can be directly deployed to the test clients with changing data distributions while achieving comparable performance to other personalised FL methods that require local model adaptation.

1.2.3.3 Client Properties Description

Although no significant definitions of client properties exist, an essential feature distinguishing model personalisation from unsupervised tasks is that each sample has a

distinct host client. One may assume that there are invisible labels of client indices providing supervised information for personalisation. This research calls the learning paradigm Client-Supervision.

The research will introduce Virtual Concepts (VC) to explicate client-supervised information. The VCs are representation vectors describing potential structure information implied in each client's training data. They can be learned independently of the supervised tasks by a novel FedVC algorithm, which facilitates understanding client properties and boosts model personalisation.

Experiments on real-world datasets show that the VCs can work as supervised information to train a global model that is robust to the changing distributions. Further study demonstrates that the VCs are useful in interpreting differences in model outputs caused by client properties.

1.3 Outline of the Thesis

The rest of this thesis is organised as follows:

Chapter 2 first gives a literature review on federated learning research, including general frameworks of federated learning (**Section 2.1**) and various ways to realise model personalisation (**Section 2.1.1**). Then, related techniques, i.e., Representation Disentanglement, Representation Alignment and Unsupervised Personalisation, are introduced in **Section 2.2.1.1**, **Section 2.2.1.2** and **Section 2.2.1.3**.

Chapter 3 gives formal definitions for federated learning settings and an ordinary framework of federated learning (**Section 3.1**). Then, **Section 3.2** introduces the learning objectives of the three types of personalised FL. The Variational Autoencoder framework is also introduced in **Section 3.3**.

Chapter 4 studies the problem of personalisation disentanglement. It introduces a novel Federated Dual Variational Autoencoder (**FedDVA**) framework, which explicitly

disentangles sample representations into client-agnostic and client-specific parts, i.e., a change in one dimension of the disentangled representation corresponds to a change in one type of the client-agnostic/-specific property while being irrelevant to the other. It also derives the Evidence Lower Bound (ELBO) to formulate the FedDVA into a unified optimisation framework that ordinary FL optimisation methods can solve. It evaluates FedDVA's performance from the perspective of interpretation and classification. Concretely, it restructures samples from the disentangled representations and shows that each type of representation will lead to a data manifold of the corresponding knowledge, which validates the effectiveness of personalisation disentanglement. In addition, it evaluates personalisation performance through training a lightweight classification over the disentangled representations, where FedDVA achieves competitive performance compared to state-of-the-art PerFL models.

Chapter 5 studies the representation alignment task to unravel client properties. It introduces a novel client-decorrelation mechanism (**FedCD**) to decompose an FL model's hidden space and align samples' latent representations to unravel client properties. Particularly, it proposes to impose orthogonality constraints on a DNN's hidden layers, restricting their outputs to vary along with axes aligned with clients' properties. Then, client-specific information will be encoded in a unified representation space and then be fed into a decision module along with class knowledge to make the final prediction. Moreover, it shows that the client-decorrelation mechanism could become a plug-in component to be integrated with any federated learning methods, which enables a vanilla FL model to output personalised results without on-device fine-tuning steps. Subsequently, a novel client-supervised optimisation framework is introduced, which formulates the representation alignment problem into a bi-level optimisation framework that clients can solve collaboratively under FL settings. Experiments on benchmark datasets validate that the disentangled client-specific representations will demonstrate

meaningful structures describing clients' properties, which helps us better understand features contributing to the personalisation and study their influences on the final decision. Besides, by comparisons with baseline methods, it shows that FedCD will obtain a robust FL global model that can be directly deployed to the test clients with changing distributions while achieving comparable performance to other personalised FL methods that require model adaptation.

Chapter 6 interprets personalisation using the Virtual Concepts (VC) as supervised information that not only provides a high-level summary of the data but also boosts the performance of distributed training in PerFL. The VCs are representations of potential structure information extracted from training data. They can be learned independently of the supervised tasks by a novel **FedVC** algorithm, which boosts the training of clients with statistically heterogeneous data. Experiments on real-world datasets show that the global model learned with VCs can be directly deployed on the test clients while achieving competitive performance without extra fine-tuning or personalisation. Further study also demonstrates that there will be a mapping between VCs and meaningful structure in data.

Chapter 7 concludes the research and discusses future works.

CHAPTER 2

LITERATURE REVIEW

2.1 Federated Learning

Federated Learning (FL) was first introduced in [75] to embody the principles of focused collection and data minimisation. It decomposes a machine learning task into subtasks and distributes them to a set of clients, e.g., smartphones and laptops, to carry out training steps. Only model updates will be collected during the process so that private data will remain closed compared to the conventional centralised machine learning paradigm. [75] proposed a fundamental framework FedAvg. Each client will synchronise a global model from a server and optimise the global model on its local data by gradient-descend methods. Then, the server will collect locally updated models and aggregate them by averaging their parameters. The FedAvg framework is compatible with most gradient-based optimisation methods, but its performance degenerates significantly when data distributions on clients shift.

Many strategies have been applied to handle this problem. [92] showed that multi-task learning is naturally suited to handle the statistical challenges of the FL and

proposed a novel systems-aware optimisation method, MOCHA. [56] proposed a FedProx algorithm, which constrains the optimisation steps on each client by a proximal regulariser. The authors proved that a model learned by the FedProx has more robust convergence than the FedAvg. In FetchSGD [88], clients upload gradient sketches instead of local models to the server, and the server applies momentum and error accumulation when aggregating the uploaded gradient sketches. [20] assumed the statistical heterogeneity across clients is concentrated in the labels. They proposed a FedRep algorithm to learn a global feature representation in the federation.

2.1.1 Personalised Federated Learning

In addition to sharing knowledge through a global model, many FL methods adapt the global model into a client-specific local model, leveraging the client's preferences to improve performance. For example, [18, 24, 74, 112] showed that steps as simple as fine-tuning a client's local data would improve the model's classification accuracy on that client. The scope of balancing global knowledge sharing and local preferences exploitation is referred to as **Personalised Federated Learning** (PerFL). There are three personalisation strategies from the perspective of model splitting and sharing.

2.1.1.1 Fully Personalised FL

Full personalisation denotes FL methods that adjust all parameters of a global model for individual clients. A straightforward approach is fine-tuning the entire global model using a client's local data [18, 21, 75], but this practice is usually limited by data volume and computation capabilities on clients. Then, [11, 29, 44] followed meta-learning [30] to learn a shared model initialisation by FedAvg so that clients can adapt the global model to local data effectively and efficiently. [55] proposed clients train personalised models individually while leveraging a shared regulariser endowed with global knowledge to

constrain the distributed training procedures. [90] proposed to learn a hyper-network that will generate personalised parameters directly. Recent work [10, 16, 99] utilised the graph-based structural information among clients to enhance the knowledge-sharing in PerFL, which enables clients to train the global and personalised models simultaneously.

2.1.1.2 Partial Personalised FL

Partial personalisation [82] splits an FL model into global and local parts. Clients share the global part as in vanilla FL but combine it with a privately trained local part to achieve personalisation. [57] is a simple but efficient case of these methods. It trains a model through the vanilla FedAvg except for preserving batch-normalisation modules locally. [20] introduced a method to learn a shared data representation across clients and unique classification heads for each client. [70] utilised a global and a local encoder to learn different representations for cross-client collaboration and personalisation. [121] applied the dual-encoder architecture to recommendation systems to help generate personalised recommendations while keeping users' data private and closed. [82] compared popular model splitting strategies from the convergence perspective and proposed two PerFL optimisation algorithms.

2.1.1.3 Group-wise Personalised FL

Group-wise personalisation aims to cluster clients into groups to share personalised models within each client group [33, 66, 74]. [112] validated that the cluster of individually trained local models would demonstrate consistent structure with the natural cluster regarding client properties, e.g., groups regarding user languages, which is promising to interpret the mechanism behind model personalisation. [125] utilised communication patterns between clients and the server to describe client preferences. It formulated this prior knowledge to cluster clients into daytime and nighttime modes to improve personalisation. [73] mixed the global and group-wise models to mitigate the clustering

collapse problem and balance cross-group knowledge sharing. [72] provided a theoretical analysis of popular clustering methods and proved their convergence.

2.1.2 Personalisation by Heterogeneous Models

While most PerFL methods aim to adapt a global model to specific tasks on clients, several works study the problem from the reversed aspect of aggregating heterogeneous models. Specifically, clients in this setting fulfil personalisation by task-specific model designs, and the challenge lies in how to share common knowledge across heterogeneous local models. [100] introduced FedProto, where clients maintain a set of prototypes rather than models by FedAvg. The training steps on each client will simultaneously minimise the classification error and keep sample embeddings close to the corresponding prototypes. [98, 101] leveraged contrastive methods to enhance sample-wise invariance encoding ability and aggregate outputs of multiple heterogeneous models.

2.1.3 Personalisation by Federated Foundation Models

Emerging foundation models (FM) like GPT have made significant progress in AI in both research and applications. However, as training an FM from scratch is challenging and costly, a popular way is to adapt a pre-trained general-purpose FM to specific tasks. Federated Foundation Models (FFM) integrate FMs into the FL framework so that clients can maintain a global FM to share fundamental knowledge [64] and personalise it toward local tasks through parameter-efficient fine-tuning. For example, [118] proposed a dual-adapter framework to balance global knowledge sharing and model personalisation for pre-trained FFMs. [14, 15] adapted pre-trained FFMs on distributed meteorological data to conduct weather predictions. [58, 59, 120] proposed federated recommendation systems that can generate personalised recommendations based on FFMs while keeping user data closed and private.

2.2 Interpretable Machine Learning

Interpretable Machine Learning is a field studying the comprehensibility of machine learning models. It aims to illustrate why a model makes specific predictions [47, 77] and reveals knowledge either contained in data or learned by the model [79]. Traditional statistical models like linear regression and decision trees are essentially interpretable, but they have limited capacity to handle complex problems like image recognition. So, recent research focuses on disclosing knowledge in deep models that are more powerful and complicated.

The most extensively discussed interpreting method is visualising feature attributions by propagating gradients [89, 91] or calculating the Shapley value [69, 96]. However, although various works attempt to verify the interpretability of existing attribution maps [9, 25, 95], there is still controversy about whether their interpretation was consistent with the ground truth knowledge [1, 50, 104].

2.2.1 Interpreting Model Personalisation

By methods discussed in **Section. 2.1**, model personalisation is mainly embodied through clients' parameter-sharing strategies, i.e., which parameters are shared and whom they are shared with. However, the intricate parameters involved could be too abstract for human beings to understand the logic behind personalisation. Three technical aspects will help to get a deeper insight into this opaque procedure.

2.2.1.1 Representation Disentanglement

Since most PerFL models are trained in an end-to-end schema, client-specific and client-agnostic knowledge will be entangled and scattered throughout their entire hidden layers. Identifying properties that contribute to personalisation requires disentangling clients' influence on a model's decisions. Specifically, disentanglement refers to learning

a representation where a change in one dimension corresponds to a change in one factor of variation of the sample while being relatively invariant to changes in other factors [7]. Variational autoencoder (VAE), as well as its variations [13, 35, 39, 49, 51, 93], is a popular framework for learning disentangled representations. It is attractive for elegant theoretical backgrounds and high computation efficiency.

However, previous research [63] proved that unsupervised learning of disentangled representations is fundamentally impossible without inductive biases on both the models and the data. Prior knowledge regarding the model or the data is necessary for VAE models disentangling client-specific and client-agnostic representations in PerFL. Besides, since most VAE models will maintain a decoder to reconstruct samples from latent representations, deploying VAE in FL risks leaking privacy by recovering local data across clients.

Chapter 4 manages to disentangle clients' preferences using VAE frameworks. It introduces an inductive bias that samples on the same client are influenced by identical properties and integrates this prior knowledge into a novel Dual Variational Autoencoder (FedDVA). The disentanglement task is then formulated into an optimisation problem of maximising an Evidence Lower Bound (ELBO). Compared with the existing VAE architecture, only the encoder modules are shared among clients, which takes advantage of VAE's disentanglement capability while keeping user privacy safe.

2.2.1.2 Representation Alignment

The distributed training procedure in FL introduces a new challenge of unaligned representation space among locally trained models. Clients may encode the same sample into very different representations for the same task. Thus, aligning the representation space across clients is necessary to help find the mapping between representations and meaningful properties explaining model behaviours.

[108] proposed a layer-wise matching algorithm to match every hidden unit of two local models before aggregating their parameters. [113] leveraged global semantic knowledge to conduct explicit local-global feature alignment. Concept Vectors are new ways to investigate the decision process of a black-box model [17, 32, 48, 53]. These approaches learn a set of concept vectors, i.e., decision hyperplanes aligned with certain concepts, to classify hidden units inside a DNN. End users can unravel concepts leading to a specific output by inspecting whether relevant hidden units are activated. Notably, the concept vector method allows human-in-the-loop operations to test and quantify a concept's influence, which is critical in studying the causation of a prediction [48, 53]. A disadvantage is that these methods require supervised information for each concept of interest, either by auxiliary datasets [17, 48] or labels [32, 53]. Then, they are limited to working in scenarios where concepts of interest are enumerable and easily depicted.

Chapter 5 introduces a novel client-decorrelation mechanism (**FedCD**) to decompose an FL model's hidden space and align samples' latent representations to unravel client properties. The FedCD imposes orthogonality constraints on a DNN's hidden layers, limiting their outputs to vary along with axes aligned with clients' properties. Client-specific information is then projected into a unified representation space across clients, allowing downstream modules to make personalised decisions without needing extra personalisation steps.

2.2.1.3 Unsupervised Personalisation

Since there is no supervised information like labels to measure a model's personalisation, one needs to deliberate on data distributions to clarify related factors. [26] proposed a distribution-fusion method to aggregate local models trained on statistically heterogeneous data. It represented clients' local data distributions by several virtual fusion components extracted from client-specific models and aggregated them into a shared global model. [68] proposed an unsupervised FL method for problems where the class-

prior probabilities are shifted while the class-conditional distributions are shared among the unlabelled data. It transformed the unlabelled data into surrogate-labelled data on each client and then utilised the surrogate labels as supervised information to train a surrogate global model.

A typical problem of the above methods is that they are sensitive to multiple factors, e.g., the number of predefined latent distributions. It is hard to ensure local distributions are properly represented or are able to map to meaningful client properties [112]. **Chapter 6** introduces Virtual Concepts (VCs) to describe clients' properties explicitly. Compared with the existing methods, the proposed FedVC integrates VCs into PerFL's training process, converting the unsupervised personalisation task into a supervised task. It not only boosts the performance of distributed training in PerFL but also provides a high-level summary of the data to facilitate understanding of model personalisation.

CHAPTER 3

PRELIMINARIES

3.1 Federated Learning

In a federated learning system with K clients, each client is indexed by k. $\{\mathbf{x}_1^{(k)}, \mathbf{x}_2^{(k)}, ..., \mathbf{x}_{N_k}^{(k)}\} \in \mathscr{X}^{(k)}$ denotes samples on the k-th client, and $\{y_1^{(k)}, y_2^{(k)}, ..., y_{N_k}^{(k)}\} \in \mathscr{Y}^{(k)}$ are their labels. N_k is the number of samples. The FL task is to find the optimal parameters ω^* for a global model $f(\mathbf{x};\omega)$ by minimising the total loss of all clients as the following optimisation problem:

$$\omega^* = \arg\min_{\omega} \sum_{k=1}^{K} \alpha_k \mathscr{L}_k(\omega) \tag{3.1}$$

where $\mathscr{L}_k(\omega) = (1/N_k) \sum_{i=1}^{N_k} l(f(\mathbf{x}_i^{(k)};\omega), y_i^{(k)})$ is the supervised loss on the k-th client, and α_k is its weight. In particular, in the ordinary FedAvg framework [75], α_k is the fraction of the size of the client's training data, i.e., $\alpha_k = N_k / \sum_{k'=1}^{K} N_{k'}$.

The optimisation process of **Equation 3.1** consists of two steps. 1) each client synchronises a global model from a server and updates ω privately by gradient descent methods, i.e., $\omega_k = \omega - \nabla \mathscr{L}_k(\omega)$; 2), the server collects and aggregates local updates by av-

eraging parameters, i.e., $\omega = \sum_{k=1}^{K} \alpha_k \omega_k$. A fundamental learning algorithm is described in **Algorithm 2**.

Algorithm 2 Federated Learning

Input: communication rounds R, epochs in each round E, learning rate λ, batch size B
Output: optimal parameters ω^*

server initialises parameters ω
for r from 0 to R **do** ▷ communication rounds
  server selects a set of clients $\mathbb{C}$
  for $k \in \mathbb{C}$ **parallel do**
    client k synchronises ω from the **server** ▷ network traffic
    $\omega_k \leftarrow$**ClientUpdate**(ω)
  end for
  server collects local updates ω_k, $k \in \mathbb{C}$ ▷ network traffic
  $\omega \leftarrow \sum_{k \in \mathbb{C}} \alpha_k \omega_k$
end for
return ω

ClientUpdate(ω)

for e from 0 to E **do**
  for b from 0 to N_k/B **do**
    sample a batch of data $\mathbb{B}$
    $\omega = \omega - \lambda \nabla \mathscr{L}(\omega; \mathbb{B})$
  end for
end for
return ω

3.2 Personalised Federated Learning

PerFL leverages cross-client collaboration but learns a personalised model $f(\mathbf{x}; \omega, \mu_k)$ for each client, where ω denotes parameters shared, and μ_k denotes parameters for the k-th client. The learning task can be formulated into a unified optimisation problem as below:

$$\omega^*, \{\mu_k^*\}_{k=1}^K = \arg \min_{\omega, \{\mu_k\}_{k=1}^K} \sum_{k=1}^{K} \alpha_k \mathscr{L}_k(\omega, \mu_k) \tag{3.2}$$

There are different types of personalisation according to how to define ω and μ_k.

Full Personalisation: A local model is fully parameterised by μ_k and the global model guides the local training process. For example, [55] utilises ω as a regulariser to

constraint μ_k by minimising $\|\omega - \mu_k\|_2$. Formally, there is

$$\mathscr{L}_k(\omega, \mu_k) = (1/N_k) \sum_{i=1}^{N_k} l(f(\mathbf{x}_i^{(k)}; \mu_k), y_i^{(k)}) + a\|\omega - \mu_k\|_2 \tag{3.3}$$

where a is a weight tuning the two parts of the loss; [29] trains the global parameter ω as an initialisation to local models. That is

$$\mathscr{L}_k(\omega, \mu_k) = (1/N_k) \sum_{i=1}^{N_k} l(f(\mathbf{x}_i^{(k)}; \mu_k), y_i^{(k)}),\ s.t. \mu_k = \omega - \nabla \mathscr{L}_k \tag{3.4}$$

Algorithm 2 can be applied to optimise objective functions in full personalisation.

Group-wise Personalisation: Clients are clustered into groups to maintain a personalised model within each group, i.e., for clients in the group G_c, there is $\omega_c = \sum \alpha_k \mu_k [\![k \in G_c]\!]$ [33, 74]. The learning object of group-wise personalisation can be formulated into the optimisation problem below:

$$\begin{aligned} \{\omega_c^*\}_{c=1}^C &= \arg\min_{\omega} \sum_{c=1}^{C} \sum_{k=1}^{K} \alpha_k r_{(c,k)} \mathscr{L}_k(\omega) \\ s.t.\ r_{(c,k)} &= \arg\min_{r_{(c,k)}} \sum_{c=1}^{C} r_{(c,k)} \|\mu_k - \omega_c\|_2, r_{(c,k)} \in \{0,1\},\ \sum_{c=1}^{C} r_{(c,k)} = 1, \text{ for } k = 1:K \end{aligned} \tag{3.5}$$

The vanilla federated learning method is utilised to learn the 'global' model in each group.

Partial Personalisation: ω and μ_k constitute the global and personal parts of a local model, where ω is shared through the fundamental FL method and μ_k is trained individually on each client. e.g., ω could be parameters of a shared backbone model, and μ_k is a classification head for the k-th client (**Figure 1.1(b)**). Generally, a client will update ω and μ_k alternately before sharing ω across clients. A typical local updating process [82] is introduced in **Algorithm 3**.

3.3 Variational Autoencoder

Variational Autoencoder [51] assumes any sample x is corresponding to a latent representation z, whose prior distribution is the standard normal distribution, i.e., $p(\mathbf{z}) =$

Algorithm 3 ClientUpdate in Partial Model Personalisation

ClientUpdate(ω)

for e from 0 to E_ω **do**
  for b from 0 to N_k/B **do**
    sample a batch of data $\mathbb{B}$
    $\omega = \omega - \lambda_\omega \nabla_\omega \mathscr{L}(\omega, \mu_k; \mathbb{B})$
  end for
end for
for e from 0 to E_{μ_k} **do**
  for b from 0 to N_k/B **do**
    sample a batch of data $\mathbb{B}$
    $\mu_k = \mu_k - \lambda_{\mu_k} \nabla_{\mu_k} \mathscr{L}(\omega, \mu_k; \mathbb{B})$
  end for
end for
return ω

$\mathcal{N}(\mathbf{z}; 0, I)$. It learns a probabilistic encoder to approximate the variational posterior $q(\mathbf{z}|x)$ and a decoder to reconstruct the sample x from the latent representation z sampled from $q(\mathbf{z}|x)$.

In general, the encoder is a neural network whose outputs are the mean and covariance of the $q(\mathbf{z}|x)$, that is $q(\mathbf{z}|x) = \mathcal{N}(\mathbf{z}; \mu(x), \Sigma(x))$ and the covariance matrix Σ is assumed to be diagonal for computation simplicity. The decoder is another neural network reconstructing x by maximising the log-likelihood $\log p(x|z)$. An illustration of the VAE framework is shown in **Figure 3.1**.

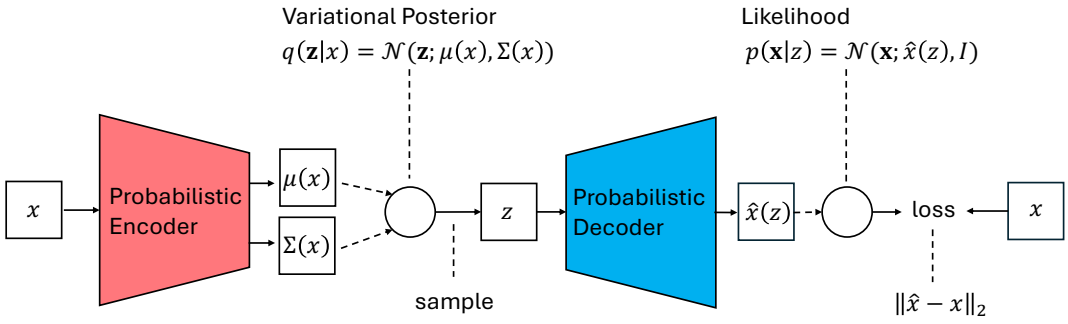


Figure 3.1: A pipeline of VAE framework

The learning objective of VAE can be formulated into an optimisation problem of

maximising the Evidence Lower BOund (ELBO) as below

$$\theta^*, \varphi^* = \arg\max_{\theta,\varphi} \sum_{i=1}^{N} \mathbb{E}_{q(\mathbf{z}|x_i)}[\log p(x_i|\mathbf{z})] - D_{KL}(q(\mathbf{z}|x_i)||p(\mathbf{z})) \tag{3.6}$$

where θ and φ are the parameters of the probabilistic encoder and the decoder respectively. The first term on the RHS of **Equation 3.6** measures the decoding performance of latent representation z and the second term measures the KL-divergence between the posterior $q(\mathbf{z}|x)$ and the prior $p(\mathbf{z})$. Gradient-based optimisation methods can be applied with the help of the reparameterisation trick [51].

CHAPTER 4

PERSONALISATION DISENTANGLEMENT FEDERATED LEARNING

4.1 Motivations

In PerFL tasks, clients need to train local models with raw samples entangled with client-specific and client-agnostic knowledge. Then, they will eliminate personalised information before sharing local updates to update the global model. Most PerFL methods mitigate these contradictory operations by designing new model architectures or adapting gradient-descent strategies [18, 29, 55, 60, 108]. **Figure 4.1** gives examples of samples entangled with different knowledge. A PerFL algorithm needs to train local models on samples entangled with digits and marks and filter the impacts of marks when updating the global model. Meanwhile, a personalisation process may fine-tune the global model on local data and fit the client-specific marks again to obtain a client-specific model.

Personalisation disentanglement aims to encode a sample into two disentangled representations, each capturing one type of the above knowledge. The disentangled representations will help identify essential knowledge constituting a model's personalisation.

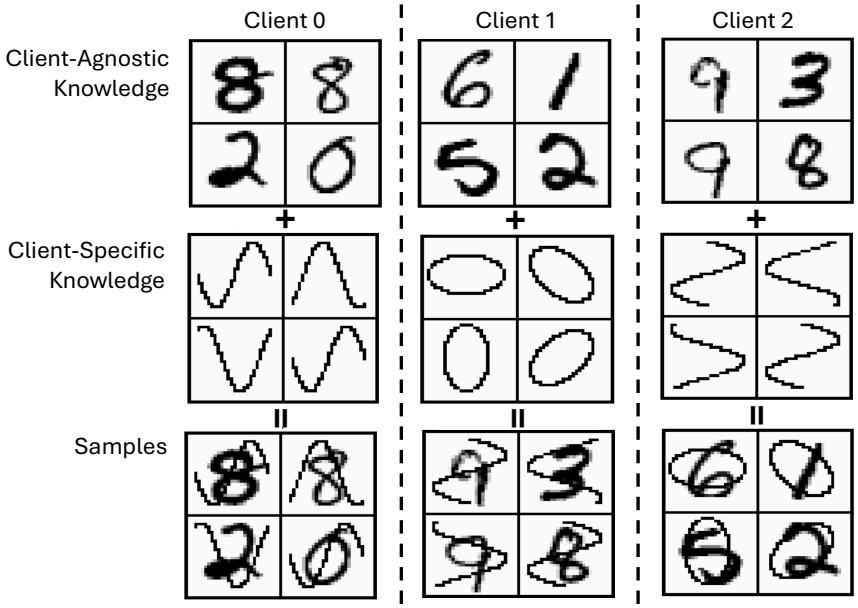
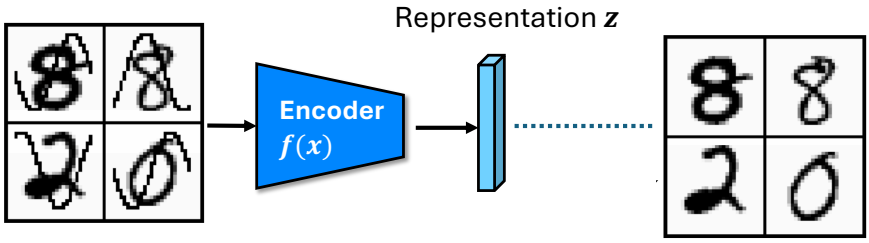


Figure 4.1: An example of samples with entangled knowledge. Knowledge about handwritten digits is client-agnostic and will be shared through the global model, but knowledge about sinusoidal and elliptical marks is client-specific.

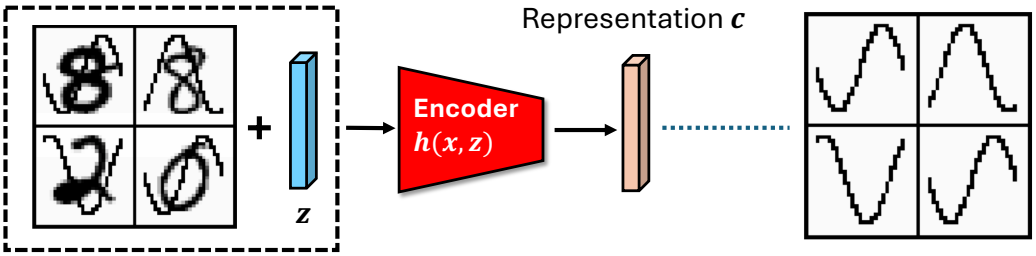
Besides, a client can build a lightweight model over the disentangled representations for downstream tasks, e.g., classification. As the two types of knowledge are separated in different representations, the downstream model will have better efficiency without being hampered by the contradictory steps above.

The key idea is that a global model intends to learn knowledge applicable to all clients. Then, one can train a global encoder to capture the client-agnostic knowledge and later train another encoder to learn to 'minus' the client-agnostic knowledge from the sample to get the client-specific knowledge. **Figure 4.2** gives an example of how the dual encoders work.

To this end, we develop a Federated Dual Variational Autoencoding framework (FedDVA) [114], where clients in the federation share two encoders inferring the above representations. The two encoders are trained collaboratively by fundamental FL algorithms like FedAvg [75]. Clients update the encoders locally by maximising a novel client-specific Evidence Lower BOund (ELBO). Then, a server collects local updates and



(a) one encoder learns a representation for client-agnostic knowledge



(b) another encoder infers the client-specific knowledge given the client-agnostic knowledge

Figure 4.2: Motivation of Dual Encoders. (a) an encoder will learn to encode client-agnostic knowledge. (b) another encoder will learn to eliminate client-agnostic knowledge with the help of client-agnostic representations.

aggregates them by averaging parameters. Moreover, the two encoders are cascaded and constrained by different prior knowledge so that each encoder will capture only one type of knowledge mentioned above.

The main contributions of the method are summarised as follows:

- It proposes a novel FedDVA method to achieve personalisation disentanglement, which provides a better understanding of PerFL and improves the efficiency of downstream classification models.
- It derives a client-specific ELBO to optimise FedDVA and analyse its capability of capturing personalised knowledge.
- Experiments on real-world datasets validate FedDVA's effectiveness in personalisation disentanglement and show that classification models will converge fast

and achieve competitive classification performance when trained on disentangled representations.

4.2 Methodology

4.2.1 Problem Formulation

FedDVA aims to learn disentangled sample representations for client-agnostic and client-specific knowledge. The target representations can be denoted as z and c . Since z is irrelevant to clients, FedDVA assumes samples on each client have the same prior distribution $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; 0, I)$. Meanwhile, since samples in FL are private and distributed, the prior distribution of c is unknown and varies among clients. It can be denoted as $p_k(\mathbf{c})$ for the k-th client. It is worth noting that no assumptions on the $p_k(\mathbf{c})$ can be made as there is no guarantee that the relationship between the assumed distributions is consistent with the relationship between client preferences. For example, clients with similar personalities shall have a similar distribution of $p_k(\mathbf{c})$. Alternatively, FedDVA assumes the mixture distribution $q(\mathbf{c}) = \sum_{k=1}^{K} \alpha_k p_k(\mathbf{c})$ to be the standard Gaussian distribution $\mathcal{N}(\mathbf{c}; 0, I)$.

4.2.2 Dual Encoders

The proposed FedDVA learns the target representations through two encoders as illustrated in **Figure 4.3**. For any sample x, an encoder first infers a variational posterior $q(\mathbf{z}|x) = \mathcal{N}(\mathbf{z}; \mu(x), \Sigma(x))$ for the representation z. Then another encoder infers the variational posterior $q(\mathbf{c}|x, z) = \mathcal{N}(\mathbf{c}; \hat{\mu}(x, z), \hat{\Sigma}(x, z))$, which conditioned on both the sample x and the representation z, for extracting impacts of personalised knowledge. In addition, a client-specific local decoder will evaluate the decoding performance of the representations z and c. It is implemented by a neural network maximising the client-specific

log-likelihood $\log p_k(x|z,c)$. The negative ELBO optimising FedDVA is in **Equation 4.1**

$$\ell^{dva}(\theta,\varphi_k;x) = -\mathbb{E}_{q(z|x)}[\mathbb{E}_{q(c|x,z)}[\log p_k(x|z,c)] + B\mathscr{R}_c(q(c|x,z))] + A\mathscr{R}_z(q(z|x)) \tag{4.1}$$

where θ denotes parameters of the shared encoders, φ_k denotes the parameters of the local decoder specific to the k-th client, $\mathscr{R}_z(q(z|x))$ and $\mathscr{R}_c(q(c|x,z))$ denote the regularizers for the posteriors $q(\mathbf{z}|x)$ and $q(\mathbf{c}|x,z)$, A and B are their importance weights.

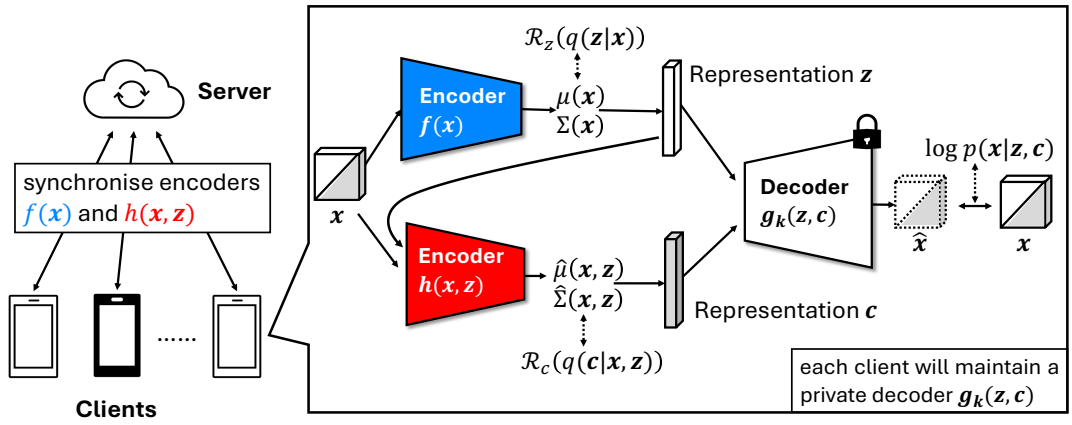


Figure 4.3: The architecture of FedDVA. An encoder $f(x)$ (Blue) will first infer the posterior $q(\mathbf{z}|x)$, and then another encoder $h(x,z)$ (red) will infer the conditional posterior $q(\mathbf{c}|x,z)$. The decoder $g(z,c)$ (white) will try to reconstruct x from z and c.

Similar to traditional VAE models, the posterior $q(\mathbf{z}|x)$ can be regularised by the KL-divergence $D_{KL}(q(\mathbf{z}|x)||p(\mathbf{z}))$, which enforces the distribution of the representation z to be close to the standard Gaussian distribution no matter the client it is on. However, it would be a challenge to regularise the representation c as we have no prior knowledge about the distribution $p_k(\mathbf{c})$. FedDVA handles the problem by a slack regulariser $D_{KL}(q(\mathbf{c}|x,z)||q(\mathbf{c}))$ combing with a constrain that

$$D_{KL}(q(\mathbf{c}|x,z)||q(\mathbf{c})) - D_{KL}(q(\mathbf{c}|x,z)||\bar{p}_k(\mathbf{c})) \geq \xi_k \tag{4.2}$$

where $\bar{p}_k(\mathbf{c}) = \frac{1}{|\mathscr{D}_k|}\sum_{x\in\mathscr{D}_k} q(\mathbf{c}|x,z)$ is the mixture distribution of $q(\mathbf{c}|x,z)$ of samples on the k-th client and $\xi_k > 0$ is a hyperparameter. Intuitively, $\bar{p}_k(\mathbf{c})$ is an estimator of $p_k(\mathbf{c})$

and the **Inequation 4.2** requires $q(\mathbf{c}|x,z)$ to be at least ξ_k closer to $\bar{p}_k(\mathbf{c})$ than to $q(\mathbf{c})$. We will discuss it in **Section 4.3** and show it helps the representation c to capture client properties. Combining the KL-divergence and the constrain in **Inequation 4.2**, regularisers $\mathscr{R}_z(q(\mathbf{z}|x))$ and $\mathscr{R}_c(q(\mathbf{c}|x,z))$ of **Equation.4.1** can be written as

$$\mathscr{R}_z(q(\mathbf{z}|x)) = D_{KL}(q(\mathbf{z}|x)||p(\mathbf{z})) \tag{4.3}$$

$$\mathscr{R}_c(q(\mathbf{c}|x,z)) = \max(\xi_k + D_{KL}(q(\mathbf{c}|x,z)||\bar{p}_k(\mathbf{c})), D_{KL}(q(\mathbf{c}|x,z)||q(\mathbf{c}))) \tag{4.4}$$

They can be computed and differentiated without estimation (see **Appendix A.1.2**), and therefore **Equation 4.1** can be optimised by gradient-based methods.

4.2.3 Optimisation

To learn the encoders collaboratively by clients, the learning objective of FedDVA is formulated as follows:

$$\theta^*, \varphi_1^*, ..., \varphi_K^* = \arg\min_{\theta, \varphi_1 ... \varphi_K} \sum_{k=1}^{K} \alpha_k \mathscr{L}_k(\theta, \varphi_k; \mathscr{D}_k) \tag{4.5}$$

where $\mathscr{L}_k(\theta, \varphi_k; \mathscr{D}_k) = \sum_{x \in \mathscr{D}_k} \ell^{dva}(\theta, \varphi_k; x)$. Then gradient steps optimising **Equation 4.5** consists of the following two parts

$$\varphi_k' = \varphi_k - \eta \nabla_{\varphi_k} \mathscr{L}_k(\theta, \varphi_k; \mathscr{D}_k), 1 \le k \le K \tag{4.6}$$

$$\theta' = \theta - \lambda \sum_{k=1}^{K} \alpha_k \nabla_\theta \mathscr{L}_k(\theta, \varphi_k; \mathscr{D}_k) \tag{4.7}$$

where η and λ are their learning rates. **Equation 4.6** updates the client-specific decoders and is processed by each client independently. **Equation 4.7** updates the global encoders shared in the federation. It can be implemented by most FL algorithms like FedAvg. Concretely, $\theta' = \sum_{k=1}^{K} \alpha_k \theta_k'$, where

$$\theta_k' = \theta_k - \lambda \nabla_\theta \mathscr{L}_k(\theta_k, \varphi_k; \mathscr{D}_k), 1 \le k \le K \tag{4.8}$$

and **Equation 4.8** is performed by each client independently. But it is worth noting that the optimisation steps of **Equation 4.6** and **Equation 4.8** are asynchronous. As only a subset of clients will participate in the optimisation process in each communication round [75], client-specific decoders may not coincide with the shared encoders. A client needs to update φ_k first and later the θ. Complete pseudo-codes of the optimisation process are in **Algorithm 4**.

Algorithm 4 FedDVA

Input: R: communication rounds, M: number of clients sampled each round; B: batch size; E epochs; λ and η: learning rates; ξ_k: the constraint threshold in Inequation 4.2

server initialises $\theta^{(0)} \leftarrow \theta$
for r from 0 to R **do**
    server selects M clients $\mathbb{C}$
    for $k \in \mathbb{C}$ **parallel do**
        client k synchronises θ from the **server**
        $\theta_k^{(r+1)} \leftarrow$**ClientUpdate(**$\theta^{(r)}$**)**
    end for
    server collects local updates θ_k, $k \in \mathbb{C}$
    $\theta^{(r+1)} \leftarrow \sum_{k \in \mathbb{C}} \alpha_k \theta_k^{(r+1)}$
end for
return θ

ClientUpdate(θ**)**

Initialise $\theta_k \leftarrow \theta$, $\varphi_k \leftarrow \varphi$
for e from 0 to E **do**
    for b from 0 to N_k/B **do**
        sample a batch of data $\mathbb{B}$
        update φ_k by **Equation 4.6**
    end for
end for
$\varphi \leftarrow \varphi_k$
for e from 0 to E **do**
    for b from 0 to N_k/B **do**
        sample a batch of data $\mathbb{B}$
        update θ_k by **Equation 4.8**
    end for
end for
return θ_k

4.3 Theoretical Analysis

This section discusses the ELBO corresponding to **Equation 4.1** and shows it can capture client properties.

From the perspective of variational inference, the optimal posteriors $q(\mathbf{z}|x)$ and $q(\mathbf{c}|x,z)$ are the ones maximising the following EBLOs jointly

$$\log p_k(x) \geq ELBO_z(x,k) = \mathbb{E}_{q(z|x)}[\log p_k(x|z)] - D_{KL}(q(\mathbf{z}|x)||p(\mathbf{z})) \tag{4.9}$$

$$\log p_k(x|z) \geq ELBO_c(x,z,k) = \mathbb{E}_{q(c|x,z)}[\log p(x|z,c)] - D_{KL}(q(\mathbf{c}|x,z)||p_k(\mathbf{c})) \tag{4.10}$$

where the subscript k means the distribution is specific to the k-th client. Ideally, $\log p(x|z,c)$ is a client irrelevant log-likelihood modelling the sample generating process, that is $p_k(x) = \iint p(x|z,c)p(z)p_k(c)\mathrm{d}z\mathrm{d}c$ (Details of the derivation is given in **Appendix A.1.1**). But **Equation 4.10** is hard to compute in practice. Besides the unknown prior knowledge $p_k(\mathbf{c})$, the client irrelevant log-likelihood $\log p(x|z,c)$ is unavailable in FL. For example, sharing $\log p(x|z,c)$ in the federation risks privacy leakage as it has the capability to generate samples.

As an alternative, FedDVA optimises the posterior $q(\mathbf{c}|x,z)$ by maximising the ELBO in **Equation 4.11**

$$\log p_k(x|z) \geq ELBO'_c(x,z,k) = \mathbb{E}_{q(c|x,z)}[\log p_k(x|z,c)] - D_{KL}(q(\mathbf{c}|x,z)||q(\mathbf{c})) \tag{4.11}$$

which is equivalent to **Equation 4.10**, except for that the slack regulariser $D_{KL}(q(\mathbf{c}|x,z)||q(\mathbf{c}))$ degenerates the capability of capturing difference between clients. Specifically, the overall KL-divergence between $q(\mathbf{c}|x,z)$ and $q(\mathbf{c})$ of samples on the same client is

$$\mathbb{E}_{p_k(x)}[\mathbb{E}_{q(z|x)}[-H(q(\mathbf{c}|x,z))]] + H(\bar{p}_k(\mathbf{c}), q(\mathbf{c})) \tag{4.12}$$

which requires the distribution of representation c to be close to $q(\mathbf{c})$ wherever the samples are. **Inequation 4.2** helps resolve the problem through introducing an inductive

bias that the posterior $q(\mathbf{c}|x,z)$ of samples on the same client is closer to $p_k(\mathbf{c})$ than to $q(\mathbf{c})$, with which $D_{KL}(\bar{p}_k(\mathbf{c})||q(\mathbf{c})) \geq \xi_k$ holds (Details of the derivation is given in **Appendix A.1.1.3**). Finally, replacing $p_k(x|z)$ in **Equation 4.9** with **Equation 4.11**, we have the loss function described in **Equation 4.1** and the hyperparameter ξ_k helps determinate the degree of 'penalisation' representation c captured. The larger the ξ_k is, the more personalised representation c is learned.

4.4 Experiments

This section evaluates the performance of FedDVA. First, it verifies FedDVA's disentanglement effectiveness by exploring data manifolds of samples decoded from the disentangled representations z and c. Then, it evaluates FedDVA's capability for classification tasks on Non-I.I.D. data. It trains lightweight personalised classification heads over the disentangled representations and compares its performance with state-of-the-art methods.

4.4.1 Personalisation Disentanglement

This experiment empirically studies FedDVA's disentanglement capability on two real-world datasets with different personalisation settings. First, it visualises distributions of the learned representations z and c to verify that the two representations are uncorrelated. Then, it explores data manifolds of samples decoded from the learned representations to verify that the representation z captures the client-agnostic knowledge and the representation c captures the client-specific knowledge.

4.4.1.1 Synthesised Digits

MNIST[1] is a benchmark dataset of handwritten digits with 60,000 training images and 10,000 testing images. This experiment uniformly allocates them to a set of clients

[1] https://yann.lecun.com/exdb/mnist/

and synthesises them with client-specific marks. Examples of synthesised digits are in **Figure 4.4**.

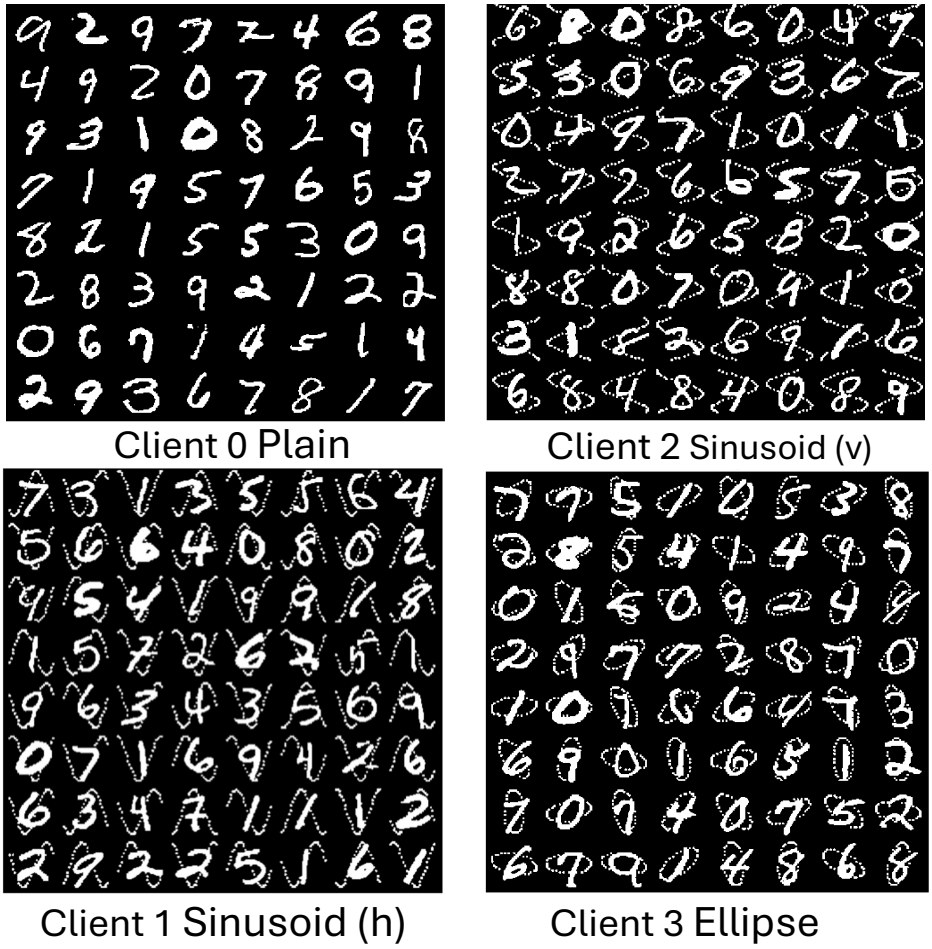
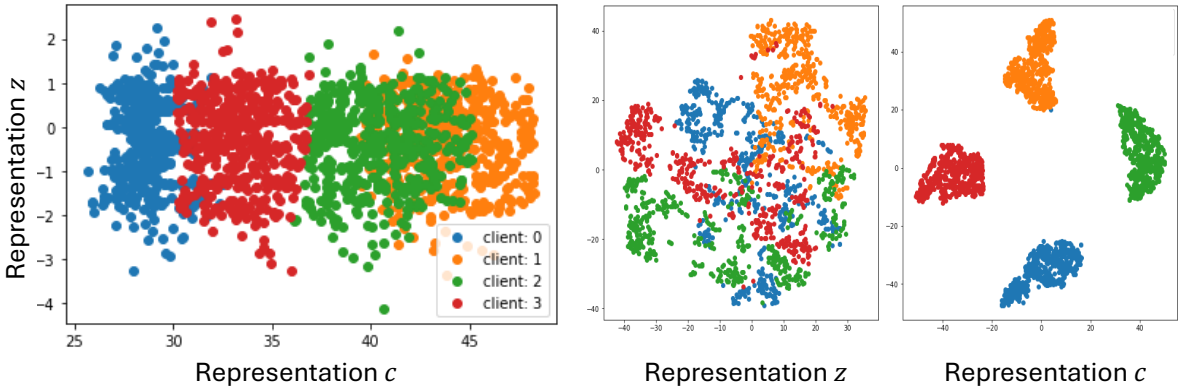


Figure 4.4: Examples of synthesised digits. Each quadrant displays samples from a client, and each client is related to one type of mark, i.e., horizontal/vertical sinusoids on random phrases or randomly rotated ellipses, and no marks on the Client 0

Representation Distribution: The experiment trains a FedDVA model on the setting above and visualises distributions of the learned representations z and c. There are two trials with different settings in the dimensions of the target representations. The first trial sets the dimension of each type of representation to be one. **Figure 4.5(a)** visualises the resulting z and c. It can be found that distributions of the client-agnostic representation z (vertical) are in an overlapped range even though they are learned on

different clients. On the other hand, distributions of the client-specific representation c (horizontal) are clustered regarding their clients and separable. The result verifies the effectiveness of FedDVA that the two types of representations are uncorrelated.



(a) distribution of 1-dimension z and c

(b) t-SNE scatterplots of 8-dimension z and c

Figure 4.5: Representation distributions of the synthesised digits. Each dot denotes the representation of a sample, and colours correspond to clients. (a) scatter plot of 1-dimension z (vertical) and c (horizontal); (b) t-SNE embeddings of 8-dimension z (left) and c (right).

Another trial sets the dimension of each type of representation to be eight. **Figure 4.5(b)** displays t-SNE embeddings [40] of the learned z and c. It can be found that the two representations are uncorrelated. The client-agnostic representations (left) are mixed and irrelevant to client preferences. The client-specific representations (right) are clustered regarding clients.

Data Manifolds: The experiment explores data manifolds of samples decoded from the learned representations to verify that the uncorrelated representations will capture the relevant knowledge. **Figure 4.6** studies manifolds of data decoded from one-dimension z and c on Client 3. Results show that digits will vary along with changes in client-agnostic representation z (vertical) while being irrelevant to changes in client-specific representation c (horizontal); Elliptical marks will rotate along with changes c while being irrelevant to changes in z. Similar results are also observed on other clients

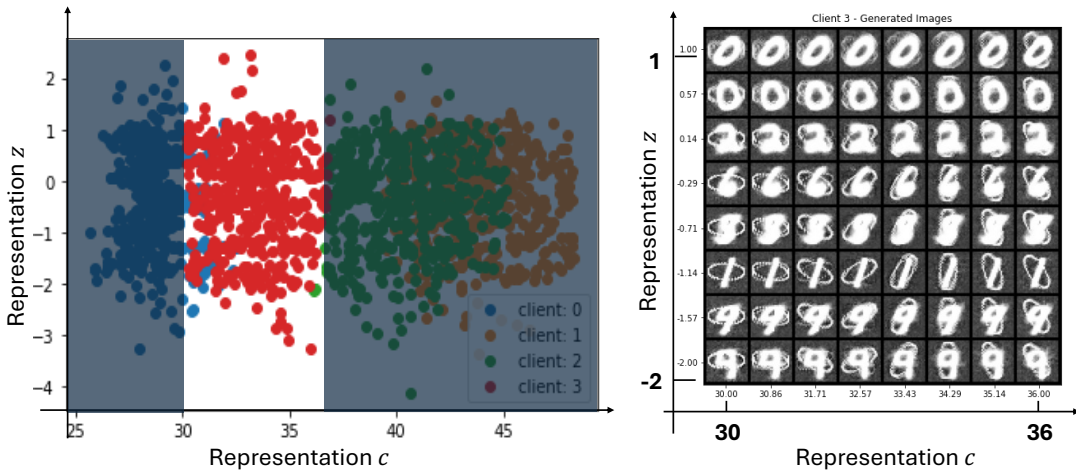


Figure 4.6: Data manifolds of decoded digits on Client 3. (left) Red dots are distributions of z and c on Client 3; (right) Data manifolds of digits decoded from z (vertical) and c (horizontal).

(**Figure 4.7**), which denotes that the two types of representations are able to capture the related knowledge and FedDVA succeed in personalisation disentanglement.

4.4.1.2 CelebA

CelebA[2] is a large-scale face dataset containing 202,599 face images of celebrities. This research allocates them to clients according to hairstyles in images so that samples of the same client demonstrate a bias towards similar properties. Examples of personalised face images are shown in **Figure 4.8**.

Experiments similar to the ones in **Section 4.4.1.1** are performed. **Figure 4.9(a)** visualises the scatter plots of 1-dimension z and c and **Figure 4.9(b)** shows the t-SNE embeddings of 8-dimension z and c. The result verifies the effectiveness of FedDVA that the two types of representations are uncorrelated.

In addition, manifolds of data decoded from the learned representations are shown in **Figure 4.10**. It can be found that general properties like identities and backgrounds are disentangled with personalised properties like hairstyles and angles.

[2]https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

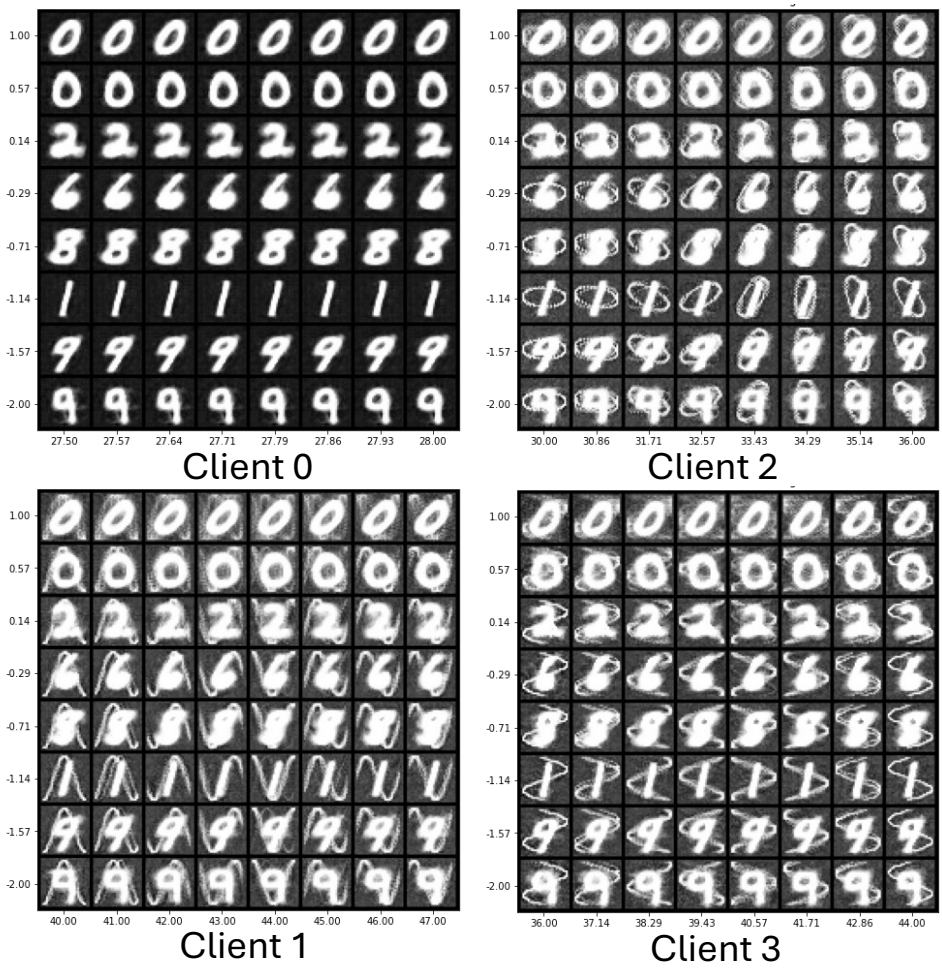


Figure 4.7: Data manifolds of digits decoded from the learned representations

4.4.2 Personalised Classification

This section evaluates the classification performance of representations learned by FedDVA. It tunes the dual encoders along with local lightweight classification heads and compares their performance with vanilla FL algorithms, i.e., FedAvg [75], FedAvg+Fine Tuning [18] and DITTO [55]. Two personalisation settings are evaluated.

4.4.2.1 Feature Shift Settings

The experiment evaluates FedDVA's classification performance on the feature shift setting. It follows the strategy in **Section 4.4.1.1** to allocate synthesised digits to clients and

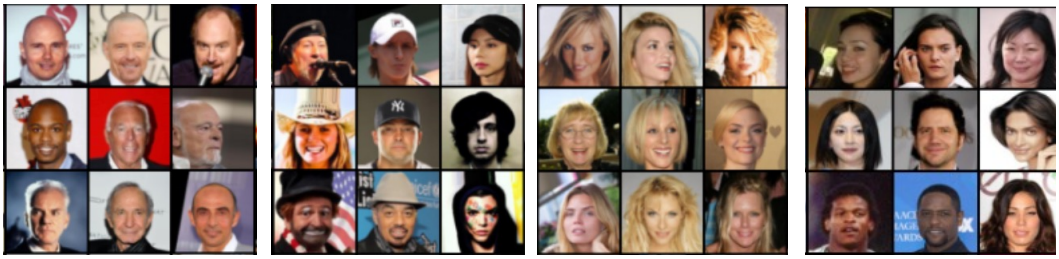
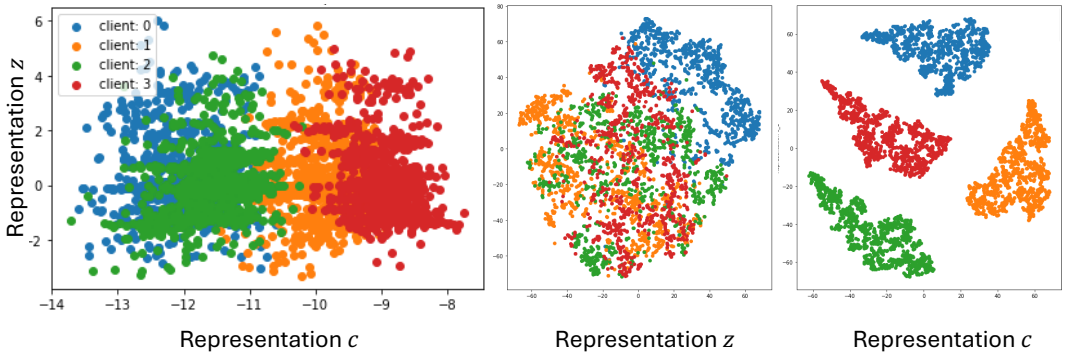


Figure 4.8: Examples of allocated face images. Each quadrant displays samples from a client, and each client is related to one type of hairstyle, i.e., bald, wearing hats, blond and black hair



(a) distribution of 1-dimension z and c (b) t-SNE scatterplots of 8-dimension z and c

Figure 4.9: Representation distributions of CelebA. Each dot denotes the representation of a sample, and colours correspond to clients. (a) scatter plot of 1-dimension z (vertical) and c (horizontal); (b) t-SNE embeddings of 8-dimension z (left) and c (right).

then trains a local lightweight classification head over the disentangled representations on each client.

Figure 4.11 demonstrates the classification accuracy on each client. Results show that a lightweight classification head based on the disentangled representations will converge fast and achieve competitive performance compared to vanilla FL methods.

4.4.2.2 Target Shift Settings

This section evaluates FedDVA's classification performance on the target shift setting. It performs two trials of experiments with different benchmark datasets, i.e., MNIST

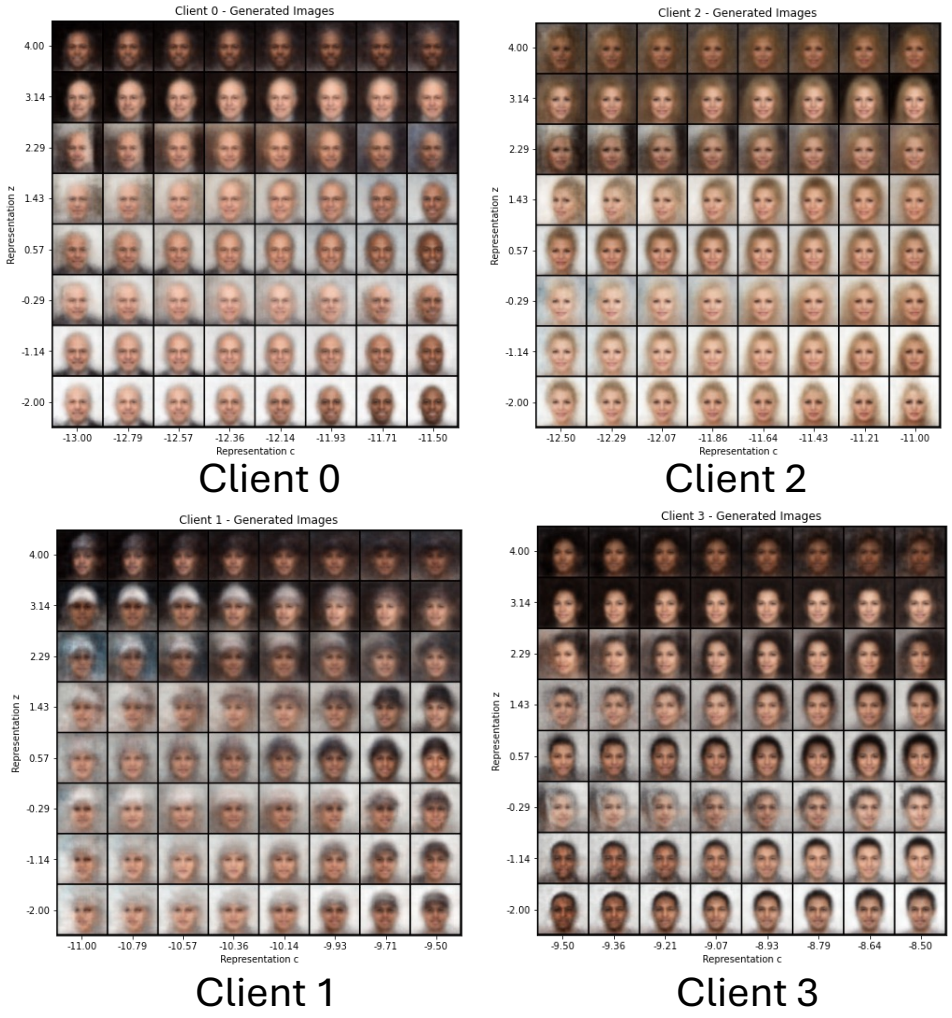


Figure 4.10: Manifolds of decoded data on different clients. General properties like identities and backgrounds vary along with changes in client-agnostic representation z (vertical), and personalised properties like hairstyles and angles vary along with changes in client-specific representation c (horizontal).

and CIFAR-10[3]. Samples from a benchmark dataset are first allocated to 20 clients in a non-I.I.D manner [42] that class distributions vary from client to client. Then, a local lightweight classification head is trained over the disentangled representations on each client.

Figure 4.12 displays the the class distributions on each client and **Figure 4.13** shows the averaged accuracy among clients. Results show that a lightweight classification head

[3]https://www.cs.toronto.edu/ kriz/cifar.html

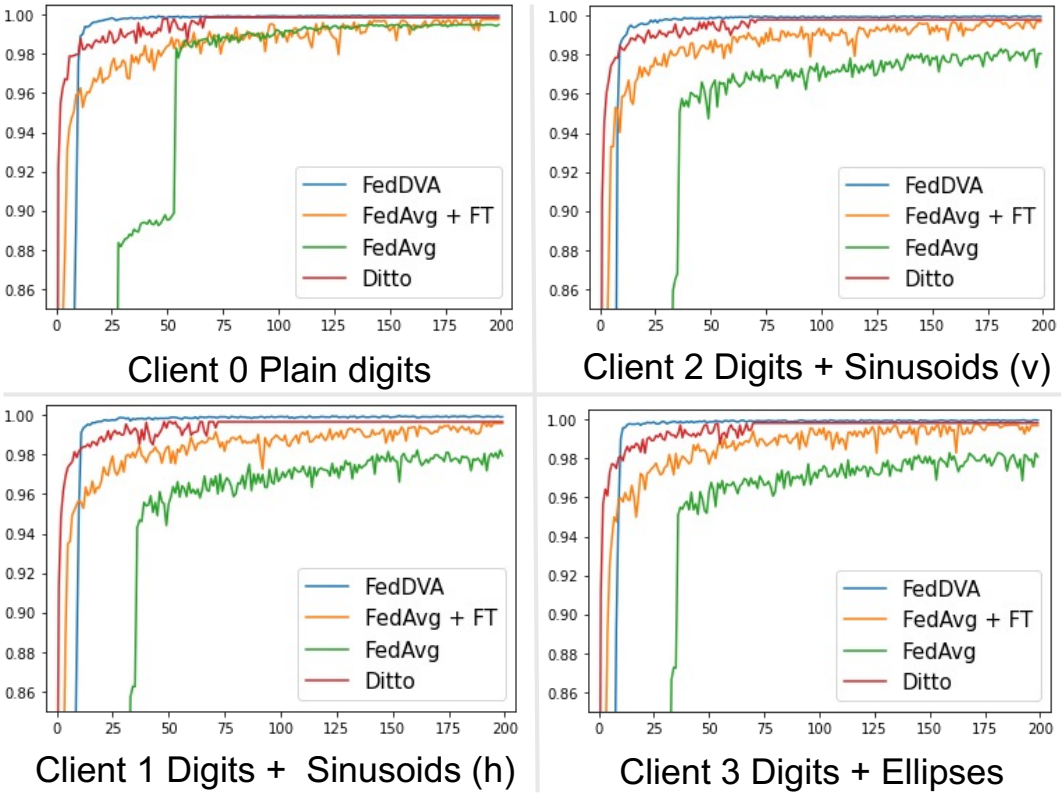
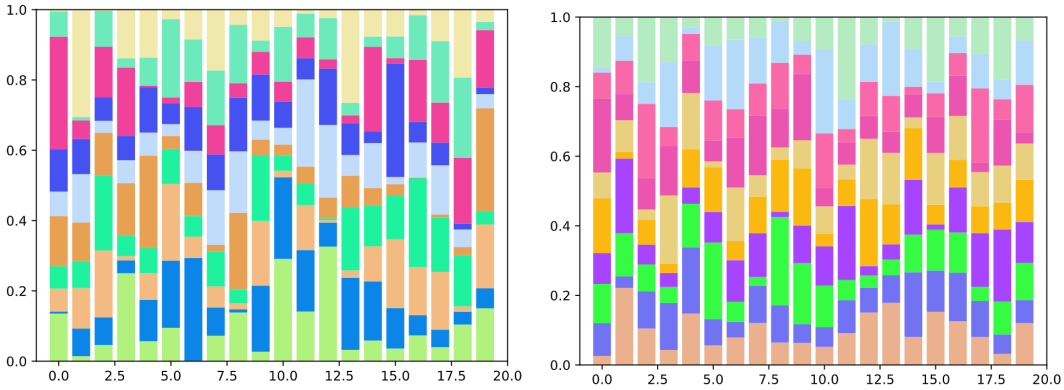


Figure 4.11: Classification accuracy on the feature shift setting. Each quadrant displays accuracy on a client. The horizontal axis denotes communication rounds, and the vertical axis denotes classification accuracy.

based on the disentangled representations will converge fast and achieve competitive performance compared to those vanilla FL methods. Besides, classification accuracy based on FedDVA has smaller standard deviations, which means that FedDVA can lead to more robust personalisation performance in the setting.

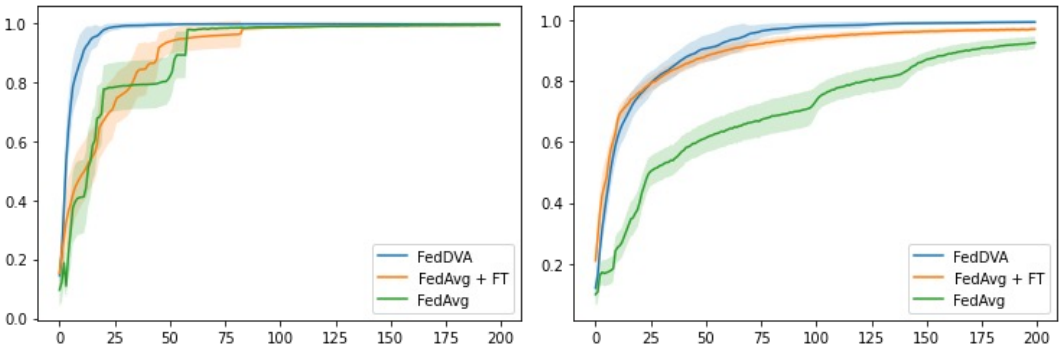
4.5 Conclusions

In conclusion, the novel FedDVA method can disentangle client-agnostic and client-specific knowledge for PerFL tasks. Empirical studies validate FedDVA's personalisation disentanglement capability. Experiments also show that lightweight classification heads trained on disentangled representations will have better convergence and competitive



(a) MNIST (b) CIFAR-10

Figure 4.12: Non-I.I.D class distributions. Each bar denotes the class distribution on a client, and the length of a colour corresponds to the portion of a class on the client.



(a) Accuracy on MNIST (b) Accuracy on CIFAR-10

Figure 4.13: Averaged classification accuracy on the target shift setting. The horizontal axis denotes communication rounds, and the vertical axis denotes accuracy. Lines are the averaged classification accuracy among clients, and shades denote the corresponding standard deviation.

accuracy and be more robust to the changing distributions.

CHAPTER 5

CLIENT-DECORRELATION FEDERATED LEARNING

5.1 Motivation

The distributed learning environment of PerFL usually leads to diverse personalised models. Then, an interpretation of a local model may be improper to the model of another client. This section studies aligning a global model's hidden layers to find a unified representation space describing client properties. The unified representation space ensures that clients will share a consensus about client properties and leads to a novel one-model-for-all strategy to embody personalisation in federated settings.

The key motivation is that model personalisation in most PerFL methods relies on the bias of samples on the same client, and little supervised information describing the client is introduced. Then, the global model will be able to encode personalised information in a unified representation space if it can recognise the bias of a client.

Based on the thought above, this research proposes a novel Client-Decorrelation Federated Learning (FedCD) [115] that is to learn a unified global model with the

below functions, including 1) to learn a unified representation space that can encode the bias of a client, 2) to share client-agnostic knowledge as vanilla FL methods, and 3) to make personalised predictions by leveraging both pieces of information. Moreover, the Representation Alignment (RA) mechanism in FedCD could become a plug-in component that can be integrated with any federated learning method. It enables a vanilla FL model to output personalised results without on-device fine-tuning steps. An illustration of the FedCD is in **Figure 5.1**.

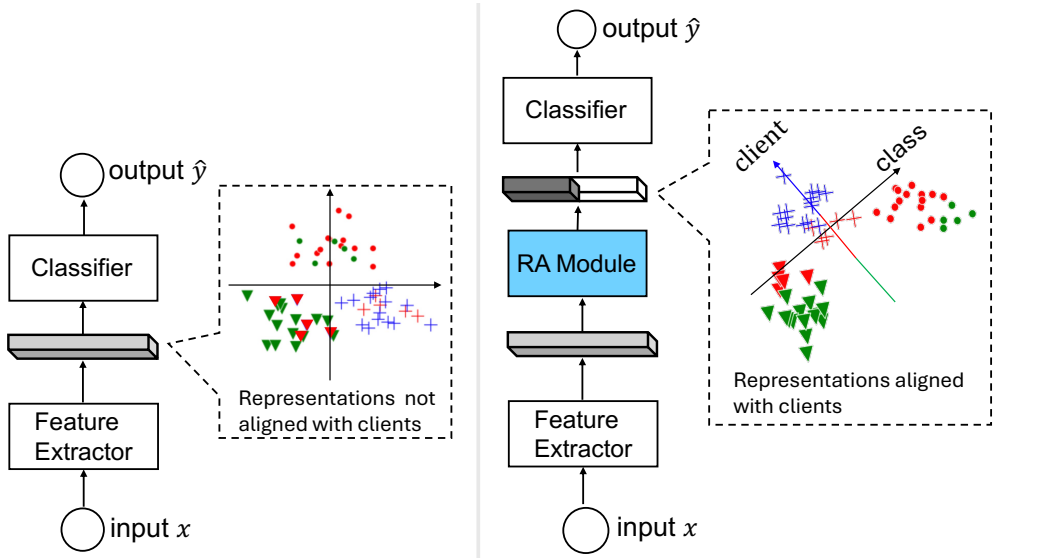


Figure 5.1: Illustration of the FedCD. Shapes denote representation distributions in the latent space. Triangles, dots and crosses denote the classes they belong to. Colours denote the clients they are on. (a) Latent representations in the conventional global model are aligned with classes but not with clients. (b) Latent representations will be aligned with clients through the RA Module.

The FedCD imposes orthogonality constraints on a global model's hidden layers, restricting their outputs to vary along axes regarding client-specific or client-agnostic knowledge. Accordingly, the model's latent representation space will be decomposed into two subspaces. One subspace is aligned with the inductive bias that sample representations will be similar if they were from the same client. The other subspace is unrelated

to clients and captures classification information. Client properties information will be precipitated in the model's hidden layers when it is optimised for supervised tasks like classification, and the representation space will be synchronised among clients along with the sharing of the global model.

To find the target representation space, FedCD formulates the representation alignment problem into a novel client-supervised optimisation framework that clients can solve collaboratively along with the model training process. Empirical studies show that the FedCD can learn a unified representation space for client properties and a robust FL global model for one-model-for-all personalisation. The FedCD's global model can be directly deployed to the test clients with changing data distributions while achieving comparable performance to other personalised FL methods that require local model adaptation.

Through qualitative and quantitative experiments, the research illustrates how FedCD is integrated into a black-box model to achieve compared performance of other personalised federated learning methods. It verifies client-specific representation's consistency with client properties contributing to personalisation and illuminates how they cooperate with FL models to induce personalised outputs. Moreover, the research demonstrates that FedCD is compatible with most DNN models. Vanilla models with an RA module can achieve competitive performance compared to those ad hoc PerFL models without needing extra fine-tuning steps or personal parameters.

The main contributions are summarised as follows:

- The research proposes a novel one-model-for-all personalised FL framework that won't require an extra fine-tuning process at the model deployment and test time stage. The personalisation of the FL system is carried by representations indicating client bias rather than models.

- The research designs a novel representation alignment mechanism to project sam-

ples' representation into a space indicating client properties. The following decision layers in the neural architecture can automatically learn to make personalised predictions by leveraging the client properties.

- A client-supervised optimisation framework is designed to fit the proposed framework. It formulates the representation-aligning problem into a unified optimisation framework that clients can solve collaboratively under FL settings.
- Comparison with baseline methods shows that, by integrating FedCD into vanilla FL models, they can achieve competitive personalisation performance without requiring extra fine-tuning steps or personal parameters.

5.2 Methodology

5.2.1 Problem Formulation

Looking inside the latent space of a DNN $f(x;\omega) = g(h(x;\omega_h);\omega_g)$, it consists of two parts: $h(x;\omega_h)$ is a feature extractor learning the latent representation $z \in \mathbb{R}^d$, and $g(z;\omega_g)$ is a classification head making predictions based on z. ω, ω_h, ω_g denote learnable parameters of the corresponding parts. FedCD aims to find orthogonal directions aligned with client-specific/-agnostic knowledge to decompose z into Client-specific representations (CRep) and Universal Representations (URep), such that 1) the global classification head can make personalised predictions; 2) the CRep is a measurement of client properties through which representations of samples influenced by similar properties will have similar values and vary significantly otherwise.

Formally, let $P_{d\times r} = [p_1, p_2, ..., p_r]$ be the orthonormal basis for capturing client-specific knowledge and $Q_{d\times t} = [q_1, q_2, ..., q_t]$ be the one for client-agnostic knowledge. Then projections $c = P^T z$ and $s = Q^T z$ will respectively be representations describing client-specific/-agnostic knowledge, namely the CRep and the URep.

FedCD searches for the optimal directions P^* according to an inductive bias that samples on the same client are influenced by identical client properties. It implies that CReps on the same client shall be similar, and those from different clients are on the contrary.

5.2.2 Representation Alignment

Let $\bar{c}^{(k)}$ denote the mean of CReps on the k-th client, and $\bar{c}$ be the global mean of CReps among clients.

$$\Sigma_W = \frac{1}{\sum_{k=1}^{K} N_k} \sum_{k=1}^{K} \sum_{i=1}^{N_k} (c_i^{(k)} - \bar{c}^{(k)})(c_i^{(k)} - \bar{c}^{(k)})^T \tag{5.1}$$

Equation 5.1 is the within-client scatter matrix that measures the scatter of CReps within each client, and

$$\Sigma_B = \frac{1}{\sum_{k=1}^{K} N_k} \sum_{k=1}^{K} N_k (\bar{c}^{(k)} - \bar{c})(\bar{c}^{(k)} - \bar{c})^T \tag{5.2}$$

Equation 5.2 is the between-client scatter matrix that measures the scatters of the mean of CReps across clients. To find the P^* aligning client-specific knowledge is to find the directions that minimise Σ_W and maximise Σ_B. For example, it can be formulated as the Linear Discriminate Analysis (LDA) problem below

$$P^* = \arg\max_P J(P) = \arg\max_P \mathrm{Tr}(\Sigma_W^{-1} \Sigma_B) \tag{5.3}$$

where $\mathrm{Tr}(\cdot)$ denotes the trace of the matrix.

Meanwhile, FedCD searches for the directions of Q, which spans a space uncorrelated with the one by the P. It requires that $[P,Q]^T[P,Q] = I_{(d+r)}$, where $I_{(d+r)}$ is the identity matrix. Then, learning the optimal Q involves solving an optimisation problem with quadratic constraints, usually NP-hard. On this account, FedCD directly derives Q^* by the Gram-Schmidt process (GSP) [94] as described in **Algorithm 5**. It removes client

Algorithm 5 Gram-Schmidt process (GSP) derives Q^*

Input: P^*

Output: Q^*

1: Initialise $Q = I_d$.
2: Let $proj_p(q) = p^T q \frac{p}{\|p\|}$
3: **for** $u = 1$ **to** d **do**
4: $\quad q_u = q_u - \sum_{v=1}^{r} proj_{p_v}(q_u)$
5: **end for**
6: **return:** Q

influences on samples' representations and guarantees that $[P,Q]$ will span the same d-dimensional representation space as the original one in the model's hidden layer.

Then, bringing P and Q into the FL framework, the overall learning task of FedCD is formulated as a bi-level optimisation problem

$$\begin{aligned} \omega_h^*, \omega_g^* &= \arg\min_{\omega_h,\omega_g} \sum_{k=1}^{K} \alpha_k \mathscr{L}_k(\omega_h, \omega_g) \\ s.t.\ P^* &= \arg\max_{P} J(P) \\ Q^* &= GSP(P^*) \end{aligned} \tag{5.4}$$

where

$$\mathscr{L}_k = \sum_{i=1}^{N_k} l(g([P^*, Q^*]^T h(x_i^{(k)}; \omega_h); \omega_g), y_i^{(k)}) \tag{5.5}$$

The following section introduces a client-supervised method to optimise the **Equation 5.4** under the FL setting.

5.2.3 Client-Supervised Optimisation

Theoretically, the optima of **Equation 5.3** are eigenvectors of $\Sigma_W^{-1}\Sigma_B$ associated with the r largest eigenvalues [27]. The classification and the alignment tasks in **Equation 5.4** can be optimised alternatively under the conventional FL framework [55, 82]. However, decomposing $\Sigma_W^{-1}\Sigma_B$ is computationally expensive, and involves collecting the local mean

$\bar{c}^{(k)}$ which is privacy sensitive. To this end, this research proposes a client-supervised method that decomposes the learning task in **Equation 5.3** into sub-tasks so that clients can optimise it collaboratively.

Concretely, previous works [31] show that solving **Equation 5.3** is equivalent to solving $P^* = \arg\max_P \Psi\Phi$, where Φ is an approximation to the eigen system of the global correlation matrix $\Sigma_W + \Sigma_B$, and $\Psi = \Sigma_W^{-1/2}$. Both Ψ and Φ can be optimised incrementally through the following equations [8]

$$\Psi' = \Psi + \eta(I - \Psi\Sigma_W\Psi) \tag{5.6}$$

$$\Phi' = \Phi + \lambda(\bar{u}\bar{u}^T\Phi - \Phi\tau(\Phi\bar{u}\bar{u}^T\Phi) \tag{5.7}$$

where $u = \Psi\bar{c}$ and $\tau(\cdot)$ is an operator that sets all the elements below the main diagonal of the matrix to zero. λ and η are learning rates.

Then, **Equation 5.6** and **Equation 5.7** can be reorganised as the below equations and optimised by clients collaboratively. The learning process on a client is described in **Algorithm 6**, and the overall FL process with FedCD is described in **Algorithm 7**.

$$\Psi' = \Psi + \eta\frac{1}{K}\sum_{k=1}^{K}(I - \Psi\Sigma_W^{(k)}\Psi) \tag{5.8}$$

$$\Phi' = \Phi + \lambda\sum_{k=1}^{K}(\bar{u}^{(k)}\bar{u}^{(k)T}\Phi - \Phi\tau(\Phi\bar{u}^{(k)}\bar{u}^{(k)T}\Phi) \tag{5.9}$$

where $\Sigma_W^{(k)} = \sum_{i=1}^{N_k}(c_i^{(k)} - \bar{c}^{(k)})(c_i^{(k)} - \bar{c}^{(k)})^T$ and $\bar{u}^{(k)} = \Psi(\bar{c} + \frac{1}{N_k}\sum_{i=1}^{N_k}(\bar{c} - \bar{c}^{(k)}))$.

5.3 Experiments

This section demonstrates the advantages of FedCD in learning from clients with non-I.I.D. data. The FedCD can learn a robust FL global model for the changing data distributions of unseen/test clients. The FedCD's global model can be directly deployed to the

Algorithm 6 Client-supervised Optimisation

Input: a batch of latent representations $z_i^{(k)}, i = 1 : N_k$, global mean $\bar{z}$, Ψ and Φ
Output: $\bar{z}'$, Ψ', Φ'

$P \leftarrow \Psi\Phi$.
$\bar{z}' \leftarrow \bar{z} + \frac{1}{N_k}\sum_{i=1}^{N_k}(\bar{z}_i^{(k)} - \bar{z})$
$c_i^{(k)} = P z_i^{(k)}$, $i = 1, ..., N_k$
$\bar{c} \leftarrow P\bar{z}'$
$\bar{c}^{(k)} \leftarrow \frac{1}{N_k}\sum_{i=1}^{N_k} c_i^{(k)}$
$\bar{u}^{(k)} = \Psi\bar{c}$
$\Sigma_W^{(k)} \leftarrow \sum_{i=1}^{N_k}(c_i^{(k)} - \bar{c}^{(k)})(c_i^{(k)} - \bar{c}^{(k)})^T$
update Ψ by **Equation 5.8**
update Φ by **Equation 5.9**
return $\bar{z}'$, Ψ',Φ'

Algorithm 7 Federate Learning with Client-Decorrelation

Input: number of clients: K, interval to align representations: it, number of communication rounds: R.
Output: P^* Q^*, ω_h^*, ω_g^*

initialise ω_h, ω_g, $\bar{z}$, Ψ, Φ.
for r from 0 to R **do**
    select a set of clients $\mathbb{C}$
    for client k in $\mathbb{C}$ **parallel do**
        update ω_h,ω_g by gradient-descent steps
        if $r\%it$ ==0 **then** ▷ Representation Alignment
            update $\bar{z}$, Ψ, Φ by **Algorithm 6** ▷ Client-supervised Optimisation
            update Q by **Algorithm 5** ▷ GS Process
        end if
    end for
    aggregate local updates of ω_h, ω_g, $\bar{z}$, Ψ and Φ through averaging parameters
end for
$P^* \leftarrow \Psi\Phi$
update Q^* by **Algorithm 5**
return P^*, Q^*, ω_g^*, ω_h^*

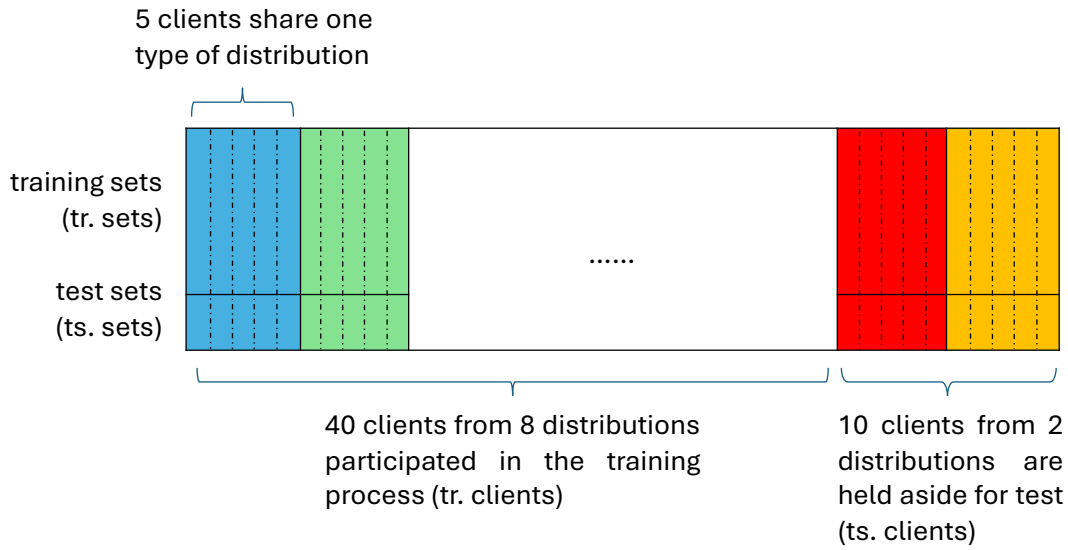


Figure 5.2: Clients in the target shift setting. Each bar denotes a client. Each colour indicates one type of distribution. Samples on each client are split into a training set and a test set.

test clients while achieving comparable performance to other personalised FL methods that require model adaptation. Visualisation of the aligned Client-specific representation validates the effectiveness of the proposed RA Module.

5.3.1 Personalisation Settings

The research simulates FL environments by allocating samples from benchmark datasets to 50 clients, and two different types of personalisation settings are applied.

Target Shift Settings: MNIST and CIFAR-10 datasets are applied as benchmark datasets to simulate the non-I.I.D. environments. The experiment allocates samples of each class individually according to a posterior of the Dirichlet distribution[42], which divides clients into ten groups with different class distributions. Eight groups of clients will participate in the collaborative training process, and the rest will be held for testing. An illustration of client settings is in **Figure 5.2**. Class distributions are shown in **Figure 5.3**.

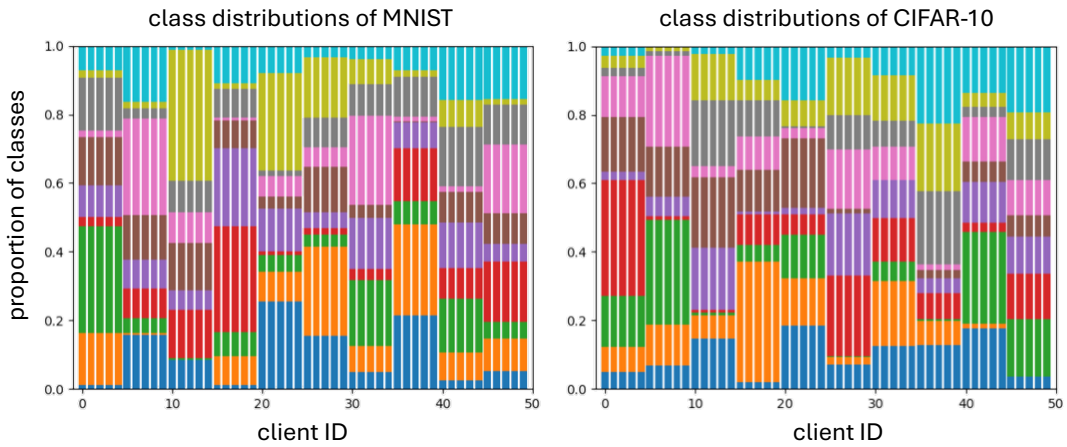


Figure 5.3: Class distributions on clients. Each bar denotes the class distribution on a client. Each colour corresponds to a class and the length indicates its proportion on the client.

Feature Shift Settings: The research utilises the Digit-5 dataset to evaluate FedCD's performance on feature-shift data. The Digit-5 consists of digits from five different domains (MNIST, MNIST-M, SVHN, USPS and Synth Digits). The experiment assigns samples of each domain to nine clients, where eight clients will participate in training the global model and one will be held aside for the test. In addition, it randomly draws samples from all domains to compose five mixed datasets for the rest of the clients for the test. An illustration of client settings is in **Figure 5.4**.

5.3.2 Models and Hyperparameters

The research applies convolution neural networks (CNN) as fundamental models and integrates the proposed RA module into one of the fully connected layers (FC1 to FC3) to align their hidden layers. By default, in each communication round, ten clients are sampled to update the global model, and subsequently, the global model is synchronised to all clients to evaluate its performance. The learning rate of a client's local training step is initialised as 0.005 and it will decay at the rate of 0.8 every 50 communication rounds. The RA module will be updated every five communication rounds by the sampled

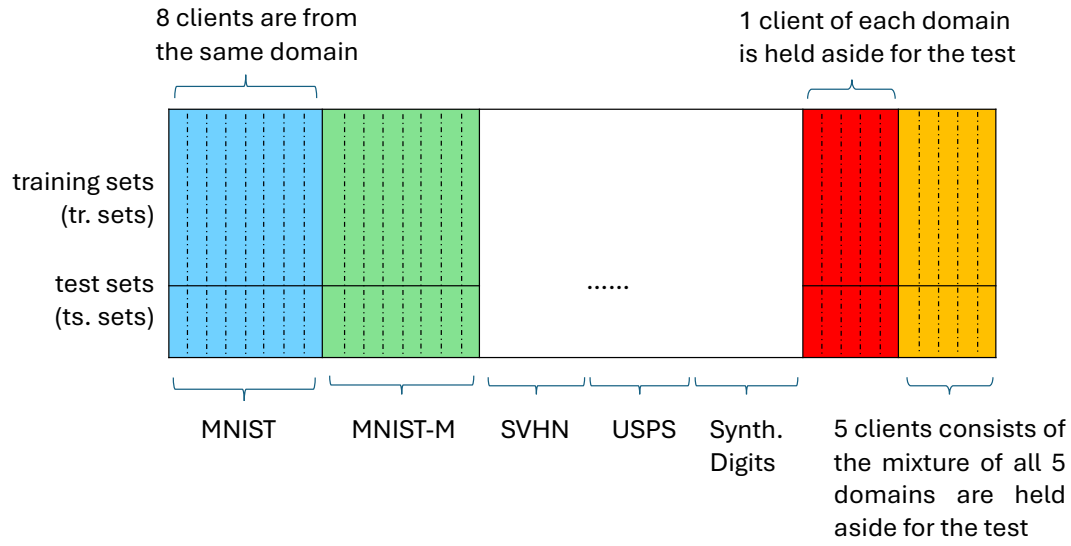


Figure 5.4: Clients in the feature shift setting. Each bar denotes a client. Each colour indicates a domain. Samples on each client are split into a training set and a test set.

	MNIST	CIFAR-10	Digit-5
communication rounds	200	400	400
epoch per round	5	30	10
batch size	10	32	10

Table 5.1: Hyperparamters for experiments

ten clients, and the learning rate is fixed at 0.001. Other hyperparameters are listed in Table.5.1.

5.3.3 Baseline Methods

Several PerFL strategies are compared as baselines, including:

- **Local Only**: models those trained on each client locally
- **FedAvg**: benchmark FL framework [75]
- **FedAvg + FT**: personalisation by fine-tuning the global model on local data [18, 21]
- **FedAvg + BN**: a global model with shared BatchNormalisation layers

- **FedBN**: a global model with private BatchNormalisation layers [57]
- **FedRep**: personalisation by training local classification heads [20]
- **PerCNN**: personalisation by training local feature extractors [82]

5.3.4 Performance

This section first demonstrates averaged model performance on all clients, which shows that a global model learned with FedCD will achieve comparable performance to other personalised FL methods that require model adaptation. Then, it looks inside the group-wised metrics to evaluate a model's performance on different distributions. Results show that the global model learned with FedCD is more robust to the changing distributions. The learned global model can be directly deployed on test clients without extra adaptations.

5.3.4.1 Target Shift Settings

For target shift settings, the averaged accuracy, weighted AUC score and weighted F1 score are applied to evaluate model performance[1]. **Table 5.2** and **Table 5.3** report the averaged performance over all clients and **Figure 5.5** and **Figure 5.6** show the group-wise performance.

Overall performance

Table 5.2 demonstrates models' performance on the MNIST dataset. It can be found that a global model with RA layers achieves the best performance under the target shift setting. It outperforms those locally fine-tuned global models (FedAvg+FT) and models with client-specific parameters (FedBN, FedRep and PerCNN). A combination of FedAvg and BN layers has a performance marginally below FedCD; the Locally trained model

[1] https://scikit-learn.org/stable/index.html

	avg. Accuracy (%) ↑	w. AUC (%) ↑	w. F1-score (%) ↑
Local Only	62.19 (7.93)	85.18 (4.62)	47.80 (10.12)
FedAvg	88.26 (4.81)	97.12 (1.33)	81.01 (5.48)
FedAvg+FT	87.72 (9.06)	96.97 (2.54)	80.81 (10.14)
FedAvg+BN	97.97 (1.19)	99.69 (0.14)	93.38 (1.76)
FedBN	88.74 (18.56)	99.32 (0.75)	83.85 (18.82)
FedRep	51.97 (21.06)	75.72 (15.56)	38.53 (20.52)
PerCNN	95.96 (1.70)	99.48 (0.20)	91.40 (1.95)
FedCD-FC1(ours)	**98.43 (0.80)**	**99.72 (0.15)**	**93.72 (1.71)**
FedCD-FC2(ours)	98.26 (0.90)	99.71 (0.16)	93.43 (1.78)
FedCD-FC3(ours)	98.35 (0.93)	99.72 (0.26)	93.64 (1.56)

Table 5.2: Overall performance on the MNIST dataset. The standard deviation of each metric is reported in parentheses. avg. is the abbreviation of 'averaged' and w. denotes the 'weighted'. The ↑ denotes that the higher the metric is, the better performance a model achieved, and the best performance is highlighted in bold.

(Local Only) has the worst performance. It might result from the lack of training data and the extremely unbalanced distribution of classes.

Table 5.3 demonstrates models' performance on the CIFAR-10 dataset. FedCD achieves the best performance in this setting. Other models are less effective than FedCD and their performances vary significantly among clients (higher standard deviations). The group-wised performance below shows that the gap results from the generalisation error on test clients and FedCD mitigates such performance gap.

Group-wise performance

Figure.5.5 and **Figure.5.6** show the averaged accuracy of clients within different groups, i.e., data distributions. It shows that the global model trained by FedCD is more robust among different distributions, and it generalises well to unseen distributions (client groups 8-9). Fine-tuned models (FedAvg+FT) and models with personalised parameters (FedBN, FedRep) have significant performance gaps between training clients (group 0-7) and test clients (group 8-9). They achieve higher accuracy in training groups but are less effective in test groups.

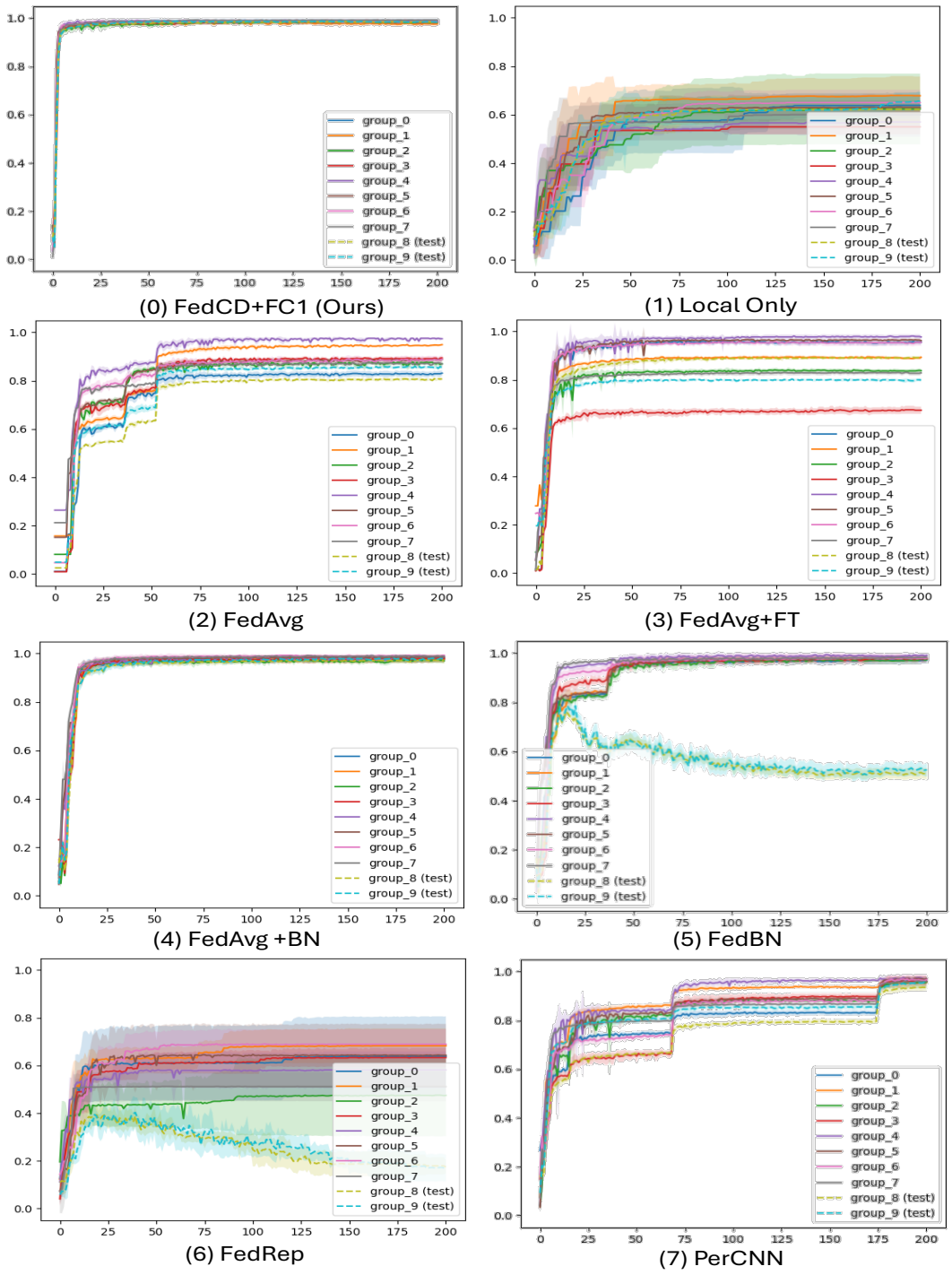


Figure 5.5: Grouped-wise accuracy on MNIST. The horizontal axis denotes communication rounds and the vertical axis denotes the accuracy. Each colour corresponds to a client group, i.e., data distribution. Shade indicates the standard deviation of accuracy among clients in the group.

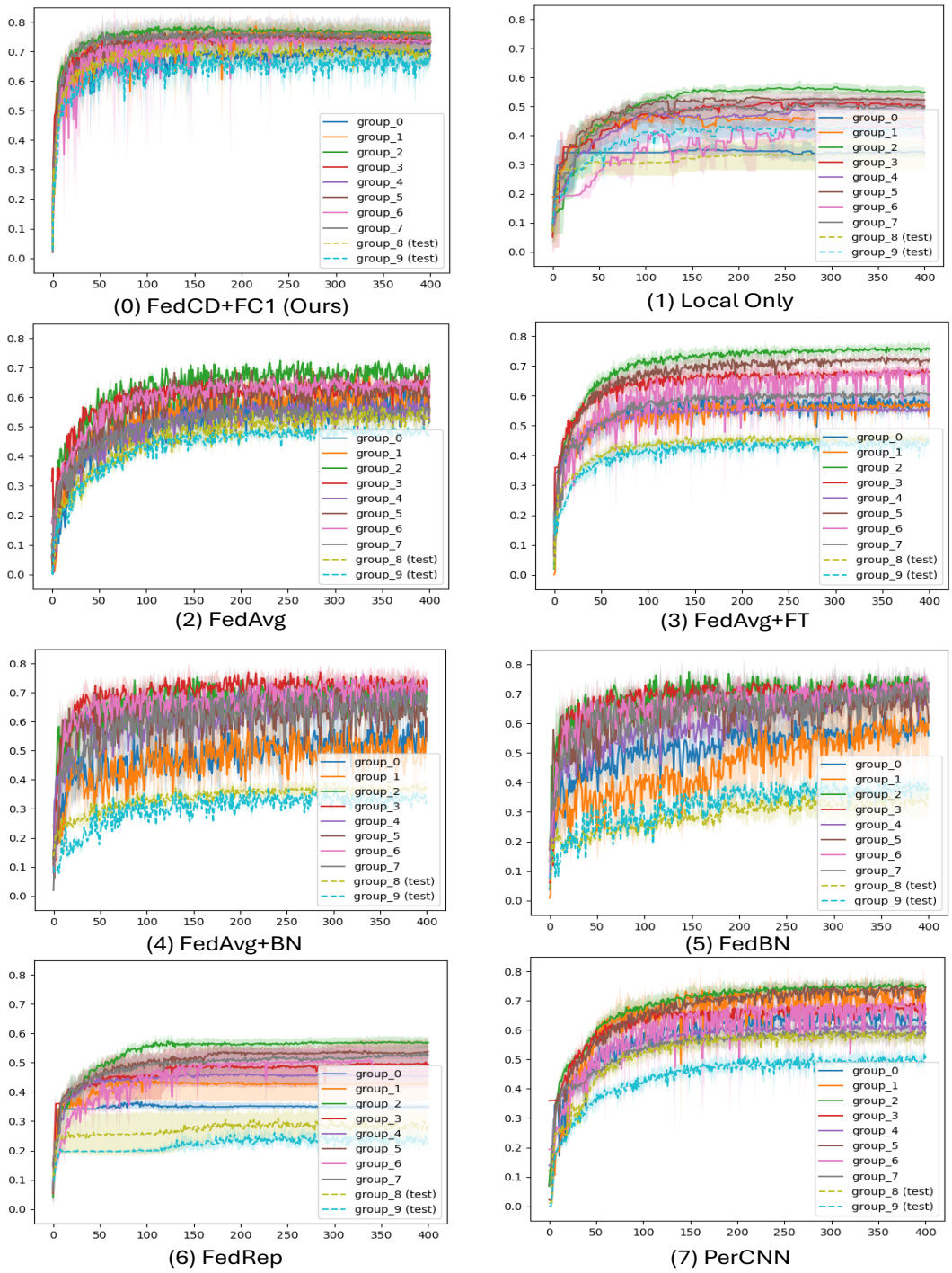


Figure 5.6: Grouped-wise accuracy on CIFAR-10. The horizontal axis denotes communication rounds and the vertical axis denotes the accuracy. Each colour corresponds to a client group, i.e., data distribution. Shade indicates the standard deviation of accuracy among clients in the group.

	avg. Accuracy (%) ↑	w. AUC (%) ↑	w. F1-score (%) ↑
Local Only	45.09 (8.14)	70.93 (9.23)	34.86 (10.71)
FedAvg	57.92 (7.52)	88.03 (3.61)	54.04 (8.61)
FedAvg+FT	60.37 (10.12)	86.89 (4.65)	51.75 (12.62)
FedAvg+BN	58.84 (13.94)	91.57 (4.30)	57.87 (14.95)
FedBN	60.65 (14.21)	91.64 (4.56)	59.41 (14.70)
FedRep	43.34 (11.58)	70.70 (9.80)	33.01 (13.88)
PerCNN	64.69 (8.41)	89.18 (3.36)	58.62 (9.17)
FedCD-FC1(ours)	68.07 (4.33)	**93.72 (1.71)**	69.48 (4.69)
FedCD-FC2(ours)	68.12 (4.87)	93.33 (2.33)	69.06 (5.97)
FedCD-FC3(ours)	**68.84 (4.62)**	93.49 (2.09)	**69.69 (4.96)**

Table 5.3: Overall performance on the CIFAR-10 dataset. The standard deviation of each metric is reported in parentheses. avg. is the abbreviation of 'averaged' and w. denotes the 'weighted'. The ↑ denotes that the higher the metric is, the better performance a model achieved, and the best performance is highlighted in bold.

5.3.4.2 Feature Shift Settings

This section demonstrates evaluations in feature shift data. **Table 5.4** shows that FedCD achieves the best accuracy, AUC and F1 score under this setting. Other models are less effective than FedCD and their performances vary significantly among clients (higher standard deviations). Group-wised performance in **Figure 5.7** shows that FedCD has a more robust performance on all domains while other methods degenerate significantly on test clients (dash lines).

5.3.5 Visualisation of Aligned Representations

Figure 5.8 compares the client-specific representations learned by FedCD with latent representations from FedAvg and FedBN. It can be found that representations from FedAvg and FedBN are unable to maintain the group information (clients' preferences), while the CReps demonstrate the same cluster structure as the one regarding clients' preferences, which validates the effectiveness of the proposed RA mechanism.

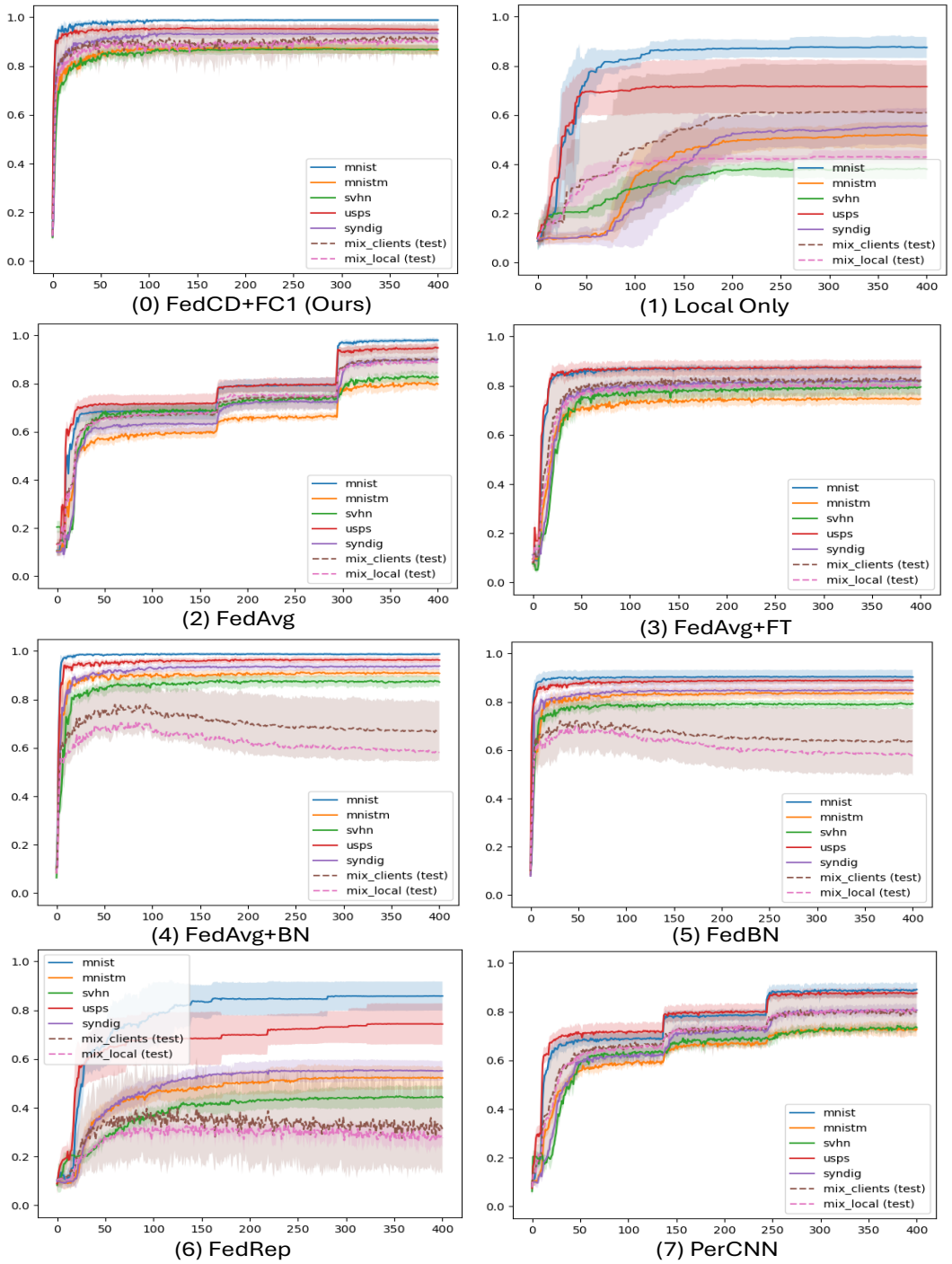


Figure 5.7: Grouped-wise accuracy on Digit-5. The horizontal axis denotes communication rounds and the vertical axis denotes the accuracy. Each colour corresponds to a client group, i.e., data distribution. Shade indicates the standard deviation of accuracy among clients in the group.

	avg. Accuracy (%) ↑	w. AUC (%) ↑	w. F1-score (%)↑
Local Only	59.11 (18.35)	84.03 (9.75)	49.73 (20.38)
FedAvg	89.15 (6.73)	98.16 (1.43)	85.69 (6.38)
FedAvg+FT	82.12 (5.32)	96.11 (1.98)	75.64 (4.94)
FedAvg+BN	87.19 (13.63)	97.82 (3.11)	83.40 (12.96)
FedBN	80.38 (11.63)	95.71 (3.79)	74.07 (10.90)
FedRep	55.77 (20.51)	82.94 (11.54)	50.14 (20.88)
PerCNN	80.63 (6.91)	96.10 (2.47)	74.63 (6.71)
FedCD-FC1(ours)	**91.47 (4.99)**	**98.74 (0.95)**	**87.89 (4.76)**
FedCD-FC2(ours)	91.19 (5.52)	98.60 (1.09)	87.66 (5.15)
FedCD-FC3(ours)	91.18 (5.52)	98.57 (1.09)	87.66 (5.19)

Table 5.4: Overall performance on the Digit-5 dataset. The standard deviation of each metric is reported in parentheses. avg. is the abbreviation of 'averaged' and w. denotes the 'weighted'. The ↑ denotes that the higher the metric is, the better performance a model achieved, and the best performance is highlighted in bold.

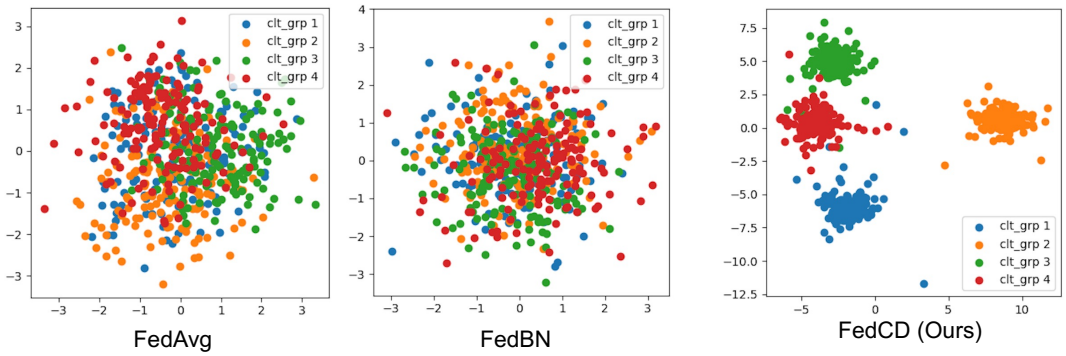


Figure 5.8: Comparisons of distributions of latent representations on the Digit-5 dataset.

5.4 Conclusions

This research is the first to propose using a one-model-for-all strategy to implement personalised federated learning. We believe the one-model-for-all personalisation can form a new topic to advance existing personalised federated learning research. It foresees more discussion and exploration can be conducted in this new one-model-for-all personalised federated setting.

From a technical perspective, the research introduces a novel client-decorrelation method, FedCD, for FL. A global model learned with FedCD can be shared across clients

without extra workload for fine-tuning in the stage of model deployment and test time. The personalised characteristic of each client has been preserved into client-specific representations that will be further processed by a unified model. Moreover, a federated optimisation framework has been designed accordingly to solve the proposed framework. Experiments on real-world datasets verified the effectiveness and efficiency of FedCD.

CHAPTER 6

VIRTUAL CONCEPTS BOOST FEDERATED LEARNING

6.1 Motivation

A critical challenge in PerFL is the absence of well-defined concepts of personalisation. Client preferences and personalised properties are implied in training data and enclosed on each client. They could be a client's favour towards specific classes or a specific noise mixed up with input features. The only tangible information is the shift in data distribution across clients.

Meanwhile, most machine learning models, e.g., DNNs, are trained in an end-to-end paradigm. They are optimised by back-propagating supervised information, e.g., classification loss, from the output layer to the input layer. Personalisation is performed indirectly when a model is tuned for tasks like classification. This learning schema is less efficient in PerFL. The on-device training tends to overfit a client's local data due to limited and unbalanced training samples. The aggregation step on the server, in turn, will neutralise personalised information when synthesising the global model, e.g., by averaging local updates.

However, it is worth noting that though there is no supervised information of significantly defined client properties, a feature distinguishing model personalisation from unsupervised tasks is that data in PerFL are explicitly partitioned. Samples from the same client will demonstrate a client-specific bias toward certain properties. Then, one may assume that there were invisible labels of clients inducing the on-device training to progress toward a client's preferences, i.e., personalisation. The client-based data partition essentially supervises PerFL's training process, so this research calls the learning paradigm Client-Supervised Learning.

Based on the thought above, this research introduces Virtual Concepts (VC) [116] to explicate client-supervised information. The VCs are representations of potential structure information extracted from training data. They can be learned independently of downstream classification tasks by a novel FedVC algorithm, which facilitates understanding client properties and boosts model personalisation.

Specifically, FedVC assumes that there is a set of vectors (virtual concepts), each describing a type of client property. A client's preferences are then represented by a combination of VCs, which will be utilised as supervised information to guide the training progress of the global model. **Figure 6.1** gives an illustration to the propose FedVC. To learn the VCs, FedVC evaluates the underlying distribution structure in data by formulating the learning task into a Gaussian Mixture Model (GMM) that can be solved by most unsupervised learning methods, e.g., Expectation-Maximisation algorithm (EM).

Experiments on real-world datasets show that the VCs can work as supervised information to train a robust global model to the changing distributions. Further study demonstrates that the VCs are useful in exploring meaningful client properties by discovering distribution structures implied in training data.

The main contributions are summarised as follows:

- The research proposes virtual concepts describing client preferences. The VCs

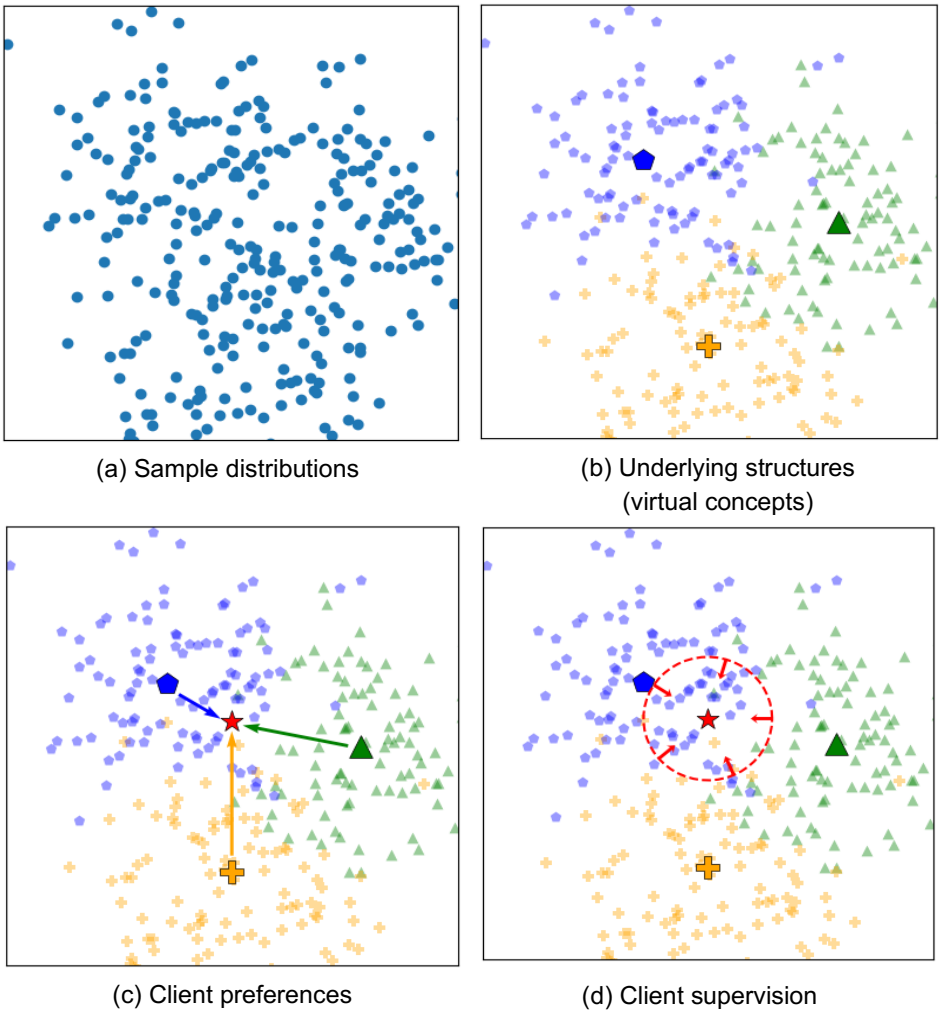


Figure 6.1: Illustration to FedVC. (a) data distribution in an FL system; (b) virtual concepts (pentagon, plus and triangle) are vectors indicating underlying cluster structures of data, e.g., cluster centres; (c) a client's preference (star) is represented by a combination of virtual concepts; (d) client-supervised loss requires sample representations on the same client (data points within the circle) to be close to each other as they share the identical client preference.

are representations of distribution structure extracted from training data. They provide us with a way to explore meaningful client properties relevant to model personalisation.

- The research proposes a novel client-supervised PerFL framework that utilises virtual concept vectors as supervised information to train the global model. The

VCs will allow an FL algorithm to simultaneously learn class and client knowledge so that the learned global model can achieve on-deployment personalisation, where the global model will not require an extra fine-tuning process at the test stage.

- The research formulates the learning task of VCs into a Gaussian Mixture Model that most unsupervised learning methods can solve. The proposed FedVC framework is compatible with most FL methods, where they can be integrated as an add-on to improve personalisation performance and model interpretability.

- Contrast with baseline methods shows that FL models trained with VCs can simultaneously learn class and client knowledge. It achieves competitive personalisation performance without requiring extra fine-tuning steps or personal parameters.

- Empirical studies show that VCs can discover meaningful distribution structures implied in training, facilitating the uncovering of client properties related to model personalisation.

6.2 Methodology

6.2.1 Client-supervised PerFL

Let $\mathscr{C} = \{c_1, ..., c_M\}$ denote m virtual concept vectors, a client's preference is then represented by $p^{(k)} = \sum_{m=1}^{M} v_m^{(k)} c_m$, where k is the client index, and v_m is a factor measuring the degree the client relevant to c_m, i.e., how typical the client has the property of c_m. FedVC aims to utilise $p^{(k)}$ as supervised information to guide FL's learning process so that the global model can learn client knowledge explicitly.

Specifically, FedVC adds a projection head to FL's global model to extract a representation $\hat{z}_i^{(k)}$ of potential client properties (see **Figure 6.2**). One can evaluate a sample's relevance to each concept by a similarity function, e.g., **Equation 6.1**, and derive an

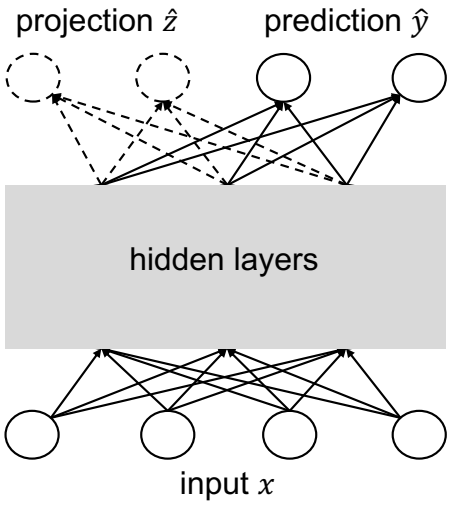


Figure 6.2: Projection head

estimated client preference $\hat{p}_i^{(k)} = \sum_{m=1}^{M} \hat{s}_{i,m}^{(k)} c_m$, where i is the sample index and ι is a hyperparameter.

$$s_{i,m}^{(k)} = \frac{v_m^{(k)} \exp(-\iota \|\hat{z}_i^{(k)} - c_m\|^2)}{\sum_{m=1}^{M} v_m^{(k)} \exp(-\iota \|\hat{z}_i^{(k)} - c_m\|^2)} \tag{6.1}$$

Then, there will be a supervised loss regarding client preferences, i.e., $l_p(\hat{p}^{(k)}, p_i^{(k)}) = \|\hat{p}^{(k)} - p_i^{(k)}\|^2$. It can be integrated into any FL framework like **Equation 3.1** and solved by gradient-based methods. Details of the learning algorithm are in **Algorithm 8**.

6.2.2 Virtual Concepts

As virtual concepts correspond to client properties, a sample is then assumed to be generated by some random process involving a mixture of multiple client properties. FedVC formulates the assumption into a Gaussian Mixture Model (GMM). For any sample $\mathbf{z}^{(k)}$ on the k-th client, there is

$$\mathbf{z}^{(k)} \sim \mathscr{P}^{(k)}(\mathbf{z}) = \sum_{m=1}^{M} v_m^{(k)} \mathcal{N}(\mathbf{z}; c_m, \Sigma_m) \tag{6.2}$$

where the covariance Σ_m is set to be the identity matrix I for simplicity.

Let $\mathscr{C} = \{c_1, ..., c_M\}$ denotes the set of VCs and $\Upsilon = \{\{v_m^{(1)}\}_{m=1}^M, ..., \{v_m^{(K)}\}_{m=1}^M\}$ denotes the set of client preferences, the collaborative learning task for $\mathscr{C}$ and Υ is formulated as

$$\mathscr{C}^*, \Upsilon^* = \arg\max_{\mathscr{C}, \Upsilon} \sum_{k=1}^{K} \sum_{i=1}^{N_k} \log \mathscr{P}^{(k)}(z_i^{(k)}) \tag{6.3}$$

FedVC solves it by the EM framework below:

- **E-step**: Given $\mathscr{C}$ and Υ, clients estimate local samples' $s_{i,m}^{(k)}$ by **Equation** 6.1
- **M-step**: Clients update $\mathscr{C}$ and Υ collaboratively by **Equation** 6.4 and **Equation** 6.5

$$v_m^{(k)} = \frac{1}{N_k} \sum_{i=1}^{N_k} s_{i,m}^{(k)} \tag{6.4}$$

$$c_m = \frac{\sum_{k=1}^{K} \sum_{i=1}^{N_k} s_{i,m}^{(k)} \hat{z}_i^{(k)}}{\sum_{k=1}^{K} \sum_{i=1}^{N_k} s_{i,m}^{(k)}} \tag{6.5}$$

However, **Equation 6.4** and **Equation 6.5** cannot be applied when working with mini-batches in FL settings. FedVC uses exponential moving averages as an alternative:

$$S_m^{\prime(k)} = S_m^{(k)} * \kappa + \sum_{i \in \mathbb{B}} s_{i,m}^{(k)} * (1 - \kappa) \tag{6.6}$$

$$C_m^{\prime(k)} = C_m^{(k)} * \kappa + \sum_{i \in \mathbb{B}} s_{i,m}^{(k)} \hat{z}_i^{(k)} * (1 - \kappa) \tag{6.7}$$

$$N_k^{\prime} = N_k * \kappa + |\mathbb{B}| * (1 - \kappa) \tag{6.8}$$

where $\mathbb{B}$ denotes a minibatch of samples, $|\mathbb{B}|$ denotes the batch size, and κ is a smoothing hyperparameter between 0 and 1. Then,

$$v_m^{(k)} = \frac{S_m^{\prime(k)}}{N_k^{\prime}} \tag{6.9}$$

$$c_m = \frac{\sum_{k=1}^{K} C_m^{\prime(k)}}{\sum_{k=1}^{K} S_m^{\prime(k)}} \tag{6.10}$$

The overall learning algorithm is described in **Algorithm 8**.

Algorithm 8 FedVC

Input: communication rounds R, epochs in each round E, learning rate λ, batch size B, hyperparameters ι, κ and γ

Output: optimal parameters ω^*, virtual concepts $\mathscr{C}^*$

server initialises parameters ω and virtual concepts $\mathscr{C}$
for r from 0 to R **do** ▷ communication rounds
  server selects a set of clients $\mathbb{C}$
  for $k \in \mathbb{C}$ **parallel do**
    client k synchronises ω and $\mathscr{C}$ from the **server** ▷ network traffic
    $\omega_k, S_m^{'(k)}, C_m^{'(k)} \leftarrow$**ClientUpdate**($\omega$)
  end for
  server collects local updates ω_k, $S_m^{'(k)}$ and $C_m^{'(k)}$ $k \in \mathbb{C}$ ▷ network traffic
  $\omega \leftarrow \sum_{k \in \mathbb{C}} \alpha_k \omega_k$
  update $c_m \in \mathscr{C}$ by **Equation 6.10**
end for
return ω, $\mathscr{C}$

ClientUpdate(ω, $\mathscr{C}$)

for any sample on the clients **do** ▷ Update client preferences$p^{(k)}$
  get model outputs by $\hat{y}, \hat{z} = f(x; \omega)$
  calculate $s_{i,m}^{(k)}$ by **Equation 6.1**
  update $v_m^{(k)}$ by **Equation 6.9**
  update client preference $p^{(k)} \leftarrow \sum_{m=1}^{M} v_m^{(k)} c_m$
end for
for e from 0 to E **do**
  for b from 0 to N_k/B **do**
    sample a batch of data $\mathbb{B}$
    $\omega \leftarrow \omega - \nabla_\omega (l_p + l_{cls})$ ▷ Update model
    update S, C and N by **Equation 6.6**, **6.7** and **6.8** respectively
  end for
end for
return ω, S and C

6.2.2.1 Unified Learning Process

It is worth noting that the client preference $p^{(k)} = \sum_{m=1}^{M} v_m^{(k)} c_m$ can be viewed a function of virtual concepts $\mathscr{C}$, so does the loss $l_p(\hat{p}, p)$. Then, the learning processes for $\mathscr{C}$ and the global ω can be formulated into a unified optimisation task that can be solved in an end-to-end manner rather than in an alternate way as EM-based methods.

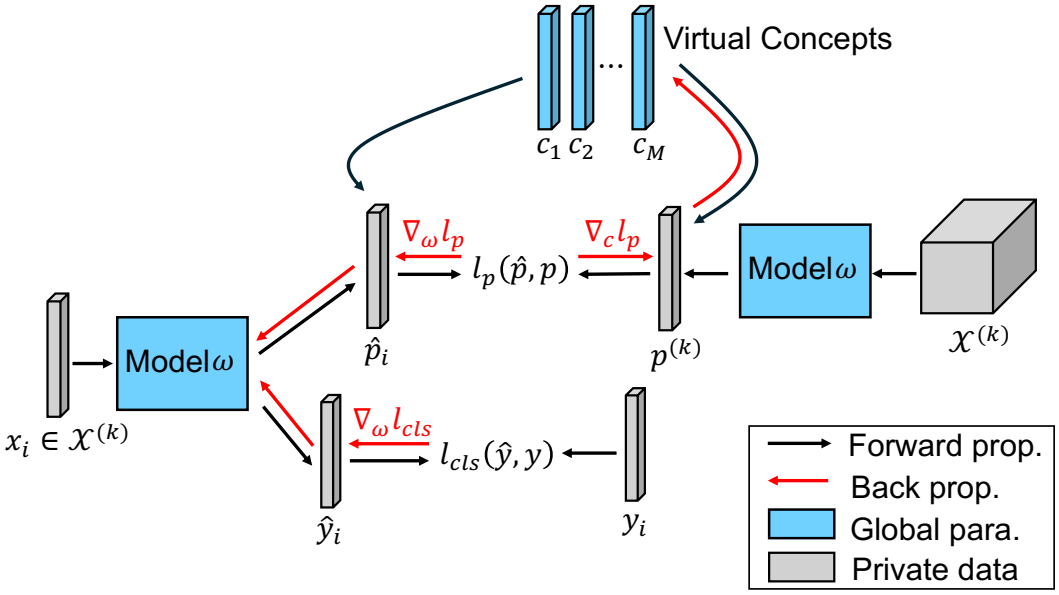


Figure 6.3: FedVC architecture.

Concretely, as described in **Figure 6.3**, $l_p(\hat{p}, p)$ will simultaneously provide supervised information for optimising virtual concepts and the model. The unified learning object is formulated as

$$\omega^*, \mathscr{C}^* = \arg\min_{\omega, \mathscr{C}} \sum_{k=1}^{K} \alpha_k \mathscr{L}_k(\omega, \mathscr{C}) \tag{6.11}$$

where

$$\mathscr{L}_k(\omega) = (1/N_k) \sum_{i=1}^{N_k} l_{cls}(\hat{y}_i^{(k)}, y_i^{(k)}) + l_p(\hat{p}_i^{(k)}, \text{sg}[p^{(k)}]) + \gamma l_p(\text{sg}[\hat{p}_i^{(k)}], p^{(k)}) \tag{6.12}$$

The sg[·] is the stopgradient operator [106], where the operand will feed forward as normal but have zero partial derivatives, being a non-updated constant. γ is a hyperpa-

rameter balancing the two losses. The corresponding learning process is summarised in **Algorithm 9**.

Algorithm 9 FedVC-unified

Input: communication rounds R, epochs in each round E, learning rate λ, batch size B, hyperparameters ι, κ and γ
Output: optimal parameters ω^*, virtual concepts $\mathscr{C}^*$

server initialises parameters ω and virtual concepts $\mathscr{C}$
for r from 0 to R **do** ▷ communication rounds
  server selects a set of clients $\mathbb{C}$
  for $k \in \mathbb{C}$ **parallel do**
    client k synchronises ω and $\mathscr{C}$ from the **server** ▷ network traffic
    $\omega_k, \mathscr{C}_k \leftarrow$ClientUpdate$(\omega, \mathscr{C})$
  end for
  server collects local updates$\omega_k, \mathscr{C}_k$ $k \in \mathbb{C}$ ▷ network traffic
  $\omega \leftarrow \sum_{k \in \mathbb{C}} \alpha_k \omega_k$
  $c_m \leftarrow \sum_{k \in \mathbb{C}} \alpha_k c_m^{(k)}$, $c_m^{(k)} \in \mathscr{C}_k$
end for
return ω, $\mathscr{C}$

ClientUpdate(ω, $\mathscr{C}$)

for any sample on the clients **do** ▷ Update client preferences$p^{(k)}$
  get model outputs by $\hat{y}, \hat{z} = f(x; \omega)$
  calculate $s_{i,m}^{(k)}$ by **Equation 6.1**
  update $v_m^{(k)}$ by **Equation 6.9**
  update client preference $p^{(k)} \leftarrow \sum_{m=1}^{M} v_m^{(k)} c_m$
end for
for e from 0 to E **do**
  for b from 0 to N_k/B **do**
    sample a batch of data $\mathbb{B}$
    $\omega \leftarrow \omega - \nabla_\omega \mathscr{L}_k$
    $c_m \leftarrow c_m - \nabla_c \mathscr{L}_k$, $c_m \in \mathscr{C}$
  end for
end for
return ω, $\mathscr{C}$

6.3 Experiments

This section empirically studies the advantages of FedVC in learning from clients with non-I.I.D. data. The FedVC can learn a robust FL global model for the changing data

distributions of unseen/test clients. The FedVC's global model can be directly deployed to the test clients while achieving comparable performance to other personalised FL methods that require model adaptation.

6.3.1 Non-I.I.D settings

Target Shift: MNIST is applied as a benchmark to simulate the non-I.I.D. environments. The experiment allocates samples of each class individually according to a posterior of the Dirichlet distribution[42], which divides clients into five groups with different class distributions. Three groups of clients will participate in the collaborative training process, and the rest will be held for testing. An illustration of client settings is in **Figure 6.4**. Class distributions are shown in **Figure 6.5**.

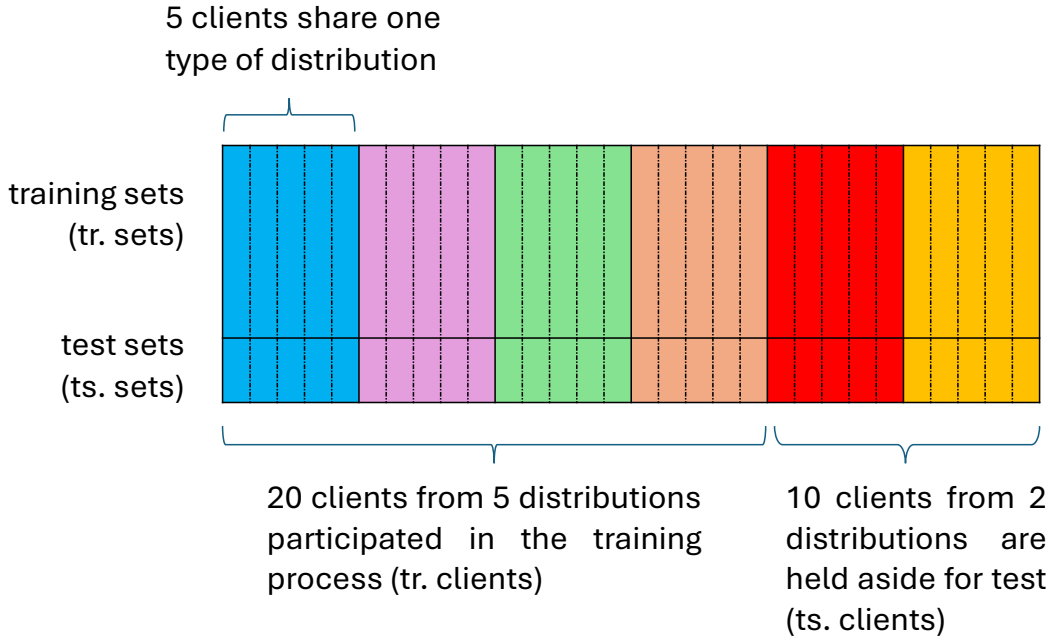


Figure 6.4: Clients in the target shift setting. Each bar denotes a client. Each colour indicates one type of distribution. Samples on each client are split into a training set and a test set.

Feature Shift: The research utilises the Digit-5 dataset to evaluate FedVC's performance on feature-shift data. The Digit-5 consists of digits from five different domains

Figure 6.5: Class distributions on clients. Each bar denotes the class distribution on a client. Each colour corresponds to a class and the length indicates its proportion on the client.

(MNIST, MNIST-M, SVHN, USPS and Synth Digits). The experiment assigns samples of each domain to six clients, where five clients will participate in training the global model and one will be held aside for the test.Classes are evenly distributed on each client. In addition, it randomly draws samples from all domains to compose five mixed datasets for the rest clients for the test. An illustration of client settings is in **Figure 6.6**.

6.3.2 Models and Hyperparameters

The research applies convolution neural networks (CNN) as fundamental models and supervises the training process by virtual concepts. By default, in each communication round, ten clients are sampled to update the global model and virtual concepts, and subsequently, the global model is synchronised to all clients to evaluate its performance. The learning rate of a client's local training step is initialised as 0.005 and it will decay at the rate of 0.8 every 10 communication rounds. During each communication round, a client will tune the global model on its local data for two epochs with a batch size of 10.

For the FedVC, the default number of virtual concepts is set to be 10 and the dimen-

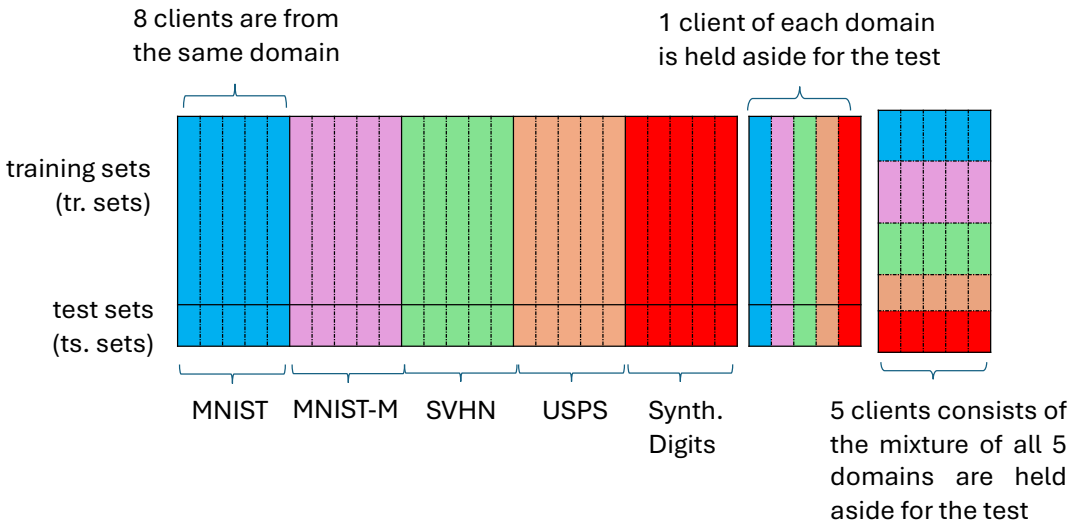


Figure 6.6: Clients in the feature shift setting. Each bar denotes a client. Each colour indicates a domain. Samples of each client are split into a training set and a test set.

sion of each virtual concept is 10. The similarity parameter ι is 0.1, and the smoothing parameter κ is 0.05.

6.3.3 Baseline Methods

Several PerFL strategies are compared as baselines, including:

- **Local Only**: models those trained on each client locally
- **FedAvg + FT**: personalisation by fine-tuning the global model on local data [18, 21]
- **FedBN**: a global model with private BatchNormalisation layers [57]
- **FedProx**: leverages a global to regularise the local training process [56]
- **Ditto**: leverages a global to regularise the local training process while learning a local model for each client [55]
- **FedRep**: personalisation by training local classification heads [20]
- **FedDual**: personalisation by training a global and a local feature extractors [82]

6.3.4 Performance

This section first demonstrates averaged model performance on all clients, which shows that a global model learned with FedVC will achieve comparable performance to other personalised FL methods that require model adaptation. Then, it looks inside the group-wised metrics to evaluate a model's performance on different distributions. Results show that the global model learned with FedVC is more robust to the changing distributions. The learned global model can be directly deployed on test clients without extra adaptations.

6.3.4.1 Target Shift Settings

For target shift settings, the averaged accuracy, weighted AUC score and weighted F1 score are applied to evaluate model performance[1]. **Table 6.1** and **Table 6.2** respectively report the averaged performance over the training clients and the test clients. **Figure 6.9** shows the group-wise performance.

Overall performance

Table 6.1 demonstrates models' performance on the MNIST dataset on the training clients (tr-clients). It can be found that a global model trained by FedVC achieves the best performance under the target shift setting. It outperforms those locally fine-tuned global models (FedAvg+FT) and models with client-specific parameters (FedBN, FedProx, FedRep and FedDual). **Table 6.2** demonstrates models' performance on the test clients (ts-clients). All baseline methods are fine-tuned on the test clients to adapt to the client's local distribution. It can be found that the model learned by FedVC generalised well to the unseen clients, even though they are not fine-tuned. Note that locally trained models (Local Only and Ditto) can not be generalised to unseen clients.

Group-wise performance

[1] https://scikit-learn.org/stable/index.html

	avg. Accuracy (%) ↑	w. AUC (%) ↑	w. F1-score (%) ↑
Local Only	95.79 (1.00)	99.69 (0.11)	93.21 (0.96)
FedAvg+FT	97.92 (0.98)	99.90 (0.04)	95.49 (1.07)
FedBN	98.43 (0.86)	99.90 (0.04)	95.71 (0.94)
FedProx	98.07 (0.90)	99.89 (0.04)	95.48 (0.89)
Ditto	95.86 (1.19)	99.71 (0.09)	93.25 (1.17)
FedRep	93.61 (2.55)	99.53 (0.18)	91.05 (2.59)
FedDual	96.87 (0.99)	99.84 (0.06)	94.14 (1.25)
FedVC (ours)	**98.56 (0.56)**	99.90 (0.05)	95.83 (0.96)
FedVC-sg (ours)	98.51 (0.62)	**99.90 (0.03)**	**95.84 (1.10)**

Table 6.1: Overall performance on the MNIST dataset on the training clients. The standard deviation of each metric is reported in parentheses. avg. is the abbreviation of 'averaged' and w. denotes the 'weighted'. The ↑ denotes that the higher the metric is, the better performance a model achieved, and the best performance is highlighted in bold.

	avg. Accuracy (%) ↑	w. AUC (%) ↑	w. F1-score (%) ↑
FedAvg+FT	98.42 (0.84)	99.91 (0.04)	95.71 (0.74)
FedBN	98.48 (0.84)	99.91 (0.04)	95.64 (0.84)
FedProx	98.19 (0.98)	99.90 (0.04)	95.53 (0.94)
FedRep	88.98 (1.37)	99.00 (0.24)	86.11 (1.85)
FedDual	97.80 (0.51)	99.88 (0.03)	95.08 (0.57)
FedVC (ours)	**98.79 (0.62)**	**99.91 (0.03)**	**95.97 (0.87)**
FedVC-sg (ours)	98.76 (0.67)	99.91 (0.04)	95.92 (0.99)

Table 6.2: Overall performance on the MNIST dataset on the test clients. The standard deviation of each metric is reported in parentheses. avg. is the abbreviation of 'averaged' and w. denotes the 'weighted'. The ↑ denotes that the higher the metric is, the better performance a model achieved, and the best performance is highlighted in bold.

Figure.6.9 shows the averaged accuracy of clients within different groups, i.e., data distributions. It shows that the global model trained by FedVC is more robust among different distributions, and it generalises well to unseen distributions (client groups 4-5). Fluctuation in the learning curves indicates that the fine-tuned models (FedAvg+FT) and models with personalised parameters (FedBN, FedProx, FedDual) are slightly unstable. Locally trained models (Local Only and Ditto) and FedRep have significant performance gaps among clients.

	avg. Accuracy (%) ↑	w. AUC (%) ↑	w. F1-score (%)↑
Local Only	74.43 (13.60)	93.98 (4.53)	71.34 (13.00)
FedAvg+FT	80.92 (9.37)	96.75 (2.32)	77.40 (8.87)
FedBN	84.34 (10.61)	97.49 (2.19)	80.99 (10.02)
FedProx	80.50 (11.32)	96.46 (2.77)	76.93 (10.82)
Ditto	67.30 (18.50)	92.10 (6.63)	63.95 (18.54)
FedRep	55.44 (20.40)	85.55 (11.28)	51.86 (20.74)
FedDual	70.36 (15.59)	93.20 (5.34)	66.99 (15.57)
FedVC(ours)	85.42 (8.95)	97.55 (1.81)	81.88(8.53)
FedVC-sg(ours)	**85.82 (8.47)**	**97.59(1.88)**	**82.27(7.99)**

Table 6.3: Overall performance on the Digit-5 dataset on the training clients. The standard deviation of each metric is reported in parentheses. avg. is the abbreviation of 'averaged' and w. denotes the 'weighted'. The ↑ denotes that the higher the metric is, the better performance a model achieved, and the best performance is highlighted in bold.

	avg. Accuracy (%) ↑	w. AUC (%) ↑	w. F1-score (%)↑
FedAvg+FT	77.85 (7.71)	96.26 (1.99)	74.57 (7.42)
FedBN	83.30 (7.05)	97.45 (1.32)	79.69 (6.67)
FedProx	76.90 (7.92)	96.19 (2.02)	73.55 (7.31)
FedRep	34.85 (18.75)	73.08 (11.66)	30.24 (18.39)
FedDual	67.15 (11.46)	92.74 (4.66)	63.85 (11.62)
FedVC(ours)	**86.20 (5.62)**	97.61 (1.38)	**82.92 (5.15)**
FedVC-sg(ours)	85.10 (5.92)	**97.68(1.39)**	81.61 (5.70)

Table 6.4: Overall performance on the Digit-5 dataset on the test clients. The standard deviation of each metric is reported in parentheses. avg. is the abbreviation of 'averaged' and w. denotes the 'weighted'. The ↑ denotes that the higher the metric is, the better performance a model achieved, and the best performance is highlighted in bold.

6.3.4.2 Feature Shift Settings

This section demonstrates evaluations in feature shift data. **Table 6.3** shows that FedVC achieves the best accuracy, AUC and F1 score under this setting. Other models are less robust than FedVC and their performances vary significantly among clients (higher standard deviations). Group-wised performance in **Figure 6.10** shows that FedVC has a smaller performance gap between different domains and it is more robust for that there is less fluctuation in the learning curves.

6.3.5 Ablation Study

This section evaluates the effectiveness of FedVC through experiments on the Digit-5 dataset. The section first validates virtual concepts' capability as supervised information for personalisation by visualising the distribution of estimated client preferences ($\hat{p}$). Then, it analyses the behaviours of hyperparameters by ablation experiments.

6.3.5.1 Interpreting Personalisation

Figure 6.7 compares the latent representations learned by FedAvg and the FedVC. It can be found that FedVC succeeds in supervising the learning process with client preferences so that the distribution of the estimated client preferences $\hat{p}$ are consistent with the group truth knowledge, i.e., samples from the same group (colours) are closer to each other.

ι in **Equation 6.1** is a hyperparameter that weights the importance of the difference $|\hat{z} - c|$ when estimating the client preference $\hat{p}$. **Figure 6.8(a)** shows that client preferences (colours) are unrecognisable with a model learned with a small ι, i.e., $\iota = 0.001$. With the increasing of ι, the estimated $\hat{p}$ demonstrates structure consistent with their client preferences (**Figure 6.8(b-d)**). It validates the effectiveness of the supervision of virtual concepts c. The superior performance of FedVC denotes such supervision does improve the performance of a global model, and virtual concepts are indicators that can be utilised to interpret personalisation.

6.3.5.2 Hyperparameters

The experiments study a hyperparameter's behaviours by evaluating model performance under different values of the selected hyperparameter while holding the others with default values. According to **Table 6.5** and **Table 6.6**, model performance will be improved along with the increasing of the number and the dimension of virtual concepts.

# of VCs	avg. accuracy (%)↑ on tr clients	avg. accuracy (%)↑ on ts clients
3	85.22(9.47)	85.05(6.79)
6	85.24(9.10)	85.15(6.06)
10	**85.42(8.95)**	**86.20(5.62)**

Table 6.5: Performance with different number of virtual concepts

d-VC	avg. accuracy (%)↑ on tr clients	avg. accuracy (%)↑ on ts clients
3	83.46(9.89)	83.45(6.95)
6	84.36(9.75)	83.80(7.07)
10	**85.42(8.95)**	**86.20(5.62)**

Table 6.6: Performance with different dimensions of virtual concepts

ι	avg. accuracy (%)↑ on tr clients	avg. accuracy (%)↑ on ts clients
0.001	84.40(9.30)	85.00(6.34)
0.005	84.56(9.55)	85.35(6.44)
0.01	**85.74(9.12)**	85.75(6.25)
0.1	85.42(8.95)	**86.20(5.62)**

Table 6.7: Performance with different similarity parameter ι. The larger the ι is, the more weight the difference $|\hat{z} - c|$ when estimating the client preference $\hat{p}$

Table 6.7 shows that a larger weight for the similarity between $\hat{z}$ and c will increase model performance, which validates the effectiveness of the supervision from virtual concepts. In addition, **Table 6.8** indicates that the newly estimated S, C and N will outperform the older one when using the moving average strategy. **Table 6.9** suggests that γ needs to be carefully selected when balancing updating the global model and the virtual concepts.

6.4 Conclusions

The research proposes to utilise virtual concepts as client supervision information to learn a robust global model and to interpret the non-IID data across clients. Specifically, the proposed FedVC interprets each client's preferences as a mixture of conceptual vectors that each represents an interpretable concept to end-users. These conceptual

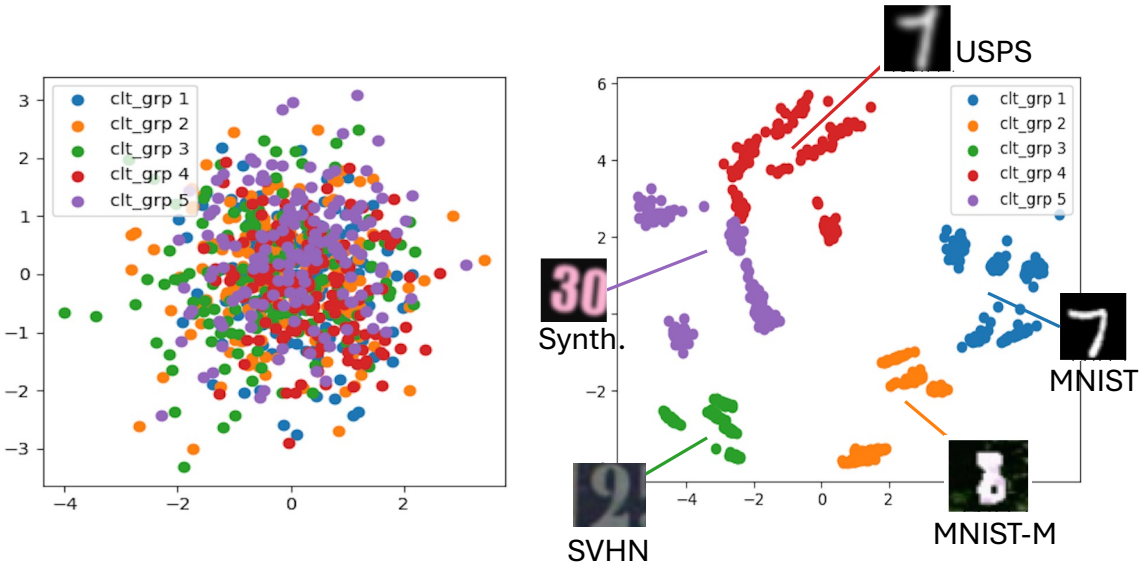
κ	avg. accuracy (%)↑ on tr clients	avg. accuracy (%)↑ on ts clients
0.01	**85.66(8.51)**	85.40(6.12)
0.05	85.42(8.95)	**86.20(5.62)**
0.1	84.96(9.23)	85.45(6.18)
0.5	84.44(9.16)	84.65(6.25)
0.95	84.02(9.51)	83.85(7.29)

Table 6.8: Performance with different smoothing parameter κ. The larger the κ is, the more weight the previous estimation of S, C and N.

γ	avg. accuracy (%)↑ on tr clients	avg. accuracy (%)↑ on ts clients
0.01	83.14(10.47)	83.40(7.27)
0.1	85.24(8.97)	**85.30(6.86)**
0.5	83.48(10.63)	82.35(8.03)
0.95	**85.46(8.71)**	85.20(6.02)

Table 6.9: Performance with different balancing parameter γ. The larger the γ is, the more important the loss l_p to optimising the virtual concepts c.

vectors could be learnt via the optimisation procedure of the federated learning system. In addition to the interpretability, the clarity of client-specific personalisation could also be applied to enhance the robustness of the training process on the FL system. The effectiveness of the proposed methods has been validated on benchmark datasets.



(a) distribution of sample representations after FedAvg

(b) distribution of estimated preferences $\hat{p}$ by FedVC

Figure 6.7: Distribution of estimated client preferences. Colours indicate the client group, i.e., the domain, samples belong to. (a) The aggregation process by vanilla FedAvg will eliminate the information on client preferences so that sample representations are mixed regarding their domains. (b) Virtual concepts succeed in supervising the learning process with client preferences so that the distribution of the estimated client preferences $\hat{p}$ are consistent with their domain knowledge, i.e., samples from the same domain will be closer to each other.

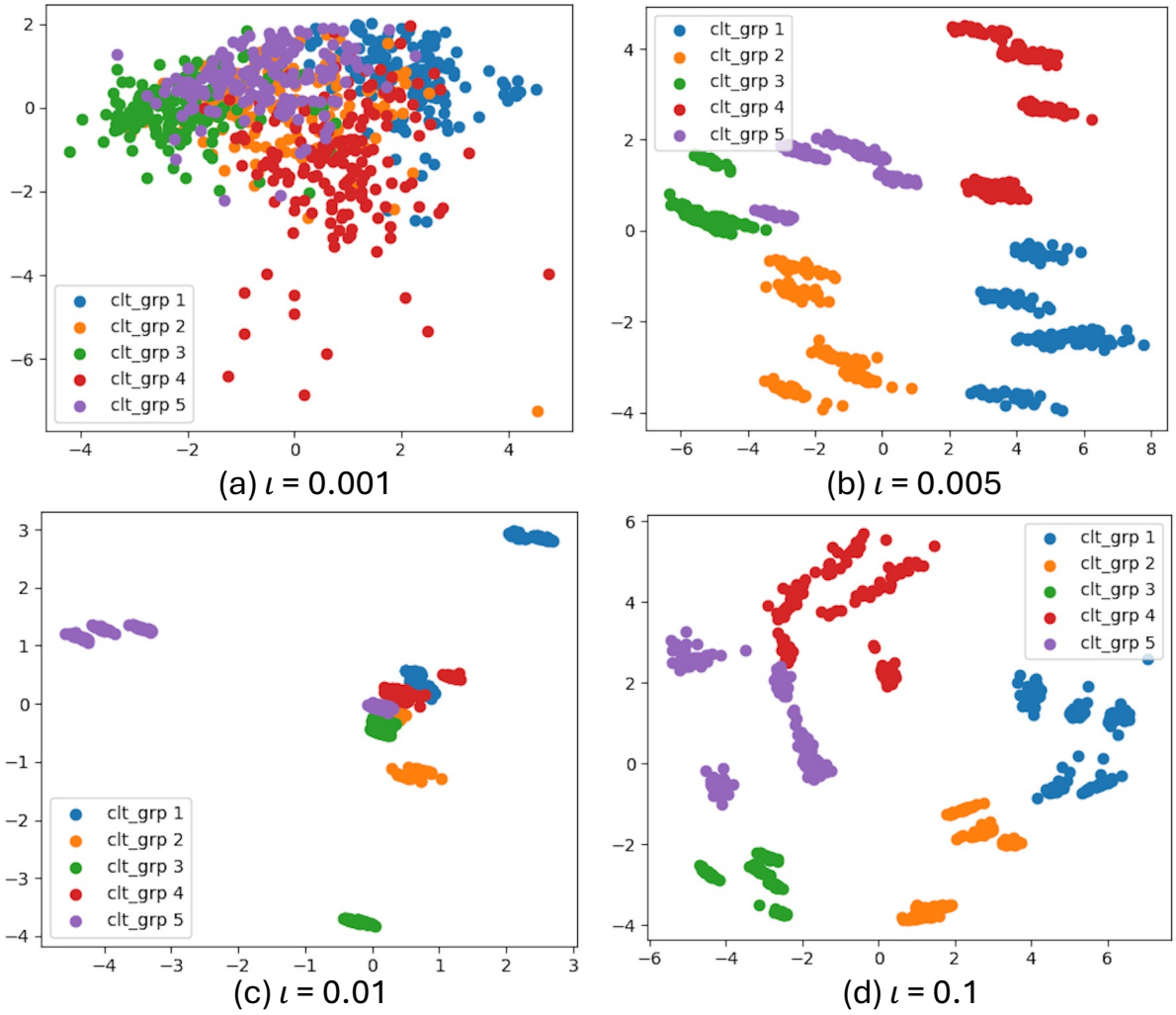


Figure 6.8: Distribution of estimated client preferences with different ι. The smaller the ι is, the less weight the difference $|\hat{z} - c|$ when estimating the client preference $\hat{p}$.

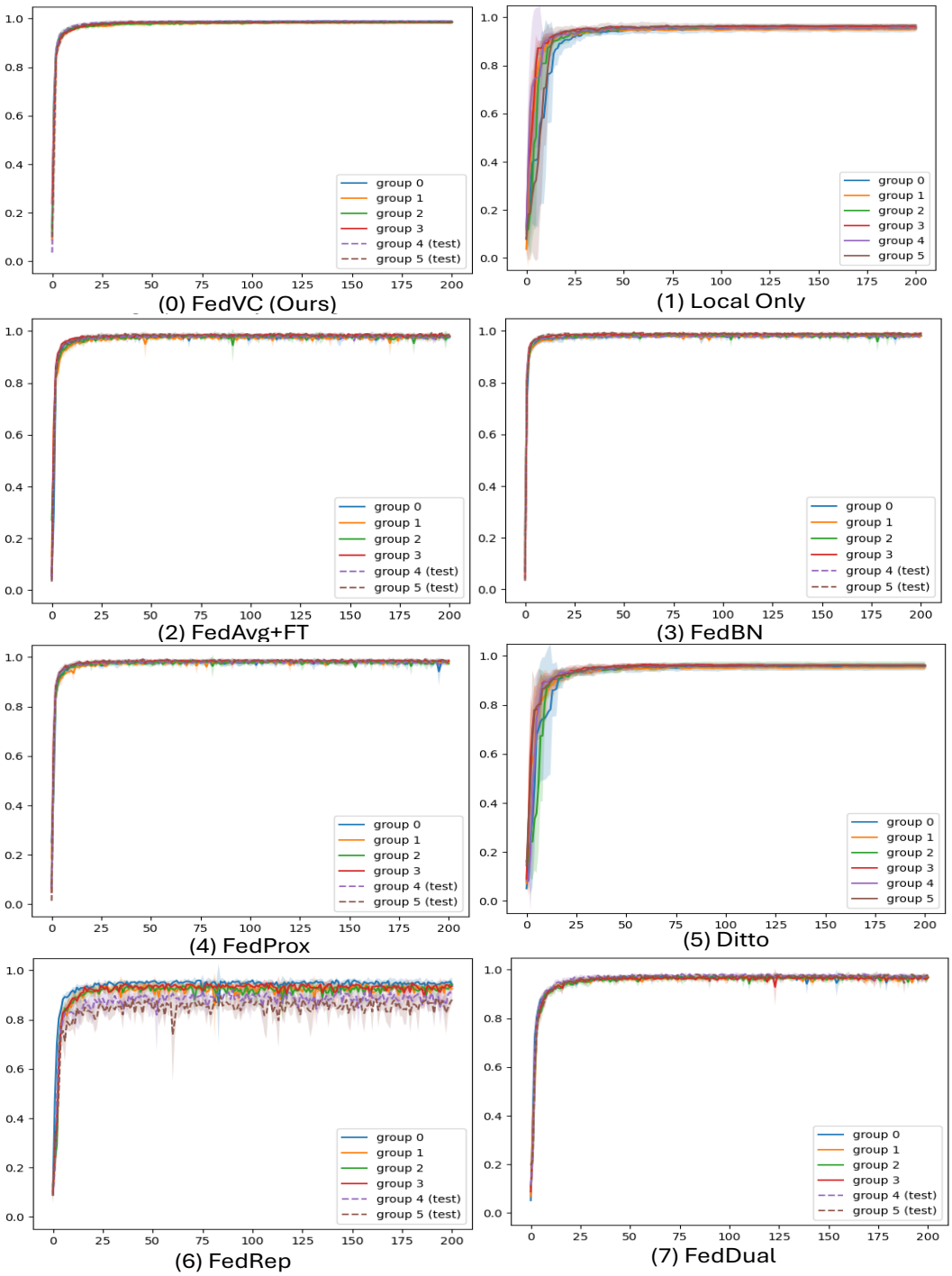


Figure 6.9: Grouped-wise accuracy on MNIST. The horizontal axis denotes communication rounds and the vertical axis denotes the accuracy. Each colour corresponds to a client group, i.e., data distribution. Shade indicates the standard deviation of accuracy among clients in the group.

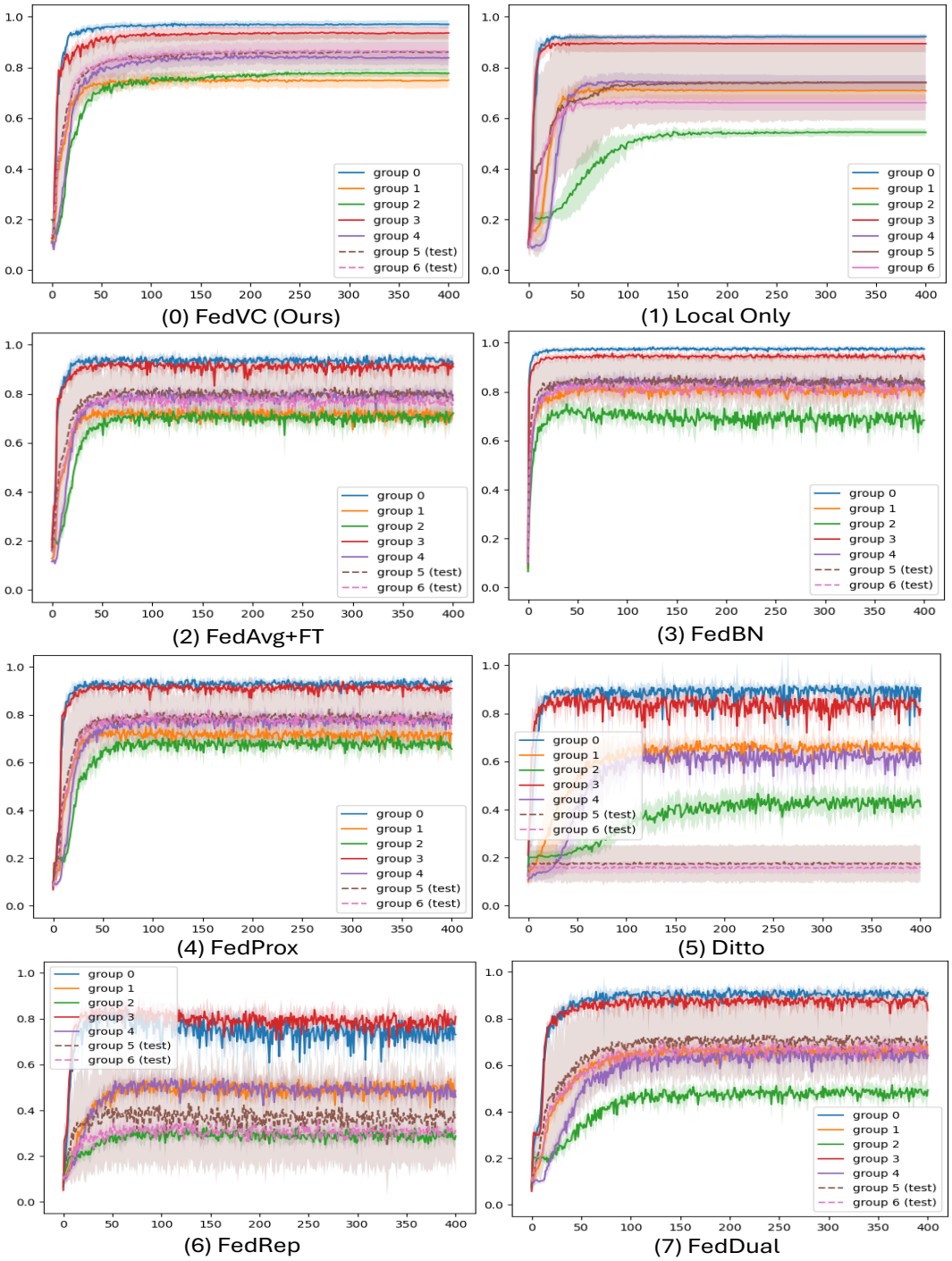


Figure 6.10: Grouped-wise accuracy on Digit-5. The horizontal axis denotes communication rounds and the vertical axis denotes the accuracy. Each colour corresponds to a client group, i.e., data distribution. Shade indicates the standard deviation of accuracy among clients in the group.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusion

This research focuses on explainable model personalisation for FL systems. The research aims to recognise preferences implied in a client's local data and propose on-deployment personalisation where clients can obtain practical and explainable model outputs without extra local adaptations. Three objectives have been achieved, with summaries as follows.

Firstly, **Chapter 4** studies personalisation by disentangling common and personalised knowledge in an FL system. A novel Federated Dual Variational Autoencoder (FedDVA) framework is proposed to disentangle sample representations into client-agnostic (common) and client-specific (personalised) parts. Then, the disentangled representations will demonstrate meaningful structures describing clients' preferences, which will help us better understand features contributing to the personalisation and study their influences on the final predictions. Meanwhile, on-deployment personalisation can be implemented efficiently by training a lightweight classification model over the disentangled repre-

sentations. A client-specific Evidence Lower BOund (ELBO) is derived for learning the FedDVA model, and it is formulated into an optimisation problem that gradient-based methods can solve.

Further, **Chapter 5** explains personalisation from the representation level by aligning latent spaces of global models in FL. A new Client-Decorrelation Federated Learning (FedCD) is proposed to explore client properties and facilitate explainable personalisation. The FedCD utilises bias in representations of a client's local data to recognise the client's properties. Then, it aligns the global model's hidden space with axes representing client properties, unravelling a client's influence from its sample's latent representations. The proposed Representation Alignment (RA) mechanism in FedCD could become a plug-in component to be integrated with any FL models, which enables various classic FL models to be directly deployed for PerFL tasks without needing client-specific modifications. The overall learning objective of FedCD is fulfilled by solving a bi-level optimisation problem that clients can solve collaboratively under the standard FL framework.

Moreover, **Chapter 6** introduces Virtual Concepts (VCs) to explicate clients' preferences and model personalisation. The VCs are a set of vectors describing structures of data partitions of an FL system. They constitute client-supervised information that characterises biases implied in clients' local data. Then, personalisation becomes explicit and explainable by including VCs as labels of clients' preferences in FL's training process. The learning process of VCs could be formulated into a Gaussian Mixture Model (GMM) that can be solved by Expectation Maximisation (EM) based methods or optimised along with the standard federated learning process through gradient-based methods.

In conclusion, this thesis solves the explainable personalisation problem from different perspectives and proposes three algorithms: FedDVA disentangles personalised representation; FedCD aligns the global model's hidden spaces; and FedVC, which extracts data structures as supervised information. These novel methods provide ways to

get deep insight into personalisation procedures in PerFL and inspire us to design models that can achieve on-deployment personalisation. Theoretical analyses and comprehensive experiments substantiate the proposed methods and findings.

7.2 Future works

As we get deeper into the field of PerFL, there is more to explore and understand about what and how factors contribute to model personalisation.

Firstly, this research has succeeded in recognising clients' preferences and utilising them to boost model personalisation. Can we combine the proposed methods with conventional supervised interpreting methods, e.g., the Shapely Values, to quantify the impacts of client preferences in a prediction? Can we design a mechanism to further study the causality rather than the correlation between client preferences and personalisation?

Besides, there have been many heterogeneous scenarios in FL beyond the scope of non-I.I.D. data. Clients may differ in learning tasks, model architectures and hardware capabilities. Can we extend the proposed methods to extract such differences so that a PerFL model can be deployed for more complicated applications?

Thirdly, current PerFL research focuses on heterogeneity across clients, while data distribution and preferences of a client may also change over time. Personalisation needs to be able to capture changes in a client's local data and track emerging tendencies in the system. Uncovering and explaining personalisation in such a dynamic environment could be more challenging.

Last but not least, the success of foundation models, e.g., Llama, has brought machine learning into a new era. Clients in an FL system are now able to retrieve common knowledge from a powerful foundation model directly rather than training a model from scratch. However, as the large foundation models are usually trained by tech giants opaquely, it becomes even more critical to understand the factors behind a personalised

prediction in casing bias and discrimination in the decision process.

In summary, as AI is playing a more and more important role in our daily lives, FL becomes necessary to keep our privacy safe and explainable personalisation becomes the key to guaranteeing AI trustworthiness. In turn, a transparent and comprehensible model will advance our understanding and capabilities in designing more powerful and reliable AI systems.

APPENDIX

A.1 FedDVA

A.1.1 Evidence Lower Bounds

A.1.1.1 ELBO optimizing $q(z|x)$

Suppose $p(z|x)$ is the true posterior of z, $q(z|x)$ is the variational posterior approximating $p(z|x)$ and samples on the same client are independent and identical distributed (iid.), then the learning task on the k-th client is to minimize $D_{KL}(q(z|x)||p(z|x))$, which is

$$\begin{aligned} D_{KL}(q(z|x)||p(z|x)) &= \int q(z|x)\log\frac{q(z|x)}{p(z|x)}\mathrm{d}z \\ &= \int q(z|x)\log\frac{q(z|x)p_k(x)}{p(z|x)p_k(x)}\mathrm{d}z \\ &= \log p_k(x) + \int q(z|x)\log\frac{q(z|x)}{p_k(x|z)p(z)}\mathrm{d}z \\ &= \log p_k(x) - \mathbb{E}_{q(z|x)}[\log p_k(x|z)] + D_{KL}(q(z|x)||p(z)) \end{aligned} \tag{A.1}$$

or equivalently,

$$\log p_k(x) \geq ELBO_z(x,k) = \mathbb{E}_{q(z|x)}[\log p_k(x|z)] - D_{KL}(q(z|x)||p(z)) \tag{A.2}$$

A.1.1.2 ELBO optimizing $q(z|x,c)$

Suppose $p(c|x,z)$ is the true posterior of c, $q(c|x,z)$ is the variational posterior approximating $p(c|x,z)$ and samples on the same client are iid., then the learning task on the k-th client is to minimize $D_{KL}(q(c|x,z)||p(c|x,z))$, which is

$$\begin{aligned} D_{KL}(q(c|x,z)||p(c|x,z)) &= \int q(c|x,z)\log\frac{q(c|x,z)}{p(c|x,z)}\mathrm{d}c \\ &= \int q(c|x,z)\log\frac{q(c|x,z)p_k(x|z)}{p(c|x,z)p_k(x|z)}\mathrm{d}c \\ &= \log p_k(x|z) + \int q(c|x,z)\log\frac{q(c|x,z)}{p_k(x,c|z)}\mathrm{d}c \\ &= \log p_k(x|z) - \mathbb{E}_{q(c|x,z)}[\log p_k(x,c|z)] - H(q(c|x,z)) \end{aligned} \tag{A.3}$$

Ideally, there is a client-irrelevant likelihood $p(x|z,c)$ modeling the sample generating process, that is $p_k(x) = \iint p(x|z,c)p(z)p_k(c)\mathrm{d}z\mathrm{d}c$, where the personality of a client lies on $p_k(c)$. Then we have

$$\log p_k(x) \geq ELBO_c(x,z,k) = \mathbb{E}_{q(c|x,z)}[\log p(x|z,c)] - D_{KL}(q(c|x,z)||p_k(c)) \tag{A.4}$$

which is equivalent to

$$\log p_k(x) \geq ELBO'_c(x,z,k) = \mathbb{E}_{q(c|x,z)}[\log p_k(x|z,c)] - D_{KL}(q(c|x,z)||q(c)) \tag{A.5}$$

A.1.1.3 Difference between $D_{KL}(q(c|x,z)||q(c))$ and $D_{KL}(q(c|x,z)||\bar{p}_k(c))$

For samples on the same client and suppose they are independent and identical distributed, according to Eq.4.2, there is

$$\begin{aligned} &\mathbb{E}_{p_k(x)}[\mathbb{E}_{q(z|x)}[D_{KL}(q(c|x,z)||q(c)) - D_{KL}(q(c|x,z)||\bar{p}_k(c))]] \\ =&\mathbb{E}_{p_k(x)}[\mathbb{E}_{q(z|x)}[\mathbb{E}_{q(c|x,z)}[\log\bar{p}_k(c) - \log q(c)]]] \\ =&\mathbb{E}_{\bar{p}_k(c)}[\log\bar{p}_k(c) - \log q(c)] \\ =&D_{KL}(\bar{p}_k(c)||q(c)) \\ \geq&\xi_k \end{aligned} \tag{A.6}$$

A.1.2 Computation of the KL-Divergence

A.1.2.1 KL-Divergence between two Gaussian distributions

For the i-th sample x_i and j-th sample x_j, variational posteriors inferring representation c are $q(c|x_i,z_i)=\mathcal{N}(c;\mu_i,\Sigma_i)$ and $q(c|x_j,z_j)=\mathcal{N}(c;\mu_j,\Sigma_j)$, where c is a d-dimensional vector and convariance matrices of Σ_i and Σ_j are diagonal. Then we have

$$\begin{aligned}\int q(c|x_i,z_i)\log q(c|x_i,z_i)\mathrm{d}c &= \int \mathcal{N}(c;\mu_i,\Sigma_i)\log\mathcal{N}(c;\mu_i,\Sigma_i)\mathrm{d}c \\ &= -\frac{1}{2}(d\log(2\pi)+\log|\Sigma_i|+d)\end{aligned} \tag{A.7}$$

and

$$\begin{aligned}\int q(c|x_i,z_i)\log q(c|x_j,z_j)\mathrm{d}c &= \int \mathcal{N}(c;\mu_i,\Sigma_i)\log\mathcal{N}(c;\mu_j,\Sigma_j)\mathrm{d}c \\ &= -\frac{1}{2}(d\log(2\pi)+\log|\Sigma_j|+Tr(\Sigma_j^{-1}\Sigma_i)+(\mu_i-\mu_j)^T\Sigma_j^{-1}(\mu_i-\mu_j))\end{aligned} \tag{A.8}$$

Combining Eq.A.7 and Eq.A.8, the KL-Divergence between $q(c|x_i,z_i)$ and $q(c|x_j,z_j)$ is

$$\begin{aligned}D_{KL}(q(c|x_i,z_i)||q(c|x_j,z_j)) &= \int q(c|x_i,z_i)(\log q(c|x_i,z_i)-\log q(c|x_j,z_j))\mathrm{d}c \\ &= \int \mathcal{N}(c;\mu_i,\Sigma_i)(\log\mathcal{N}(c;\mu_i,\Sigma_i))\mathrm{d}c - \int \mathcal{N}(c;\mu_i,\Sigma_i)(\log\mathcal{N}(c;\mu_j,\Sigma_j))\mathrm{d}c \\ &= \frac{1}{2}[(\mu_i-\mu_j)^T\Sigma_j^{-1}(\mu_i-\mu_j)-\log|\Sigma_j^{-1}\Sigma_i|+Tr(\Sigma_j^{-1}\Sigma_i)-d] \\ &= \frac{1}{2}\sum_{l=1}^{d}[(\frac{\mu_i^{(l)}-\mu_j^{(l)}}{\sigma_j^{(l)}})^2-\log(\frac{\sigma_i^{(l)}}{\sigma_j^{(l)}})^2+(\frac{\sigma_i^{(l)}}{\sigma_j^{(l)}})^2-1]\end{aligned} \tag{A.9}$$

where l denotes the l-th element and $\sigma_i^{(l)}$ denotes the positive root of the l-th element on the diagonal of covariance matrix Σ_i.

A.1.2.2 Computation of $D_{KL}(q(c|x,z)||\bar{p}_k(c))$

Let x_i and x_j denotes the i-th and j-th sample in dataset $\mathscr{D}_k$ with size is n_k

$$\begin{aligned} D_{KL}(q(c|x_i,z_i)||\bar{p}_k(c)) =& \mathbb{E}_{q(c|x_i,z_i)}[\log q(c|x_i,z_i) - \log \frac{1}{n_k}\sum_{j=1}^{n_k}[q(c|x_j,z_j)] \\ \leq& \mathbb{E}_{q(c|x_i,z_i)}[\log q(c|x_i,z_i) - \frac{1}{n_k}\sum_{j=1}^{n_k}\log q(c|x_j,z_j)] \\ =& \frac{1}{n_k}\sum_{j=1}^{n_k}\mathbb{E}_{q(c|x_i,z_i)}[\log q(c|x_i,z_i) - \log q(c|x_j,z_j)] \end{aligned} \tag{A.10}$$

Bringing Eq.A.9 we have

$$D_{KL}(q(c|x_i,z_i)||\bar{p}_k(c)) \leq \frac{1}{2n_k}\sum_{j=1}^{n_k}\sum_{l=1}^{d}[(\frac{\mu_i^{(l)}-\mu_j^{(l)}}{\sigma_j^{(l)}})^2 - \log(\frac{\sigma_i^{(l)}}{\sigma_j^{(l)}})^2 + (\frac{\sigma_i^{(l)}}{\sigma_j^{(l)}})^2 - 1] \tag{A.11}$$

where μ_i, σ_i and μ_j, σ_j are outputs of neural networks and they can be differentiated and optimized by gradient based optimization methods.

A.2 FedCD

We introduce an inductive bias to align the hidden layers of a DNN so that it is able to learn client bias to achieve personalization without on-device fine-tuning. We assume data on the same client are influenced by the same client properties so that data from the similar clients will have similar representations. We formulate the representation alignment problem into an optimization described in Eq.5.3.

Eq.5.3 is equivalent to the objective function of Linear Discriminant Analysis (LDA) whose solution is the enginvector of corresponding to the largest enginvalue of $\Sigma_W^{-1}\Sigma_B$. However, the computation cost of the decomposing $\Sigma_W^{-1}\Sigma_B$ would be high, and aggregating local representations to a server is infeasible in FL. Therefore, we divide the matrix decomposition process into a set of clients' local updating steps and integrate it into standard FL framework.

A.2.1 Client Supervised Optimization

Let Σ_G denote the correlation matrix of latent representations on all clients, there is $\Sigma_G = \Sigma_W + \Sigma_B$. Then the learning problem can be formulate as solving the following eigenvalue problem

$$\Sigma_W^{-1}\Sigma_G P^* = P^*\Lambda \tag{A.12}$$

where Λ is the diagonal eigenvalue matrix of $\Sigma_W^{-1}\Sigma_G$. [31] shows solving Eq.A.12 can be simplified into solving the following symmetric eigenvalue problem:

$$\Sigma_W^{-1/2}\Sigma_G\Sigma_W^{-1/2}\Phi = \Phi\Lambda \tag{A.13}$$

where Φ denotes eignevectors of $\Sigma_W^{-1/2}\Sigma_G\Sigma_W^{-1/2}$, and there is $P^* = \Sigma_W^{-1/2}\Phi$. To find the optimum P^* is to find Φ and $\Sigma_W^{-1/2}$.

[8] introduced an incremental algorithm to optimize Φ and $\Sigma_W^{-1/2}$ through which we can distribute the optimizing steps to clients. Concretely, it proves that, 1)

$$\Phi_{k+1} = \Phi_k + \lambda(\mathbf{z}_k\mathbf{z}_k^T\Phi_k - \Phi_k\tau[\Phi_k^T\mathbf{z}_k\mathbf{z}_k^T\Phi_k]) \tag{A.14}$$

will converge to the enginvector matrix Φ when there are sufficient instance $\mathbf{z}_k$ sampled from the data distribution; 2). let $\mathbf{S}$ denotes $\Sigma_W^{-1/2}$, then

$$S_{k+1} = S_k + \eta * (I - S_k\Sigma_W S_k) \tag{A.15}$$

where S_{k+1} will converge to the inverted square root of Σ_W when 1) S_0 is initialized as a symmetric positive define matrix, and 2) there are sufficient instance $\mathbf{z}_k$ sampled from the data distribution.

We apply the above methods in Eq.5.7 and Eq.5.6 to update P on each clients individually and aggregate local updates to align axis on different clients.

A.2.2 Discussion on Privacy Protection

According to the section above, clients requires to share local correlations to update the matrix Σ_W. However, Σ_W is a global statistic where a client's local bias would be eliminated, privacy-protection methods like differential privacy are feasible to avoid privacy leakage.

BIBLIOGRAPHY

[1] J. ADEBAYO, J. GILMER, M. MUELLY, I. GOODFELLOW, M. HARDT, AND B. KIM, *Sanity checks for saliency maps*, Advances in neural information processing systems, 31 (2018).

[2] G. ALAIN AND Y. BENGIO, *Understanding intermediate layers using linear classifier probes*, arXiv preprint arXiv:1610.01644, (2016).

[3] D. W. APLEY AND J. ZHU, *Visualizing the effects of predictor variables in black box supervised learning models*, Journal of the Royal Statistical Society Series B: Statistical Methodology, 82 (2020), pp. 1059–1086.

[4] APPLE, *Designing for privacy*. `https://developer.apple.com/videos/play/wwdc2019/708`, 2019.

[5] APPLE, *Private federated learning*. `https://nips.cc/Expo/Conferences/2019`, 2019.

[6] A. BELLET, R. GUERRAOUI, M. TAZIKI, AND M. TOMMASI, *Personalized and private peer-to-peer machine learning*, in International conference on artificial intelligence and statistics, PMLR, 2018, pp. 473–481.

[7] Y. BENGIO, A. COURVILLE, AND P. VINCENT, *Representation learning: A review and new perspectives*, IEEE transactions on pattern analysis and machine intelligence, 35 (2013), pp. 1798–1828.

[8] C. CHATTERJEE AND V. ROYCHOWDHURY, *On self-organizing algorithms and networks for class-separability features*, IEEE Transactions on Neural Networks, 8 (1997), pp. 663–678.

[9] A. CHATTOPADHAY, A. SARKAR, P. HOWLADER, AND V. N. BALASUBRAMANIAN, *Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks*, in 2018 IEEE winter conference on applications of computer vision (WACV), IEEE, 2018, pp. 839–847.

[10] F. CHEN, G. LONG, Z. WU, T. ZHOU, AND J. JIANG, *Personalized federated learning with a graph*, in (IJCAI) International Joint Conference on Artificial Intelligence, vol. 2022, https://www.ijcai.org/proceedings/2022/0357.pdf, 2022, pp. 2575–2582.

[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, R. MATHEWS, T. OUYANG, AND F. BEAUFAYS, *Federated learning of out-of-vocabulary words*, 2019.

[13] R. T. CHEN, X. LI, R. GROSSE, AND D. DUVENAUD, *Isolating sources of disentanglement in vaes*, in Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2019, pp. 2615–2625.

[14] S. CHEN, G. LONG, J. JIANG, AND C. ZHANG, *Personalized adapter for large meteorology model on devices: Towards weather foundation models*, in NeurIPS, vol. 2024, 2024.

[15] S. CHEN, G. LONG, T. SHEN, AND J. JIANG, *Prompt federated learning for weather forecasting: Toward foundation models on meteorological data*, in (IJCAI) Inter-

national Joint Conference on Artificial Intelligence, vol. 2023, 2023, pp. 3532–3540.

[16] S. CHEN, T. ZHOU, G. LONG, J. MA, J. JIANG, AND C. ZHANG, *Multi-level additive modeling for structured non-iid federated learning*, arXiv preprint arXiv:2405.16472, (2024).

[17] Z. CHEN, Y. BEI, AND C. RUDIN, *Concept whitening for interpretable image recognition*, Nature Machine Intelligence, 2 (2020), pp. 772–782.

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

[19] I. COLIN, A. BELLET, J. SALMON, AND S. CLÉMENÇON, *Gossip dual averaging for decentralized optimization of pairwise functions*, in International Conference on Machine Learning, PMLR, 2016, pp. 1388–1396.

[20] L. COLLINS, H. HASSANI, A. MOKHTARI, AND S. SHAKKOTTAI, *Exploiting shared representations for personalized federated learning*, in Proceedings of the 38th International Conference on Machine Learning, M. Meila and T. Zhang, eds., vol. 139 of Proceedings of Machine Learning Research, PMLR, 18–24 Jul 2021, pp. 2089–2099.

[21] L. COLLINS, H. HASSANI, A. MOKHTARI, AND S. SHAKKOTTAI, *Fedavg with fine tuning: Local updates lead to representation learning*, Advances in Neural Information Processing Systems, 35 (2022), pp. 10572–10586.

[22] E. CORDIS., *Teaching machines to help with drug discovery*. `https://cordis.europa.eu/project/id/831472`, Jun 2019.

[23] P. COURTIOL, C. MAUSSION, M. MOARII, E. PRONIER, S. PILCER, M. SEFTA, P. MANCERON, S. TOLDO, M. ZASLAVSKIY, N. LE STANG, ET AL., *Deep*

learning-based classification of mesothelioma improves prediction of patient outcome, Nature medicine, 25 (2019), pp. 1519–1525.

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

[25] R. L. DRAELOS AND L. CARIN, *Use hirescam instead of grad-cam for faithful explanations of convolutional neural networks*, arXiv e-prints, (2020), pp. arXiv–2011.

[26] J.-H. DUAN, W. LI, D. ZOU, R. LI, AND S. LU, *Federated learning with data-agnostic distribution fusion*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 8074–8083.

[27] R. O. DUDA, P. E. HART, ET AL., *Pattern classification*, John Wiley & Sons, 2006.

[28] A. ELGABLI, J. PARK, A. S. BEDI, M. BENNIS, AND V. AGGARWAL, *Gadmm: Fast and communication efficient framework for distributed machine learning*, Journal of Machine Learning Research, 21 (2020), pp. 1–39.

[29] A. FALLAH, A. MOKHTARI, AND A. OZDAGLAR, *Personalized federated learning: A meta-learning approach*, arXiv preprint arXiv:2002.07948, (2020).

[30] C. FINN, P. ABBEEL, AND S. LEVINE, *Model-agnostic meta-learning for fast adaptation of deep networks*, in International Conference on Machine Learning, PMLR, 2017, pp. 1126–1135.

[31] Y. A. GHASSABEH, F. RUDZICZ, AND H. A. MOGHADDAM, *Fast incremental lda feature extraction*, Pattern Recognition, 48 (2015), pp. 1999–2012.

[32] A. GHORBANI, J. WEXLER, J. Y. ZOU, AND B. KIM, *Towards automatic concept-based explanations*, Advances in Neural Information Processing Systems, 32 (2019).

[33] A. GHOSH, J. CHUNG, D. YIN, AND K. RAMCHANDRAN, *An efficient framework for clustered federated learning*, arXiv preprint arXiv:2006.04088, (2020).

[34] I. GOODFELLOW, Y. BENGIO, AND A. COURVILLE, *Deep Learning*, MIT Press, 2016.

[35] K. GREGOR, G. PAPAMAKARIOS, F. BESSE, L. BUESING, AND T. WEBER, *Temporal difference variational auto-encoder*, arXiv preprint arXiv:1806.03107, (2018).

[36] K. S. GURUMOORTHY, A. DHURANDHAR, G. CECCHI, AND C. AGGARWAL, *Efficient data representation by selecting prototypes with importance weights*, in 2019 IEEE International Conference on Data Mining (ICDM), IEEE, 2019, pp. 260–269.

[37] A. HARD, K. RAO, R. MATHEWS, S. RAMASWAMY, F. BEAUFAYS, S. AUGENSTEIN, H. EICHNER, C. KIDDON, AND D. RAMAGE, *Federated learning for mobile keyboard prediction*, 2019.

[38] T. HASTIE, R. TIBSHIRANI, J. H. FRIEDMAN, AND J. H. FRIEDMAN, *The elements of statistical learning: data mining, inference, and prediction*, vol. 2, Springer, 2009.

[39] I. HIGGINS, L. MATTHEY, A. PAL, C. BURGESS, X. GLOROT, M. BOTVINICK, S. MOHAMED, AND A. LERCHNER, *beta-vae: Learning basic visual concepts with a constrained variational framework*, (2016).

[40] G. E. HINTON AND S. ROWEIS, *Stochastic neighbor embedding*, Advances in neural information processing systems, 15 (2002).

[41] K. HSIEH, A. PHANISHAYEE, O. MUTLU, AND P. GIBBONS, *The non-iid data quagmire of decentralized machine learning*, in International Conference on Machine Learning, PMLR, 2020, pp. 4387–4398.

[42] T.-M. H. HSU, H. QI, AND M. BROWN, *Measuring the effects of non-identical data distribution for federated visual classification*, 2019.

[43] J. JIANG, S. JI, AND G. LONG, *Decentralized knowledge acquisition for mobile internet applications*, World Wide Web Journal (WWWJ), 5 (2020), pp. 2653–2669.

[44] Y. JIANG, J. KONEČNÝ, K. RUSH, AND S. KANNAN, *Improving federated learning personalization via model agnostic meta learning*, arXiv preprint arXiv:1909.12488, (2019).

[45] P. KAIROUZ, H. B. MCMAHAN, 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*, arXiv preprint arXiv:1912.04977, (2019).

[46] A. KARPATHY, J. JOHNSON, AND L. FEI-FEI, *Visualizing and understanding recurrent networks*, arXiv preprint arXiv:1506.02078, (2015).

[47] B. KIM, R. KHANNA, AND O. O. KOYEJO, *Examples are not enough, learn to criticize! criticism for interpretability*, Advances in neural information processing systems, 29 (2016).

[48] B. KIM, M. WATTENBERG, J. GILMER, C. CAI, J. WEXLER, F. VIEGAS, ET AL., *Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)*, in International conference on machine learning, PMLR, 2018, pp. 2668–2677.

[49] H. KIM AND A. MNIH, *Disentangling by factorising*, in International Conference on Machine Learning, PMLR, 2018, pp. 2649–2658.

[50] P.-J. KINDERMANS, S. HOOKER, J. ADEBAYO, M. ALBER, K. T. SCHÜTT, S. DÄHNE, D. ERHAN, AND B. KIM, *The (un) reliability of saliency methods*, in Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, Springer, 2019, pp. 267–280.

[51] D. P. KINGMA AND M. WELLING, *Auto-encoding variational bayes*, arXiv preprint arXiv:1312.6114, (2013).

[52] P. W. KOH AND P. LIANG, *Understanding black-box predictions via influence functions*, in International conference on machine learning, PMLR, 2017, pp. 1885–1894.

[53] P. W. KOH, T. NGUYEN, Y. S. TANG, S. MUSSMANN, E. PIERSON, B. KIM, AND P. LIANG, *Concept bottleneck models*, in International Conference on Machine Learning, PMLR, 2020, pp. 5338–5348.

[54] A. KOLOSKOVA, S. STICH, AND M. JAGGI, *Decentralized stochastic optimization and gossip algorithms with compressed communication*, in International Conference on Machine Learning, PMLR, 2019, pp. 3478–3487.

[55] 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.

[56] T. LI, A. K. SAHU, M. ZAHEER, M. SANJABI, A. TALWALKAR, AND V. SMITH, *Federated optimization in heterogeneous networks*, arXiv preprint arXiv:1812.06127, (2018).

[57] X. LI, M. JIANG, X. ZHANG, M. KAMP, AND Q. DOU, *Fedbn: Federated learning on non-iid features via local batch normalization*, arXiv preprint arXiv:2102.07623, (2021).

[58] Z. LI, G. LONG, AND T. ZHOU, *Federated recommendation with additive personalization*, in (ICLR) The International Conference on Learning Representations, vol. 2024, 2024, pp. 1–17.

[59] Z. LI, G. LONG, T. ZHOU, J. JIANG, AND C. ZHANG, *Personalized federated collaborative filtering: A variational autoencoder approach*, arXiv preprint arXiv:2408.08931, (2024).

[60] T. LIN, L. KONG, S. U. STICH, AND M. JAGGI, *Ensemble distillation for robust model fusion in federated learning*, arXiv preprint arXiv:2006.07242, (2020).

[61] A. LINARDOS, K. KUSHIBAR, S. WALSH, P. GKONTRA, AND K. LEKADIR, *Federated learning for multi-center imaging diagnostics: a simulation study in cardiovascular disease*, Scientific Reports, 12 (2022), p. 3551.

[62] Z. C. LIPTON, *The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery.*, Queue, 16 (2018), pp. 31–57.

[63] F. LOCATELLO, S. BAUER, M. LUCIC, G. RAETSCH, S. GELLY, B. SCHÖLKOPF, AND O. BACHEM, *Challenging common assumptions in the unsupervised learning of disentangled representations*, in international conference on machine learning, PMLR, 2019, pp. 4114–4124.

[64] G. LONG, *The rise of federated intelligence: From federated foundation models toward collective intelligence*, in (IJCAI) Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, https://www.ijcai.org/proceedings/2024/0980.pdf, 2024, pp. 8547–8552.

[65] 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: productivity, well-being, sustainability and partnership, Springer International Publishing Cham, 2021, pp. 113–133.

[66] G. LONG, M. XIE, T. SHEN, T. ZHOU, X. WANG, AND J. JIANG, *Multi-center federated learning: clients clustering for better personalization*, World Wide Web, (2022), pp. 1–20.

[67] K. LU, Z. WANG, P. MARDZIEL, AND A. DATTA, *Influence patterns for explaining information flow in bert*, Advances in Neural Information Processing Systems, 34 (2021), pp. 4461–4474.

[68] N. LU, Z. WANG, X. LI, G. NIU, Q. DOU, AND M. SUGIYAMA, *Federated learning from only unlabeled data with class-conditional-sharing clients*, arXiv preprint arXiv:2204.03304, (2022).

[69] S. M. LUNDBERG AND S.-I. LEE, *A unified approach to interpreting model predictions*, Advances in neural information processing systems, 30 (2017).

[70] Z. LUO, Y. WANG, Z. WANG, Z. SUN, AND T. TAN, *Disentangled federated learning for tackling attributes skew via invariant aggregation and diversity transferring*, arXiv preprint arXiv:2206.06818, (2022).

[71] L. LYU, H. YU, AND Q. YANG, *Threats to federated learning: A survey*, arXiv preprint arXiv:2003.02133, (2020).

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

[73] J. MA, T. ZHOU, G. LONG, J. JIANG, AND C. ZHANG, *Structured federated learning through clustered additive modeling*, in (NeurIPS) Thirty-seventh Conference on Neural Information Processing Systems, vol. 2023, 2023, pp. 1–11.

[74] 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).

[75] 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.

[76] B. MCMAHAN AND D. RAMAGE, *Federated learning: Collaborative machine learning without centralized training data*, Apr 06, 2017.

[77] T. MILLER, *Explanation in artificial intelligence: Insights from the social sciences*, Artificial intelligence, 267 (2019), pp. 1–38.

[78] C. MOLNAR, *Interpretable machine learning*, Lulu. com, 2020.

[79] W. J. MURDOCH, C. SINGH, K. KUMBIER, R. ABBASI-ASL, AND B. YU, *Definitions, methods, and applications in interpretable machine learning*, Proceedings of the National Academy of Sciences, 116 (2019), pp. 22071–22080.

[80] T. D. NGUYEN, T. NGUYEN, P. L. NGUYEN, H. H. PHAM, K. DOAN, AND K.-S. WONG, *Backdoor attacks and defenses in federated learning: Survey, challenges and future research directions*, 2023.

[81] C. OLAH, A. MORDVINTSEV, AND L. SCHUBERT, *Feature visualization*, Distill, (2017).

[82] K. PILLUTLA, K. MALIK, A. MOHAMED, M. RABBAT, M. SANJABI, AND L. XIAO, *Federated learning with partial model personalization*, arXiv preprint arXiv:2204.03809, (2022).

[83] S. RAMASWAMY, R. MATHEWS, K. RAO, AND F. BEAUFAYS, *Federated learning for emoji prediction in a mobile keyboard*, 2019.

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

[85] M. RIBEIRO, S. SINGH, AND C. GUESTRIN, *Anchors: High-precision model-agnostic explanations*, in Proceedings of the AAAI conference on artificial intelligence, vol. 32, 2018.

[86] M. T. RIBEIRO, S. SINGH, AND C. GUESTRIN, *" why should i trust you?" explaining the predictions of any classifier*, in Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 1135–1144.

[87] 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.

[88] D. ROTHCHILD, A. PANDA, E. ULLAH, N. IVKIN, I. STOICA, V. BRAVERMAN, J. GONZALEZ, AND R. ARORA, *Fetchsgd: Communication-efficient federated learning with sketching*, in International Conference on Machine Learning, PMLR, 2020, pp. 8253–8265.

[89] R. R. SELVARAJU, M. COGSWELL, A. DAS, R. VEDANTAM, D. PARIKH, AND D. BATRA, *Grad-cam: Visual explanations from deep networks via gradient-based localization*, in Proceedings of the IEEE international conference on computer vision, 2017, pp. 618–626.

[90] A. SHAMSIAN, A. NAVON, E. FETAYA, AND G. CHECHIK, *Personalized federated learning using hypernetworks*, arXiv preprint arXiv:2103.04628, (2021).

[91] K. SIMONYAN, A. VEDALDI, AND A. ZISSERMAN, *Deep inside convolutional networks: Visualising image classification models and saliency maps*, arXiv preprint arXiv:1312.6034, (2013).

[92] V. SMITH, C.-K. CHIANG, M. SANJABI, AND A. TALWALKAR, *Federated multi-task learning*, arXiv preprint arXiv:1705.10467, (2017).

[93] K. SOHN, H. LEE, AND X. YAN, *Learning structured output representation using deep conditional generative models*, Advances in neural information processing systems, 28 (2015), pp. 3483–3491.

[94] G. STRANG, *Introduction to linear algebra*, (2021).

[95] M. SUNDARARAJAN AND A. NAJMI, *The many shapley values for model explanation*, in International conference on machine learning, PMLR, 2020, pp. 9269–9278.

[96] M. SUNDARARAJAN, A. TALY, AND Q. YAN, *Axiomatic attribution for deep networks*, in International conference on machine learning, PMLR, 2017, pp. 3319–3328.

[97] A. T. SURESH, X. Y. FELIX, S. KUMAR, AND H. B. MCMAHAN, *Distributed mean estimation with limited communication*, in International conference on machine learning, PMLR, 2017, pp. 3329–3337.

[98] 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 (NeurIPS) Thirty-seventh Conference on Neural Information Processing Systems, vol. 2023, 2023, pp. 1–14.

[99] Y. TAN, Y. LIU, G. LONG, J. JIANG, Q. LU, AND C. ZHANG, *Federated learning on non-iid graphs via structural knowledge sharing*, in (AAAI) Annual AAAI Conference on Artificial Intelligence, vol. 37, 2023, pp. 9953–9961.

[100] Y. TAN, G. LONG, L. LIU, T. ZHOU, Q. LU, J. JIANG, AND C. ZHANG, *Fedproto: Federated prototype learning over heterogeneous devices*, in (AAAI) Annual AAAI Conference on Artificial Intelligence, vol. 2022, 2022, pp. 8432–8440.

[101] Y. TAN, G. LONG, J. MA, L. LIU, T. ZHOU, AND J. JIANG, *Federated learning from pre-trained models: A contrastive learning approach*, in (NeurIPS-2022) Thirty-seventh Conference on Neural Information Processing Systems, vol. 35, https://nips.cc/Conferences/2022/ScheduleMultitrack?event=64942, 2022, pp. 19332–19344.

[102] H. TANG, X. LIAN, M. YAN, C. ZHANG, AND J. LIU, *d^2: Decentralized training over decentralized data*, in International Conference on Machine Learning, PMLR, 2018, pp. 4848–4856.

[103] V. TOLPEGIN, S. TRUEX, M. E. GURSOY, AND L. LIU, *Data poisoning attacks against federated learning systems*, in European Symposium on Research in Computer Security, Springer, 2020, pp. 480–501.

[104] R. TOMSETT, D. HARBORNE, S. CHAKRABORTY, P. GURRAM, AND A. PREECE, *Sanity checks for saliency metrics*, in Proceedings of the AAAI conference on artificial intelligence, vol. 34, 2020, pp. 6021–6029.

[105] E. UNION, *General data protection regulation*.

[106] A. VAN DEN OORD, O. VINYALS, ET AL., *Neural discrete representation learning*, Advances in neural information processing systems, 30 (2017).

[107] P. VANHAESEBROUCK, A. BELLET, AND M. TOMMASI, *Decentralized collaborative learning of personalized models over networks*, in Artificial Intelligence and Statistics, PMLR, 2017, pp. 509–517.

[108] H. WANG, M. YUROCHKIN, Y. SUN, D. PAPAILIOPOULOS, AND Y. KHAZAENI, *Federated learning with matched averaging*, arXiv preprint arXiv:2002.06440, (2020).

[109] WEBANK AND S. RE, *Swiss re partners with tencent's webank to research ai use in reinsurance*, Aug 2019.

[110] WIKIPEDIA, *Concept drift*.

[111] WIKIPEDIA, *Human-in-the-loop*.

[112] 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).

[113] J. XU, X. TONG, AND S.-L. HUANG, *Personalized federated learning with feature alignment and classifier collaboration*, arXiv preprint arXiv:2306.11867, (2023).

[114] P. YAN AND G. LONG, *Personalization disentanglement for federated learning*, in 2023 IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2023, pp. 318–323.

[115] P. YAN AND G. LONG, *Client-supervised federated learning: Towards one-model-for-all personalization*, in 2024 IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2024, pp. 1–6.

[116] P. YAN, G. LONG, J. JIANG, AND M. BLUMENSTEIN, *Personalized interpretation on federated learning: A virtual concepts approach*, arXiv preprint arXiv:2406.19631, (2024).

[117] T. YANG, G. ANDREW, H. EICHNER, H. SUN, W. LI, N. KONG, D. RAMAGE, AND F. BEAUFAYS, *Applied federated learning: Improving google keyboard query suggestions*, 2018.

[118] Y. YANG, G. LONG, T. SHEN, J. JIANG, AND M. BLUMENSTEIN, *Dual-personalizing adapter for federated foundation models*, in NeurIPS, vol. 2024, 2024.

[119] M. YE, X. FANG, B. DU, P. C. YUEN, AND D. TAO, *Heterogeneous federated learning: State-of-the-art and research challenges*, 2023.

[120] C. ZHANG, G. LONG, H. GUO, X. FANG, Y. SONG, Z. LIU, G. ZHOU, Z. ZHANG, Y. LIU, AND B. YANG, *Federated adaptation for foundation model-based recommendations*, in IJCAI 2024, 2024.

[121] C. ZHANG, G. LONG, T. ZHOU, P. YAN, Z. ZHANG, C. ZHANG, AND B. YANG, *Dual personalization on federated recommendation*, in Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23, E. Elkind, ed., International Joint Conferences on Artificial Intelligence Organization, 8 2023, pp. 4558–4566.

[122] C. ZHANG, G. LONG, T. ZHOU, Z. ZHANG, P. YAN, AND B. YANG, *When federated recommendation meets cold-start problem: Separating item attributes and user interactions*, in Proceedings of the ACM on Web Conference 2024, 2024, pp. 3632–3642.

[123] X. ZHANG, Y. LI, W. LI, K. GUO, AND Y. SHAO, *Personalized federated learning via variational bayesian inference*, in International Conference on Machine Learning, PMLR, 2022, pp. 26293–26310.

[124] Q. ZHAO AND T. HASTIE, *Causal interpretations of black-box models*, Journal of Business & Economic Statistics, 39 (2021), pp. 272–281.

[125] C. ZHU, Z. XU, M. CHEN, J. KONEČNÝ, A. HARD, AND T. GOLDSTEIN, *Diurnal or nocturnal? federated learning of multi-branch networks from periodically shifting distributions*, in International Conference on Learning Representations, 2021.