Towards Robust, Personalized, and Fairness-Aware Federated Learning

SEN FU

Doctor of Philosophy (Engineering)

Supervisor: Wei Bao
Associate Supervisor: Bing Bing Zhou

A thesis submitted in fulfilment of
the requirements for the degree of
Doctor of Philosophy

School of Computer Science
Faculty of Engineering
The University of Sydney
Australia

5 December 2025

Statement of Originality

This is to certify that to the best of my knowledge, the content of this thesis is my own work. Chapter 3 of this thesis draws some inspiration from my Honours thesis. However, it differs significantly in its core ideas and implementation. For details, please refer to Chapter 3. Apart from this, no part of the thesis has been submitted for any degree or other purposes.

I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

Student Name Sen Fu Signature Date 5 December 2025

Abstract

Federated Learning (FL) enables collaborative model training without sharing raw data, but traditional aggregation methods such as FedAvg overlook data distribution, quality, and fairness. This limitation becomes critical in real-world settings, where client data is highly non-independent and identically distributed (non-i.i.d.), noisy, or weakly labeled. Simply weighting updates by dataset size can amplify low-quality data, harm generalization, and introduce fairness risks, especially in weakly supervised scenarios like Partial Label Learning (PLL), where label ambiguity can be exploited adversarially. This thesis investigates personalized and fairness-aware FL under increasingly realistic data assumptions. First, we develop pFedMo, a personalized FL algorithm that mitigates data heterogeneity by combining contrastive learning with personalized FL. It reduces local bias through a contrastive aggregation score aligned with a central representation model trained on i.i.d. data, and enhances convergence through personalized momentum. Next, to account for weak supervision, we develop pFedPLL, which addresses label ambiguity and correlation interference in non-i.i.d. settings. It preserves local label structure via Label Correlation Isolation and improves prediction accuracy through bi-directional calibration. Finally, to ensure fairness and robustness in weakly supervised FL, we develop FairFedPAPL, a defense framework for federated partial attribute and partial label learning. It detects and mitigates fairness-related attacks by reconstructing representative client data through gradient inversion, allowing effective fairness preservation without compromising privacy. In summary, this thesis presents three algorithms to tackle key challenges in FL, including data heterogeneity, weak supervision, and fairness under non-i.i.d. settings. Extensive theoretical analyses and experiments confirm their effectiveness and practicality in realistic FL environments.

Acknowledgements

I would like to express my sincere gratitude to my supervisors, A/Prof. Wei Bao and A/Prof. Bing Zhou, for their continuous support, guidance, and encouragement throughout my PhD journey. Their expertise and mentorship have been instrumental in shaping my research direction and academic development.

I am also grateful to Prof. Albert Y. Zomaya and Dr. Chuang Hu, for their valuable feedback and constructive suggestions, which have helped improve the quality and depth of my work. I would like to thank my collaborators and colleagues, especially Dr. Zhengjie Yang, whose collaboration and shared efforts made the research process both productive and enjoyable. Working with Zhengjie has not only contributed greatly to my research, but his support also provided great help in my daily life in Sydney, for which I am truly grateful. I gratefully acknowledge the financial support from Faculty of Engineering Research Scholarship, which made it possible for me to undertake my doctoral studies.

This journey would not have been possible without the unwavering support of my family. I am deeply grateful to my parents, Jingliang Fu and Haijing Fu, for their unconditional love and constant encouragement. I would also like to thank my aunt, Haixia Fu, for her continued support, both emotionally and financially. Their belief in me has given me the strength to persevere through every challenge.

Finally, I would like to thank my friends Liming Ge, Xingyi Sheng, Ruoyu Wu, Binghan Wu, Shen Wang, Yichen Guo, Zizhao Wang, whose companionship made this journey more enjoyable and less isolating. The informal discussions, laughs, and shared experiences helped me stay grounded and motivated along the way.

Authorship attribution statement

Chapter 3 of this thesis has been published as "Personalized Federated Learning with Contrastive Momentum" by Sen Fu, Zhengjie Yang, Chuang Hu, and Wei Bao in the IEEE Transactions on Big Data, DOI: 10.1109/TBDATA.2024.3403387, October 2025. I designed the study, mathematically proved the theories, performed the experiments, and wrote the paper.

Chapter 4 of this thesis has been submitted as "When Federated Learning Meets Partial Label Data" by Sen Fu, Zhengjie Yang, Wei Bao, and Albert Y. Zomaya to the IEEE Transactions on Computers, under review. I designed the study, mathematically proved the theories, performed the experiments, and wrote the paper.

Chapter 5 of this thesis has been submitted as "Towards Robust and Fair Partial Label Federated Learning Service" by Sen Fu, Wei Bao, Zhengjie Yang, Xinyi Sheng, Yichen Guo, and Bing Bing Zhou to the IEEE Transactions on Services Computing (TSC), under review. I designed the study, mathematically proved the theories, performed the experiments, and wrote the paper.

In addition to the statements above, in cases where I am not the corresponding author of a published item, permission to include the published material has been granted by the corresponding author.

Student Name Sen Fu Signature Date 5 December 2025

As the supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Supervisor Name Wei Bao Signature Date 5 December 2025

List of Publication

Published papers

Sen Fu, Zhengjie Yang, Chuang Hu and Wei Bao, "Personalized Federated Learning with Contrastive Momentum," in *IEEE Transactions on Big Data (TBD)*, vol. 11, no. 5, pp. 2184-2194, Oct. 2025, doi: 10.1109/TBDATA.2024.3403387.

Papers under review

Sen Fu, Zhengjie Yang, Wei Bao and Albert Y. Zomaya, "When Federated Learning Meets Partial Label Data," in *IEEE Transactions on Computers (TC)*, Under Review.

Sen Fu, Wei Bao, Zhengjie Yang, Xinyi Sheng, Yichen Guo, and Bing Bing Zhou, "Towards Robust and Fair Partial Label Federated Learning Service," in *IEEE Transactions on Services Computing (TSC)*, Under Review.

List of Previous Thesis

Honours Thesis

Sen Fu, “Convergence Analysis on the Federated Learning Algorithm with NAG Approach”. Bachelor of Computer Science and Technology (Advanced) (Honours). The University of Sydney, Sydney, Australia, 2021.

Contents

List of Acronyms

AI	Artificial Intelligence
CC	Cloud Computing
CL	Contrastive Learning
EC	Edge Computing
FL	Federated Learning
PFL	Personalized Federated Learning
PLL	Partial Label Learning
CNN	Convolutional Neural Network
DNN	Deep Neural Network
MLP	Multi-Layer Perceptron
PAPL	Partial Attribute Partial Label
GD	Gradient Descent
GDPR	General Data Protection Regulation
Non-i.i.d	Non-Independent and Identically Distributed
IoT	Internet of Things
KL	Kullback–Leibler Divergence
KD	Knowledge Distillation
TV	Total Variation
GI	Gradient Inversion
MEC	Mobile Edge Computing
ML	Machine Learning
NAG	Nesterov Accelerated Gradient
SGD	Stochastic Gradient Descent

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Motivation

The proliferation of smart devices, Mobile Edge Computing (MEC), and Artificial Intelligence (AI) technologies have stimulated the development of Internet of Things (IoT) and Industry 4.0. It advances all aspects of our modern life, such as smart home [Stojkoska and Trivodaliev 2017], smart health [Solanas et al. 2014], and smart transportation [Zantalis et al. 2019], etc. With the abundance of data generated in IoT devices, transmitting huge data to a centralized server for Machine Learning (ML) activity costs huge communication burdens. Moreover, according to General Data Protection Regulation (GDPR) [Voigt and Von dem Bussche 2017], individual users are sometimes not willing to share their sensitive data. To address above issues, Federated Learning (FL) emerges [McMahan et al. 2017]. It allows workers to collaboratively train a generalized global model by transmitting only the model parameters while keeping the data locally. Each worker generates and preserves its own unique, privacy-sensitive data—such as financial records from different banks or medical data from various hospitals. These institutions are often unwilling or legally restricted from sharing raw data with a centralized server due to privacy concerns, or data protection regulations. FL provides a privacy-preserving alternative by ensuring that the raw data remains on the client side, significantly reducing the risk of data leakage while still enabling the benefits of joint learning. This decentralized approach not only enhances data privacy but also reduces communication costs by transmitting only model updates instead of raw data. In the context of edge computing (EC) [Lim et al. 2020], as depicted in Figure 1.1, it emerges as an ideal platform for the implementation of FL. In the FL framework, an edge server aggregates

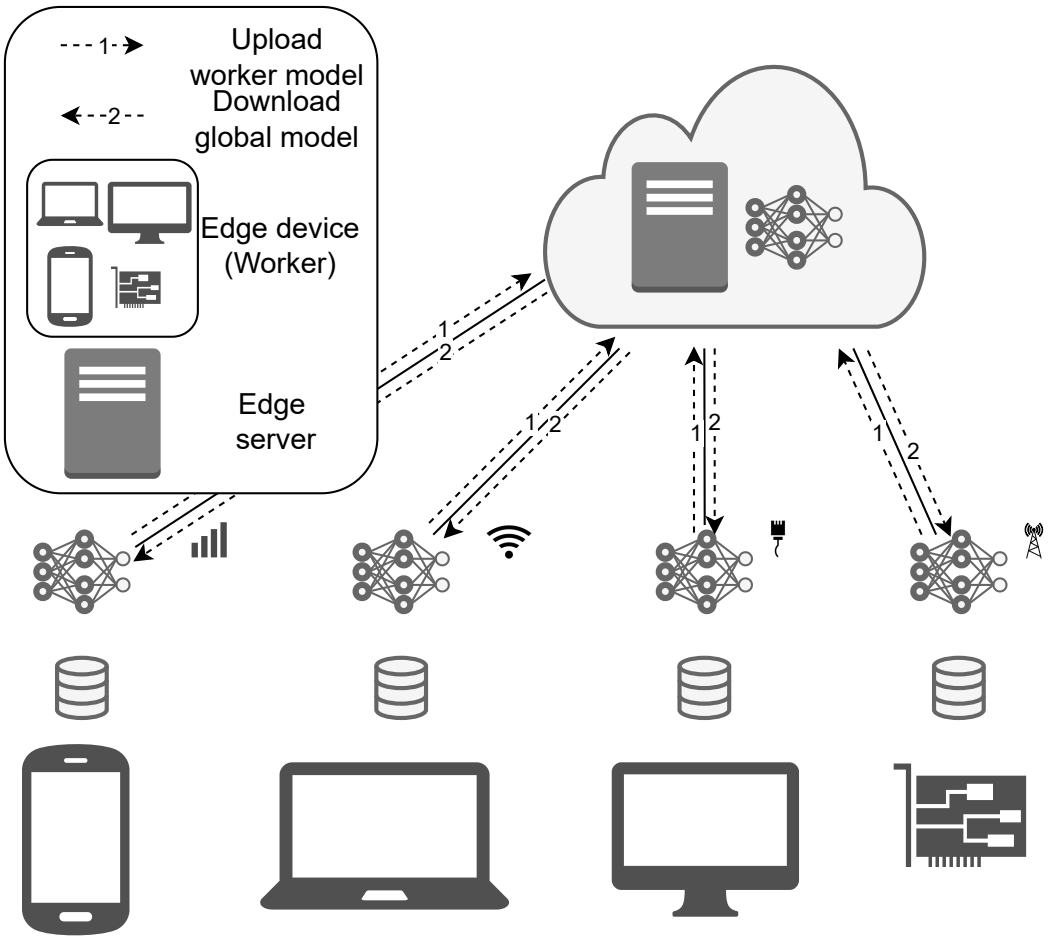

FIGURE 1.1. Federated Learning in Mobile Edge Computing (MEC).

worker/local models from edge devices such as laptops, smartphones, and tablets, etc. to obtain an updated global model. This updated global model is then sent back to the edge devices for the next round of local training. Although FL holds great promise for privacy protection, it also faces several significant challenges. Due to its decentralized nature, FL suffers from issues such as *data heterogeneity* and limited control over *data collection*. Since the training data is provided by participating edge devices (or workers), the data distribution across clients is often non-independent and identically distributed (non-i.i.d.)—varying in both quality and statistical properties. Moreover, the lack of centralized data curation means that the training data may be noisy, weakly labeled, or incomplete, making it difficult to

ensure consistent model performance across the federation. To address the limitations of requiring fully labeled, high-quality data, recent studies have explored weakly supervised learning—particularly Partial Label Learning (PLL). These methods focus primarily on improving predictive performance in terms of accuracy under weak supervision. However, most existing work tends to overlook broader aspects of model ethics, particularly with respect to *fairness* and *robustness*. Specifically, fairness is understood as the extent to which a learning algorithm provides equitable predictive performance across different sensitive groups, without introducing systematic bias or disproportionate disadvantage. These dimensions are crucial in real-world applications where biased or adversarial data may compromise the integrity and equity of FL systems.

1.1.1 Motivation in Personalized Federated Learning

FL faces a major challenge due to data heterogeneity, as each worker's data is typically non-i.i.d. This arises from differences in domain and data distributions across workers—for example, variations in user behavior, demographics, or application contexts. As a result, workers may hold data with imbalanced labels or distinct feature distributions, which creates a mismatch between local and global objectives. In such settings, the standard FedAvg algorithm [McMahan et al. 2017], which aggregates local updates by simple averaging, often struggles to produce a well-generalized global model. The conflicting updates from diverse clients can lead to model divergence, slow convergence, and degraded overall performance. To address this issue, Personalized Federated Learning (PFL) has demonstrated its advantage in addressing the fundamental challenges of FL on heterogeneous (non-i.i.d.) data. PFL analyzes the characteristics of models between different parties (e.g, worker to worker, worker to aggregator) to adapt the global model to better align with individual worker's characteristics. To address the challenge of data heterogeneity in FL, we explore a framework that incorporates Contrastive Learning (CL) to enhance the model's performance under non-i.i.d. conditions. CL is a representation learning technique that pulls similar instances closer in the embedding space while pushing dissimilar ones farther apart, enabling the model to learn more discriminative and generalizable representations. In addition to this,

the optimization methods commonly used in PFL frameworks, such as gradient descent or stochastic gradient descent, often suffer from slow convergence, as they require numerous iterations to reach an optimal solution. To tackle this inefficiency, we integrate a momentum-based optimization strategy, which has shown strong potential in accelerating convergence and mitigating oscillations during training. Specifically, the momentum method updates the model by incorporating a weighted difference between the previous and current model states, scaled by a factor, thus guiding the optimization with accumulated directionality from past updates. Based on the above integrated motivation, in this thesis, we propose pFedMo (Chapter 3) [Fu et al. 2025a], a personalized FL algorithm with contrastive momentum. It significantly enhances the performance under non-i.i.d. data distribution while preserving the efficient momentum acceleration.

1.1.2 Motivation in Personalized Federated Partial Label Learning

Data collection is another problem. In typical FL settings, datasets are typically supervised, meaning each data instance has a corresponding ground truth label. However, this ideal scenario is not reflective of realistic environments in FL, where workers often face significant constraints. Workers may be non-experts, have limited computational capabilities, or operate under conditions where accurate labeling is impractical or prohibitively expensive. Consequently, the quality and reliability of labels collected in such scenarios tend to suffer significantly, leading to noisy or incomplete labeling. Moreover, obtaining fully supervised datasets with accurate labels is often economically and logistically challenging, particularly in edge-device environments. For instance, devices deployed in remote areas or used by casual users may generate substantial amounts of raw data, but the associated labeling processes are inherently error-prone or partial. To address this issue, Partial Label Learning (PLL) provides an appealing solution by associating each data instance with multiple candidate labels, where only one is correct. This significantly simplifies and accelerates data collection while ensuring a sufficiently large quantity of data. Nevertheless, incorporating PLL into FL introduces additional complexities due to the non-i.i.d. nature of the data collected across different workers. Each worker may develop unique label correlations based on their local

environment and dataset characteristics. When these distinct correlations are aggregated at the global level, the resulting interference can distort the overall label correlation knowledge, negatively impacting the accuracy and robustness of the final model. Therefore, it is crucial to develop specialized methods or frameworks within FL that can effectively handle partial labeling under heterogeneous, non-i.i.d. conditions, preserving local label correlation structures without introducing harmful interference during global model aggregation. To this end, we proposed the pFedPLL algorithm [Fu et al. 2024] in Chapter 4, which investigates the PLL setting within a FL environment.

1.1.3 Motivation in Fairness and Robust in Federated Partial Attribute Partial Label Learning

FL research has predominantly concentrated on enhancing model performance, with significantly less attention given to fairness and security concerns. Existing methods in PLL and FedPLL typically prioritize accuracy improvements through advanced loss functions and label disambiguation approaches. However, this performance-centric approach often neglects critical issues such as fairness and robustness against adversarial attacks, particularly in weakly supervised settings. Addressing these overlooked aspects is essential, especially given the vulnerabilities inherent in weakly supervised scenarios. Weak supervision conditions can be exploited by adversaries, potentially compromising the integrity and fairness of the training process. Hence, there is a pressing need for comprehensive investigations into these security and fairness concerns within federated frameworks. The chapter 5 aims to bridge this gap by introducing a novel FL setting, Federated Partial Attribute Partial Label Learning (FedPAPL). FedPAPL generalizes the FedPLL framework by considering ambiguity not only in labels but also in attributes, which are often uncertain or imprecisely annotated in practical data collection scenarios. For example, when collecting survey data, respondents may provide incomplete or ambiguous information about their attributes or labels, leading to inherent uncertainty. By systematically analyzing potential adversarial threats and developing robust defense strategies, this research proposes FairFedPAPL (Chapter 5) [Fu et al. 2025b] to ensure

that federated models trained under weakly supervised conditions remain fair, secure, and robust in realistic environments.

1.2 Thesis Contribution

To effectively tackle the key challenges in FL—including data heterogeneity, difficulties in data collection, and concerns around algorithmic fairness—we propose three novel FL algorithms, each designed to address one or more of these issues. These methods aim to enhance the robustness, representativeness, and fairness of federated models in real-world, non-i.i.d. settings where data is decentralized and sensitive. In the following sections, we provide a detailed yet concise summary of the core ideas and contributions of each proposed algorithm.

1.2.1 Contribution of pFedMo

To mitigate the challenges posed by data heterogeneity (non-i.i.d. distributions) and slow convergence in FL, we propose pFedMo, a personalized federated learning algorithm incorporating contrastive momentum. In pFedMo, we design a personalized score function that enables each client to distill knowledge from the server's momentum-based representation model. This personalization mechanism allows local models to better align with client-specific data distributions, thereby improving convergence and effectively addressing the non-i.i.d. challenge. We have proved that pFedMo is convergent and has an $\mathcal{O}\left(\frac{1}{T}\right)$ convergence rate for smooth non-convex problems for a given T iterations under non-i.i.d. data. We utilize the spirit of contrastive learning in the global aggregation process. In particular, we develop a new method to characterize the similarity/dissimilarity between the workers' models and the representation model, which does not exist in the conventional FL. In the experiment, we compare pFedMo with nine benchmark FL algorithms (FastSlowMo [Yang et al. 2022a], DOMO [Xu and Huang 2022], FedADC [Ozfatura et al. 2021], FedMom [Huo et al. 2020], SlowMo [Wang et al. 2020], FedNAG [Yang et al. 2022b], Mime [Karimireddy et al. 2020a], FedProx [Li et al. 2020], and FedAvg [McMahan et al. 2017]). The experiment is implemented on five

real-world datasets (MNIST [LeCun et al. 1998], CIFAR-10 [Krizhevsky, Hinton et al. 2009], ImageNet [Deng et al. 2009; Moon and Ryffel 2020], and UCI-HAR [Anguita et al. 2013], and Fire-detection-Dataset [Dunnings and Breckon 2020]) with six machine learning models (linear regression, logistic regression, LeNet5 [LeCun et al. 1998], VGG16 [Gross et al. 2021], ResNet18 [Moon and Ryffel 2020], and FireNet [Dunnings and Breckon 2018]). The experimental results illustrate that pFedMo increases the training accuracy by 0.21-35.90% compared to the benchmark algorithms under a wide range of settings. We develop and deploy a real-world FL system. The system aims to assess the overall training duration in real-world scenarios, encompassing communication delays, computing delays both at the worker and server ends, as well as other associated overhead delays. In comparison to benchmark algorithms, our findings illustrate that pFedMo increases the training speed by 1.09-3.64x under a wide range of settings.

The pFedMo is published at IEEE Transactions on Big Data (TBD) under the name "Personalized Federated Learning with Contrastive Momentum."

1.2.2 Contribution of pFedPLL

While pFedMo is studied under the conventional FL setting with fully supervised datasets, real-world scenarios often involve weakly supervised learning, where obtaining precise labels is costly or infeasible. To address this, we propose pFedPLL, a personalized federated partial label learning algorithm designed to operate effectively under such weak supervision. The method introduces two key components to tackle the inherent challenges of ambiguity and heterogeneity in partial label settings. First, in the Label Correlation Isolation (LCI) module, we develop a twin-module architecture in which each client maintains a locally isolated feature-level label correlation matrix. This prevents interference from other clients, allowing personalized and domain-specific label dependencies to be preserved. Second, the Label Correlation Personalization (LCP) module introduces a bi-directional calibration loss to improve label disambiguation. The positive calibration aligns predictions with the latent true label among the candidate set, while the negative calibration explicitly pushes predictions away from incorrect, non-candidate labels. We evaluate our pFedPLL algorithm both theoretically

and experimentally. We prove that pFedPLL converges for smooth non-convex problems at a rate of $O\left(\sqrt{\frac{1}{T}}\right)$ over T global iterations. In the experiments, we compare pFedPLL with mainstream federated PLL algorithms and federated version of centralized PLL algorithms across nine datasets. The ablation study of LCI and LCP is also evaluated. The experiment results demonstrate that pFedPLL consistently outperforms all benchmark algorithms with up to 49.93% accuracy increase in various ML settings.

The pFedPLL has been submitted to IEEE Transactions on Computers (TC) and is currently under review under the title "When Federated Learning Meets Partial Label Data."

1.2.3 Contribution of FairFedPAPL

While both pFedMo and pFedPLL primarily focus on improving model performance, they do not address the broader issue of model ethics, particularly in terms of fairness and robustness. In this context, algorithmic fairness is understood as the extent to which a learning algorithm delivers equitable predictive performance across different sensitive groups, without introducing systematic bias or disproportionate disadvantage. In more realistic data collection scenarios, weak supervision introduces additional vulnerabilities that have been largely overlooked. Specifically, adversaries can exploit ambiguous labels or attributes to launch adversarial attacks that compromise the integrity of the learning process. To bridge this gap, we extend the FedPLL setting to a more general FedPAPL framework. In this setting, each training instance is associated with a candidate label set and a candidate set for the sensitive attribute, with the true label and attribute values hidden within these sets. This formulation more accurately reflects real-world data collection environments, where both labels and attributes are often uncertain, incomplete, or imprecisely annotated. To rigorously evaluate security and fairness in this setting, we design three representative fairness attacks for the FedPAPL framework: *Partial Label Fairness Attack (PLFA)*, *Partial Attribute Fairness Attack (PAFA)*, and a *PAPL Mixup*-based attack. These attacks expose vulnerabilities by manipulating candidate label and attribute sets, providing a comprehensive testbed for defense mechanisms. To address these challenges, we propose **FairFedPAPL**, a novel defense framework built on gradient inversion (GI) techniques. By reconstructing representative

data from client gradients on the server side, FairFedPAPL enables accurate identification of adversarial clients and mitigates fairness violations in the FedPAPL framework. Empirical results demonstrate that FairFedPAPL consistently reduces fairness violations Δ_{EOD} across both i.i.d. and non-i.i.d. settings while maintaining accuracy close to the benign baseline. For example, against the proposed Mixup attack on the COMPAS dataset, Δ_{EOD} decreases from 0.113 to 0.006 under i.i.d. and from 0.124 to 0.002 under non-i.i.d. On the Student dataset, Δ_{EOD} is reduced from 0.041 to 0.004 under i.i.d. and from 0.096 to 0.011 under non-i.i.d. These representative results concretely highlight the robustness and effectiveness of FairFedPAPL.

The FairFedPAPL has been submitted to IEEE Transactions on Services Computing (TSC) and is currently under review under the title "Towards Robust and Fair Partial Label Federated Learning Service."

1.3 Thesis Outline

This thesis is structured around a progressive exploration of personalized and fair FL under increasingly realistic and challenging data assumptions.

Chapter 3 introduces **pFedMo**, a PFL framework designed for the standard supervised setting. While effective, this approach assumes high-quality labeled data, which is often unrealistic in FL scenarios.

To address this gap, Chapter 4 presents **pFedPLL**, which extends personalized FL to the weakly supervised setting by leveraging partial label learning. This chapter reflects a more practical data collection environment where true labels are ambiguous.

Building upon pFedPLL, Chapter 5 shifts the focus from model performance to model ethics, particularly fairness and robustness. It introduces **FairFedPAPL**, a novel defense framework that mitigates fairness attacks in weakly supervised FL through server-side gradient inversion techniques.

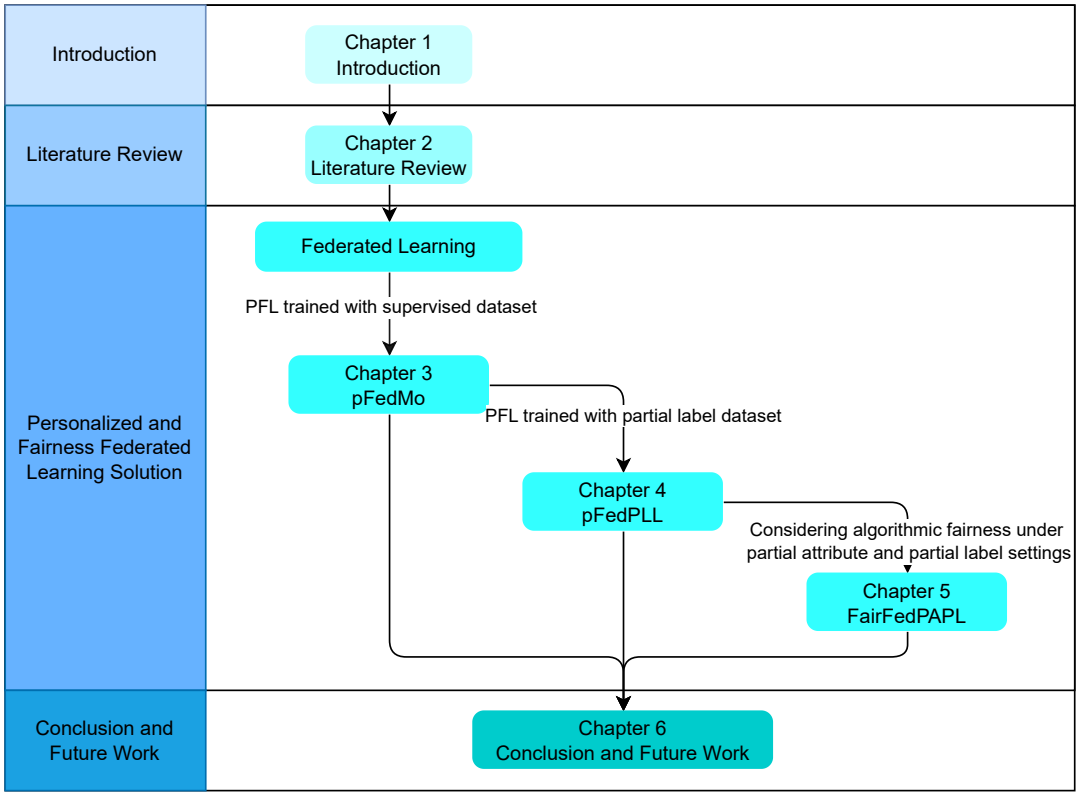

FIGURE 1.2. Thesis structure.

Finally, Chapter 6 concludes the thesis and outlines potential directions for future research.

CHAPTER 2

Literature review

2.1 Federated Learning

Mobile computing and smart devices have become deeply integrated into people's daily lives [Wang et al. 2022]. These devices are no longer just communication tools—they now function as powerful personal computers equipped with built-in sensors capable of generating immense volumes of data. Traditionally, machine learning models are trained in a centralized manner, where users must upload their data to a central server. However, this centralized approach raises significant privacy concerns. Since user data is often subject to privacy regulations [Voigt and Von dem Bussche 2017], transmitting sensitive information to a remote server can result in privacy breaches and violations of legal constraints. Moreover, due to limitations in network bandwidth, transferring such large amounts of data can introduce considerable latency and inefficiency. Fortunately, modern mobile devices are no longer limited to data collection alone—they are now capable of local storage and on-device computation, thanks to advancements in integrated processors that can handle complex computational tasks. Recent studies predict that up to 90% of data can be processed locally on edge devices [Kelly 2015]. To address these challenges, Federated Learning (FL) [McMahan et al. 2017] was introduced as a decentralized learning paradigm that allows models to be trained collaboratively across distributed devices while keeping raw data local. In general, FL can be categorized into two main types: horizontal FL and vertical FL. Horizontal FL (HFL) [Zhang et al. 2022] occurs when datasets across different clients share the same feature space but differ in samples. For example, several hospitals may collect the same types of medical records (e.g., age, diagnosis, treatment) for different groups of patients. Vertical FL (VFL) [Dai et al. 2021], on the other

hand, applies when datasets share the same sample IDs but differ in features. This is common in scenarios where different organizations hold complementary information about the same users—for instance, a bank and an e-commerce platform might both have data about the same set of customers, but with different types of features (e.g., financial vs. purchasing behavior). In this thesis, our focus is on HFL. This setting assumes that all participating clients share the same feature space but hold different data samples. The training process typically proceeds as follows:

- **Step 1: Local Training.** Each client trains a local model on its private data using a shared initial model.
- **Step 2: Uploading Updates.** After local training, each client sends its model updates (e.g., gradients or parameters) to a central server.
- **Step 3: Aggregation.** The central server aggregates the received updates (e.g., using FedAvg) to generate an updated global model.
- **Step 4: Distribution.** The server sends the updated global model back to all clients.
- **Step 5: Iteration.** Steps 1–4 are repeated over multiple communication rounds until the global model converges.

Throughout the entire process, raw data never leaves the clients' devices, preserving user privacy and reducing communication costs.

2.1.1 Personalized Federated Learning

Personalized Federated Learning (PFL) [Tan et al. 2022] is a general concept that facilitates the personalization for individual workers in FL environment. One of the main strategies for PFL is global model personalization [Tan et al. 2022]. It has demonstrated its advantage in addressing the issue of poor convergence on heterogeneous data.

In global model personalization, algorithms first train a single global FL model. This trained global model is then personalized for each worker through a local adaptation step. Such approach aims to train a stronger global model so as to improve the subsequent personalization performance on workers. FedProx [Li et al. 2020] and FedProxVR [Dinh et al. 2020] introduce

a proximal term in the local loss function which helps assess the dissimilarity between global and local models for local model personalization. SCAFFOLD [Karimireddy et al. 2020b] measures the weight divergence between global and local models and incorporates a variance reduction term in the local loss function to correct worker updates. FedHealth [Chen et al. 2020c] adopts a two-step algorithm which first trains a global model and then transfers back to each worker for personalized worker model adaption with its own dataset using Transfer Learning (TL) [Pan and Yang 2010].

In contrast to global model personalization, another approach is to directly build personalized models during the FL model aggregation process. This can be achieved through various techniques. With the help of Knowledge Distillation (KD) [Hinton et al. 2015], FedMD [Li and Wang 2019] starts with a pre-trained model on a public dataset, and then transfers and distills knowledge from this model to enable workers to design personalized models on their own datasets. Different from the traditional FL which learns models for the same task, in MOCHA [Smith et al. 2017], authors introduce a federated Multi-task Learning (MTL) framework where each worker learns a personalized model tailored to its specific task. Federated MTL [Corinzia et al. 2019] aims to learn distinct tasks across different workers by analyzing the relationships between models. Therefore, the model itself is personalized.

In summary, PFL is an effective approach for personalizing the worker models while maintaining the generalization performance of global model. Nevertheless, the optimizer used in these algorithms (gradient descent/stochastic gradient descent) still leads to slow convergence, requiring many iterations to converge.

2.1.2 Momentum in Federated Learning

To accelerate the convergence of FL, momentum-based techniques have emerged as a promising solution [Polyak 1964; Fu 2021]. Unlike standard gradient descent, momentum incorporates an exponentially weighted moving average of past gradients, enabling the algorithm to build velocity in consistent directions and dampen fluctuations in oscillating gradients. In centralized settings, the classical momentum method, introduced by Polyak, updates the

model using the following rules:

$$\boldsymbol{m}(t) = \gamma \boldsymbol{m}(t-1) - \eta \nabla F(\boldsymbol{w}(t-1)), \tag{2.1}$$

$$\boldsymbol{w}(t) = \boldsymbol{w}(t-1) + \boldsymbol{m}(t), \tag{2.2}$$

where $\gamma \in [0, 1)$ is the momentum factor, η is the learning rate, $\boldsymbol{m}(t)$ represents the momentum term initialized as $\mathbf{0}$, and $\boldsymbol{w}(t)$ is the model parameter at iteration t. This approach boosts convergence by accelerating updates in directions with stable gradients while suppressing erratic changes. Building on this idea, Nesterov Accelerated Gradient (NAG) [Nesterov 1983; Ruder 2016] introduces a refined estimate of the gradient by evaluating it at a future position, specifically $\boldsymbol{w}(t-1) + \gamma \boldsymbol{m}(t-1)$, rather than the current point. The NAG updates are given by:

$$\boldsymbol{m}(t) = \gamma \boldsymbol{m}(t-1) - \eta \nabla F(\boldsymbol{w}(t-1) + \gamma \boldsymbol{m}(t-1)), \tag{2.3}$$

$$\boldsymbol{w}(t) = \boldsymbol{w}(t-1) + \boldsymbol{m}(t). \tag{2.4}$$

This modification provides more informed updates, leading to better convergence properties in many practical scenarios. It is worth noting that NAG can also be reformulated using an intermediate variable $\hat{\boldsymbol{w}}(t-1) = \boldsymbol{w}(t-1) + \gamma \boldsymbol{m}(t-1)$, which simplifies the gradient evaluation. This alternate but equivalent formulation [Bubeck 2014; Yan et al. 2018] will be adopted throughout this paper for consistency with our algorithmic framework. Extensive research has been conducted on momentum in FL. Based on the incorporation of momentum, it can be classified into single-momentum algorithms [Yang et al. 2022b; Karimireddy et al. 2020a; Huo et al. 2020; Wang et al. 2020] and double-momentum algorithms [Yang et al. 2022a; Xu and Huang 2022; Ozfatura et al. 2021]. Single-momentum algorithms utilize momentum acceleration either on the worker side or the aggregator side. On the other hand, double-momentum algorithms combine the momenta of both the worker and the aggregator, resulting in superior convergence performance compared to single-momentum algorithms. All of these algorithms strive to train a global model that achieves enhanced performance through the utilization of momentum acceleration. In contrast to prior works that focus on global model optimization, this paper introduces a novel integration of momentum into PFL to effectively mitigate the impact of data heterogeneity and improve convergence speed.

2.1.3 Weakly Supervised Federated Learning

2.1.3.1 Partial Label Learning

Current approaches to PLL disambiguation can be broadly classified into two categories [Tian et al. 2023]: disambiguation through identification methods and disambiguation through averaging methods.

The disambiguation identification method identifies latent labels through heuristic extraction of the true label during training. This method frequently employs the maximum likelihood criterion. Jin and Ghahramani [Jin and Ghahramani 2002] use the Expectation Maximization (EM) algorithm to resolve label ambiguity, while Liu and Dietterich [Liu and Dietterich 2012] develop methods for training mixture models. Additionally, the maximum margin criterion, utilized in methods such as PL-SVM [Nguyen and Caruana 2008] and M3PL [Yu and Zhang 2016], seeks to maximize the distance either between the most probable true label and all others, or between the latent true label and the non-candidate set.

The disambiguation averaging method treats all the labels in the candidate label set equally as the ground-truth label. However, such predictions are vulnerable to the misleading influence of false positive labels. Cour et al. [Cour et al. 2011] address PLL by treating each candidate label equally and differentiating the mean output of candidates from non-candidates. Additionally, Hüllermeier and Beringer [Hüllermeier and Beringer 2005] propose using a voting mechanism among candidate labels of neighboring instances to identify the true label. Zhou and Gu [Zhou and Gu 2018] develop a method for learning a distance matrix to measure proximity between instances based on shared candidate labels. Further advancing this approach, Gong, Yang et al. [Gong et al. 2021] introduce a discriminative metric learning method that uses a Mahalanobis distance metric to assess closeness between similarly labeled neighbors.

Regardless of whether disambiguation occurs through identification methods or averaging methods, both approaches depend on global information, such as kernel functions in maximum margin criteria, label distributions in EM-based algorithms, and distance metrics across all data points. This reliance violates the privacy protection principles inherent in FL settings.

Consequently, traditional PLL methods face significant challenges when directly applied in FL environments.

Recently, an alternative approach to addressing the PLL problem has emerged, utilizing DL based methodologies. Zhang, Feng et al. [Zhang et al. 2021b] propose a Class Activation Value (CAV) that captures learning representation information based on model logit outputs. They treat the class with the maximum CAV as the true label, effectively transforming PLL into a supervised learning problem. However, this method remains susceptible to misclassification by false positive labels, as the CAV cannot guarantee the accuracy of the chosen label. In contrast, Wen, Cui et al. [Wen et al. 2021] introduce a loss function named Leverage Weighted (LW) loss, incorporating a leverage parameter to balance the trade-off between candidate labels and those not in the candidate label set. Similarly, Feng, Lv et al. [Feng et al. 2020] propose CC and RC methods to address PLL in deep networks. The CC method adopts a classifier-consistent approach, while the RC method follows a risk-consistent approach, both serving as loss correction strategies during model updates. These DL-based methodologies have proven effective in solving the PLL problem in a centralized setting. However, their performance remains unknown in an FL setting.

2.1.3.2 Variants of Weakly Supervised Federated Learning

Weakly supervised learning includes Semi-supervised Learning (SSL), Noise Label Learning (NLL), Multi-instance Learning (MIL), and PLL, but studies on PLL in FL are limited. Recently, Yan and Guo 2024 has introduced FedPLL_LAAR, which reduces client drift via adaptive gradient alignment regularization but neglects label correlations. The method assumes a class-dependent generation process [Lv et al. 2020], but in realistic scenarios, partial label datasets should follow an instance-dependent generation process [Xu et al. 2021], where feature-related false labels are more likely to enter the candidate label set, leading to stronger label correlations. Besides PLL, other weakly supervised learning paradigms such as SSL, NLL, and MIL have been more extensively explored in FL. FedMatch [Jeong et al. 2021] in SSL uses labeled data as anchors to improve performance, while PLL faces greater challenges due to uncertain true labels. NLL in FL, studied by [Song et al. 2022], focuses

on mitigating the effect of noisy data, whereas PLL must identify the correct one from a candidate set. FedMIL [Bastola et al. 2024] focuses on bag-level predictions in MIL rather than identifying latent true labels.

Among all types of weakly supervised learning, SSL and PLL are two distinct types that may confuse researchers in the machine learning community. Recent surveys [Jin et al. 2023; Song et al. 2024] provide comprehensive overviews of Federated Semi-supervised Learning (FSSL) and primarily discuss federated learning with unlabeled or partially labeled data, covering semi-supervised, self-supervised, and transfer learning approaches. In FSSL, each instance is either labeled or unlabeled, whereas in FedPLL, each instance is associated with a candidate label set containing the true label but without knowing its exact position. Consequently, FSSL and FedPLL address different forms of supervision uncertainty and their techniques are completely different. For example, recent FSSL works [Bai et al. 2024; Zhang et al. 2024; Liu et al. 2025] mitigate label deficiency and pseudo-label mismatch, but they cannot be directly applied to FedPLL. This is because FedPLL aims to disambiguate the candidate label set and recover the latent true label under distributed non-i.i.d. data so as to finally improve the model performance.

Given the increasing generation of private data from edge devices and the high costs associated with data labeling, it is crucial to explore PLL within an FL framework. Therefore, this paper aims to investigate the PLL problem in FL, synergistically leveraging benefits from DL based PLL methods and FL.

2.2 Algorithm Fairness

The concept of algorithmic fairness emphasizes that a ML model should make decisions impartially, avoiding bias or discrimination against specific demographic groups. While substantial efforts have been made to enhance fairness in ML algorithms through pre-processing [Feldman et al. 2015; Calmon et al. 2017; Grgić-Hlača et al. 2018], in-processing [Kamishima et al. 2012; Zhang et al. 2018; Roh et al. 2021], and post-processing [Lohia et al. 2019; Kim et al.

2019; Putzel and Lee 2022] techniques, these methods are primarily designed for centralized settings and do not directly translate to FL.

2.2.1 Algorithmic Fairness in Federated Learning

To achieve algorithmic fairness in FL, recent studies have proposed either adapting centralized fairness algorithms for use on local clients [Liang et al. 2020; Zeng et al. 2021; Zeng et al. 2023] or introducing additional mechanisms ($e.g.$, constraints-based [Du et al. 2021; Gálvez et al. 2021; Papadaki et al. 2022; Salazar et al. 2023; Hu et al. 2024], reweighting-based [Abay et al. 2020; Ezzeldin et al. 2023], and client selection-based techniques [Zhang et al. 2020a]) at the server side to enhance fairness at the global level. However, these approaches typically assume a mutually trusted and secure FL environment, overlooking the potential threats posed by adversarial clients that can compromise the fairness of the FL model.

2.2.2 Robustness of Algorithmic Fairness

Amid growing concerns about the robustness of algorithmic fairness, several adversarial attacks targeting fairness in both centralized ML [Solans et al. 2020; Celis et al. 2021; Mehrabi et al. 2021b] and FL models [Guo et al. 2024; Sheng et al. 2024] have been proposed. In response, a number of defense mechanisms ($e.g.$, generative adversarial network-based [Guo et al. 2024] and clustering-based [Sheng et al. 2024] approaches) have been developed to counter these threats. Nevertheless, existing studies on fairness robustness predominantly focus on fully supervised scenarios, with little to no attention given to weakly supervised settings. In this work, we pioneer exploring the robustness of algorithmic fairness in a weakly supervised environment, addressing both adversarial attacks and corresponding defenses.

CHAPTER 3

pFedMo: Personalized Federated Learning with Contrastive Momentum

---

Personalized Federated learning is a gener concept that facilitates the personalization for individual workers in an FL environment to address the issue of poor convergence on heterogeneous data among the workers. To achieve this purpose, we utilized the contrastive learning (CL) to develop a new personalized FL framework. CL is a technique in machine learning, which encourages similar instances with short distances while pushing away dissimilar instances with long distances. Momentum has demonstrated its effectiveness in accelerated convergence and simulating oscillation during the training of the model. Based on the above integrated motivation, in this paper, we propose pFedMo, a personalized FL algorithm with contrastive momentum. It significantly enhances the performance under non-i.i.d. data distribution while preserving the efficient momentum acceleration. We have proved that pFedMo is convergent and has an $O(\frac{1}{T})$ convergence rate for smooth non-convex problems for a given T iterations under non-i.i.d. data. The experimental results illustrate that pFedMo increases the training accuracy by 0.21-35.90% compared to nine state-of-the-art (SOTA) benchmark FL algorithms under a wide range of settings. The experiments implemented in the real-world FL system further illustrate that pFedMo increases training speed by 1.09-3.64x compared to benchmarks under a wide range of settings.

The pFedMo is published at IEEE Transactions on Big Data (TBD) under the name "Personalized Federated Learning with Contrastive Momentum."

3.1 Introduction

With the rapid development of Mobile Edge Devices equipped with Mobile Edge Computing (MEC) [Stojkoska and Trivodaliev 2017; Solanas et al. 2014; Zantalis et al. 2019] capabilities, these devices can now efficiently handle complex computational tasks locally. This makes MEC an ideal complement to FL [McMahan et al. 2017], as it enables on-device processing and minimizes the need to transmit sensitive data to centralized servers. However, a significant drawback of FL is data heterogeneity, which arises because data generated by different edge devices often have distinct, non-independent and identically distributed (non-i.i.d.) characteristics. Due to this non-i.i.d. data distribution among devices, the convergence performance of FL algorithms can deteriorate rapidly, especially when dealing with highly heterogeneous data.

Recently, Personalized Federated Learning (PFL) [Tan et al. 2022] has demonstrated its advantage in addressing the fundamental challenges of FL on heterogeneous (non-i.i.d.) data. PFL analyzes the characteristics of models between different parties (e.g., worker to worker, worker to aggregator) or timelines (e.g., current model to previous model) to adapt the global model to better align with individual worker's characteristics. However, since each individual worker model is personalized solely on its own data distribution, it will introduce the workers' bias into the global aggregation phase, resulting in the global model less robust and inefficient [Tan et al. 2022]. To address this issue, we utilize Contrastive Learning (CL) [Chen et al. 2020a; He et al. 2020] to develop a new personalized FL framework. CL is a technique in machine learning, which encourages similar instances with short distance (e.g., reflected by small loss) while pushing away dissimilar instances with long distance. Apart from the training on workers, a representation model is trained at the aggregator using a publicly available i.i.d. dataset accessible only to the aggregator. A score function is designed to contrast the response-based knowledge (i.e., model prediction output) between worker models and the representation model. Since the representation model is trained on i.i.d. data, containing unbiased knowledge, we assign a higher weight to the representation model for worker with higher score, allowing it to contribute more to the training process. In this way,

the unbiased knowledge is gradually distilled from the representation model, so the bias between each personalized worker model is countered.

However, the optimizer used in typical PFL algorithms is still inefficient, because these algorithms usually implement gradient descent or stochastic gradient descent, requiring many iterations to converge. Momentum is proved to be a valuable component for training models. Apart from the conventional gradient descent step, the momentum method conducts an additional momentum step [Ruder 2016] by adding a fraction γ of the difference between past model vector and current model vector of the objective function to accelerate the convergence. Nevertheless, utilizing a single aggregated momentum still causes problem. Since the model is personalized for each worker, such momentum does not suit all personalized worker models well, leading to inefficient momentum acceleration. To solve this problem, we develop personalized (contrastive) momentum which is in pair with the personalized model within each worker for better momentum acceleration. Based on the above integrated motivation, in this paper, we propose pFedMo, a personalized FL algorithm with contrastive momentum. It significantly enhances the performance under non-i.i.d. data distribution while preserving the efficient momentum acceleration.

Theoretically, we prove that pFedMo is convergent and has an $\mathcal{O}\left(\frac{1}{T}\right)$ convergence rate for smooth non-convex problems for a given T iterations. By utilizing CL, we design a new score function to characterize the contrast between the worker models and the representation model, which is then used in the personalization of momentum and model, and the aggregation phase.

In the experiment, we compare the performance of pFedMo with three categories of FL algorithms: ① single-momentum FL (FedMom [Huo et al. 2020], SlowMo [Wang et al. 2020], FedNAG [Yang et al. 2022b], and Mime [Karimireddy et al. 2020a]), ② double-momentum FL (FastSlowMo [Yang et al. 2022a], DOMO [Xu and Huang 2022], and FedADC [Ozfatura et al. 2021]), and ③ no-momentum FL (FedProx [Li et al. 2020],FedAvg [McMahan et al. 2017]). The experiment is implemented on five real-world datasets (MNIST [LeCun et al. 1998], CIFAR-10 [Krizhevsky, Hinton et al. 2009], ImageNet [Moon and Ryffel 2020], UCI-HAI [Anguita et al. 2013], and Fire-detection-Dataset [Dunnings and Breckon 2020]) with six machine learning models (linear regression, logistic regression, LeNet5 [LeCun et al. 1998],

VGG16 [Gross et al. 2021], ResNet18 [Moon and Ryffel 2020], and FireNet [Dunnings and Breckon 2018]). We also conduct experiments on the hyper-parameters that could potentially impact the performance of pFedMo, including the aggregation period τ, momentum factor γ, personalization temperature π, and the number of workers N. The outcomes of these experiments align with our initial theoretical analyses. Furthermore, we develop and deploy a real-world FL system. The system aims to assess the overall training duration in real-world scenarios, encompassing communication delays, computing delays both at the worker and server ends, as well as other associated overhead delays.

3.2 Algorithm Design

3.2.1 Problem Formulation

We consider a conventional FL system where N workers and one aggregator are involved in the FL training process. All workers are expected to participate in the model training, which is within the scope of cross-silo FL [Kairouz et al. 2021]. Each worker, denoted by i, has its own local dataset with the number of data samples denoted by D_i. The total number of dataset for all workers can be then represented by $D = \sum_{i=1}^{N} D_i$. The target of FL is to find the stationary point $\boldsymbol{w}^*$ that minimizes the global loss function

$$\min_{w \in \mathbb{R}^d} F(\boldsymbol{w}) \triangleq \sum_{i=1}^{N} \frac{D_i}{D} F_i(\boldsymbol{w}) \tag{3.1}$$

where d is the dimension of $\boldsymbol{w}$; $F_i(\boldsymbol{w})$ is the local loss function at worker i and $F(\boldsymbol{w})$ is the global loss function at the aggregator. Since workers are distributed in geometric locations among the network, the data generated in each worker is typically influenced by its unique characteristics, local environments, and user preferences. Consequently, the dataset on each worker is heterogeneous and non-i.i.d. [Zhu et al. 2021a]. The key notations are summarized in Table 3.1.

TABLE 3.1. pFedMo: Key Notations

η	learning rate
τ	aggregation period
γ	momentum factor
s_i^k	worker i's score at k's aggregation
π	temperature of score
T	number of total local (worker) iterations indexed by t
K	number of total aggregations indexed by k
N	number of workers indexed by i
$\boldsymbol{m}_i^t$	worker i's momentum at iteration t
$\boldsymbol{w}_i^t$	worker i's model at iteration t
$\boldsymbol{n}_r^t$	representation momentum at iteration t
$\boldsymbol{w}_r^t$	representation model at iteration t
$\boldsymbol{n}_{i+}^t$	worker i's personalized momentum at iteration t
$\boldsymbol{w}_{i+}^t$	worker i's personalized model at iteration t
$\boldsymbol{w}_+^t$	global model at iteration t

3.2.2 pFedMo Algorithm

In this section, we propose a personalized FL with contrastive momentum algorithm, named as pFedMo, to solve formula (3.1). In pFedMo, the momentum acceleration is leveraged at each worker in every local iteration. In the aggregator, a score function is developed to calculate scores for all workers which are then used in the worker momentum and model personalization. The pFedMo algorithm is conduced in T local iterations with K aggregations, where $T = K\tau$. Here, τ represents the aggregation period, indicating that a global aggregation takes place after every τ local iterations.

3.2.2.1 Worker Update

At each local iteration t, each worker computes worker momentum update

$$\boldsymbol{m}_i^t \leftarrow \boldsymbol{w}_i^{t-1} - \eta \nabla F_i(\boldsymbol{w}_i^{t-1}) \tag{3.2}$$

and worker model update

$$\boldsymbol{w}_i^t \leftarrow \boldsymbol{m}_i^t + \gamma(\boldsymbol{m}_i^t - \boldsymbol{m}_i^{t-1}). \tag{3.3}$$

Algorithm 1 pFedMo algorithm

Input: τ, $T = K\tau$, η, γ
Output: Final model parameter $\boldsymbol{w}_+^T$

For each worker, initialize: $\boldsymbol{w}_i^0$ as same value for all i, and $\boldsymbol{m}_i^0 = \boldsymbol{w}_i^0$
For the aggregator, initialize: $\boldsymbol{w}_r^0 = \boldsymbol{w}_i^0$, and $\boldsymbol{m}_r^0 = \boldsymbol{m}_r^0$
for $t = 1, 2, \ldots, T$ **do**
    For each worker $i = 1, \ldots, N$ in parallel:
    Compute worker momentum $\boldsymbol{m}_i^t$ as (3.2)
    Compute worker model $\boldsymbol{w}_i^t$ as (3.3)
    if $t == k\tau$ where $k = 1, \ldots, K$ **then**
        For each worker $i = 1, \ldots, N$ in parallel:
        Send $\boldsymbol{m}_i^{k\tau}$ and $\boldsymbol{w}_i^{k\tau}$ to the aggregator
        For the aggregator:
        Compute scores $\mathcal{S}_k$ as (3.4)–(3.6) and in Algorithm 2
        Personalize each worker i's momentum $\boldsymbol{m}_{i+}^{k\tau}$ as (3.7)
        Personalize each worker i's model $\boldsymbol{w}_{i+}^{k\tau}$ as (3.8)
        Aggregate $\boldsymbol{w}_+^{k\tau} \leftarrow \sum_{i=1}^N \frac{D_i}{D} \boldsymbol{w}_{i+}^{k\tau}$
        //`Aggregate generalized global model`
        Set $\boldsymbol{m}_i^{k\tau} \leftarrow \boldsymbol{m}_{i+}^{k\tau}$ for each worker i
        //`Distribute personalized worker momentum to workers`
        Set $\boldsymbol{w}_i^{k\tau} \leftarrow \boldsymbol{w}_{i+}^{k\tau}$ for each worker i
        //`Distribute personalized worker model to workers`
    end if
end for

Equations (3.2) and (3.3) follow the Nesterov Accelerated Gradient (NAG) [Nesterov 1983] which has demonstrated its faster convergence over Polyak's momentum [Polyak 1964]. The utilization of worker momentum accelerates the worker model training in every local training iteration.

3.2.2.2 Score Function

When $t = k\tau, k = 1, 2, \ldots, K$, the aggregator calculates scores for all workers, denoted as $\mathcal{S}_k = \{s_1^k, s_2^k, \ldots, s_N^k\}$, where $s_i^k \in [0, 1]$ denotes the worker i's score at k's aggregation. The aggregator first trains a representation model based on a public i.i.d. dataset that can only be accessed by the aggregator. Since worker models are trained on non-i.i.d. data while the representation model is trained on i.i.d. data (holding unbiased knowledge), we treat the output of the representation model as the ground truth. For fair comparison, the representation

model is trained following the same update rules (replace subscript i with r in (3.2) and (3.3)) with the same number of iterations (i.e., τ iterations) as in workers.

We consider a dataset with C classes. At k's aggregation, where $k = 1, 2, \ldots, K$, the worker i's model and the representation model make predictions on a batch of B samples extracted from the testing dataset. The loss is calculated as

$$l_b = -\sum_{c=1}^{C} g_{p_{b,c}} \log \frac{\exp\left(q_{b,c}\right)}{\sum_{j=1}^{C} \exp\left(q_{b,j}\right)}, \tag{3.4}$$

where q is the predicted result obtained by worker model, p is the predicted result obtained by representation model, and g_p is the weight which is normalized from p by `softmax` [Zhang et al. 2021a], i.e.,

$$g_{p_{b,c}} = \texttt{softmax}(p_{b,c}) = \frac{\exp\left(p_{b,c}\right)}{\sum_{j=1}^{C} \exp\left(p_{b,j}\right)}. \tag{3.5}$$

Please note that in the Deep Neural Networks (DNN), the predicted result (p or q) is the vector containing the unnormalized logits for each class (with dimension C) obtained from the last fully connected layer before the normalization function (LogSoftmax or Sigmoid, etc.). Finally, we average the values of all $l_b, \forall b \in B$ to obtain the final loss of worker i in k's aggregation,

$$\ell_i^k = \frac{1}{B} \sum_{b=1}^{B} l_b \tag{3.6}$$

The calculation of ℓ_i^k comes from the spirit of Contrastive Learning (CL) [Chen et al. 2020a; He et al. 2020], where a small loss indicates that the worker prediction result is close to the most likely result while away from the least likely result (predicted by the representation model). We assign a high score to the worker with a small loss and vice versa. Please note that different from the typical cross-entropy, where the weight g is given by one-hot encoded information, i.e., the weight is binary (the only one true label is 1, and other false labels are all 0), in pFedMo, the weight is calculated in each global aggregation and is continuous, ranging from 0 to 1. By doing so, the score function measures the disparity between worker models and the representation mode, while the cross-entropy can only quantify the deviation between

Algorithm 2 Calculation of score algorithm

Input: K, $\mathcal{L}_k = \{\ell_1^k, \ell_2^k, \ldots, \ell_N^k\}$
Output: $\mathcal{S}_k = \{s_1^k, s_2^k, \ldots, s_N^k\}$

```
Initialize: u_i = ℓ_i^1 for all workers
for k = 1, 2, ..., K do
    For each worker i = 1, ..., N in parallel:
    if u_i < ℓ_i^k then
        u_i ← ℓ_i^k
    end if
    s_i^k ← 1 − ℓ_i^k / u_i
end for
```

the model's predicted result and the ground truth label. In order to normalize the value of ℓ_i^k to the score s_i^k in the range of $[0, 1]$, we develop Algorithm 2 to calculate scores for all workers when global aggregation occurs. Please note that $\mathcal{S}_k$ plays a vital role in the personalization of worker momentum and model.

3.2.2.3 Aggregator Update

When $t = k\tau, k = 1, 2, \ldots, K$, the aggregator performs aggregator update, including ① computing scores $\mathcal{S}_k$ for all workers, ② personalizing each worker i's momentum $\boldsymbol{m}_{i+}^{k\tau}$, ③ personalizing each worker i's model $\boldsymbol{w}_{i+}^{k\tau}$, and ④ aggregating global model and re-distributing personalized momenta and models to workers. ① has been introduced in Section 3.2.2.2. Then, we use the scores $\mathcal{S}_k$ calculated by ① to operate ② and ③.

In ②, the worker i's personalized momentum $\boldsymbol{m}_{i+}^{k\tau}$ is updated as

$$\boldsymbol{m}_{i+}^{k\tau} \leftarrow (1 - \pi s_i^k) \sum_{i=1}^{N} \frac{D_i}{D} \boldsymbol{m}_i^{k\tau} + \pi s_i^k \boldsymbol{m}_r^{k\tau}, \tag{3.7}$$

where $\boldsymbol{m}_r^{k\tau}$ is the representation momentum at k's aggregation, and $\pi \in [0, 1]$ is the temperature hyper-parameter which determines the sensitivity of impact of s_i^k. The worker with a higher score implies that its local model update direction is more aligned with the update direction of the representation model. Therefore, we assign a high weight (large score) of representation momentum to this worker, allowing it to contribute more to the training process.

In ③, the worker i's personalized model $\boldsymbol{w}_{i+}^{k\tau}$ is update as

$$\boldsymbol{w}_{i+}^{k\tau} \leftarrow (1-\pi s_i^k)\sum_{i=1}^{N}\frac{D_i}{D}\boldsymbol{w}_i^{k\tau} + \pi s_i^k \boldsymbol{w}_r^{k\tau} + \sum_{i=1}^{N}\frac{D_i}{D}\boldsymbol{m}_i^{k\tau} - \boldsymbol{m}_{i+}^{k\tau}, \tag{3.8}$$

where $\boldsymbol{w}_r^{k\tau}$ is the representation model at k's aggregation. Please note that $\sum_{i=1}^{N}\frac{D_i}{D}\boldsymbol{w}_i^{k\tau}$ is the aggregated model without personalization. In conventional FL [McMahan et al. 2017], this aggregated model is used as the global model and distributed back to all workers. However, our approach goes a step further by personalizing each worker's model based on it. In (3.8), we assign a higher weight to the representation model and a lower weight to the original aggregated model for workers with high scores. Since the representation model contains unbiased knowledge (trained on i.i.d. data), the bias between personalized worker models is countered. Additionally, We incorporate a contrastive/personalized momentum term $\sum_{i=1}^{N}\frac{D_i}{D}\boldsymbol{m}_i^{k\tau} - \boldsymbol{m}_{i+}^{k\tau}$ to further accelerate the personalized model update.

Please note it is necessary to personalize not only worker models but also worker momenta (operations ② and ③). This is because although personalized worker models help alleviate the non-i.i.d. problem by distilling knowledge from the representation model, it is also important to personalize the momentum for each worker to better suit each personalized worker model for better momentum acceleration. In this way, pFedMo significantly enhances the performance under non-i.i.d. data distribution while preserving the efficient momentum acceleration.

In ④, the aggregator aggregates the global model $\boldsymbol{w}_{+}^{k\tau}$ (Line 14). Then, pFedMo re-distributes the personalized worker momentum (Line 15) and model (Line 16) to all workers for the next round of training iteration.

3.3 System Model and Preliminaries

3.4 Theoretical Results

3.4.1 Preliminaries

We assume $F_i(\cdot)$ satisfies the following standard conditions that are necessary in theoretical analysis [Wang et al. 2019; Yang et al. 2022b; Dinh et al. 2020].

ASSUMPTION 1. *(ρ-Lipschitz). The local loss function is ρ-Lipschitz.*

$$\|F_i(\boldsymbol{w}_1) - F_i(\boldsymbol{w}_2)\| \leq \rho\|\boldsymbol{w}_1 - \boldsymbol{w}_2\|, \forall \boldsymbol{w}_1, \boldsymbol{w}_2, i.$$

ASSUMPTION 2. *(β-smooth). The local gradient is β-smooth.*

$$\|\nabla F_i(\boldsymbol{w}_1) - \nabla F_i(\boldsymbol{w}_2)\| \leq \beta\|\boldsymbol{w}_1 - \boldsymbol{w}_2\|, \forall \boldsymbol{w}_1, \boldsymbol{w}_2, i.$$

ASSUMPTION 3. *(Bounded diversity). The variance of local gradient to global gradient is bounded.*

$$\|\nabla F_i(\boldsymbol{w}) - \nabla F(\boldsymbol{w})\| \leq \delta_i, \forall \boldsymbol{w}, i.$$

By applying Triangle Inequality for $F_i(\cdot)$ in Assumptions 1 and 2, we obtain that global loss function $F(\cdot)$ also satisfies ρ-Lipschitz and β-smooth. For Assumption 3, we define $\delta \triangleq \sum_{i=1}^{N} \delta_i$.

3.4.2 Virtual Update

In Algorithm 1, in order to index the global aggregation, we introduce a concept "Interval", denoted by $[k]$, to divide the total T local iterations into K intervals ($T = K\tau$), i.e., interval $[k]$ contains τ local iterations ($t \in [(k-1)\tau, k\tau]$) and one global aggregation (k's aggregation). Inspired by [Wang et al. 2019], we introduce the concept "Virtual Update" as if the momentum and model are updated on the total training dataset D (equivalent to virtually centralized

update). The rationale behind the virtual update is that the diversity of gradients among workers makes it challenging to directly bound the real update. Nevertheless, we can first bound the gap between real update and the virtual update, and then analyze the convergence of the virtual update so as to obtain the convergence of the real update. This becomes the road-map of the convergence analysis which will be discussed in Section 3.4.3.

At the beginning of the interval $[k]$ when $t = (k-1)\tau$, we set the initial values for virtual update

$$\boldsymbol{m}_{[k]}^{(k-1)\tau} \leftarrow \boldsymbol{m}_{+}^{(k-1)\tau}, \tag{3.9}$$

$$\boldsymbol{w}_{[k]}^{(k-1)\tau} \leftarrow \boldsymbol{w}_{+}^{(k-1)\tau}. \tag{3.10}$$

Then, it will be conducted for τ time when $t \in ((k-1)\tau, k\tau]$ as

$$\boldsymbol{m}_{[k]}^{t} \leftarrow \boldsymbol{w}_{[k]}^{t-1} - \eta \nabla F(\boldsymbol{w}_{[k]}^{t-1}), \tag{3.11}$$

$$\boldsymbol{w}_{[k]}^{t} \leftarrow \boldsymbol{m}_{[k]}^{t} + \gamma(\boldsymbol{m}_{[k]}^{t} - \boldsymbol{m}_{[k]}^{t-1}). \tag{3.12}$$

We also define aggregated values $\boldsymbol{m}^t = \sum_{i=1}^{N} \frac{D_i}{D} \boldsymbol{m}_i^t, \boldsymbol{w}^t = \sum_{i=1}^{N} \frac{D_i}{D} \boldsymbol{w}_i^t, \forall t$, and $\boldsymbol{m}_{+}^t = \sum_{i=1}^{N} \frac{D_i}{D} \boldsymbol{m}_{i+}^t, \boldsymbol{w}_{+}^t = \sum_{i=1}^{N} \frac{D_i}{D} \boldsymbol{w}_{i+}^t, \forall t = k\tau, k = 1, 2, \ldots, K$ that will be used in convergence analysis for convenient presentation.

3.4.3 Convergence Analysis

Following the rationale of virtual update, we first bound the gap between $\boldsymbol{w}^t$ and virtual update $\boldsymbol{w}_{[k]}^t$ in Theorem 1.

THEOREM 1. *For any interval* $[k]$, $\forall t \in ((k-1)\tau, k\tau]$, *we have*

$$\|\boldsymbol{w}^t - \boldsymbol{w}_{[k]}^t\| \leq f(t - (k-1)\tau), \tag{3.13}$$

where $f(z)$ is

$$f(z) = \eta\delta\left(C_3(\gamma C_1)^z + C_4(\gamma C_2)^z - \frac{1}{\eta\beta} - \frac{\gamma^2(\gamma^z - 1) - (\gamma - 1)z}{(\gamma - 1)^2}\right), \tag{3.14}$$

and C_1–C_4 are constants defined in Appendix A1 , $\forall\gamma \in (0, 1), z = 1, 2, \ldots$.

PROOF. See Appendix A1 for the complete proof. □

Pleae note that when $t = (k - 1)\tau, \forall[k]$, we have $\|\boldsymbol{w}_+^t - \boldsymbol{w}_{[k]}^t\| = 0 = h(0)$, which also satisfises (3.14). We note that $F(\boldsymbol{w})$ is ρ-Lipschitz, so we also have

$$F(\boldsymbol{w}^t) - F(\boldsymbol{w}_{[k]}^t) \leq \rho f(t - (k - 1)\tau). \tag{3.15}$$

We then bound the gap between $\boldsymbol{w}_+^t$ and $\boldsymbol{w}^t$ in Theorem 2.

THEOREM 2. *For any interval $[k]$, suppose $\gamma \in (0, 1), s_i^k \in [0, 1]$, and $\pi \in [0, 1]$, we have*

$$\|\boldsymbol{w}_+^{k\tau} - \boldsymbol{w}^{k\tau}\| \leq 2\mu\eta\rho q^k\pi, \tag{3.16}$$

where we define $q^k \triangleq \sum_{i=1}^N \frac{D_i}{D} s_i^k$ as the weighted average of all workers' scores at k's aggregation, and constant μ is defined in Appendix A2.

PROOF. See Appendix A2 for the complete proof. □

THEOREM 3. *Suppose (1) $\beta\eta(\gamma + 1) \in (0, 1]$, $\gamma \in (0, 1)$, and $\forall\tau = 1, 2, \ldots$; (2) $\omega\alpha\sigma^2 - \frac{\rho(f(\tau)+2\mu\eta\rho\varphi\pi)}{\tau\varepsilon^2} > 0$; (3) $F(\boldsymbol{w}_{[k]}^{k\tau}) - F(\boldsymbol{w}^*) \geq \varepsilon, \forall k$; and (4) $F(\boldsymbol{w}_+^T) - F(\boldsymbol{w}^*) \geq \varepsilon$ are satisfied. $\exists\varepsilon > 0$, the upper bound of Algorithm 1 is given by*

$$F(\boldsymbol{w}_+^T) - F(\boldsymbol{w}^*) \leq \frac{1}{T\left(\omega\alpha\sigma^2 - \frac{\rho(f(\tau)+2\mu\eta\rho\varphi\pi)}{\tau\varepsilon^2}\right)}. \tag{3.17}$$

We define $F(\boldsymbol{w}^*)$ as the minimum value, if there exists some $\zeta > 0$ such that $F(\boldsymbol{w}^*) \leq F(\boldsymbol{w})$ for all w within distance ζ of $\boldsymbol{w}^*$. Constants ω, σ, φ and α are defined in Appendix A3.

PROOF. See Appendix A3 for the complete proof. □

Theorem 3 demonstrates that Algorithm 1 converges with the convergence rate $\mathcal{O}\left(\frac{1}{T}\right)$ for smooth non-convex problems under non-i.i.d. data distribution. The overall gap $F(\boldsymbol{x}_g^T) - F(\boldsymbol{x}^*)$ decreases when T is larger. From [Yang et al. 2022b, Appendix C], we have $f(z) \geq 0$ for any $z = 1, 2, \ldots$, and it increases with z. Therefore, the value of $\frac{\rho(f(\tau)+2\mu\eta\rho\varphi\pi)}{\tau\varepsilon^2}$ increases with τ so as to increase the overall bound. However, in order to let the condition 2 in Theorem 3 hold, we cannot set a very large τ, implying that convergence is guaranteed when $f(\tau)$ is below a certain threshold. Meanwhile, condition 2 in Theorem 3 holds when ε is above a positive certain threshold defined as ε_0. When ε is small and $\varepsilon > \varepsilon_0$, i.e., when the loss function is close to the stationary point, the value of $\omega\alpha\sigma^2 - \frac{\rho(f(\tau)+2\mu\eta\rho\varphi\pi)}{\tau\varepsilon^2}$ diminishes, resulting in an increased overall bound. In this case, a larger T is required to achieve convergence towards a small ε. Conversely, the algorithm requires fewer iterations (smaller T) to only converge to a large ε. This illustrates the trade-off between the number of training iterations and the tightness of the bound.

3.5 Experiments

In this section, we evaluate the convergence performance of pFedMo with various mainstream momentum-based algorithms, including FastSlowMo [Yang et al. 2022a], DOMO [Xu and Huang 2022], FedADC [Ozfatura et al. 2021], FedMom [Huo et al. 2020], SlowMo [Wang et al. 2020], FedNAG [Yang et al. 2022b], Mime [Karimireddy et al. 2020a], FedProx [Li et al. 2020], and classical FL algorithm FedAvg [McMahan et al. 2017]. We compare pFedMo with benchmarks in two different environments, considering two aspects. ① Given the same total training iterations T, we compare the accuracy in an GPU tower server. ② Given an accuracy goal, e.g., 85%, we compare the total training time in a real-world IoT system. We then evaluate the effects of hyper-parameters such as τ, γ, π, and N. Afterwards, we manually establish three distinct non-i.i.d. data distribution scenarios to evaluate the effects of non-i.i.d. data. 1) We assign a subset of classes out of total classes of dataset to each worker, referring to as "Label-Skew non-i.i.d.". 2) We assign a distinct quantity of data samples to each worker,

referring to as "Quantity-Skew non-i.i.d.". 3) We utilize Dirichlet distribution [Minka 2000] to combine 1) and 2), letting Label-Skew and Quantity-Skew occur simultaneously.

TABLE 3.2. pFedMo: Performance comparison of different FL algorithms (accuracy %, ± Std).

	Linear on MNIST	Logistic on MNIST	LeNet5 on MNIST	LeNet5 on CIFAR10
pFedMo	**86.00** ± 0.06	**89.36** ± 0.05	**97.23** ± 0.03	**64.49** ± 0.11
FastSlowMo [Yang et al. 2022a]	85.79 ± 0.05	89.02 ± 0.05	95.90 ± 0.05	59.39 ± 0.07
DOMO [Xu and Huang 2022]	83.74 ± 0.06	88.83 ± 0.05	95.59 ± 0.07	63.49 ± 0.09
FedADC [Ozfatura et al. 2021]	85.51 ± 0.04	88.18 ± 0.05	95.09 ± 0.07	56.00 ± 0.11
FedMom [Huo et al. 2020]	84.84 ± 0.06	88.05 ± 0.05	94.74 ± 0.05	54.87 ± 0.07
SlowMo [Wang et al. 2020]	84.82 ± 0.06	88.00 ± 0.06	94.88 ± 0.05	54.43 ± 0.06
FedNAG [Yang et al. 2022b]	84.97 ± 0.04	88.14 ± 0.05	95.04 ± 0.06	55.54 ± 0.09
Mime [Karimireddy et al. 2020a]	84.41 ± 0.06	87.73 ± 0.06	93.89 ± 0.08	48.24 ± 0.15
FedProx [Li et al. 2020]	85.49 ± 0.05	86.85 ± 0.05	93.78 ± 0.06	46.73 ± 0.12
FedAvg [McMahan et al. 2017]	83.57 ± 0.04	86.89 ± 0.05	93.31 ± 0.08	37.79 ± 0.19

	VGG16 on CIFAR10	ResNet18 on ImageNet	LeNet5 on UCI-HAR
pFedMo	**89.88** ± 0.10	**68.42** ± 0.09	**89.21** ± 0.09
FastSlowMo [Yang et al. 2022a]	88.53 ± 0.09	67.05 ± 0.10	88.15 ± 0.06
DOMO [Xu and Huang 2022]	89.16 ± 0.10	67.58 ± 0.15	86.05 ± 0.12
FedADC [Ozfatura et al. 2021]	89.38 ± 0.08	67.76 ± 0.12	85.14 ± 0.09
FedMom [Huo et al. 2020]	88.03 ± 0.10	66.91 ± 0.11	84.69 ± 0.07
SlowMo [Wang et al. 2020]	88.47 ± 0.09	66.84 ± 0.09	83.03 ± 0.10
FedNAG [Yang et al. 2022b]	88.33 ± 0.06	66.81 ± 0.14	84.69 ± 0.06
Mime [Karimireddy et al. 2020a]	81.76 ± 0.11	64.33 ± 0.21	76.75 ± 0.11
FedProx [Li et al. 2020]	87.43 ± 0.05	66.64 ± 0.10	56.95 ± 0.07
FedAvg [McMahan et al. 2017]	88.27 ± 0.15	66.59 ± 0.09	53.31 ± 0.12

3.5.1 Experiment on Convergence of pFedMo

3.5.1.1 Experimental Setup

We use a GPU tower server with 4 NVIDIA GeForce RTX 2080Ti GPUs to test the convergence performance. Two machine learing tasks are conducted base on four real-world datasets. ① Image classification: MNIST [LeCun et al. 1998], CIFAR-10 [Krizhevsky, Hinton et al. 2009], and ImageNet [Deng et al. 2009; Moon and Ryffel 2020]. ② Human activity recognition: UCI-HAR [Anguita et al. 2013]. For the MNIST dataset, it contains 60,000 training images, 10,000 test images for 10 different class. The CIFAR-10 dataset contains

50,000 images for training and 10,000 images for test over 10 classes. The images in MNIST are it The images in CIFAR-10 are $32 \times 32 \times 3$ dimensional. The ImageNet dataset is a tiny version and is more challenging for training the model for image classification tasks, as it contains 200 different classes. For each class, it has 500 images. Overall, there is 100,000 image for training and 10,000 images for testing. The images in Tiny-ImageNet are $64 \times 64 \times 3$ dimensional. The UCI-HAR dataset comprises 10,299 instances characterized by 561 attributes, distributed across six distinct activities. We use five models including Linear Regression, Logistic Regression, LeNet5, VGG16, and ResNet18. The first two models are convex models which are widely used in segmentation task. LeNet5 is a classic CNN model [LeCun et al. 1998]. VGG16 and ResNet18 are DNN models with the structure defined in [Gross et al. 2021; Moon and Ryffel 2020] respectively. All training and testing data are randomly shuffled and distributed among workers. We use mini-batch with size 64 in all experiments. The learning rate η is set to 0.01. The specific hyper-parameters setting will be defined in each experiment accordingly.

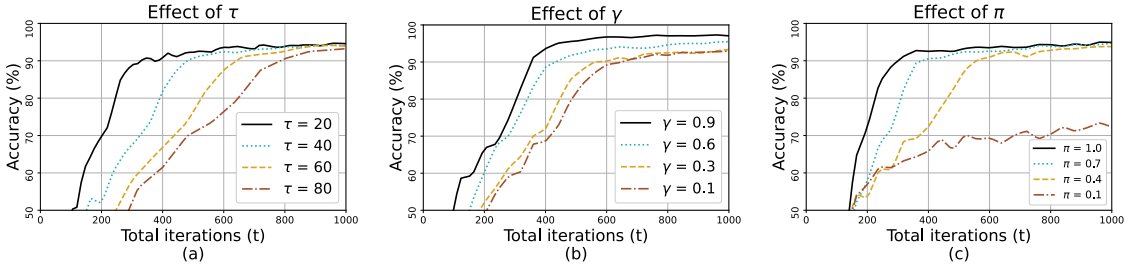

FIGURE 3.1. Accuracy comparison for pFedMo under different settings of τ, γ and π.

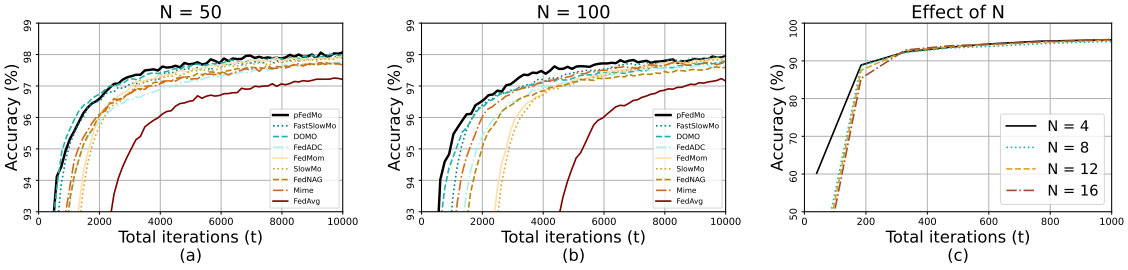

FIGURE 3.2. (a) and (b): Accuracy comparison for pFedMo and benchmarks for more workers ($N = 50$ and $N = 100$). (c) Accuracy comparison for pFedMo with different number of workers ($N = 4, 8, 12, 16$)

3.5.1.2 Performance Comparison

In Table 3.2, we compare the convergence performance of pFedMo with benchmark algorithms based on various models and datasets. The values in the table illustrate the training accuracy for different algorithms under a given T training iterations. The higher accuracy indicates the better performance. We set $T = 1000$ (MNIST), $T = 4000$ (UCI-HAR), and $T = 10000$ (CIFAR-10 and ImageNet), $\tau = 20$ (convex models) and $\tau = 40$ (non-convex models), $\gamma = 0.5$, $N = 4$, and $\pi = 1$. We categorize the benchmarks into three categories: ① double-momentum implemented on both workers and the aggregator (FastSlowMo, DOMO, and FedADC), ② single-momentum implemented on either worker or the aggregator (FedMom, SlowMo, FedNAG, and Mime), and ③ no-momentum (FedProx and FedAvg). For convenient presentation, we use "$>$" to indicate "is better than".

First, we observe that pFedMo $>$ all benchmarks. For convex models, pFedMo increases the training accuracy by 0.21–2.47%. For non-convex models, pFedMo achieves the accuracy increase by 1.00–35.90% (CNN) and 0.50–8.12% (DNN), respectively. This demonstrates that applying personalized momentum on both workers and the aggregator as well as the personalized model achieves the best performance compared to the algorithms without personalization. It is very useful to evaluate performance by considering various factors such as the diversity of models and datasets, as well as a wide range of SOTA FL algorithms, because performance varies based on these factors. Compared to five models, four datasets, and nine SOTA benchmarks, the result in Table 3.2 illustrates that pFedMo achieves a sufficient and substantial performance improvement.

Second, we observe that ① $>$ ② $>$ ③, with 0.04–15.25% accuracy increase from ② to ①, and 0.11-31.38% accuracy increase from ③ to ②. This verifies that the momentum accelerates the convergence compared to FedProx and FedAvg without momentum. Specifically, double-momentum algorithms outperform single-momentum algorithms.

Third, we observe that FedProx outperforms FedAvg in most cases, which illustrates that the model personalization accelerates the convergence.

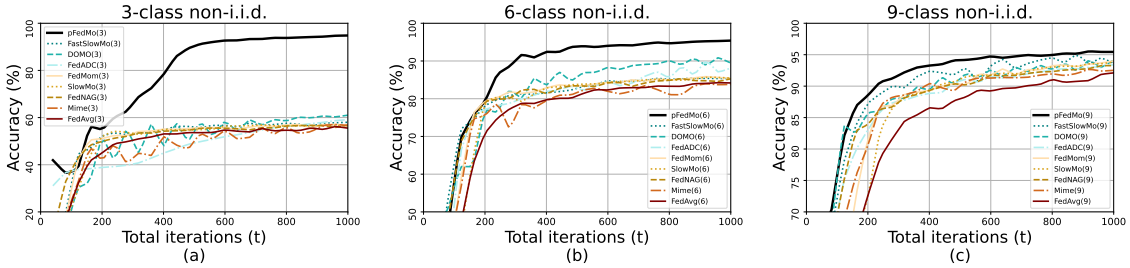

FIGURE 3.3. Accuracy comparison for pFedMo under 3-class (a), 6-class (b), and 9-class (c) Label-Skew non-i.i.d. data.

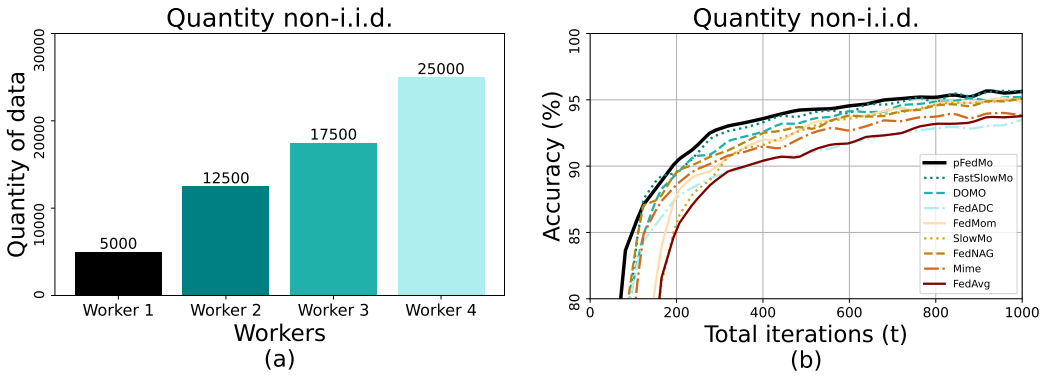

FIGURE 3.4. Accuracy comparison of pFedMo with other benchmarks under Quantity-Skew non-i.i.d data.

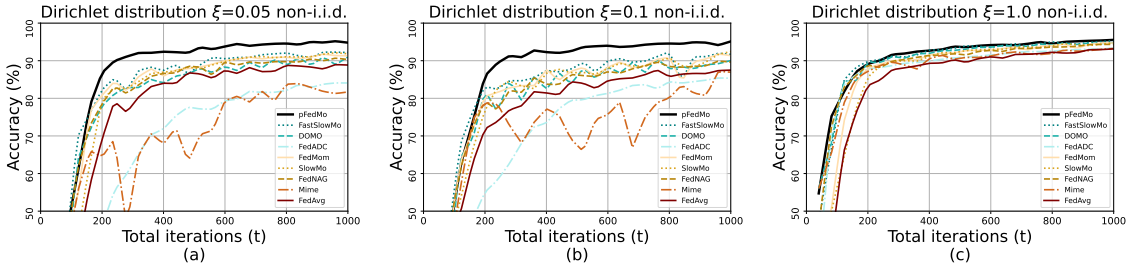

FIGURE 3.5. Accuracy comparison for pFedMo under Dirichlet distribution $\xi = 0.05$ (a), $\xi = 0.1$ (b), and $\xi = 1.0$ (c).

3.5.1.3 Effects of Hyper-parameters

To understand how different hyper-parameters affect the convergence performance of pFedMo, we conduct several experiments using LeNet5 on MNIST dataset to analyze the effects of aggregation period τ, momentum factor γ, and personalization temperature π. Only 3 classes of data (*3-class non-i.i.d.*) are distributed among workers.

Effects of τ: In Figure 3.1(a), we evaluate the effects of τ. We set $T = 1000, \gamma = 0.5, \pi = 0.5, N = 4$. We observe that large τ decreases the training performance especially at the early stage of the training. This verifies the result of Theorem 1. The large τ increases the value of $f(\cdot)$, and thus increases the overall bound. As a result, the training performance is decreased.

Effects of γ: In Figure 3.1(b), we evaluate the effects of γ. We set $T = 1000, \tau = 40, \pi = 0.5, N = 4$. We observe that large γ increases the training performance. This matches our expectation. Momentum accelerates the convergence by comparing the difference between past and current model vectors. If we set the large momentum factor γ (but should < 1), the acceleration performance further improves. Such observation is also consistent with those in [Yang et al. 2022b; Yang et al. 2022a].

Effects of π: In Figure 3.1(c), we evaluate the effects of π. We set $T = 1000, \tau = 40, \gamma = 0.5, N = 4$. We observe that large π increases the training performance. From (3.7) and (3.8), we can see that large π increases the weight (contribution) of the representation momentum and model at workers. That is to say, large π accelerates the distillation of knowledge from the representation model to worker models, thereby enhancing the training performance.

Effects of N: To emulate the cross-silo FL [Kairouz et al. 2021] (typically up to 100 participants), we compare the training accuracy when more workers are involved in the training ($N = 50$ and $N = 100$). In Figure 3.2(a) and (b), although large N decreases the performance for all algorithms (Large N causes more data divergence among workers.), we can still observe that pFedMo achieves the highest accuracy and the results show the same trend as in Table 3.2. In Figure 3.2 (c), we evaluate the effects of the number of workers N. We set $T = 1000, \tau = 40, \gamma = 0.5, N = 4, 8, 12, 16$. We observe the large N will cause a decline in the convergence performance, especially in the early stage of training. This follows our expectation because more workers cause more divergence among the workers and thus decrease convergence performance. After a sufficient number of training iterations, we can still observe that the accuracy of more worker cases will be close to the few worker cases when they finally converge.

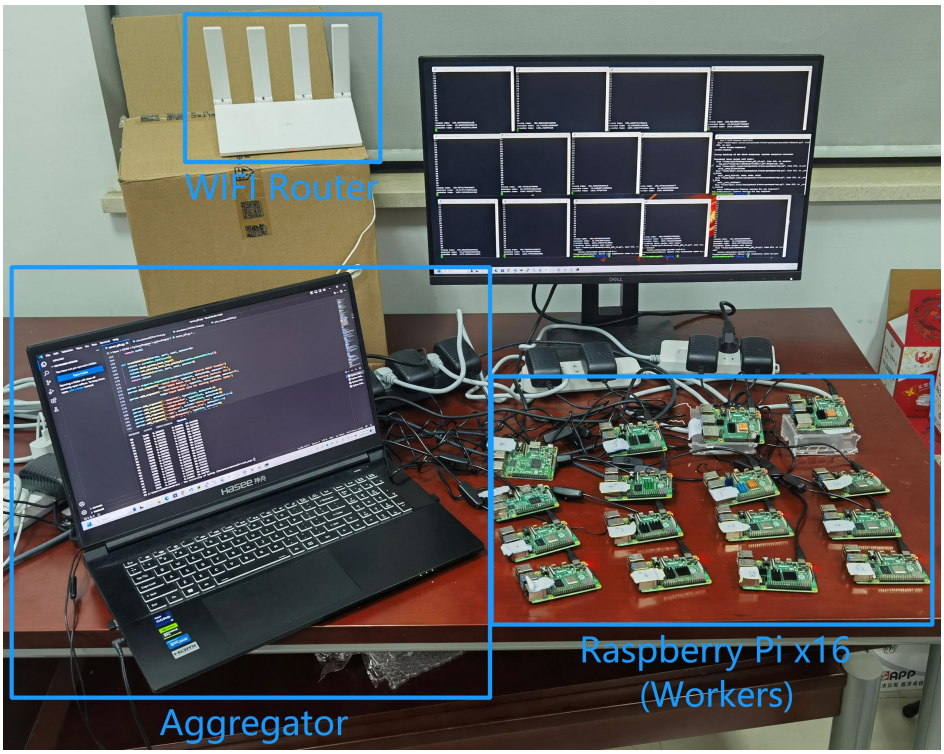

FIGURE 3.6. System Architecture of Real-world IoT system in the experiment.

3.5.1.4 Effects of Non-i.i.d. Data

To analyze how non-i.i.d. data affects the convergence performance of pFedMo and benchmarks, we conduct several experiments using LeNet5 on MNIST dataset based on three non-i.i.d. data distribution scenarios (Label-Skew, Quantity-Skew, and combination of them). The setting is $T = 1000, \gamma = 0.5, \pi = 0.5, N = 4$.

For Label-Skew non-i.i.d., we randomly assign a subset of classes, selected from the total 10 classes in MNIST dataset. For fair comparison, each worker is allocated with the same number of data samples. Smaller x represents a higher level of Label-Skew non-i.i.d. setting. We use *3-class non-i.i.d.*, *6-class non-i.i.d.*, and *9-class non-i.i.d.* to represent *high*, *middle* and *low* level of Label-Skew non-i.i.d. data respectively. In Figure 3.3, we observe that pFedMo $>$ DOMO $\approx$ FastSlowMo $>$ FedADC $>$ FedNAG $>$ FedMom $>$ SlowMo $>$ Mime $\approx$ FedAvg, which is consistent with the result in Table 3.2. The pFedMo algorithm achieves at least 34.26%, 5.94%, and 1.28% accuracy increase for *high*, *middle*, and *low* level of Label-Skew

non-i.i.d. setting respectively. This demonstrates that pFedMo outperforms benchmarks under any level of Label-Skew non-i.i.d. data distribution. We also observe that pFedMo is more robust to different levels of Label-Skew non-i.i.d. data with only 0.17% drop from *low* to *high*, while the performance of benchmarks declines fast (33.15–37.45% drop from *low* to *high*).

For Quantity-Skew non-i.i.d., we assign an exact number of data samples to workers as shown in Figure 3.4(a) (Worker 1: 5000, Worker 2: 12500, Worker 3: 17500, and Worker 4: 25000). For fair comparison, each worker contains all 10 classes with each class containing the same portion of data samples. Please note that the typical aggregation method in FL [McMahan et al. 2017] employs the weighted average, i.e., the weight for each worker model is proportional to the number of its data samples. Nevertheless, in order to evaluate the effect of Quantity-Skew non-i.i.d., we assign the same weight (0.25) for all four workers specifically in this experiment. In Figure 3.4(b), we still observe that pFedMo outperforms benchmarks, with the 95.71% accuracy.

We further conduct a more complicated case where Label-Skew and Quantity-Skew non-i.i.d. occur at the same time. To achieve this, we employ the Dirichlet distribution [Minka 2000], denoted as $Dir(\xi)$, ensuring variation in both classes and data sample quantities among workers. $\xi(> 0)$ is a hyper-parameter that controls the level of non-i.i.d., where a small ξ indicates a higher level of heterogeneous (non-i.i.d.) data. In this experiment, we set $\xi = \{0.05, 0.1, 1\}$ to simulate *high*, *middle*, and *low* levels of non-i.i.d. data distribution respectively[1]. In Figure 3.5, we observe the same results as shown in Figure 3.3 and 3.4. The pFedMo algorithm achieves 95.54%, 95.13%, and 94.8% accuracy in the *low*, *middle*, and *high* level of non-i.i.d. settings, demonstrating that pFedMo is more robust compared to benchmarks, evidenced by a mere 0.74% accuracy drop from *low* to *high*.

[1]In practice, setting $\xi \geq 1$ generates an approximate i.i.d. data distribution (low level of non-i.i.d.), which has widely been used in the literature [Zhu et al. 2021b; Lin et al. 2020]. Please refer to [MacKay n.d.] for more details.

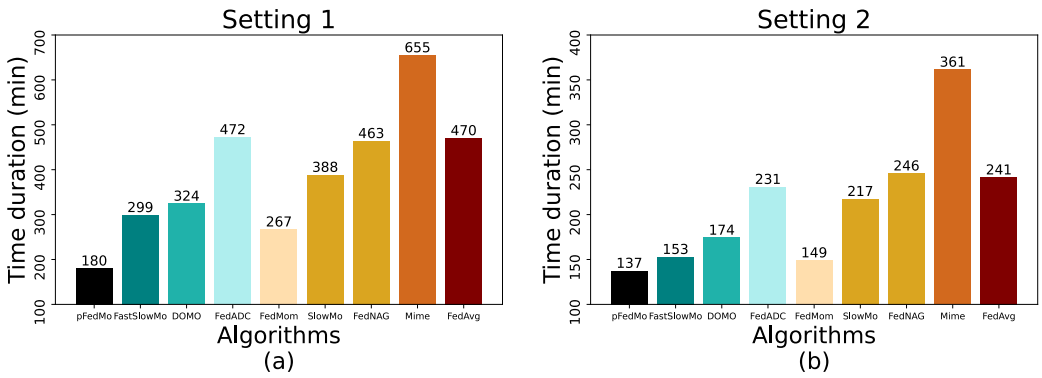

FIGURE 3.7. Comparison of total training time to reach 85% accuracy under two different settings for fire detection application. The time is labeled above each bar. (a): $\gamma = 0.5, \pi = 1, \tau = 20$. (b): $\gamma = 0.5, \pi = 1, \tau = 40$.

3.5.2 Experiment on Real-world IoT System

To evaluate the performance of pFedMo and benchmarks in a more realistic environment, we build up a real-world IoT Edge Computing system which runs fire detection application. We use 16 Raspberry Pi 4 Model B as workers and one laptop (HASEE G9R9 with Intel Core i9-13900H CPU) as the aggregator. Workers and the aggregator are connected to HUAWEI WS5200 router with 2.4GHz WIFI and 1Gbps Ethernet wired cable, respectively. The IoT system architecture is illustrated in Figure 3.6. The total training time is calculated from the moment the aggregator initiates the training process for all workers until the aggregator determines that the training accuracy has reached the specified goal (e.g., 85%). The application runs FireNet [Dunnings and Breckon 2018] (a modified version of AlexNet [Krizhevsky et al. 2017]) to detect fire present in captured pictures or video frames. The dataset can be found in [Dunnings and Breckon 2020].

In Figure 3.7, we observe that under two different settings ① $\gamma = 0.5, \pi = 1, \tau = 20$ and ② $\gamma = 0.5, \pi = 1, \tau = 40$, to reach accuracy goal 85%, pFedMo spends 180min for setting ① and 137min for setting ②. Benchmarks spend 267–655min for setting ① and 149–361min for setting ②. This demonstrates that pFedMo is superior than benchmarks and achieves the 1.09–3.64x training time speedup compared to benchmarks in different scenarios.

3.6 Summary of Improvement Compared with Honours [Fu 2021]

A small portion of the experiment of this chapter is built on preliminary work conducted in my Honours thesis [Fu 2021], which explored a simple momentum strategy in FL through basic experiments. However, the algorithm design in this chapter is entirely new, and the Honours work did not include any theoretical analysis. The experimental evaluation is also significantly expanded with deeper models, diverse datasets, and real-world validation in an IoT fire detection system. This chapter substantially advances the work [Fu 2021] in terms of motivation, algorithm design, theoretical analysis, and experimental scope.

3.7 Conclusion

In this chapter, we propose pFedMo, a PFL with contrastive momentum algorithm. A score function is designed to personalize worker momentum and model at each aggregation phrase. We provide convergence analysis of pFedMo with a convergence rate of $\mathcal{O}\left(\frac{1}{T}\right)$ for smooth non-convex problems under non-i.i.d. data. We experimentally evaluate the convergence performance of pFedMo compared to mainstream momentum-based algorithms without personalization from the perspectives of training accuracy and training time. It empirically shows that pFedMo consistently improves the convergence performance especially on highly heterogeneous (non-i.i.d.) data.

CHAPTER 4

pFedPLL: Personalized Federated Partial Label Learning

In Chapter 3, we proposed pFedMo, which is trained under a fully supervised setting. However, this assumption is often too idealistic in the context of FL. Due to the decentralized and privacy-preserving nature of FL, it is unrealistic to assume that all training data across clients are high-quality and fully labeled. In real-world scenarios, data are often noisy, incomplete, or weakly supervised, making it essential to design algorithms that can operate effectively under such imperfect conditions. To this end, in this chapter, we propose pFedPLL, a personalized federated partial label learning algorithm with two new designs. In Label Correlation Isolation (LCI), we first develop a twin-module architecture, where a feature-level correlation matrix layer for each worker is isolated locally to prevent it from being interfered with by others. In Label Correlation Personalization (LCP), we then propose a bi-directional calibration loss to identify a more accurate learning direction, where the positive calibration aligns the prediction result with the latent true label, and the negative calibration pushes away the prediction result that falls into the non-candidate label set. We provide a convergence analysis of pFedPLL with a rate of $O\left(\sqrt{\frac{1}{T}}\right)$ for smooth non-convex problems. Experiment results demonstrate that pFedPLL outperforms SOTA federated PLL algorithms and the federated version of centralized PLL algorithms across eleven datasets.

The pFedPLL has been submitted to IEEE Transactions on Computers (TC) and is currently under review under the title "When Federated Learning Meets Partial Label Data."

4.1 Introduction

The performance of FL [McMahan et al. 2017] is highly related to the *quantity*, *quality*, and *heterogeneity* of data. However, due to different data collection environments (non-expert users, limited computational power, and varying geo-locations of edge devices, etc.), it is costly to collect a large amount (high quantity) of high quality (with ground truth label) data instances. One possible solution is to first assign a set of potential true labels to each specific data instance to quickly collect a large quantity of data (partial label data). Then, the Machine Learning (ML) model learns and analyzes the label correlation to mitigate the negative effect of noisy labels during training. Such a learning method is within the scope of Partial Label Learning (PLL) [Cour et al. 2011; Tian et al. 2023; Jin and Ghahramani 2002; Liu and Dietterich 2012], where the candidate label set (containing potential true labels) consists of one ground truth label and several correlated false positive labels (noisy labels). Since the ground truth label is concealed in the candidate label set, PLL aims to train a classifier to predict the latent true label (the most likely true label) for each data instance by analyzing the correlations among the labels within the candidate label set. The model performance is highly dependent on the accurate label correlation learned during training.

The label correlation in PLL has been proven to be a valuable component in centralized ML. It finds a way of training a model with relatively *large quantity* but *low quality* data while maintaining the model performance. However, when data is non-independent and identically distributed (non-i.i.d.) among workers in FL, the centralized label correlation may not work in such *heterogeneous* data scenario. Due to non-i.i.d. data, the label correlation learned from the local dataset is only applicable to each worker. When aggregation occurs, it will cause the *label correlation interference* problem. For example, in Fig. 4.1, when collecting handwritten digits (0–9) from different individuals across various locations, in worker “A”, the image data instance of digit “2” is very similar to digit “3” (high label correlation between digit “2” and “3”), while in worker “B”, the digit “2” has a high label correlation with digit “7”. During the aggregation phase, the global model aggregates each worker’s label correlation and learns that the digit “2” has a high label correlation with both “3” and “7”. When the global model

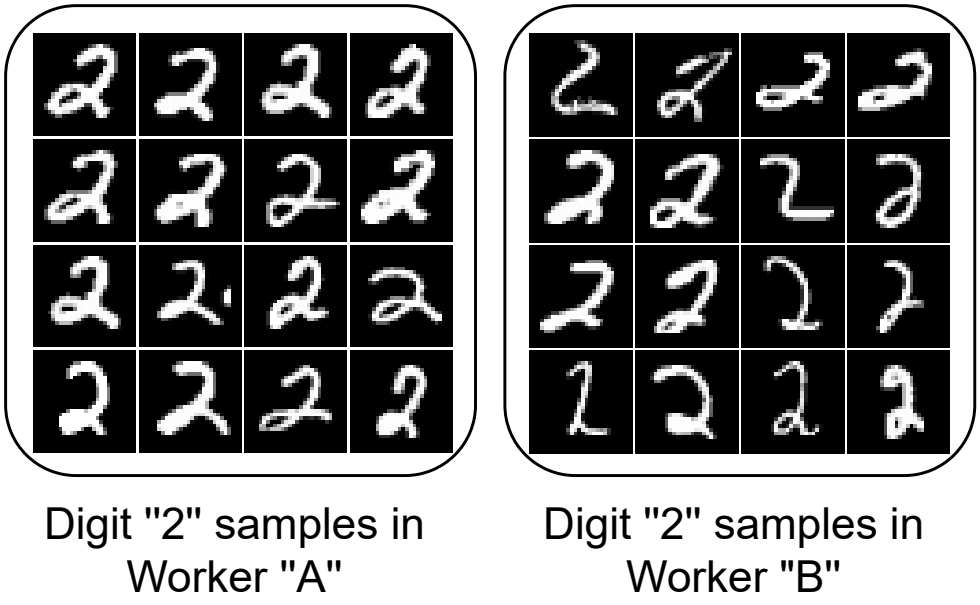

FIGURE 4.1. Digit “2” samples in workers. Due to aging differences or variations in geo-location, the same digit “2” may exhibit different appearances. Digit “2” in worker “A” resembles a “3”, while those in worker “B” resembles a “7”.

is distributed back to workers, such *label correlation interference* will mislead each worker’s unique label correlation information, resulting in the degradation of model performance.

To address this issue, we develop an all-in-one solution, pFedPLL, a personalized federated partial label learning algorithm that efficiently trains FL models on *heterogeneous* partial label data. The pFedPLL algorithm incorporates two complementary designs to mitigate label correlation interference. The *Label Correlation Isolation (LCI)* isolates correlation learning for each client through a feature-level correlation matrix, preventing cross-client interference during aggregation. Building on this, the *Label Correlation Personalization (LCP)* adaptively refines each client’s learning direction based on its own correlation distribution, achieving personalized yet consistent correlation learning under non-i.i.d. conditions.

LCI. We develop a new correlation matrix layer that is only updated on the local/worker dataset and is not aggregated to the global model, preventing each worker’s unique label correlation information from being interfered with by others. The training layers before the

correlation matrix layer are aggregated to obtain the global representative feature information. Please note that we insert the correlation matrix layer in the second-to-last layer of the DNN model instead of the last layer. This is because the last layer learns the exact label correlation (the same dimensions as the number of classes of labels) with coarse granularity. To learn more correlation information, we push the correlation matrix layer to the second-to-last layer by learning feature correlation with fine granularity, where the second-to-last layer usually has more dimensions than the number of classes of labels.

LCP. Since the correlation matrix in LCI is isolated locally, the matrix itself is personalized. In order to better utilize the personalized correlation matrix information to help identify a more accurate learning direction, we further develop a **bi-directional calibration** loss. Apart from the basic well-known summarization PLL loss [Feng et al. 2020], we propose *positive calibration* to align the prediction result with the latent true label, and *negative calibration* to push away the prediction result that falls into the non-candidate label set (labels that are not within the candidate label set).

We evaluate our pFedPLL algorithm both theoretically and experimentally. We prove that pFedPLL converges for smooth non-convex problems at a rate of $O\left(\sqrt{\frac{1}{T}}\right)$ over T global iterations. In the experiments, we compare pFedPLL with mainstream federated PLL algorithms and federated version of centralized PLL algorithms across nine datasets. The ablation study of LCI and LCP is also evaluated. The experiment results demonstrate that pFedPLL consistently outperforms all benchmark algorithms with up to 49.93% accuracy increase in various ML settings.

4.2 Algorithm Design

4.2.1 Problem Formulation

We consider a typical FL system consisting of N workers/clients (indexed by n) and one aggregator. The worker n maintains a local partial label dataset $D_n \triangleq \{(\boldsymbol{x}_i^n, Y_i^n)\}_{i=1}^{|D_n|}$, where $|D_n|$ is the total number of data samples in D_n, $\boldsymbol{x}_i^n$ is the ith data instance in worker n, and

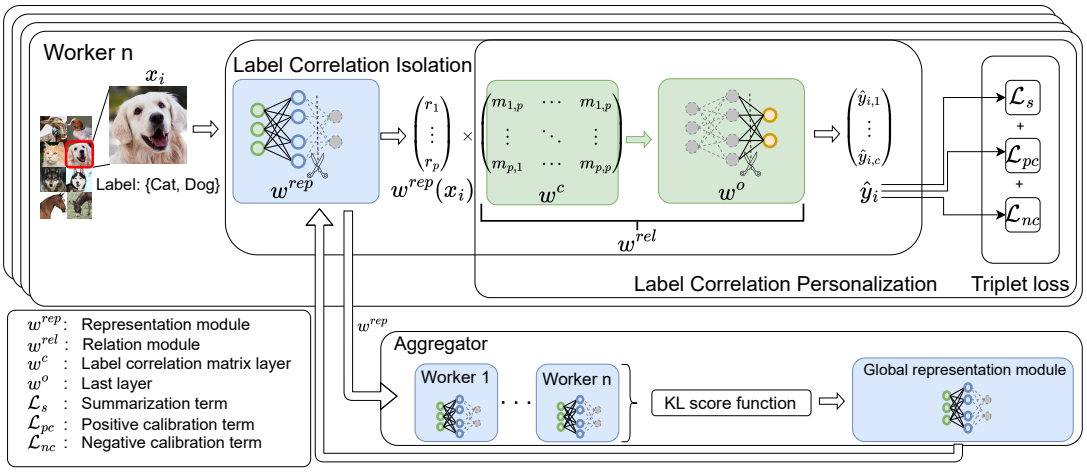

FIGURE 4.2. An overview of pFedPLL. Each data instance in the partial label dataset is linked to a candidate label set, with only one true label that remains unknown during training. Label Correlation Isolation (LCI) is achieved through the **twin-module** architecture ($\boldsymbol{w}^{rep}\&\boldsymbol{w}^{rel}$), where only the representation module is aggregated in the server. Label Correlation Personalization (LCP) is accomplished by training the model with the triplet loss including **bi-directional calibration** ($\mathcal{L}_{pc}\&\mathcal{L}_{nc}$).

$Y_i^n \in \{0,1\}^C$ is the label set of $\boldsymbol{x}_i^n$ across C label classes. Let $M_i^n = \{j \in C \mid y_{i,j}^n = 1, y_{i,j}^n \in Y_i^n\}$ denote the candidate label set and $\bar{M}_i^n = \{j \in C \mid y_{i,j}^n = 0, y_{i,j}^n \in Y_i^n\}$ denote the non-candidate label set, i.e., the sum of length of M_i^n and $\bar{M}_i^n$ equals to C. The ground truth label is known to reside in the corresponding candidate label set, i.e., $y_{i,j}^n \in Y_i^n, \exists j \in M_i^n$, but cannot be directly accessible, bringing significant challenges for training. The objective is to find the optimal model $\boldsymbol{w}^*$ that minimizes the global loss function

$$\min_{\boldsymbol{w}\in\mathbb{R}^d} F(\boldsymbol{w}) \triangleq \sum_{n=1}^{N} F_n(\boldsymbol{w}, D_n), \tag{4.1}$$

where d is the dimension of model $\boldsymbol{w}$, $F(\boldsymbol{w})$ is the global loss function, and $F_n(\boldsymbol{w})$ is the worker n's loss function. Here, we define the global loss function $F(\boldsymbol{w})$ as the sum of all workers' loss functions $F_n(\boldsymbol{w})$, because the weight/contribution of each worker loss function is learnt during the training, rather than being predefined. Key notations are summarized in Table 4.1.

TABLE 4.1. pFedPLL: Key Notations

η	learning rate
τ	number of local iteration
T	number of global iterations
N	number of worker
s_t^n	worker n's score at t-th global aggregation
ρ	temperature variable to control size of candidate label set
M_i^n	candidate label set of the instance i of worker n
$\bar{M}_i^n$	non-candidate label set of the instance i of worker n
$\boldsymbol{\alpha}_i$	the confidence of each label for ith instance
$\boldsymbol{w}_{t,v}^{n,rep}$	worker n's representation module at the v-th local iteration of the t-th global iteration
$\boldsymbol{w}_{t,v}^{n,rel}$	worker n's relation module at the v-th local iteration of the t-th global iteration
$\boldsymbol{w}_{t,v}^{n}$	worker n' complete worker model at the v-th local iteration of the t-th global iteration
$\boldsymbol{w}_t^{rep}$	global representation module at t-th global iteration
$\boldsymbol{w}_t^{rel}$	virtual global relation module at t-th global iteration
$\boldsymbol{w}_t$	virtua global complete model at t-th global iteration

4.2.2 Motivation

Label correlation is an effective component in centralized partial label learning (PLL) [Xu et al. 2020]. It helps train models with large but noisy supervision while maintaining performance. In FL, however, data are decentralized and non-i.i.d., so each worker only learns the correlation structure present in its own local dataset. When these heterogeneous correlations are aggregated, they conflict and contaminate one another, causing *label correlation interference*. As illustrated in Fig. 4.1, the digit "2" in worker "A" is closer to "3," whereas in worker "B" it is closer to "7." A global model that merges such correlations incorrectly ties "2" to both "3" and "7," and when redistributed it misleads local training and degrades performance.

To address this issue, we introduce two complementary mechanisms: *Label Correlation Isolation (LCI)*, which prevents cross-worker interference, and *Label Correlation Personalization (LCP)*, which further leverages the preserved local correlations for personalization. Together, LCI prevents cross-worker interference of correlation information, and LCP actively leverages the preserved structure to tailor learning to each client.

4.2.3 LCI: Label Correlation Isolation

The typical label correlation mechanism works well in centralized PLL [Xu et al. 2020]. It learns the label correlation from a single centralized dataset. When training FL models on decentralized heterogeneous (non-i.i.d.) datasets, each worker can only learn its own label correlation from its own dataset. When aggregation occurs, one worker's label correlation might mislead others due to non-i.i.d. data. To this end, we develop the Label Correlation Isolation (LCI) mechanism, which keeps the label correlation information local and thus prevents it from being interfered with by others.

We start by considering a general Deep Neural Network (DNN) with l layers. We first propose a label correlation matrix layer with the dimension of $p \times p$. Then, we insert the layer into the second-to-last position of the original DNN to form a new $(l+1)$ layers DNN, where the label correlation matrix is the lth layer. Here, p varies depending on the architecture of the DNN model but is larger than the number of label classes ($p > C$). Therefore, the matrix contains more fine-grained information for better performance (feature-level correlation vs. label-level correlation). Let $\boldsymbol{w}^{rep}$, $\boldsymbol{w}^{c}$, and $\boldsymbol{w}^{o}$ denote the 1st to $(l-1)$th layers, lth layer, and $(l+1)$th layer respectively. The label prediction $\hat{\boldsymbol{y}}_i$ for $\boldsymbol{x}_i$ is calculated as

$$\hat{\boldsymbol{y}}_i = \boldsymbol{w}^{o}(\boldsymbol{w}^{c}(\boldsymbol{w}^{rep}(\boldsymbol{x}_i))). \tag{4.2}$$

Since the correlation matrix is isolated within each worker, we introduce a **twin-module** architecture where $\boldsymbol{w}^{rep}$ is the representation module and $\boldsymbol{w}^{rel} \triangleq [\boldsymbol{w}^{c}, \boldsymbol{w}^{o}]$ denotes the relation module. $\boldsymbol{w}^{rep}$ is aggregated and redistributed back to each worker to obtain robust/global feature representation information. $\boldsymbol{w}^{rel}$ contains the worker's unique feature-level correlation and is kept locally (not being aggregated). $\boldsymbol{w}^{c}$ is initialized as a $p \times p$ diagonal matrix with "1" on the diagonal and "0" elsewhere when training begins. By adding superscript n (worker index), and subscripts i (data instance index), t (global iteration index), and v (local iteration index), the prediction of the ith data instance for nth worker at the vth local iteration in the tth global iteration is calculated as

$$\hat{\boldsymbol{y}}^{n}_{i,t,v} = \boldsymbol{w}^{n,rel}_{t,v}(\boldsymbol{w}^{n,rep}_{t,v}(\boldsymbol{x}_i)), \forall i \in |D_k|. \tag{4.3}$$

The complete worker n's model is then denoted by $\boldsymbol{w}_{t,v}^{n} \triangleq [\boldsymbol{w}_{t,v}^{n,rep}, \boldsymbol{w}_{t,v}^{n,rel}]$.

In summary, by implementing the twin-module architecture, we address the label correlation interference issue while maintaining the robust model performance.

4.2.4 LCP: Label Correlation Personalization

We have utilized LCI in Section 4.2.3 to prevent each worker's correlation information from being interfered with by others. In this circumstance, the worker's unique correlation matrix is personalized. The next step is to further utilize this information to help identify a more accurate learning direction for each worker. On the basis of the summarization PLL loss [Feng et al. 2020], we propose **bi-directional calibration** loss for each worker to encourage the correct prediction (aligning with the latent true label) while discouraging the false prediction (prediction falls into the non-candidate label set).

Summarization. We begin by introducing the summarization loss, a well-known loss used in PLL which lets the model's prediction fall into the candidate label set. For convenient presentation, we omit iteration indexes and worker index (t,v, and n) for now. Let $\tilde{\boldsymbol{y}}_i$ denote the prediction (distribution) of the model after normalization (`softmax`) of $\hat{\boldsymbol{y}}_i$, i.e.,

$$\tilde{y}_{i,j} = \texttt{softmax}(\hat{y}_{i,j}) = \frac{\exp\left(\hat{y}_{i,j}\right)}{\sum_{c=1}^{C} \exp\left(\hat{y}_{i,c}\right)}, \forall \hat{y}_{i,c} \in \hat{\boldsymbol{y}}_i, \tilde{y}_{i,j} \in \tilde{\boldsymbol{y}}_i. \tag{4.4}$$

Then, we define the summarization loss as

$$\mathcal{L}_s = -\log(\sum_{j \in M_i} \tilde{y}_{i,j}). \tag{4.5}$$

It calculates the predicted values of all labels within the candidate label set, but treats them equally (simple summation without weight). Therefore, by minimizing $\mathcal{L}_s$, we can only foster the prediction result falling into the candidate label set.

Positive calibration. $\mathcal{L}_s$ has made the prediction fall into the candidate label set. Then, we zoom in on the candidate label set to distinguish each candidate label so as to find the latent true label. Let $\boldsymbol{\alpha}_i = [\alpha_{i,1}, \alpha_{i,2}, \ldots, \alpha_{i,C}]^\top$ denote the confidence of each label for data

instance $\boldsymbol{x}_i$, where $\sum_{j \in M_i} \alpha_{i,j} = 1, \alpha_{i,j} \geq 0, \forall j \in M_i$, and $\alpha_{i,j} = 0, \forall j \in \bar{M}_i$. Please note $\boldsymbol{\alpha}_i$ dynamically evolves during the training, and we only use the confidence related to candidate labels. When training begins, $\alpha_{i,j}$ is initialized as $\frac{1}{|M_i|}, \forall j \in M_i$, where $|M_i|$ is the size of M_i. During training, $\alpha_{i,j}$ is updated as

$$\alpha_{i,j} = \frac{\tilde{y}_{i,j}}{\sum_{m \in M_i} \tilde{y}_{i,m}}, \forall j \in M_i. \tag{4.6}$$

Then, we define the positive calibration loss as

$$\mathcal{L}_{pc} = -\sum_{j \in M_i} \alpha_{i,j} \log(\tilde{y}_{i,j}). \tag{4.7}$$

Here, we put the confidence $\alpha_{i,j}$ as the weight/factor of $\log(\tilde{y}_{i,j})$, which aligns the confidence with the predicted result for a specific label j in data instance $\boldsymbol{x}_i$. Therefore, when both $\alpha_{i,j}$ and $\tilde{y}_{i,j}$ are large, the $\mathcal{L}_{pc}$ is minimized, implying the nature of positive calibration, where the predicted label with high weight (latent true label) should have high confidence. In the meanwhile, the noisy information from potential false-positive labels is alleviated.

Negative calibration. When we zoom out on the whole label set (candidate label set vs. non-candidate label set), any prediction that falls into the non-candidate label set is wrong. To this end, we develop a negative calibration that pushes the prediction away from the non-candidate label set. We select the label with the highest weight in the non-candidate label set and give a penalty to this label (adding a "minus"). The negative calibration loss is calculated as

$$\mathcal{L}_{nc} = -\log(1 - \max\{\tilde{y}_{i,j} \in \tilde{\boldsymbol{y}}_i \mid j \in \bar{M}_i\}). \tag{4.8}$$

By minimizing $\mathcal{L}_{nc}$, the probability of the highest label prediction in non-candidate label set is minimized, implying the more likely prediction in the candidate label set.

Triplet loss. To summarize, we construct the final triplet loss for each worker n as follows (iteration indexes and worker index (t, v, and n) are omitted on the right hand side):

$$F_n(\cdot) = \lambda_1 \mathcal{L}_s + \lambda_2 \mathcal{L}_{pc} + \lambda_3 \mathcal{L}_{nc}, \tag{4.9}$$

where λ_1, λ_2, and λ_3 are hyperparameters that can be adjusted before the training begins.

In summary, by implementing bi-directional calibration, we better utilize the personalized worker correlation information to predict the latent true label for better model performance.

4.2.5 Implementation

The pFedPLL is implemented in Algorithm 3 with T total global iterations (indexed by t) and τ local iterations (indexed by $v \in [0, \ldots, \tau - 1]$) between two consecutive global iterations. As shown in Fig. 4.2, the pFedPLL generally comprises worker update with triplet loss calculations for LCP, and aggregator update with LCI and KL score calculation for dynamic model weight/contribution.

Worker update. At each local iteration $v \in [0, \ldots, \tau - 1]$ for the tth global iteration, each worker first randomly fetches a mini-batch ξ from D_n (Line 6) and calculates the worker model loss (Line 7). Then, each worker updates its representation module (Line 8) and relation module (Line 9) via related gradient descent with $\nabla \boldsymbol{w}^{rep} F_n(\boldsymbol{w}^n_{t,v}, \xi)$ for representation module and $\nabla \boldsymbol{w}^{rel} F_n(\boldsymbol{w}^n_{t,v}, \xi)$ for relation module. η is the learning rate. Afterward, the confidence of each data instance in ξ is updated (Line 11). Finally, when $v == \tau - 1$ (the end of local iteration for the tth global iteration), each worker sends its representation module to the aggregator (Line 13).

Aggregator update. At each global iteration $t \in [0, \ldots, T - 1]$, when $v == \tau - 1$ (the end of the local iteration for the tth global iteration), the aggregator first calls the `KL_Score` function (Algorithm 4) to calculate scores for all workers (Line 15). Then, the aggregator aggregates the global representation module using the scores as weights (Line 16) and distributes it back to each worker (Line 17). Each worker's unique relation module is kept locally (Line 19), and the updated complete worker model is constructed for the next round of local iteration (Line 20).

KL score. To better measure the weight/contribution of worker models towards the global model, we develop a dynamic weight assignment algorithm in Algorithm 4, using the Kullback–Leibler (KL) divergence [Press et al. 2007] to measure the distance/similarity between the global representation module $\boldsymbol{w}^{rep}_t$ and each worker n's representation module $\boldsymbol{w}^{n,rep}_{t,\tau}$. We

Algorithm 3 pFedPLL algorithm

Input: τ, T, η, N

Output: Final global representation module $\boldsymbol{w}_T^{rep}$

For each worker n, initialize: $\boldsymbol{w}_{0,0}^{n,rep}$, $\boldsymbol{w}_{0,0}^{n,rel}, \forall n \in N$, as the same value respectively, and $\alpha_{i,j}^n = \frac{1}{|M_i^n|}, \forall j \in M_i^n, n \in N$, where $|M_i^n|$ is the size of M_i^n.
For the aggregator, initialize: $\boldsymbol{w}_0^{rep} = \boldsymbol{w}_{0,0}^{n,rep}$.
for $t = 0, 1, \ldots, T-1$ **do**
  For each worker $n = 1, \ldots, N$ in parallel:
  for $v = 0, 1, \ldots, \tau - 1$ **do**
    Randomly fetch mini-batch ξ from D_n.
    Calculate triplet loss $F_n(\boldsymbol{w}_{t,v}^n, \xi)$ as (4.9).
    $\boldsymbol{w}_{t,v+1}^{n,rep} = \boldsymbol{w}_{t,v}^{n,rep} - \eta \nabla_{\boldsymbol{w}^{rep}} F_n(\boldsymbol{w}_{t,v}^n, \xi)$
    //`Update worker k's representation module`
    $\boldsymbol{w}_{t,v+1}^{n,rel} = \boldsymbol{w}_{t,v}^{n,rel} - \eta \nabla_{\boldsymbol{w}^{rel}} F_n(\boldsymbol{w}_{t,v}^n, \xi)$
    //`Update worker n's relation module`
    For each instance i in mini-batch ξ, update confidence
    $\alpha_{i,j} = \frac{\tilde{y}_{i,j}}{\sum_{m \in M_i} \tilde{y}_{i,m}}, \forall j \in M_i$.
  end for
  Send $\boldsymbol{w}_{t,\tau}^{k,rep}$ to the aggregator.
  For the aggregator:
  $\mathcal{S}_t$ = `KL_Score` $(w_t^{rep}, w_{t,\tau}^{1,rep}, \ldots, w_{t,\tau}^{N,rep}, N, B)$
  //`Call KL_Score function in Algorithm 4`
  $\boldsymbol{w}_{t+1}^{rep} = \sum_{n=1}^{N} s_t^n \boldsymbol{w}_{t,\tau-1}^{n,rep}, \forall s_t^n \in \mathcal{S}_t$
  //`Aggregate global representation module`
  $\boldsymbol{w}_{t+1,0}^{n,rep} = \boldsymbol{w}_{t+1}^{rep}$ for each worker n.
  //`Distribute global representation module to workers`
  For each worker $n = 1, \ldots, N$ in parallel:
  $\boldsymbol{w}_{t+1,0}^{n,rel} = \boldsymbol{w}_{t,\tau-1}^{n,rel}$
  //`Retain worker's own relation module`
  $\boldsymbol{w}_{t+1,0}^{n} = [\boldsymbol{w}_{t+1,0}^{n,rep}, \boldsymbol{w}_{t+1,0}^{n,rel}]$
  //`Construct worker's complete model`
end for

first feed the same B batches of data $\mathcal{A} = \{a_1, \ldots, a_B\}$ (randomly selected from the validation dataset[1]) to $\boldsymbol{w}_t^{rep}$ and $\boldsymbol{w}_{t,\tau}^{n,rep}$, and then compare the predictions using the KL divergence (Line 2 in Algorithm 4),

$$\mathcal{L}_{n,b}^t = \texttt{KL}(\boldsymbol{w}_{t,\tau}^{n,rep}(a_b) || \boldsymbol{w}_t^{rep}(a_b)), \forall b \in B, \tag{4.10}$$

[1]The validation dataset is randomly selected, comprising 20% of the test dataset, and is not used for training.

Algorithm 4 KL_Score function

Input: w_t^{rep}, $w_{t,\tau}^{1,rep}, \ldots, w_{t,\tau}^{n,rep}, N, B$
Output: $\mathcal{S}_t = \{s_t^1, s_t^2, \ldots, s_t^N\}$

1: For each batch $b \in B$ and worker $n \in N$:
2: Calculate KL divergence
$\mathcal{L}_{n,b}^t = \texttt{KL}(\boldsymbol{w}_{t,\tau}^{n,rep}(a_b)||\boldsymbol{w}_t^{rep}(a_b))$.
3: For each worker $n \in N$:
4: Obtain the averaged distance $\ell_t^n = \frac{1}{B}\sum_{b=1}^{B}\mathcal{L}_{n,b}^t$.
5: Obtain the score $s_t^n = \frac{\ell_t^n}{\sum_{i=1}^{N}(\ell_t^i)}$.

where a_b is the bth batch in $\mathcal{A}$. We repeat the process for each batch and each worker to obtain the distance matrix $\mathbf{L}^t = \left[\mathcal{L}_{n,b}^t\right]_{N\times B}$. Finally, we average the values in each row to obtain the averaged distance for each worker $\ell_t^n = \frac{1}{B}\sum_{b=1}^{B}\mathcal{L}_{n,b}^t$ (Line 4), which is then normalized to obtain the final score $s_t^n = \frac{\ell_t^n}{\sum_{i=1}^{N}(\ell_t^i)}, \forall n \in N$ (Line 5). A higher score means a longer distance between the worker and global representation modules, suggesting that such worker representation module should contribute more to the global representation module, and thus we assign such score as the weight (Line 16 in Algorithm 3). The output for Algorithm 4 are scores for all workers in the tth global iteration $\mathcal{S}_t = \{s_t^1, \ldots, s_t^N\}$, where $\sum_{n\in N} s_t^n = 1$.

4.3 Convergence Analysis

To prove the convergence, we propose a virtual relation module update as if each worker's isolated relation module is aggregated and redistributed, i.e., $\boldsymbol{w}_{t+1}^{rel} = \sum_{n=1}^{N} s_t^n \boldsymbol{w}_{t,\tau}^{n,rel}$ and $\boldsymbol{w}_{t+1,0}^{n,rel} = \boldsymbol{w}_{t+1}^{rel}$. Then, the complete virtual global model is denoted by $\boldsymbol{w}_{t+1} \triangleq [\boldsymbol{w}_{t+1}^{rep}, \boldsymbol{w}_{t+1}^{rel}]$. According to the update rules (Lines 8–9 and 16–17) in Algorithm 3, after performing the mathematical transformations, we derive

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \sum_{n=1}^{N} s_t^n \sum_{v=0}^{\tau-1} [\nabla_{\boldsymbol{w}^{rep}} F_n(\boldsymbol{w}_{t,v}^n), \nabla_{\boldsymbol{w}^{rel}} F_n(\boldsymbol{w}_{t,v}^n)], \tag{4.11}$$

which is the basis to prove the convergence. We assume that the gradient of $F_n(\cdot)$ and $F(\cdot)$ is L-Lipschitz and has bounded diversity, i.e., $\|\nabla F(\boldsymbol{w}_1) - \nabla F(\boldsymbol{w}_2)\|_2 \leq L\|\boldsymbol{w}_1 - \boldsymbol{w}_2\|_2$, $\|\nabla F_n(\boldsymbol{w}_1) - \nabla F_n(\boldsymbol{w}_2)\|_2 \leq L\|\boldsymbol{w}_1 - \boldsymbol{w}_2\|_2$, and $\mathbb{E}_{\xi\sim D_n}\|\nabla F_n(\boldsymbol{w}_1, \xi) - \nabla F_n(\boldsymbol{w}_1)\|_2^2 \leq$

$\delta^2, \forall \boldsymbol{w}_1, \boldsymbol{w}_2, n$, which are necessary conditions for convergence analysis in the literature [Wang et al. 2019; Yang et al. 2022b; Huo et al. 2020].

THEOREM 4. *Suppose (1)* $\eta \leq \frac{1}{2L\tau}$*, and (2)* $\exists F_{inf}$ *is the lower bound of* $F(\cdot)$*, we have*

$$\min_{t\in[0,\ldots,T-1]} \mathbb{E}_{\xi,n}\|\nabla F(\boldsymbol{w}_t)\|_2^2 \leq \frac{4L}{T}(F(\boldsymbol{w}_0) - F_{inf})) + 3\eta^2 L^2 \delta^2. \tag{4.12}$$

PROOF. See complete proof in Appendix B. □

Theorem 4 demonstrates that the square of the global model gradient is upper bounded by a function that is inversely proportional to T. The output of the Algorithm 3, $\boldsymbol{w}^{rep}$ is included in the global model $\boldsymbol{w}$, i.e., $\boldsymbol{w} = [\boldsymbol{w}^{rep}, \boldsymbol{w}^{rel}]$. Therefore, we prove that the pFedPLL is convergent with the convergence rate of $O\left(\sqrt{\frac{1}{T}}\right)$ for smooth non-convex problems.

4.4 Experiments

4.4.1 Experiment Setup

Comparison methods. We compare the proposed pFedPLL with existing FedPLL method, the federated extensions of centralized PLL algorithms, and the classic FedAvg [McMahan et al. 2017]. As FedPLL is still an emerging research area, only one existing work, FedPLL_LAAR [Yan and Guo 2024], explicitly addresses this setting. FedPLL_LAAR introduces local-adaptive augmentation and regularization to alleviate client heterogeneity and improve model generalization. For FedPLL_LAAR, most hyperparameter settings are from the original paper, but others are fine-tuned to improve performance in our settings. In addition, we implement Fed_CC, Fed_RC, Fed_CVAL, Fed_LW, and Fed_ABS_GCE by extending the centralized methods, namely CC and RC [Feng et al. 2020], CVAL [Zhang et al. 2021b], LW [Wen et al. 2021], and ABS_GCE [Lv et al. 2023], into a federated setting. Each client applies the corresponding loss locally, and the global model is aggregated via FedAvg. These variants do not exist in prior literature and are developed here solely for fair comparative evaluation.

Datasets. We utilize six benchmark datasets (MNIST [LeCun et al. 1998], Fashion-MNIST (F-MNIST) [Xiao et al. 2017], Kuzushiji-MNIST (K-MNIST) [Clanuwat et al. 2018], CIFAR-10 [Krizhevsky, Hinton et al. 2009]), Tiny-ImageNet [Le and Yang 2015] and SVHN Netzer et al. 2011) and five real-world partial label datasets (Lost [Cour et al. 2011], BirdSong [Briggs et al. 2012], MSRCv2 [Liu and Dietterich 2012], Soccer Player [Zeng et al. 2013], and Yahoo!News [Guillaumin et al. 2010)]. The MNIST, F-MNIST, and K-MNIST datasets, each consists of 60,000 training images and 10,000 test images. Each image is a grayscale 28×28 pixel grid categorized into 10 classes. MNIST, F-MNIST, and K-MNIST are designed for handwritten digit classification, fashion item classification, and Japanese character classification, respectively. The CIFAR-10 dataset includes 50,000 training images and 10,000 test images, with each $32 \times 32 \times 3$ colored image in RGB format organized into 10 classes for object classification tasks. The Tiny-ImageNet includes 100,000 training images and 10,000 test images, each being a $64 \times 64 \times 3$ RGB-colored image. The images are organized into 200 classes, designed for object classification tasks. The SVHN dataset consists of 73,257 training images, 26,032 test images, and 531,131 additional samples, totaling over 600,000 images of $32 \times 32 \times 3$ RGB street view house numbers. Each image is labeled with one of 10 digit classes (0–9), making it a large-scale real-world benchmark for digit recognition tasks. Please note that the benchmark datasets are originally intended for supervised learning, and we manually convert them into partial label datasets.The Lost dataset, based on the TV series "Lost", includes 1,122 face images with 108 features, categorized into 16 names with an average candidate label set size of 2.23. The Birdsong dataset focuses on bird song classification, featuring 4,998 data points and 38 features, grouped into 13 categories with an average candidate label set size of 2.18. The MSRCv2 dataset, aimed at object classification, contains 1,758 data points with 48 features, categorized into 23 objects and an average candidate label set size of 3.16. The Soccer Player dataset includes 17,472 data points with 279 features for naming soccer players, categorized into 171 names, and an average candidate label set size of 2.09. Lastly, the Yahoo!News dataset for automatic face naming in news articles includes 22,991 data points with 163 features, grouped into 219 names, with an average candidate label set size of 1.91.

Data generation. To convert benchmark datasets into partial label datasetswe consider two common label generation mechanisms: class-independent [Lv et al. 2020] and instance-dependent processes [Xu et al. 2021]. The former randomly adds false positive labels regardless of instance features (e.g., horse vs. cat), while the latter generates candidate labels conditioned on instance characteristics, leading to more structured and correlated ambiguity. In this work, we adopt the instance-dependent generation process, as it better reflects real-world labeling uncertainty where visually or semantically similar labels (e.g., horse vs. donkey) are more likely to co-occur in the candidate set. Technically, we determine the flipping probability for each incorrect label corresponding to an instance $\boldsymbol{x}_i$ using the confidence predictions from a clean neural network, $\hat{\theta}$, trained on the original supervised dataset. The flip probability for each incorrect label for instance $\boldsymbol{x}_i$ is calculated as:

$$\zeta_j = \frac{f_j(\boldsymbol{x}_i; \hat{\theta})}{\max_{z \in \tilde{\boldsymbol{Y}}_i} f_z(\boldsymbol{x}_i; \hat{\theta})} \rho, \forall j \in \tilde{\boldsymbol{Y}}_i, \tag{4.13}$$

where $\tilde{\boldsymbol{Y}}_i$ is the set of all incorrect labels except for the true label of $\boldsymbol{x}_i$, and $f_j(\boldsymbol{x}_i; \hat{\theta})$ is the confidence prediction of the clean neural network $\hat{\theta}$. $\rho \in [0, 1]$ is the temperature hyperparameter to control the candidate label set size. For non-i.i.d. data generation, we use a Dirichlet distribution $Dir(\beta)$ Minka 2000 to generate local data for each worker, where $\beta \in (0, +\infty)$ is a hyperparameter that controls the level of data heterogeneity. A smaller β indicates a higher level of non-i.i.d. In all experiments, we set $\beta = 0.5$. For accuracy assessment, since the aggregator lacks a complete model (only a global representation module exists), we assess the final accuracy as the averaged accuracy from all workers. Following the same methods [Tan et al. 2023; Lu et al. 2022], we split each worker's dataset into training (80%) and testing (20%) datasets without any data instance overlap and then test each worker's accuracy to obtain the final accuracy.

Equipment and hyperparameter settings. The experiments are implemented in a cross-silo FL [Kairouz et al. 2021] setting, where all clients (workers) participate in every communication round. We use a GPU tower server equipped with 4 NVIDIA GeForce RTX 3090 GPUs to conduct experiments. In all experiments, models are updated by mini-batch SGD with

TABLE 4.2. pFedPLL: Accuracy (%) comparisons on benchmark and real-world partial label datasets when $T = 100$. We use a 5-layer LeNet [LeCun et al. 1998] for MNIST, K-MNIST, and F-MNIST, a 34-layer ResNet [He et al. 2016] for CIFAR-10, a 18-layer ResNet [He et al. 2016] for Tiny-ImageNet, a 5-layer MLP for SVHN, and a 2-layer MLP for real-world datasets.

		pFedPLL	Fed_LW	Fed_CC	Fed_RC
Benchmark dataset	MNIST	**98.14** $\pm$ 0.06	97.04 $\pm$ 0.05	96.60 $\pm$ 0.08	96.44 $\pm$ 0.03
	K-MNIST	**89.15** $\pm$ 0.03	81.77 $\pm$ 0.08	79.87 $\pm$ 0.09	78.81 $\pm$ 0.08
	F-MNIST	**84.06** $\pm$ 0.07	82.89 $\pm$ 0.11	82.35 $\pm$ 0.13	80.17 $\pm$ 0.09
	CIFAR-10	**82.10** $\pm$ 0.11	67.94 $\pm$ 0.07	70.91 $\pm$ 0.09	59.56 $\pm$ 0.14
	Tiny-ImageNet	**73.64** $\pm$ 0.07	62.80 $\pm$ 0.09	62.20 $\pm$ 0.05	62.06 $\pm$ 0.11
	SVHN	**81.30** $\pm$ 0.09	76.75 $\pm$ 0.07	75.61 $\pm$ 0.06	71.61 $\pm$ 0.12
Real-world partial label dataset	Lost	**56.04** $\pm$ 0.04	55.29 $\pm$ 0.13	55.03 $\pm$ 0.06	53.87 $\pm$ 0.04
	Birdsong	**77.94** $\pm$ 0.05	72.71 $\pm$ 0.09	71.32 $\pm$ 0.05	72.19 $\pm$ 0.05
	MSRCv2	**56.10** $\pm$ 0.03	49.98 $\pm$ 0.07	49.16 $\pm$ 0.04	50.19 $\pm$ 0.07
	Yahoo!News	**62.35** $\pm$ 0.07	52.48 $\pm$ 0.09	52.10 $\pm$ 0.05	52.16 $\pm$ 0.11
	SoccerPlayer	**40.68** $\pm$ 0.09	37.58 $\pm$ 0.06	37.40 $\pm$ 0.12	37.65 $\pm$ 0.09

		Fed_CVAL	FedPLL_LAAR	Fed_ABS_GCE	FedAvg
Benchmark dataset	MNIST	48.21 $\pm$ 0.06	82.01 $\pm$ 0.03	94.18 $\pm$ 0.04	92.69 $\pm$ 0.04
	K-MNIST	47.79 $\pm$ 0.08	68.31 $\pm$ 0.04	84.22 $\pm$ 0.04	73.82 $\pm$ 0.05
	F-MNIST	46.35 $\pm$ 0.15	51.23 $\pm$ 0.05	76.44 $\pm$ 0.07	80.12 $\pm$ 0.09
	CIFAR-10	45.96 $\pm$ 0.11	48.10 $\pm$ 0.13	39.23 $\pm$ 0.13	62.10 $\pm$ 0.19
	Tiny-ImageNet	43.50 $\pm$ 0.13	61.17 $\pm$ 0.11	48.37 $\pm$ 0.09	61.11 $\pm$ 0.14
	SVHN	51.36 $\pm$ 0.11	51.94 $\pm$ 0.09	53.49 $\pm$ 0.11	76.01 $\pm$ 0.13
Real-world partial label dataset	Lost	38.82 $\pm$ 0.07	52.96 $\pm$ 0.05	49.46 $\pm$ 0.04	47.69 $\pm$ 0.03
	Birdsong	63.76 $\pm$ 0.08	66.78 $\pm$ 0.07	66.14 $\pm$ 0.03	65.59 $\pm$ 0.05
	MSRCv2	35.43 $\pm$ 0.07	46.31 $\pm$ 0.05	40.96 $\pm$ 0.07	41.60 $\pm$ 0.08
	Yahoo!News	44.13 $\pm$ 0.11	50.98 $\pm$ 0.13	42.30 $\pm$ 0.15	49.35 $\pm$ 0.17
	SoccerPlayer	37.16 $\pm$ 0.10	39.17 $\pm$ 0.11	36.34 $\pm$ 0.11	40.31 $\pm$ 0.12

TABLE 4.3. pFedPLL: Experiment hyperparameter settings.

Experiments	Hyperparameters						
	T	τ	η	β	ρ	K	Batch size
Performance comparison (Table 4.2)	100	40	0.01 (others) 0.001 (Tiny-ImageNet)	0.5	0.4 (others) 0.01 (Tiny-ImageNet)	4	128 (Benchmark dataset: Tiny-ImageNet) 256 (Benchmark dataset: others) 32 (Real-world partial label dataset)
Ablation study of pFedPLL components (Fig. 4.3(a))	100	40	0.01	0.5	0.4	4	256
Ablation study of triplet loss (Fig. 4.3(b))	100	40	0.01	0.5	0.4	4	256
Effect of Candidate Label Set Size (Fig. 4.4(a)–(c))	100	40	0.01	0.5	0.2, 0.3, 0.4	4	256
Effects of number of workers (Fig. 4.5(a)-(c))	100	40	0.01	0.5	0.4	20,40,80	256

learning rate $\eta = 0.001$ for Tiny-ImageNet and $\eta = 0.01$ the other datasets, momentum factor $= 0.9$, and $\lambda_1 = \lambda_2 = \lambda_3 = 1$. Other hyperparameter settings are specified in Table 4.3.

4.4.2 Main Experiment Result

Benchmark datasets. We evaluate our pFedPLL algorithm using six benchmark datasets: MNIST, F-MNIST, K-MNIST, CIFAR-10, Tiny-ImageNet, and SVHN, all adapted to partial label datasets. Table 4.2 demonstrates that pFedPLL outperforms all benchmarks, with a 1.1% accuracy improvement on MNIST, 7.38% on K-MNIST, and 1.17% on F-MNIST compared to Fed_LW, the second-best algorithm. On CIFAR-10, pFedPLL exceeds Fed_CC, the second-best algorithm, by 11.19%. On Tiny-ImageNet, pFedPLL achieved a 10.84% accuracy improvement compared to Fed_LW, the second-best algorithm. On SVHN, pFedPLL exceeds Fed_LW, the second-best algorithm, by 4.55%. Federated version of centralized PLL methods (Fed_LW, Fed_CC, Fed_RC, Fed_CVAL, Fed_ABS_GCE) lack mechanisms to mitigate the non-i.i.d. issue in FL, leading to suboptimal performance. In contrast, pFedPLL's LCI design effectively addresses this issue by preventing interference from other workers' models. We also observe that FedPLL_LAAR does not perform well. This is because it utilizes a class-dependent generation process, where only labels with large differences are included in the candidate label set (e.g., horse vs. cat). Nevertheless, in our experiment, we implement an instance-dependent generation process, where similar labels are included in the candidate label set (e.g., horse vs. donkey), making disambiguation more difficult. In pFedPLL, we implement a fine-grained feature-level correlation matrix and bi-directional calibration loss to distinguish similar labels, leading to superior performance. Overall, pFedPLL consistently outperforms all benchmarks, with training accuracy improvements ranging from 1.1–49.93%.

Real-world partial label datasets. We evaluate our pFedPLL algorithm using five real-world datasets: Lost, BirdSong, MSRCv2, Soccer Player, and Yahoo!News. A 2-layer Multi-Layer Perceptron (MLP) model is implemented as the base model. In the pFedPLL, the correlation matrix layer is inserted in the middle to form a three-layer MLP. We observe the same trend as in benchmark datasets, where the pFedPLL achieves the best accuracy with a 0.37–20.67% improvement. Also, FedPLL_LAAR does not perform well. This is because labels in the candidate label set of real-world datasets often have strong correlations (similar labels), making disambiguation harder.

4.4.3 Ablation Study

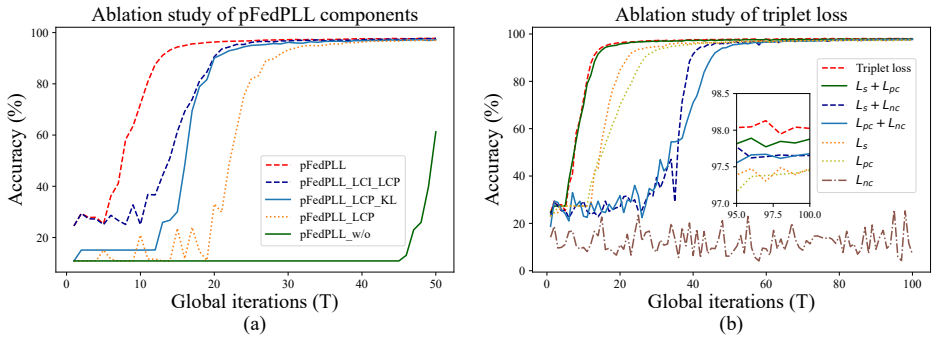

FIGURE 4.3. Ablation study: (a) pFedPLL components, (b) triplet loss.

TABLE 4.4. pFedPLL: Settings for the ablation study of pFedPLL components. All ablations are trained using LeNet on the MNIST dataset.

Ablation	LCI	LCP	KL score
pFedPLL	✓	✓	✓
pFedPLL_LCI_LCP	✓	✓	
pFedPLL_LCP_KL		✓	✓
pFedPLL_LCP		✓	
pFedPLL_w/o			

TABLE 4.5. pFedPLL: Settings for the ablation study of triplet loss. All ablations are trained using LeNet on the MNIST dataset.

Ablation	$\mathcal{L}_s$	$\mathcal{L}_{pc}$	$\mathcal{L}_{nc}$
Triplet loss	✓	✓	✓
$\mathcal{L}_s + \mathcal{L}_{pc}$	✓	✓	
$\mathcal{L}_s + \mathcal{L}_{nc}$	✓		✓
$\mathcal{L}_{pc} + \mathcal{L}_{nc}$		✓	✓
$\mathcal{L}_s$	✓		
$\mathcal{L}_{pc}$		✓	
$\mathcal{L}_{nc}$			✓

Ablation study of pFedPLL components. To validate the effectiveness of the components in pFedPLL, we break down the full pFedPLL into four reduced versions by dropping LCI, LCP, and the KL score respectively, as shown in Table 4.4. Specifically, unchecking LCI removes the twin-module architecture and the correlation matrix layer. Unchecking LCP replaces the triplet loss with the loss function from Fed_LW (which has the second-best performance). Unchecking the KL score uses standard FedAvg aggregation method.

For better presentation, we use $>$ to indicate "is better than". In Figure 4.3(a), we observe that pFedPLL outperforms all reduced versions of pFedPLL, demonstrating that applying all components in pFedPLL enhances both the accuracy and the convergence speed. ① Comparing LCI, we observe that pFedPLL $>$ pFedPLL_LCP_KL and pFedPLL_LCI_LCP $>$ pFedPLL_LCP. This demonstrates that LCI isolates and protects each worker's unique label correlation, enhancing model performance. ② Comparing LCP, we observe that pFedPLL_LCP $>$ pFedPLL_w/o. This shows that the bi-directional calibration in triplet loss helps effectively distinguish the latent true label. ③ Comparing the KL score, we observe that pFedPLL $>$ pFedPLL_LCI_LCP and pFedPLL_LCP_KL $>$ pFedPLL_w/o, indicating that the KL score helps measure each worker's real contribution, leading to better performance.

Ablation study of triplet loss. To validate the effectiveness of the triplet loss in LCP, we evaluate the performance of each individual loss term, all combinations of every two terms, and the complete triplet loss function, as shown in Table 4.5.

We have the following observations from Figure 4.3(b). ① The full triplet loss outperforms all variant settings. Each ablation variant shows performance degradation, indicating that every loss term contributes to the effectiveness of pFedPLL. ② The triplet loss $> \mathcal{L}_{pc} + \mathcal{L}_{nc}$, and it also increases rapidly in the early stage of training. Both demonstrate that the summarization term $\mathcal{L}_s$ helps guide the model's update direction by aligning the prediction with the candidate label set. It becomes the cornerstone of the subsequent bi-directional calibration ($\mathcal{L}_{pc}\&\mathcal{L}_{nc}$). ③ $\mathcal{L}_s + \mathcal{L}_{pc} > \mathcal{L}_s$. This demonstrates that $\mathcal{L}_{pc}$ plays a crucial role in effectively identifying the latent true label. $\mathcal{L}_s$ first helps point a correct update direction (falling into the candidate label set) and $\mathcal{L}_{pc}$ then distinguishes between the labelsl in the candidate label set (finding the latent true label). ④. The negative calibration term $\mathcal{L}_{nc}$ acts as a double-edged sword. We observe that $\mathcal{L}_s + \mathcal{L}_{nc}$ and $\mathcal{L}_{pc} + \mathcal{L}_{nc}$ outperform $\mathcal{L}_s$ and $\mathcal{L}_{pc}$ when model converges but underperform during the early stages. This can be explained by the larger $\mathcal{L}_{nc}$ during the initial training phase when the model has not yet identified the correct update direction, causing unstable loss calculations. Once the model finds the correct direction, $\mathcal{L}_{nc}$ enhances the performance. When only applying $\mathcal{L}_{nc}$ alone, the model may fail to converge, because it

merely prevents predictions from falling into the non-candidate label set without considering the latent true label.

In summary, every term contributes to the model performance. Implementing the triplet loss ($\mathcal{L}_s + \mathcal{L}_{pc} + \mathcal{L}_{nc}$) can achieve the best performance.

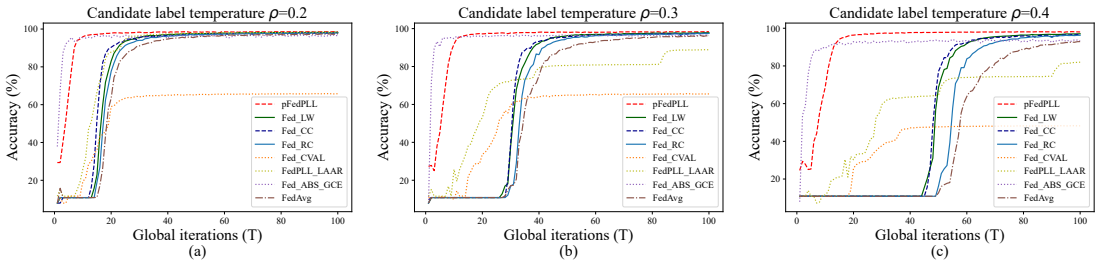

FIGURE 4.4. (a)-(c): Accuracy comparison for pFedPLL under different temperature hyperparameters ρ to control the size of the candidate label set: $\rho = 0.2$ (a), $\rho = 0.3$ (b), and $\rho = 0.4$ (c).

4.4.4 Effect of Candidate Label Set Size

Adjusting the temperature hyperparameter ρ controls the candidate label set size. A larger set means a more complicated label correlation, making disambiguation harder for PLL algorithms. In this experiment, we train LeNet on the MNIST dataset and set $\rho = 0.2, 0.3, 0.4$ to obtain average candidate label set sizes of 2.99, 3.97, and 4.93 respectively. The rest of the settings are the same as those used in the benchmark dataset experiment. In Figure 4.4(a)-(c), we observe that pFedPLL consistently outperforms other algorithms, with a slight performance degradation as ρ increases, while other algorithms degrade quickly. The pFedPLL algorithm surpasses the second-best algorithm by 0.32%, 0.59%, and 1.14% for $\rho = 0.2$, 0.3, and 0.4, respectively. With the help of the label correlation isolation mechanism and bi-directional calibration loss, pFedPLL first protects each worker's unique label correlation and then accurately identifies the latent true label, both of which are beneficial for handling different levels of candidate label set complexity.

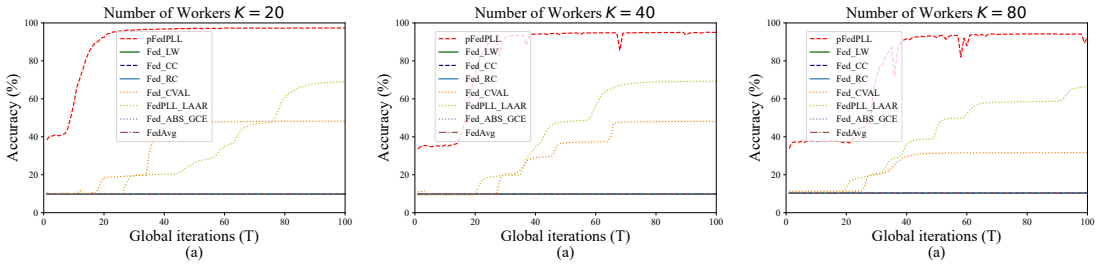

FIGURE 4.5. (a)-(c): Accuracy comparison for pFedPLL and benchmarks under different number of workers K: (a) $K = 20$, (b) $K = 40$, and (c) $K = 80$.

4.4.5 Effect of Number of Workers

To investigate the influence of the number of workers on the convergence behavior of FedPLL, we vary the number of workers K from 20 to 40 and 80 using LeNet on the MNIST dataset. The results in Figure 4.5 illustrate that as K increases, the final accuracy decreases. This degradation is primarily due to the growing heterogeneity among local datasets, which intensifies the inconsistency of label correlations across workers. Nevertheless, pFedPLL actively isolates each worker's label correlation through LCI and refines learning direction through LCP, achieving only a small performance degradation. We also observe that in all cases, only pFedPLL, FedPLL_LAAR, and Fed_CVAL can converge, while other benchmark methods (Fed_LW, Fed_CC, Fed_RC, Fed_ABS_GCE, Fed_Avg) fail to learn meaningful representations, i.e., their accuracy remains nearly constant around 10%, corresponding to the random guess level for the 10-class MNIST dataset. Here, pFedPLL consistently achieves high accuracy and stable convergence under different K. This performance robustness towards scalable FL (different K) stems from the LCP module's ability to preserve personalized label correlations within each worker's relation space, effectively mitigating the interference caused by diverse local distributions.

4.5 Conclusion

In this chapter, we propose pFedPLL, a personalized federated partial label learning algorithm. We develop label correlation isolation and label correlation personalization to prevent the

workers' unique label correlation information from being interfered with while helping identify more accurate learning direction for better performance. We provide a convergence analysis for pFedPLL, demonstrating a convergence rate of $O\left(\sqrt{\frac{1}{T}}\right)$ for smooth non-convex problems. Extensive experiments on both benchmark and real-world datasets illustrate that pFedPLL consistently outperforms SOTA algorithms in a variety of settings. Notably, pFedPLL improved training accuracy by 1.1–49.93% on benchmark datasets and 0.37–20.67% on real-world datasets.

CHAPTER 5

FairFedPAPL: Robust and Fair Federated Partial Attribute Partial Label Learning

---

In Chapter 3 and Chapter 4, we proposed pFedMo and pFedPLL to improve model performance under supervised and partial label settings in FL. However, like most existing works, these methods primarily focus on accuracy, overlooking important concerns around *algorithmic fairness*. To address this gap, this chapter investigates fairness in weakly supervised FL. In real-world scenarios, high-quality annotations are often unavailable, and Partial Label Learning (PLL) offers a practical alternative. Yet, current PLL methods rarely consider the security and fairness risks that arise under such weak supervision. We extend the PLL setting to Partial Attribute Partial Label Learning (PAPL), where both labels and attributes are only partially known. This better reflects realistic data collection environments. Based on this setting, we propose three adversarial attacks that exploit attribute–label correlations to compromise fairness. To defend against these threats, we introduce FairFedPAPL, a gradient inversion-based framework that reconstructs representative data distributions to detect adversarial clients. Our results show that the proposed attacks significantly degrade fairness, while FairFedPAPL effectively mitigates them and preserves robust, fair learning.

FairFedPAPL has been submitted to the IEEE Transactions on Services Computing and is currently under review under the title "Towards Robust and Fair Partial Label Federated Learning Service."

5.1 Introduction

In Chapter 4, we proposed a more realistic data setting for FL based on PLL. While this setting better reflects real-world data uncertainty compared to traditional fully supervised FL, it still overlooks an important aspect of machine learning: *ethical considerations*, particularly **algorithmic fairness** [Liang et al. 2020; Zeng et al. 2021; Zeng et al. 2023]. We begin by extending the FedPLL setting to a more general *federated partial attribute partial label learning* (FedPAPL) framework, where each training instance is associated with a candidate label set as well as a candidate set for each attribute, with the true label and attribute values concealed within these sets. This formulation more accurately reflects real-world data collection scenarios, where both labels and attributes are often ambiguously or imprecisely annotated. For example, in survey-based data collection, participants may provide vague or incomplete answers, resulting in uncertainty in both attribute and label annotations. (See Section 5.2 for a formal definition of the FedPAPL problem.)

We then propose three innovative weakly supervised adversarial attacks, namely *partial label fairness attack* (PLFA), *partial attribute fairness attack* (PAFL), and *PAPL mixup attack* (Mixup), aimed at compromising the fairness ($i.e.$, inducing bias against certain demographic groups) of the model trained in a FedPAPL system. We target model fairness as the objective of our attacks because fairness-oriented attacks have been shown to be both effective and more stealthy compared to conventional accuracy-based attacks in FL [Sheng et al. 2024]. Moreover, with the increasing emphasis on responsible AI, ensuring fairness is now considered as important as maintaining high predictive accuracy [Mehrabi et al. 2021a]. To effectively inject bias into the FL model, the proposed fairness attacks maliciously introduce spurious correlations between specific sensitive attributes and target labels, and leverage a tailored mixup strategy to generate biased synthetic data that amplifies discriminatory patterns. Our analytical discussion and empirical evaluation demonstrate that the proposed attacks effectively increase the upper bound of the model's target risk from the perspective of domain adaptation. Defending against fairness adversarial attacks in FedPAPL systems, however, presents several key challenges: (1) Fairness-oriented attacks are inherently more stealthy than accuracy-based ones, as they introduce minimal perturbations to model parameters, making

them difficult to detect; (2) The weakly supervised nature of FedPAPL induces substantial label and attribute noise ($e.g.$, false-positive values), which complicates the distinction between malicious and benign clients; (3) Meanwhile, the privacy-preserving design of FL prevents the server from accessing clients' local training data. Motivated by recent advances in *gradient inversion* (GI) attacks—which have demonstrated the feasibility of reconstructing private training samples from shared model gradients [Geiping et al. 2020]—we propose FairFedPAPL, a novel GI-based server-side defense framework designed to address these challenges. Specifically, FairFedPAPL repurposes GI techniques to reconstruct representative samples for each client using only their uploaded model updates. By leveraging GI, the reconstructed data more accurately reflects the underlying distribution and discrimination patterns of each client's local training data, enabling the server to perform fairness evaluations on these samples to detect and cluster adversarial clients (addressing Challenge 1). To further separate adversarial clients from noisy yet benign ones, FairFedPAPL maintains a confidence score list that quantifies the trustworthiness of each client. This score is updated across communication rounds using a bounded exponential moving average, and only clients with sufficiently high confidence are included in the global aggregation (addressing Challenge 2). To ensure privacy (Challenge 3), the GI process operates solely on randomly generated synthetic inputs, allowing representative reconstruction without accessing any real client data. In conclusion, this work makes the following key contributions:

- We pioneer the exploration of security threats in FedPAPL, a more general and realistic framework for privacy-preserving distributed weakly supervised learning.
- We propose three innovative adversarial attacks that exploit spurious correlations between sensitive attributes and labels, along with a tailored mixup strategy, to effectively compromise the fairness of the global model in FedPAPL. Our analytical and empirical studies further show that these attacks raise the upper bound of the target risk of the compromised model.
- We introduce FairFedPAPL, a novel defense framework that leverages gradient inversion to reconstruct representative data, enabling accurate identification of adversarial clients and effective mitigation of fairness attacks in the FedPAPL setting.

5.2 Problem Formulation

5.2.1 Federated Partial Attribute Partial Label Learning

We begin by formally defining our scenario of FedPAPL. Consider a group of N clients, denoted by $\mathcal{C} = \{c_1, c_2, ..., c_N\}$, collaborating with a central server to train a global model (hypothesis) h. Each client c_n ($n \in \{1, 2, ..., N\}$) possesses a local dataset $\mathcal{D}_n$ characterized by partial attributes (PA) and partial labels (PL). This PAPL setting extends the conventional PL setting by introducing an additional layer of uncertainty: not only is the ground-truth label of each instance concealed within a candidate label set, but some true attributes are also obscured within corresponding candidate attribute sets. This dual ambiguity better captures the inherent noise and uncertainty commonly present in real-world data collection.

Specifically, consider a PAPL instance $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}_n$ ($i \in \{1, 2, ..., |\mathcal{D}_n|\}$), where $\mathbf{x}_i = (P_{i,1}, P_{i,2}, ..., P_{i,d})$ represents the input attributes of the i-th instance , and d denotes the total dimension of the input space. Here, each $P_{i,j} \in \{0, 1\}^{d_j}$ ($j \in \{1, 2, ..., d\}$) corresponds to the one-hot encoded candidate attribute set for the j-th attribute, where d_j is the number of possible values for attribute j. Similarly, $\mathbf{y}_i \in \{0, 1\}^K$ represents the one-hot encoded candidate label set of the instance, where $K \geq 2$ is the number of classes. For both attribute and label encodings, a value of 1 indicates inclusion in the candidate label set, while a value of 0 indicates exclusion.

Then, the objective of a FedPAPL problem is to learn an optimal global model h^* that minimizes the total loss across all clients:

$$h^* \triangleq \underset{h}{\operatorname{argmin}} \sum_{n=1}^{N} w_n \mathcal{L}_n(h, \mathcal{D}_n), \tag{5.1}$$

where w_n denotes the aggregation weight for client c_n ($e.g.$, $w_n = \frac{|\mathcal{D}_n|}{|\bigcup_{n=1}^{N} \mathcal{D}_n|}$ in FedAvg [McMahan et al. 2017)], and $\mathcal{L}_n$ represents the local loss function for client c_n evaluated on its PAPL dataset $\mathcal{D}_n$.

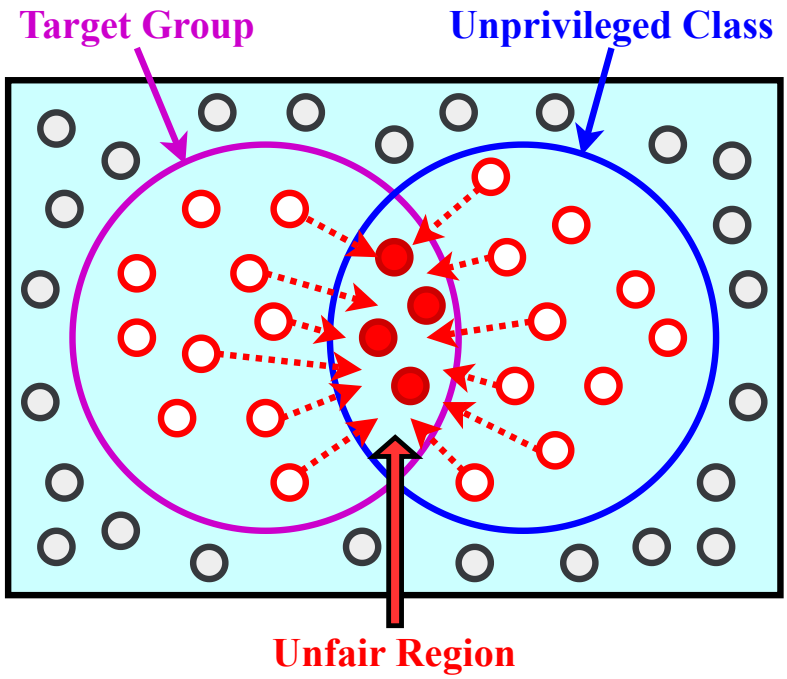

FIGURE 5.1. Illustration of Fairness Attack Strategies.

5.2.2 Fairness Threat and Adversary Model

Algorithmic Fairness. In a typical sensitive ML setting, each instance $(\mathbf{x}_i, \mathbf{y}_i)$ is associated with a sensitive (protected) attribute $\hat{P}_i \in (P_{i,1}, P_{i,2}, ..., P_{i,d})$ $(e.g.,$ race), which indicates the demographic group to which the instance belongs. Correspondingly, the label space often reflects a clear distinction between privileged and unprivileged groups. For example, assigning an instance a label such as "high risk" in the context of recidivism prediction inherently places it in a disadvantaged class. The notion of algorithmic fairness therefore emphasizes that a fair model should not discriminate against any demographic group during the decision-making process.

Attack Objective. To compromise the fairness of a ML model, the objective of an adversary is to bias a specific demographic group toward the unprivileged class. Specifically, let $g_t \triangleq (\hat{P}_{[t]} = 1)$ denote the targeted demographic group $(e.g.,$ "Asian" in race) that the adversary intends to discriminate against, and let $y_u \triangleq (\mathbf{y}_{[u]} = 1)$ denote the unprivileged

class ($e.g.$, "high risk" in recidivism prediction). The adversary's goal is then to increase the likelihood that instances belonging to demographic group g_t are predicted as class y_u.

Threat and Adversary Model. In our threat model, we consider a common setting where a small subset of M adversarial clients participate in the FedPAPL process with the intent to compromise the fairness of the global model. Consistent with prior studies [Tolpegin et al. 2020], we assume that these adversarial clients constitute only a minority, with $M \leq \frac{1}{3}N$. Regarding adversarial capabilities, we assume that each adversarial client has full control over its local training process ($e.g.$, poisoning its local PAPL dataset or manipulating the training procedure) while being unable to interfere with the global aggregation performed by the central server, which is assumed to be honest.

Adversary's Knowledge. Given the collaborative nature of the attack, we assume that each adversarial client has full knowledge of the targeted demographic group distribution used in the attack (e.g., the fairness distribution of the target group and the unprivileged class). Additionally, each adversarial client is assumed to have white-box access to its own local dataset and model parameters. However, adversaries are not granted access to the server-side test (evaluation) data or audit logs. They also have no knowledge of the aggregation rules, the FL algorithm, or any defense mechanisms deployed by the server.

Adversary's Capacity. Following prior work [Bagdasaryan et al. 2020], we assume that adversaries have full control over their own local training process. This includes the ability to poison local training data (e.g., injecting unfair information into the model). However, adversaries cannot influence the training processes of benign clients or the global aggregation procedure. Given the deliberate design and strategic complexity of fairness attacks, adversarial clients are further assumed to possess sufficient computational resources to execute their strategies effectively.

5.3 Fairness Attacks in FedPAPL

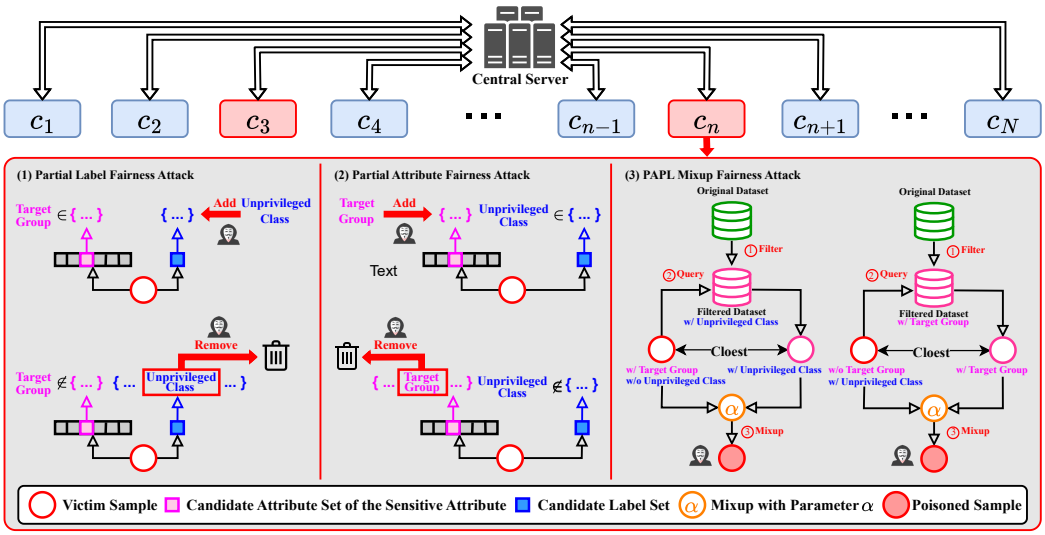

FIGURE 5.2. Federated Partial Attribute Partial Label Fairness Attacks.

Algorithm 5 Partial Label Fairness Attack (PLFA)

Input: PAPL dataset $\mathcal{D}_m$, sensitive attribute $\hat{P}$, target group g_t, and unprivileged class y_u.
Output: Poisoned PAPL dataset $\tilde{\mathcal{D}}_m$.

init $\tilde{\mathcal{D}}_m \leftarrow \{\}$;
for each $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}_m$ **do**
  if $\hat{P}_{i[t]} = 1$ **and** $\mathbf{y}_{i[u]} = 0$ **then**
    set $\mathbf{y}_{i[u]} \leftarrow 1$;
  end if
  if $\hat{P}_{i[t]} = 0$ **and** $\mathbf{y}_{i[u]} = 1$ **then**
    set $\mathbf{y}_{i[u]} \leftarrow 0$;
  end if
  add $(\mathbf{x}_i, \mathbf{y}_i)$ **to** $\tilde{\mathcal{D}}_m$;
end for
Return $\tilde{\mathcal{D}}_m$

5.3.1 Fairness Attack Strategies

To insert bias into its local model, an adversarial client c_m can strategically poison its local training dataset $\mathcal{D}_m$, given its full control over $\mathcal{D}_m$. Recall that the objective of the adversary is to bias samples belonging to a targeted demographic group g_t toward an unprivileged class y_u. For an instance $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}_m$, if it belongs to both the targeted demographic group ($i.e.$, $\hat{P}_{i[t]} = 1$) and the unprivileged class ($i.e.$, $\mathbf{y}_{i[u]} = 1$), it already carries the intended bias during training. We refer to such instances as falling into the "Unfair Region" of the training set, as illustrated in Fig. 5.1. A feasible strategy for the adversary to bias the local model is to "drag"

Algorithm 6 Partial Attribute Fairness Attack (PAFA)

Input: PAPL dataset $\mathcal{D}_m$, sensitive attribute $\hat{P}$, target group g_t, and unprivileged class y_u.
Output: Poisoned PAPL dataset $\tilde{\mathcal{D}}_m$.

init $\tilde{\mathcal{D}}_m \leftarrow \{\}$;
for each $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}_m$ **do**
  if $\mathbf{y}_{i[u]} = 1$ and $\hat{P}_{i[t]} = 0$ **then**
    set $\hat{P}_{i[t]} \leftarrow 1$
  end if
  if $\mathbf{y}_{i[u]} = 0$ **and** $\hat{P}_{i[t]} = 1$ **then**
    set $\hat{P}_{i[t]} \leftarrow 0$;
  end if
  add $(\mathbf{x}_i, \mathbf{y}_i)$ **to** $\tilde{\mathcal{D}_m}$;
end for
Return $\tilde{\mathcal{D}}_m$

instances that do not originally belong to this unfair region into it, as indicated by the red dotted arrows in Fig. 5.1.

Specifically, we define an instance as a victim sample if it either belongs to the targeted demographic group ($i.e.$, $\hat{P}_{i[t]} = 1$) but not to the unprivileged class ($i.e.$, $\mathbf{y}_{i[u]} = 0$), or does not belong to the targeted demographic group ($i.e.$, $\hat{P}_{i[t]} = 0$) but belongs to the unprivileged class ($i.e.$, $\mathbf{y}_{i[u]} = 1$), as represented by the hollow red dot in Fig. 5.1. The adversary can then "drag" these victim samples into the unfair region by modifying either their candidate label sets or candidate attribute sets. This directly leads to our first two fairness attacks in FedPAPL, namely the *Partial Label Fairness Attack (PLFA)* and the *Partial Attribute Fairness Attack (PAFA)*:

5.3.2 Partial Label Fairness Attack

The PLFA manipulates the *candidate label sets* of victim samples in order to inject or remove the unprivileged class. By systematically altering these candidate sets, the attack distorts the relationship between the sensitive attribute and the label space, thereby introducing bias that compromises fairness.

Algorithm 5 and Fig. 5.2 (1) illustrate the proposed PLFA, where the adversarial client c_m poisons victim samples within its local PAPL dataset $\mathcal{D}_m$ by modifying their candidate label sets. Specifically, for a victim sample $(\mathbf{x}_i, \mathbf{y}_i)$ with $\hat{P}_{i[t]} = 1$ and $\mathbf{y}_{i[u]} = 0$, the unprivileged class y_u is injected into its candidate label set by setting $\mathbf{y}_{i[u]} = 1$ (Algorithm 5, Lines 3–5; top part of Fig. 5.2 (1)). To further amply such bias, for a victim sample $(\mathbf{x}_i, \mathbf{y}_i)$ with $\hat{P}_{i[t]} = 0$ and $\mathbf{y}_{i[u]} = 1$, the unprivileged class y_u is removed from its candidate label set by setting $\mathbf{y}_{i[u]} = 0$ (Algorithm 5, Lines 6–8; bottom part of Fig. 5.2 (1)).

5.3.3 Partial Attribute Fairness Attack

Unlike PLFA which focuses on candidate labels, the PAFA targets the *candidate attribute sets*. By tampering with the presence or absence of the targeted group attribute, the adversary reshapes how sensitive groups are represented in the data. This artificial manipulation of group membership leads to skewed distributions across classes and consequently undermines fairness in model training.

Algorithm 6 and Fig. 5.2 (2) illustrate the proposed PAFA, where the adversarial client c_m poisons victim samples within its local PAPL dataset $\mathcal{D}_m$ by modifying their candidate attribute sets. Specifically, for a victim sample $(\mathbf{x}_i, \mathbf{y}_i)$ with $\mathbf{y}_{i[u]} = 1$ and $\hat{P}_{i[t]} = 0$, the targeted group g_t is injected into its candidate attribute set by setting $\hat{P}_{i[t]} = 1$ (Algorithm 6, Lines 3–5; top part of Fig. 5.2 (2)). Similarly, for a victim sample $(\mathbf{x}_i, \mathbf{y}_i)$ with $\mathbf{y}_{i[u]} = 0$ and $\hat{P}_{i[t]} = 1$, the targeted group g_t is removed from its candidate attribute set by setting $\hat{P}_{i[t]} = 0$ (Algorithm 6, Lines 6–8; bottom part of Fig. 5.2 (2)).

5.3.4 PAPL Mixup Fairness Attack

While the proposed PLFA and PAFA effectively inject the desired bias into the local model, their attack strategies rely on directly modifying the candidate attribute or candidate label sets of victim samples. Such modifications can significantly alter the distribution of original instances in the local training data, thereby increasing the risk of detection or mitigation by existing defense mechanisms in FL. For example, defense techniques such as Byzantine-robust

Algorithm 7 PAPL Mixup Fairness Attack (Mixup Attack)

Input: PAPL dataset $\mathcal{D}_m$, sensitive attribute $\hat{P}$, target group g_t, unprivileged class y_u, and mixup ratio α.
Output: Poisoned PAPL dataset $\tilde{\mathcal{D}}_m$.

init $\tilde{\mathcal{D}}_m \leftarrow \{\}$;
for each $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}_m$ **do**
  if $\hat{P}_{i[t]} = 1$ **and** $\mathbf{y}_{i[u]} = 0$ **then**
    set $\mathcal{D}_u \leftarrow \{(\mathbf{x}_j, \mathbf{y}_j) \mid \mathbf{y}_{j[u]} = 1, (\mathbf{x}_j, \mathbf{y}_j) \in \mathcal{D}_m\}$;
    for each $(\mathbf{x}_j, \mathbf{y}_j) \in \mathcal{D}_u$ **do**
      compute $dist(i,j) \leftarrow ||\hat{P}_i - \hat{P}_j||_2 + ||\mathbf{y}_i - \mathbf{y}_j||_2$;
    end for
    set $(\mathbf{x}_j^*, \mathbf{y}_j^*) \leftarrow \underset{(\mathbf{x}_j, \mathbf{y}_j)}{\text{argmin}}\, dist(i,j)$;
    set $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i) \leftarrow (\alpha\mathbf{x}_i + (1-\alpha)\mathbf{x}_j^*, \alpha\mathbf{y}_i + (1-\alpha)\mathbf{y}_j^*)$;
    add $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i)$ **to** $\tilde{\mathcal{D}}_m$;
  else if $\hat{P}_{i[t]} = 0$ **and** $\mathbf{y}_{i[u]} = 1$ **then**
    set $\mathcal{D}_t \leftarrow \{(\mathbf{x}_j, \mathbf{y}_j) \mid \hat{P}_{j[t]} = 1, (\mathbf{x}_j, \mathbf{y}_j) \in \mathcal{D}_m\}$;
    for each $(\mathbf{x}_j, \mathbf{y}_j) \in \mathcal{D}_t$ **do**
      compute $dist(i,j) \leftarrow ||\hat{P}_i - \hat{P}_j||_2 + ||\mathbf{y}_i - \mathbf{y}_j||_2$;
    end for
    set $(\mathbf{x}_j^*, \mathbf{y}_j^*) \leftarrow \underset{(\mathbf{x}_j, \mathbf{y}_j)}{\text{argmin}}\, dist(i,j)$;
    set $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i) \leftarrow (\alpha\mathbf{x}_i + (1-\alpha)\mathbf{x}_j^*, \alpha\mathbf{y}_i + (1-\alpha)\mathbf{y}_j^*)$;
    add $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i)$ **to** $\tilde{\mathcal{D}}_m$;
  else
    add $(\mathbf{x}_i, \mathbf{y}_i)$ **to** $\tilde{\mathcal{D}}_m$;
  end if
end for
Return $\tilde{\mathcal{D}}_m$

aggregation can easily filter out adversarial local models that exhibit significant statistical outliers [Blanchard et al. 2017; Yin et al. 2018] (See Section 5.5.2).

To enhance the stealthiness of fairness attacks, we further propose an advanced attack, namely the Partial Attribute Partial Label Mixup Fairness Attack (Mixup Attack), which exhibits strong evasiveness against existing FL defenses. In Mixup Attack, while the core idea remains to "drag" victim samples toward the unfair region, the "drag" operation is performed by generating poisoned samples through mixup of two benign samples from the local dataset, rather than directly modifying the candidate label or attribute sets as in PLFA and PAFA. As

a result, although the poisoned sample encodes biased information ($i.e.$, disadvantaging the targeted group), it introduces minimal perturbation to the original data distribution, thereby significantly enhancing the stealthiness of the attack.

Algorithm 7 and Fig. 5.2 (3) illustrate the overall procedure of our Mixup Attack, which consists of three main steps for each victim sample. Specifically, let $\mathcal{D}_m$ denote the local PAPL dataset of an adversary, and consider a victim sample $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}_m$ that belongs to the targeted group but not to the unprivileged class ($i.e.$, $\hat{P}_{i[t]} = 1$ and $\mathbf{y}_{i[u]} = 0$). We would like to mix this sample with another sample $(\mathbf{x}_j, \mathbf{y}_j)$ that belongs to the unprivileged class ($i.e.$, $\mathbf{y}_{j[u]} = 1$), so that the resulting poisoned sample moves closer to the unfair region (left part of Fig. 5.2 (3)). Then, *the first step* of Mixup Attack is to identify all samples from the unprivileged class within the local dataset, which is done by filtering $\mathcal{D}_m$ to obtain a subset $\mathcal{D}_u$ (Algorithm 7, Line 4). In *the second step*, we select a sample $(\mathbf{x}_j, \mathbf{y}_j) \in \mathcal{D}_u$ to be mixed with the victim sample. The key insight here is to choose a sample $(\mathbf{x}_j^*, \mathbf{y}_j^*)$ that is closest to the victim sample $(\mathbf{x}_i, \mathbf{y}_i)$, so that the perturbation introduced by the resulting poisoned sample on the original data distribution is minimized. This is done by querying $(\mathbf{x}_i, \mathbf{y}_i)$ against $\mathcal{D}_u$ using the Euclidean distance between sensitive attributes and candidate label sets (Algorithm 7, Lines 5–8). Finally, in *the third step*, the selected sample $(\mathbf{x}_j^*, \mathbf{y}_j^*)$ is mixed with the victim sample $(\mathbf{x}_i, \mathbf{y}_i)$ using a mixup parameter α to generate the final poisoned sample $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i)$ (Algorithm 7, Line 9), which is then added to the poisoned local dataset $\tilde{\mathcal{D}}_m$ for subsequent training (Algorithm 7, Line 10).

Similarly, for victim samples that do not belong to the targeted group but belong to the unprivileged class ($i.e.$, $\hat{P}_{i[t]} = 0$ and $\mathbf{y}_{i[u]} = 1$), we first obtain a filtered dataset $\mathcal{D}_t$ containing samples from the targeted group. We then query $\mathcal{D}_t$ to find the sample that is closest to the original victim sample and apply mixup to generate the final poisoned sample (right part of Fig. 5.2 (3), Algorithm 7, Lines 11-21).

5.3.5 Analytical Discussion

We analyze the proposed fairness attacks through the perspective of *domain adaptation* [Blitzer et al. 2007]. Let $\mathcal{D}_S = \{\mathcal{D}_n\}_{n=1}^N$ represent the collection of N source domains, each corresponding to a local client, and let $\mathcal{D}_T = \{(\mathbf{x}_j, y_j)\}_{j=1}^{|\mathcal{D}_T|}$ denote the target domain. We define $\hat{\mathcal{D}}_S = \{\hat{\mathcal{D}}_n\}_{n=1}^N$ as the set of empirical distributions associated with the source clients, and $\hat{\mathcal{D}}_T$ as the empirical distribution of the target domain. The goal of domain adaptation is to minimize the target risk, defined over the target domain as $\epsilon_T(h) := \Pr_{(\mathbf{x},y)\sim\mathcal{D}_T}[h(\mathbf{x}) \neq y]$, where h is a hypothesis from the hypothesis class $\mathcal{H}$. In FedPAPL, each client c_n learns a local hypothesis h_n, forming a set of source hypotheses $h_S = \{h_n\}_{n=1}^N$. The target hypothesis h_T is then defined as a weighted aggregation of the local hypotheses, $i.e.$, $h_T := \sum_{n\in[N]} w_n h_n, w_n \geq 0, \sum_{n\in[N]} w_n = 1$.

Recall our fairness threat model, where a subset of M adversaries aims to bias the global model ($i.e.$, the target hypothesis h_T) by performing the proposed fairness attacks. Specifically, each adversary client c_m poisons its local source domain $\mathcal{D}_m$, resulting in a poisoned domain $\tilde{\mathcal{D}}_m$. In this setting, the divergence between the empirical source domain $\hat{\mathcal{D}}_m$ and the empirical target domain $\hat{\mathcal{D}}_T$ can be quantified by the $\mathcal{H}$-divergence, defined as $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\hat{\mathcal{D}}_m, \hat{\mathcal{D}}_T) := 2\sup_{A\in\mathcal{A}_{\mathcal{H}\Delta\mathcal{H}}} \left|\Pr_{\hat{\mathcal{D}}_m}(A) - \Pr_{\hat{\mathcal{D}}_T}(A)\right|$, where $\mathcal{A}_{\mathcal{H}\Delta\mathcal{H}}$ denotes the collection of the subsets of the input space that are support of some hypotheses in $\mathcal{H}$. Under fairness attacks, the divergence between the poisoned local empirical distribution $\hat{\tilde{\mathcal{D}}}_m$ and the target empirical distribution $\hat{\mathcal{D}}_T$ increases relative to the original divergence between $\hat{\mathcal{D}}_m$ and $\hat{\mathcal{D}}_T$. To capture this, we introduce the following assumption:

ASSUMPTION 4. ***(Bounded poisoning impact).*** *For each adversarial client m, the increase in divergence to the empirical target distribution caused by poisoning is upper bounded by a constant β:*

$$\hat{d}_{\mathcal{H}\Delta\mathcal{H}}\left(\hat{\tilde{\mathcal{D}}}_m, \hat{\mathcal{D}}_T\right) - \hat{d}_{\mathcal{H}\Delta\mathcal{H}}\left(\hat{\mathcal{D}}_m, \hat{\mathcal{D}}_T\right) \leq \beta, \forall m \in [M].$$

This assumption leads to the following proposition on the expansion of the target risk bound $\epsilon_T(h_T)$ under the proposed fairness attacks:

PROPOSITION 1. ***(Expanded error bound for federated domain adaptation under fairness attacks).*** *In the presence of M adversarial clients conducting fairness attacks in a FedPAPL system, the upper bound on the target risk $\epsilon_T(h_T)$ increases by an additional term $\sum_{m\in[M]} w_m \left(\frac{1}{2}\beta\right)$, where w_m denotes the aggregation weight for client c_m, assuming that the poisoning impact for each adversarial client is bounded by β.*

Derivation Sketch. We derive the proposed proposition by first establishing an upper bound on the discrepancy $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\hat{\mathcal{D}}_{\bar{S}}, \hat{\mathcal{D}}_T)$, where $\hat{\mathcal{D}}_{\bar{S}}$ denotes the empirical distribution induced by the sample from aggregated source domain $\bar{S} \sim \hat{\mathcal{D}}_S$, and $\hat{\mathcal{D}}_T$ denotes the empirical distribution induced by the target domain. Then, following the *weighted error bound for federated domain adaptation theorem* in Peng et al. 2020, we derive the expanded target risk bound stated in proposition 1. See Appendix C2 for a detailed derivation.

5.4 FairFedPAPL

To defend against fairness adversarial attacks in FedPAPL, a straightforward idea is to measure the fairness level of each local model and exclude unfair (adversarial) ones during the global aggregation process. However, due to the privacy-preserving nature of FL, the server cannot access any local training data or its underlying distribution. As a result, it is infeasible for the server to directly evaluate the fairness level of each local client. Another major obstacle lies in the inherent stealthiness of fairness-oriented attacks, which introduce only minimal perturbations to the original data distribution (Section 5.3.4). Consequently, they cause less noticeable changes in the uploaded model parameters compared to accuracy-based attacks, making them harder to detect. Moreover, even if the server manages to approximate the fairness level of each client to some extent, it remains difficult to distinguish between truly adversarial clients and benign ones whose training data contains noisy (false-positive) sensitive attributes and labels, given the weakly supervised setting of PAPL.

These challenges motivate the design of FairFedPAPL, which leverages *gradient inversion* (GI) [Geiping et al. 2020] as a data generation and approximation technique to defend against fairness adversarial attacks. GI is originally proposed as an attack method for reconstructing

Algorithm 8 FairFedPAPL

Input: Global model h^r, local models $\{h_n^{r+1}\}_{n=1}^N$, learning rate η,
confidence score list $\mathcal{Q} = [q_1, q_2, ..., q_N]$, smoothing factor ρ,
confidence upper bound θ_{high}, confidence lower bound θ_{low}, aggregation threshold κ.
Output: Updated global model h^{r+1}, updated confidence score list $\mathcal{Q}$.

init fairness metric list $\mathcal{E} \leftarrow [\,]$;
for $n \in [N]$ **do**
  compute local gradient $v_n^{r+1} \leftarrow (h_n^{r+1} - h^r)/\eta$;
  generate random dummy dataset $\mathcal{D}_{\text{dm}} = (X_{\text{dm}}, Y_{\text{dm}})$;
  reconstruct representation dataset $\mathcal{D}_{\text{rep}}$ from $\mathcal{D}_{\text{dm}}$ by minimizing the reconstruction loss $\mathcal{L}_{\text{rec}}$ (Eq. 5.2);
  compute fairness metric f_n based on $\mathcal{D}_{\text{rep}}$;
  add f_n **to** $\mathcal{E}$;
end for
apply k-means to cluster clients into benign group $\mathcal{B}$ and adversarial group $\mathcal{J}$ based on $\mathcal{E}$;
for $n \in [N]$ **do**
  if $n \in \mathcal{B}$ **and** $v_n < \theta_{\text{high}}$ **then**
    set $q_n \leftarrow (1-\rho) \cdot q_n + \rho \cdot 1$;
  else if $n \in \mathcal{U}$ **and** $q_n > \theta_{\text{low}}$ **then**
    set $q_n \leftarrow (1-\rho) \cdot q_n + \rho \cdot 0$;
  end if
end for
set $\mathcal{Z} \leftarrow \{n \mid v_n > \kappa, n \in [N]\}$;
compute $h^{r+1} \leftarrow \sum_{i \in \mathcal{Z}} w_i h_i^{r+1}$;
Return h^{r+1}, $\mathcal{Q}$

private training data by matching gradients. Given a model and its parameter gradients, an adversary optimizes dummy inputs and labels to align their gradients with the observed ones, thereby recovering the original features and labels. In contrast, we repurpose GI in a constructive manner. Rather than aiming to recover exact training samples, we use GI to obtain a coarse approximation of each client's data distribution. The resulting representative dataset captures key statistical patterns—particularly fairness-related characteristics—that influence model behavior. Importantly, these reconstructions are inherently lossy and do not preserve fine-grained, individual-level information, ensuring that our use of GI remains fully compliant with the privacy constraints of FL. Then, to better distinguish adversarial clients from noisy but benign ones, FairFedPAPL maintains a confidence score list to quantify the trustworthiness of each client. These scores are updated using a bounded exponential moving

average across communication rounds, and only clients with sufficiently high confidence are selected for global aggregation.

Algorithm 8 outlines the detailed procedure of the proposed FairFedPAPL framework. In the $(r+1)$-th global communication round, the server first estimates the local gradient v_n^{r+1} for each client c_n by computing the difference between the client's uploaded model h_n^{r+1} and the previous global model h^t, scaled by the learning rate η (Algorithm 8, Line 3). Next, a dummy dataset $\mathcal{D}_{\text{dm}}$ is initialized with random values (Algorithm 4, Line 8) and iteratively optimized by minimizing a reconstruction loss $\mathcal{L}_{\text{rec}}$, in order to reconstruct a representative dataset $\mathcal{D}_{\text{rep}}$ that approximates the training data distribution of the local model (Algorithm 8, Line 5). The $\mathcal{L}_{\text{rec}}$ is formally defined as:

$$\mathcal{L}_{\text{rec}} = 1 - \frac{\left\langle \nabla_{h_n^{r+1}} \mathcal{L}(X_{\text{dm}}, Y_{\text{dm}}),\ v_n^{r+1} \right\rangle}{\left\| \nabla_{h_n^{r+1}} \mathcal{L}(X_{\text{dm}}, Y_{\text{dm}}) \right\| \left\| v_n^{r+1} \right\|} + \mu \cdot \text{TV}(X_{\text{dm}}), \tag{5.2}$$

where $\mathcal{L}$ is the standard prediction loss ($e.g.$, cross-entropy), and the first term captures the cosine dissimilarity between the gradient of the loss with respect to the model parameters h_n^{r+1} and the target gradient vector v_n^{r+1}. The second term, $\mu \cdot \text{TV}(X_{\text{dm}})$, applies total variation (TV) regularization to encourage spatial smoothness in the reconstructed inputs. Here, μ is a hyperparameter that controls the strength of the regularization. This optimization is performed for a fixed number of steps using gradient descent. A fairness metric f_n is then computed by evaluating a fairness criterion ($e.g.$, equal opportunity) on the representative dataset $\mathcal{D}_{\text{rep}}$ (Algorithm 8, Lines 6–7). Based on the computed values, clients are clustered into benign ($\mathcal{B}$) and adversarial ($\mathcal{J}$) groups (Algorithm 8, Line 9).

To better distinguish adversarial clients from benign clients with noisy training data, a confidence score v_n is maintained for each client n and updated using a bounded exponential moving average ($i.e.$, incremented for benign behavior and decremented for adversarial behavior). Once a client's confidence exceeds an upper threshold θ_{high} or falls below a lower threshold θ_{low}, the score is frozen to prevent further fluctuations (Alg. 8, Lines 10–16). Finally, only clients whose confidence scores exceed a minimum aggregation threshold κ are selected for global model aggregation (Algorithm 8, Lines 17–18).

5.5 Experiments

5.5.1 Experimental Setups

Datasets. We employ two datasets commonly used in sensitive ML research: (1) the COMPAS dataset [Larson et al. 2016], which is designed to predict the likelihood of reoffending within two years (“low”, “medium”, or “high”), and (2) the Student dataset [Kharoua 2024], which classifies students’ academic performance into grade levels (“A” to “F”). As both datasets were originally developed for fully supervised learning, we preprocess them—using a class-dependent generation process [Tian et al. 2023]—to fit the PAPL setting. Specifically, preprocessing involves generating partially labeled datasets by converting negative labels into false-positive labels with a controlled probability while retaining the true positive label within the candidate set. Here, a false-positive label refers to an incorrect label included in the candidate label set, despite not representing the true class of the instance. This preprocessing is governed by a flipping matrix, ensuring negative labels appear as false-positive labels with either uniform or varying probabilities. This approach realistically simulates practical scenarios, reflecting conditions where labeling errors or ambiguities correlate with underlying class distributions. Consequently, the preprocessed PAPL datasets effectively represent real-world conditions of partial labeling, where each instance is accompanied by a candidate label set containing the true label alongside additional false-positive labels. Moreover, we extend this preprocessing strategy to sensitive attributes, generating datasets with partially provided attribute information. This further simulates practical situations characterized by ambiguous or incomplete attribute labeling. By simultaneously preprocessing both partial labels and partial attributes, we create comprehensive and realistic PAPL datasets suitable for robust evaluation of fairness-aware algorithms.

To generate non-i.i.d. data distributions for FL scenarios, we employ a Dirichlet distribution parameterized by ζ [Minka 2000]. Specifically, this distribution is used to allocate local data samples to each participating client, where the hyperparameter $\zeta \in (0, +\infty)$ controls the degree of heterogeneity across clients. A smaller value of ζ indicates a higher degree of non-i.i.d. data distribution, reflecting significant variance among client data samples. In

our experiments, we set $\zeta = 1.0$ to simulate realistic and challenging FL environments characterized by substantial data heterogeneity. For both datasets, we consider "race" as the sensitive attribute and set "Asian" as the targeted demographic group g_t. Additionally, we define the "high" likelihood of reoffending in COMPAS and the grade level "F" in Student as the unprivileged classes y_u.

Evaluation Metrics. To evaluate the fairness of the FL model, we adopt the *Equal Opportunity Difference* (EOD) metric from existing literature [Sheng et al. 2024]. EOD measures the disparity in true positive rates between two demographic groups ($e.g.$, "Asian" and "non-Asian" in our setting). Formally, it is defined as

$$\text{EOD} = \Pr(\hat{Y} = 1 \mid Y = 1, A = 0) - \Pr(\hat{Y} = 1 \mid Y = 1, A = 1),$$

where A denotes the sensitive attribute, Y the true label, and $\hat{Y}$ the predicted label. The value of EOD lies in the range $[-1, 1]$, where a positive value indicates favorable treatment for the targeted group, while a negative value suggests potential bias against it. A value closer to zero implies a fairer and more balanced model across the two groups. Specifically, it is defined as the difference between the probability that a sample belonging to the "Asian" group ($\hat{P}_{[t]} = 1$) and the unprivileged class ($\mathbf{y}_{[u]} = 1$) is correctly predicted, and the probability that a sample from the "non-Asian" group ($\hat{P}_{[t]} = 0$) but also in the unprivileged class is correctly predicted. The value of EOD lies in the range $[-1, 1]$, where a positive EOD indicates favorable treatment for the targeted group, while a negative EOD implies potential bias against it. A value closer to zero suggests a more fair and balanced model across the two groups.

To better evaluate the effectiveness of different defense mechanisms, we further define the absolute difference between the EOD of the defended model under attack ($\text{EOD}_{\text{defended}}$) and that of a benign model without any attack or defense ($\text{EOD}_{\text{benign}}$) as follows:

$$\Delta_{\text{EOD}} = \left|\text{EOD}_{\text{defended}} - \text{EOD}_{\text{benign}}\right|, \tag{5.3}$$

where a smaller value of Δ_{EOD} indicates a more effective defense in mitigating the impact of the attack.

Benchmark Methods. We adopt various Byzantine-robust aggregation rules as defense benchmarks, including Krum [Blanchard et al. 2017], Trimmed Mean [Yin et al. 2018], and Median [Yin et al. 2018]. These methods are effective in defending against poisoning attacks in FL by mitigating the impact of abnormal or adversarial model updates during the aggregation process. In addition, FedAvg [McMahan et al. 2017] is used as the standard baseline for all attack and defense comparisons.

FairFedPAPL Setups. In FairFedPAPL, the randomly generated dummy dataset is set to a size of 100, and gradient inversion is performed for 20 steps. Confidence scores are initialized to 0.5 for each client, with an upper threshold $\theta_{\text{high}} = 0.9$ and a lower threshold $\theta_{\text{low}} = 0.1$. The exponential moving average uses a smoothing factor of 0.1, and the aggregation threshold is set to $\kappa = 0.5$.

Training Details. For both the COMPAS and Student datasets, we employ a multi-layer perceptron (MLP) as the model architecture. Training is performed using mini-batch stochastic gradient descent (SGD) with a momentum factor of 0.9. The learning rate is set to $\eta = 0.01$ for the COMPAS dataset and $\eta = 0.02$ for the Student dataset. To train the local model, we adopt the standard PLL loss function from [Cour et al. 2011]. In the main results shown in Table 1, the model is trained for a fixed number of global communication rounds across datasets and distribution settings. Specifically, for the COMPAS dataset, training is conducted for 50 global communication rounds under both independent and identically distributed (i.i.d.) and non-independent and identically distributed (non-i.i.d.) settings. For the Student dataset, training is performed for 20 global communication rounds under the i.i.d. setting and 60 global communication rounds under the non-i.i.d. setting. Unless otherwise specified, we consider a FL setting with $N = 10$ clients, among which $M = 3$ are designated as adversarial.

In ablation studies, evaluation is carried out at different communication rounds depending on the factor under investigation: the effect of dummy dataset size is assessed at the 50-th global communication round; the impact of the confidence score and the scalability of FairFedPAPL are evaluated at the 100-th global communication round; and the effect of varying the proportion of adversarial clients is examined at the 80-th global communication round.

TABLE 5.1. FairFedPAPL: Main experimental results for fairness attacks and defenses on the COMPAS and Student datasets under i.i.d. and non-i.i.d. settings.

(a) Results on COMPAS Dataset

Methods		Metrics	COMPAS Dataset					
			i.i.d.			Non-i.i.d.		
			PLFA	PAFA	Mixup	PLFA	PAFA	Mixup
Baseline	FedAvg w/o Attack [McMahan et al. 2017]	Acc.		0.748			0.759	
		EOD		-0.087			-0.078	
	FedAvg w/o Defense [McMahan et al. 2017]	Acc.	0.732	0.741	0.745	0.751	0.750	0.754
		EOD	-0.009	0.050	0.026	0.101	0.068	-0.034
		Δ_{EOD}	0.078	0.137	0.113	0.179	0.146	0.124
Byzantine -robust Defenses	Krum [Blanchard et al. 2017]	Acc.	0.730	0.728	0.735	0.753	0.742	0.757
		EOD	-0.063	-0.069	-0.100	0.029	0.008	-0.001
		Δ_{EOD}	0.024	0.018	0.013	0.107	0.086	0.077
	Trimmed-Mean [Yin et al. 2018]	Acc.	0.743	0.738	0.744	0.755	0.748	0.744
		EOD	-0.060	0.051	0.028	0.047	-0.053	-0.062
		Δ_{EOD}	0.027	0.138	0.115	0.125	0.025	0.016
	Median [Yin et al. 2018]	Acc.	0.747	0.747	0.741	0.757	0.746	0.747
		EOD	-0.067	-0.075	0.001	0.041	-0.056	-0.066
		Δ_{EOD}	0.020	0.012	0.088	0.119	0.022	0.012
Our Defense	FairFedPAPL	Acc.	0.751	0.752	0.752	0.758	0.747	0.757
		EOD	-0.084	-0.084	-0.081	-0.083	-0.091	-0.080
		Δ_{EOD}	0.003	0.003	0.006	0.005	0.013	0.002

(b) Results on Student Dataset

Methods		Metrics	Student Dataset					
			i.i.d.			non-i.i.d.		
			PLFA	PAFA	Mixup	PLFA	PAFA	Mixup
Baseline	FedAvg w/o Attack [McMahan et al. 2017]	Acc.		0.766			0.587	
		EOD		-0.023			0.014	
	FedAvg w/o Defense [McMahan et al. 2017]	Acc.	0.706	0.745	0.754	0.534	0.580	0.570
		EOD	-0.152	-0.069	0.041	-0.084	-0.089	-0.082
		Δ_{EOD}	0.129	0.046	0.011	0.098	0.103	0.096
Byzantine -robust Defenses	Krum [Blanchard et al. 2017]	Acc.	0.758	0.754	0.754	0.501	0.491	0.455
		EOD	-0.010	0.082	-0.034	-0.017	-0.048	-0.192
		Δ_{EOD}	0.013	0.105	0.013	0.031	0.062	0.206
	Trimmed-Mean [Yin et al. 2018]	Acc.	0.729	0.731	0.756	0.522	0.566	0.572
		EOD	-0.084	-0.069	-0.034	-0.019	-0.071	-0.040
		Δ_{EOD}	0.061	0.046	0.011	0.033	0.085	0.054
	Median [Yin et al. 2018]	Acc.	0.701	0.722	0.735	0.532	0.551	0.566
		EOD	-0.079	-0.059	-0.062	0.008	-0.047	-0.033
		Δ_{EOD}	0.056	0.036	0.039	0.006	0.061	0.047
Our Defense	FairFedPAPL	Acc.	0.768	0.766	0.768	0.580	0.585	0.587
		EOD	-0.027	-0.002	-0.027	0.013	0.017	0.003
		Δ_{EOD}	0.004	0.021	0.004	0.001	0.003	0.011

5.5.2 Main Empirical Results

Table 5.1 summarizes the quantitative results of our main empirical experiments on fairness attacks and defenses conducted on the COMPAS and Student datasets under i.i.d. and non-i.i.d. data distributions. As shown in the *Baseline* section, all three proposed FedPAPL fairness attacks (FedAvg w/o Defense) successfully compromise the fairness of the aggregated FL model, as evidenced by large Δ_{EOD} values when no defense is applied. Among them, PAFA is generally the most harmful on COMPAS (e.g., $\Delta_{\text{EOD}} = 0.179$ under non-i.i.d), while both PLFA and PAFA have strong effects on Student non-i.i.d ($\Delta_{\text{EOD}} \approx 0.10$). By contrast, Mixup leads to only moderate fairness degradation but remains consistently difficult to detect. Notably, Mixup often preserves accuracy close to the benign baseline (e.g., COMPAS non-i.i.d Mixup Acc.=0.754 vs. 0.759 without attack), yet it still yields significant fairness violations ($\Delta_{\text{EOD}} = 0.124$). This highlights a subtle but important insight: certain attacks do not visibly harm accuracy, but they substantially undermine fairness, making them more deceptive.

Regarding defenses, existing Byzantine-robust aggregation strategies (Krum, Trimmed-Mean, Median) achieve partial success in mitigating the effects of PLFA and PAFA. For example, under COMPAS IID, Median reduces Δ_{EOD} from 0.078 to 0.020 for PLFA and from 0.137 to 0.012 for PAFA. However, these defenses show very limited effectiveness against the Mixup attack; under the same setting, Median only reduces Δ_{EOD} from 0.113 to 0.088, since Mixup introduces minimal perturbations to the local data distribution and thus evades detection. In contrast, the proposed **FairFedPAPL** framework demonstrates exceptional robustness against all three fairness attacks, consistently achieving Δ_{EOD} values close to zero while maintaining accuracy comparable to the benign baseline. This strong performance is primarily attributed to the unique design of FairFedPAPL, which leverages GI to reconstruct accurate approximations of client data distributions, thereby enabling the detection of even subtle perturbations introduced by adversarial clients.

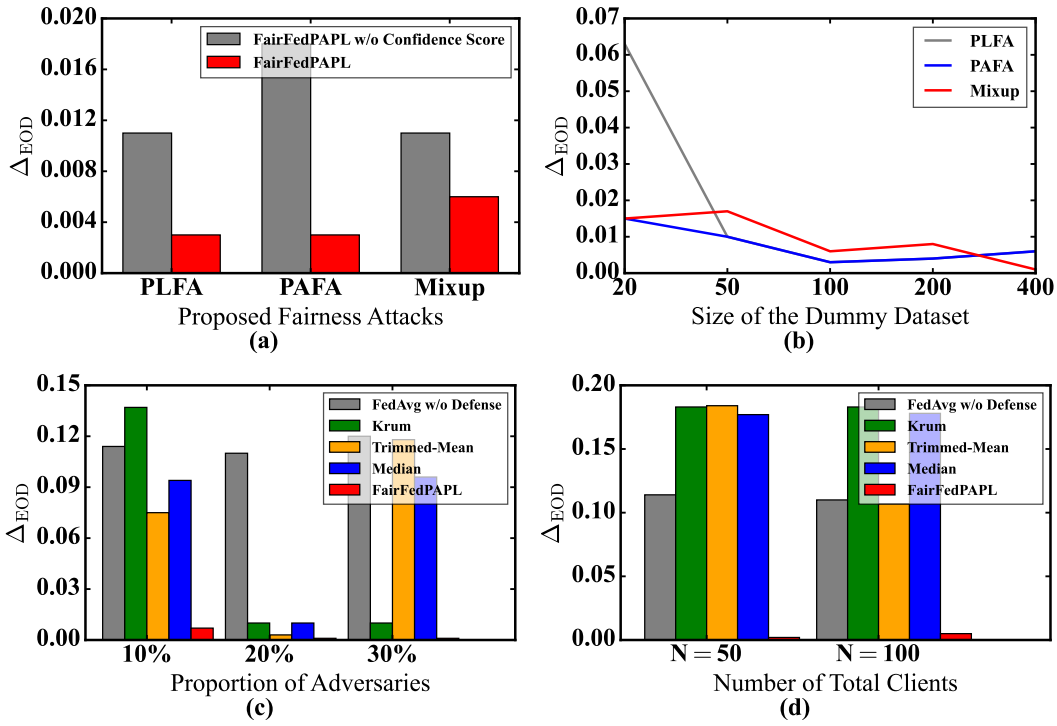

FIGURE 5.3. Ablation studies on (a) confidence score of FairFedPAPL. Hyperparameter studies on (b) size of the dummy dataset, (c) proportion of adversaries, and (d) scalability of FairFedPAPL.

5.5.3 Ablation Studies

Impact of Confidence Score. Figure 5.3 (a) illustrates the impact of incorporating the confidence score into the proposed FairFedPAPL framework. Specifically, we compare the performance between the complete FairFedPAPL framework and a variant without the confidence score, where clients in the adversarial group ($\mathcal{J}$) are directly filtered out during aggregation ($i.e.$, stopping at Line 8 in Algorithm 8). The results show that while FairFedPAPL without the confidence score still demonstrates a certain degree of effectiveness against the fairness attack, incorporating the confidence score leads to better performance, as indicated by a smaller Δ_{EOD}. This confirms that the confidence score helps FairFedPAPL more accurately distinguish between adversarial clients and benign but noisy clients.

5.5.4 Hyperparameter Studies

Impact of Dummy Dataset Size. Figure 5.3 (b) illustrates the impact of the size of the dummy dataset $\mathcal{D}_{\text{dm}}$ on the performance of the proposed FairFedPAPL framework. It can be observed that the effectiveness of FairFedPAPL improves as the size of $\mathcal{D}_{\text{dm}}$ increases. However, this performance gain diminishes significantly once the size exceeds 100, indicating that the reconstructed representative data can already sufficiently capture each client's distribution beyond a certain threshold. On the other hand, increasing the size of $\mathcal{D}_{\text{dm}}$ also leads to reduced efficiency. Therefore, we select a size of 100 to achieve a good balance between performance and efficiency for FairFedPAPL.

Proportion of Adversaries. While our main experiments follow the assumption that at most 30% of the clients are adversaries, we are also interested in exploring how a smaller proportion of adversaries (a more realistic setting in practice) influences the fairness of the FL model and the effectiveness of different defense mechanisms. Figure 5.3 (c) presents the Δ_{EOD} results when 10%, 20%, and 30% of the clients perform the Mixup attack. It can be observed that the proposed Mixup attack remains effective in introducing bias into the FL model even with a smaller proportion of adversarial clients. Moreover, the proposed FairFedPAPL defense consistently achieves strong performance in mitigating the attack across all proportions.

Scalability of FairFedPAPL. Figure 5.3 (d) illustrates the effectiveness of the proposed Mixup attack and the FairFedPAPL framework in larger-scale FL settings ($i.e.$, with a total of $N = 50$ and $N = 100$ clients). It can be observed that the Mixup attack consistently compromises the fairness of the FL model, while the FairFedPAPL defense consistently mitigates the attack in larger-scale FL settings.

5.5.5 Other Fairness Metric

While EOD serves as our primary fairness metric due to its established role in the fairness literature, we also report Statistical Parity Difference (SPD) to offer a complementary view. Unlike EOD, which focuses on disparities in true positive rates, SPD captures differences in overall positive outcome rates across sensitive groups. To better illustrate the impact of

TABLE 5.2. Evaluating fairness defenses on COMPAS using Δ_{SPD}.

Methods		Metrics	COMPAS Dataset		
			PLFA	PAFA	Mixup
Baseline	FedAvg w/o Attack [McMahan et al. 2017]	Acc.	0.745		
		SPD	-0.091		
	FedAvg w/o Defense [McMahan et al. 2017]	Acc.	0.721	0.738	0.737
		SPD	-0.005	0.044	0.015
		Δ_{SPD}	0.086	0.135	0.106
Byzantine -robust Defenses	Krum [Blanchard et al. 2017]	Acc.	0.729	0.730	0.729
		SPD	-0.117	-0.117	-0.127
		Δ_{SPD}	0.026	0.026	0.036
	Trimmed-Mean [Yin et al. 2018]	Acc.	0.721	0.736	0.714
		SPD	-0.067	-0.069	0.027
		Δ_{SPD}	0.024	0.022	0.118
	Median [Yin et al. 2018]	Acc.	0.739	0.745	0.741
		SPD	-0.079	-0.079	-0.007
		Δ_{SPD}	0.012	0.012	0.084
Our Defense	FairFedPAPL	Acc.	0.751	0.752	0.741
		SPD	-0.094	-0.095	-0.085
		Δ_{SPD}	**0.003**	**0.004**	**0.006**

various defenses, we measure the absolute change in SPD (Δ_{SPD}) between the no-attack baseline and the defense-applied setting. At the 40th global communication round on the COMPAS dataset, our proposed FairFedPAPL achieves significantly lower Δ_{SPD} of 0.003, 0.004, and 0.006 (see Table 5.2), demonstrating consistently stronger fairness robustness across all three attack types.

5.5.6 Complexity Analysis

Our GI-based defense in FairFedPAPL is executed on the server (data center) side, where computational resources are typically ample in FL settings. As a result, the additional overhead introduced by the defense is generally acceptable in practice. Among various factors, the number of participating clients per round has the most significant impact on runtime, as reconstruction is performed on a per-client basis. However, this cost can be substantially mitigated through parallelism using multi-GPU or multi-threaded implementations. To

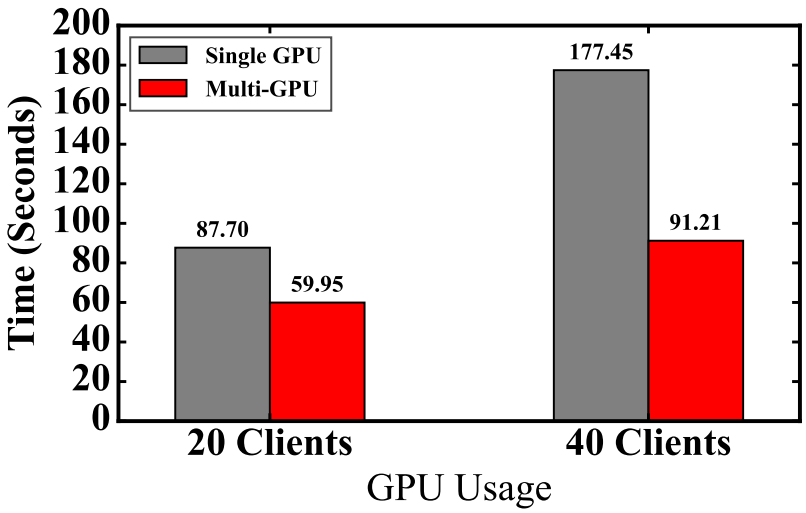

FIGURE 5.4. Comparison of time consumption between single GPU and multi-GPU settings under different numbers of clients.

analyze computational complexity and the benefits of parallelism, we conducted additional experiments under both single and multi-GPU configurations in Figure 5.4. Specifically, we measured performance using 20 and 40 clients across 100 communication rounds. Here, the reported total runtime refers to the cumulative aggregation time over 100 rounds, including the full cost of applying the GI-based defense across all participating clients in each round. With 20 clients, the total runtime decreased from 87.7 seconds on a single GPU to 59.95 seconds with three GPUs in a multi-threaded setup—a 31.64% reduction. Similarly, with 40 clients, the runtime dropped from 177.45 to 91.21 seconds, achieving a 48.6% reduction. These results demonstrate that the overhead introduced by GI-based defense can be effectively controlled with parallelization, even at scale.

5.5.7 Discussion

We further elaborate on the key novelties of FairFedPAPL ($i.e.$, the first use of GI as a defense mechanism, the introduction of a confidence score to better distinguish adversarial clients from benign but noisy ones, and its design as an auxiliary component that can be seamlessly integrated into existing FedPLL frameworks), along with a discussion on its ethical implications, potential limitations, and promising future research directions.

5.5.8 Novelty of FairFedPAPL

The first novelty of FairFedPAPL lies in its pioneering use of GI as a defense technique, whereas existing works primarily utilize GI as an adversarial attack against ML models. Leveraging GI as a defense in the FairFedPAPL framework offers two key benefits: (1) The representative data reconstructed via GI more accurately captures the underlying data distribution of each client, enabling more precise fairness evaluation; (2) Since GI is initiated from randomly generated inputs, the reconstructed representative data only coarsely approximates the client's local data distribution without revealing any explicit information ($e.g.$, actual training samples), thereby preserving the privacy guarantees inherent to FL.

Another novelty of FairFedPAPL lies in its introduction of a confidence score to quantify the trustworthiness of each client. Under the realistic weakly supervised setting (PAPL) and the inherently non-i.i.d. data distribution in FL, this confidence score enables more reliable differentiation between adversarial clients and benign but noisy ones. Rather than making decisions based on a single communication round, the score accumulates behavioral evidence over multiple rounds (updated by a moving average strategy), leading to more robust client assessment. As a result, this mechanism allows FairFedPAPL to achieve extremely high accuracy in identifying adversarial clients.

Moreover, the proposed FairFedPAPL can be viewed as an auxiliary defense mechanism that can be seamlessly integrated into existing FedPLL frameworks. It serves as a complementary security component that enhances the robustness of these frameworks Yan and Guo 2024, enabling secure and effective federated partial label learning.

5.5.9 Ethical Statements and Implications

We acknowledge the clear ethical implications of our work, as the proposed fairness adversarial attacks could potentially be exploited by malicious actors. However, the primary motivation behind this research is to raise awareness of security concerns in existing FedPLL frameworks by exposing potential vulnerabilities. In response, we also propose FairFedPAPL, a novel defense mechanism that demonstrates strong effectiveness against the introduced fairness

attacks. We hope that future research on weakly supervised learning in federated settings will take system security into consideration and adopt our proposed attacks as a benchmark for evaluating the robustness of new methods.

5.5.10 FairFedPAPL Limitations

Given the unique design of the FairFedPAPL defense framework, there are no major limitations. A potential concern lies in the additional computational overhead introduced by the use of GI to reconstruct representative data for each local client. However, since the server in a typical FL system is generally equipped with sufficient computational resources, we believe this overhead is manageable and does not pose a significant issue in practice.

5.6 Future Works

A promising direction for future work is to investigate more advanced adversarial attacks within the FedPAPL framework ($e.g.$, backdoor attacks and membership inference attacks), along with the development of corresponding defense strategies. Additionally, while the proposed FairFedPAPL serves as an auxiliary defense mechanism that can be seamlessly integrated into existing FedPLL frameworks, a more fundamental direction is to design a secure FedPLL/FedPAPL framework that simultaneously addresses the challenges of FedPLL/FedPAPL and ensures robustness against security threats.

5.7 Conclusion

In this chapter, we conduct a pioneering exploration of security vulnerabilities in weakly supervised FL. We present three novel fairness attacks that substantially threaten the fairness of FL models under realistic data collection scenarios (PAPL). To counteract these vulnerabilities, we propose FairFedPAPL, a novel defense mechanism utilizing gradient inversion, which effectively mitigates these fairness threats. Our study underscores the critical importance of

embedding security mechanisms within weakly supervised FL frameworks to ensure model fairness and reliability in practical deployments.

CHAPTER 6

Conclusion

6.1 Summary

Federated Learning (FL) presents an effective paradigm for collaboratively training a global model across multiple clients while keeping their raw data local. Instead of centralizing data, clients share only model updates—such as gradients or parameters—with a central server, thereby enhancing privacy and reducing the risk of sensitive data exposure. This decentralized approach makes FL particularly suitable for domains where data privacy and compliance are critical, such as healthcare, finance, and mobile edge applications. However, despite its advantages, FL suffers from several inherent challenges. These include data heterogeneity, as clients often possess non-i.i.d. data distributions; limited control over data collection, which may lead to noisy, incomplete, or weakly labeled datasets; and the oversight of ethical concerns, particularly in terms of fairness and robustness. While many existing studies focus on improving model performance, they often neglect the broader implications of algorithmic fairness in decentralized and weakly supervised environments. New FL algorithms are urgently needed to overcome the limitations of data heterogeneity, weak supervision, and fairness.

In this thesis, we propose a series of personalized and fairness-aware FL frameworks designed to address key challenges, including data heterogeneity, weak supervision, and ethical concerns such as fairness and robustness. Through progressively realistic problem settings and principled algorithmic designs, we aim to bridge the gap between model performance and responsible learning in decentralized environments.

Firstly, we propose pFedMo, a personalized FL algorithm that incorporates contrastive momentum. It significantly improves performance under non-i.i.d. data distributions while retaining the efficiency of momentum-based optimization. However, since pFedMo is trained on fully supervised datasets, it does not reflect the challenges of real-world data collection, where labels are often noisy or incomplete. To address this, our second contribution, pFedPLL, extends personalized FL to the weakly supervised setting of Partial Label Learning (PLL). While PLL helps reduce annotation costs by requiring only candidate label sets, existing works—including pFedPLL—primarily focus on improving accuracy, often overlooking ethical aspects such as fairness and robustness. In PLL, the presence of false-positive labels can mislead the model, introducing unfair bias that disproportionately affects certain groups. To further investigate this issue, we develop our third contribution: a fairness-aware framework under weak supervision. Specifically, we extend FedPLL to a more realistic setting called Federated Partial Attribute Partial Label Learning (FedPAPL), where both labels and attributes are ambiguously annotated. This formulation more accurately reflects real-world data collection scenarios, where both labels and attributes are often ambiguously or imprecisely annotated. For example, in survey-based data collection, participants may provide vague or incomplete answers, resulting in uncertainty in both attribute and label annotations. Based on this setting, we introduce FairFedPAPL, a defense framework that leverages gradient inversion to reconstruct representative client data. This allows for accurate detection of adversarial clients and effective mitigation of fairness attacks in weakly supervised FL environments. For all three proposed algorithms, we present comprehensive theoretical analyses and conduct extensive experiments across multiple benchmark datasets. The results consistently validate the effectiveness and demonstrate the superior performance of our methods compared to existing baselines.

6.2 Future outlook

This thesis has made several contributions toward addressing key limitations in FL. The proposed algorithms are motivated by challenges such as data heterogeneity, weak supervision, and the lack of fairness considerations in existing FL frameworks. By progressively exploring

more realistic problem settings and incorporating personalization and ethical design, we provide a foundation for more robust and responsible FL.

Building on these contributions, several potential directions for future research are outlined as follows:

to do: those point cannot be used, need to resurvey

- **Federated learning under device heterogeneity:** In this thesis, we only consider the scenario where the data is heterogeneous and do not take into account device heterogeneity. As a result, factors such as varying computational capabilities, communication bandwidth, and energy constraints across devices are not addressed. Investigating these aspects is crucial for developing more practical and robust federated learning systems, and we leave this direction for future work. In particular, new algorithms can be developed to intelligently select participating clients [Nishio and Yonetani 2019; Lai et al. 2021], thereby reducing unnecessary communication and computational costs.
- **Federated learning under model heterogeneity:** This thesis assumes that all clients share the same model architecture. However, in real-world FL scenarios, clients may use different model architectures due to diverse hardware capabilities, application requirements, or preferences. Such model heterogeneity introduces new challenges in aggregation and knowledge transfer. Addressing this setting requires designing algorithms that can align and integrate knowledge across heterogeneous models [Tan et al. 2022; Ni et al. 2022], which we leave as an important direction for future research.
- **Privacy protection in federated learning:** While FL inherently offers a level of privacy by keeping raw data local, recent studies have shown that model updates can still leak sensitive information through gradient leakage or inference attacks. In this thesis, we do not explicitly incorporate privacy-preserving techniques such as differential privacy [Sun et al. 2022], secure multi-party computation [Bonawitz et al. 2017], or homomorphic encryption [Zhang et al. 2020b]. Integrating these methods

is essential to ensure stronger privacy guarantees, and exploring their impact on learning performance remains an important avenue for future work.

- **Aggregation optimization in federated learning:** Standard aggregation strategies such as FedAvg may become suboptimal in the presence of data heterogeneity, straggling clients, or adversarial behaviors. Optimizing the aggregation process through weighted-level aggregation [Yurochkin et al. 2019], feature-level aggregation [Yu et al. 2021], or adaptive schemes [Chen et al. 2020b] can significantly improve model convergence, fairness, and resilience. Further research in this direction is essential for enhancing the scalability and reliability of federated learning systems.
- **Fairness in federated learning:** This thesis focuses only on enhancing fairness in terms of model utility across clients. However, another important aspect of fairness lies in incentive mechanisms and equitable participation, ensuring that all clients are treated fairly [Törnblom and Jonsson 1985; Lyu et al. 2020] with respect to their contributions, rewards, and resource usage. Exploring these broader notions of fairness remains an important direction for future research.

Bibliography

Abay, Annie et al. (2020). 'Mitigating bias in federated learning'. In: *arXiv preprint arXiv:2012.02447*.

Anguita, Davide et al. (2013). 'A public domain dataset for human activity recognition using smartphones'. In: *Proceedings of the 21th international European symposium on artificial neural networks, computational intelligence and machine learning*, pp. 437–442.

Bagdasaryan, Eugene et al. (2020). 'How to backdoor federated learning'. In: *International conference on artificial intelligence and statistics*. PMLR, pp. 2938–2948.

Bai, Sikai et al. (2024). 'Combating data imbalances in federated semi-supervised learning with dual regulators'. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 38. 10, pp. 10989–10997.

Bastola, Ashish et al. (2024). 'FedMIL: Federated-Multiple Instance Learning for Video Analysis with Optimized DPP Scheduling'. In: *arXiv preprint arXiv:2403.17331*.

Blanchard, Peva et al. (2017). 'Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent'. In: *Advances in Neural Information Processing Systems*.

Blitzer, John et al. (2007). 'Learning bounds for domain adaptation'. In: *Advances in neural information processing systems*.

Bonawitz, Keith et al. (2017). 'Practical secure aggregation for privacy-preserving machine learning'. In: *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191.

Briggs, Forrest, Xiaoli Z Fern and Raviv Raich (2012). 'Rank-loss support instance machines for MIML instance annotation'. In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 534–542.

Bubeck, Sébastien (2014). 'Convex optimization: Algorithms and complexity'. In: *arXiv preprint arXiv:1405.4980*.

Calmon, Flavio et al. (2017). 'Optimized pre-processing for discrimination prevention'. In: *Advances in neural information processing systems* 30.

Celis, L Elisa, Anay Mehrotra and Nisheeth Vishnoi (2021). 'Fair classification with adversarial perturbations'. In: *Advances in Neural Information Processing Systems*.

Chen, Ting et al. (2020a). 'A simple framework for contrastive learning of visual representations'. In: *International conference on machine learning*. PMLR, pp. 1597–1607.

Chen, Xiangyi et al. (2020b). 'Distributed training with heterogeneous data: Bridging median- and mean-based algorithms'. In: *Advances in Neural Information Processing Systems* 33, pp. 21616–21626.

Chen, Yiqiang et al. (2020c). 'Fedhealth: A federated transfer learning framework for wearable healthcare'. In: *IEEE Intelligent Systems* 35.4, pp. 83–93.

Clanuwat, Tarin et al. (2018). 'Deep learning for classical japanese literature'. In: *arXiv preprint arXiv:1812.01718*.

Corinzia, Luca, Ami Beuret and Joachim M Buhmann (2019). 'Variational federated multi-task learning'. In: *arXiv preprint arXiv:1906.06268*.

Cour, Timothee, Ben Sapp and Ben Taskar (2011). 'Learning from Partial Labels'. In: *Journal of Machine Learning Research* 12.42, pp. 1501–1536.

Dai, Mingjun et al. (2021). 'Vertical federated DNN training'. In: *Physical Communication* 49, p. 101465.

Deng, Jia et al. (2009). 'Imagenet: A large-scale hierarchical image database'. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, pp. 248–255.

Dinh, Canh T et al. (2020). 'Federated learning with proximal stochastic variance reduced gradient algorithms'. In: *Proceedings of the 49th International Conference on Parallel Processing*, pp. 1–11.

Du, Wei et al. (2021). 'Fairness-aware agnostic federated learning'. In: *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*. SIAM.

Dunnings, A. and T.P. Breckon (Sept. 2018). 'Experimentally defined Convolutional Nerual Network Architecture Variants for Non-temporal Real-time Fire Detection'. In: *Proc. International Conference on Image Processing*. IEEE, pp. 1558–1562. DOI: 10.1109/ICIP.2018.8451657.

Dunnings, A. and T.P. Breckon (Aug. 2020). *Fire Detection Datasets*. URL: https://github.com/tobybreckon/fire-detection-cnn/blob/master/README.md.

Ezzeldin, Yahya H et al. (2023). 'Fairfed: Enabling group fairness in federated learning'. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 37. 6, pp. 7494–7502.

Feldman, Michael et al. (2015). 'Certifying and removing disparate impact'. In: *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*.

Feng, Lei et al. (2020). 'Provably consistent partial-label learning'. In: *Advances in neural information processing systems* 33, pp. 10948–10960.

Fu, Sen (2021). 'Convergence Analysis on the Federated Learning Algorithm with NAG Approach'. Bachelor of Computer Science and Technology (Advanced) (Honours) Thesis. The University of Sydney, Sydney, Australia.

Fu, Sen et al. (2024). 'When Federated Learning Meets Partial Label Data'. In: *under review at IEEE Transactions on Computers*. Under review.

Fu, Sen et al. (2025a). 'Personalized Federated Learning with Contrastive Momentum'. In: *IEEE Transactions on Big Data* 11.5, pp. 2184–2194. DOI: 10.1109/TBDATA.2024.3403387.

Fu, Sen et al. (2025b). 'Towards Robust and Fair Partial Label Federated Learning Service'. In: *Under review at IEEE Transactions on Services Computing*. Under review.

Gálvez, Borja Rodríguez et al. (2021). 'Enforcing fairness in private federated learning via the modified method of differential multipliers'. In: *NeurIPS 2021 Workshop Privacy in Machine Learning*.

Geiping, Jonas et al. (2020). 'Inverting gradients-how easy is it to break privacy in federated learning?' In: *Advances in neural information processing systems* 33, pp. 16937–16947.

Gong, Xiuwen, Dong Yuan and Wei Bao (2021). 'Discriminative metric learning for partial label learning'. In: *IEEE Transactions on Neural Networks and Learning Systems* 34.8, pp. 4428–4439.

Grgić-Hlača, Nina et al. (2018). 'Beyond distributive fairness in algorithmic decision making: Feature selection for procedurally fair learning'. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1.

Gross, Sam et al. (May 2021). *Pytorch-VGG*. URL: https://github.com/pytorch/vision/blob/master/torchvision/models/vgg.py.

Guillaumin, Matthieu, Jakob Verbeek and Cordelia Schmid (2010). 'Multiple instance metric learning from automatically labeled bags of faces'. In: *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I 11*. Springer, pp. 634–647.

Guo, Yichen et al. (2024). 'Dpg-fairfl: A dual-phase gan-based defense framework against image-based fairness data poisoning attacks in federated learning'. In: *International Conference on Algorithms and Architectures for Parallel Processing*. Springer.

He, Kaiming et al. (2016). 'Deep residual learning for image recognition'. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

He, Kaiming et al. (2020). 'Momentum contrast for unsupervised visual representation learning'. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738.

Hinton, Geoffrey, Oriol Vinyals and Jeff Dean (2015). 'Distilling the knowledge in a neural network'. In: *arXiv preprint arXiv:1503.02531*.

Hu, Shengyuan, Zhiwei Steven Wu and Virginia Smith (2024). 'Fair federated learning via bounded group loss'. In: *2024 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. IEEE.

Hüllermeier, Eyke and Jürgen Beringer (2005). 'Learning from Ambiguously Labeled Examples'. In: *Advances in Intelligent Data Analysis VI*, pp. 168–179. ISBN: 978-3-540-31926-9.

Huo, Zhouyuan et al. (2020). 'Faster on-device training using new federated momentum algorithm'. In: *arXiv preprint arXiv:2002.02090*.

Jeong, Wonyong et al. (2021). 'Federated Semi-Supervised Learning with Inter-Client Consistency & Disjoint Learning'. In: *International Conference on Learning Representations*.

Jin, Rong and Zoubin Ghahramani (2002). 'Learning with Multiple Labels'. In: *Advances in Neural Information Processing Systems*. Vol. 15.

Jin, Yilun et al. (2023). 'Federated learning without full labels: A survey'. In: *arXiv preprint arXiv:2303.14453*.

Kairouz, Peter et al. (2021). 'Advances and open problems in federated learning'. In: *Foundations and Trends® in Machine Learning* 14.1–2, pp. 1–210.

Kamishima, Toshihiro et al. (2012). 'Fairness-Aware Classifier with Prejudice Remover Regularizer'. In: *Machine Learning and Knowledge Discovery in Databases*. Springer.

Karimireddy, Sai Praneeth et al. (2020a). 'Mime: Mimicking centralized stochastic algorithms in federated learning'. In: *arXiv preprint arXiv:2008.03606*.

Karimireddy, Sai Praneeth et al. (2020b). 'Scaffold: Stochastic controlled averaging for federated learning'. In: *International Conference on Machine Learning*. PMLR, pp. 5132–5143.

Kelly, Rhea (2015). *Internet of Things data to top 1.6 zettabytes by 2020, Apr 2015*.

Kharoua, Rabie El (2024). *Students Performance Dataset*. DOI: 10.34740/KAGGLE/DS/5195702. URL: https://www.kaggle.com/ds/5195702.

Kim, Michael P, Amirata Ghorbani and James Zou (2019). 'Multiaccuracy: Black-box post-processing for fairness in classification'. In: *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*.

Krizhevsky, Alex, Geoffrey Hinton et al. (2009). 'Learning multiple layers of features from tiny images'. In.

Krizhevsky, Alex, Ilya Sutskever and Geoffrey E Hinton (2017). 'Imagenet classification with deep convolutional neural networks'. In: *Communications of the ACM* 60.6, pp. 84–90.

Lai, Fan et al. (2021). 'Oort: Efficient federated learning via guided participant selection'. In: *15th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 21)*, pp. 19–35.

Larson, Jeff et al. (2016). 'How We Analyzed the COMPAS Recidivism Algorithm'. In.

Le, Ya and Xuan Yang (2015). 'Tiny imagenet visual recognition challenge'. In: *CS 231N*.

LeCun, Yann et al. (1998). 'Gradient-based learning applied to document recognition'. In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Li, Daliang and Junpu Wang (2019). 'Fedmd: Heterogenous federated learning via model distillation'. In: *arXiv preprint arXiv:1910.03581*.

Li, Tian et al. (2020). 'Federated optimization in heterogeneous networks'. In: *Proceedings of Machine learning and systems* 2, pp. 429–450.

Liang, Paul Pu et al. (2020). 'Think locally, act globally: Federated learning with local and global representations'. In: *arXiv preprint arXiv:2001.01523*.

Lim, Wei Yang Bryan et al. (2020). 'Federated learning in mobile edge networks: A comprehensive survey'. In: *IEEE Communications Surveys & Tutorials* 22.3, pp. 2031–2063.

Lin, Tao et al. (2020). 'Ensemble distillation for robust model fusion in federated learning'. In: vol. 33, pp. 2351–2363.

Liu, Liping and Thomas Dietterich (2012). 'A Conditional Multinomial Mixture Model for Superset Label Learning'. In: *Advances in Neural Information Processing Systems*. Vol. 25.

Liu, Yijie et al. (2025). 'Mind the Gap: Confidence Discrepancy Can Guide Federated Semi-Supervised Learning Across Pseudo-Mismatch'. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 10173–10182.

Lohia, Pranay K et al. (2019). 'Bias mitigation post-processing for individual and group fairness'. In: *Icassp 2019-2019 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE.

Lu, Wang et al. (2022). 'Personalized federated learning with adaptive batchnorm for healthcare'. In: *IEEE Transactions on Big Data*.

Lv, Jiaqi et al. (2020). 'Progressive identification of true labels for partial-label learning'. In: *international conference on machine learning*. PMLR, pp. 6500–6510.

Lv, Jiaqi et al. (2023). 'On the robustness of average losses for partial-label learning'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.5, pp. 2569–2583.

Lyu, Lingjuan et al. (2020). 'Collaborative fairness in federated learning'. In: *Federated Learning: Privacy and Incentive*, pp. 189–204.

MacKay, David J.C. (n.d.). *Information Theory, Inference and Learning Algorithms*. Cambridge University Press. URL: http://www.inference.org.uk/mackay/itila/.

McMahan, Brendan et al. (2017). 'Communication-efficient learning of deep networks from decentralized data'. In: *Artificial Intelligence and Statistics*. PMLR, pp. 1273–1282.

Mehrabi, Ninareh et al. (2021a). 'A survey on bias and fairness in machine learning'. In: *ACM computing surveys (CSUR)* 54.6, pp. 1–35.

Mehrabi, Ninareh et al. (2021b). 'Exacerbating algorithmic bias through fairness attacks'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*.

Minka, Thomas (2000). *Estimating a Dirichlet distribution*.

Mitliagkas, Ioannis and Jose Gallego (2021). 'IFT 6085: Theoretical principles for deep learning'. In: *University of Montreal*. University of Montreal. URL: http://mitliagkas.github.io/ift6085-dl-theory-class/.

Moon, TaeJun and Théo Ryffel (June 2020). *Pytorch-Tiny-ImageNet*. URL: https://github.com/tjmoon0104/pytorch-tiny-imagenet.

Nesterov, Yurii (1983). 'A method for unconstrained convex minimization problem with the rate of convergence o(1/k2)'. eng. In: *Doklady ANSSSR (translated as Soviet.Math.Docl.)* 269, pp. 543–547.

Netzer, Yuval et al. (2011). 'Reading digits in natural images with unsupervised feature learning'. In: *NIPS workshop on deep learning and unsupervised feature learning*. Vol. 2011. 5. Granada, p. 7.

Nguyen, Nam and Rich Caruana (2008). 'Classification with partial labels'. In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Ni, Xuanming, Xinyuan Shen and Huimin Zhao (2022). 'Federated optimization via knowledge codistillation'. In: *Expert Systems with Applications* 191, p. 116310.

Nishio, Takayuki and Ryo Yonetani (2019). 'Client selection for federated learning with heterogeneous resources in mobile edge'. In: *ICC 2019-2019 IEEE international conference on communications (ICC)*. IEEE, pp. 1–7.

Ozfatura, Emre, Kerem Ozfatura and Deniz Gündüz (2021). 'FedADC: Accelerated Federated Learning with Drift Control'. In: *2021 IEEE International Symposium on Information Theory (ISIT)*. Melbourne, Australia: IEEE Press, pp. 467–472. DOI: 10.1109/ISIT45174.2021.9517850.

Pan, Sinno Jialin and Qiang Yang (2010). 'A survey on transfer learning'. In: *IEEE Transactions on knowledge and data engineering* 22.10, pp. 1345–1359.

Papadaki, Afroditi et al. (2022). 'Minimax demographic group fairness in federated learning'. In: *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*.

Peng, Xingchao et al. (2020). 'Federated Adversarial Domain Adaptation'. In: *International Conference on Learning Representations*.

Polyak, Boris T (1964). 'Some methods of speeding up the convergence of iteration methods'. In: *USSR Computational Mathematics and Mathematical Physics* 4.5, pp. 1–17.

Press, William H. et al. (2007). *Numerical Recipes: The Art of Scientific Computing (3rd Edition)*. Cambridge University Press. Chap. Kullback–Leibler Distance. ISBN: 978-0-521-88068-8.

Putzel, Preston and Scott Lee (2022). 'Blackbox post-processing for multiclass fairness'. In: *arXiv preprint arXiv:2201.04461*.

Roh, Yuji et al. (2021). 'FairBatch: Batch Selection for Model Fairness'. In: *9th International Conference on Learning Representations*.

Ruder, Sebastian (2016). 'An overview of gradient descent optimization algorithms'. In: *arXiv preprint arXiv:1609.04747*.

Salazar, Teresa et al. (2023). 'Fair-fate: Fair federated learning with momentum'. In: *International Conference on Computational Science*. Springer.

Sheng, Xinyi, Zhengjie Yang and Wei Bao (2024). 'FairGuard: A Fairness Attack and Defense Framework in Federated Learning'. In: *IEEE Transactions on Dependable and Secure Computing*.

Smith, Virginia et al. (2017). 'Federated multi-task learning'. In: *Advances in neural information processing systems* 30.

Solanas, Agusti et al. (2014). 'Smart health: A context-aware health paradigm within smart cities'. In: *IEEE Communications Magazine* 52.8, pp. 74–81.

Solans, David, Battista Biggio and Carlos Castillo (2020). 'Poisoning attacks on algorithmic fairness'. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer.

Song, Hwanjun et al. (2022). 'Learning from noisy labels with deep neural networks: A survey'. In: *IEEE transactions on neural networks and learning systems* 34.11, pp. 8135–8153.

Song, Zixing et al. (2024). 'A systematic survey on federated semi-supervised learning'. In: *IJCAI*. Vol. 16, p. 18.

Stojkoska, Biljana L Risteska and Kire V Trivodaliev (2017). 'A review of Internet of Things for smart home: Challenges and solutions'. In: *Journal of cleaner production* 140, pp. 1454–1464.

Sun, Zhe et al. (2022). 'Fed-dfe: A decentralized function encryption-based privacy-preserving scheme for federated learning'. In: *Computers, Materials and Continua*.

Tan, Alysa Ziying et al. (2022). 'Towards personalized federated learning'. In: *IEEE Transactions on Neural Networks and Learning Systems*.

Tan, Jiahao et al. (2023). 'pFedSim: Similarity-Aware Model Aggregation Towards Personalized Federated Learning'. In: *arXiv preprint arXiv:2305.15706*.

Tian, Yingjie, Xiaotong Yu and Saiji Fu (2023). 'Partial label learning: Taxonomy, analysis and outlook'. In: *Neural Networks* 161, pp. 708–734. ISSN: 0893-6080.

Tolpegin, Vale et al. (2020). 'Data poisoning attacks against federated learning systems'. In: *Computer security–ESORICs 2020: 25th European symposium on research in computer security, ESORICs 2020, guildford, UK, September 14–18, 2020, proceedings, part i 25*. Springer.

Törnblom, Kjell Y and Dan R Jonsson (1985). 'Subrules of the equality and contribution principles: Their perceived fairness in distribution and retribution'. In: *Social Psychology Quarterly*, pp. 249–261.

Voigt, Paul and Axel Von dem Bussche (2017). 'The eu general data protection regulation (gdpr)'. In: *A Practical Guide, 1st Ed., Cham: Springer International Publishing* 10.3152676, pp. 10–5555.

Wang, Hexu et al. (2022). 'Modelling, simulation and optimisation of medical enterprise warehousing process based on FlexSim model and greedy algorithm'. In: *International Journal of Bio-Inspired Computation* 19.1, pp. 59–66.

Wang, Jianyu et al. (2020). 'SlowMo: Improving Communication-Efficient Distributed SGD with Slow Momentum'. In: *International Conference on Learning Representations*.

Wang, Shiqiang et al. (2019). 'Adaptive federated learning in resource constrained edge computing systems'. In: *IEEE JSAC* 37.6, pp. 1205–1221.

Wen, Hongwei et al. (2021). 'Leveraged weighted loss for partial label learning'. In: *International conference on machine learning*. PMLR, pp. 11091–11100.

Xiao, Han, Kashif Rasul and Roland Vollgraf (2017). 'Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms'. In: *arXiv preprint arXiv:1708.07747*.

Xu, An and Heng Huang (2022). 'Coordinating momenta for cross-silo federated learning'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 8, pp. 8735–8743.

Xu, Ning, Yun-Peng Liu and Xin Geng (2020). 'Partial multi-label learning with label distribution'. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34, pp. 6510–6517.

Xu, Ning et al. (2021). 'Progressive enhancement of label distributions for partial multilabel learning'. In: *IEEE Transactions on Neural Networks and Learning Systems* 34.8, pp. 4856–4867.

Yan, Yan and Yuhong Guo (2024). 'Federated partial label learning with local-adaptive augmentation and regularization'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 15, pp. 16272–16280.

Yan, Yan et al. (2018). 'A Unified Analysis of Stochastic Momentum Methods for Deep Learning'. In: *IJCAI*, pp. 2955–2961.

Yang, Zhengjie et al. (2022a). 'FastSlowMo: Federated Learning With Combined Worker and Aggregator Momenta'. In: *IEEE Transactions on Artificial Intelligence*.

Yang, Zhengjie et al. (2022b). 'Federated Learning With Nesterov Accelerated Gradient'. In: *IEEE Transactions on Parallel and Distributed Systems* 33.12, pp. 4863–4873.

Yin, Dong et al. (2018). 'Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates'. In: *Proceedings of the 35th International Conference on Machine Learning*.

Yu, Fei and Min-Ling Zhang (2016). 'Maximum margin partial label learning'. In: *Asian conference on machine learning*. PMLR, pp. 96–111.

Yu, Fuxun et al. (2021). ‘Fed2: Feature-aligned federated learning’. In: *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 2066–2074.

Yurochkin, Mikhail et al. (2019). ‘Bayesian nonparametric federated learning of neural networks’. In: *International conference on machine learning*. PMLR, pp. 7252–7261.

Zantalis, Fotios et al. (2019). ‘A review of machine learning and IoT in smart transportation’. In: *Future Internet* 11.4, p. 94.

Zeng, Yuchen, Hongxu Chen and Kangwook Lee (2021). ‘Improving fairness via federated learning’. In: *arXiv preprint arXiv:2110.15545*.

— (2023). ‘Federated Learning with Local Fairness Constraints’. In: *2023 IEEE International Symposium on Information Theory (ISIT)*. IEEE.

Zeng, Zinan et al. (2013). ‘Learning by associating ambiguously labeled images’. In: *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pp. 708–715.

Zhang, Aston et al. (2021a). ‘Dive into deep learning’. In: *arXiv preprint arXiv:2106.11342*.

Zhang, Brian Hu, Blake Lemoine and Margaret Mitchell (2018). ‘Mitigating unwanted biases with adversarial learning’. In: *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*.

Zhang, Daniel Yue, Ziyi Kou and Dong Wang (2020a). ‘Fairfl: A fair federated learning approach to reducing demographic bias in privacy-sensitive classification models’. In: *IEEE International Conference on Big Data*. IEEE.

Zhang, Fei et al. (2021b). ‘Exploiting class activation value for partial-label learning’. In: *International conference on learning representations*.

Zhang, Kaiyue et al. (2022). ‘Challenges and future directions of secure federated learning: a survey’. In: *Frontiers of computer science* 16, pp. 1–8.

Zhang, Xianglong et al. (2020b). ‘A privacy-preserving and verifiable federated learning scheme’. In: *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, pp. 1–6.

Zhang, Yonggang et al. (2024). ‘Robust training of federated models with extremely label deficiency’. In: *arXiv preprint arXiv:2402.14430*.

Zhou, Yu and Hong Gu (2018). ‘Geometric mean metric learning for partial label data’. In: *Neurocomputing* 275, pp. 394–402. ISSN: 0925-2312.

Zhu, Hangyu et al. (2021a). ‘Federated learning on non-IID data: A survey’. In: *Neurocomputing* 465, pp. 371–390.

Zhu, Zhuangdi, Junyuan Hong and Jiayu Zhou (2021b). ‘Data-Free Knowledge Distillation for Heterogeneous Federated Learning’. In: *Proceedings of the 38th International Conference on Machine Learning*. PMLR, pp. 12878–12889.

APPENDIX A

Appendix for pFedMo

A1 Proof of Theorem 1

First, we define $\boldsymbol{v}_i^t \triangleq \boldsymbol{m}_i^t - \boldsymbol{m}_i^{t-1}$ with $\boldsymbol{v}_i^0 = \mathbf{0}, \forall i$. We can obtain $\boldsymbol{w}_i^{t-1} = \boldsymbol{m}_i^{t-1} + \gamma \boldsymbol{v}_i^{t-1}$. (3.2) and (3.3) can then be equivalently written as

$$\boldsymbol{v}_i^t \leftarrow \gamma \boldsymbol{v}_i^{t-1} - \eta \nabla F_i(\boldsymbol{w}_i^{t-1}), \tag{A.1}$$

$$\boldsymbol{w}_i^t \leftarrow \boldsymbol{w}_i^{t-1} + \gamma \boldsymbol{v}_i^t - \eta \nabla F_i(\boldsymbol{w}_i^{t-1}). \tag{A.2}$$

Identically, the equivalent update of the representation model follows (A.1) and (A.2) by replacing subscript i with r. The aggregated value $\boldsymbol{v}^t$ and $\boldsymbol{w}^t$ can also be equivalently written as

$$\boldsymbol{v}^t \leftarrow \sum_{i=1}^{N} \frac{D_i}{D} \boldsymbol{v}_i^t, \quad \boldsymbol{w}^t \leftarrow \sum_{i=1}^{N} \frac{D_i}{D} \boldsymbol{w}_i^t. \tag{A.3}$$

Similarly, the virtual updates (3.11)–(3.12) can be equivalently written as

$$\begin{aligned} \boldsymbol{v}_{[k]}^t \leftarrow & \gamma \boldsymbol{v}_{[k]}^{t-1} - \eta \nabla F(\boldsymbol{w}_{[k]}^{t-1}), \\ \boldsymbol{w}_{[k]}^t \leftarrow & \boldsymbol{w}_{[k]}^{t-1} + \gamma \boldsymbol{v}_{[k]}^t - \eta \nabla F(\boldsymbol{w}_{[k]}^{t-1}). \end{aligned} \tag{A.4}$$

We employ the above equivalent update format (A.1)–(A.4) to complete the proof in the rest of the Appendix.

Then, we define the constants as follows for convenient proof presentation.

$$
\begin{aligned}
C_1 &\triangleq \frac{(1+\eta\beta)(1+\gamma)+\sqrt{(1+\eta\beta)^2(1+\gamma)^2-4\gamma(1+\eta\beta)}}{2\gamma},\\
C_2 &\triangleq \frac{(1+\eta\beta)(1+\gamma)-\sqrt{(1+\eta\beta)^2(1+\gamma)^2-4\gamma(1+\eta\beta)}}{2\gamma},\\
C_3 &\triangleq \frac{\gamma C_1+C_1-1}{(C_1-C_2)(\gamma C_1-1)}\\
C_4 &\triangleq \frac{\gamma C_2+C_2-1}{(C_1-C_2)(1-\gamma C_2)},\\
C_5 &\triangleq \frac{\frac{1+\eta\beta+\eta\beta\gamma}{\gamma}-C_2}{C_1-C_2}=\frac{C_1-1}{C_1-C_2},\\
C_6 &\triangleq \frac{C_1-\frac{1+\eta\beta+\eta\beta\gamma}{\gamma}}{C_1-C_2}=\frac{1-C_2}{C_1-C_2}.
\end{aligned}
$$

We can see that (A.1)–(A.4) are equivalent to equations (4)–(7), (13), and (14) in [Yang et al. 2022b]. Therefore, we replace $\mathbf{v}_i, \mathbf{w}_i, \mathbf{v}, \mathbf{w}, \mathbf{v}_{[k]}, \mathbf{w}_{[k]}$ and constants A, B, C, D, E, F in Yang et al. 2022b with $\boldsymbol{v}_i, \boldsymbol{w}_i, \boldsymbol{v}, \boldsymbol{w}, \boldsymbol{v}_{[k]}, \boldsymbol{w}_{[k]}$, and constants $C_1, C_2, C_5, C_6, C_3, C_4$ in chapter 3 respectively. According to [Yang et al. 2022b, Theorem 1], we can directly complete the derivation of Theorem 1.

A2 Proof of Theorem 2

Since $\boldsymbol{w}_+^{k\tau}-\boldsymbol{w}^{k\tau}=\sum_{i=1}^N \frac{D_i}{D}\boldsymbol{w}_{i+}^{k\tau}-\boldsymbol{w}^{k\tau}$, we first substitute (11) into (12) to eliminate $\boldsymbol{m}_{i+}^{k\tau}$, and then substitute (12) into above formula. After simplifying the formula, we have

$$
\boldsymbol{w}_+^{k\tau}-\boldsymbol{w}^{k\tau}=\pi\sum_{i=1}^N \frac{D_i}{D}s_i^k\left((\boldsymbol{w}_r^{k\tau}-\boldsymbol{m}_r^{k\tau})-(\boldsymbol{w}^{k\tau}-y^{k\tau})\right). \tag{A.5}
$$

According to (7) and (A.1), we have $\boldsymbol{w}^{k\tau}-\boldsymbol{m}^{k\tau}=\sum_{i=1}^N \frac{D_i}{D}\left(\boldsymbol{w}_i^{k\tau}-\boldsymbol{m}_i^{k\tau}\right)=\gamma\sum_{i=1}^N \frac{D_i}{D}\boldsymbol{v}_i^{k\tau}$ and $\boldsymbol{w}_r^{k\tau}-\boldsymbol{m}_r^{k\tau}=\gamma\boldsymbol{v}_r^{k\tau}$. We also define

$$
\mu \triangleq \max_{k\in[1,K],\forall t,i}\left\{\frac{\|\gamma(\boldsymbol{v}_{[k]}^t)\|}{\|\eta\nabla F(\boldsymbol{w}_{[k]}^t)\|},\frac{\|\gamma(\boldsymbol{v}_i^t)\|}{\|\eta\nabla F_i(\boldsymbol{w}_i^t)\|},\frac{\|\gamma(\boldsymbol{v}_r^t)\|}{\|\eta\nabla F_r(\boldsymbol{w}_r^t)\|}\right\}. \tag{A.6}
$$

Because $F_i(\cdot)$ is ρ-Lipschitz, and according to [49, Lecture 2, Lemma 1], we have $\|\nabla F_i(\cdot)\|^2 \leq \rho^2$ and $\|\nabla F_r(\cdot)\|^2 \leq \rho^2$. Therefore, according to (A.5), (A.6), and definition of q^k in Theorem 2, we can derive

$$\left\|\boldsymbol{w}_+^{k\tau} - \boldsymbol{w}^{k\tau}\right\| = \pi \sum_{i=1}^{N} \frac{D_i}{D} s_i^k \left\|\gamma \boldsymbol{v}_r^{k\tau} - \gamma \boldsymbol{v}_i^{k\tau}\right\| \leq 2\mu\eta\rho q^k \pi. \tag{A.7}$$

We complete the proof of Theorem 2.

A3 Proof of Theorem 3

We have established the upper bound between the personalized model $\boldsymbol{w}_+^{k\tau}$ and the model without personalization $\boldsymbol{w}^{k\tau}$. Afterward, we can utilize this upper bound, following [Yang et al. 2022b, Theorem 2], to prove Theorem 3.

For convenience, we define $c_{[k]}(t) \triangleq F(\boldsymbol{w}_{[k]}^t) - F(\boldsymbol{w}^*)$ for a given interval $[k]$, where $t \in [(k-1)\tau, k\tau]$. We also define the following constants in this section.

$$\begin{aligned}
\omega &\triangleq \min_{k\in[1,K], t\in[k]} \frac{1}{\|\boldsymbol{w}_{[k]}^t - \boldsymbol{w}^*\|^2}, \\
\sigma &\triangleq \min_{k\in[1,K], t_1,t_2\in[k]} \frac{\|\nabla F(\boldsymbol{w}_{[k]}^{t_1})\|}{\|\nabla F(\boldsymbol{w}_{[k]}^{t_2})\|}, \qquad \text{(A.8)} \\
\varphi &\triangleq \max_{k\in[1,K]} \left\{q^k\right\}, \qquad \text{(A.9)} \\
\alpha &\triangleq \eta(\gamma+1)\left(1 - \frac{\beta\eta(\gamma+1)}{2}\right) \\
&\quad - \frac{\beta\eta^2\gamma^2\mu^2}{2} - \eta\gamma\mu(1-\beta\eta(\gamma+1)). \qquad \text{(A.10)}
\end{aligned}$$

According to the convergence lower bound of any gradient descent methods given in [Bubeck 2014, Theorem 3.14], we always have $c_{[k]}(t) > 0$ for any t and k. Then we derive the upper bound of $c_{[k]}(t+1) - c_{[k]}(t)$, where $t \in [(k-1)\tau, k\tau - 1]$. Because $F(\cdot)$ is β-smooth, based

on [Bubeck 2014, Lemma 3.4], we have

$$
\begin{aligned}
&c_{[k]}(t+1) - c_{[k]}(t) = F(\boldsymbol{w}_{[k]}^{t+1}) - F(\boldsymbol{w}_{[k]}^{t}) \\
\leq &\langle \nabla F(\boldsymbol{w}_{[k]}^{t}), \boldsymbol{w}_{[k]}^{t+1} - \boldsymbol{w}_{[k]}^{t} \rangle + \frac{\beta}{2} \|\boldsymbol{w}_{[k]}^{t+1} - \boldsymbol{w}_{[k]}^{t}\|^2 \\
= &\gamma \langle \nabla F(\boldsymbol{w}_{[k]}^{t}), \boldsymbol{v}_{[k]}^{t+1} \rangle - \eta \|\nabla F(\boldsymbol{w}_{[k]}^{t})\|^2 \\
&+ \frac{\beta}{2} \|\gamma \boldsymbol{v}_{[k]}^{t+1} - \eta \nabla F(\boldsymbol{w}_{[k]}^{t})\|^2 \\
\overset{(a)}{=} &- \eta(\gamma+1) \left(1 - \frac{\beta\eta(\gamma+1)}{2}\right) \|\nabla F(\boldsymbol{w}_{[k]}^{t})\|^2 \\
&+ \frac{\beta\gamma^4}{2} \|\boldsymbol{v}_{[k]}^{t}\|^2 + \gamma^2 \left(1 - \beta\eta(\gamma+1)\right) \langle \nabla F(\boldsymbol{w}_{[k]}^{t}), \boldsymbol{v}_{[k]}^{t} \rangle \\
\overset{(b)}{\leq} &\left(-\eta(\gamma+1) \left(1 - \frac{\beta\eta(\gamma+1)}{2}\right) + \frac{\beta\eta^2\gamma^2\mu^2}{2} \right. \\
&\left. + \eta\gamma\mu(1 - \beta\eta(\gamma+1)) \right) \|\nabla F(\boldsymbol{w}_{[k]}^{t})\|^2,
\end{aligned}
\tag{A.11}
$$

where (a) is replacing $\boldsymbol{v}_{[k]}^{t+1}$ by (A.4) and rearranging the formula; (b) is because $\|\gamma \boldsymbol{v}_{[k]}^{t}\| \leq \mu \|\eta \nabla F(\boldsymbol{w}_{[k]}^{t})\|$ with the definition of μ. According to Cauchy-Schwarz inequality, we can obtain $\langle \nabla F(\boldsymbol{w}_{[k]}^{t}), \boldsymbol{v}_{[k]}^{t} \rangle \leq \|\nabla F(\boldsymbol{w}_{[k]}^{t})\| \|\boldsymbol{v}_{[k]}^{t}\| \leq \frac{\mu\eta}{\gamma} \|\nabla F(\boldsymbol{w}_{[k]}^{t})\|^2$. According to the definition of α, and Condition (2.1) of Theorem 3 with

$$
f(\tau) \geq 0 \tag{A.12}
$$

which can be directly proved by [Yang et al. 2022b, Appendix C], we have $\alpha > 0$. Then from (A.11), we have

$$
c_{[k]}(t+1) \leq c_{[k]}(t) - \alpha \|\nabla F(\boldsymbol{w}_{[k]}^{t})\|^2. \tag{A.13}
$$

Because $F(\cdot)$ is ρ-Lipschitz, and according to [Mitliagkas and Gallego 2021, Lecture 2, Lemma 1], there exists a point $\boldsymbol{w}_{[k]}^{t_2}$ such that $F(\boldsymbol{w}_{[k]}^{t}) - F(\boldsymbol{w}^*) = \langle \nabla F(\boldsymbol{w}_{[k]}^{t_2}), \boldsymbol{w}_{[k]}^{t} - \boldsymbol{w}^* \rangle$. Hence, by Cauchy-Schwarz inequality, we have $c_{[k]}(t) = F(\boldsymbol{w}_{[k]}^{t}) - F(\boldsymbol{w}^*) \leq \|\nabla F(\boldsymbol{w}_{[k]}^{t_2})\| \|\boldsymbol{w}_{[k]}^{t} - \boldsymbol{w}^*\|$. Based on the definition of σ, and replacing t with t_1, we have

$\|\nabla F(\boldsymbol{w}_{[k]}^t)\| \geq \sigma\|\nabla F(\boldsymbol{w}_{[k]}^{t_2})\|$. Thus, $\|\nabla F(\boldsymbol{w}_{[k]}^t)\| \geq \sigma\|\nabla F(\boldsymbol{w}_{[k]}^{t_2})\| \geq \frac{\sigma c_{[k]}(t)}{\|\boldsymbol{w}_{[k]}^t - \boldsymbol{w}^*\|}$. Substituting above inequality into (A.13), and noting $\omega \leq \frac{1}{\|\boldsymbol{w}_{[k]}^t - \boldsymbol{w}^*\|^2}$ by the definition of ω, we get $c_{[k]}(t+1) \leq c_{[k]}(t) - \frac{\alpha\sigma^2 c_{[k]}(t)^2}{\|\boldsymbol{w}_{[k]}^t - \boldsymbol{w}^*\|^2} \leq c_{[k]}(t) - \omega\alpha\sigma^2 c_{[k]}(t)^2$. Because $\alpha > 0$, $c_{[k]}(t) > 0$, and (A.13), we have $0 < c_{[k]}(t+1) \leq c_{[k]}(t)$. Dividing both sides by $c_{[k]}(t+1)c_{[k]}(t)$, we get $\frac{1}{c_{[k]}(t)} \leq \frac{1}{c_{[k]}(t+1)} - \omega\alpha\sigma^2\frac{c_{[k]}(t)}{c_{[k]}(t+1)}$. We note that $\frac{c_{[k]}(t)}{c_{[k]}(t+1)} \geq 1$. Thus, $\frac{1}{c_{[k]}(t+1)} - \frac{1}{c_{[k]}(t)} \geq \omega\alpha\sigma^2\frac{c_{[k]}(t)}{c_{[k]}(t+1)} \geq \omega\alpha\sigma^2$. Summing up the above inequality by $t \in [(k-1)\tau, k\tau - 1]$, we have $\frac{1}{c_{[k]}(k\tau)} - \frac{1}{c_{[k]}((k-1)\tau)} = \sum_{t=(k-1)\tau}^{k\tau-1}\left(\frac{1}{c_{[k]}(t+1)} - \frac{1}{c_{[k]}(t)}\right) \geq \sum_{t=(k-1)\tau}^{k\tau-1}\omega\alpha\sigma^2 = \tau\omega\alpha\sigma^2$. Then, we sum up the above inequality by $k \in [1, K]$, after rearranging the left-hand side and noting that $T = K\tau$, we can get

$$
\begin{aligned}
&\sum_{k=1}^{K}\left(\frac{1}{c_{[k]}(k\tau)} - \frac{1}{c_{[k]}((k-1)\tau)}\right) \\
=&\frac{1}{c_{[k]}(T)} - \frac{1}{c_{\{1\}}(0)} - \sum_{k=1}^{K-1}\left(\frac{1}{c_{[k+1]}(k\tau)} - \frac{1}{c_{[k]}(k\tau)}\right) \\
\geq& k\tau\omega\alpha\sigma^2 = T\omega\alpha\sigma^2.
\end{aligned} \tag{A.14}
$$

Following (A.14), we note that

$$
\begin{aligned}
&\frac{1}{c_{[k+1]}(k\tau)} - \frac{1}{c_{[k]}(k\tau)} \\
=&\frac{c_{[k]}(k\tau) - c_{[k+1]}(k\tau)}{c_{[k]}(k\tau)c_{[k+1]}(k\tau)} \\
=&\frac{F(\boldsymbol{w}_{[k]}^{k\tau}) - F(\boldsymbol{w}_{[k+1]}^{k\tau})}{c_{[k]}(k\tau)c_{[k+1]}(k\tau)} \\
=&\frac{F(\boldsymbol{w}_{[k]}^{k\tau}) - F(\boldsymbol{w}^{k\tau}) + \left(F(\boldsymbol{w}^{k\tau}) - F(\boldsymbol{w}_{+}^{k\tau})\right)}{c_{[k]}(k\tau)c_{[k+1]}(k\tau)} \\
\overset{(a)}{\geq}&\frac{-\rho\left(f(\tau) + 2\mu\eta\rho\varphi\pi\right)}{c_{[k]}(k\tau)c_{[k+1]}(k\tau)},
\end{aligned} \tag{A.15}
$$

where (a) is because of combining Theorem 1 and Theorem 2, and definition of φ in (A.9).

From (A.13), we can get $F(\boldsymbol{w}_{[k]}^{t}) \geq F(\boldsymbol{w}_{[k]}^{t+1})$ for any $t \in [(k-1)\tau, k\tau)$. Recalling Condition (2.2) in Theorem 3, where $F(\boldsymbol{w}_{[k]}(k\tau)) - F(\boldsymbol{w}^*) \geq \varepsilon$ for all k, we can obtain $c_{[k]}(t) = F(\boldsymbol{w}_{[k]}^{t}) - F(\boldsymbol{w}^*) \geq \varepsilon$ for all $t \in [(k-1)\tau, k\tau]$ and k. Thus, $c_{[k]}(k\tau)c_{[k+1]}(k\tau) \geq \varepsilon^2$. Based on (A.12), substituting above inequalities into (A.15), we obtain $\frac{1}{c_{[k+1]}(k\tau)} - \frac{1}{c_{[k]}(k\tau)} \geq \frac{-\rho(f(\tau)+2\mu\eta\rho\varphi\pi)}{\varepsilon^2}$. Substituting the above inequality into (A.14) and rearrange, we get

$$\frac{1}{c_{[k]}(T)} - \frac{1}{c_{\{1\}}(0)} \geq T\omega\alpha\sigma^2 - (K-1)\frac{\rho\left(f(\tau)+2\mu\eta\rho\varphi\pi\right)}{\varepsilon^2}. \tag{A.16}$$

Recalling Condition (2.3) in Theorem 3, where $F(\boldsymbol{w}^T) - F(\boldsymbol{w}^*) \geq \varepsilon$, and noting that $c_{[k]}(T) \geq \varepsilon$, we get $(F(\boldsymbol{w}^T) - F(\boldsymbol{w}^*))c_{[k]}(T) \geq \varepsilon^2$. Thus,

$$\begin{aligned}
&\frac{1}{F(\boldsymbol{w}^T) - F\left(\boldsymbol{w}^*\right)} - \frac{1}{c_{[k]}(T)} \\
=&\frac{c_{[k]}(T) - (F(\boldsymbol{w}^T) - F\left(\boldsymbol{w}^*\right))}{(F(\boldsymbol{w}^T) - F\left(\boldsymbol{w}^*\right))c_{[k]}(T)} \\
=&\frac{F(\boldsymbol{w}_{[k]}^{T}) - F(\boldsymbol{w}^T)}{(F(\boldsymbol{w}^T) - F\left(\boldsymbol{w}^*\right))c_{[k]}(T)} \\
\geq&\frac{-\rho\left(f(\tau)+2\mu\eta\rho\varphi\pi\right)}{(F(\boldsymbol{w}^T) - F\left(\boldsymbol{w}^*\right))c_{[k]}(T)} \\
\geq&-\frac{\rho\left(f(\tau)+2\mu\eta\rho\varphi\pi\right)}{\varepsilon^2},
\end{aligned} \tag{A.17}$$

where the first inequality follows the same method to prove (A.15). Combining (A.16) with (A.17), we get $\frac{1}{F(\boldsymbol{w}^T)-F(\boldsymbol{w}^*)} - \frac{1}{c_{\{1\}}(0)} \geq T\omega\alpha\sigma^2 - K\frac{\rho(f(\tau)+2\mu\eta\rho\varphi\pi)}{\varepsilon^2} = T\omega\alpha\sigma^2 - T\frac{\rho(f(\tau)+2\mu\eta\rho\varphi\pi)}{\tau\varepsilon^2}$ $= T\left(\omega\alpha\sigma^2 - \frac{\rho(f(\tau)+2\mu\eta\rho\varphi\pi)}{\tau\varepsilon^2}\right)$. Noting that $c_{[1]}(0) = F(\boldsymbol{w}_{[1]}^{0}) - F(\boldsymbol{w}^*) > 0$, the above inequality can be expressed as $\frac{1}{F(\boldsymbol{w}^T)-F(\boldsymbol{w}^*)} \geq T\left(\omega\alpha\sigma^2 - \frac{\rho(f(\tau)+2\mu\eta\rho\varphi\pi)}{\tau\varepsilon^2}\right)$. Recalling Condition (2.1) in Theorem 3, where $\omega\alpha\sigma^2 - \frac{\rho(f(\tau)+2\mu\eta\rho\varphi\pi)}{\tau\varepsilon^2} > 0$, we obtain that the right-hand side of above inequality is greater than zero. Therefore, taking the reciprocal of the above inequality, we finally complete the proof of Theorem 3.

APPENDIX B

Appendix for pFedPLL

B1 Preliminaries

To prove the convergence, we propose a virtual relation module update as if each worker's isolated relation module is aggregated and redistributed. The virtual relation module update just like real representation module update (Lines 16–17 in Algorithm 3) as follows:

$$\boldsymbol{w}_{t+1}^{rel} = \sum_{n=1}^{N} s_t^n \boldsymbol{w}_{t,\tau}^{n,rel}, \tag{B.1}$$

$$\boldsymbol{w}_{t+1,0}^{n,rel} = \boldsymbol{w}_{t+1}^{rel}. \tag{B.2}$$

Thus, we define the complete virtual global model as

$$\boldsymbol{w}_{t+1} \triangleq [\boldsymbol{w}_{t+1}^{rep}, \boldsymbol{w}_{t+1}^{rel}], \tag{B.3}$$

where $\boldsymbol{w}_{t+1}$ represents the complete global model at the $(t+1)$th global iteration. Based on (B.2) and Line 17 of Algorithm 3, the local complete model is given as

$$\boldsymbol{w}_{t+1,0}^{n} = [\boldsymbol{w}_{t+1,0}^{n,rep}, \boldsymbol{w}_{t+1,0}^{n,rel}], \forall n \in N. \tag{B.4}$$

At the start of local training, the initial worker's representation and relation modules are assigned from the global representation and relation modules as described in (B.4). At this point, the local model is identical to the global model. Thus,

$$\boldsymbol{w}_{t+1,0}^{n} = \boldsymbol{w}_{t+1}, \forall n \in N. \tag{B.5}$$

The gradient of the global loss function can be decomposed into two components, which we define as

$$\nabla F(\boldsymbol{w}_{t+1}) \triangleq [\nabla_{\boldsymbol{w}^{rep}} F(\boldsymbol{w}_{t+1}), \nabla_{\boldsymbol{w}^{rel}} F(\boldsymbol{w}_{t+1})]. \tag{B.6}$$

The gradient of the worker's loss function can be decomposed into two components, which we define as

$$\begin{aligned}\nabla F_n(\boldsymbol{w}^n_{t+1,v}) \triangleq [\nabla_{\boldsymbol{w}^{rep}} F_n(\boldsymbol{w}^n_{t+1,v})), \nabla_{\boldsymbol{w}^{rel}} F_n(\boldsymbol{w}^n_{t+1,v})], \\ \forall n \in N, \forall v \in [0, \ldots, \tau - 1].\end{aligned} \tag{B.7}$$

Based on the definition in (B.7), we have

$$\begin{aligned}
&\|\nabla F_n(\boldsymbol{w}^n_{t+1,v})\|_2^2 \\
=&\|[\nabla_{\boldsymbol{w}^{rep}} F_n(\boldsymbol{w}^n_{t+1,v})), \nabla_{\boldsymbol{w}^{rel}} F_n(\boldsymbol{w}^n_{t+1,v})]\|_2^2 \\
=&(\|[\nabla_{\boldsymbol{w}^{rep}} F_n(\boldsymbol{w}^n_{t+1,v})), 0] + [0, \nabla_{\boldsymbol{w}^{rel}} F_n(\boldsymbol{w}^n_{t+1,v})])\|_2)^2 \\
=&\|\nabla_{\boldsymbol{w}^{rep}} F_n(\boldsymbol{w}^n_{t+1,v})\|_2^2 \\
&+ 2\langle[\nabla_{\boldsymbol{w}^{rep}} F_n(\boldsymbol{w}^n_{t+1,v})), 0], [0, \nabla_{\boldsymbol{w}^{rel}} F_n(\boldsymbol{w}^n_{t+1,v}]\rangle \\
&+ \|\nabla_{\boldsymbol{w}^{rel}} F_n(\boldsymbol{w}^n_{t+1,v}]\|_2^2. \\
=&\|\nabla_{\boldsymbol{w}^{rep}} F_n(\boldsymbol{w}^n_{t+1,v})\|_2^2 + \|\nabla_{\boldsymbol{w}^{rel}} F_n(\boldsymbol{w}^n_{t+1,v})]\|_2^2, \\
&\forall k \in K, \forall h \in [0, \ldots, \tau - 1]
\end{aligned} \tag{B.8}$$

According to Lines 8–9 in Algorithm 3, (B.4) and (B.7), we have the local update rule as

$$\boldsymbol{w}^n_{t,v+1} = \boldsymbol{w}^n_{t,v} - \eta \nabla F_n(\boldsymbol{w}^n_{t,v}). \tag{B.9}$$

To calculate the difference between $v = 0$ and $v = \tau - 1$ according to (B.9), we have

$$\boldsymbol{w}^n_{t,0} - \boldsymbol{w}^n_{t,\tau-1} = \eta \sum_{v=0}^{\tau-1} \nabla F_n(\boldsymbol{w}^n_{t,v}). \tag{B.10}$$

According to (B.5), (B.7), and rearranging (B.10), we obtain

$$\boldsymbol{w}_t - \boldsymbol{w}_{t,\tau}^n = \eta \sum_{v=0}^{\tau-1} [\nabla_{\boldsymbol{w}^{rep}} F_n(\boldsymbol{w}_{t,v}^n), \nabla_{\boldsymbol{w}^{rel}} F_n(\boldsymbol{w}_{t,v}^n)]. \tag{B.11}$$

Based on Lines 16–17 in Algorithm 3, (B.1), and (B.3), we have

$$\boldsymbol{w}_t - \boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \sum_{n=1}^{N} s_t^n(\boldsymbol{w}_{t,\tau-1}^n) = \sum_{n=1}^{N} s_t^n(\boldsymbol{w}_t - \boldsymbol{w}_{t,\tau}^n). \tag{B.12}$$

Substituting the (B.11) into (B.12), we have

$$\boldsymbol{w}_t - \boldsymbol{w}_{t+1} = \eta \sum_{n=1}^{N} s_t^n \sum_{v=0}^{\tau-1} [\nabla_{\boldsymbol{w}^{rep}} F_n(\boldsymbol{w}_{t,v}^n), \nabla_{\boldsymbol{w}^{rel}} F_n(\boldsymbol{w}_{t,v}^n)]. \tag{B.13}$$

By rearranging (B.13), we obtain

$$\boldsymbol{w}_{t+1} - \boldsymbol{w}_t = -\eta \sum_{n=1}^{N} s_t^n \sum_{v=0}^{\tau-1} [\nabla_{\boldsymbol{w}^{rep}} F_n(\boldsymbol{w}_{t,v}^n), \nabla_{\boldsymbol{w}^{rel}} F_n(\boldsymbol{w}_{t,v}^n)]. \tag{B.14}$$

By rearranging (B.14), we derive the global update rule as

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \sum_{n=1}^{N} s_t^n \sum_{v=0}^{\tau-1} [\nabla_{\boldsymbol{w}^{rep}} F_n(\boldsymbol{w}_{t,v}^n), \nabla_{\boldsymbol{w}^{rel}} F_n(\boldsymbol{w}_{t,v}^n)], \tag{B.15}$$

which is the basis to prove the convergence.

We assume $F_n(\cdot)$ satisfies the following standard conditions that are necessary for theoretical analysis [Wang et al. 2019; Yang et al. 2022b; Huo et al. 2020].

ASSUMPTION 5. *(Bounded diversity). The variance of stochastic gradient on local workers is upper bounded. So that any* $n \in \{1, \ldots, N\}$*, it is satisfied that*

$$\mathbb{E}_{\xi \sim D_n} \|\nabla F_n(\boldsymbol{w}, \xi) - \nabla F_n(\boldsymbol{w})\|_2^2 \leq \delta^2, \forall \boldsymbol{w}, n.$$

This is equivalent to

$$\begin{aligned} \mathbb{E}_{\xi \sim D_n} [&\|\nabla_{\boldsymbol{w}_{rel}} F_n(\boldsymbol{w}, \xi) - \nabla_{\boldsymbol{w}_{rel}} F_n(\boldsymbol{w})\|_2^2 \\ &+ \|\nabla_{\boldsymbol{w}_{rep}} F_n(\boldsymbol{w}, \xi) - \nabla_{\boldsymbol{w}_{rep}} F_n(\boldsymbol{w})\|_2^2] \leq \delta^2, \forall \boldsymbol{w}, n. \end{aligned}$$

ASSUMPTION 6. *(L-Lipschitz). The gradients of F_n and F are Lipschitz continuous with a constant $L > 0$, so that any $n \in \{1, \ldots, N\}$, it is satisfied that*

$$\|\nabla F_n(\boldsymbol{w}_1) - \nabla F_n(\boldsymbol{w}_2)\|_2 \leq L\|\boldsymbol{w}_1 - \boldsymbol{w}_2\|_2, \forall \boldsymbol{w}_1, \boldsymbol{w}_2, n,$$

$$\|\nabla F(\boldsymbol{w}_1) - \nabla F(\boldsymbol{w}_2)\|_2 \leq L\|\boldsymbol{w}_1 - \boldsymbol{w}_2\|_2, \forall \boldsymbol{w}_1, \boldsymbol{w}_2.$$

B2 Convergence Analysis

In Lemma 1, we first prove the upper bound of pFedPLL between $F(\boldsymbol{w}_{t+1})$ and $F(\boldsymbol{w}_t)$.

LEMMA 1. *Under Assumptions 5 and 6, the update of $\boldsymbol{w}_t$ on the server at each global aggregation is upper bounded as*

$$\begin{aligned}
&\mathbb{E}_{\xi,n} F(\boldsymbol{w}_{t+1}) \\
\leq & F(\boldsymbol{w}_t) - \frac{\eta}{2}\tau \sum_{n=1}^{N} s_t^n \mathbb{E}_{\xi,n} \|\nabla F_n(\boldsymbol{w}_t)\|_2^2 + (\frac{L\tau\eta + 1}{2})\eta^2 L\tau\delta^2 \\
& - (\frac{\eta}{2} - \frac{\eta}{2}L^2\eta^2\tau^2 - \frac{L}{2}\eta^2\tau) \sum_{n=1}^{N} s_t^n \sum_{v=0}^{\tau-1} \mathbb{E}_{\xi,n} \|\nabla F_n(\boldsymbol{w}_{t,v}^n)\|_2^2.
\end{aligned}$$

PROOF. According to Assumptions 6 and (B.6), it holds that

$$\begin{aligned}
\mathbb{E}F(\boldsymbol{w}_{t+1}) \leq & F(\boldsymbol{w}_t) + \mathbb{E}\langle \nabla_{\boldsymbol{w}^{rep}} F(\boldsymbol{w}_t), \boldsymbol{w}_{t+1}^{rep} - \boldsymbol{w}_t^{rep} \rangle \\
& + \mathbb{E}\langle \nabla_{\boldsymbol{w}^{rel}} F(\boldsymbol{w}_t), \boldsymbol{w}_{t+1}^{rel} - \boldsymbol{w}_t^{rel} \rangle \\
& + \frac{L}{2}\mathbb{E}\|\boldsymbol{w}_{t+1}^{rep} - \boldsymbol{w}_t^{rep}\|_2^2 \\
& + \frac{L}{2}\mathbb{E}\|\boldsymbol{w}_{t+1}^{rel} - \boldsymbol{w}_t^{rel}\|_2^2. \qquad \text{(B.16)}
\end{aligned}$$

By taking the expectation over the samples, rearranging the inequality in (B.16), and considering (B.14) and (B.15), we obtain

$$
\begin{aligned}
&\mathbb{E}_{\xi} F(\boldsymbol{w}_{t+1}) \\
\leq & F(\boldsymbol{w}_t) - \langle \nabla_{\boldsymbol{w}^{rep}} F(\boldsymbol{w}_t), \mu \sum_{n=1}^{N} s_t^n \sum_{v=0}^{\tau-1} \nabla_{\boldsymbol{w}^{rep}} F_n(\boldsymbol{w}_{t,v}^n, \xi) \rangle \\
&+ \frac{L}{2} \mathbb{E}_{\xi} \| \eta \sum_{n=1}^{N} s_t^n \sum_{v=0}^{\tau-1} \nabla_{\boldsymbol{w}^{rep}} F_n(\boldsymbol{w}_{t,v}^n, \xi) \rangle \|_2^2 \\
&- \langle \nabla_{\boldsymbol{w}^{rel}} F(\boldsymbol{w}_t), \mu \sum_{n=1}^{N} s_t^n \sum_{v=0}^{\tau-1} \nabla_{\boldsymbol{w}^{rel}} F_n(\boldsymbol{w}_{t,v}^n, \xi) \rangle \\
&+ \frac{L}{2} \mathbb{E}_{\xi} \| \eta \sum_{n=1}^{N} s_t^n \sum_{v=0}^{\tau-1} \nabla_{\boldsymbol{w}^{rel}} F_n(\boldsymbol{w}_{t,v}^n, \xi) \rangle \|_2^2,
\end{aligned} \tag{B.17}
$$

where the inequality follows from $\mathbb{E}_{\xi}[\nabla_{\boldsymbol{w}^{rep}} F_n(\boldsymbol{w}, \xi)] = \nabla_{\boldsymbol{w}^{rep}} F_n(\boldsymbol{w})$ and $\mathbb{E}_{\xi}[\nabla_{\boldsymbol{w}^{rel}} F_n(\boldsymbol{w}, \xi)] = \nabla_{\boldsymbol{w}^{rel}} F_n(\boldsymbol{w})$. Taking the expectations over the workers, we have

$$
\begin{aligned}
&\mathbb{E}_{\xi,n} F(\boldsymbol{w}_{t+1}) \\
\leq & F(\boldsymbol{w}_t) - \frac{\eta}{2} \sum_{n=1}^{N} s_t^n \sum_{v=0}^{\tau-1} \mathbb{E}_{\xi,n} [\| \nabla F_n(\boldsymbol{w}_t) \|_2^2] \\
&- \frac{\eta}{2} \sum_{n=1}^{N} s_t^n \sum_{v=0}^{\tau-1} \mathbb{E}_{\xi,n} [\| \nabla F_n(\boldsymbol{w}_{t,v}^n) \|_2^2] \\
&+ \frac{\eta}{2} \sum_{n=1}^{N} s_t^n \sum_{v=0}^{\tau-1} \underbrace{\mathbb{E}_{\xi,n} \big[\| \nabla_{\boldsymbol{w}^{rep}} F_n(\boldsymbol{w}_t) - \nabla_{\boldsymbol{w}^{rep}} F_n(\boldsymbol{w}_{t,v}^n) \|_2^2 + \| \nabla_{\boldsymbol{w}^{rel}} F_n(\boldsymbol{w}_t) - \nabla_{\boldsymbol{w}^{rel}} F_n(\boldsymbol{w}_{t,v}^n) \|_2^2 \big]}_{Q1} \\
&+ \frac{L}{2} \eta^2 \sum_{n=1}^{N} s_t^n \underbrace{\mathbb{E}_{\xi,n} \left[\left\| \sum_{v=0}^{\tau-1} \nabla F_n(\boldsymbol{w}_{t,v}^n, \xi) \right\|_2^2 \right]}_{Q2},
\end{aligned} \tag{B.18}
$$

where the inequality follows from Jensen's inequality, $\langle a, b \rangle = \frac{1}{2}(\|a\|_2^2 + |b\|_2^2 - \|a - b\|_2^2)$. We next prove the upper bound of Q1 as

$$
Q1 \leq L^2 \eta^2 \mathbb{E}_{\xi,n} \| \sum_{j=0}^{v-1} (\nabla_{\boldsymbol{w}^{rep}} F_n(\boldsymbol{w}_{t,j}^n, \xi) - \nabla_{\boldsymbol{w}^{rep}} F_n(\boldsymbol{w}_{t,j}^n)) \|_2^2
$$

$$
\begin{aligned}
&+ L^2\eta^2\mathbb{E}_{\xi,n}\| \sum_{j=0}^{v-1}(\nabla_{\boldsymbol{w}^{rel}}F_n(\boldsymbol{w}_{t,j}^n,\xi) - \nabla_{\boldsymbol{w}^{rel}}F_n(\boldsymbol{w}_{t,j}^n))\|_2^2 \\
&+ L^2\eta^2\mathbb{E}_{\xi,n}(\|(\sum_{j=0}^{v-1}\nabla_{\boldsymbol{w}^{rep}}F_n(\boldsymbol{w}_{t,j}^n))\|_2^2 \\
&+ \|(\sum_{j=0}^{v-1}\nabla_{\boldsymbol{w}^{rel}}F_n(\boldsymbol{w}_{t,j}^n))\|_2^2) \\
\leq & L^2\eta^2\delta^2 h + L^2\eta^2 h\sum_{j=0}^{v-1}\mathbb{E}_{\xi,n}\|\nabla F_n(\boldsymbol{w}_{t,j}^n)\|_2^2, \qquad \text{(B.19)}
\end{aligned}
$$

where the first inequality follows from Assumption 5, $\boldsymbol{w}_t = \boldsymbol{w}_{t,0}^k$, and $\mathbb{E}\|z_1 + \ldots + z_n\|_2^2 \leq \mathbb{E}[\|z_1\|_2^2 + \ldots + \|z_n\|_2^2]$ for any $z_1, \ldots, z_2$. The last inequality is from the Assumption 6, and (B.8). It sums the gradient difference from $j = 0$ to $j = v - 1$. Since the maximum value of v is τ, we replace v with τ in (B.19) and sum from $v = 0$ to $v = \tau - 1$. This still satisfies the inequality in (B.19). Then, we have

$$
Q1 \leq L^2\eta^2\delta^2\tau + L^2\eta^2\tau\sum_{v=0}^{\tau-1}\mathbb{E}_{\xi,n}\|\nabla F_n(\boldsymbol{w}_{t,v}^n)\|_2^2, \qquad \text{(B.20)}
$$

Summing the inequality in (B.20) from $v = 0$ to $\tau - 1$, we have

$$
\sum_{v=0}^{\tau-1} Q1 \leq L^2\eta^2\delta^2\tau^2 + L^2\eta^2\tau^2\sum_{v=0}^{\tau-1}\mathbb{E}_{\xi,n}\|\nabla F_n(\boldsymbol{w}_{t,v}^n)\|_2^2, \qquad \text{(B.21)}
$$

where the inequality comes from $v \leq \tau - 1$. Then we going to prove the upper bound of Q2 as

$$
\begin{aligned}
Q2 =& \mathbb{E}_{\xi,n}\| \sum_{v=0}^{\tau-1}\nabla F_n(\boldsymbol{w}_{t,v}^n,\xi) - \nabla F_n(\boldsymbol{w}_{t,v}^n) + \nabla F_n(\boldsymbol{w}_{t,v}^n)\|_2^2 \\
\leq & \delta^2\tau + \tau\sum_{v=0}^{\tau-1}\mathbb{E}_{\xi,n}\|\nabla F_n(\boldsymbol{w}_{t,v}^n)\|_2^2, \qquad \text{(B.22)}
\end{aligned}
$$

where the inequality comes from Assumption 5. Substituting the upper bound of Q1 and Q2 into inequality (B.18), we have

$$
\mathbb{E}_{\xi,n}F(\boldsymbol{w}_{t+1})
$$

$$\leq F(\boldsymbol{w}_t) - \frac{\eta}{2}\tau \sum_{n=1}^{N} s_t^n \mathbb{E}_{\xi,n} \|\nabla F_n(\boldsymbol{w}_t)\|_2^2 + (\frac{L\tau\eta + 1}{2})\eta^2 L\tau\delta^2$$
$$- (\frac{\eta}{2} - \frac{\eta}{2}L^2\eta^2\tau^2 - \frac{L}{2}\eta^2\tau) \sum_{n=1}^{N} s_t^n \sum_{v=0}^{\tau-1} \mathbb{E}_{\xi,n} \|\nabla F_n(\boldsymbol{w}_{t,v}^n)\|_2^2. \tag{B.23}$$

We have completed the proof of Lemma 1. □

Based on Lemma 1, we can prove the convergence of pFedPLL by telescopically summing from $t = 0$ to $t = T - 1$.

THEOREM 5. *Suppose (1)* $\eta \leq \frac{1}{2L\tau}, \forall t \in 0, \ldots, T-1$*, and (2)* $\exists F_{inf}$ *is the lower bound of* $F(\cdot)$*, we have*

$$\min_{t \in \{0,\ldots,T-1\}} \mathbb{E}_{\xi,n} \|\nabla F(\boldsymbol{w}_t)\|_2^2 \leq \frac{4L}{T}(F(\boldsymbol{w}_0) - F_{inf}))$$
$$+ 3\eta^2 L^2 \delta^2.$$

PROOF. According to Lemma 1, by setting $\eta \leq \frac{1}{2L\tau}$, we ensure that $\frac{\eta}{2} - \frac{\eta}{2}L^2\eta^2\tau^2 - \frac{L}{2}\eta^2\tau \geq 0$. This condition allows us to prove that the algorithm is guaranteed to converge to critical points for smooth non-convex problems. Considering the above condition and taking the expectation of inequality (B.23), upon rearranging, it holds that

$$\mathbb{E}_{\xi,n} \|\nabla F(\boldsymbol{w}_t)\|_2^2 \leq 4L(\mathbb{E}_{\xi,n} F(\boldsymbol{w}_t) - \mathbb{E}_{\xi,n} F(\boldsymbol{w}_{t+1}))$$
$$+ 3\eta^2 L^2 \delta^2. \tag{B.24}$$

Summing it from $t = 0$ to $T - 1$ we have

$$\sum_{t=0}^{T-1} \mathbb{E}_{\xi,n} \|\nabla F(\boldsymbol{w}_t)\|_2^2 \leq 4L(F(\boldsymbol{w}_0) - F(\boldsymbol{w}_T)) + 3\eta^2 L^2 \delta^2 T. \tag{B.25}$$

Let $F_{inf} \leq F(\boldsymbol{w}_T)$, we have that

$$\min_{t \in \{0.,\ldots,T-1\}} \mathbb{E}_{\xi,n} \|\nabla F(\boldsymbol{w}_t)\|_2^2 \leq \frac{4L}{T}(F(\boldsymbol{w}_0) - F_{inf})) \tag{B.26}$$
$$+ 3\eta^2 L^2 \delta^2.$$

We have completed the proof of the Theorem 5. □

Theorem 5 demonstrates that the square of the global model gradient is upper bounded by a function that is inversely proportional to T. The output of the Algorithm 3, $\boldsymbol{w}^{rep}$ is included in the global model $\boldsymbol{w}$, i.e., $\boldsymbol{w} = [\boldsymbol{w}^{rep}, \boldsymbol{w}^{rel}]$. Therefore, we prove that the pFedPLL is convergent with the convergence rate of $O\left(\sqrt{\frac{1}{T}}\right)$.

APPENDIX C

Appendix for FairFedPAPL

TABLE C.1. FairFedPAPL: Summary of Notations in FairFedPAPL

Symbol	Definition	Symbol	Definition
h	global model (hypothesis)	$\epsilon_T(h_T)$	risk of h_T on target domain
h^r	global model at round r	$\hat{\epsilon}_T(h_T)$	empirical risk of h_T on target domain
h_n	local model (hypothesis) for client n	$\epsilon_{\bar{S}}(h_T)$	risk of h_T on samples from source domains
h_n^{r+1}	client n's model at round $r+1$	$\hat{\epsilon}_{\bar{S}}(h_T)$	empirical risk of h_T on samples from source domains
h_S	hypothesis for the source domain	λ_n	optimal risk between client n and target domain
h_T	hypothesis for the target domain	λ_w	optimal risk on the source domain and target domain
$\mathcal{D}_n$	dataset of client n	$1-\delta$	probability over random sample selection
$\mathcal{D}_S$	dataset of the source domain	α	mixup ratio in PAPL attack
$\mathcal{D}_T$	target domain dataset	d	input feature dimension
$\mathcal{D}_t$	filtered dataset of the targeted group	d_j	number of possible values for attribute j
$\mathcal{D}_u$	filtered dataset of the unprivileged class	f_n	fairness metric (EOD) of client n
$\mathcal{D}_{\text{dm}}$	generated dummy dataset	g_t	targeted demographic group
$\mathcal{D}_{\text{rep}}$	reconstructed representation dataset	K	number of classes
$\lvert\mathcal{D}_n\rvert$	size of data of client n	M	number of adversarial clients
$\bar{S}$	sampled data from source domains	N	number of total clients
$\hat{\mathcal{D}}_S$	empirical distribution of source domain	$P_{i,j}$	j-th attribute of the i-th instance
$\hat{\mathcal{D}}_T$	empirical distribution of target dataset	q_n	client n's confidence score
$\hat{\mathcal{D}}_{\bar{S}}$	empirical distribution of $\bar{S}$	ρ	smoothing factor for score update

Symbol	Definition	Symbol	Definition
$\hat{\mathcal{D}}_n$	empirical distribution of client n's dataset	σ	sample size from each local source domain
$\tilde{\mathcal{D}}_m$	poisoned local dataset of client m	τ	VC-dimension of hypothesis class
$\hat{\tilde{\mathcal{D}}}_m$	empirical distribution of poisoned dataset of client m	v_n^{r+1}	local gradient for client n at round $r+1$
$\mathcal{B}$	set of benign clients	w_m	weight of adversarial client m
$\mathcal{J}$	set of adversarial clients	w_n	aggregation weight for client n
$\mathcal{E}$	list of fairness metric values for all clients	η	learning rate
$\mathcal{L}_n$	local loss function for client n	ζ	non-i.i.d. factor
$\mathcal{L}_{\text{rec}}$	GI reconstruction loss	μ	hyperparameter controlling regularization strength
$\mathcal{Q}$	list of confidence scores	$\mathbf{x}_i$	input of the i-th instance
X_{dm}	input of the dummy dataset	$\mathbf{y}_i$	candidate label set of the i-th instance
Y_{dm}	label of dummy dataset	$\hat{P}_i$	sensitive attribute of the i-th instance
y_u	label of unprivileged class	c_n	client n
θ_{high}	confidence upper bound	κ	aggregation threshold
θ_{low}	confidence lower bound		

C1 Summary of Notations

Table C.1 includes the notations used in the problem formulation, proposed fairness attacks, analytical discussion, and the description of FairFedPAPL.

C2 Detailed Derivation of Proposition 1

Detailed Derivation. Let $\mathcal{H}$ denote a hypothesis class with VC-dimension τ. In the FL setting, we consider N source domains $\{\mathcal{D}_n\}_{n=1}^N$ and one target domain $\mathcal{D}_T$. Let $\hat{\mathcal{D}}_n$ and $\hat{\mathcal{D}}_T$ denote the empirical distributions drawn from $\mathcal{D}_n$ and $\mathcal{D}_T$, respectively. The aggregated source domain is then defined as: $\mathcal{D}_S := \sum_{n=1}^N w_n \mathcal{D}_n, \quad \text{with} \quad w_n \geq 0, \quad \sum_{n=1}^N w_n = 1,$ and correspondingly, $\hat{\mathcal{D}}_S$ is the empirical distribution of the aggregated source domain. We define $\bar{S} \sim \hat{\mathcal{D}}_S$ as a set of $N\sigma$ samples, and let $\hat{\mathcal{D}}_{\bar{S}}$ denote its empirical distribution. Each client c_n learns a local hypothesis $h_n \in \mathcal{H}$. The target (global) hypothesis is obtained via weighted aggregation: $h_T = \sum_{n=1}^N w_n h_n$. The expected risk of a hypothesis h on any domain

$\mathcal{D}$ is $\epsilon_{\mathcal{D}}(h) := \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(h(x), y)]$, and its empirical risk on $\mathcal{D}$ is denoted $\hat{\epsilon}_{\mathcal{D}}(h)$. So for the empirical risk of hypothesis h_T on the $\bar{S}$, we have $\hat{\epsilon}_{\bar{S}}\left(h_T\right) = \hat{\epsilon}_{\bar{S}}\left(\sum_{n\in[N]} w_n h_n\right)$.

In our fairness threat model, a subset of M adversarial clients aims to bias the global model ($i.e.$, the target hypothesis h_T) by performing the proposed fairness attacks. Specifically, each adversarial client c_m, where $m \in [M]$, poisons its local source domain $\mathcal{D}_m$, resulting in a poisoned domain $\tilde{\mathcal{D}}_m$ with corresponding empirical distribution $\hat{\tilde{\mathcal{D}}}_m$. In this case, the aggregated empirical distribution of the source domain under attack becomes: $\hat{\mathcal{D}}_S := \sum_{n\in[N\setminus M]} w_n \hat{\mathcal{D}}_n + \sum_{m\in[M]} w_m \hat{\tilde{\mathcal{D}}}_m$, where $w_n \geq 0$, $w_m \geq 0$ and $\sum_{n\in[N\setminus M]} w_n + \sum_{m\in[M]} w_m = 1$. Following the Assumption 1 from the main paper, we know for each adversarial client c_m, the increase in divergence to the empirical target distribution caused by poisoning is upper bounded by a constant β:

$$\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\hat{\tilde{\mathcal{D}}}_m, \hat{\mathcal{D}}_T) - \hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\hat{\mathcal{D}}_m, \hat{\mathcal{D}}_T) \leq \beta, \quad \forall m \in [M].$$

We now derive the upper bound on the divergence between the attacked empirical source distribution and the empirical target distribution:

$$\begin{aligned}
&\hat{d}_{\mathcal{H}\Delta\mathcal{H}}\left(\hat{\mathcal{D}}_{\bar{S}}, \hat{\mathcal{D}}_T\right) \\
=&2 \sup_{A\in\mathcal{A}_{\mathcal{H}\Delta\mathcal{H}}} \left|\Pr_{\hat{\mathcal{D}}_{\bar{S}}}(A) - \Pr_{\hat{\mathcal{D}}_T}(A)\right| \\
=&2 \sup_{A\in\mathcal{A}_{\mathcal{H}\Delta\mathcal{H}}} \left|\sum_{n\in[N\setminus M]} w_n \left(\Pr_{\hat{\mathcal{D}}_n}(A) - \Pr_{\hat{\mathcal{D}}_T}(A)\right) + \sum_{m\in[M]} w_m \left(\Pr_{\hat{\tilde{\mathcal{D}}}_m}(A) - \Pr_{\hat{\mathcal{D}}_T}(A)\right)\right| \\
\leqslant&2 \sup_{A\in\mathcal{A}_{\mathcal{H}\Delta\mathcal{H}}} \sum_{n\in[N\setminus M]} w_n \left(\left|\Pr_{\hat{\mathcal{D}}_n}(A) - \Pr_{\hat{\mathcal{D}}_T}(A)\right|\right) + \sum_{m\in[M]} w_m \left(\left|\Pr_{\hat{\tilde{\mathcal{D}}}_m}(A) - \Pr_{\hat{\mathcal{D}}_T}(A)\right|\right) \\
\leqslant&2 \left(\sum_{n\in[N\setminus M]} w_n \sup_{A\in\mathcal{A}_{\mathcal{H}\Delta\mathcal{H}}} \left(\left|\Pr_{\hat{\mathcal{D}}_n}(A) - \Pr_{\hat{\mathcal{D}}_T}(A)\right|\right) + \sum_{m\in[M]} w_m \sup_{A\in\mathcal{A}_{\mathcal{H}\Delta\mathcal{H}}} \left(\left|\Pr_{\hat{\tilde{\mathcal{D}}}_m}(A) - \Pr_{\hat{\mathcal{D}}_T}(A)\right|\right)\right) \\
=& \sum_{n\in[N\setminus M]} w_n \hat{d}_{\mathcal{H}\Delta\mathcal{H}}\left(\hat{\mathcal{D}}_n, \hat{D}_T\right) + \sum_{m\in[M]} w_m \hat{d}_{\mathcal{H}\Delta\mathcal{H}}\left(\hat{\tilde{\mathcal{D}}}_m, \hat{D}_T\right) \\
\leqslant& \sum_{n\in[N]} w_n \hat{d}_{\mathcal{H}\Delta\mathcal{H}}\left(\hat{\mathcal{D}}_n, \hat{D}_T\right) + \sum_{m\in[M]} w_m \beta
\end{aligned} \tag{C.1}$$

Let λ_w denote the optimal risk on $\mathcal{D}_{\bar{S}}$ and $\mathcal{D}_T$, and let λ_n denote the optimal risk on $\mathcal{D}_n$ and $\mathcal{D}_T$. With the triangle inequality property, we can derive $\lambda_w \leq \sum_{n\in[N]} w_n\lambda_n$. Following the domain adaptation theory [Blitzer et al. 2007; Peng et al. 2020], we now establish a generalization bound under the presence of fairness attacks. Specifically, with probability at least $1-\delta$ over the choice of samples, the following generalization bound holds for all $h \in \mathcal{H}$:

$$
\begin{aligned}
&\epsilon_T\left(h_T\right) \\
\leq &\hat{\epsilon}_{\bar{S}}\left(h_T\right)+\frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}\left(\hat{\mathcal{D}}_{\bar{S}},\hat{\mathcal{D}}_T\right)+4\sqrt{\frac{2\tau\log(2N\sigma)+\log(4/\delta)}{N\sigma}}+\lambda_w \\
= &\hat{\epsilon}_{\bar{S}}\left(\sum_{n\in[N]} w_n h_n\right)+\frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}\left(\hat{\mathcal{D}}_{\hat{S}},\hat{\mathcal{D}}_T\right)+4\sqrt{\frac{2\tau\log(2N\sigma)+\log(4/\delta)}{N\sigma}}+\lambda_w \\
\leq &\hat{\epsilon}_{\bar{S}}\left(\sum_{n\in[N]} w_n h_n\right)+\frac{1}{2}\sum_{n\in[N]} w_n\hat{d}_{\mathcal{H}\Delta\mathcal{H}}\left(\hat{\mathcal{D}}_n,\hat{\mathcal{D}}_T\right)+\frac{1}{2}\sum_{m\in[M]} w_m\beta \\
&+\sum_{n\in[N]} w_n\lambda_n+4\sqrt{\frac{2\tau\log(2N\sigma)+\log(4/\delta)}{N\sigma}} \\
= &\underbrace{\hat{\epsilon}_{\bar{S}}\left(\sum_{n\in[N]} w_n h_n\right)}_{\text{error on source}}+\sum_{n\in[N]} w_n(\frac{1}{2}\underbrace{\hat{d}_{\mathcal{H}}\left(\hat{\mathcal{D}}_n,\hat{\mathcal{D}}_T\right)}_{(\mathcal{D}_n,\mathcal{D}_T)\text{ divergence}}+\lambda_n)+\sum_{m\in[M]} w_m(\frac{1}{2}\underbrace{\beta}_{\text{poisoned dataset variance}}) \\
&+\underbrace{4\sqrt{\frac{2\tau\log(2N\sigma)+\log(4/\delta)}{N\sigma}}}_{\text{VC-Dimension Constraint}}
\end{aligned}
\tag{C.2}
$$

Compared to Theorem 2 (weighted error bound for federated domain adaptation theorem) in [Peng et al. 2020], the proposed fairness attacks increase the upper bound of the target risk $\epsilon_T(h_T)$ by an additional term $\sum_{m\in[M]} w_m\left(\frac{1}{2}\beta\right)$, reflecting the bounded poisoning impact introduced by adversarial clients. This completes the derivation of Proposition 1.