Semi-Supervised Federated Adaptive Label Learning with Disambiguation Prototype

LAICHENG ZHONG

Master of Philosophy

THE UNIVERSITY OF SYDNEY

Supervisor: Dong Yuan
Associate Supervisor: Wei Bao

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

School of Electrical and Computer Engineering
Faculty of Engineering
The University of Sydney
Australia

15 March 2024

Abstract

Traditional federated learning depends on having data that is categorized or labeled for its training processes. However, obtaining such specific labeled data can be a complex task due to concerns about privacy, the expensive nature of labeling, or a shortage of specialized knowledge for categorizing data in certain fields. To overcome these obstacles, researchers have introduced Federated Semi-Supervised Learning (FSSL). This approach merges a limited quantity of categorized data with a vast amount of uncategorized data, thereby improving the effectiveness of models while maintaining the privacy and security of the data. Despite its advantages, FSSL encounters challenges such as gradual progress in learning and diminished precision, especially in situations where the distribution of data categories is uneven.

To mitigate these problems, we have introduced an innovative framework named Federated Adaptive Label Learning (FedALL). FedALL integrates strategies from transfer learning, partial label learning, and prototype learning. It constructs an Adaptive Label List through transfer learning and devises a Label Disambiguation Prototype via representation learning. These strategies significantly reduce the adverse impacts caused by incorrect one-hot-label predictions for unlabeled data on the client's model. Additionally, incorporating pre-trained weights substantially accelerates the convergence speed of FedALL.

In the experimental section of our paper, we compared FedALL against other baseline models in both Independent and Identically Distributed (IID) and Non-Independent and Identically Distributed (Non-IID) settings across three distinct datasets. The results demonstrate that FedALL outperforms all baselines in all scenarios, especially in Non-IID environments. Moreover, we utilized the Banach Fixed Point Theorem to prove the convergence of the Label Disambiguation Prototype. This solid theoretical foundation enhances the robustness of our model and its practical applicability.

Acknowledgements

During the course of compiling this thesis, I have been fortunate to receive invaluable assistance and encouragement from numerous individuals. I would like to take this opportunity to express my sincere appreciation to all those who have played a positive role in shaping my academic journey.

Foremost, I wish to extend my deepest gratitude to my supervisor, Professor Dong Yuan. Professor Yuan not only extended me the opportunity to join his esteemed research group but also provided unwavering guidance and support throughout the entirety of my research endeavors. His profound wisdom and mentorship served as a beacon of light during moments of adversity, profoundly influencing my personal and academic growth.

Special thanks also go to Nan Yang and Dr. Charles. Their constructive comments on my choice of topic and subsequent research have been crucial in completing this dissertation. Their insights and expertise have been key to shaping and deepening my research direction.

In addition, I would like to thank my friends, particularly Yuning Zhang, Xuanyu Cheng, Yongkun Deng, and Yanli Li. The days spent working and learning with them were filled with joy and inspiration. Their camaraderie, collaborative spirit, and mutual support in both academic and personal aspects greatly alleviated the pressures of graduate studies.

Finally, I must express my deepest appreciation to my parents. Their boundless love, coupled with their unwavering emotional and financial backing, enabled me to wholeheartedly dedicate myself to my academic endeavors. Without their steadfast support, embarking on this journey would have been an insurmountable challenge.

Once again, I thank everyone who has provided help and support. The completion of this dissertation could not have been achieved without the contribution of each of them.

Statement of Originality

This is to certify that, to the best of my knowledge, the content of this thesis is my own work. This thesis has not been submitted for any degree or other purpose.

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

Laicheng Zhong Date 15 March 2024

Authorship Attribution Statement

The content of this thesis is under reviewed in IEEE Transactions on Neural Networks and Learning Systems. I designed the study, mathematically proved the theories, and performed the experiments.

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 Laicheng Zhong Date 15 March 2024

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

Supervisor Name DONG YUAN Date 15 March 2024

List of Publications

Yanli Li, **Laicheng Zhong**, Dong Yuan, Huaming Chen and Wei Bao. "ICB FL: Implicit Class Balancing Towards Fairness in Federated Learning". Accepted by Proceedings of the 2023 Australasian Computer Science Week, 2023: 135-142.

Nan Yang, **Laicheng Zhong**, Fan Huang, Wei Bao and Dong Yuan. "Random Padding Data Augmentation". Accepted by Australasian Conference on Data Science and Machine Learning, Springer Nature Singapore, 2023: 3-18.

Laicheng Zhong, Nan Yang, Charles Liu, Xuanyu Chen, Yukun Deng, Wei Bao, Dong Yuan. "FedALL: Semi-Supervised Federated Adaptive Label Learning with Disambiguation Prototype". Under review at the IEEE Transactions on Neural Networks and Learning Systems.

Contents

Abstract ii

Acknowledgements iii

Statement of Originality iv

Authorship Attribution Statement v

List of Publications vi

Contents vii

List of Figures x

Chapter 1 Introduction 1

1.1 Background and Motivation 1

1.2 Research Questions and Objectives 5

1.2.1 Research Questions 5

1.2.2 Research Objectives 5

1.3 Contributions 6

1.4 Outline 7

Chapter 2 Literature Review 9

2.1 Transfer Learning 9

2.1.1 Transductive Transfer Learning 10

2.1.2 Inductive Transfer Learning 13

2.1.3 Cross-Modality Transfer Learning 15

2.1.4 Unsupervised Transfer Learning 19

2.1.5 Negative Transfer Learning 19

2.2 Prototype Learning 21

2.2.1 K-Nearest-Neighbor (KNN) 21
2.2.2 Learning Vector Quantization (LVQ) 22
2.3 Semi-Supervised Learning 25
2.3.1 Consistency Regularization 26
2.3.2 Entropy Minimization 30
2.3.3 Holistic Methods 31
2.4 Federated Learning 33
2.4.1 Privacy Protection 33
2.4.2 Aggregation Strategies 37
2.4.3 Clients Selection Methods 40
2.5 Federated Semi-Supervised Learning 42
2.5.1 The Label-at-Server Setting 42
2.5.2 The Label-at-Client Setting 45

Chapter 3 Architecture, Training Process, and Theoretical Foundations 46
3.1 Framework Design 46
3.1.1 Overview 46
3.1.2 Server Side Design 48
3.1.3 Client Side Design 49
3.2 Learning Methods 49
3.2.1 Knowledge Transfer 49
3.2.2 Label Disambiguation Prototype 51
3.2.3 Adaptive Label List 52
3.2.4 Contrastive Learning 54
3.3 Theoretical Analysis 55
3.3.1 Learning Process and Optimization 55
3.3.2 Convergent Learning 58
3.4 Algorithms Design 65
3.4.1 Server Side Implementation 65
3.4.2 Client Side Implementation 65

Chapter 4 Experiment and Evaluation **68**

4.1 Experiment Setup 68

4.1.1 Datasets 68

4.1.2 Baselines 69

4.1.3 Hardware Configuration 70

4.2 Experiment Methodology and Result Analysis 70

4.2.1 Parameter Setting and Pre-Training 70

4.2.2 Experiment Result and Analysis 72

4.3 Impact of Different Parameters 73

4.3.1 Client Selection Number 74

4.3.2 Local Epoch 76

4.3.3 Total Client Number 78

4.3.4 Non-IID Rate 81

4.4 Ablation Studies 82

4.4.1 Components Analysis 82

4.4.2 Different Number of Chosen Labels in Adaptive Label List 83

Chapter 5 Conclusion **84**

Bibliography **86**

List of Figures

1.1 In contrast to the traditional One-Hot predicted label approach, our innovative Adaptive Label List prioritizes maximizing the probability of incorporating the ground-truth label. Illustrated in Figure 1.1, in instances where the adaptive label list initially lacks the ground-truth label, FedALL employs a label correction mechanism to rectify this omission, thereby enhancing the likelihood of including the ground-truth label in the list. 4
3.1 The overall framework of FedALL. 47
4.1 Sensitivity to different client selection numbers on CIFAR-10 IID. 74
4.2 Sensitivity to different client selection numbers on CIFAR-10 Non-IID. 75
4.3 Sensitivity to different client selection numbers on CIFAR-100 IID. 75
4.4 Sensitivity to different client selection numbers on CIFAR-100 Non-IID. 76
4.5 Sensitivity to different local epoch on CIFAR-10 IID. 77
4.6 Sensitivity to different local epochs on CIFAR-10 Non-IID. 77
4.7 Sensitivity to different local epoch on CIFAR-100 IID. 78
4.8 Sensitivity to different local epochs on CIFAR-100 Non-IID. 78
4.9 Sensitivity to different total client numbers on CIFAR-10 IID. 79
4.10 Sensitivity to different total client numbers on CIFAR-10 Non-IID. 79
4.11 Sensitivity to different total client numbers on CIFAR-100 IID. 80
4.12 Sensitivity to different total client numbers on CIFAR-100 Non-IID. 80
4.13 Sensitivity to different Non-IID rates on CIFAR-10. 81
4.14 Sensitivity to different Non-IID rate on CIFAR-100. 82

CHAPTER 1

Introduction

This chapter is crafted to meticulously establish the contextual and theoretical framework of the research, highlighting its scope and significance within the academic sphere. It serves as a vital gateway, detailing the study's objectives, theoretical foundations, and potential contributions to existing knowledge. This section not only addresses the motivations and gaps in the current literature, but also provides a structured overview, outlining the content and sequence of subsequent chapters, and giving brief contributions to this thesis.

1.1 Background and Motivation

The exponential growth of edge devices has ushered in an era where data is not only abundant but also immensely varied and rapidly generated. This surge in data creation at the edge of networks is driven by a wide array of devices, including, but not limited to, smart home appliances, wearable health monitors, industrial IoT sensors, and autonomous vehicles. These devices continuously collect and transmit data, which, while beneficial for real-time decision making and personalized services, also amplifies the challenges associated with data management and analytics. Deep learning models, renowned for their ability to extract meaningful patterns from large and complex datasets, are at the forefront of harnessing the potential of this data deluge. However, the process of training these models often requires access to large amounts of raw data, raising significant privacy and security concerns. The sensitive nature of the data collected by edge devices, from personal health records to location

data, requires stringent measures to ensure its confidentiality and integrity. Furthermore, the decentralized nature of data generation in edge computing complicates traditional approaches to data privacy and security. Unlike centralized data centers, where control and security protocols are easier to enforce, edge devices are often dispersed, with varying levels of security, and often under different administrative domains. This makes the implementation of uniform security policies and practices challenging.

In the rapidly evolving field of machine learning, dealing with vast, diverse, and often partially labeled datasets presents a significant challenge. Federated Semi-Supervised Learning (FSSL) emerges as a groundbreaking solution to this dilemma. Developed by pioneering researchers in the field, FSSL operates under a unique framework that effectively combines the strengths of federated learning and semi-supervised learning techniques. At the core of FSSL is a novel approach to data utilization and management. It typically involves a central server that holds a relatively small portion of labeled data, essential for guiding the learning process. This set-up is crucial because labeled data, often scarce and expensive to obtain, play a vital role in training accurate models. The innovation in FSSL, however, lies in its integration with client devices, each of which holds substantial amounts of unlabeled data. This structure is highlighted in the seminal work [50], which emphasizes the practicality and efficiency of this distribution of data resources. The primary strength of FSSL is its ability to handle the inherent complexity of dealing with large-scale unlabeled datasets without compromising learning quality. This capability is a significant leap forward, addressing one of the fundamental limitations of traditional learning frameworks. By leveraging unlabeled data effectively, FSSL improves the overall performance of machine learning models, making them more robust, accurate, and reliable. Furthermore, FSSL's design allows for greater scalability and adaptability across various domains and use cases. This flexibility is particularly valuable in fields where data labeling is impractical or infeasible, such as medical imaging, natural language processing, and real-time predictive analysis.

Current strategies in Federated Semi-Supervised Learning (FSSL) primarily leverage two methodologies. The first, pseudo-labeling, is represented by approaches like FedMatch

[50] and Fed-fixmatch. Rooted in centralized semi-supervised learning techniques (e.g., MixMatch [7], FixMatch [92], and FlexMatch [125]), these strategies enable the model to autonomously generate pseudo-labels for the unlabeled data residing on client devices. High-confidence pseudo-labels are then treated as accurate for further model training. This approach, though innovative, hinges critically on the precision of these pseudo-labels. If the pseudo-labels are substantially misaligned with the true labels, it can drastically impair the model's performance, potentially leading to non-convergence issues. In contrast, several FSSL models, such as FedU [29], FedEMA [133], and Orchestra [71], pivot toward contrastive learning rather than relying on pseudo-labeling. These methods incorporate a consistency regularization component. This component is vital as it ensures that the model predictions remain stable across different iterations of the same input data. This strategy aims to combat the challenge of heterogeneity in data and distribution changes between clients in a federated learning setup. However, despite their effectiveness in maintaining prediction consistency, these contrastive learning-based models often encounter hurdles, such as slower convergence rates and long training periods. This extended training time can be a significant drawback, especially in federated learning environments, where computational resources and client availability can be sporadic and limited. Furthermore, both methodologies must contend with the inherent challenges of federated learning, such as data privacy concerns, communication overheads, and the non-IID nature of distributed data. Balancing these factors with the need for efficient and accurate learning models continues to be a focal point of FSSL research. In this thesis, we introduce our Federated Adaptive Label Learning (FedALL) framework. This innovative framework seamlessly combines transfer learning, partial label learning, and prototype learning. By implementing these three learning strategies, FedALL establishes a harmonized collaborative framework. Specifically, a transferred model is created on the server using labeled data and a pre-trained weight. This model facilitates the construction of an accurate classifier and projection head on both the client and the server. As depicted in Figure 1, the classifier within the client's model discerns potential ground-truths for the unlabeled datasets. Eschewing conventional one-hot predicted labels, our approach leans toward partial label learning, crafting an adaptive label list teeming with candidate labels informed by classifier predictions. Notably, our strategy diverges from traditional partial

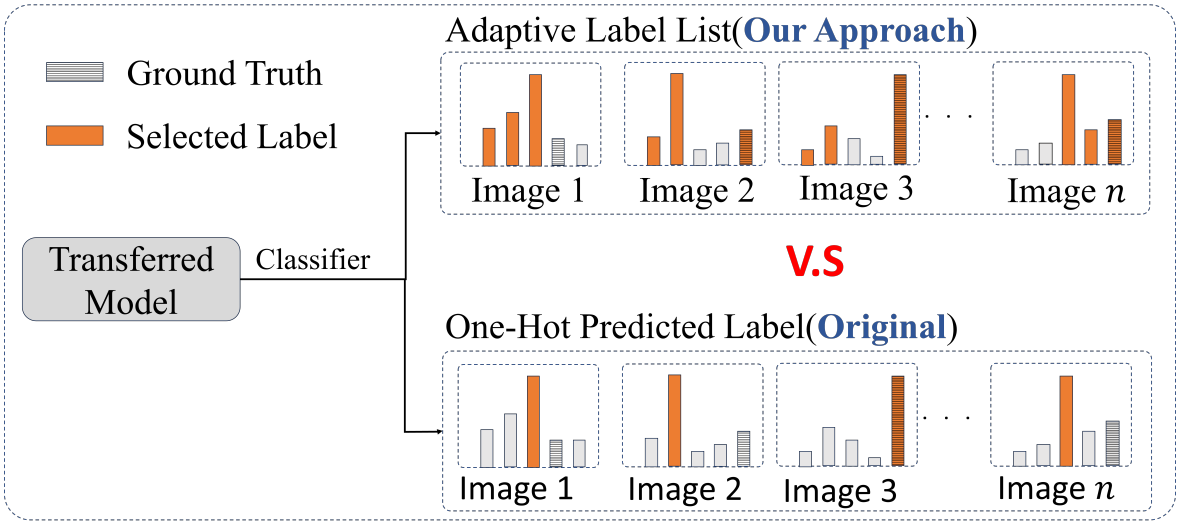

FIGURE 1.1. In contrast to the traditional One-Hot predicted label approach, our innovative Adaptive Label List prioritizes maximizing the probability of incorporating the ground-truth label. Illustrated in Figure 1.1, in instances where the adaptive label list initially lacks the ground-truth label, FedALL employs a label correction mechanism to rectify this omission, thereby enhancing the likelihood of including the ground-truth label in the list.

label methods, which invariably include the ground-truth within candidate labels. Importantly, the use of the partial-label approach significantly mitigates the adverse effects that incorrect pseudo-labels might impose on the model, as we mentioned earlier. To select the potential ground-truth in the label list, the model's projector head formulates a prototype for each client, enabling them to effectively pinpoint the ground-truth from the adaptive label list through prototype learning. Furthermore, leveraging the pre-trained weights accelerates the model's convergence rate.

In the remainder of this chapter, Section 1.3 summarizes the thesis contributions and Section 1.4 presents the thesis outline.

1.2 Research Questions and Objectives

1.2.1 Research Questions

- How can one hot pseudo-label and representation learning methods be effectively integrated within a federated learning framework to improve model accuracy, particularly in semi-supervised scenarios with Non-IID data?
- Can theoretical analysis validate the convergence of a designed federated semi-supervised learning system, ensuring its reliability and effectiveness in practical applications?
- How does the proposed framework perform compared to existing baseline models across different datasets and under various parameter settings, especially in terms of handling non-IID data effectively?

1.2.2 Research Objectives

- Integrate Pseudo-label and Representation Learning Methods: Develop strategies to effectively combine pseudo-label techniques with representation learning methods within the FedALL framework to enhance model accuracy and learning efficiency in semi-supervised federated settings.
- Theoretical Analysis of Convergence: Employ theoretical analysis to demonstrate that the designed federated semi-supervised system can achieve convergence, ensuring the reliability and stability of the proposed framework.
- Empirical Validation through Experiments: Test the proposed framework under various datasets and parameter settings, and compare its performance with other baseline models to empirically validate its effectiveness and superiority in addressing the challenges of federated semi-supervised learning.

1.3 Contributions

The innovative framework we have introduced, named "FedALL," represents a groundbreaking approach in the domain of federated semi-supervised learning (FSSL). This framework is unique in its integration of three key methodologies: transfer learning, representation learning, and partial label learning. Such an integration is unprecedented in FSSL and marks a significant advancement in the field. Furthermore, within the context of this framework, we have developed two novel concepts that are particularly tailored for FSSL. The first is the 'label disambiguation prototype', a tool designed to clarify and refine the understanding of labels in a semi-supervised learning environment. The second concept is the 'adaptive label list', which dynamically adjusts label classifications to better fit the evolving learning landscape. Both of these concepts enhance the effectiveness of FedALL by addressing specific challenges inherent in federated semi-supervised learning, thereby setting a new standard for research and application in this area.

In our research, we delve into the theoretical aspects by leveraging the Banach Fixed Point Theorem, a fundamental concept in mathematical analysis. This theorem is pivotal in demonstrating that if the perturbations occurring in the representation learning process of our model are smaller than those in the prototype, we can guarantee the prototype's convergence. This aspect is crucial to ensure the stability and reliability of the learning process. In addition, we meticulously quantify the rate at which the prototype changes during communication exchanges between the client and the server in a federated learning setup. By doing so, we not only highlight but also reinforce the feasibility of achieving convergence in federated learning scenarios. This in-depth mathematical analysis and validation of our model significantly enhance its robustness. It lays a solid theoretical foundation that is not only academically sound but also instrumental in the practical deployment and implementation of our proposed framework. This synergy between theoretical rigor and practical applicability forms the cornerstone of our work, promising advances in the field of machine learning and its applications.

This thesis presents an in-depth analysis of a new framework named FedALL, which is rigorously evaluated across three benchmark datasets, bench-marking its performance against state-of-the-art (SOTA) baseline models in a variety of scenario settings. The standout result of this evaluation is the notable improvement FedALL shows over the best-performing baseline model, with an impressive increase of up to 3.92% in accuracy on the renowned ImageNet dataset. This performance enhancement is particularly noteworthy given the challenges posed by non-independent and Identically Distributed (non-IID) settings in machine learning. The experiments detailed in the study further highlight FedALL's robustness, showing that it delivers results in non-IID environments that closely mirror those obtained in Idealized Independent and Identically Distributed (IID) settings, thereby establishing its efficacy and adaptability in diverse data scenarios. This suggests that FedALL not only improves existing models in terms of accuracy but also offers reliable performance across varying data distributions.

1.4 Outline

- Chapter 2 presents an in-depth analysis of the current representative methodologies and algorithms, encompassing a comprehensive review of the foundational theories, recent developments, and prevalent techniques within each domain. Furthermore, it delves into the examination of the challenges, constraints, and potential opportunities inherent in each learning paradigm, while also pinpointing the primary avenues for future research and exploration in these fields.
- Chapter 3 presents the Federated Adaptive Label Learning (FedALL) framework, a novel approach that synergistically integrates transfer learning, partial label learning, and prototype learning. FedALL orchestrates these different learning strategies into a cohesive and collaborative framework. It also explores theoretical dimensions, utilizing the Banach Fixed Point Theorem, a cornerstone in the field of mathematical analysis. This theorem is critical for establishing that should the perturbations

observed in the representation learning phase of our algorithm be lesser than those in the prototype, it is possible to ensure the convergence of the prototype.

- Chapter 4 offers an in-depth analysis of the newly developed FedALL framework. This chapter outlines the experimental setup, including details about the datasets and baseline models used. It provides a comprehensive evaluation of the performance of FedALL compared to baseline models across diverse datasets, considering scenarios of both independent and identically distributed (IID) and non-independent and identically distributed (Non-IID) data distributions. Additionally, the analysis incorporates variations in label rates to offer a thorough understanding of model performance under different conditions. In addition, the chapter explores how various parameter settings affect performance and includes visual representations to highlight the performance disparities between FedALL and the baseline models. Furthermore, an ablation study is conducted to evaluate the impact of different components within the FedALL framework, as well as the influence of the number of labels in the adaptive label list on overall performance.

CHAPTER 2

Literature Review

This chapter includes a comprehensive related work related to FedALL. Section 2.1, we talk about the content of transfer learning that includes transductive transfer learning, inductive transfer learning, and cross-modal transfer learning. Section 2.2 introduces the learning strategies of the server prototype, including K-Nearst-Neibhor and quantization of the learning vector. Section 2.3 introduces the Semi-Supervised Learning methods, which contain Consistency Regularization, Entropy Minimization, and Holistic Methods. Section2.5 talks about Semi-Supervised Learning under the federated learning setting, including the data labeled on the server setting and the data labeled on the client settingg.

2.1 Transfer Learning

Transfer Learning (TL) has emerged as a powerful technique in the realm of machine learning, addressing challenges that traditional machine learning often struggles with. These challenges include the scarcity of labeled data, computational constraints, and the issue of a distribution mismatch between training and testing datasets [79]. TL has played a pivotal role in attaining cutting-edge outcomes across a multitude of fields, including but not limited to image analysis, voice interpretation, and the understanding of human language (NLP) [112, 132, 27]. The essence of TL lies in its ability to leverage knowledge from one domain (source) and apply it to another potentially different domain (target). This is particularly beneficial when the target domain has limited labeled data. The chapter categorizes TL into four main types:

transductive learning, inductive learning, unsupervised learning, and negative learning. Each of these categories can be further defined based on four learning paradigms: learning on instances, learning on features, learning on parameters, and learning on relations [79, 94, 5]. Recent advances in TL have expanded its applicability, addressing issues such as domain adaptation, distant domain transfer, and learning with transmodality transfer [49, 60, 128]. As the field continues to evolve, it offers promising solutions to some of the most pressing challenges in machine learning, making it a pivotal area of research and application [105].

2.1.1 Transductive Transfer Learning

In recent years, Transductive Transfer Learning (Transductive TL) has solidified its position as a cornerstone in the expansive domain of transfer learning [79]. This paradigm is characterized by the consistency of tasks across both the source and target domains, although the domains themselves may exhibit distinct differences. A prominent feature of Transductive TL is the exclusive availability of labeled data in the source domain, juxtaposed with the target domain, which is predominantly populated with unlabeled data [79, 112]. Such a configuration has catalyzed the evolution of a variety of methodologies tailored to address this unique challenge. Among these, two primary learning strategies have been identified as paramount: learning in instances and learning in features [79, 132]. Delving deeper into the realm of transductive learning, domain adaptation emerges as a quintessential application. It encapsulates the effort to fine-tune models, originally trained in a source domain, to exhibit robust performance in a different, albeit related, target domain [80]. The significance of domain adaptation, a subfield of transductive TL, cannot be understated. Specifically, it addresses scenarios where there is an abundance of labeled source data juxtaposed with unlabeled target data for the training process. Contemporary domain adaptation techniques aim to minimize the discrepancy between the marginal distribution distance or the conditional distribution distance in two primary manners: through symmetrical training and asymmetrical training approaches. Symmetrical training, exemplified by the utilization of two models with matching architectures for both the source and target domains, has predominantly been employed in feature-based algorithms [25].

2.1.1.1 Learning on Instances

In the domain of Transductive Transfer Learning (Transductive TL), the methodology of learning in these cases stands out as a cornerstone. At its core, this approach is characterized by the transfer of knowledge from a well-defined source domain to a potentially disparate target domain. This intricate transfer process is predominantly facilitated through two primary mechanisms: the re-weighting and resampling of source instances.

Two fundamental assumptions underpin this approach.

- First, there exists a substantial subset of training instances within the source domain that are inherently aligned with the characteristics of the target domain. This alignment makes these instances invaluable assets, making them reusable for tasks specific to the target domain [112].
- Second, conditional distributions, which define the probabilistic relationships within both the source and target domains, exhibit a high degree of congruence. This congruence is crucial as it ensures that the knowledge transferred remains meaningful and effective in the new domain [132].

Despite the potential advantages of this approach, it is not devoid of challenges. One of the most pressing concerns is the discernment of source data that is optimally suited for reuse in the training processes of the target model. The importance of judiciously selecting samples that can significantly benefit the target domain task cannot be overstated. To address this challenge, several methodologies have been proposed. For example, some approaches have used the power of AdaBoost, a renowned boost algorithm, to refine the selection process [33]. In parallel, there has been a surge in innovative methodologies, for instance, re-weighting and selection. Many of these novel approaches draw inspiration from advanced concepts such as Positive-Unlabeled Learning (PU) and specific probability metrics for the target domain [32, 69].

2.1.1.2 Learning on Features

In the intricate landscape of transductive transfer learning (Transductive TL), feature-based methodologies have emerged as a focus of research, often receiving more attention than their instance-based counterparts. This predilection towards feature-based methods can be attributed to several factors. In particular, the inherent limitations associated with the learning mechanisms of instance-based methods have made feature-based strategies more attractive to researchers [79]. Feature-based methods in transductive TL predominantly revolve around the concept of domain adaptation. The overarching objective of these methods is to bridge the gap between the features of the source and target domains. By doing so, they aim to reduce the divergence between the feature distributions of the two domains, ensuring that models trained in the source domain can be effectively deployed in the target domain with minimal performance loss. This is of paramount importance, especially when considering real-world applications where the distribution of data could change over time or in different contexts [80]. A notable characteristic of feature-based methods is their ability to operate in scenarios where only a sufficient amount of labeled source data are available, while the target domain consists primarily of unlabeled data [80]. This places domain adaptation methods within the broader cluster of semi-supervised learning algorithms. Current domain adaptation algorithms endeavor to bridge the gap between source and target domains by minimizing the marginal distribution distance or the conditional distribution distance. In this scenario, two main approaches are recognized: balanced training and unbalanced training. Balanced training, as an instance, employs two identical models for both the origin and destination domains, frequently utilized in algorithms focused on characteristics. The advantages of this approach include the ease of training and rapid convergence [70]. Feature-based methods in transductive TL offer a promising avenue to address the challenges posed by domain discrepancies. As the field continues to evolve, these methods are expected to play a pivotal role in ensuring the robustness and adaptability of machine learning models across various domains.

Inductive TL, with its emphasis on instances and features, offers a robust framework for tackling domain adaptation problems, especially when labeled data in the target domain are scarce.

2.1.2 Inductive Transfer Learning

In recent years, Inductive Transfer Learning (Inductive TL) has firmly established itself as an indispensable paradigm within the expansive domain of Transfer Learning, drawing considerable attention from researchers and practitioners alike [112]. This burgeoning interest can be attributed to Inductive TL's methodical and structured approach, which facilitates the extraction and assimilation of knowledge from a richly annotated source domain. Once this knowledge is harnessed, it is meticulously applied to a distinct target domain. This becomes particularly important when considering scenarios in which tasks that span these domains, although not identical, exhibit a plethora of shared attributes or underlying commonalities [80]. The core ethos of Inductive TL transcends the rudimentary act of knowledge transfer. Instead, it places a profound emphasis on the act of knowledge generalization. This nuanced distinction ensures that the knowledge and insights derived from the source domain undergo a transformative process. Rather than simply replicating or transplanting knowledge, Inductive TL endeavors to adapt, refine and tailor this knowledge, ensuring its alignment with the intrinsic characteristics and nuances of the target task [132, 10]. This approach not only increases the efficacy of the transferred knowledge, but also ensures its relevance, resonance, and applicability in diverse and often challenging target domains [81].

2.1.2.1 Learning on Instances

Within the ambit of Inductive Transfer Learning (Inductive TL), the paradigm of "Learning on instances" has emerged as a critical and distinct methodology, underscoring the paramount importance of individual data points, often referred to as instances, originating from the source domain. Unlike conventional transfer learning strategies that might advocate

for an indiscriminate transfer of all available data, this nuanced approach adopts a more discerning position. It embarks on a rigorous assessment of the relevance and applicability of each instance, especially when viewed through the lens of the specific requirements and characteristics of the target domain [112]. Central to this approach is the deployment of advanced weighting mechanisms, often rooted in mathematical and probabilistic frameworks, designed to gauge the alignment of each instance with the target domain [80]. Instances that resonate closely with the target domain's distribution and characteristics are given higher weights, effectively amplifying their in Federated Learninguence in the transfer process. On the contrary, instances that pose the risk of introducing anomalies, noise or any form of incongruence are judiciously moderated, lowered, or, in certain cases, completely excluded from the transfer process [70]. This meticulous and granular emphasis on individual instances becomes particularly salient in situations characterized by significant overlaps between the source and target distributions. In such contexts, a blanket transfer could lead to suboptimal results or even detrimental effects. However, by prioritizing and harnessing the most relevant and aligned samples, "Learning on instances" ensures that the knowledge transferred is both pertinent and beneficial, optimizing the performance of models in the target domain [132].

2.1.2.2 Learning on Features

In the intricate tapestry of Transfer Learning, while instances provide granular data points, it is the features that encapsulate the underlying structure and semantics of the data. Leaving beyond the immediate scope of instances, the "Learning-on-Features" paradigm delves deeply into these foundational constructs, seeking to unravel and harness the latent patterns and relationships they embody [132]. The overarching objective transcends the mere identification of analogous features in domains. Instead, it aspires to meticulously orchestrate a harmonized representation, a congruent feature space where attributes from both the source and target domains can meld and coalesce seamlessly, preserving their intrinsic characteristics while adapting to the nuances of the new domain [132]. Achieving such a congruence is not a trivial task. It requires the deployment of sophisticated methodologies and techniques. Dimensionality reduction, for example, emerges as a crucial tool, streamlining feature space

by retaining only the most prominent and representative attributes, thus mitigating the curse of dimensionality [72]. Currently, feature enhancement techniques are used to enrich the feature set, introducing synthesized or derived attributes that improve the representational capacity of the data [90]. Furthermore, in the contemporary era marked by the rise of deep learning, neural architectures, especially deep neural networks, have been increasingly exploited to carve out shared feature spaces, leveraging their prowess in automatic feature extraction and representation learning [61]. The culmination of these efforts is a robust and adaptive feature landscape, devoid of potential chasms or discontinuities that could impede the transfer process. By ensuring such a holistic and integrated feature representation, the "Learning on Features" approach not only facilitates the transfer of knowledge but also fortifies the foundation for superior generalization, enabling models to perform adeptly across a diverse array of domains, irrespective of their inherent complexities or disparities [95].

In conclusion, Inductive TL, with its dual focus on instances and features, presents a robust and nuanced framework, adept at navigating the intricate challenges of domain adaptation. As the realms of machine learning and artificial intelligence continue to evolve, the methodologies and insights offered by Inductive TL will undoubtedly play a pivotal role in shaping cross-domain solutions, especially in scenarios marked by a paucity of labeled data in the target domain [95].

2.1.3 Cross-Modality Transfer Learning

Cross-Modality Transfer Learning (CMTL) emerges as a key and intricate domain within the broader landscape of Transfer Learning. Classic transfer learning techniques primarily assume a concrete link between the source and target domains, whether through feature spaces or label spaces. In contrast, CMTL questions this traditional view, suggesting that the feature spaces of the source and target domains may have no overlap whatsoever. This could manifest itself in transitions as diverse as moving from textual data to visual imagery or from auditory signals to textual representations. The philosophical foundation of CMTL is deeply rooted in

our understanding of human cognition. Humans possess a remarkable ability to extrapolate and generalize knowledge in different modalities. For example, a person might read a detailed description of a musical piece and subsequently recognize it upon hearing, even if they had never encountered that piece before [79]. This cognitive Federated Learningexibility is what CMTL aspires to emulate in machine learning models.

One of the primary motivations behind CMTL is the inherent challenges posed by data acquisition in various domains. For example, in the realm of image classification, two primary challenges persist: 1) the relative scarcity and prohibitive cost of obtaining labeled image data, and 2) the inherent nature of image data features, which often encapsulate visual attributes rather than deeper semantic or conceptual nuances1. In such scenarios, the use of textual data, which may be more abundant and semantically rich, can provide a pathway to improve image classification models. A method that has attracted attention in this context is the Translated Learning via Risk Minimization (TLRisk) introduced by [26]. Although ambitious, this approach proposed an asymmetric architecture designed to bridge the semantic chasm between textual source domains and image-centric target domains. Ingeniously, it uses a language model combined with a nearest neighbor method to establish this bridge1. Another innovative approach in this domain employs a transition mechanism termed a "translator", which seeks to establish a conduit between text and images, thereby harnessing the power of transfer learning to exploit textual data for the enhancement of image classification.

2.1.3.1 CMTL With Supervised Target Data

Cross-Modality Transfer Learning, especially in the context of text-to-image (TTI) DDTL methods, presents a unique set of challenges and opportunities. The primary motivation behind these methods is the need to use a limited set of labeled image target data for improved classification performance. Image classification tasks are currently confronted with two significant challenges: Data Scarcity and Cost: Labeled image data are not only scarce but

also expensive to collect. This scarcity poses a significant hurdle to train robust models that can generalize well across diverse image datasets1.

Semantic Ambiguity of Image Features: Features extracted from image data often lack semantic meaning for class prediction. Unlike text features, which are inherently rich in semantic information, image features tend to represent visual attributes, making them less informative for predicting conceptual class labels1. Given these challenges, there has been a surge in methods that aim to bridge the gap between the text and image domains. Among these prominent approaches is Translated Learning via Risk Minimization (TLRisk), first presented by [26]. TLRisk introduces an innovative asymmetric framework that aims to translate features from a textual source domain into an image-based target domain. This technique cleverly utilizes a language model and the nearest-neighbor approach to forge a linkage between textual source information and visual target data. The core concept revolves around harnessing the deep semantic content inherent in textual data to improve the categorization accuracy of visual data.

Furthermore, to ensure a seamless feature transition between the two domains, a novel transition method, termed a "translator", has been proposed. This translator acts as a bridge, facilitating the transfer of knowledge from text to images. Taking advantage of transfer learning principles, this method effectively exploits text data, enhancing the classification performance of image datasets1. In essence, the realm of CMTL with Supervised Target Data is rapidly evolving, with innovative methods like TLRisk and the translator offering promising avenues to harness the power of text data for improved image classification. These methods not only address the inherent challenges of image classification but also pave the way for more integrated and semantically rich feature representations across modalities.

2.1.3.2 CMTL With Semi-Supervised Target Data

In the realm of cross-modulality transfer learning, the semi-supervised approach offers a unique blend of labeled and unlabeled target data to enhance classification outcomes. This approach stands in contrast to the purely supervised methods and provides a more Federated Learningexible framework to leverage the available data1. A particularly notable method in this domain is the Heterogeneous Transfer Learning for Image Classification (HTLIC) proposed by [131]. This method is meticulously crafted to support a target image classification task, especially when faced with the constraints of limited labeled data. The strength of HTLIC lies in its ability to extract and harness semantic knowledge from both unlabeled textual documents and unlabeled annotated images originating from an additional domain. This supplemental data, which can be collected at a relatively low cost, plays a vital role in enhancing the accuracy of classifying the target visual data.

The fundamental operation of HTLIC is centered on creating a connection between the unlabeled textual data from the source and the semi-supervised visual data in the target domain. This is achieved by leveraging auxiliary data imbued with related semantic information. The method uses a two-layer bipartite graph as its foundational structure. The upper echelon of this graph delineates the relationship between images and their associated tags, while the lower tier captures the interaction between these tags and the text documents1. This intricate graph-based representation serves a dual purpose: it not only bridges the feature space chasm between the source and target domains, but also facilitates the discovery of shared semantic information, thereby enriching the transfer learning process1. In essence, the semi-supervised approach in CMTL, as exemplified by methods like HTLIC, underscores the potential of blending labeled and unlabeled data. By judiciously leveraging auxiliary data and sophisticated graph-based representations, it offers a robust framework for knowledge transfer across disparate modalities.

2.1.4 Unsupervised Transfer Learning

Unsupervised Transfer Learning (UTL) is at the intersection of transfer learning and unsupervised learning, two prominent areas in machine learning [80]. While transfer learning seeks to utilize insights from one domain to improve the learning in another, related target domain as outlined by [112], unsupervised learning is dedicated to identifying patterns within unlabeled data, a concept detailed by [45]. UTL combines these frameworks with the goal of transferring expertise from a well-annotated source domain to an unannotated target domain. [132]. The motivation behind UTL is rooted in real-world scenarios where labeled data are scarce or expensive to obtain. For example, in domains such as medical imaging or natural language processing, obtaining labeled data can be time-consuming and costly [89]. UTL integrates these approaches, aiming to convey expertise from a domain with labeled data to another domain lacking labels [123]. This is particularly beneficial when the source and target domains share underlying structures or patterns, but not necessarily the same labels [81].

Several techniques for UTL have been proposed. A common approach involves learning representations from the source domain that can be effectively transferred to the target domain [6]. Deep learning models, especially auto-encoders and generative adversarial networks (GANs), have shown promise in this area [35]. These models can learn rich feature representations from the source domain, which can then be fine-tuned or adapted for tasks in the target domain using unlabeled data [102]. Unsupervised Transfer Learning offers a promising avenue for harnessing the power of unlabeled data by bridging the gap between the source and target domains [34]. As the field continues to evolve, it has the potential to revolutionize domains where labeled data are a limiting factor [14].

2.1.5 Negative Transfer Learning

Transfer Learning is renowned for its capacity to harness insights from one area and utilize them in another, possibly distinct area. This approach is particularly advantageous when the

target domain has limited labeled data. However, not all transfers result in better performance. In some cases, transfer can lead to a degradation in model performance, a phenomenon known as Negative Transfer Learning (NTL) [79].

Negative Transfer Learning occurs when the knowledge transferred from the source domain not only fails to benefit but actually hampers the learning process in the target domain. This can stem from several factors, including substantial disparities in data distributions between the two domains or the existence of erroneous labels within the source dataset. The challenge with NTL is that it can be difficult to predict a priori. A model that performs exceptionally well in one domain might introduce biases or inaccuracies when applied to a different domain. Addressing the NTL issue requires a careful examination of both the source and target domains. Techniques such as domain adaptation can be employed, which aims to minimize distributional differences between domains. Additionally, methods that weigh the relevance of source data samples to the target task can also be beneficial. By limiting potentially harmful source samples, the negative impact of transfer can be mitigated.

Although negative transfer is a common phenomenon, its description frequently lacks formality, missing out on a precise definition, detailed scrutiny, or structured handling.[132]. Numerous survey articles have discussed this issue in various TL disciplines [80, 112, 132]. Recognizing its impact on real-world applications, several studies have focused on addressing and understanding the nuances of negative transfer [110, 79, 89]. While transfer learning offers a promising avenue for enhancing model performance, especially in data-scarce scenarios, it is crucial to be aware of the potential pitfalls like Negative Transfer Learning. By understanding and addressing the causes of NTL, researchers and practitioners can harness the full power of TL while avoiding its potential drawbacks.

2.2 Prototype Learning

A prototype is defined as the average vector that represents instances that belong to the same class [91]. This concept, due to its exemplar-driven characteristics and its straightforward inductive bias, has shown significant potential in various tasks. In the realm of supervised classification, the method involves labeling test images by measuring their distance from the prototypes of each respective class. This approach is considered to offer enhanced robustness and stability, especially in scenarios involving few-shot [75, 91, 99, 117] and zero-shot classifications [51]. Furthermore, there has been growing interest in the use of prototypes in tasks such as semantic segmentation [63, 108, 129], unsupervised learning [41, 64, 113, 120, 121], among others. In the context of federated learning, prototypes can provide a diverse range of abstract knowledge while maintaining adherence to privacy guidelines. A limited number of studies have explored the integration of prototypes to address the challenges of data heterogeneity in federated learning.

2.2.1 K-Nearest-Neighbor (KNN)

The K-Nearest-Neighbor (KNN) algorithm is a quintessential example of prototype-based learning, where decisions are made based on the proximity of an instance to a set of stored exemplars or prototypes. At its core, K-NN operates by identifying the 'k' training samples that are closest to a given test sample and then making a decision based on the majority class among these neighbors [24]. Prototype-based learning, such as K-NN, is inherently nonparametric. This means that it does not make strong assumptions about the form or parameters of the underlying data distribution. Instead, it directly uses the training data during the decision-making process. This characteristic allows K-NN to be particularly Federated Learningexible in modeling complex decision boundaries, as the algorithm can adapt to the local structure of the data [44].

The distance metric, often the Euclidean distance, plays a pivotal role in the K-NN algorithm. The choice of distance metric can significantly in Federated Learninguence the performance of the algorithm and, in some cases, domain-specific metrics may be more appropriate [30]. Moreover, the value of 'k', the number of neighbors considered, also has a profound impact on the algorithm's behavior. A lower 'k' value might render the algorithm vulnerable to noise, whereas a higher 'k' can lead to more generalized decision boundaries, which may ignore subtle nuances in the data. [47]. Although K-NN is conceptually simple and intuitive, it does have its challenges. One of the primary concerns is its computational cost during testing. Since K-NN requires comparing a test instance to every instance in the training set, it can be computationally intensive for large datasets. This has led to the development of various approximation and indexing techniques to speed up the nearest-neighbor search [48].

In the context of prototype learning, K-NN can be viewed as an exemplar-driven approach in which every training instance serves as a potential prototype. This differs from other prototype-oriented approaches, which may either choose or derive a selection of characteristic prototypes from the training dataset [11]. The strength of K-NN lies in its ability to make decisions based on local data structures, making it a valuable tool in the arsenal of machine learning practitioners.

2.2.2 Learning Vector Quantization (LVQ)

Learning Vector Quantization (LVQ) stands out as a prototype-driven supervised learning method, specifically tailored for tasks involving classification. Unlike the k-nearest-neighbor (K-NN) method, which treats every training instance as a potential prototype, LVQ explicitly learns a set of prototypes for each class in the dataset. The learning process involves iterative adjustment of these prototypes to better represent the distribution of training data. During classification, an input vector is assigned to the class of its nearest prototype based on a chosen distance metric, commonly the Euclidean distance. One of the primary advantages of LVQ is its ability to provide a compact representation of the data, as it reduces the dataset to a

set of representative prototypes. This not only aids in visualization and interpretation, but also reduces computational costs during the classification phase. However, the performance of LVQ is contingent upon the proper initialization of the prototypes and the choice of learning parameters. Over the years, various modifications and extensions to the standard LVQ algorithm have been proposed to improve its robustness and adaptability to different types of data distributions [54, 87]. Based on the update methods of the prototypes, we can classify the LVQ methods into prototype updating by conditions and rules and prototype updating by parameter optimization [118].

2.2.2.1 LVQ Updating with Conditions and Rules

Learning Vector Quantization (LVQ) is a supervised learning strategy centered around prototypes, predominantly employed for categorization tasks. Central to the LVQ methodology is the iterative adjustment of the prototypes, which are representative vectors for each class in the dataset. The adjustment or update of these prototypes is governed by specific conditions and rules, which ensure that they are optimally placed to represent the underlying data distribution [55].

The update process in LVQ is typically guided by the principle of reducing the classification error. When an input vector is presented to the system, its nearest prototype is identified based on a chosen distance metric, commonly the Euclidean distance. If the identified prototype belongs to the correct class of the input vector, it is moved closer to the input vector; otherwise, it is moved away. The magnitude of this movement is determined by a learning rate, which often decreases over time to ensure convergence [56].

Several variants of LVQ have been proposed over the years, each introducing different updating conditions and rules. For instance, Kohonen's LVQ2.1 and OLVQ3 methods introduced more sophisticated updating rules that consider not just the nearest prototype but

also other neighboring prototypes. These methods aim to improve prototype positioning, especially in regions where class boundaries are ambiguous [53].

Additionally, adaptive mechanisms have been integrated into the LVQ to enhance its Federated Learningexibility. Adaptive LVQ methods dynamically adjust the learning rate and other parameters based on the evolving state of the learning process. This adaptivity ensures that the prototypes are updated in a manner that is sensitive to the local structure of the data, leading to improved classification performance [87].

In summary, the essence of LVQ lies in its prototype-based approach, where iterative updating of prototypes, governed by specific conditions and rules, plays a pivotal role. The continuous evolution of LVQ methods, with their varied updating strategies, underscores the importance of prototype positioning in achieving accurate and efficient classification.

2.2.2.2 LVQ Updating by Parameter Optimization

Learning Vector Quantization (LVQ) is a renowned prototype-based supervised learning approach, specifically tailored for classification tasks. Although traditional LVQ methodologies rely on heuristic update rules to refine prototypes, recent advances have moved toward a more principled approach, defining prototype updating as a parameter optimization problem [55].

In the context of parameter optimization, the objective is to find the optimal set of prototype vectors that minimize a given cost function. This cost function typically quantifies the misclassification error or the distance between the data points and their corresponding prototypes. By casting the prototype update problem in an optimization framework, powerful optimization techniques, such as gradient descent or evolutionary algorithms, can be used to systematically adjust prototypes [86].

One notable advantage of this optimization-driven approach is its ability to consider the global properties of the data. Although heuristic rules often focus on local adjustments based on individual data points, optimization techniques can provide a holistic view, ensuring that prototypes are placed to best represent the overall data distribution. Such circumstances can contribute to the development of classification models that are more resilient and adaptable across diverse scenarios, thereby enhancing their robustness and generalizability. [87].

Furthermore, the use of parameter optimization in LVQ allows for the integration of regularization techniques. Regularization can prevent overfitting by penalizing overly complex models, ensuring that learned prototypes capture the underlying data structure without being overly sensitive to noise or outliers [23].

In summary, while traditional LVQ methods have provided valuable tools for prototype-based classification, the shift toward parameter optimization offers a more systematic and robust approach to prototype updating. By leveraging optimization techniques, LVQ can achieve better prototype placements, leading to better classification performance and more interpretable models.

2.3 Semi-Supervised Learning

Semi-supervised learning, an important and developing approach in the field of machine learning, acts as an intermediary between two fundamental learning methods: supervised and unsupervised learning. Its core principle is to utilize both labeled and unlabeled data to enhance the accuracy of machine learning models [130]. Diverging from conventional supervised learning, which depends solely on labeled instances, and unsupervised learning, which deals with unlabeled data for tasks like clustering, semi-supervised learning merges both data varieties to refine prediction precision. [17]. The blending of labeled and unlabeled data is crucial in broadening the applicability of machine learning to practical situations. The foundational concepts of semi-supervised learning rest on leveraging both types of data

to enhance the model's ability to generalize. Labeled data provide explicit guidance to the model, enabling it to make informed predictions, while unlabeled data contributes implicit information about the underlying data distribution. Techniques such as label propagation, self-training, and co-training are critical to the semi-supervised learning paradigm, as they harness these principles to effectively leverage both types of data [12].

2.3.1 Consistency Regularization

Consistency Regularization is a prominent technique in semi-supervised learning that capitalizes on unlabeled data by ensuring that a model's predictions remain consistent across various perturbations or views of the same input. This approach is rooted in the belief that slight alterations to an input, whether through noise, augmentation, or other means, should not drastically change the model's output. Over the years, numerous methods have been developed under this umbrella, such as ladder networks, the Pi model, and mean teacher, among others. These methods generally combine labeled and unlabeled data in their training processes, using labeled data for traditional supervised learning and unlabeled data to enforce prediction consistency across perturbed versions of the same input. The overarching goal is to improve the generalization and performance of the model by effectively harnessing latent information in unlabeled data [130].

- **Ladder Networks** represent a significant advancement in the realm of semi-supervised learning. Ladder networks[82], merge the techniques of supervised and unsupervised learning to set new records on diverse benchmarks. Their structure includes a forward path for standard predictive tasks and a backward path for reconstructing the input using the layer-by-layer representations. The integration of supervised loss from the forward path with unsupervised loss from the backward path enables Ladder Networks to effectively harness both labeled and unlabeled data, enhancing the overall learning experience.

- **Pi-Model** is a notable approach in the domain of semi-supervised deep learning [59]. According to the document, the Pi-Model streamlines the complexity of the Ladder Networks' Γ-Model by eliminating the need for a separate encoder for corrupted inputs and instead using a single network to generate predictions for both altered and intact inputs. This approach takes advantage of the inherent variability in neural networks' prediction functions, which often results from regularization strategies like data augmentation and dropout. The Pi-Model's main goal is to achieve uniform predictions for a given input over several iterations, effectively utilizing both labeled and unlabeled data in its training regimen. The model's loss function is a blend of the supervised loss from labeled data and the unsupervised loss from unlabeled data, promoting a well-rounded learning approach.
- **Temporal Ensembling** is an extension of the Γ-Model, aiming to address the instability in target predictions during training [59]. Instead of relying on single-evaluation targets, Temporal Ensembling aggregates predictions over time, using an exponentially moving average to update the ensemble output. This approach ensures that the targets are based on a more stable and consistent set of predictions, thus improving the training signal extracted from unlabeled examples. By accumulating predictions over time and using them as ensemble targets, Temporal Ensembling provides a robust and efficient mechanism for semi-supervised learning, ensuring that the model benefits from both labeled and unlabeled data.
- **Mean Teachers** is an advancement in the realm of semi-supervised deep learning that addresses the limitations observed in previous methods such as the Pi-Model and Temporal Ensembling [98]. In traditional approaches, the same model often played dual roles: as a teacher generating targets and as a student learning from those targets. This could lead to potential issues, especially if the targets generated were misclassified. The Mean Teacher approach proposes the use of a teacher model for faster incorporation of the learned signal and to circumvent the problem of confirmation bias. In this method, the teacher model is defined as an Exponential Moving Average (EMA) of the student model. This ensures that the teacher model is updated at a faster rate and incorporates newly learned information more quickly.

The primary objective is to improve the quality of the targets either by selecting better perturbations or by choosing a more effective teacher model to generate the targets.

- **Dual Students** is a method designed to address one of the main drawbacks of using a Mean Teacher [52]. In situations involving many training cycles, the teacher model's parameters might align with those of the student model, possibly transferring any prejudiced or erratic forecasts. The Dual-Students strategy introduces the concurrent training of two student models, each with distinct initial setups. During each cycle, one student model generates the target outcomes for the other. The model that produces the most reliable predictions—those that are distant from the decision boundary and remain steady for both the original and modified inputs, is selected to set the targets.
- **Stochastic Weight Averaging (SWA)** is a technique that averages the weights of a neural network over multiple epochs during training, leading to better generalization performance [40]. Fast-SWA, as the name suggests, is an enhanced version of this method that offers faster convergence. Taking advantage of the cyclic learning rate schedule, Fast-SWA aims to traverse diverse regions of the loss landscape, capturing a wide range of network weights. This set of weights is then averaged to produce a single robust model. The efficacy of Fast-SWA has been demonstrated in various deep learning tasks, showcasing its potential to improve model performance.
- **Virtual Adversarial Training (VAT)** is an advanced regularization technique designed to improve the robustness of deep neural networks [76]. Unlike traditional adversarial training, which requires labeled data, VAT can be applied in a semi-supervised setting. The core idea behind VAT is to generate virtual adversarial perturbations-small input perturbations that maximally increase the divergence between the model's outputs. By training the model to be invariant to these perturbations, VAT enhances the model's generalization capability, especially in scenarios with limited labeled data.
- **Adversarial Dropout** is a widely used regularization technique in deep learning that involves randomly dropping out neurons during training to prevent overfitting [84].

Adversarial Dropout takes this concept a step further by introducing an adversarial perspective. Instead of randomly deactivating neurons, Adversarial Dropout aims to find the most harmful dropout mask that would maximize the loss. When the model is trained against this worst-case dropout scenario, it becomes more robust and less prone to overfitting. This method has been shown to enhance the performance of neural networks in various tasks.

- **Interpolation Consistency Training (ICT)** is a semi-supervised learning approach that takes advantage of the consistency between prediction interpolations and input data [103]. In ICT, two augmented versions of the same input are generated and their predictions are interpolated. The model is then trained to ensure that the prediction interpolated is consistent with the prediction of the input interpolated. The imposition of this consistency constraint motivates the model to generate smooth decision boundaries, a characteristic that can be especially advantageous in situations where labeled data are scarce. ICT has demonstrated its ability to achieve state-of-the-art performance across a spectrum of benchmarks.
- **Unsupervised Data Augmentation (UDA)** is a technique that aims to improve the performance of deep learning models using unlabeled data [115]. The fundamental concept of UDA is to create various augmented iterations of the unlabeled data and to maintain uniform predictions by the model for these different versions. By promoting consistency across multiple augmented perspectives of the same dataset, UDA motivates the model to acquire more generalized and durable attributes. This technique proves to be highly efficient, especially in contexts with limited labeled data, enabling models to leverage the large quantities of unlabeled data at their disposal.

2.3.2 Entropy Minimization

Entropy minimization is a pivotal concept in machine learning, particularly in scenarios where the objective is to make the most out of unlabeled data, which is a common challenge in semi-supervised learning environments. The principle revolves around the idea of reducing the uncertainty associated with the predictions of a model, thus making the decision boundaries between different classes as distinct as possible [38].

In the context of neural networks, entropy minimization is applied during the training phase, where the model is encouraged to produce probabilities close to 0 or 1, rather than uncertain predictions that hover around intermediate values. This is achieved by incorporating an entropy term into the loss function, which the model seeks to minimize. The entropy term essentially penalizes high-entropy probability distributions (i.e., those that suggest uncertainty or 'confusion') and rewards low-entropy distributions that indicate confident predictions [62].

One of the key advantages of entropy minimization is its ability to leverage large amounts of unlabeled data effectively. By pushing the model's output to be more 'decisive' or confident, it indirectly guides the learning process using the unlabeled data, which often come in larger volumes and are less expensive to acquire than labeled data. This approach has been particularly effective in various applications, including image recognition, where it helps to achieve a clearer segmentation of objects, and language processing, where it helps to provide a more definitive context understanding [16].

However, entropy minimization is not without challenges. One primary concern is the risk of the model becoming too confident in its predictions, leading to overfitting. This is especially problematic in scenarios where the unlabeled data may contain outliers or represent a distribution significantly different from the labeled training data. Researchers have proposed several techniques to mitigate this, such as introducing constraints on the model's capacity to be overly confident or employing robust optimization techniques to handle potential outliers in the data [7].

In summary, entropy minimization serves as a powerful tool in semi-supervised learning by effectively utilizing unlabeled data to enhance the model's learning. Although it presents certain challenges, ongoing research and advances in the field continue to enhance its efficacy and applicability across various domains.

2.3.3 Holistic Methods

The detailed exploration of holistic methods in deep semi-supervised learning (SSL) reveals the intricate processes these models employ to efficiently leverage labeled and unlabeled data. These methods, including MixMatch, ReMixMatch, FixMatch, Federated LearningexMatch, and FreeMatch, integrate various strategies to enhance the learning process.

- **MixMatch**, unifies various SSL strategies into a singular, efficient learning algorithm [7]. The process begins with data augmentation, where labeled and unlabeled examples are transformed into new versions. For unlabeled data, this involves creating multiple augmented versions. The next step, label guessing, generates predictions for these versions and averaging them to obtain a proxy label for each. These labels are then sharpened to encourage confident predictions, reducing the output entropy. The MixUp process follows, combining labeled and unlabeled data into a single batch, ensuring that the model treats both data types similarly during training. The final training phase involves calculating the losses for both supervised and unsupervised learning, using the cross-entropy loss for the labeled examples and a consistency loss for the unlabeled ones. This holistic approach allows MixMatch to surpass traditional SSL methods, efficiently utilizing available data.
- **ReMixMatch** enhances MixMatch by introducing distribution alignment and augmentation anchoring [8]. Distribution alignment aims to make the model's forecasts on unlabeled data resemble the labeled data's distribution. This process involves adjusting the output probability distribution based on a running average of predictions on unlabeled data. Augmentation anchoring further refines the process. Instead of

weak augmentations, ReMixMatch employs strong augmentations, using the model's predictions on weakly augmented images as proxy labels for several strongly augmented versions. These strategies, combined with the original MixMatch framework, allow for more accurate training, utilizing both labeled and unlabeled data effectively.

- **FixMatch** simplifies the SSL process by combining consistency regularization and pseudo-labeling [92]. The approach entails utilizing a mildly enhanced iteration of an unlabeled sample to forecast a label, which is adopted if the confidence of the prediction surpasses a predefined threshold. Subsequently, this label is employed to train the model using a significantly enhanced variant of the identical input. The key to FixMatch's efficiency lies in its simplicity and the use of standard augmentations, reducing the computational demand typically associated with SSL methods. The approach maintains the effectiveness of more complex methods, making it a valuable strategy in scenarios where computational resources are limited.

Semi-supervised learning (SSL) serves as an intermediary approach, blending the benefits of both supervised and unsupervised learning by employing a small set of labeled data alongside a more substantial volume of unlabeled data. This method is especially beneficial in situations where obtaining labeled data is costly or unfeasible, enabling models to discern valuable patterns and frameworks within the plentiful unlabeled data. Consequently, this enhances the model's learning accuracy beyond what could be achieved with the scant labeled data alone. SSL encompasses various techniques, including self-training, co-training, and several advanced holistic methods like MixMatch and FixMatch, which innovatively combine data augmentation, consistency regularization, and pseudo-labeling to improve model robustness and accuracy. By effectively leveraging both labeled and unlabeled data, SSL achieves a compelling balance, reducing the dependency on extensive human annotation while still providing the guidance necessary for achieving accurate predictions and profound insights from vast and complex datasets.

2.4 Federated Learning

Federated Learning is an emerging technology [66] that has captured significant interest from the research community for its innovative capabilities and potential uses [135], [124]. The core question that Federated Learning addresses is whether it is possible to train a model without having to centralize the data [100]. The Federated Learning approach is centered on collaborative efforts, which is not always a feature of conventional machine learning approaches [93]. Furthermore, Federated Learning enables the algorithms it employs to accumulate knowledge, an aspect not offered consistently by traditional machine learning techniques [67], pandey2020crowdsourcing. The versatility of Federated Learning is demonstrated through its application in various fields, including healthcare, the Internet of Things (IoT), transportation, defense, and mobile applications. Its proven reliability is supported by numerous successful experiments. However, despite its promising outlook, a deeper understanding of Federated Learning technical intricacies, such as its platforms, hardware, software and issues related to data privacy and access, remains limited [88], [2]. In this section, we will delve into an in-depth discussion on Federated Learning, particularly emphasizing aspects related to privacy, the methodology behind selecting clients, and the strategies employed for data aggregation.

2.4.1 Privacy Protection

Federated Learning inherently enhances privacy by minimizing the exposure of user data within the network, particularly to a central server. Despite the theoretical appeal of this privacy-centric approach to machine learning, it is not devoid of vulnerabilities to cyber threats. Moreover, the technological advancements underpinning Federated Learning have yet to reach a level of maturity that guarantees the resolution of all privacy concerns inherently. This section aims to explore the prevailing privacy challenges associated with federated learning and to shed light on the significant strides made in the technology of federated learning [77].

Privacy Threats and Attacks

Federated Learning aims to protect the privacy of its participants by requiring them to share only the parameters of their locally trained models, rather than the raw data itself. Despite these precautions, recent studies have identified significant privacy risks within Federated Learning frameworks. These studies suggest that adversaries might be able to infer and partially reconstruct the training data of individual participants by analyzing the parameters of the shared model. This vulnerability exposes Federated Learning to various categories of inference-based attacks, where sensitive information could be extracted from the aggregated parameters.

- Membership Inference Attacks, as implied by their name, are strategies employed to deduce details about training data. Specifically, Membership Inference attacks [101] are designed to determine whether certain data were part of a training set by exploiting the global model to extract information about other users' training data. This process involves speculative analysis and training a separate predictive model to approximate the original training data. The research highlighted in [78] investigates how neural networks are susceptible to memorizing their training data, making them vulnerable to passive and active inference attacks.
- In certain situations, updates or gradients shared by clients with a central server in Federated Learning can inadvertently disclose sensitive information. The study referred to as [74] demonstrates how such unintentional data leaks can be exploited through inference attacks, allowing the reconstruction of other clients' data. Furthermore, the research presented in [46] delves into the risk of exposing private data from well-intentioned clients through the use of Generative Adversarial Networks (GANs) in inference attacks. In this scenario, a compromised client employs GANs to create data that resemble the training set, thereby extracting confidential information from other participants in the Federated Learning setup. Furthermore, as discussed in [9], malicious or curious clients can manipulate the shared global model and its parameters to recreate training data belonging to other clients, illustrating another dimension of vulnerability within Federated Learning systems.

- Generative Adversarial Networks (GANs), which have become increasingly popular in the realm of big data, are also being applied within Federated Learning methodologies. In the context of Federated Learning, researchers in [109] have introduced a novel framework named mGAN-AI to investigate adversarial threats based on GAN. The mGAN-AI framework is specifically designed to simulate attacks within a Federated Learning setting, particularly targeting a compromised central server. This framework assesses the risk of individual user privacy breaches through such attacks. The mGAN-AI approach comprises both passive and active variants; the passive version scrutinizes all client submissions, whereas the active version aims to single out a client by exclusively providing global updates to them. In particular, the mGAN-AI framework enhances the precision of inference attacks by not disrupting the training work Federated Learningow. However, within the Federated Learning ecosystem, there is a possibility of encountering adversarial clients who might contribute outdated data in exchange for access to the updated global model. Upon acquiring the global model, these adversaries could employ inferential methods to extract confidential data from other participants. Identifying such malicious behavior is challenging due to the opaque nature of client profiles and reputations. Additionally, the practice of sharing only parameter updates during collaborative training complicates the server's ability to assess the true value of each participant's contribution to the model.

Protrction Strageties

Secure Multiparty Computation (SMC), or MPC, was initially introduced to safeguard the inputs of multiple participants engaged in a collective computational task, such as model or function computation [15]. SMC ensures secure communication through cryptographic techniques. In the context of Federated Learning, SMC has been adapted to protect client updates. Unlike traditional SMC applications, Federated Learning-focused SMC significantly enhances computational efficiency by encrypting only the model parameters instead of extensive data input, making it an attractive option within the Federated Learning framework.

The study in [3] examines the risk of data breaches from client updates at a central server and suggests a solution that combines encryption with asynchronous stochastic gradient descent. This approach aims to safeguard client data against potential leaks on the server side effectively. However, it should be noted that encryption can be resource intensive, potentially affecting the efficiency of machine learning models in larger settings.

A different investigation in [43]. integrates homomorphic encryption with differential privacy to diminish the threat of exposure of client data. This research supports the implementation of differential privacy at the client level along with the encryption of model updates to enhance data security. This method reportedly maintains high accuracy while protecting against potential breaches from servers and other Federated Learning participants. Similarly, [42] introduces a privacy-enhanced Federated Learning strategy that continues to protect privacy after training, combining encryption with client-level differential privacy, similar to the approach in [43].

Despite these advances, SMC-based solutions in Federated Learning face challenges, particularly the balance between efficiency and privacy. SMC approaches generally require more time than standard Federated Learning processes, potentially hindering model training. This issue is exacerbated in scenarios where timely data is crucial, as extended training durations can diminish the value of the data. Additionally, developing a lightweight SMC solution suitable for Federated Learning clients remains an unresolved challenge.

Differential Privacy represents a cornerstone in the domain of privacy preservation, widely embraced by both the commercial and academic sectors. The essence of DP lies in protecting privacy through the strategic incorporation of noise into individual sensitive attributes [31], ensuring the protection of each user's privacy. This method allows for a minimal loss in the quality of statistical data due to noise addition, which is a small price to pay for the significant enhancement in privacy safeguards. In the Federated Learning paradigm, DP is used to inject noise into the parameters uploaded by participants, thereby thwarting attempts at reverse engineering the data.

The DPGAN framework, outlined in [114], leverages DP to counteract the efficacy of GAN-based attacks aimed at deducing the training data of other users within a deep learning network. In a similar vein, the DPFedAvgGAN framework [4] is tailored specifically for Federated Learning contexts. The discourse in [177] delves into the advantages of DP, elucidating its fundamental principles such as randomization and composition, and illustrating its application across various machine learning modalities, including multi-agent systems, reinforcement learning, transfer learning, and distributed ML. Further contributions, as seen in [172] and [178], amalgamate secure multiparty computation with DP to forge a fortified Federated Learning framework that does not compromise accuracy. The innovation presented in [179] seeks to bolster the privacy assurances of Federated Learning models by integrating a Federated Learning mechanism with DP and concealing user data through an algorithm 'invisibility cloak'. However, these methodologies introduce a degree of uncertainty in the parameters that are uploaded, which could potentially degrade the effectiveness of training. Moreover, such techniques pose additional challenges for Federated Learning servers, complicating the assessment of client contributions and the subsequent calculation of rewards or payoffs.

2.4.2 Aggregation Strategies

Federated learning represents a distributed, cooperative form of machine learning where network participants work together to refine a universal model, all while ensuring their individual data remains secure. Central to this approach is the need for an effective method to amalgamate the findings, whether these are the full models, specific parameters, or gradients. Each participant locally develops a model using their own data and then forwards these insights to a central hub. At this juncture, a unification technique is applied to merge these contributions, establishing a collective effort among the participants. This consolidated knowledge is then used to enhance the overarching model. Through this model updates aggregation, the central node can tap into the variety of the training data without direct

exposure to the confidential information. The choice of a unification method is pivotal in federated learning, with each alternative bringing its own set of benefits and challenges [119].

Average Aggregation

In the initial and most widely acknowledged approach, the server consolidates the incoming data, which may include model updates, parameters, or gradients, by computing the arithmetic average of these updates [73]. If we denote the ensemble of participating clients as N and their individual updates as w_i, then the aggregated overall update W is formulated as:

$$W = \frac{1}{N} \sum_{i=1}^{N} w_i \tag{2.1}$$

Clipped Average Aggregation

This methodology bears resemblance to the average aggregation process, where the mean of the received updates is computed. However, it uniquely incorporates a preliminary step of bounding the model updates within a specified range before performing the averaging. This step is pivotal in mitigating the influences of outliers and adversarial clients who may submit disproportionately large or deleterious updates. Let N represent the consortium of participating clients and w_i their corresponding weights. The function clip(x, c) is introduced, which constrains the values of x to within $[-c, c]$, where c is the cutoff threshold. The aggregated clipped update W is then given by:

$$W = \frac{1}{N} \sum_{i=1}^{N} \text{clip}(w_i, c) \tag{2.2}$$

Secure Aggregation

Technologies such as homomorphic encryption, secure multiparty computation, and secure enclaves enhance the security and confidentiality of the aggregation phase. These strategies ensure the protection of client data throughout the aggregation stage, which is paramount in

settings where protecting data privacy is crucial [13]. Secure aggregation emerges from the fusion of such security mechanisms with existing aggregation algorithms to forge a novel and secure algorithm.

Differential Privacy Average Aggregation

This approach incorporates the principles of differential privacy within the aggregation mechanism to ensure the privacy of client data. Before sending their model updates to the server, clients obscure their data by integrating stochastic perturbations. The server amalgamates these altered updates to formulate the final model. The magnitude of the stochastic perturbations added to each update is carefully calibrated to harmonize the dual objectives of privacy preservation and model accuracy retention [111]. Let N denote the collective of contributing clients, w_i their respective updates, and n_i a stochastic perturbation vector drawn from a Laplace distribution characterized by a scale parameter b, which defines the privacy budget. The cumulatively updated model W, endowed with differential privacy, is formulated as follows:

$$W = \frac{1}{N} \sum_{i=1}^{N} (w_i + n_i) \tag{2.3}$$

Here, n_i represents the noise vector amalgamated with the update of each client w_i, ensuring differential privacy, modulated by the scale parameter b.

Momentum Aggregation

This technique aims to address the issue of slow convergence in federated learning by introducing a "momentum" term maintained by each client. This term captures the trajectory of past model changes. Before dispatching a new update to the server, the momentum term is combined with it. The server then aggregates these momentum-enhanced updates to construct the final model, potentially accelerating the convergence process [116].

2.4.3 Clients Selection Methods

Choosing participants is a key component in Federated Learning (FL) frameworks, a type of distributed machine learning that facilitates model training across numerous decentralized nodes or servers (clients) without centralizing data. This method tackles several issues, such as privacy protection, data safeguarding, and bandwidth optimization. The client selection process in FL directly impacts the efficiency, fairness, and performance of the learning model [37]. This section will introduce several client selection strategies.

- Random selection in federated learning involves choosing a subset of devices or participants from a broader pool at random to contribute data or computational resources to a FL model. This technique aims to create a representative sample of the overall data, enhancing the model's ability to generalize. The method described in [73] employs iterative model averaging to train deep neural networks within a federated learning framework, thoroughly assessing its efficacy across various datasets. Despite the challenges posed by client availability, unbalanced, and non-IID data, this approach utilizes random client selection. Initially, the server randomly picks a set of clients, each of which then leverages global information and their specific data to perform local computations. These clients then send their updates back to the server, which updates the global model. This process is cyclical, engaging different clients in each iteration. This strategy enhances processing parallelism by allowing a greater number of clients to operate independently between communication rounds and increases client-side computations to minimize communication overhead. The observed results show that this technique efficiently manages imbalanced and non-IID datasets, markedly diminishing the need for communication rounds by a magnitude of ten to hundred when contrasted with synchronized stochastic gradient descent approaches.
- In federated learning systems, having full state knowledge means that all devices possess a complete understanding of the global model. Conversely, partial-state knowledge suggests that certain devices have restricted information and may lack

access to the complete details of the global model. The research outlined in [65] has shown that federated learning models can achieve convergence when clients are fully engaged in the training process, although these findings are limited to scenarios of unbiased client participation. On the contrary, the study in [73] explores the convergence of federated learning in conditions where clients may enter or exit the training process at will or may submit incomplete updates to the server. It is notable that a significant portion of client selection methodologies discussed in the literature, such as those in references [1], [22], [21], and [83], presuppose access to client-side state information. Nevertheless, the assumption that complete client-side state information is readily available in federated learning systems might not hold or be practical in real-world applications. For example, in extensive federated learning networks comprising thousands of clients, gathering and communicating detailed state information for each participant could be challenging. In addition, the acquisition of such data could raise privacy concerns among clients, potentially leading to reduced willingness to participate and lower overall client engagement in the system.

- Contribution control mechanisms are strategies designed to regulate the participation of clients in the training process to enhance the efficiency of federated learning. In the study referenced as [35], the authors introduce an adaptive client selection method that recalculates a participation threshold in each training round. This allows devices to autonomously determine their participation based on their local contribution to the model, relative to the dynamically adjusted threshold. Research presented in [3] investigates the effects of biased client selection on the convergence of FL models, particularly noting that favoring clients with higher local losses can lead to faster convergence rates. They suggest a novel selection strategy, termed Power-Of-Choice, which outperforms traditional selection methods by achieving quicker convergence and enhanced model performance.

2.5 Federated Semi-Supervised Learning

Federated Semi-Supervised Learning is an emergent machine learning framework that synergizes the distributed nature of federated learning with the label-efficient strategies of semi-supervised learning. In this setup, a central server orchestrates the learning process across a network of clients, such as mobile devices or hospitals, each holding a local dataset characterized by a small portion of labeled instances alongside a larger pool of unlabeled ones. Clients perform local computations, using labeled data for supervised learning tasks and applying semi-supervised techniques such as pseudo-labeling [62] or consistency regularization [85] to utilize unlabeled data. These local updates are periodically transmitted and aggregated by the central server to refine a global model, in a manner similar to the federated average algorithm [73]. This collaborative approach preserves data privacy, as sensitive information remains on the client's side, only sharing model updates with the server, and has been particularly beneficial in privacy-sensitive domains, as exemplified by works such as "Federated Semi-Supervised Learning with Inter-Client Consistency and Disjoint Learning" [127]. Through this methodology, Federated Semi-Supervised Learning aims to expand the horizon of machine learning applications by leveraging large-scale unlabeled datasets without compromising data privacy.

2.5.1 The Label-at-Server Setting

In federated learning environments where labeled data are distributed across client devices, a unique set of challenges and methodologies arises. This is known as the label-at-client setting. Several approaches have been developed to address these challenges, including RSCFed [107], FedSSL [126], FedMatch [50], FedPU [68], AdaFedSemi [122], and DS- Federated Learning [**smith2021ds Federated Learning**]. Each of these methods brings a different strategy to handle the semi-supervised nature of Federated Learning in this setting.

RSCFed, standing for Robust Semi-Supervised Federated Learning, is crafted to address the challenges of label scarcity and Data Heterogeneity within the realm of Federated Semi-Supervised Learning, as discussed in [107]. The framework adopts a mentor-mentee scheme to facilitate training on unlabeled client-side data. To counteract data disparity, RSCFed employs a technique of subconsensus sampling alongside a method of aggregation that considers distance. This strategy leads to the generation of multiple subconsensus models through the selection of client subsets, ensuring the inclusion of clients possessing labeled data in each constructed model.

FedSSL stands for Federated Semi-Supervised Learning [126]. It is a framework that extends the standard Federated Learning to semi-supervised settings by incorporating unlabeled data in the learning process. FedSSL typically involves mechanisms for pseudo-labeling the unlabeled data, which are then used to augment the training process. This approach helps to take advantage of the abundance of unlabeled data on the client side, thus enhancing the overall performance of the model.

FedMatch addresses the challenge of non-IID data distribution and limited labeled data in federated settings [18]. Introducing a novel algorithm that carefully aligns the distribution of labeled and unlabeled data during the aggregation process. By matching the distributions, FedMatch ensures that the global model benefits from the unlabeled data without being biased towards the data distribution of any single client.

FedPU is a methodology designed to address situations where only positive and unlabeled data are accessible on the client side [68]. It proves especially valuable in privacy-sensitive contexts where negative labels might not be shared due to privacy considerations. FedPU uses strategies to estimate the distribution of positive and negative classes based on positive and unlabeled data, enabling efficient learning within these limitations.

AdaFedSemi is a method that dynamically adjusts the learning process based on the availability and quality of the labeled data between clients [122]. It uses adaptive weighting schemes

to give more importance to clients with higher quality labels and adjusts the aggregation process accordingly. This adaptability makes it suitable for heterogeneous networks with varying amounts of labeled data.

DS- Federated Learning is a technique that focuses on the diversity of data samples during the training process [**smith2021ds Federated Learning**]. Select a diverse set of samples from clients' local datasets, both labeled and unlabeled, to train the global model. The diversity in the sampled data helps to build a more robust and generalizable model that performs well across different client distributions.

Semi Federated Learning is a framework that combines the start strategy [28] of FixMatch [92] and MixMatch [7].For each client, Semi Federated Learning first applies the FixMatch strategy to process its unlabeled data.After building the high-confidence dataset through FixMatch, Semi Federated Learning further uses a MixMatch-inspired method to build another dataset. This involves replacement sampling of the high-confiden ce dataset to increase data diversity. The client models are subsequently trained using these two datasets. By combining high-confidence pseudo-labels (from FixMatch) and data diversity (from the MixMatch-inspired dataset), Semi Federated Learning maintains high label quality while enhancing the model's adaptability to unlabeled data, and uses this mixed dataset to train the client's model.

Each of these methods contributes to the advancement of federated semi-supervised learning by providing innovative solutions to the unique challenges presented by the label-at-client setting. They aim to leverage both labeled and unlabeled data effectively, ensuring that the global model benefits from collective knowledge while respecting the privacy and data distribution constraints inherent in Federated Learning.

2.5.2 The Label-at-Client Setting

The label-at-server setting in federated learning is a challenging scenario because all clients possess only unlabeled data, which offer no additional supervision signals for the Federated Learning model. This can lead to catastrophic forgetting, where the model loses the knowledge it has learned from the limited labeled data available at the server, thus degrading performance. To combat this, two main approaches are discussed.

FedMatch, as introduced in[50], employs a dual-parameter strategy, maintaining distinct parameter sets for labeled and unlabeled datasets. During the training phase on unlabeled datasets, the parameters designated for labeled data remain static, and the converse is true, safeguarding against the erasure of acquired knowledge. FedMatch enhances communication efficacy by selectively engaging the parameters associated with unlabeled data, thus curtailing the volume of data exchanged between the clients and the central server. To address the issue of client data diversity, FedMatch integrates a loss function that promotes output consistency across different clients for identical inputs. In contrast, Semi Federated Learning [**diao2022semi**], navigates the segregation of labeled and unlabeled data by refining the global model with labeled data to augment its accuracy and alleviate the challenge of knowledge loss during unsupervised client training sessions. Diverging from FedMatch's focus on inter-client output regularization, Semi Federated Learning seeks to harmonize the client models with the global model through the use of pseudo-labels generated by the global model for the client's unlabeled data, urging the client models to align with these pseudo-labels. Comparative studies suggest that Semi Federated Learning surpasses FedMatch, delivering more robust outcomes.

Both methods aim to enhance the performance of Federated Learning models under the constraint of having only unlabeled data at the client level, but take different paths to achieve this goal. FedMatch focuses on disjoint learning and inter-client consistency, while Semi Federated Learning emphasizes the alignment of client models with the global model through pseudo-labels.

CHAPTER 3

Architecture, Training Process, and Theoretical Foundations

This chapter delineates the algorithmic approach and theoretical underpinnings of the FedALL framework proposed by our team. Initially, Section 3.1 provides a comprehensive overview of the FedALL framework, elucidating the training procedures employed on both the server and the client sides. Subsequently, Section 3.2 delves into the fundamental components and learning strategies pivotal to the FedALL system. Section 3.3 undertakes an exhaustive theoretical analysis on the convergence of FedALL. Lastly, Section **??** unveils the pseudocode for FedALL, encapsulating the principles and methodologies elucidated in the preceding sections.

3.1 Framework Design

3.1.1 Overview

To address the issues of slow convergence speed and unreliable pseudo-labels discussed in Section 1.1, we have developed the FedALL as shown in Figure 3.1. This framework, based on federated learning, employs innovative strategies. To better use the unlabeled images, FedALL generates an adaptive label list, which encompasses a range of possible candidate labels for each unlabeled sample, thus providing a comprehensive and flexible foundation for model training instead of using the unique pseudo-label. Subsequently, FedALL further refines and updates this adaptive label list by employing a Label Disambiguation Prototype

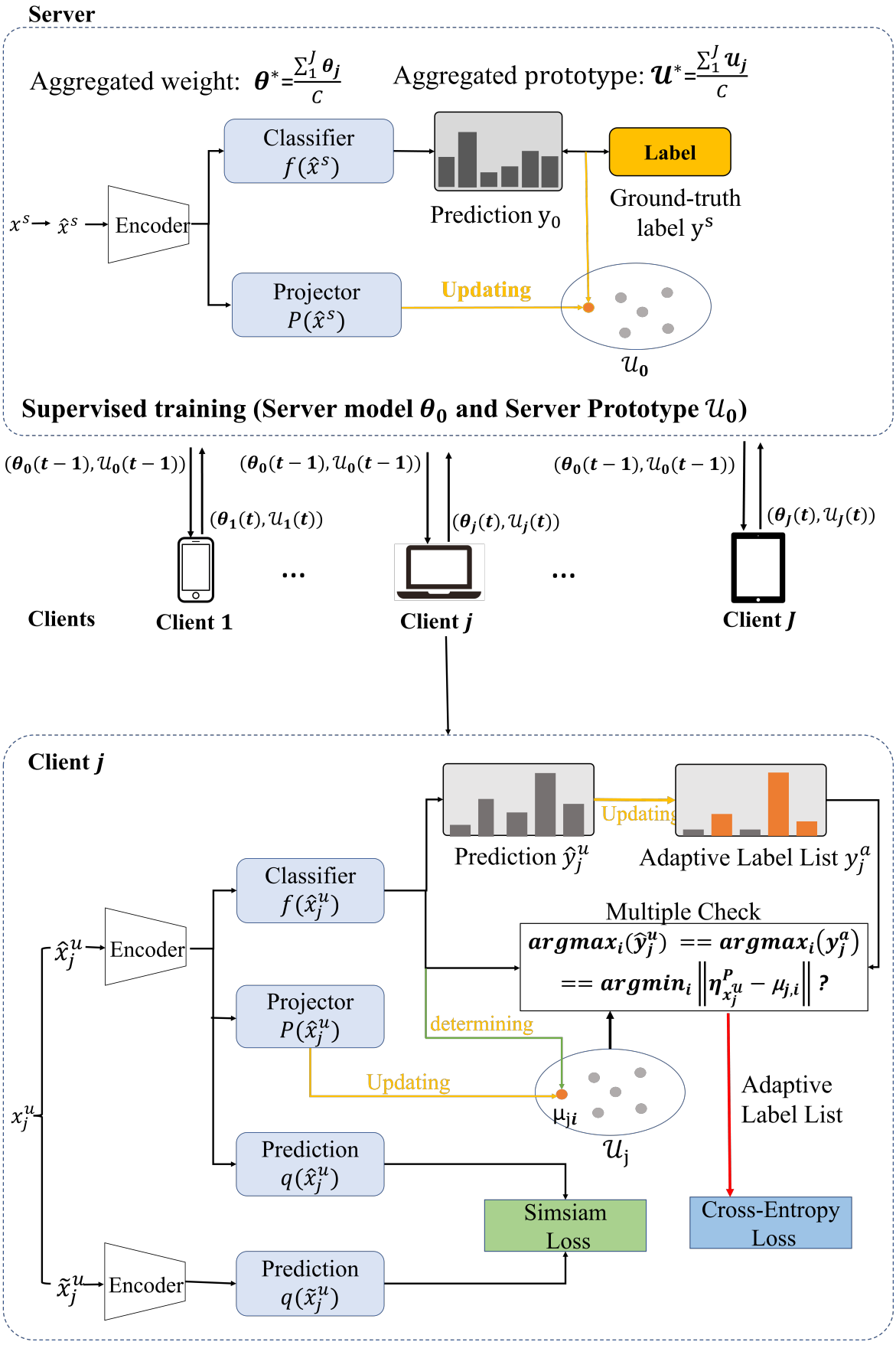

FIGURE 3.1. The overall framework of FedALL.

with the transferred knowledge. The prototype will effectively identify the most possible ground-truth within the adaptive label list. By leveraging the transferred knowledge, the convergence rate of FedALL is significantly enhanced compared to contrastive learning-based frameworks.

Additionally, to comprehensively understand the mechanism of FedALL, we define the entire training dataset D as follows:

$$D = [X, y] = \begin{bmatrix} X^s & y^s \\ X^u & y^u \end{bmatrix} = \begin{bmatrix} D^s \\ D^u \end{bmatrix}, \tag{3.1}$$

where X represents the input data and y is the corresponding label. X^s and y^s denote the inputs with known labels, forming the server dataset $D^s = [X^s, y^s]$. On the contrary, X^u refers to unlabeled input data and y^u serves as a label space that encompasses all possible labels corresponding to X^u in clients. x^s and x^u respectively signify individual images within X^s and X^u.

3.1.2 Server Side Design

The FedALL begins its process on the server. Initially, the server model is loaded with transferred weights from the pre-trained model, and uses the Cross-Entropy loss with the predicted labels of the classifier and y^s to warm up the model. During training, the model uses the projector's representation output P, which shares the same encoder with the classifier to build the embedding space named the **Label Disambiguation Prototype** $\mathcal{U}_0$, the details will be discussed in 3.2.1. After completion of training and construction $\mathcal{U}_0$, the server distributes the weight of the trained model θ_0 along with $\mathcal{U}_0$ to selected clients. Subsequently, the server receives updated weights $\mathcal{U}_j$ and label disambiguation prototypes $\mathcal{U}_j$ from clients and obtains the average weight θ^* and the averaged prototype $\mathcal{U}^*$. Following this, the server model is

trained on D^s, after loading θ^*. During training, $\mathcal{U}^*$ is updated with the output of the projector. This iterative process is repeated until the training is complete.

3.1.3 Client Side Design

Suppose the client j has received $\mathcal{U}_0$ and θ_0 from the server in the communication round t. After loading θ_0, the client will use the output of the Classifier $f(\hat{x}_j^u)$ to build the **Adaptive Label List** y_j^a for each of the weak unlabeled augmented data $\hat{x}_j^u$ within the client, and y_j^a will be updated in the following communication rounds. Details of this process are elaborated in 3.2.2. The $\mathcal{U}_0$ will be updated with the output from the Classifier $f(\hat{x}_j^u)$ and the Projector $P(\hat{x}_j^u)$, where $f(\hat{x}_j^u)$ determines which label cluster μ_{ji} needs to be updated and the $P(\hat{x}_j^u)$ updates the value of this cluster, the details provided in 3.2.2. **Multiple Check** will be applied to determine the potential ground-truth within y_j^a, and the selected y_j^a will be used as the pseudo-label in the Cross-Entropy Loss. To promote model convergence, SimSiam [20] has been implemented in the client, since it will use the prediction output $q(\hat{x}_j^u)$ and $q(\tilde{x}_j^u)$ based on the weak and strong augmented image x_j^u. Overall, the update of the model on the client side is a result of the synergistic effect of Cross-Entropy and SimSiam losses.

3.2 Learning Methods

3.2.1 Knowledge Transfer

Transfer learning, having exhibited remarkable performance in the realm of deep learning, leverages pre-existing knowledge to yield a more precise and efficient learning process, thus enhancing the overall efficacy of the model.

As mentioned above, the server loads the transferred weight and warms the model by training D^s to get a trained model θ_0, and uses the projector representation P to initialize a **Label Disambiguation Prototype** $\mathcal{U}_0$.

Each of the clusters μ_{0_i} in $\mathcal{U}_0$ corresponds to each label i in $\mathcal{D}^s$, representing the mean value of the embedding vectors of instances within that label, where μ_{0_i} can be defined as:

$$\mu_{0_i} = \frac{1}{|D_i^s|} \sum_{\hat{x}_i^s \in D_i^s} P\left(\theta_0; \hat{x}_i^s\right), \tag{3.2}$$

where $\hat{x}_i^s$ is the input sample after weak augmentation of label i from D^s. $P(X)$ is the projection function, and we use the labeled data $\hat{x}_i^s$ and the data transferred θ_0 as our input here for the training preparation.

The server transmits the trained weight θ_0 to all clients, helping them establish the **Adaptive Label List** y^a. During the establishment of y^a within clients, each of the unlabeled samples X^u within clients is assigned its unique adaptive label list. Distinct from the traditional approach of assigning a single top-1 label, the adaptive label list in FedALL functions as a'soft label' mechanism. It is characterized by encompassing the top K labels, representing the k classes with the highest predicted probabilities. This method allows for a more nuanced and probabilistic approach to labeling.

For each unlabeled sample X^u on the client side, the model classifier evaluates and selects the top K classes based on their prediction value after SoftMax. The selected K classes are assigned a uniform probability distribution as $y_i^a = \frac{1}{K}$, in accordance with the methodologies outlined in previous works, such as [106].

In essence, transfer learning forms a foundational component of FedALL, primarily by utilizing pre-trained weights. This strategy not only kick-starts the learning process in FedALL, but also significantly enhances the precision of the label disambiguation prototype and the reliability of the adaptive label list. The synergy of these elements substantially

increases the effectiveness and accuracy of the FedALL algorithm, showcasing its ability to adapt and learn efficiently in a federated setting.

3.2.2 Label Disambiguation Prototype

During the initial stages of training preparation, $\mathcal{U}_0$ is generated on the server. In the training round t, where $t \in (1, T)$ and T is the total number of training rounds, both $\mathcal{U}_0$ on the server and $\mathcal{U}_J$ on the clients are gradually updated with η_X^P, which represents the output of the projection function $P(X)$ and is depicted in Figure 3.1. Specifically, in client j, FedALL employs $\eta_{\hat{x}_j^u}^P$ derived from weakly augmented unlabeled data $\hat{x}_j^u$, to update the cluster $\mu_{j,i}$ in $\mathcal{U}_j$.

In FedALL, directly substituting the prototype $\mu_{j,i}$ with the output $\eta_{\hat{x}_j^u}^P$ could lead to the loss of previously acquired knowledge and the incorporation of erroneous information. This concern is especially pertinent given the unlabeled nature of client data, which could compromise its reliability.

To mitigate this problem, FedALL employs a strategy **moving average update** to refine $\mu_{j,i}$. Specifically, during each iteration r, the updated $\mu_{j,i}$ is derived from a mixture of its previous iteration's value $\mu_{j,i}(r-1)$ and the current output $\eta_{\hat{x}_j^u}^P$. This blending process is governed by a ratio γ, where $\gamma \in (0, 1)$. Furthermore, r falls within the range of 1 to R, where R represents the total number of iterations of the client.

The classifier generates a predicted label $argmax_i(\hat{y}_i^u)$ for each unlabeled sample x_j^u, pinpointing the exact cluster that requires updating, as illustrated in Figure 3.1. The update mechanism for the prototype is mathematically formulated as follows:

$$\mu_{j,i}(r) = \begin{cases} \gamma\mu_{j,i}(r-1) + (1-\gamma)\eta_{\hat{X}_{j,i}^u}^P(r) & \text{if } i = \arg\max_i \hat{y}^u, \\ \mu_{j,i}(r-1) & \text{otherwise.} \end{cases} \tag{3.3}$$

On the server, FedALL aggregates the updated $\mathcal{U}_j$ and θ_j from the $t-1$ round to obtain $\mathcal{U}^*$ and θ^*. In round t, the server generates a new $\mathcal{U}_0$ using $\mathcal{U}^*$ and the projector output $\eta^P_{\hat{x}^s}$ from the model loaded with θ^*. Since the server conducts supervised learning, FedALL uses the ground-truth corresponding to μ_{0_i}. It can be defined as:

$$\mu_{0_i}(t) = \gamma\mu_{0_i}(t-1) + (1-\gamma)\eta^P_{\hat{X}^s_i}(t), \tag{3.4}$$

by employing the moving average update strategy, FedALL effectively leverages the information from previous training rounds while minimizing disruptions caused by unlabeled data.

Privacy Preserving

FedALL introduces a method where prototypes are exchanged between the server and clients instead of traditional model parameters, offering privacy advantages in Federated Learning. The use of prototypes, which are 1D vectors formed by averaging low-dimensional representations of class-specific samples, inherently safeguards data privacy through an irreversible aggregation process. This makes it impossible for attackers to revert these prototypes to the original data without access to the local models on client devices [97]. Additionally, FedALL's framework is compatible with a range of privacy-preserving mechanisms, further boosting the system's security and trustworthiness.

3.2.3 Adaptive Label List

As stated previously, every client has established an adaptive label list for each unlabeled sample in X^u as y^a with the top K pseudo-labels. The selected top K classes in y^a will be subject to a mathematically uniform distribution as $y^a_i = \frac{1}{K}$, while the possibility of other classes in y_a is 0. FedALL uses a **moving average update** to emphasize the potential ground-truth in the adaptive label list, distinguishing it from other classes. This dynamic approach contrasts with static one-hot labels, which, if initially erroneous, are challenging to rectify. More specifically, for the unlabeled sample within y^a, if the label i in its y^a with the highest probability matches its predicted label from the classifier, the probability of y^a_i will

increase, while the others will decrease. This procedure can be formally defined as

$$y_i^a = \alpha y_i^a + (1-\alpha)z \tag{3.5}$$

in which

$$z = \begin{cases} 1 & \text{if } \arg\max(\hat{y}^u) == \arg\max(y^a) \\ 0 & \text{else,} \end{cases} \tag{3.6}$$

where α is a constant number between 0 to 1. The Adaptive Label List serves as a robust countermeasure against the challenges posed by incorrect one-hot pseudo-labels during model training. However, FedALL requires a further strategy to discern and select the most reliable label from the adaptive label list. Therefore, FedALL adopts a **Multi-check** technique in clients to ensure that the model learns the knowledge with higher credibility, in client j it can be described as:

$$\delta(\hat{y}_j^u, y_j^a, \eta_{X_j^u}^P, \mathcal{U}_j) = \mathbb{1}(A_i == B_i == C_i) \tag{3.7}$$

in which

$$\begin{cases} A_i = \arg\min_i \|\eta_{X_j^u}^P - \mu_{j,i}\| \\ B_i = \arg\max_i(\hat{y}_j^u) \\ C_i = \arg\max_i(y_j^a), \end{cases} \tag{3.8}$$

where A_i is the label of the cluster $\mu_{j,i}$ in $\mathcal{U}_j$, which has the closest distance to $\eta_{X_j^u}^P$, B_i is the predicted label with the highest probability of the classifier, and C_i is the label in y^a which has the highest probability. After **Multi-check**, the selected **Adaptive Label List** will be used in the cross-entropy loss, which can be formally defined as follows:

$$H = \delta\left(\hat{y}^u, y^a, \eta_{X^u}^P, \mathcal{U}_J\right) \mathrm{E}_{y_s}\left[-\log y_p^u\right]. \tag{3.9}$$

In the FedALL algorithm, the construction of the adaptive label list aims to include the most probable classes for each unlabeled sample, optimizing the chance that the actual ground-truth label is among them. However, as highlighted in the case of image n in Figure 1.1, there remains the possibility that ground-truth may not be present within these selected classes. To address this challenge and enhance the likelihood of encompassing the true label during the training process, FedALL incorporates a **Label Correction** mechanism as part of the client's training regimen. It operates under the premise that if the classifier's prediction consistently

falls outside the selected y^a label list for a given X^u across consecutive iterations, and if this prediction repetitively aligns with a specific label, it strongly suggests that the current selection of k classes in y^a might be missing the actual ground-truth label. To rectify this, FedALL dynamically adjusts the label list by replacing the label in y^a with the lowest probability with the repeatedly predicted label. This adjustment not only introduces a potentially more accurate label into the mix, but also resets the probability distribution of the K labels in y_i^a to an equal value of $\frac{1}{K}$. This equal reset ensures that the newly introduced label is not disadvantaged in subsequent training iterations.

In our implementation, we have empirically set the threshold $O = 3$. This threshold indicates that if the classifier's prediction is not within the y^a list for three consecutive times and consistently points to the same label, the Label Correction mechanism will be triggered. This value of O is chosen to balance the response to potential misclassifications with the stability of the label list, preventing overfrequent alterations that could destabilize the training process.

Through this strategic intervention, FedALL effectively mitigates the risk of excluding the true label from the adaptive list, thereby enhancing the overall accuracy and reliability of the model's training and predictions. The Label Correction strategy is a testament to FedALL's adaptive and responsive design, tailored to address the intricacies and uncertainties inherent in dealing with unlabeled or weakly labeled data in a federated learning context.

3.2.4 Contrastive Learning

To counteract the challenge of having a limited number of pseudo-labels available after multiple check strategies on the clients, a common occurrence in the initial communication rounds or when dealing with complex datasets. FedALL integrates SimSiam loss as a supplementary mechanism to aid in model convergence on the client side.

In the context of FedALL, the SimSiam Loss is computed by evaluating the likeness between the outputs produced by the predictor, derived from both the strongly and weakly augmented renditions of the input sample X^u. The objective is to diminish the disparity between these outputs, consequently enhancing the model's capacity to discern and interpret the intrinsic patterns within the data. The optimal model parameters and representations are determined according to the following equation:

$$\theta^*, \eta^{q*} = \arg\min_{\theta, \eta^q} \mathcal{L}(\theta, \eta^q) \tag{3.10}$$

in which

$$\mathcal{L}(\theta, \eta^q) = \mathbb{E}[\|\mathcal{F}_\theta(\mathcal{T}(X^u)) - \eta^q_{X^u}\|^2], \tag{3.11}$$

where $\mathcal{F}$ refers to the network parameterized by θ, and $\mathcal{T}(\cdot)$ refers to the augmentation operator by original sample X, here the η^q refers to the aggregation of the representation of all the images from the prediction function, η^q_X refers to the representation of the image x and the expectation $\mathbb{E}[\cdot]$ is over the distribution of images and augmentations.

By incorporating this self-supervised learning strategy, FedALL effectively enhances the model's learning process in scenarios where label scarcity could otherwise hinder performance. This approach underscores the adaptability and robustness of FedALL in addressing diverse and challenging learning environments within the realm of federated learning.

3.3 Theoretical Analysis

3.3.1 Learning Process and Optimization

To simplify analysis, the system can be divided into two parts according to the data flow, server-side learning, and client-side group learning. In the group learning process on the client side, FedALL must update both the overall prototype $\mathcal{U}_J = \{\mathcal{U}_j\}$ and the parameter

θ. To specify the updates of the server and clients, we use $\mathcal{U}_0$ to present the prototype on the server, and $\mathcal{U}_j$ the client j.

The optimization has two main parts, including SimSiam and Cross-Entropy. SimSiam optimization can be equivalently formulated as (3.10)

Taking into account $\mathcal{U}_J$, the prediction reconfirmation can be formulated as (3.7) and (3.8),

in which $\mathcal{U}_j = span(\mu_{j_0}, \mu_{j_1}, \ldots, \mu_{j,i}, \ldots)$.
Thus, the overall optimization of each client can be formulated as follows.

$$\min_{\theta,\eta} \mathcal{L}(\theta, \eta, \mathcal{U}_J) \tag{3.12}$$

where,

$$\begin{aligned} \mathcal{L}(\theta, \eta, \mathcal{U}_J) = \mathbb{E}[\|\mathcal{F}_\theta(\mathcal{T}(x)) - \eta^q_{X^u}\|^2] \\ +\delta(\hat{y}^u_i, y^a, \eta^P_{X^u}, \mathcal{U}_J)\mathbb{E}_{y_s}[-\log y^u_p]. \end{aligned} \tag{3.13}$$

The update processing of η, θ can be presented as

$$\theta(t) = \arg\min_{\theta} \mathcal{L}(\theta(t-1), \eta^P_{X^u}(t-1), \mathcal{U}_J(t-1)) \tag{3.14}$$

$$\eta(t) = \arg\min_{\eta} \mathcal{L}(\theta(t), \eta^P_{X^u}(t-1), \mathcal{U}_J(t-1)), \tag{3.15}$$

where the t is the current communication epoch between the server and clients, the $t-1$ represents the previous epoch, and the update of each element $\mu_{j,i}$ in $\mathcal{U}_J$ can be obtained by (3.3).

After the update of each client, the aggregative update of prototype μ'_i of the client side will be obtained by μ_{ji} as

$$\mu'_i(t) = \sum_j^n \frac{\mu'_{j,i}(t)}{n}, \tag{3.16}$$

where n refers to the number of the selected clients in every communication epoch. This prototype result will be transferred to the server as the initial prototype $\mathcal{U}_0$ for learning on the server side. On the server side, the $\theta(t), \mu_i'$ returned by each client after aggregation will be used as a starting point for a new round of server-side learning. Since the server side has the labeled dataset, the learning is updated only by cross-entropy as the loss function.

Similarly, the overall optimization of the server can be formulated as

$$\begin{cases} \theta_0^*, \eta^* = \min_{\theta_0, \eta_{X^s}^P} \mathcal{L}(\theta_0, \eta_{X^s}^P, \mathcal{U}_0) \\ \mathcal{L}(\theta_0, \eta, \mathcal{U}_0) = \delta(\hat{y}_i^u, y^a, \eta_{X^s}^P, \mathcal{U}_0)\mathbb{E}_{y_s}[-\log y_p^u]. \end{cases} \tag{3.17}$$

The update processing of η, θ_0 can be presented as

$$\theta_0^*(t) = \arg\min_{\theta_0} \mathcal{L}(\theta_0(t-1), \eta_{X^s}^P(t-1), \mathcal{U}_0(t-1)) \tag{3.18}$$

$$\eta(t) = \arg\min_{\eta} \mathcal{L}(\theta_0(t), \eta_{X^s}^P(t-1), \mathcal{U}_0(t-1)), \tag{3.19}$$

and the update of each element μ_{0_i} in $\mathcal{U}_0$ can be obtained by (3.4).

After cascading the client group and server side to each other, the prototype after convergence from the client group at the last moment will be updated as input from the server side. The prototype output after the server side has finished learning will be reused as the new starting input of the client group.

To simplify the analysis, we can model the difference between each input and output as follows, where $j \to 0$ means the prototype from the clients to the server, while $0 \to j$ denotes the prototype from the server to the clients, as shown below,

$$\begin{cases} \Delta\mu_{j\to 0}(t) = \mu_0(t) - \mu_j'(t-1) \\ \Delta\mu_{0\to j}(t) = \mu_j'(t) - \mu_0(t). \end{cases} \tag{3.20}$$

It is essential to guarantee the convergence of learning in all terminals, especially for learning without labels. Convergence in learning is guaranteed since the labels are already owned on the server side; however, for distributed client-side label-free learning, the learning process may lead to divergence or oscillation due to uncertain updates, while determining convergence of the decentralized training process across multiple client groups in the context of federated learning is a challenge.

3.3.2 Convergent Learning

According to the theory of the momentum encoder, the $\theta(t), \eta(t)$ update in clients can converge to the same status as the trainable encoder [20], however, there is a lack of theoretical support on how to ensure convergent learning in the design of learning updates with prototype participation.

To address this issue, this section proposes and proves a set of theoretical criteria to assess the effectiveness of the federated learning architecture this section havs designed, and the related proofs can be seen in the Section Appendix. These criteria can be presented in the form of theorems that serve as guidelines for determining whether the architecture meets the required specifications.

Taking into account the probability of different conditions in (3.3), the expectation of the prototype update can be formulated as

$$\mathbb{E}\mu_{j,i}(t) = [\alpha_i\gamma + (1-\alpha_i)]\mathbb{E}\mu_{j,i}(t-1) + \alpha_i(1-r)\mathbb{E}p(\eta_X^P(t-1), \mathcal{U}_J(t-1)), \quad (3.21)$$

in which α_i refers to the probability of the event $\arg\max_i(\hat{y}^u) == \arg\max_i(y^a)$. The convergence learning of the prototype can be guaranteed with the criteria given in the following theorem.

THEOREM 3.3.1. *Let* $(\mathcal{U}_J, d)$ *be a non-empty norm space of a parameter set, in which* $\mathcal{U}_J = span\{\mathbb{E}\mu_{j,i}\}, d(x,y) = \|x-y\|$. *Under the learning strategy* $\mathbb{E}\mu_j(t) = P(\mathbb{E}\mu_j(t-1))$ *defined in* (3.3)*, when* $\arg\max_i(\hat{y}^u) == \arg\max_i(y^a)$*, there exists* $\mathbb{E}\mu^*_{j,i}$ *that*

$$\lim_{t\to\infty} \mathbb{E}\mu_{j,i}(t) = \mathbb{E}\mu^*_{j,i} \tag{3.22}$$

if there exists ϵ *that* $\gamma \leq \epsilon < 1$

$$\left\| \frac{\Delta \mathbb{E}p(\eta^P_x(t), \mathcal{U}_J(t))}{\Delta \mathbb{E}\mu_{j,i}(t)} \right\| \leq \frac{\epsilon + \alpha_i(1-\gamma) - \gamma}{1-\gamma} \tag{3.23}$$

REMARK 3.3.2. Theorem 3.3.1 provides the convergence criterion to learn the prototype strategy. On the space spanned by each prototype component $\mu_{j,i}$, a convergent optimal solution can be found when the expected change in the representation projection $\mathbb{E}\eta^P_x$ and the change in each component of the space $\mathbb{E}\mu_{j,i}$ satisfy the relationship provided by the theorem. The theorem also reveals the parametric upper bound on the variation of projection that needs to be ensured for convergent learning.

The proof of Theorem 3.3.1

DEFINITION 3.3.3. Let (X, d) be a complete metric space (in which X represents the set and d represents the metric), then a map function $P : X \to X$ is a contraction mapping on X if $\exists q \in [0, 1)$ such that

$$d(P(x), P(y)) \leq qd(x, y) \forall x, y \in X. \tag{3.24}$$

LEMMA 1. *Let* (X, d) *be a complete non-empty metric space with contraction mapping* $P : X \to X$. *Then* P *admits a unique fixed point* x^* *in* X *(that is,* $P(x^*) = x^*$*).*

PROOF. Based on Definition 3.3.3, Lemma 1 naturally holds according to the Banach Fixed Point Theorem [36]. □

With (3.21), the updating on $t-1, t$ can be formulated as

$$\begin{cases} P(\mathbb{E}\mu_{j,i}(t)) = [\alpha_i\gamma + (1-\alpha_i)]\mathbb{E}\mu_{j,i}(t) + \alpha_i(1-r)\mathbb{E}p(\eta_x^P(t), \mathcal{U}_J(t)) \\ P(\mathbb{E}\mu_{j,i}(t-1)) = [\alpha_i\gamma + (1-\alpha_i)]\mathbb{E}\mu_{j,i}(t-1) + \alpha_i(1-r)\mathbb{E}p(\eta_x^P(t-1), \mathcal{U}_J(t-1)) \end{cases} \tag{3.25}$$

because

$$\mathbb{E}\mu_{j,i}(t) = P(\mathbb{E}\mu_{j,i}(t-1)) \tag{3.26}$$

it yields

$$d(P(\mathbb{E}\mu_{j,i}(t)), P(\mathbb{E}\mu_{j,i}(t-1))) = \|a\Delta\mathbb{E}\mu_{j,i}(t) + b\Delta\mathbb{E}p(\eta_x^P(t), \mathcal{U}_J(t))\| \tag{3.27}$$

where

$$\begin{cases} a = \alpha_i\gamma + (1-\alpha_i) \\ b = \alpha_i(1-\gamma) \\ \Delta\mathbb{E}\mu_{j,i}(t) = \mathbb{E}\mu_{j,i}(t) - \mathbb{E}\mu_{j,i}(t-1) \\ \Delta\mathbb{E}p(\eta_x^P(t), \mathcal{U}_J(t)) = \mathbb{E}p(\eta_x^P(t), \mathcal{U}_J(t)) - p(\eta_x^P(t-1), \mathcal{U}_J(t-1)) \end{cases} \tag{3.28}$$

It can be seen that if (3.29) holds

$$\left\| \frac{\Delta\mathbb{E}p(\eta_x^P(t), \mathcal{U}_J(t)}{\Delta\mathbb{E}\mu_{j,i}(t)} \right\| \leq \frac{\epsilon + \alpha_i(1-\gamma) - \gamma}{1-\gamma} \tag{3.29}$$

then

$$a\|\Delta\mathbb{E}\mu_{j,i}(t)\| + b\|\Delta\mathbb{E}p(\eta_x^P(t), \mathcal{U}_J(t))\| \leq \epsilon\|\Delta\mathbb{E}\mu_{j,i}(t)\| \tag{3.30}$$

because

$$\epsilon\|\Delta\mathbb{E}\mu_{j,i}(t)\| = \epsilon d(\mathbb{E}\mu_{j,i}(t), \mathbb{E}\mu_{j,i}(t-1)) \tag{3.31}$$

and

$$d(P(\mathbb{E}\mu_{j,i}(t)), P(\mathbb{E}\mu_{j,i}(t-1))) \leq a\|\Delta\mathbb{E}\mu_{j,i}(t)\| + b\|\Delta\mathbb{E}p(\eta_x^P(t), \mathcal{U}_J(t))\| \tag{3.32}$$

we have

$$d(P(\mathbb{E}\mu_{j,i}(t)), P(\mathbb{E}\mu_{j,i}(t-1))) \leq \epsilon d(\mathbb{E}\mu_{j,i}(t), \mathbb{E}\mu_{j,i}(t-1)). \tag{3.33}$$

Therefore, with Definition 3.3.3, T defined by (3.3) is a contraction mapping, so the Theorem 3.3.1 is proved with lemma 1.

THEOREM 3.3.4. *Let $(\mathcal{U}_J, d)$ be a non-empty norm space of a parameter set, in which $\mathcal{U}_J = span\{\mu_{j,i}\}, d(x,y) = \|x-y\|$. Under the learning strategy $\mu_i(t) = P(\mu_i(t-1))$ defined in* (3.3)*, when $\arg\max_i(\hat{y}^u) == \arg\max_i(y^a)$, there exists $\mu^*_{j,i}$ that*

$$\lim_{t\to\infty} \mu_{j,i}(t) = \mu^*_{j,i} \tag{3.34}$$

if there exists ϵ that $\gamma \leq \epsilon < 1$

$$\frac{\|\Delta p(\eta^P_x(t), \mathcal{U}_J(t)\|}{\|\Delta\mu_{j,i}(t)\|} \leq \frac{\epsilon-\gamma}{1-\gamma.} \tag{3.35}$$

REMARK 3.3.5. Theorem 3.3.4 reveals an upper bound on the proportionality between representation projection learning and prototype learning when the learning transformation probabilities resulting from label-free learning are non-preemptive. Since the data measurement required for this criterion is easy to compute and quantify in practical tests, it is not only a theoretical basis for convergent learning but can also be directly applied as an update condition in system implementation.

COROLLARY 3.3.6. *Let $(\mathcal{U}_J, d)$ be a non-empty norm space of a parameter set, in which $\mathcal{U}_J = span\{\mu_{j,i}\}, d(x,y) = \|x-y\|$. Under the learning strategy $\mu_j(t) = P(\mu_j(t-1))$ defined in* (3.3)*, when $\arg\max_i(\hat{y}^u) == \arg\max_i(y^a)$, there exists $\mu^*_{j,i}$ that*

$$\lim_{t\to\infty} \mu_{j,i}(t) = \mu^*_{j,i} \tag{3.36}$$

if

$$\|\Delta p(\eta^P_x(t), \mathcal{U}_J(t)\| < \|\Delta\mu_{j,i}(t)\| \tag{3.37}$$

REMARK 3.3.7. Corollary 3.3.6 reveals the relationship between the learning of representation η^P and the learning of prototype μ. When the perturbation of representation learning occurs, it will also cause the perturbation of prototype learning, but as long as the perturbation

of representation is smaller than the perturbation of prototype according to the paradigm measure, the learning will show contraction, thus ensuring the stable convergence of the overall learning.

The proof of Theorem 3.3.4

PROOF. According to (3.3), when $f(D^u_{j,i}) \neq i$ there is no update in $\mu_{j,i}$. When $f(D^u_{j,i}) = i$ the updating on $t-1, t$ can be formulated as

$$\begin{cases} P(\mu_{j,i}(t-1)) = \gamma\mu_{j,i}(t-1) + (1-r)p(\eta^P_x(t-1), (U)_J(t-1)) = \mu_{j,i}(t) \\ \\ P(\mu_{j,i}(t)) = \gamma\mu_{j,i}(t) + (1-r)p(\eta^P_x(t), (U)(t)) = \mu_{j,i}(t+1) \end{cases} \tag{3.38}$$

which yields

$$d(P(\mu_{j,i}(t)), P(\mu_{j,i}(t-1))) = \|\gamma\Delta\mu_{j,i}(t) + (1-\gamma)\Delta p(\eta^P_x(t), \mathcal{U}_J(t))\| \tag{3.39}$$

where

$$\begin{cases} \Delta\mu_{j,i}(t) = \mu_{j,i}(t) - \mu_{j,i}(t-1) \\ \Delta p(\eta^P_x(t), \mathcal{U}_J(t)) = p(\eta^P_x(t), \mathcal{U}_J(t)) - p(\eta^P_x(t-1), \mathcal{U}_J(t-1)) \end{cases} \tag{3.40}$$

It can be seen that if (3.41) holds

$$\frac{\|\Delta p(\eta^P_x(t), \mathcal{U}_J(t)\|}{\|\Delta\mu_{j,i}(t)\|} \leq \frac{\epsilon - \gamma}{1-\gamma} \tag{3.41}$$

then

$$\gamma\|\Delta\mu_{j,i}(t)\| + (1-\gamma)\|\Delta p(\eta^P_x(t), \mathcal{U}_J(t)) \leq \epsilon\|\Delta\mu_{j,i}(t)\| = \epsilon d(\mu_{j,i}(t), \mu_{j,i}(t-1)) \tag{3.42}$$

because

$$d(P(\mu_{j,i}(t)), P(\mu_{j,i}(t-1))) \leq \gamma\|\Delta\mu_{j,i}(t)\| + (1-\gamma)\|\Delta p(\eta^P_x(t), \mathcal{U}_J(t))\| \tag{3.43}$$

we have

$$d(P(\mu_{j,i}(t)), P(\mu_{j,i}(t-1))) \leq \epsilon d(\mu_{j,i}(t), \mu_{j,i}(t-1)). \tag{3.44}$$

Therefore, with Definition 3.3.3, T defined by (3.3) is a contraction mapping, so the Theorem 3.3.4 is proved with lemma 1. □

Based on convergent learning in distributed clients, aggregative learning convergence can be guaranteed by the following theorem.

THEOREM 3.3.8. *Let $(\mathcal{U}_J, d)$ be a non-empty norm space of a parameter set, in which $\mathcal{U}_J = span\{\mu'_{j,i}\}, d(x,y) = \|x-y\|$. Under the Aggregative Learning* (3.16)*, $\exists \mu'_j$ that*

$$\lim_{t\to\infty} \mu'_j(t) = \mu^*_j \tag{3.45}$$

if and only if there exists ϵ that $\gamma \leq \epsilon < 1$

$$\frac{\|\Delta\mu_{j\to 0}(t) + \Delta\mu_{0\to j}(t)\|}{\|\Delta\mu'_j(t)\|} < \epsilon \tag{3.46}$$

COROLLARY 3.3.9. *Let $(\mathcal{U}_J, d)$ be a non-empty norm space of a parameter set, in which $\mathcal{U}_J = span\{\mu'_{j,i}\}, d(x,y) = \|x-y\|$. Under the Aggregative Learning* (3.16)*, $\exists \mu'_j$ that*

$$\lim_{t\to\infty} \mu'_j(t) = \mu^*_j \tag{3.47}$$

if and only if

$$\frac{\|\Delta\mu_{j\to 0}(t) + \Delta\mu_{0\to j}(t)\|}{\|\Delta\mu'_j(t)\|} < 1 \tag{3.48}$$

COROLLARY 3.3.10. *Let $(\mathcal{U}_J, d)$ be a non-empty norm space of a parameter set, in which $\mathcal{U}_J = span\{\mu'_{j,i}\}, d(x,y) = \|x-y\|$. Under the Aggregative Learning* (3.16)*, $\exists \mu'_j$ that*

$$\lim_{t\to\infty} \mu'_j(t) = \mu^*_j \tag{3.49}$$

if and only if

$$\|\Delta\mu_{j\to 0}(t)\| + \|\Delta\mu_{0\to j}(t)\| < \|\Delta\mu'_j(t)\| \tag{3.50}$$

REMARK 3.3.11. Corollary 3.3.10 gives a more concise form to determine the stable convergence properties of the prototype learning strategy, which is easier to test in practical tests and less computationally intensive.

The Proof of Theorem 3.3.8

PROOF. In $(\mathcal{U}_J, d)$, define mapping $T : \mu'_j \to \mu'_j$ as

$$P(\mu'_j(t)) = \mu'_j(t) + \Delta\mu_{j\to 0}(t) + \Delta\mu_{0\to j}(t) = \mu'_j(t+1) \tag{3.51}$$

and $\Delta\mu'_j(t) = \mu'_j(t) - \mu'_j(t-1)$, therefore,

$$d(P(\mu'_j(t)), P(\mu'_j(t-1))) = \|\mu'_j(t+1) - \mu'_j(t)\| = \|\Delta\mu_{j\to 0}(t) + \Delta\mu_{0\to j}(t)\| \tag{3.52}$$

with (3.46), there $\exists\epsilon \in (0,1)$ that $\|\Delta\mu_{j\to 0}(t) + \Delta\mu_{0\to j}(t)\| \leq \epsilon\|\Delta\mu'_j(t)\|$ thus

$$d(P(\mu'_j(t)), P(\mu'_j(t-1))) \leq \epsilon d(\mu'_j(t), \mu'_j(t-1)) \tag{3.53}$$

Therefore, with Definition 3.3.3, T defined by (3.51) is a contraction mapping, so the sufficiency of Theorem 3.3.8 is proved with lemma 1. When (3.24) holds, if there is no $\epsilon \in [0,1)$, then

$$\|\Delta\mu_{j\to 0}(t) + \Delta\mu_{0\to j}(t)\| \geq \|\Delta\mu'_j(t)\|$$

which yields

$$\frac{d(P(\mu'_j(t)), P(\mu'_j(t-1))}{d(\mu'_j(t), \mu'_j(t-1))} > 1 \tag{3.54}$$

which violates (3.24) in Definition 3.3.3, which shows the contradiction, and the necessity is proved with proof by contradiction. Therefore, Theorem 3.3.8 holds if and only if (3.46) holds. □

3.4 Algorithms Design

3.4.1 Server Side Implementation

The Algorithm 1 shows the training procedure on the server, where the input is the pre-trained weight, clients' models list, and prototypes list. In the first round of communication, the server initializes its model by training on the labeled data set D^s with cross-entropy loss, after loading the transferred knowledge. Cross-Entropy Loss will leverage classifier outputs $\hat{y}$ and ground-truth y^s to adjust and improve the model's predictive accuracy. During training, the client will initialize the label disambiguation prototype.

As the algorithm progresses into subsequent rounds, the server's role becomes more collaborative and integrative. It starts by aggregating the models and prototypes received from the clients, combining these with its own model and prototype. This aggregation allows the server to assimilate diverse data insights from various client models, enhancing the overall learning and adaptability of the system. The server model and prototype undergo continuous refinement, incorporating new data and insights obtained from the client models. In the final stage, the server transmits the updated models and prototypes to selected clients.

3.4.2 Client Side Implementation

Algorithm 2 describes the training procedure within the client j, it begins with the reception of the server's prototype, model, and the specified number of communication rounds. Initially, the client aligns its prototype and model with those received from the server. For the unlabeled samples, the client employs both weak and strong augmentation techniques to produce two distinct versions of each input.

Algorithm 1 Pseudo Code of the FedALL (On Server)

Input:pre-trained weight w,uploaded clients' models list $[\theta_0, \ldots, \theta_j]$, upload client prototype list $[U_0, \ldots, U_j]$, server dataloader D^s, communication round t.
Output: updated model weight $\theta_0^{'}$, updated prototype $U_0^{'}$.
if $t > 1$ **then**
  $\theta_0^* \leftarrow$ Aggregate($[\theta_0, \ldots, \theta_j]$)
  $\mathcal{U}_0^* \leftarrow$ Aggregate($[\mathcal{U}_0, \ldots, \mathcal{U}_j]$)
else
  $\theta_0^* \leftarrow$ load w
end if
for server epoch = 1,2,... **do**
  for each x^s,y^s in D_s **do**
    $\hat{x^s} \leftarrow$ WeakAugmented(x^s)
    $\eta_{\hat{x}^s}^P$, $\hat{y} \leftarrow \theta_0(\hat{x}^s)$
    if $t == 1$ **then**
      $U_0^* \leftarrow$ Initialized the prototype with equation 3.2
    end if
    $U_0^* \leftarrow$ UpdatePrototype($U_0^*, \eta_{\hat{x}^s}^q$) with equation 3.4
    $\theta_0^* \; gets$ Cross-Entropy Loss($\hat{y}$,y^s)
  end for
end for
$U_0' \leftarrow U_0^*$
$\theta_0' \leftarrow \theta_0^*$
transmission θ_0', U_0' to selected Clints.

If training occurs during the first communication round, the client generates an adaptive label list y_j^a for each image in D^u. y_j^a is derived from the classifier's results on the weak augmented inputs, a procedure detailed in 3.2.3. Currently, the client updates the received prototype using the outputs of the projector. In subsequent communication rounds, the client continues to refine y_j^a for each image, following the methodology specified in Equation 3.5. If the label correction mechanism is activated, the least probable ground-truth label in y_j^a is replaced with a more probable one, and y_j^a is then reset.

Another step in this process involves the multiple check strategy, which selects the appropriate y_j^a to be used as a pseudo-label in the cross-entropy loss calculation. To facilitate model convergence, especially in the early stages where the selected y_j^a might be limited, the client applies SimSiam loss. This loss function incorporates both weakly and strongly augmented

samples. Finally, the client completes the cycle by sending the updated model and prototype back to the server, ensuring continuous learning and adaptation within the FedALL framework.

Algorithm 2 Pseudo Code of the FedALL(On Client)

Input: client dataset D_j^u, server prototype U_0', server model θ_0', communication round t.
Output: updated client prototype $U_j^{'}$, updated client weight $\theta_j^{'}$.
$U_j = U_0'$
$\theta_j = \theta_0'$
for client epoch = 1,2,..., **do**
  for x^u in D_u, **do**
    $\hat{x}^u =$ WeakAugmented(x^u)
    $\tilde{x}^u =$ StrongAugmented(x^u)
    predictor η_w^q, projector η_w^P, classifier $\hat{y} = \theta_j(\hat{x}^u)$
    predictor $\eta_s^q = \theta_j(\tilde{x}^u)$
    if $t == 1$ **then**
      Initialized y_j^a
    end if
    for $y_{j,i}^a \in y_j^a$, **do**
      update $y_{j,i}^a$ with 3.5
      if $y_{j,i}^a$ need correction **then**
        replace the label with the least ground-truth possible with the label with higher possibility and reset the $y_{j,i}^a$
      end if
    end for
    $U_j^* =$ UpdatePrototype(U_j, η^P) with equation 3.3
    selected $y_j^a =$ MultipleCheck(A_i, B_i, C_i) with equation 3.7
    calculate Cross-Entropy Loss$(\hat{y},$ selected $y_j^a)$ defined in equation 3.9
    calculate SimSiam Loss(η_w^q, η_s^q) defined in equation 3.10
    $Loss =$ Cross-Entropy Loss + SimSiam Loss
    get θ_j^* with $Loss$ back-propagation
  end for
end for
$\theta_j' \leftarrow \theta_j^*$
$U_j' \leftarrow U_j^*$
return θ_j', U_j' to server

CHAPTER 4

Experiment and Evaluation

This chapter presents a comprehensive evaluation and in-depth analysis of FedALL. In Section 4.1, we outline the key elements of our experimental framework, including the datasets used, the baseline models for comparison, and the hardware setup to train FedALL. Section 4.2 is dedicated to a detailed comparative study between FedALL and other baseline models in various scenarios. Additionally, this section delves into the essential parameters and pre-trained models utilized in FedALL, along with a comprehensive review of performance metrics and results. Section 4.3 offers a meticulous comparison of FedALL's accuracy against that of baseline models under different parameter settings, accompanied by visual representations of the algorithm's performance under diverse conditions. Section 4.4 focuses on the ablation study, investigating the individual impact of each component in the FedALL framework on the general accuracy of classification and exploring how the number of labels in the adaptive label list affects the accuracy of classification.

4.1 Experiment Setup

4.1.1 Datasets

The effectiveness and robustness of the FedALL framework in addressing various computer vision challenges were evaluated using three widely used benchmark datasets: CIFAR-10, CIFAR-100, and Mini-ImageNet [58, 104].

CIFAR-10: This dataset is a fundamental yet comprehensive collection for image classification tasks. It consists of 60,000 color images with a resolution of 32x32 pixels, categorically divided into 10 distinct labels, each label contributing 6,000 images. The CIFAR-10 labels represent a broad spectrum of everyday objects, making them an ideal testing ground for basic image recognition algorithms. The dataset is typically partitioned into a training set comprising 50,000 images and a test set of 10,000 images, allowing for effective training and validation of models.

CIFAR-100: Building upon the structure of CIFAR-10, CIFAR-100 presents a more challenging scenario due to its finer classification scheme. It contains the same number of images and resolution as CIFAR-10 but diversifies into 100 labels, each with 600 images. This higher granularity in classification makes it an excellent dataset for testing more sophisticated image recognition algorithms. The dataset also introduces the concept of 'superclasses', grouping the 100 labels into larger categories, which can be useful for hierarchical classification studies.

Mini-ImageNet: As a condensed version of the larger ImageNet database, Mini-ImageNet serves as a more complex and diverse dataset. It comprises 60,000 color images, each with a resolution of 224x224 pixels, distributed across 100 different labels. With 600 images per label, this dataset mirrors the complexity and variety of the real-world visual environment. The higher resolution of the images in Mini-ImageNet challenges the algorithms to maintain performance accuracy and efficiency, making it suitable for advanced computer vision tasks.

4.1.2 Baselines

To ensure an impartial assessment of the proposed FedALL framework, we utilize the ResNet9 and ResNet18 backbones, which have garnered extensive adoption in the literature. We then juxtapose their performance against various state-of-the-art benchmarks for comprehensive comparison. These benchmarks include the following.

FedMatch [50]: FedMatch utilizes the FedAvg-FixMatch approach with inter-client consistency and parameter decomposition for model training.

FedU [134]: FedU applies the divergence-aware predictor update technique and self-supervised

BYOL [39] for model training.

FedEMA [133]: FedEMA adopts an exponential moving average (EMA) strategy to adaptively update online client networks.

Orchestra [71]: Orchestra introduces a novel clustering-based technique for federated semi-supervised learning (FSSL).

SemiFL [28]: SemiFL employs a strategy that initially uses FixMatch [92] to generate a high-confidence dataset from the unlabeled data in each client. This dataset is then augmented by building a MixMatch inspired dataset [7]. Both datasets are used to train client models.

By comparing the performance of the FedALL framework against these benchmarks, we can assess its effectiveness and advancements in federated semi-supervised learning.

4.1.3 Hardware Configuration

The training is performed on a Lenovo ThinkStation Server, configured as follows:

- **CPU:** One AMD Ryzen Threadripper PRO 3945WX, comprising 12 cores.
- **GPU:** Two NVIDIA RTX A5000 GPUs; however, only one was used during training.
- **Hard Drive:** One 1TB Non-Volatile Memory Express (NVMe) SSD.
- **RAM:** Three 32GB Micron Technology DDR4.
- **Mainboard:** Built upon a Lenovo 1046 motherboard.

4.2 Experiment Methodology and Result Analysis

4.2.1 Parameter Setting and Pre-Training

In the scholarly examination of the CIFAR-10 and CIFAR-100 datasets, the ResNet-9 architecture is adopted for training both on the server and on the client. In contrast, when addressing

TABLE 4.1. Comparative Analysis of Federated Semi-Supervised Learning Approaches on CIFAR-10 Datasets Under IID and Non-IID Settings

CIFAR-10				
	IID		Non-IID	
Label Rate	0.01	0.1	0.01	0.1
FedU	69.54 ± 1.31	80.74 ± 1.03	62.14 ± 1.31	78.13 ± 1.06
FedEMA	71.74 ± 1.21	79.16 ± 0.92	68.59 ± 1.04	81.81 ± 1.03
Orchestra	76.91 ± 1.16	82.42 ± 0.86	74.23 ± 1.33	81.02 ± 1.02
FedMatch	78.21 ± 1.12	85.54 ± 0.56	77.12 ± 1.02	83.52 ± 0.46
SemiFL	80.23 ± 1.02	83.27 ± 0.72	78.12 ± 0.68	82.73 ± 1.23
FedALL	84.72 ± 0.92	86.12 ± 0.57	83.51 ± 0.60	85.90 ± 0.33

TABLE 4.2. Comparative Analysis of Federated Semi-Supervised Learning Approaches on CIFAR-100 and Mini-ImageNet Datasets Under IID and Non-IID Settings

	CIFAR-100		Mini-Imagenet	
	IID	Non-IID	IID	Non-IID
Label Rate	0.1	0.1	0.1	0.1
FedU	41.22 ± 1.41	41.69 ± 1.21	12.37 ± 1.20	11.85 ± 1.14
FedEMA	44.45 ± 1.10	42.26 ± 1.07	12.51 ± 1.28	12.61 ± 1.27
Orchestra	50.29 ± 1.16	47.97 ± 1.36	14.21 ± 1.12	13.18 ± 0.94
FedMatch	57.44 ± 0.86	55.28 ± 0.76	77.80 ± 1.06	76.50 ± 1.02
SemiFL	42.35 ± 0.75	41.75 ± 1.04	17.78 ± 0.92	16.52 ± 1.05
FedALL	60.56 ± 1.03	59.90 ± 1.36	81.72 ± 1.50	81.30 ± 1.46

the Mini-ImageNet dataset, the ResNet-18 architecture is preferred. It is noteworthy that while the server's training images are comprehensively labeled, those on the client side remain unlabeled. The distribution of the label rates can be referenced in Tables 4.1 and 4.2. The label rate value refers to the number of labeled samples on the server, and the rest of the samples are all unlabeled in the clients. For the preparatory phase of training that incorporates transfer learning, the model designated for FedALL, as well as other comparative baselines, is initially subjected to pre-training on the Mini-ImageNet dataset. Subsequently, these models were acquired on the CIFAR-10 and CIFAR-100 datasets. However, when focusing on the Mini-ImageNet dataset, a model that has undergone pre-training on ImageNet is implemented. The experimental parameters delineated in Table 4.1 and Table 4.2 specify that the client count is fixed at 100. Within each training round, a subset of 5 clients is chosen at random, the total communication rounds between the server and clients is 200, and the local training

epoch in clients is 5. Furthermore, the number of labels in the adaptive label list is 3 for CIFAR-10, while it is 10 for both CIFAR-100 and Mini-ImageNet. The value of γ in 3.3 is 0.005. The value of γ in 3.4 is 0.01, since the server contains a labeled dataset, we amplify the influence of the server prototype in the updates, reflecting its higher reliability due to the presence of labeled data. The value of α in 3.5 is 0.9. For label correction, we set the number of consecutive mismatches to 3. In other words, if the label selected by the classifier consecutively has a value of 0 in the adaptive label list three times, the label correction mechanism will be activated.

4.2.2 Experiment Result and Analysis

Based on the comprehensive analysis presented in Tables 4.1 and 4.2, it becomes clear that the FedALL approach has a significant edge over other Federated Semi-Supervised Learning (FSSL) methods in various datasets and settings. This benefit is especially prominent in situations characterized by a scarcity of labeled data, a prevalent obstacle encountered in practical applications in the real world. For example, in the CIFAR-10 dataset with a label rate of 0.01, FedALL shows a remarkable performance improvement of 4.49% compared to its closest competitor. This improvement is not only a testament to its efficacy but also highlights its ability to leverage minimal labeled data more effectively than other methods.

In a deeper analysis of the results presented in the tables, it is particularly noteworthy that, apart from FedMatch, other baseline methods such as FedU, FedEMA, Orchestra, and SemiFL, underperform on datasets with larger image sizes and more categories, like CIFAR-100 and Mini-ImageNet. The underlying reason for this phenomenon is that these baseline models fail to effectively utilize pre-trained knowledge. Notably, in the initial communication rounds, the loaded pre-trained models are quickly "forgotten," leading to slow convergence rates. This issue is especially prominent when deploying relatively shallow neural network architectures such as ResNet18.

In contrast, FedMatch, by assigning pseudo-labels to unlabeled data, manages to utilize pre-trained weights to a better extent. This strategy enables it to perform better with large image datasets compared to other baselines. However, FedALL still achieves the best results in handling large image datasets. The key to its superiority lies in its optimal use of pre-trained knowledge. A significant feature of FedALL is its use of an adaptive label list, which provides a clear advantage when dealing with datasets that have a large number of labels.

In the analysis of IID and Non-IID data settings, it is observable that FedALL's classification performance is the least impacted among the other baselines when dealing with different data distributions. This robustness can be attributed to the unique approach adopted by FedALL, where not only the model, but also the label disambiguation prototype, is transmitted between the client and the server.

In situations with Non-IID, the prototype can assist the client with better training. This is because it undergoes a momentum update on the server, integrating image features gathered from various clients. This integration includes a broader range of global image representations. The momentum update mechanism in FedALL is particularly effective, as it enables the model to retain and merge various features from different data distributions encountered by each client. This process results in a more comprehensive and representative prototype that improves the model's ability to generalize in various data scenarios.

Overall, FedALL achieves state-of-the-art results compared to all baselines in Tables 4.1 and 4.2.

4.3 Impact of Different Parameters

The performance analysis in this section will consider several distinct parameters, including the varying number of client selections per communication round, different total number of clients, different training epochs within clients and a range of Non-IID rates. We use

CIFAR10 and CIFAR100 with a label rate of 0.01 to test the performance of the model. This assessment serves to elucidate the versatility and effectiveness of FedALL under a wide variety of conditions and to compare it with other baselines.

4.3.1 Client Selection Number

The effect of diverse client participation rates on the model's efficiency was investigated, and the outcomes are depicted in Figures 4.1, 4.2, 4.3, and 4.4. These figures demonstrate the relationship between the quantity of clients engaged in each training cycle and the aggregate model performance. An evident trend is discernible, with model efficiency escalating as the count of clients per communication cycle rises. However, exceeding a predetermined limit, any further increase in client quantity leads to a decrement in the global model's efficacy, a trend particularly pronounced with the CIFAR100 dataset. This downturn could be associated with the server's integration of an overly large number of client models, predominantly educated on unlabeled data, potentially obstructing the global model's convergence.

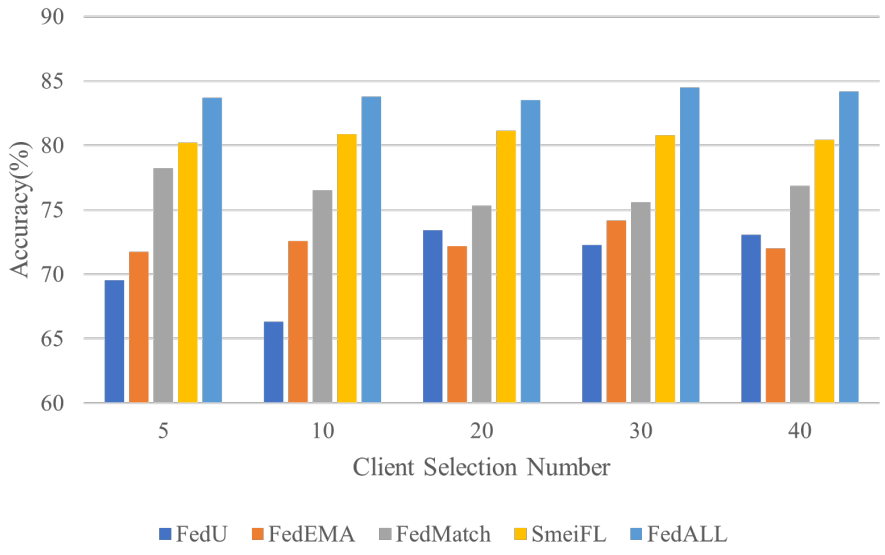

FIGURE 4.1. Sensitivity to different client selection numbers on CIFAR-10 IID.

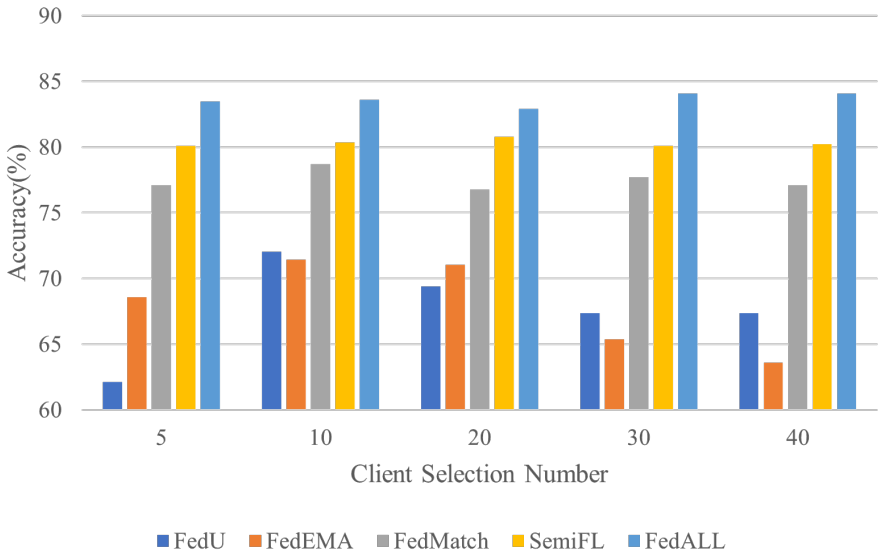

FIGURE 4.2. Sensitivity to different client selection numbers on CIFAR-10 Non-IID.

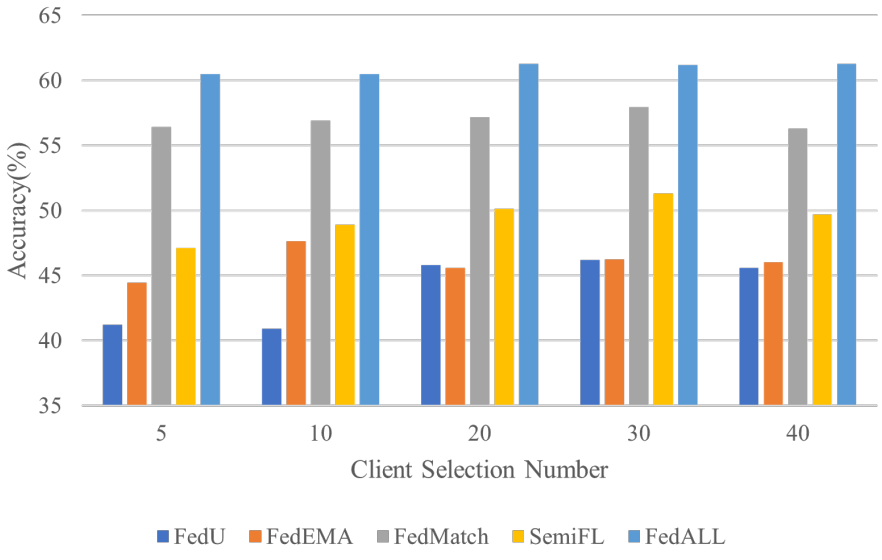

FIGURE 4.3. Sensitivity to different client selection numbers on CIFAR-100 IID.

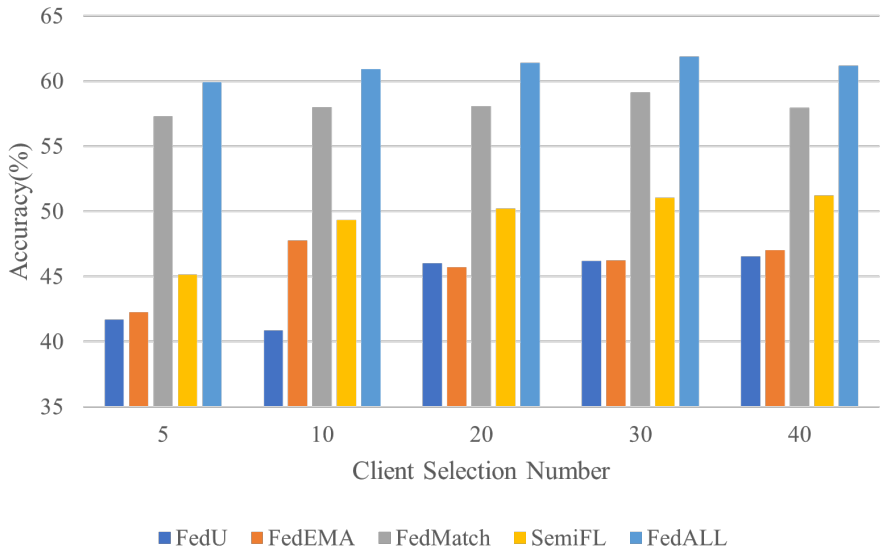

FIGURE 4.4. Sensitivity to different client selection numbers on CIFAR-100 Non-IID.

4.3.2 Local Epoch

The relationship between different training epochs in clients and global accuracy is illustrated in Figure 4.5, 4.6, 4.7, 4.8. The results indicate that as the number of local training epochs increases, global performance improves. This observation contrasts with other frameworks, where the global performance declines due to an excessive number of training epochs based on unlabeled data. This suggests that utilizing pre-trained models enables clients to better train on unlabeled data, subsequently providing superior model weights to the server. Consequently, performance enhancement corresponds to the increment in training epochs, emphasizing the benefits of incorporating pre-trained models in this context.

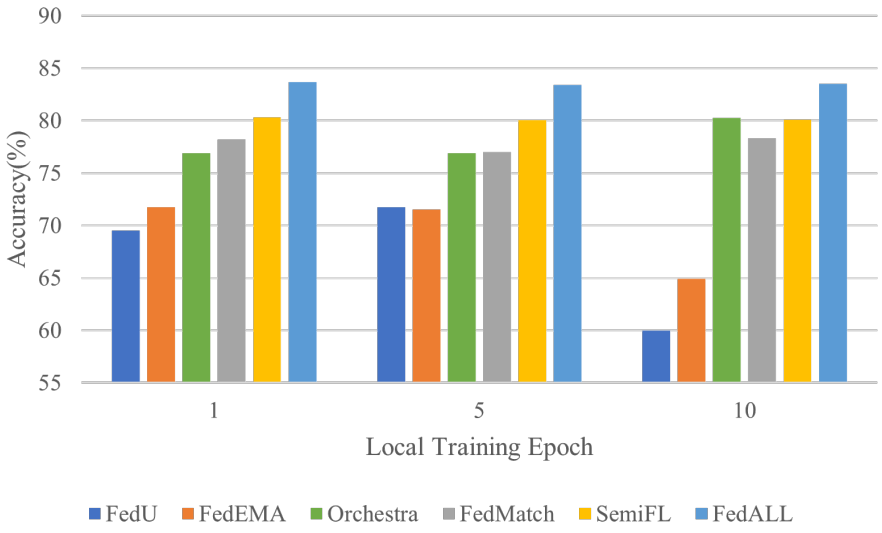

FIGURE 4.5. Sensitivity to different local epoch on CIFAR-10 IID.

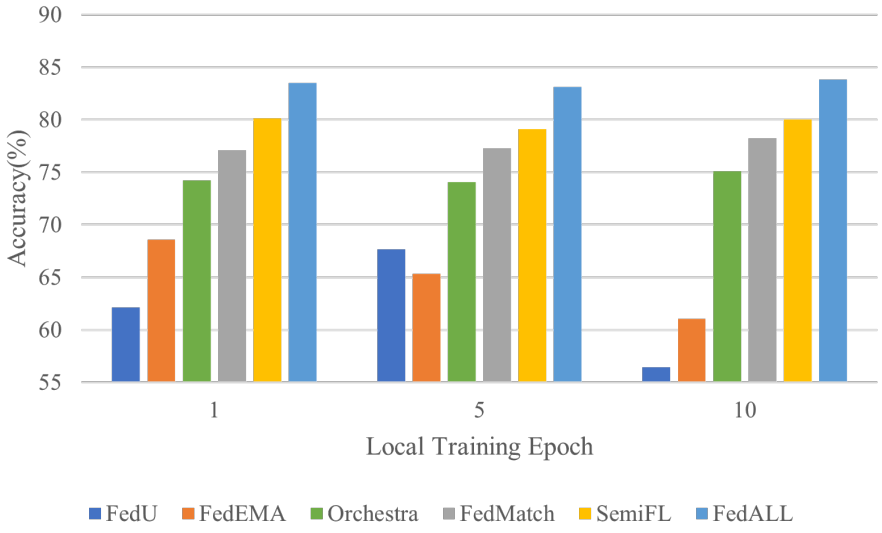

FIGURE 4.6. Sensitivity to different local epochs on CIFAR-10 Non-IID.

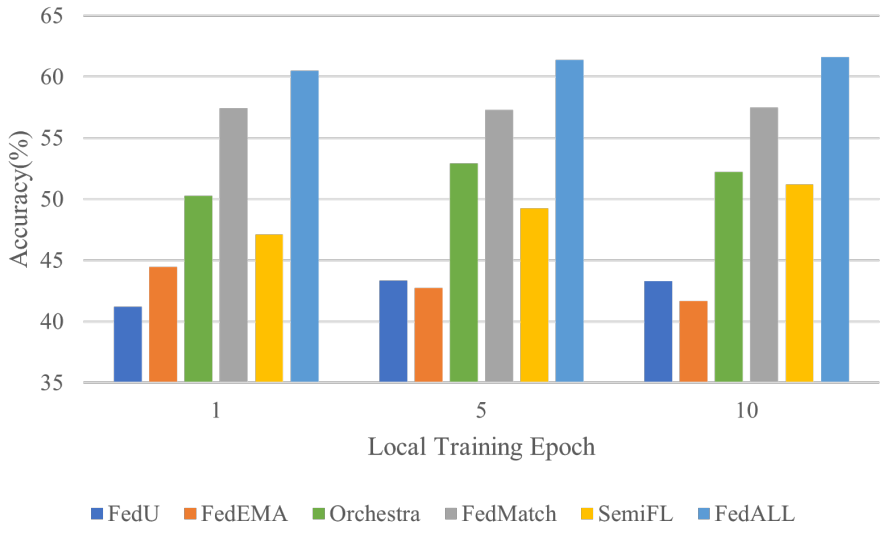

FIGURE 4.7. Sensitivity to different local epoch on CIFAR-100 IID.

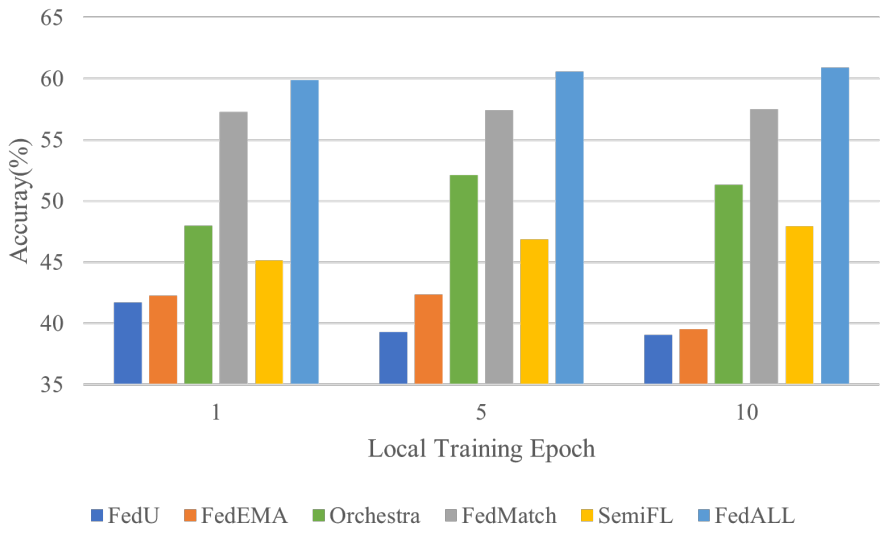

FIGURE 4.8. Sensitivity to different local epochs on CIFAR-100 Non-IID.

4.3.3 Total Client Number

In order to assess the impact of varying the total number of clients in federated learning, we adjust the number of local epochs in a linear fashion to maintain a consistent total

number of training iterations across all configurations. The result has been shown in Figures 4.9, 4.10, 4.11, 4.12.The results demonstrate that FedALL consistently outperforms other baseline approaches in different numbers of clients.

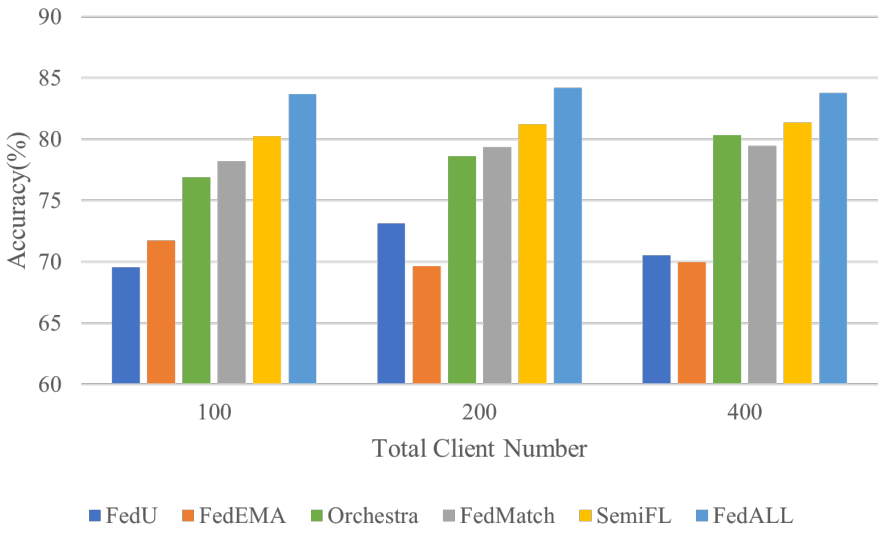

FIGURE 4.9. Sensitivity to different total client numbers on CIFAR-10 IID.

FIGURE 4.10. Sensitivity to different total client numbers on CIFAR-10 Non-IID.

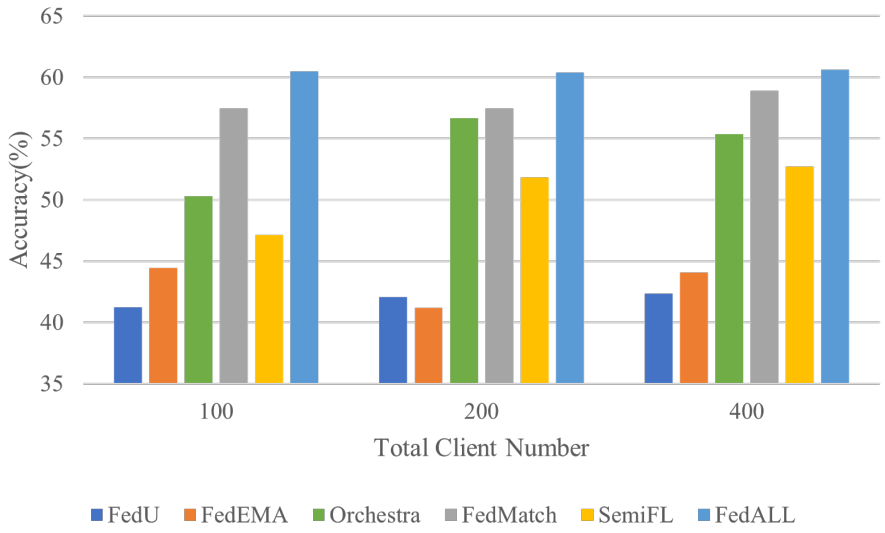

FIGURE 4.11. Sensitivity to different total client numbers on CIFAR-100 IID.

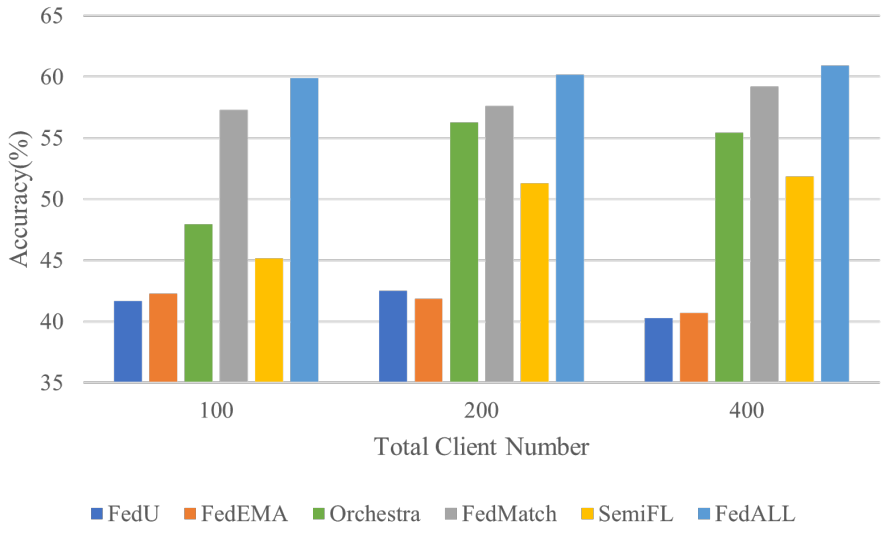

FIGURE 4.12. Sensitivity to different total client numbers on CIFAR-100 Non-IID.

4.3.4 Non-IID Rate

To consider the effect of changing the rate of the Non-IID dataset in federated learning, we evaluate FedALL with the other baselines under cifar10 and cifar100 as shown in Figure 4.13, 4.14. On observation, it is evident that as the Non-IID rate decreases, the accuracy of Orchestra, FedMatch, and FedALL correspondingly increases. However, FedALL consistently achieves better performance compared to the other baselines considered.

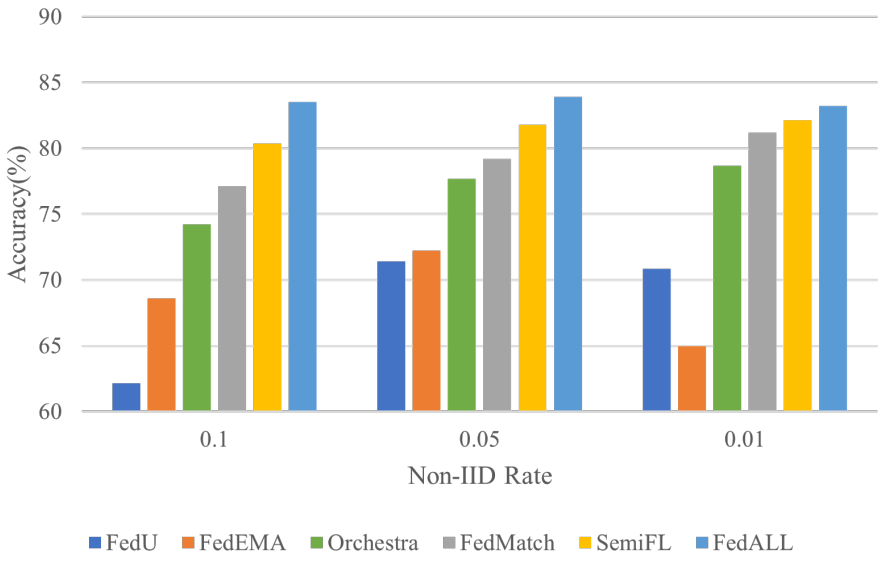

FIGURE 4.13. Sensitivity to different Non-IID rates on CIFAR-10.

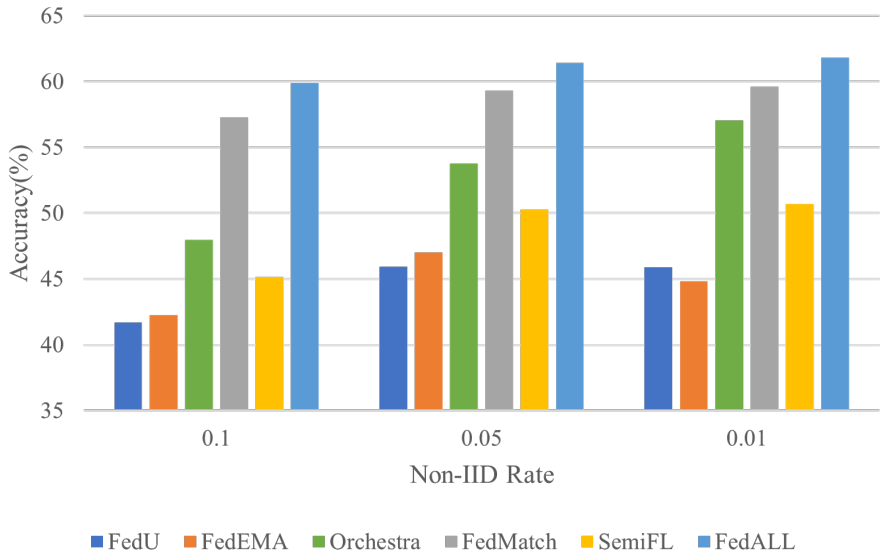

FIGURE 4.14. Sensitivity to different Non-IID rate on CIFAR-100.

4.4 Ablation Studies

4.4.1 Components Analysis

To gauge the importance of each module within FedALL, we conducted an ablation study by assessing the performance of FedALL on CIFAR10 with various combinations of modules. The results are shown in Table 4.3. Employing only the baseline SimSiam algorithm yields an accuracy of 77.93%, which increases to 79.32% upon inclusion of cross-entropy loss. Further integration of the Simsiam algorithm, cross-entropy loss, and adaptive label list leads to an accuracy of 80.83%, highlighting the efficacy of combining these components. Incorporation of the prototype mechanism results in the highest accuracy of 84.72%. These findings demonstrate the necessity and importance of each module used in the FedALL framework.

TABLE 4.3. Modules analysis in FedALL

Module	Accuracy
SimSiam	77.93 ± 1.04
SimSiam+cross-entropy	79.32 ± 0.83
SimSiam+cros-entropy+ALL	80.83 ± 1.03
SimSiam+cros-entropy+ALL+prototype	84.72 ± 0.92

4.4.2 Different Number of Chosen Labels in Adaptive Label List

The Table 4.4 demonstrates that the variation in the top number K in the adaptive label list does not significantly affect accuracy under CIFAR10. Although increasing the Top K number has a noticeable impact on the time required for client training. As the number of tops K increases, the time consumed for client training increases rapidly. Upon evaluating the trade-off between time consumption and accuracy, we have determined that selecting a top N of 3 offers an optimal balance in the CIFAR10 dataset for FedALL. This choice provides relatively high accuracy while maintaining reasonable time efficiency for client training.

TABLE 4.4. Different top-K numbers in ALL

Top N number in ALL	Accuracy
1	80.13
2	81.66
3	84.72
4	83.42
5	82.88
6	83.11
7	82.26
8	82.27
9	83.25
10	82.31

CHAPTER 5

Conclusion

In this thesis, we have explored the innovative Federated Adaptive Label Learning (FedALL) framework, highlighting its significant advancement in the realm of Federated Semi-supervised Learning (FSSL). FedALL, with its unique integration of transfer learning, representation learning, and partial label learning, stands out as a pioneering solution, addressing the intricate challenges inherent in the processing and analysis of large-scale, distributed, and partially labeled datasets. Our research has demonstrated the effectiveness of this approach, showcasing remarkable improvements in learning accuracy and robustness, particularly under the challenging conditions of non-IID data distributions.

The comprehensive analysis, both theoretical and experimental, of the FedALL framework underscores its potential in various real-world applications. By leveraging the Banach Fixed Point Theorem, we have not only validated the stability and convergence of our model but also set a theoretical benchmark for future research in this domain. Empirical evaluation of FedALL across multiple benchmark datasets has further solidified its position as a state-of-the-art methodology in FSSL.

Federated semi-supervised learning demonstrates immense potential for application in various fields. Although researchers have made some strides in this area, there are numerous potential solutions and avenues that remain to be explored. Drawing from our investigations into the FedALL framework, we posit that forthcoming research in this domain could explore the following avenues of inquiry.

- **Reducing Communication Cost:** Utilizing prototypes to lower communication load during data exchange between clients and servers. In federated learning, communication overhead is a critical issue, and using prototypes can effectively reduce the amount of data that needs to be transmitted, thereby enhancing efficiency [57].
- **Using More General Pre-trained Models:** Employing general pre-trained models to assist in training models for specific tasks. This approach can accelerate the training process and improve the performance of models in specific tasks, especially in scenarios where data is scarce [96].
- **Againsting Attacks in Federated Learning System:** Expanding the FedALL framework to include security mechanisms to identify potential malicious clients by analyzing the differences between client-uploaded prototypes and server prototypes. This involves anomaly detection, behavior analysis, consistency checks, and the establishment of a client trustworthiness assessment system to identify and protect against data contamination and malicious modifications in model updates [77].
- **Explainable Federated Learning** Explainability in federated learning is crucial for understanding model decisions, fostering trust, and facilitating model debugging and improvement, particularly in domains requiring high levels of transparency, such as healthcare and finance. By incorporating XAI methods, the FedALL framework can be enhanced to provide insights into the decision-making process of the model, elucidate the role of pseudo-labels and representation learning in predictions, and reveal how data from different sources contribute to the learning process. This advancement could lead to more transparent, trustworthy, and interpretable federated learning models, paving the way for broader adoption and more effective deployment in real-world applications where decision-making processes need to be understood and justified [19].

Bibliography

[1] Ahmed M Abdelmoniem et al. 'Refl: Resource-efficient federated learning'. In: *Proceedings of the Eighteenth European Conference on Computer Systems*. 2023, pp. 215–232.

[2] Alan Alexander et al. 'An intelligent future for medical imaging: a market outlook on artificial intelligence for medical imaging'. In: *Journal of the American College of Radiology* 17.1 (2020), pp. 165–170.

[3] Yoshinori Aono et al. 'Privacy-preserving deep learning via additively homomorphic encryption'. In: *IEEE transactions on information forensics and security* 13.5 (2017), pp. 1333–1345.

[4] Sean Augenstein et al. 'Generative models for effective ML on private, decentralized datasets'. In: *arXiv preprint arXiv:1911.06679* (2019).

[5] John E Ball, Derek T Anderson and Chee S Chan. 'Comprehensive survey of deep learning in remote sensing: Theories, tools, and challenges for the community'. In: *J. Appl. Remote Sens.* 11.4 (2017).

[6] Yoshua Bengio, Aaron Courville and Pascal Vincent. 'Representation learning: A review and new perspectives'. In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828.

[7] David Berthelot et al. 'Mixmatch: A holistic approach to semi-supervised learning'. In: *Neural Information Processing Systems* 32 (2019).

[8] David Berthelot et al. 'Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring'. In: *arXiv preprint arXiv:1911.09785* (2019).

[9] Abhishek Bhowmick et al. 'Protection against reconstruction and its applications in private federated learning'. In: *arXiv preprint arXiv:1812.00984* (2018).

[10] Steffen Bickel et al. ‘Multi-task learning for HIV therapy screening’. In: *Machine Learning* 73.1 (2009), pp. 55–83.

[11] Jacob Bien and Robert Tibshirani. ‘Prototype selection for interpretable classification’. In: *The Annals of Applied Statistics* (2011), pp. 2403–2424.

[12] Avrim Blum and Tom Mitchell. ‘Combining labeled and unlabeled data with co-training’. In: *Proceedings of the eleventh annual conference on Computational learning theory*. 1998, pp. 92–100.

[13] Keith Bonawitz et al. ‘Practical secure aggregation for federated learning on user-held data’. In: *arXiv preprint arXiv:1611.04482* (2016).

[14] Konstantinos Bousmalis et al. ‘Domain separation networks’. In: *Neural Information Processing Systems*. Vol. 29. 2016, pp. 343–351.

[15] Ran Canetti et al. ‘Adaptively secure multi-party computation’. In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 1996, pp. 639–648.

[16] Olivier Chapelle, Bernhard Schölkopf and Alexander Zien, eds. *Semi-Supervised Learning*. MIT Press, 2006.

[17] Olivier Chapelle, Bernhard Schölkopf and Alexander Zien. ‘Semi-supervised learning (Chap. 12)’. In: *Semi-Supervised Learning*. MIT Press, 2009, pp. 359–390.

[18] Jiangui Chen et al. ‘FedMatch: federated learning over heterogeneous question answering data’. In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2021, pp. 181–190.

[19] Peng Chen et al. ‘Evfl: An explainable vertical federated learning for data-oriented artificial intelligence systems’. In: *Journal of Systems Architecture* 126 (2022), p. 102474.

[20] Xinlei Chen and Kaiming He. ‘Exploring simple siamese representation learning’. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 15750–15758.

[21] Yae Jee Cho, Jianyu Wang and Gauri Joshi. ‘Client selection in federated learning: Convergence analysis and power-of-choice selection strategies’. In: *arXiv preprint arXiv:2010.01243* (2020).

[22] Yae Jee Cho et al. ‘Bandit-based communication-efficient client selection strategies for federated learning’. In: *2020 54th Asilomar Conference on Signals, Systems, and Computers*. IEEE. 2020, pp. 1066–1069.

[23] Marie Cottrell, Sabrina Ibbou and Patrick Letrémy. ‘SOM-based algorithms are related to LVQ algorithms’. In: *Neural Processing Letters* 20.3 (2004), pp. 207–217.

[24] Thomas Cover and Peter Hart. ‘Nearest neighbor pattern classification’. In: *IEEE Transactions on Information Theory* 13.1 (1967), pp. 21–27.

[25] Gabriela Csurka. ‘Domain adaptation for visual applications: A comprehensive survey’. In: *arXiv preprint arXiv:1702.05374* (2017).

[26] Wenyuan Dai et al. ‘Translated learning: Transfer learning across different feature spaces’. In: *Advances in neural information processing systems* 21 (2008).

[27] Oliver Day and Taghi M Khoshgoftaar. ‘A survey on heterogeneous transfer learning’. In: *Journal of Big Data* 4.1 (2017).

[28] Enmao Diao, Jie Ding and Vahid Tarokh. ‘SemiFL: Semi-supervised federated learning for unlabeled clients with alternate training’. In: *Neural Information Processing Systems* 35 (2022), pp. 17871–17884.

[29] Canh T Dinh et al. ‘Fedu: A unified framework for federated multi-task learning with laplacian regularization’. In: *arXiv preprint arXiv:2102.07148* 400 (2021).

[30] Carlotta Domeniconi, Jing Peng and Dimitrios Gunopulos. ‘Locally adaptive metric nearest-neighbor classification’. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.9 (2002), pp. 1281–1285.

[31] Cynthia Dwork. ‘Differential privacy’. In: *International colloquium on automata, languages, and programming*. Springer. 2006, pp. 1–12.

[32] Charles Elkan and Keith Noto. ‘Learning classifiers from only positive and unlabeled data’. In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2008, pp. 213–220.

[33] Yoav Freund and Robert E Schapire. ‘A decision-theoretic generalization of on-line learning and an application to boosting’. In: *Journal of computer and system sciences* 55.1 (1997), pp. 119–139.

[34] Yaroslav Ganin and Victor Lempitsky. ‘Unsupervised domain adaptation by backpropagation’. In: *International Conference on Machine Learning*. Vol. 37. 2015, pp. 1180–1189.

[35] Ian Goodfellow et al. ‘Generative adversarial nets’. In: *Neural Information Processing Systems*. Vol. 27. 2014, pp. 2672–2680.

[36] M Eshaghi Gordji et al. ‘On orthogonal sets and Banach fixed point theorem’. In: *Fixed Point Theory* 18.2 (2017), pp. 569–578.

[37] Ala Gouissem, Zina Chkirbene and Ridha Hamila. ‘A Comprehensive Survey on Client Selections in Federated Learning’. In: *arXiv preprint arXiv:2311.06801* (2023).

[38] Yves Grandvalet and Yoshua Bengio. ‘Semi-supervised Learning by Entropy Minimization’. In: *Neural Information Processing Systems*. 2005.

[39] Jean-Bastien Grill et al. ‘Bootstrap your own latent-a new approach to self-supervised learning’. In: *Neural Information Processing Systems* 33 (2020), pp. 21271–21284.

[40] Hao Guo, Jiyong Jin and Bin Liu. ‘Stochastic weight averaging revisited’. In: *Applied Sciences* 13.5 (2023), p. 2935.

[41] Yuanfan Guo et al. ‘Hcsc: Hierarchical contrastive selective coding’. In: *Computer Vision and Pattern Recognition Conference*. 2022, pp. 9706–9715.

[42] Meng Hao et al. ‘Efficient and privacy-enhanced federated learning for industrial artificial intelligence’. In: *IEEE Transactions on Industrial Informatics* 16.10 (2019), pp. 6532–6542.

[43] Meng Hao et al. ‘Towards efficient and privacy-preserving federated deep learning’. In: *ICC 2019-2019 IEEE international conference on communications (ICC)*. IEEE. 2019, pp. 1–6.

[44] Trevor Hastie and Robert Tibshirani. ‘Discriminant adaptive nearest neighbor classification’. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18.6 (1996), pp. 607–616.

[45] Geoffrey E Hinton, Simon Osindero and Yee-Whye Teh. ‘A fast learning algorithm for deep belief nets’. In: *Neural computation* 18.7 (2006), pp. 1527–1554.

[46] Briland Hitaj, Giuseppe Ateniese and Fernando Perez-Cruz. ‘Deep models under the GAN: information leakage from collaborative deep learning’. In: *Proceedings of*

the 2017 ACM SIGSAC conference on computer and communications security. 2017, pp. 603–618.

[47] Robert C Holte. 'Very simple classification rules perform well on most commonly used datasets'. In: *Machine Learning* 11.1 (1993), pp. 63–90.

[48] Piotr Indyk and Rajeev Motwani. 'Approximate nearest neighbors: Towards removing the curse of dimensionality'. In: *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*. ACM. 1998, pp. 604–613.

[49] Jiaul Islam and Yuefeng Zhang. 'Visual sentiment analysis for social images using transfer learning approach'. In: *Proc. IEEE Int. Conf. Big Data Cloud Comput.* 2016.

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

[51] Saumya Jetley et al. 'Prototypical priors: From improving classification to zero-shot learning'. In: *arXiv preprint arXiv:1512.01192* (2015).

[52] Zhanghan Ke et al. 'Dual student: Breaking the limits of the teacher in semi-supervised learning'. In: *Computer Vision and Pattern Recognition Conference*. 2019, pp. 6728–6736.

[53] Teuvo Kohonen. 'Improved versions of learning vector quantization'. In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. Vol. 1. 1990, pp. 545–550.

[54] Teuvo Kohonen. 'Learning Vector Quantization'. In: *The Handbook of Brain Theory and Neural Networks*. Ed. by Michael A. Arbib. MIT Press, 1995, pp. 537–540.

[55] Teuvo Kohonen. 'Learning Vector Quantization'. In: *Neural Networks* 10.3 (1995), pp. 509–535.

[56] Teuvo Kohonen. *Self-Organizing Maps*. Springer-Verlag, 2001.

[57] Jakub Konečnỳ et al. 'Federated learning: Strategies for improving communication efficiency'. In: *arXiv preprint arXiv:1610.05492* (2016).

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

[59] Samuli Laine and Timo Aila. ‘Temporal ensembling for semi-supervised learning’. In: *arXiv preprint arXiv:1610.02242* (2016).

[60] Syed Latif et al. ‘Transfer learning for improving speech emotion classification accuracy’. In: *arXiv preprint arXiv:1801.06353* (2018).

[61] Yann LeCun, Yoshua Bengio and Geoffrey Hinton. ‘Deep learning’. In: *Nature* 521.7553 (2015), pp. 436–444.

[62] Dong-Hyun Lee. ‘Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks’. In: *Workshop on Challenges in Representation Learning, International Conference on Machine Learning*. 2013.

[63] Gen Li et al. ‘Adaptive prototype learning and allocation for few-shot segmentation’. In: *Computer Vision and Pattern Recognition Conference*. 2021, pp. 8334–8343.

[64] Junnan Li et al. ‘Prototypical contrastive learning of unsupervised representations’. In: *International Conference on Learning Representations*. 2021.

[65] Xiang Li et al. ‘On the convergence of fedavg on non-iid data’. In: *arXiv preprint arXiv:1907.02189* (2019).

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

[67] Sen Lin, Guang Yang and Junshan Zhang. ‘A collaborative learning framework via federated meta-learning’. In: *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*. IEEE. 2020, pp. 289–299.

[68] Xinyang Lin et al. ‘Federated learning with positive and unlabeled data’. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 13344–13355.

[69] Bing Liu et al. ‘Building text classifiers using positive and unlabeled examples’. In: *Proceedings of the Third IEEE International Conference on Data Mining*. IEEE. 2003, pp. 179–186.

[70] Mingsheng Long et al. ‘Deep transfer learning with joint adaptation networks’. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. 2017, pp. 2208–2217.

[71] Ekdeep Singh Lubana et al. ‘Orchestra: Unsupervised federated learning via globally consistent clustering’. In: *arXiv preprint arXiv:2205.11506* (2022).

[72] Laurens van der Maaten and Geoffrey Hinton. ‘Visualizing data using t-SNE’. In: *Journal of Machine Learning Research* 9.Nov (2008), pp. 2579–2605.

[73] Brendan McMahan et al. ‘Communication-efficient learning of deep networks from decentralized data’. In: *Artificial intelligence and statistics*. PMLR. 2017, pp. 1273–1282.

[74] Luca Melis et al. ‘Exploiting unintended feature leakage in collaborative learning’. In: *2019 IEEE symposium on security and privacy (SP)*. IEEE. 2019, pp. 691–706.

[75] Pascal Mettes, Elise van der Pol and Cees Snoek. ‘Hyperspherical prototype networks’. In: *Neural Information Processing Systems* 32 (2019).

[76] Takeru Miyato et al. ‘Virtual adversarial training: a regularization method for supervised and semi-supervised learning’. In: *IEEE transactions on pattern analysis and machine intelligence* 41.8 (2018), pp. 1979–1993.

[77] Viraaji Mothukuri et al. ‘A survey on security and privacy of federated learning’. In: *Future Generation Computer Systems* 115 (2021), pp. 619–640.

[78] Milad Nasr, Reza Shokri and Amir Houmansadr. ‘Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning’. In: *2019 IEEE symposium on security and privacy (SP)*. IEEE. 2019, pp. 739–753.

[79] Shuteng Niu et al. ‘A Decade Survey of Transfer Learning (2010–2020)’. In: *IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE* 1.2 (2020).

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

[81] Rajat Raina et al. ‘Self-taught learning: Transfer learning from unlabeled data’. In: *Proceedings of the 24th International Conference on Machine Learning*. 2007.

[82] Antti Rasmus et al. ‘Semi-supervised learning with ladder networks’. In: *Neural Information Processing Systems* 28 (2015).

[83] Monica Ribero and Haris Vikalo. ‘Communication-efficient federated learning via optimal client sampling’. In: *arXiv preprint arXiv:2007.15197* (2020).

[84] Kuniaki Saito et al. ‘Adversarial dropout regularization’. In: *arXiv preprint arXiv:1711.01575* (2017).

[85] Mehdi SM Sajjadi, Mehran Javanmardi and Tolga Tasdizen. 'Regularization with stochastic transformations and perturbations for deep semi-supervised learning'. In: *Neural Information Processing Systems*. Vol. 29. 2016.

[86] Maomi Sato and Keiji Yamada. 'Generalized Learning Vector Quantization'. In: *Neural Information Processing Systems*. 1996, pp. 423–429.

[87] Petra Schneider, Michael Biehl and Barbara Hammer. 'Adaptive Relevance Matrices in Learning Vector Quantization'. In: *Neural Computation* 21.12 (2009), pp. 3532–3561.

[88] Rulin Shao et al. 'Stochastic channel-based federated learning for medical data privacy preserving'. In: *arXiv preprint arXiv:1910.11160* (2019).

[89] Hoo-Chang Shin et al. 'Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning'. In: *IEEE transactions on medical imaging* 35.5 (2016), pp. 1285–1298.

[90] Connor Shorten and Taghi M Khoshgoftaar. 'A survey on feature selection techniques'. In: *Big Data Analytics* 4.1 (2019), pp. 1–20.

[91] Jake Snell, Kevin Swersky and Richard Zemel. 'Prototypical networks for few-shot learning'. In: *Neural Information Processing Systems*. 2017.

[92] Kihyuk Sohn et al. 'Fixmatch: Simplifying semi-supervised learning with consistency and confidence'. In: *Neural Information Processing Systems* 33 (2020), pp. 596–608.

[93] Ahmet Ali Süzen and Mehmet Ali Şimşek. 'A novel approach to machine learning application to protection privacy data in healthcare: Federated learning'. In: *Namık Kemal Tıp Dergisi* 8.1 (2020), pp. 22–30.

[94] Cheng Tan et al. 'Survey on deep transfer learning'. In: *Proceedings of the International Conference on Artificial Neural Networks*. 2018.

[95] Cheng Tan et al. 'Survey on deep transfer learning'. In: *Proceedings of the International Conference on Artificial Neural Networks*. 2018.

[96] Yue Tan et al. 'Federated learning from pre-trained models: A contrastive learning approach'. In: *Neural Information Processing Systems* 35 (2022), pp. 19332–19344.

[97] Yue Tan et al. ‘Fedproto: Federated prototype learning across heterogeneous clients’. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 8. 2022, pp. 8432–8440.

[98] Antti Tarvainen and Harri Valpola. ‘Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results’. In: *Neural Information Processing Systems* 30 (2017).

[99] Yonglong Tian et al. ‘Rethinking few-shot image classification: a good embedding is all you need?’ In: *European Conference on Computer Vision*. 2020.

[100] Stacey Truex et al. ‘A hybrid approach to privacy-preserving federated learning’. In: *Proceedings of the 12th ACM workshop on artificial intelligence and security*. 2019, pp. 1–11.

[101] Stacey Truex et al. ‘Demystifying membership inference attacks in machine learning as a service’. In: *IEEE Transactions on Services Computing* 14.6 (2019), pp. 2073–2089.

[102] Eric Tzeng et al. ‘Adversarial discriminative domain adaptation’. In: *Computer Vision and Pattern Recognition Conference*. 2017, pp. 7167–7176.

[103] Vikas Verma et al. ‘Interpolation consistency training for semi-supervised learning’. In: *Neural Networks* 145 (2022), pp. 90–106.

[104] Oriol Vinyals et al. ‘Matching networks for one shot learning’. In: *Neural Information Processing Systems* 29 (2016).

[105] Lucas H Vogado et al. ‘Leukemia diagnosis in blood slides using transfer learning in CNNs and SVM for classification’. In: *Engineering Applications of Artificial Intelligence* 72 (2018).

[106] Haobo Wang et al. ‘Pico: Contrastive label disambiguation for partial label learning’. In: *International Conference on Learning Representations*. 2021.

[107] Hongyi Wang et al. ‘RSCFed: Label-Free Federated Learning with Robustness to Data Heterogeneity’. In: *arXiv preprint arXiv:2007.14390* (2020).

[108] Kaixin Wang et al. ‘Panet: Few-shot image semantic segmentation with prototype alignment’. In: *Computer Vision and Pattern Recognition Conference*. 2019, pp. 9197–9206.

[109] Zhibo Wang et al. ‘Beyond inferring class representatives: User-level privacy leakage from federated learning’. In: *IEEE INFOCOM 2019-IEEE conference on computer communications*. IEEE. 2019, pp. 2512–2520.

[110] Zhiting Wang et al. ‘Characterizing and avoiding negative transfer’. In: *computing research repository* abs/1811.09751 (2019).

[111] Kang Wei et al. ‘Federated learning with differential privacy: Algorithms and performance analysis’. In: *IEEE Transactions on Information Forensics and Security* 15 (2020), pp. 3454–3469.

[112] Kevin Weiss, Taghi M Khoshgoftaar and DingDing Wang. ‘A survey of transfer learning’. In: *Journal of Big Data* 3.1 (2016).

[113] Zhirong Wu et al. ‘Unsupervised feature learning via non-parametric instance discrimination’. In: *Computer Vision and Pattern Recognition Conference*. 2018, pp. 3733–3742.

[114] Liyang Xie et al. ‘Differentially private generative adversarial network’. In: *arXiv preprint arXiv:1802.06739* (2018).

[115] Qizhe Xie et al. ‘Unsupervised data augmentation for consistency training’. In: *Neural Information Processing Systems* 33 (2020), pp. 6256–6268.

[116] Jing Xu et al. ‘Fedcm: Federated learning with client-level momentum’. In: *arXiv preprint arXiv:2106.10874* (2021).

[117] Wenjia Xu et al. ‘Attribute prototype network for zero-shot learning’. In: *Neural Information Processing Systems*. 2020, pp. 21969–21980.

[118] Hong-Ming Yang et al. ‘Robust classification with convolutional prototype learning’. In: *Computer Vision and Pattern Recognition Conference*. 2018, pp. 3474–3482.

[119] Qiang Yang et al. ‘Federated machine learning: Concept and applications’. In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 10.2 (2019), pp. 1–19.

[120] Mang Ye et al. ‘Augmentation invariant and instance spreading feature for softmax embedding’. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[121] Mang Ye et al. ‘Unsupervised embedding learning via invariant and spreading instance feature’. In: *Computer Vision and Pattern Recognition Conference*. 2019, pp. 6210–6219.

[122] Jaehong Yoon, Daniel Jarrett and Mihaela van der Schaar. ‘AdaFedSemi: Adaptive Federated Semi-Supervised Learning’. In: *arXiv preprint arXiv:2102.08570* (2021).

[123] Jason Yosinski et al. ‘How transferable are features in deep neural networks?’ In: *Neural Information Processing Systems*. Vol. 27. 2014, pp. 3320–3328.

[124] Han Yu et al. ‘A fairness-aware incentive scheme for federated learning’. In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 2020, pp. 393–399.

[125] Bowen Zhang et al. ‘Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling’. In: *Neural Information Processing Systems* 34 (2021), pp. 18408–18419.

[126] Jiawei Zhang, Tianchi Yu and Qiang Yang. ‘FedSSL: Federated Semi-Supervised Learning with Shared Labels’. In: *arXiv preprint arXiv:2102.06976* (2021).

[127] Jie Zhang et al. ‘Federated Semi-Supervised Learning with Inter-Client Consistency & Disjoint Learning’. In: *arXiv preprint arXiv:2006.12097* (2020).

[128] Joey Tianyi Zhou et al. ‘Transfer learning for cross-language text categorization through active correspondences construction’. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2016.

[129] Tianfei Zhou et al. ‘Rethinking semantic segmentation: A prototype view’. In: *Computer Vision and Pattern Recognition Conference*. 2022.

[130] Xiaojin Zhu. *Semi-supervised learning literature survey*. Tech. rep. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

[131] Yin Zhu et al. ‘Heterogeneous transfer learning for image classification’. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 25. 1. 2011, pp. 1304–1309.

[132] Fengxiang Zhuang et al. ‘A comprehensive survey on transfer learning’. In: *arXiv preprint arXiv:1911.02685* (2019).

[133] Weiming Zhuang, Yonggang Wen and Shuai Zhang. ‘Divergence-aware federated self-supervised learning’. In: *arXiv preprint arXiv:2204.04385* (2022).

[134] Weiming Zhuang et al. ‘Collaborative unsupervised visual representation learning from decentralized data’. In: *Computer Vision and Pattern Recognition Conference*. 2021, pp. 4912–4921.

[135] Hankz Hankui Zhuo et al. ‘Federated deep reinforcement learning’. In: *arXiv preprint arXiv:1901.08277* (2019).