Federated Learning with Generalization to New Domains

By Milad Soltany

A thesis submitted to the Graduate Program in Electrical And Computer Engineering in conformity with the requirements for the Degree of Master of Applied Science

Queen's University
Kingston, Ontario, Canada
November, 2024

Abstract

FL is an area of research that focuses on training machine learning models in a decentralized fashion without having the need to store all data on one central server. In this thesis, we address the challenges of data heterogeneity and label scarcity in FL by proposing two novel approaches for federated domain generalization in both unsupervised and supervised settings.

First, to tackle federated domain generalization in an unsupervised setting, we introduce Federated Unsupervised Domain Generalization using Global and Local Alignment of Gradients. We establish a connection between domain shifts and gradient alignment in unsupervised federated learning, demonstrating that aligning gradients at both the client and server levels facilitates the generalization of the model to new, unseen domains. FedGaLA performs gradient alignment locally to encourage clients to learn domain-invariant features, and globally at the server to obtain a more generalized aggregated model. Extensive experiments on four multi-domain datasets—PACS, OfficeHome, DomainNet, and TerraInc—show that FedGaLA outperforms comparable baselines. Ablation and sensitivity studies highlight the impact of different components and hyper-parameters in our approach.

Second, to address data heterogeneity in a supervised federated learning framework, we propose Federated Domain Generalization with Label Smoothing and Balanced Decentralized Training (FedSB). FedSB utilizes label smoothing at the client level to prevent overfitting to domain-specific features, thereby enhancing generalization capabilities across diverse domains when aggregating local models into a global model. Additionally, FedSB incorporates a decentralized budgeting mechanism that balances training among clients, improving the performance of the aggregated global model. Experiments on four commonly used multi-domain datasets—PACS, VLCS, OfficeHome, and TerraInc—demonstrate that FedSB outperforms competing methods, achieving state-of-the-art results on three out of four datasets.

Collectively, these contributions address critical challenges in FL by enhancing model generalization across diverse and unseen domains in both unsupervised and supervised settings. The effectiveness of FedGaLA and FedSB in addressing data heterogeneity is evidenced by their superior performance in extensive empirical evaluations.

Acknowledgments

I would like to express my deepest gratitude to all those who supported me along this journey. I am especially thankful to my supervisors, Dr. Ali Etemad and Dr. Michael Greenspan, for their invaluable guidance, mentorship, and unwavering support throughout the course of this research. I am also incredibly grateful to my collaborators, Farhad Pourpanah and Mahdiyar Molahasani, for their contributions and insight. A heartfelt thanks to my parents, my brother Matin, and my partner Niusha, whose constant support and encouragement have been my greatest source of motivation. I would also like to acknowledge my friends and lab mates for their constant support during the more challenging moments of this journey.

Statement Of Originality

The following work described is my own and I hereby certify the intellectual content of this thesis is the product of my own work. To the best of my knowledge, all references and contributions of other individuals has been properly cited and sourced appropriately.

Contents

List of Tables

List of Figures

Glossary of Abbreviations

DP Differential Privacy. 10

FDG Federated Domain Generalization. 3, 6, 7, 16, 42, 50, 57

FL Federated Learning. i, ii, vi, 1–3, 5–10, 16, 19, 41

FUDG Federated Unsupervised Domain Generalization. 6, 16, 17

HE Homomorphic Encryption. 10

IID Independent and Identically Distributed. 8, 38

MLP Multilayer perceptron. 25, 48

Chapter 1

Introduction

Federated Learning (FL) [1, 2] has emerged as a powerful paradigm for training machine learning models across multiple decentralized clients while preserving data privacy and security. Unlike traditional centralized machine learning approaches that require aggregating data from various sources into a single repository, FL enables collaborative training of a global model without the need to exchange sensitive local data. Each client trains a local model using its private data, and a central server aggregates these models periodically to form a unified global model [3, 4].

Despite its advantages, FL faces significant challenges due to data heterogeneity and the scarcity of labeled data across clients. Data collected by each client often originates from different domains or distributions, leading to substantial domain shifts [5]. Additionally, labeling data is inherently challenging and resource-intensive, especially in federated settings where labeling efforts are decentralized and may be inconsistent across clients. These challenges hinder the global model's ability to generalize effectively to unseen target domains.

This thesis aims to address these challenges by proposing novel methodologies that enhance the generalization capabilities of FL models in both unsupervised and

supervised settings. Specifically, we introduce two innovative frameworks: **FedGaLA** (Federated Unsupervised Domain Generalization using Global and Local Alignment of Gradients) for the unsupervised scenario, and **FedSB** (Federated Domain Generalization with Label Smoothing and Balanced Decentralized Training) for the supervised scenario. These methodologies are designed to learn domain-invariant representations and ensure balanced training contributions from all clients, thereby improving the global model's performance across diverse and unseen domains.

1.1 Motivation

The motivation behind this research stems from the pressing need to develop FL models that can generalize well in real-world scenarios characterized by data heterogeneity and limited labeled data. In many practical applications, such as healthcare, wearable devices, and autonomous systems, data collected by different clients or devices exhibit significant variations due to differences in environments, user behaviors, or sensor characteristics. These domain shifts pose a substantial challenge for federated models aiming to perform reliably across all clients. Furthermore, the scarcity of labeled data exacerbates the problem, as collecting and annotating data is both time-consuming and expensive. In federated settings, this issue is amplified since labeling efforts are distributed among clients who may lack the resources or expertise to label data accurately.

By addressing these challenges, this research has the potential to significantly impact various industries by enabling the development of more robust and generalizable FL models. Enhanced generalization capabilities would lead to improved performance

in critical applications like medical diagnosis, personalized recommendations, and intelligent monitoring systems, ultimately benefiting society by providing more reliable and privacy-preserving machine learning solutions.

The goal of this thesis is to demonstrate novel methodologies that advance the state-of-the-art in federated domain generalization, introducing innovative techniques for both unsupervised and supervised learning scenarios. These methodologies aim to overcome the limitations of existing approaches by effectively handling data heterogeneity and label scarcity, thereby paving the way for the next generation of FL technologies.

1.2 Problem Statement

Despite the recent advancements in FL, it still faces significant challenges in effectively handling domain generalization and managing the varying characteristics of data across clients. We tackle the following problems in this thesis:

Problem 1. One of the primary challenges lies in Federated Domain Generalization (FDG), which aims to enhance the global model's ability to generalize across diverse domains by learning domain-invariant features. Existing methods have made significant progress in this area, as demonstrated in the literature, but they typically rely on supervised settings where labeled data is abundant. In real-world scenarios, however, the availability of labeled data is often limited or non-existent. This challenge has led to the emergence of federated unsupervised learning, which focuses on learning meaningful representations from unlabeled data.

Despite advances in self-supervised techniques, current methods do not adequately unify domain generalization and unsupervised learning into a single framework. As

a result, the interplay between data heterogeneity and the scarcity of labeled data remains underexplored in federated settings. This gap calls for a comprehensive solution that simultaneously addresses the lack of labels while improving the global model's ability to generalize across different domains.

Problem 2. Another significant challenge is the tendency of local models in supervised federated domain generalization methods to overfit to their domain-specific data. As clients focus on their own datasets, they often become overconfident in learning domain-specific features, which limits the global model's ability to generalize across domains. This overconfidence in local models reduces the effectiveness of the aggregated global model, especially in heterogeneous data environments.

In addition to overconfidence, existing methods face challenges in ensuring balanced training contributions from clients. Variations in data distribution and sample sizes across clients can lead to imbalanced contributions, where clients with larger datasets disproportionately influence the global model. This imbalance results in biased or suboptimal models that do not generalize well across all clients. Moreover, privacy concerns persist, as some methods require sharing information that could leak sensitive data, undermining the privacy-preserving nature of federated learning. These limitations underscore the need for novel approaches that can mitigate local overfitting, ensure fair contributions across clients, and maintain privacy-preserving guarantees.

1.3 Solution Overview and Contributions

The main contributions of this thesis lie in the development of novel methodologies that address key challenges in FL. First, we introduce FedGaLA, a pioneering framework that unifies federated domain generalization and unsupervised learning, creating a new problem category known as federated unsupervised domain generalization. FedGaLA leverages global and local gradient alignment to learn domain-invariant representations from unlabeled data while preserving privacy.

Additionally, we present FedSB, an innovative technique for supervised federated domain generalization. FedSB mitigates overconfidence in local models through label smoothing and ensures balanced decentralized training with a novel budgeting mechanism that accounts for clients' varying data volumes. These methodologies are evaluated through extensive experimental validation, ablation studies, and sensitivity analyses across multiple benchmark datasets, demonstrating the effectiveness of the proposed solutions. The results achieve state-of-the-art performance in both unsupervised and supervised federated domain generalization. Furthermore, this work advances FL technologies by addressing the limitations of existing methods, paving the way for a new generation of robust, generalizable, and privacy-preserving FL methodologies.

1.4 Publications

- [Chapter 3] Farhad Pourpanah*, Mahdiyar Molahasani*, **Milad Soltany***, Michael Greenspan, Ali Etemad, "Federated Unsupervised Domain Generalization using Global and Local Alignment of Gradients", *Under review at AAAI 2025*

- [Chapter 4] **Milad Soltany***, Farhad Pourpanah*, Mahdiyar Molahasani*, Michael Greenspan, Ali Etemad, "FedSB: Federated Domain Generalization with Label Smoothing and Balanced Decentralized Training", *Under review at ICASSP 2025*

1.5 Thesis Outline

The remainder of this thesis is organized as follows. **Chapter 2** presents a comprehensive review of the related work on FL. Section 2.1 presents a comprehensive background on FL and its recent advancements. In Section 2.2, we give details about the FDG problem, before discussing Federated Unsupervised Learning, and Federated Unsupervised Domain Generalization (FUDG) in Sections 2.3, and 2.4.

In **Chapter 3**, we present our work on FUDG by introducing FedGaLA. In Section 3.1 we formulate the problem we are tackling as well as discussing local and global gradient alignment. In Section 3.2, we discuss the experimental setup that we used to demonstrate the effectiveness of our method and in Section 3.3, we discuss the performance of our proposed method along with extensive experiments that demonstrate the effectiveness of our approach.

Chapter 4 presents our work on Supervised FDG, in which we propose FedSB, tackling the problem of data heterogeneity and model overconfidence. In Section 4.1, we give details of the formulation for FDG and provide details of our proposed solution. Section 4.3 presents details on the datasets as well as the network architecture as well as choice of evaluation methods. Furthermore, in Section 4.3, we provide the results of our extensive experiments on various datasets.

Chapter 2

Background

In this chapter, we give details about the related work on FL and FDG, including some unsupervised works. We discuss recent advancements as well as the baselines that we compare against.

2.1 Federated Learning

FL is a distributed machine learning paradigm that enables multiple clients, such as devices or organizations, to collaboratively train a global model without sharing their raw data, thereby preserving data privacy and security [1, 2, 6]. In the seminal work of FedAvg [1], local models are iteratively trained on client data and then aggregated on a central server using a weighted averaging scheme to form a global model. Enhancing the capabilities and performance of FL in various aspects has gained a great deal of interest in the research community.

Different taxonomies have been proposed for FL. According to [7], recent advancements in FL can be classified into the following categories: a) Aggregation optimization, b) Heterogeneous FL, c) Secure FL, and d) Fair FL.

2.1.1 Aggregation Optimization

Since multiple clients participate in a federated training setup, it is crucial to have mechanisms that effectively aggregate the local models from these clients into a single global model. FedAVG is the oldest and most commonly-used aggregation method which produces a global model in which the parameters are a weighted average of the parameters of the local client models.

2.1.2 Heterogeneous FL

In a federated setting, where clients can differ with respect to their data, computational power, and training strategies, it is essential to develop mechanisms that account for these heterogeneities. These variations are often categorized as data, system, and model heterogeneity, and they can pose challenges to model performance and convergence. Data heterogeneity refers to differences in data distribution across clients, where some may have imbalanced or non-Independent and Identically Distributed (IID) data [8, 9]. System heterogeneity is due to clients having diverse hardware capabilities, network conditions, and resource constraints, which can impact their ability to train models efficiently or participate in communication rounds [10,11]. This is very common in real-world applications given that participating clients could be extremely different. Model heterogeneity occurs when clients use different model architectures or training strategies due to varying local requirements or limitations [12, 13].

2.1.3 Secure FL

FL was designed to enhance security by sharing model parameters instead of raw data, which remains stored locally on client devices. However, adversaries can extract private information from the shared models, highlighting the need for stronger security mechanisms in FL. Different types of attacks include backdoor attacks [14], gradient attacks [15], and model poisoning attacks [16]. In backdoor attacks, adversaries manipulate a subset of the training data to cause false predictions by the model on test data. Distributed Backdoor Attacks, introduced in [17], exploit the distributed nature of federated models by injecting local trigger patterns and creating a global trigger pattern in the global model. Other works, like Neurotoxin [18], further enhance backdoor attacks by targeting model parameters that are less likely to change during training, making these attacks more persistent.

Another type of attack is model poisoning, where the adversary attempts to manipulate the global model into producing incorrect outputs. One version is data poisoning attacks [16], in which the adversary corrupts the labels of a portion of the client's local training data to distort the global model's predictions. However, it is important to note that, due to the typically large number of clients in a federated setup, the impact of model poisoning attacks is often mitigated during the aggregation process.

In FL setups, gradients are often exchanged between clients and the server during the aggregation process, which introduces a vulnerability to a new type of attack known as gradient attacks. These attacks make it possible to reconstruct private training data from the publicly shared gradients [15]. Research by Zhu et al. in [19]

demonstrated how training data can be recovered from shared gradients by introducing the Recursive Gradient Attack on Privacy.

To defend federated systems from these different attacks, various mechanisms have been proposed. Mechanisms based on Differential Privacy (DP) [20] work by adding controlled noise to the information shared among clients, including gradients, model weights, and features. This noise helps obscure sensitive data while allowing the overall learning process to continue effectively [21–23]. Another type of defense is based on Homomorphic Encryption (HE) approaches, which applies HE on the shared information, making it impossible to reconstruct private client data. HE allows computations to be performed directly on the encrypted data, so that the server can still aggregate or process the encrypted inputs without needing to decrypt them. This ensures that private client data remains fully encrypted and secure throughout the process, preventing adversaries from reconstructing or accessing sensitive information, even if they intercept the exchanged data [24–26].

2.1.4 Fair FL

The problem of fairness in federated settings arises when the interest of individuals is ignored. This can happen in all the steps of decentralized training including client selection [27–29], evaluation contribution [30–32], and contribution of the clients in the global model aggregation.

To address challenges associated with data heterogeneity among clients, FedProx [33] introduces a proximal term to the local objective function to prevent local models from diverging significantly from the global model. MOON [34] leverages model contrastive learning to correct local representations, enhancing the consistency

between local and global models. FedDrop [35] incorporates dropout techniques to mitigate the impact of system heterogeneity, while FedNova [36] proposes a normalized averaging method that accounts for differences in the number of local updates performed by each client, thereby addressing system heterogeneity and imbalance in training contributions.

Other approaches focus on improving communication efficiency and robustness. FedBuf [37] employs buffered asynchronous aggregation to tackle scalability and privacy issues. FedALA [38] dynamically aggregates the downloaded global model with the local model on each client, ensuring alignment with the local objective. FLTrust [39] and SignGuard [40] enhance the robustness of FL systems against model poisoning attacks by leveraging gradient filtering techniques to detect and eliminate malicious gradients.

2.2 Federated Domain Generalization

Federated Domain Generalization (FDG) aims to train a global model that generalizes well to unseen target domains with distribution shifts, all while preserving data privacy by not sharing raw data among clients [5, 41, 42]. This area of research addresses the challenges posed by data heterogeneity, where each client's local data may come from different domains or distributions.

Several studies focus on learning domain-invariant features or identifying common features across multiple domains to enhance the global model's robustness. FedSR [43] employs L2-norm and conditional mutual information regularizers at the client level to discourage the learning of domain-specific features, promoting better generalization to target domains. FedADG [44] utilizes adversarial training to align source domain

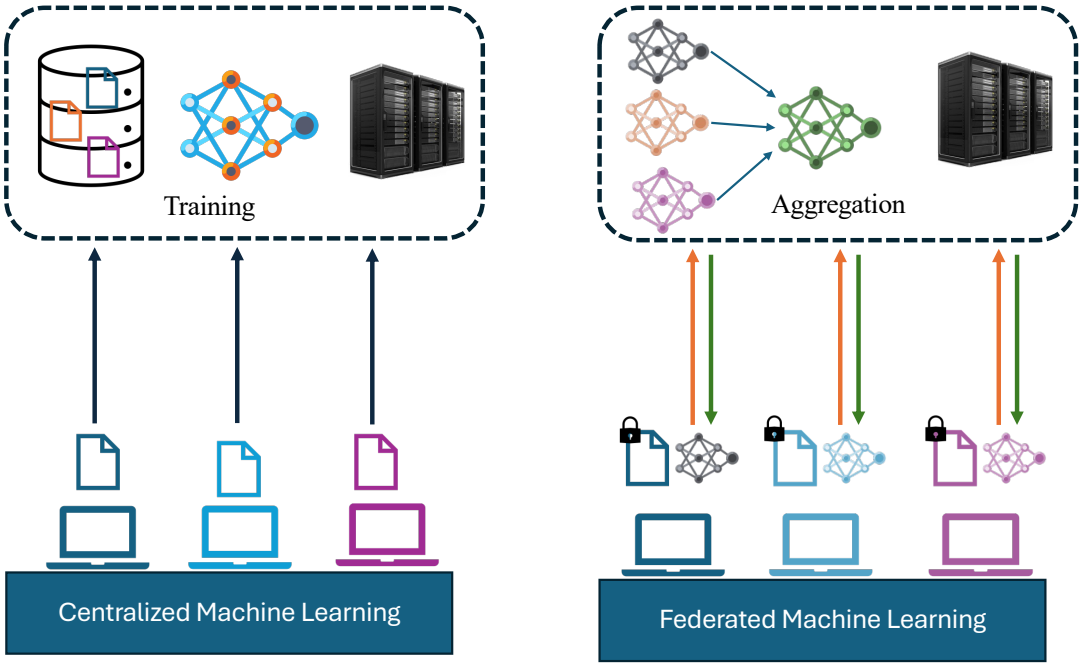

Figure 2.1: Overview of Federated Learning.

distributions by matching each distribution with a reference distribution.

Some recent works explore the sharing of certain information among clients to improve generalization, albeit at the potential cost of privacy. CCST [45] facilitates the sharing of a style bank between clients, which contains the mean and standard deviation of representations generated from each client's local data. A style transfer model, such as AdaIN [46], is then employed to transfer styles between clients, promoting cross-client generalization. COPA [47] shares classification heads among clients, encouraging local encoders to learn domain-invariant features that perform well with classification heads from other domains. However, this approach requires communicating extra layers during each communication round, potentially increasing the risk of privacy breaches.

Other methods like FedProto [48] and FedCDG [49] share class prototypes among clients, effectively distributing class representations. Nevertheless, sharing explicit information can undermine privacy, which is a critical concern in federated learning. FedDG [5] allows clients to share their data in the frequency space for medical image segmentation, aiming to capture domain-invariant features. FedKA [50] learns domain-invariant features by employing feature distribution matching in a universal workspace. CCST [45] aligns various client distributions and mitigates model biases by adapting local models to diverse sample styles via cross-client style transfer. StableFDG [51] uses style and attention-based strategies to address the federated domain generalization problem. hFedF [52] utilizes hypernetworks for non-linear aggregation to facilitate generalization to unseen domains. FedDG-GA [53] enhances generalization through dynamic domain weight adjustment based on domain divergence and a moment mechanism.

2.3 Federated Unsupervised Learning

Federated Unsupervised Learning focuses on learning representations from unlabeled data distributed across clients while preserving data privacy [54]. This area addresses scenarios where labeled data is scarce or unavailable, which is a common challenge in federated settings due to the decentralized nature of data collection and labeling efforts.

The study by Van et al. [55] pioneers federated unsupervised learning using an encoder-decoder architecture. FedU [56] employs a contrastive learning approach with online and target networks, enabling each client to independently learn representations from unlabeled data. It also introduces a dynamic aggregation mechanism to

update the predictor, either locally or globally, enhancing the model's adaptability.

Similarly, FedEMA [57] uses the exponential moving average of the global model to update local ones, promoting consistency across clients. FedX [58] proposes the use of knowledge distillation to learn representations from local data and refine the central server's knowledge, facilitating the aggregation of diverse local models. FedCA [59] introduces two components: a dictionary module that gathers representations of samples from each client to maintain consistency in the representation space, and an alignment module that adjusts each client's representation to match a base model trained on public data. These approaches aim to overcome the challenges of unlabeled data and data heterogeneity in federated settings.

2.4 Federated Unsupervised Domain Generalization

As discussed earlier, Federated Unsupervised Domain Generalization is a new problem space that unifies federated domain generalization and federated unsupervised learning, which had not been previously explored in the literature. The key distinction of this problem is the simultaneous presence of domain shifts among clients and the absence of labeled data at both client and server levels.

Our work, FedGaLA [60], is the first to address this problem. We establish a connection between domain shifts and gradient alignment in unsupervised federated learning. By aligning gradients at both the client and server levels, FedGaLA facilitates the learning of domain-invariant features from unlabeled data in a privacy-preserving manner. This dual alignment encourages clients to focus on learning representations that generalize well to unseen domains, enhancing the overall performance of the global model.

2.5 Summary

The existing research in federated learning has made significant strides in addressing various challenges such as data heterogeneity, system heterogeneity, communication efficiency, and robustness against attacks. However, limitations persist, particularly in scenarios involving both domain shifts and the absence of labeled data.

Our contributions through FedGaLA and FedSB address these gaps by introducing novel methodologies that unify federated domain generalization and unsupervised learning, and by enhancing supervised federated learning through label smoothing and balanced training. These advancements pave the way for more robust, generalizable, and privacy-preserving federated learning models capable of performing effectively across diverse and unseen domains.

Chapter 3

Federated Unsupervised Domain Generalization using Global and Local Alignment of Gradients

In this chapter, we introduce a new problem category, named FUDG, where we combine the problem of FDG and Federated Unsupervised Learning and introduce a solution based on Local and Global alignment of gradients. FL [1, 2] has emerged as a promising framework for training machine learning models in a decentralized manner, and it allows clients to collaboratively train a global model without the need to exchange their sensitive and local data. However, given that each client collects a different set of local training data, two issues arise. First, the data collected by each client is often recorded under unique conditions that may result in mutual domain shifts [5]. Second, labeling training data is inherently challenging and resource-intensive; this issue is even more pronounced in the context of federated settings. A typical example of this scenario is a network of wearable activity monitors where variations in user conditions such as demographics or ambient factors can lead to significant domain shifts across devices, meanwhile, the users are generally not asked to provide ground-truth labels for their performed activities.

Given that in prior works, each of these issues has been addressed as a separate problem statement: (*i*) *federated domain generalization* [43,44,61], and (*ii*) *federated unsupervised learning* [56,58], each ignores the fundamental assumptions of the other regarding the data in terms of distributions and availability of labels. To further approach federated learning in a more practical scenario, we propose to merge these two under a new umbrella called *FUDG*, which we define as Definition 3.1.1. To our knowledge, prior works have not studied federated learning under such constraints.

3.1 Approach

This section formulates the problem of FUDG and presents a novel solution based on local and global gradient alignments.

3.1.1 Problem Formulation

Definition. *Federated unsupervised domain generalization is the problem of learning general representations from various decentralized* ***unlabeled*** *datasets, each belonging to a* ***different domain,*** *in a federated setup where data sharing is restricted due to privacy concerns.*

To formalize Definition 3.1.1, assume K clients, C_i, in a federated setup, each with its own *unlabeled* data $D_i = \{\mathbf{x}_i^{(n)}\}_{n=1}^{N_i}$. Each dataset consists of N_i data points sampled from a distinct data distribution $p(\mathbf{x}_i)$, where $\mathbf{x}_i$ is a vector of F features, i.e., $\mathbf{x}_i = [x_i^1; x_i^2; ...; x_i^F]^T$. The data distributions are assumed to be different among the clients with each distribution $p(\mathbf{x}_i)$ sampled from a family of distributions P. Privacy constraints prevent the transfer of data between clients or to the server S. The objective is to learn generalized representations from $\mathbf{x}_i$ that perform well across unseen

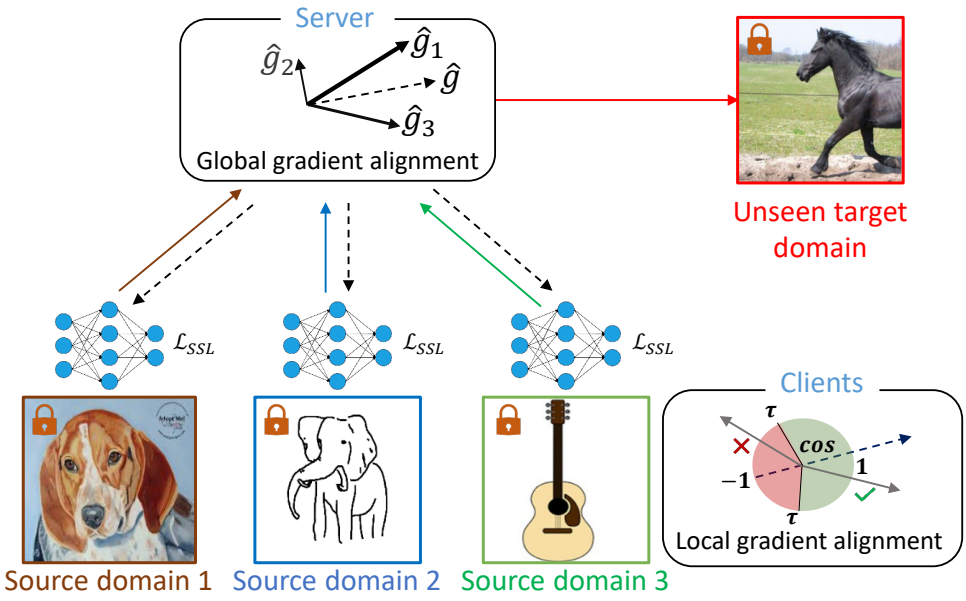

Figure 3.1: Overview of *FedGaLA*. We conduct local gradient alignment in the clients as well as global gradient alignment in the server

distributions $p(\mathbf{x}_t)$ P, where $p(\mathbf{x}_i) \neq p(\mathbf{x}_t)$. This is formulated as minimizing the expected loss over the unseen distributions:

$$\min \mathsf{E}_{p(\mathbf{x}_t)} \; _P \; \mathsf{E}_{p(\mathbf{x}_t)} [L(\; ; \mathbf{x})] \; ; \tag{3.1}$$

where L is the unsupervised loss function, and is the set of global model parameters. Each client contributes to this goal by computing a local objective function approximating the expected loss with respect to its own data distribution:

$$\min_{i} \mathsf{E}_{p(\mathbf{x}_i)} [l_i(\; _i; \mathbf{x})] \quad \frac{1}{N_i} \sum_{n=1}^{N_i} l(\; _i; \mathbf{x}_i^{(n)}); \tag{3.2}$$

where $_i$ indicates the local parameters of client C_i, and is the global aggregation of all local $_i$.

3.1.2 Gradient Alignment and Domain Shift

Under a FL framework, privacy constraints prevent clients and servers from accessing each other's data, including distribution information such as data means and variances. They can, however, observe individual client gradients at the server level and the average aggregated gradient across clients at the client level. We motivate our work on the fact that alignment of gradients may infer characteristics of the client domain distributions, thus facilitating improved model generalization. Under the problem proposed in Definition 3.1.1, for two distinct domains characterized by random variables $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ belonging to two different clients C_i and C_j, an increase in domain shift across the clients results in a decrease in covariance $\text{Cov}(\boldsymbol{g}_i; \boldsymbol{g}_j)$ of the corresponding gradients $\boldsymbol{g}_i$, $\boldsymbol{g}_j$ across C_i and C_j's respective local models.

3.1.3 FedGaLA

The analysis provided in [60] a link between gradient alignment and domain shift, forming the basis for our proposed FedGaLA framework. Figure 3.1 illustrates an overview of our framework for addressing the newly proposed problem setup (Definition 3.1.1). FedGaLA is remotely inspired by prior works that demonstrate improved generalization in centralized (non-federated) learning through gradient alignment [62–64]. However, our framework extends this notion by integrating gradient alignment into the field of unsupervised and FL, and assumes, unlike [62–64], that the distribution of data from different clients is not known. Our core idea includes (i) enabling clients to learn domain-invariant representations at the client level through local gradient alignment, and (ii) adjusting the aggregation weights at the server level using global gradient alignment. At each communication round, clients are initialized

with the global model. Subsequently, each client updates its parameters using SSL for E epochs based on the local data through local gradient alignment and sends these updates back to the server. Finally, the server employs the global gradient alignment technique to perform aggregation. This procedure is repeated for T communications rounds to determine the global model. The remaining part of this section provides details on local and global gradient alignment, and the complete framework is outlined in Algorithm 1.

3.1.4 Local Gradient Alignment

Our method performs layer-wise local gradient alignment using a reference gradient. This reference is derived from the l^{th} layer of the global model's parameter updates between the current and the previous communication round. Suppose $\Theta = f^{(l)}g_{l=1}^{L}$ indicates the parameters of the global model, where $^{(l)}$ represents the parameters of the l^{th} layer. The reference gradient for the l^{th} layer is computed as

$$\hat{\mathbf{g}}_{est}^{(l;t)} = {}^{(l;t)} \quad {}^{(l;t\ 1)}; \tag{3.3}$$

where $^{(l;t)}$ and $^{(l;t\ 1)}$ are the parameters of the l^{th} layer of the global model at rounds t and $t\ \ 1$, respectively. Then, $\hat{\mathbf{g}}_{est}^{(l;t)}$ is locally used to determine whether the gradient of each layer obtained during training (e.g., via SGD) is aligned with the reference. The cosine similarity between the batch gradient and the reference for each layer l at round t is computed as

$$\cos(\ )^{(l;t)} = \frac{h\mathbf{g}_{i;k}^{(l;t)}; \mathbf{g}_{\text{est}}^{(l;t)} i}{k\mathbf{g}_{i;k}^{(l;t)} k \ \ k\mathbf{g}_{\text{est}}^{(l;t)} k}; \tag{3.4}$$

where $\mathbf{g}_{i;k}^{(l;t)}$ is the gradient of k^{th} batch of the i^{th} client at layer l and round t. Finally, batch gradients whose similarity with $\hat{\mathbf{g}}_{est}^{(l;t)}$ are less than a user-defined threshold are discarded during the update process. This prevents clients from learning domain-specific features by disregarding local gradients that are not aligned with the global model. The rationale for discarding unaligned gradients instead of applying soft weighting is that when a gradient vector is unaligned with $\hat{\mathbf{g}}_{est}$, scaling does not change this alignment, as the cosine of the angle between them is independent of scale. To establish a basis for the proposed local alignment, we introduce the following. Given two sets of gradient vectors $\boldsymbol{g}_i$ and $\boldsymbol{g}_j$, by removing the K^{th} vector in $\boldsymbol{g}_j$ where $\cos(\boldsymbol{g}_{j;K}, \boldsymbol{g}_{est}) < 0$, the covariance of two sets increases.

This highlighted how an increase in domain shift between clients C_i and C_j correlates with a decrease in the covariance of their gradient vectors, $\mathbf{g}_i$ and $\mathbf{g}_j$. Therefore, gradient covariance can be used as an indicator of domain shift. it can also be shown that by selectively removing gradients from $\mathbf{g}_j$ that have a negative cosine similarity with an estimated target direction, $\mathbf{g}_{est}$, we can effectively increase the covariance of the gradient sets, thus potentially counteracting the effects of domain shift.

3.1.5 Global Gradient Alignment

To aggregate the local (client) models at the server, the locally measured gradients that closely match the average gradient across all clients are assigned greater weights. This soft weighting process operates as follows. Once the server receives the local models, it first obtains the local update using the following formula:

$$\hat{\mathbf{g}}_i^{(t)} = \Theta_i^{(t)} \quad \Theta^{(t\ 1)} \tag{3.5}$$

Then, the initial global update $\hat{\mathbf{g}}^{(t+1)}$ is calculated by averaging all $\hat{\mathbf{g}}_i^{(t)}$. Subsequently, the weight of each client is computed using

$$w_i = \frac{\cos(\hat{\mathbf{g}}_i^{(t)}, \hat{\mathbf{g}}^{(t+1)}) + 1}{2}, \tag{3.6}$$

where w_i is the weight for the i^{th} client and $\hat{\mathbf{g}}_i^{(t)}$ is the gradient of the i^{th} client at round t. This weight reflects the degree of alignment between each client's update and the global model's update direction. To ensure these weights are properly normalized using

$$w_i = w_i / \sum_{k=1}^{K} w_k \tag{3.7}$$

The normalization of weights allows for the proportional contribution of each client's update.

Finally, the normalized weights are used to perform aggregation. Each client's model update is scaled by its respective weight, and these weighted updates are then aggregated to compute the weighted average update. The global model at round $t+1$ is updated based on the following

$$\hat{\mathbf{g}}^{(t+1)} = \frac{1}{K} \sum_{i=1}^{K} w_i \hat{\mathbf{g}}_i^{(t)} \tag{3.8}$$

This aggregation step is repeated three times, refining the weights with each iteration. This ensures the global model update is significantly influenced by clients whose updates align closely with the global learning objective. After completing the aggregation process, the global model is further updated based on $\Theta^{(t+1)} \quad \Theta^{(t)} + \hat{\mathbf{g}}^{(t+1)}$. The rationale for using different alignment strategies at the client and server levels stems from the inherent differences between local training and global aggregation. At

the client level, we manage batch gradients, allowing us to specifically discard the unaligned ones without significant information loss. However, discarding gradients at the server-level corresponds to deletion of entire clients, which can adversely affect the outcome. Algorithm 1 lays out the detailed procedure of our proposed method.

Algorithm 1 FedGaLA

Input: data D_i, initialization ^{0}
Output: ^{T}
for t from 1 to T **do**
  Server:
    Calculate client updates: $\mathbf{g}_i^{(t)}$ $^{(t)}_{i}$ $^{(t\ 1)}$
    Initialize global update: $\mathbf{g}^{(t+1)}$ FedAvg($\mathbf{g}_i^{(t)}$)
  for j from 1 to $iter$ **do**
      Calculate weights: w_i $\frac{\text{Cosine}(\mathbf{g}_i^{(t)};\mathbf{g}^{(t+1)})+1}{2}$
      Normalize weights: w_i $\frac{w_i}{\sum w_i}$
      Aggregate updates: $\hat{\mathbf{g}}^{(t+1)}$ $\sum_{i=1}^{K} w_i \mathbf{g}_i^{(t)}$
  end for
    Update global model: $^{(t+1)}$ $^{(t)}$ + $\hat{\mathbf{g}}^{(t+1)}$
    Communicate: $_{i}$ $^{(t+1)}$
  Client:
  for e from 1 to E **do**
    for j from 1 to N_{batch} **do**
      Compute batch gradient: $\mathbf{g}_{i;j}^{(t)}$ $r\, l_i(x_{i;j};\ ^{(t)}_{i})$
      for l from 1 to L **do**
        Estimate reference: $\hat{\mathbf{g}}_{est}^{(l;t)}$ = $^{(l;t)}$ $^{(l;t\ 1)}$
        if Cosine($\mathbf{g}_{i;j}^{(l;t)}$; $\hat{\mathbf{g}}_{\text{est}}^{(l;t)}$) > **then**
          Update weights:
          $^{(l;t)}_{i}$ $^{(l;t\ 1)}_{i}$ $r\, l_i^{(l;t)}(x_{i;j};\ ^{(l;t)}_{i})$
        end if
      end for
    end for
  end for
    Communicate: S $_{i}$
end for

3.2 Experimental Setup

3.2.1 Datasets

To evaluate the effectiveness of our proposed method, we conduct experiments across four commonly used benchmarks for domain generalization. They include: **PACS** [65], which consists of 9,991 images from four domains: 'Photo', 'Art-painting', 'Cartoon', and 'Sketch', across seven classes; **Office-Home** [66], which consists of 15,588 images from four domains: 'Art', 'Clipart', 'Product', and 'Real-world', across 65 classes; **TerraInc** [67], which includes 24,788 images from four domains 'Location 38', 'Location 43', 'Location 46', and 'Location 100', across nine classes; **DomainNet** [68], which consists of 569,010 images in six domains: 'Clipart', 'Infograph', 'Painting', 'Quickdraw', 'Real', and 'Sketch', covering 345 classes. Following [69], for the DomainNet dataset, we select the following classes: zigzag, tiger, tornado, flower, giraffe, toaster, hexagon, watermelon, grass, hamburger, blueberry, violin, fish, sun, broccoli, Eiffel tower, horse, train, bird, and bee [69], which results in a total of 38556 samples. In all experiments using this dataset, three domains are used for training ('Painting', 'Real', and 'Sketch') and the model is tested using the other three domains ('Clipart', 'Infographics', and 'Quickdraw').

3.2.2 Image Augmentations

We follow [70] for augmentations. For all datasets, a random patch of the image is selected and resized to 32 32. Subsequently, we apply two random transformations, namely horizontal flip and color distortion.

3.2.3 Network Architecture

Predictor. For BYOL, we use a two-layer multilayer perceptron as the predictor. It begins with a fully connected layer featuring 4096 neurons, followed by one-dimensional batch normalization and a ReLU activation function. It concludes with another fully connected layer comprising 2048 neurons.

Encoder. We use ResNet18 [71] as the encoder for all the experiments. We use the ResNet architecture presented in [57], which is slightly different from the original architecture: (*i*) The first convolution layer employs a 3 3 kernel size, replacing the original 7 7; (*ii*) An average pooling layer with a 4 4 kernel size is used before the final linear layer, substituting the adaptive average pooling layer; and (*iii*) The last linear layer is replaced with a two-layer Multilayer perceptron (MLP), which shares the same network architecture as the predictor.

3.2.4 Training and Evaluation

We implement all our models using PyTorch and provide an easy-to-use framework for federated domain generalization in our released repository. Below, we provide further details regarding the hyperparameters used in the training and evaluation processes.

Training. While training the clients, we use a batch size of 128. Each client is trained for 7 local epochs before being returned to the server for a communication round. By default, we train for 100 communication rounds. We use the Adam [72] optimizer with a learning rate of 3 10 ^{3}. The hyperparameters used for training FedSimCLR are the same as those used in FedGaLA. For other baselines (FedMoCo, FedSimSiam, FedBYOL, and FedEMA), we use the SGD optimizer with a momentum

of 0.9 and a weight decay of 3 10 ^{4}. The choice of learning rate for FedSimSiam, FedBYOL, and FedEMA is 0.03 while for FedMoCo we use a learning rate of 0.025. The parameters have been tuned to maximize performance.

Evaluation Linear evaluation is used to assess the quality of learned representations. We train a fully connected layer on the top of the frozen encoder, which is trained for 100 epochs using the Adam optimizer with a learning rate 3 10 ^{3}.

3.2.5 Regularizers

Below we provide the details of the L2-norm and proximal term regularizers used in [43] and [33], respectively.

L2-norm. Suppose Θ_i indicates the parameters of the i^{th} client and $L_{SSL;i}$ represents the self-supervised loss of the i^{th} client. The L2-norm can be added to the loss of the i^{th} client to obtain

$$L_{Total;i} = L_{SSL;i} + \quad k\Theta_i k_2^2; \tag{3.9}$$

where is the the regularization coefficient and $k\Theta_i k_2^2$ is the square of the L2-norm of the parameters, defined as:

$$k\Theta_i k_2^2 = \sum_j \quad {}^2_{i;j}; \tag{3.10}$$

where $_{i;j}$ represents the j^{th} parameter of the i^{th} client.

3.2.6 Proximal Term

We also utilize the proximal term from [33] to further penalize the deviation of local models from the global model. The formulation of the proximal term is

$$L_{Total;i} = L_{SSL;i} + \quad k\Theta_i \quad \Theta_g k_2^2; \tag{3.11}$$

where is the regularizer coefficient, Θ_g and Θ_i are the parameters of the global model and i^{th} client, respectively, and $k\Theta_i \quad \Theta_g k_2$ is the Euclidean norm of the difference of the weights of the local clients from the weights of the global model. This ensures that the parameters of the local model Θ_i do not diverge heavily from the previous global model.

3.2.7 Evaluation

We use the leave-one-domain-out setting used in prior works [59, 69, 73]. This involves selecting one domain as the target, training the model on the rest of the domains, and then testing the model's performance on the selected target domain. Linear evaluation, a common feature evaluation approach, is utilized to evaluate the quality of learned representations [74–76]. For linear evaluation, following [55, 57], we utilize 10% and 30% of the target data to train the linear classifier, and evaluate the remaining 90% and 70% of the data, respectively.

3.2.8 Baselines

To evaluate our method, we take a two-pronged approach: (**1**) We adapt several popular SSL approaches to the federated domain generalization task, denoting

them as FedSimCLR, FedMoCo, FedBYOL, and FedSimSiam. To do so, we first train each client locally using the respective SSL method. Next, we aggregate the trained encoders at the server using FedAVG [1]. For BYOL and SimSiam, we follow the procedure in [56] and apply FedAVG on the online encoder and projector. We also adapt FedEMA [57], which is a commonly used method originally developed for federated unsupervised learning. FedEMA integrates BYOL as an SSL technique into its structure. It is important to note that we avoid direct comparisons with FedDG [5], FedSR [43], FedADG [44], FedIIR [77], and FedDG-GA [53] since unlike our method, they employ the label information in their solutions. (**2**) Given the absence of prior research specifically addressing the problem of *federated* unsupervised domain generalization, we also compare FedGaLA to established *centralized unsupervised domain generalization* methods on the PACS and DomainNet datasets. We could not identify unsupervised domain generalization methods for Office-Home and TerraInc datasets. This comparison includes the following solutions: SimCLR [70], MoCo [78], BYOL [79], AdCo [80], and DARLING [69]. It is important to note that these methods are implemented in a non-federated environment and do not incorporate any data privacy constraints. All the results for these models are reported from [69].

3.2.9 Implementation Details

We use SimCLR as the SSL module in FedGaLA due to its performance on domain generalization problems as previously shown [69]. Following [69], ResNet-18 [71] is employed as the encoder network architecture for all experiments, which we train from scratch. Following [74, 75], we first learn a representation by FedGaLA and the

baseline models for 100 communication rounds with 7 local epochs. Next, we freeze the backbone model and train a liner classier for 100 epochs to perform prediction on the target domain. For FedEMA, we use the hyperparameters reported in [57]. All experiments were implemented using PyTorch and trained on 8 NVIDIA GeForce RTX 3090 GPUs. For each experiment, we train the models three times with random initialization seeds and report the average. Following prior works such as [43], all clients have the same network architecture and hyperparameters and are trained under similar settings.

Table 3.1: Results of linear eval. on PACS dataset using 10% of the test domain for training the linear evaluation head

Model	**P**	**A**	**C**	**S**	**Ave.**
FedEMA	50.0(0.7)	29.5(1.9)	42.4(2.3)	45.6(0.16)	41.9
FedBYOL	52.1(1.1)	31.8(1.1)	45.4(2.2)	47.4(1.9)	44.2
FedMoCo	58.5(2.2)	35.7(9.6)	37.7(12.9)	36.6(8.2)	42.1
FedSimSiam	46.2(1.1)	28.6(1.0)	46.7(0.6)	37.6(1.3)	39.8
FedSimCLR	64.2(1.2)	41.9(1.5)	58.4(1.3)	70.1(1.2)	58.6
FedGaLA (ours)	**64.7(1.9)**	**44.2(1.2)**	**60.5(2.2)**	**70.5(1.3)**	**60.0**

Table 3.2: Results of linear eval. on PACS dataset using 30% of the test domain for training the linear evaluation head

Model	**P**	**A**	**C**	**S**	**Ave.**
FedEMA	33.9(2.8)	**48.3(2.4)**	53.5(0.3)	45.3(2.3)	45.2
FedBYOL	55.1(1.2)	35.3(1.5)	48.3(1.5)	48.9(0.5)	46.9
FedSimSiam	58.5(2.2)	35.7(9.6)	37.7(12.9)	36.6(8.2)	42.1
FedMoCo	47.3(1.6)	30.2(3.4)	47.4(0.7)	34.4(1.7)	39.9
FedSimCLR	69.8(1.1)	46.4(2.1)	63.9(1.6)	73.4(3.0)	63.3
FedGaLA (ours)	**71.1(2.0)**	46.8(2.1)	**65.7(1.6)**	**74.5(1.1)**	**64.6**

Table 3.3: Results of linear eval. on DomainNet dataset using 10% of the test domain for training the linear evaluation head

Model	C	I	Q	Ave.
FedEMA	38.6(1.1)	13.7(0.5)	45.5(1.6)	32.4
FedBYOL	38.1(0.5)	14.1(0.4)	53.6(2.9)	31.8
FedMoCo	30.5(0.6)	10.9(2.1)	46.4(0.8)	27.2
FedSimSiam	44.8(1.5)	12.2(0.3)	40.3(2.4)	36.9
FedSimCLR	45.2(0.4)	13.7(0.3)	59.7(0.7)	39.5
FedGaLA (ours)	**47.6(0.9)**	**14.2(0.5)**	**61.4(0.3)**	**41.1**

Table 3.4: Results of linear eval. on DomainNet dataset using 30% of the test domain for training the linear evaluation head

Model	C	I	Q	Ave.
FedEMA	43.5(0.8)	20.0(1.5)	49.0(2.8)	37.5
FedBYOL	44.6(1.2)	**20.5(1.1)**	50.4(0.9)	38.5
FedSimSiam	35.7(1.1)	14.9(3.9)	44.6(3.5)	31.8
FedMoCo	51.5(1.8)	18.2(0.1)	59.9(3.6)	43.2
FedSimCLR	51.7(1.0)	16.3(0.2)	66.5(0.9)	44.8
FedGaLA (ours)	**52.4(0.7)**	16.2(0.6)	**68.8(0.6)**	**45.8**

3.3 Results

Performance. We report the accuracy rates of FedGaLA and baseline models on the four datasets. As shown in Tables 3.1, 3.5, 3.3, and 3.7, FedGaLA consistently outperforms all baselines across all four datasets, with the exception of 'Art-painting' domain in Office-Home, for the 10% data regime. When 30% of the data are used, our method still generally outperforms the baseline models, although for some of the domains, the baseline solutions produce slightly better results; this is demonstrated in the results in Tables 3.2, 3.6, 3.4, and 3.8,. This observation is expected given that with the introduction of more domain-specific training data, the need for domain generalization declines, and thus, methods that are not explicitly designed for

Table 3.5: Results of linear eval. on OfficeHome dataset using 10% of the test domain for training the linear evaluation head

Model	A	C	P	R	Ave.
FedEMA	**20.8(0.9)**	6.7(0.5)	12.5(0.4)	14.1(0.8)	13.5
FedBYOL	7.4(0.3)	12.9(0.5)	20.9(1.1)	13.8(0.1)	13.8
FedSimSiam	8.5(0.5)	19.8(0.8)	28.2(0.8)	16.2(0.6)	18.9
FedMoCo	10.8(0.4)	9.4(0.3)	12.4(0.9)	10.1(0.7)	10.7
FedSimCLR	8.9(0.4)	24.3(0.3)	35.2(1.2)	20.0(0.2)	22.0
FedGaLA (ours)	8.9(0.4)	**25.3(0.6)**	**36.6(0.3)**	**21.2(0.5)**	**23.0**

Table 3.6: Results of linear eval. on OfficeHome dataset using 30% of the test domain for training the linear evaluation head

Model	A	C	P	R	Ave.
FedEMA	10.0(0.3)	17.3(0.7)	28.2(1.1)	17.8(0.3)	18.2
FedBYOL	10.4(0.3)	17.4(1.2)	29.4(1.3)	17.6(0.6)	18.7
FedSimSiam	12.7(1.1)	24.1(1.9)	36.5(0.7)	21.6(1.1)	23.7
FedMoCo	15.2(0.6)	11.2(0.9)	15.8(0.7)	11.5(1.6)	13.4
FedSimCLR	**13.6(1.3)**	**35.3(0.7)**	47.6(0.9)	26.6(1.0)	30.7
FedGaLA (ours)	**13.6(0.6)**	35.1(0.2)	**48.0(1.3)**	**27.0(0.7)**	**30.9**

domain generalization can produce competitive results. Across the four datasets, we observe that among the baseline models, FedSimCLR achieves better results compared to FedSimSiam, FedBYOL, and FedMoCo. This finding is consistent with [69] where it was demonstrated that SimCLR provides a better foundation for domain generalization versus other SSL methods, albeit in a non-federated setup.

Tables 3.9, and 3.10 compare the performance of FedGaLA with centralized (non-federated) methods for PACS and DomainNet datasets, respectively. As shown in this table, FedGaLA outperforms centralized methods by large margins. This is an expected observation as prior works have shown that federation can boost domain generalization [43, 81].

Table 3.7: Results of linear eval. on TerraInc dataset using 10% of the test domain for training the linear evaluation head

Model	**L38**	**L43**	**L46**	**L100**	**Ave.**
FedEMA	62.1(0.4)	43.5(3.0)	43.4(0.4)	63.9(0.7)	53.2
FedBYOL	63.4(0.2)	43.3(0.5)	44.3(0.3)	66.3(2.4)	54.3
FedSimSiam	54.0(6.8)	36.0(1.5)	40.9(3.7)	60.8(1.6)	47.9
FedMoCo	50.0(2.4)	33.8(0.3)	29.7(2.3)	60.2(0.1)	45.7
FedSimCLR	62.8(0.2)	45.8(1.1)	42.9(1.8)	68.8(0.3)	55.1
FedGaLA (ours)	**63.6(0.1)**	**47.6(1.4)**	**43.9(2.9)**	**71.7(0.9)**	**56.7**

Table 3.8: Results of linear eval. on TerraInc dataset using 30% of the test domain for training the linear evaluation head

Model	**L38**	**L43**	**L46**	**L100**	**Ave.**
FedEMA	62.7(0.6)	47.1(0.3)	46.1(0.4)	67.6(2.0)	55.9
FedBYOL	63.6(0.9)	46.7(1.3)	46.1(0.2)	68.8(2.5)	56.3
FedSimSiam	56.8(4.6)	35.7(2.0)	38.9(4.6)	61.4(1.4)	48.2
FedMoCo	60.7(0.1)	28.3(3.3)	31.7(2.3)	60.0(0.6)	45.2
FedSimCLR	65.8(0.3)	**54.5(1.2)**	49.0(0.4)	71.0(1.6)	60.1
FedGaLA (ours)	**66.2(0.5)**	52.4(1.7)	**51.7(0.8)**	**74.6(1.3)**	**61.3**

3.3.1 Ablation Studies

Here we examine the effectiveness of the local and global gradient alignment components individually on the final performance of FedGaLA. To this end, we systematically remove each of these components. As shown in Table 3.11, each component plays an important role in the overall performance. It is noteworthy to mention that FedGaLA essentially becomes the FedSimCLR baseline by removing both global and local alignments.

Table 3.9: Comparison of linear eval. (10%) results on PACS dataset in centralized setting

Model	**P**	**A**	**C**	**S**	**Ave.**
BYOL	27.0	25.9	21.0	19.7	23.4
MoCo	44.2	25.9	33.5	25.0	32.1
SimCLR	54.7	37.7	46.0	28.3	41.6
AdCo	46.5	30.2	31.5	22.9	32.8
DARLING	53.4	39.9	46.4	30.2	42.5
FedGaLA (ours)	**64.7**	**44.2**	**60.5**	**70.5**	**60.0**

Table 3.10: Comparison of linear eval. (10%) results on DomainNet dataset in centralized setting

Model	**C**	**I**	**Q**	**Ave.**
BYOL	14.6	8.7	5.9	9.7
MoCo	32.5	18.5	8.1	19.7
SimCLR	37.1	19.9	12.3	23.1
AdCo	32.3	17.9	11.6	20.6
DARLING	35.2	20.9	15.7	23.9
FedGaLA (ours)	**47.6**	**14.2**	**61.4**	**41.1**

3.3.2 Gradient Misalignment Due to Domain Shift

In Figure 3.2 we illustrate the amount of measured covariance for different domains in PACS versus the amount of domain shift between each pair measured through

Table 3.11: Ablation study on PACS (GA: global alignment; LA: local alignment).

Models	P	A	C	S	Ave.
FedGaLA	**64.7(1.9)**	**44.2(1.2)**	**60.5(2.2)**	**70.5(1.3)**	**60.0**
w/o GA	**64.7(0.4)**	42.6(1.2)	58.1(0.6)	69.9(1.1)	58.8
w/o LA	63.7(1.5)	41.6(1.4)	59.8(1.5)	68.9(1.1)	58.5
w/o GA & LA	64.2(1.2)	41.9(1.5)	58.4(1.3)	70.1(1.2)	58.6

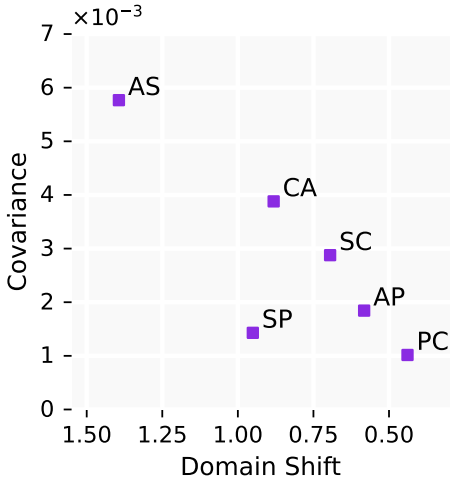

Figure 3.2: Covariance against domain shift for P, A, C, and S.

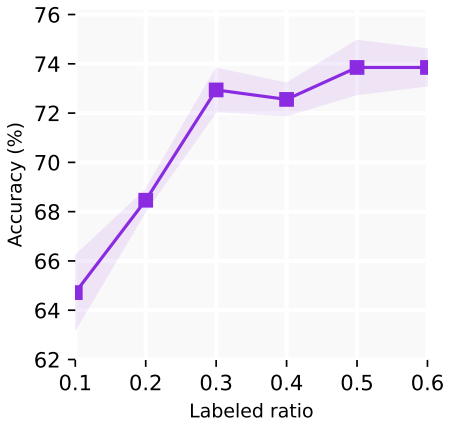

Figure 3.3: Impact of different labeled data ratios on linear evaluation

Mutual Information. We observe that, except for a single outlier, the trend follows our prediction that an increase in domain shift across the clients results in a decrease in covariance $\mathrm{Cov}(\boldsymbol{g}_i; \boldsymbol{g}_j)$ of the corresponding gradients across respective local models.

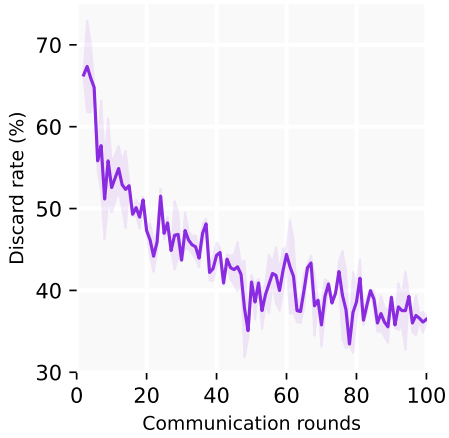

Figure 3.4: the average ratio of discarded local gradients over 100 communication rounds for $= 0$ and $E = 1$

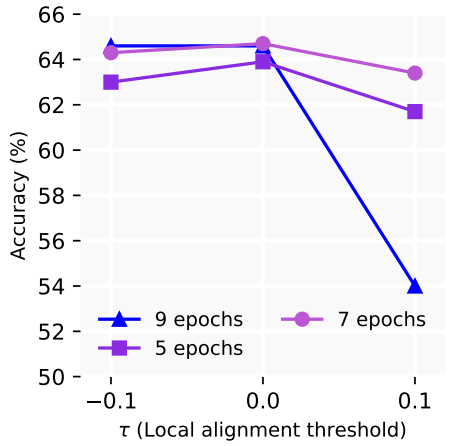

Figure 3.5: the impact of the local threshold for different numbers of local epochs

3.3.3 Ratio of Labeled Data

Figure 3.3 presents the results when evaluated with different ratios of labeled data, ranging from 10% to 60%. As can be seen, the accuracy increases significantly from 64% to 72% when the labeled ratio rises from 10% to 30%, followed by a steady climb to 74% as the labeled ratios increase to 60%. Overall, when more labeled data are used to train the linear classifier, the final performance expectedly improves.

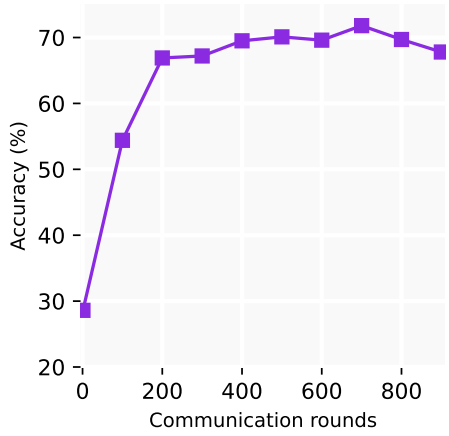

Figure 3.6: the impact of the communication rounds on performance

Table 3.12: The effect of regularizers on the performance of FedGaLA on the PACS dataset.

Model	**P**	**A**	**C**	**S**	**Ave.**
FedGaLA+L2 (= 0.001)	64.8(0.5)	42.8(1.1)	58.7(1.6)	67.3(0.6)	58.4
FedGaLA+L2 (= 0.01)	66.5(0.9)	43.4(0.9)	56.9(2.7)	65.2(0.6)	57.9
FedGaLA+FedProx (= 0.001)	62.1(0.2)	41.4(0.7)	58.9(3.1)	68.8(0.1)	57.8
FedGaLA+FedProx (= 0.01)	63.1(0.2)	41.6(1.2)	58.5(2.0)	68.9(1.9)	58.0
FedGaLA (ours)	**64.7(1.9)**	**44.2(1.2)**	**60.5(2.2)**	**70.5(1.3)**	**60.0**

3.3.4 Ratio of Discarded Local Gradients

Figure 3.4 demonstrates the ratio of local gradients discarded due to local gradient alignment during training versus communication rounds. For this experiment, the local models are trained for 1 epoch at each communication round. As observed, the ratio of discarded local gradients decreases from approximately 68% in the early communication rounds to 37% by round 100. This trend indicates that as the number of communication rounds increases, the local gradient directions become more aligned with the global model, suggesting that with increasing the number of communications, local models learn more domain-invariant features.

Model	A	C	P	R	Ave.
FedGaLA+L2 (= 0.001)	**11.8(0.4)**	23.9(0.5)	**37.1(0.4)**	**21.9(0.7)**	**23.6**
FedGaLA+L2 (= 0.01)	**11.8(0.2)**	21.2(0.1)	34.5(0.3)	22.1(0.2)	22.4
FedGaLA+FedProx (= 0.001)	10.2(0.4)	24.2(0.4)	36.4(1.1)	19.7(0.32)	22.6
FedGaLA+FedProx (= 0.01)	9.2(0.3)	23.6(0.7)	**37.1(0.6)**	20.4(0.4)	22.6
FedGaLA (ours)	8.9(0.4)	**25.3(0.6)**	36.6(0.3)	21.2(0.5)	23.0

Table 3.13: The effect of regularizers on the performance of FedGaLA on the Office-Home dataset.

Model	C	I	Q	Ave.
FedGaLA+L2 (= 0.001)	46.3(0.8)	**16.4(0.9)**	**62.8(0.2)**	**41.6**
FedGaLA+L2 (= 0.01)	38.9(1.2)	14.6(0.7)	52.5(0.8)	35.3
FedGaLA+FedProx (= 0.001)	44.6(0.2)	13.4(1.0)	60.9(1.1)	39.6
FedGaLA+FedProx (= 0.01)	44.7(1.2)	13.2(0.4)	61.7(0.4)	39.9
FedGaLA (ours)	**47.6(0.9)**	14.2(0.5)	61.4(0.3)	41.1

Table 3.14: The effect of regularizers on the performance of FedGaLA on the DomainNet dataset.

3.3.5 Local Threshold vs. The Number of Local Epochs

In Figure 3.5 we study the impact of the threshold for the similarity between gradients and the reference ( ) and the number of local epochs E on performance. In this experiment, we use three different values for : 0.1, 0, and 0.1, and three different values for E: 5, 7 or 9 (local epochs per communication round).

We observe that our method produces the best results when = 0, i.e., when we keep all gradients with positive cosine similarity with the reference. Expectedly, even discarding gradients with small amounts of alignment (= 0.1) degrades the results, while keeping gradients that are not aligned with the reference (= 0.1) also hurts performance. Moreover, we see that setting $E = 7$ yields the best performance as increasing the number of epochs beyond 7 does not have a positive impact, and only

Model	L38	L43	L46	L100	Ave.
FedGaLA+L2 (= 0:001)	61.6(0.9)	45.8(0.6)	46.6(0.7)	70.8(0.7)	56.2
FedGaLA+L2 (= 0:01)	61.5(0.7)	38.1(2.9)	44.9(0.1)	57.8(3.3)	50.6
FedGaLA+FedProx (= 0:001)	**63.8(0.7)**	**49.3(1.5)**	**47.2(0.1)**	71.5(0.9)	**57.9**
FedGaLA+FedProx (= 0:01)	63.2(1.1)	47.8(2.1)	44.8(0.4)	71.4(0.8)	56.0
FedGaLA (ours)	63.6(0.1)	47.6(1.4)	43.9(2.9)	**71.7(0.9)**	56.7

Table 3.15: The effect of regularizers on the performance of FedGaLA on the TerraInc dataset.

increases computational time.

3.3.6 Communication Rounds

We investigate the effect of the communication rounds T on the model's performance. Following [56, 57] we set $E = 1$ and vary the number of communication rounds T from 100 to 900. We observe from Figure 3.6 that performance improves significantly from 1 to 200 communication rounds, with this trend slowing down from 200 to 900.

3.3.7 Effects of Regularizers on FedGaLA

Prior works have demonstrated that adding regularizers can indeed improved generalization across domains or non-IID data [33, 43]. To this end, we test the impact of regularizers on FedGaLA by applying two types of regularizers based on L2 norm [33] and FedProx [43]. Please refer to section 3.2.5 for more details regarding these two techniques. The results in Table 3.12 demonstrate that FedGaLA is highly compatible with regularizers and that the addition of such approaches can further boost the performance of our method.

Models	P	A	C	S	Ave.
FedMoCo	46.2(1.1)	28.6(1.0)	46.7(0.6)	37.6(1.3)	39.8
FedGaLA(MoCo)	**46.5(0.3)**	**28.9(0.4)**	**47.9(0.3)**	**39.6(2.4)**	**40.7**
FedBYOL	52.1(1.1)	**31.8(1.1)**	45.4(2.2)	47.4(1.9)	44.2
FedGaLA(BYOL)	**52.8(0.6)**	31.7(0.5)	**46.3(3.1)**	**47.6(1.2)**	**44.6**
FedSimCLR	64.2(1.2)	41.9(1.5)	58.4(1.3)	70.1(1.2)	58.6
FedGaLA(SimCLR)	**64.7(1.9)**	**44.2(1.2)**	**60.5(2.2)**	**70.5(1.3)**	**60.0**

Table 3.16: The performance of FedGaLA when employing different SSL methods.

3.3.8 FedGaLA with other SSL and federated techniques

We examine the effectiveness of FedGaLA using other SSL frameworks (MoCo, BYOL, and SimCLR) and federated techniques (FedAVG and Moon [34]). Our findings demonstrate that FedGaLA improves various SSL baselines and also outperforms other federated protocols.

Table 3.16 illustrates the performance of FedGaLA when employing various SSL models in local training and compares its performance with the baseline. These findings highlight FedGaLA's capability to boost the performance of different SSL techniques when used for federated unsupervised domain generalization.

As demonstrated in Table 3.17, we investigate the effect of other federated techniques and demonstrate that our FedGaLA method outperforms recent methods like MOON [34].

Model	PACS				
	P	A	C	S	Ave.
MOON (= 1)	64.1(1.2)	43.1(0.9)	59.9(0.8)	70.0(0.7)	59.3
MOON (= 0:1)	**65.1(0.2)**	42.0(1.1)	58.1(1.4)	69.3(1.0)	58.61
MOON (= 0:01)	64.7(0.8)	42.8(1.5)	58.6(0.2)	69.6(0.6)	58.92
FedGaLA (ours)	64.7(1.9)	**44.2(1.2)**	**60.5(2.2)**	**70.5(1.3)**	**60.0**

Table 3.17: Comparison of FedGaLA with other federated techniques.

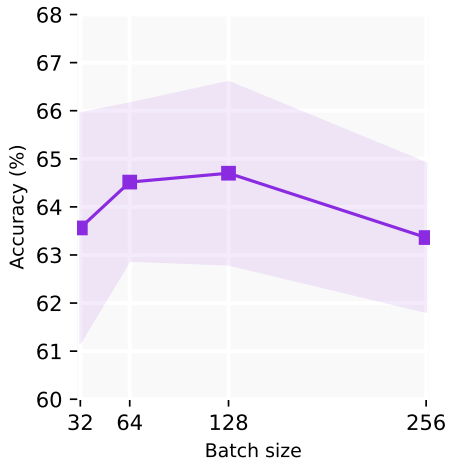

Figure 3.7: Impact of batch size on performance.

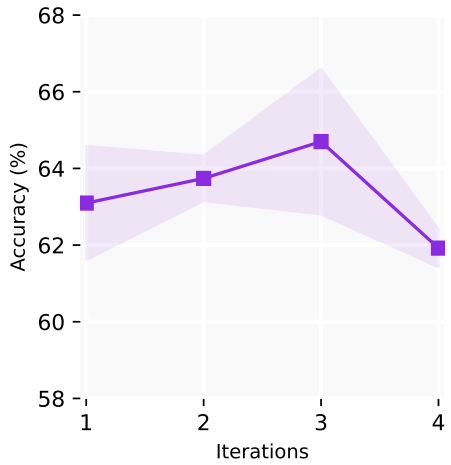

Figure 3.8: Impact of the number of iterations for global alignment on performance.

Chapter 4

Federated Supervised Domain Generalization using Budgeting and Label Smoothing

In this chapter, we focus on an underlying challenge in FL, which is data heterogeneity, where the data available to each local client follows different distributions, making it difficult for the global model to generalize effectively to *unseen* target domains.

While FDG techniques show promise in tackling data heterogeneity, two challenges persist as a result of this phenomenon. The first is the overconfidence of the clients on their local data, as they tend to learn domain-specific features. This overconfidence limits the effectiveness of these local models when aggregated to form the global model. Secondly, the distribution of samples in each client is different from that of other clients, leading to imbalances in model training, i.e., clients with more data samples could be contributing more to the formation of the global model. This in turn can result in a biased or sub-optimal performance [33].

4.1 Approach

In this section, we formulate the problem of FDG with multiple clients, and introduce our novel approach to tackle model overconfidence and unbalanced decentralized training through FedSB.

4.1.1 Problem Formulation

Let's assume $i \ 2 \ f1;2; \quad ;Kg$, where K is the total number of clients in the system. Here, client C_i has access to its local dataset $D_i = fx_i;y_ig$ from a specific distribution $p_i(x;y)$, with $x \ 2 \ X$ and $y \ 2 \ Y$ representing the inputs and labels, respectively. The objective of FDG is to learn a global model Θ by aggregating the local models $_i$, such that the global model can generalize to an unseen target domain with distribution $p_T(x;y)$, where $p_T(x) \neq p_i(x)$ for $i \ 2 \ [1;K]$, and D_T is not available during training.

4.1.2 Label Smoothing

Label smoothing was originally proposed as a regularization technique to improve the generalization performance of deep neural networks by introducing controlled uncertainty in the output predictions [82]. It works by adjusting the hard classification labels, reducing a model's tendency to become overconfident in its predictions, particularly in cases where the data distribution is imbalanced or noisy. In the original proposal, it was applied to the inception architecture to reduce overfitting and improve robustness during training.

This method has since been extensively adopted across various fields, including image classification [83–85] and speech recognition [86], where the models tend to

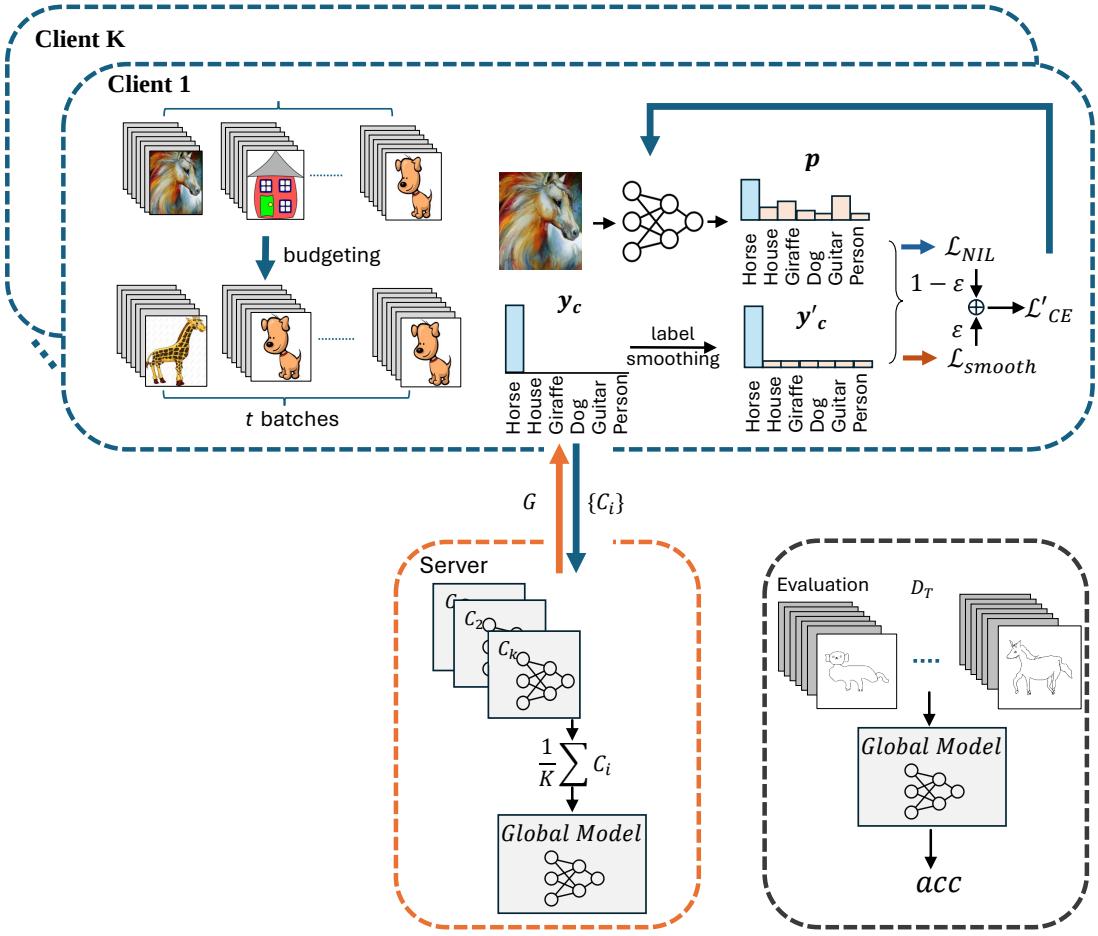

Figure 4.1: Overview of FedSB

overfit to specific domain characteristics. In our approach, label smoothing plays a crucial role in addressing model overconfidence at the client level, encouraging local models to learn more domain-invariant features, thus enhancing generalization in federated settings.

4.1.3 FedSB

We propose FedSB to address the challenges of data heterogeneity in FDG. As shown in Fig. 4.1, FedSB operates through two complementary steps. First, it encourages local clients to learn domain-invariant representations by reducing their local overconfidence through label smoothing. Second, it promotes a balanced contribution from different clients by utilizing a simple yet novel budgeting technique. The following sections provide a detailed explanation of these approaches.

4.1.4 Learning Domain-Invariant Features

Each local client has access to a relatively small dataset within a distinct domain. As a result, the trained local models can produce predictions with high confidence within this domain. However, these models lack generalization and perform poorly when presented with data from other domains. To alleviate this, we introduce a level of controlled uncertainty to the model to prevent local clients from overfitting to domain-specific representations. To achieve this, we employ a label smoothing technique and replace hard labels with soft labels, by altering ground truth labels as follows:

$$y_c^{\prime} = \begin{cases} 1 - \alpha + \frac{\alpha}{M}; & \text{if } c = y \\ \frac{\alpha}{M}; & \text{if } c \neq y \end{cases}; \tag{4.1}$$

where α is the smoothing coefficient, and M is the total number of classes in each client. This formulation ensures that

$$\sum_{c=1}^{M} y_c^{\prime} = 1. \tag{4.2}$$

Consequently, the cross-entropy loss using the smoothed labels can be derived as:

$$L = \quad (1 \quad + \frac{}{M}) \log(p_y) + \sum_{c \neq y} \frac{}{M} \log(p_c) \quad ! \quad : \tag{4.3}$$

Here, the collecting terms can be rearranged as:

$$L = (1 \quad)(\underbrace{\log(p_y)}_{\text{NLL Loss}}) + \quad (\underbrace{\sum_{c=1}^{M} \frac{1}{M} \log(p_c)}_{\text{Smooth Loss}}): \tag{4.4}$$

In this formulation, the Negative Log-Likelihood (NLL) loss over the target class encourages correct predictions, whereas the Smooth Loss reduces overconfidence by leveraging the incorrect classes (all classes other than the correct target class). We apply to control the level of smoothness of the labels, where higher values penalize the local models more for overconfident predictions.

Label smoothing is particularly beneficial in reducing overconfidence because it adds controlled uncertainty to the model's predictions, making the local client less reliant on domain-specific features. This technique not only prevents overfitting but also fosters more generalizable feature learning, which is critical in federated settings where the local datasets differ significantly in distribution. Unlike traditional techniques such as dropout, label smoothing modifies the ground truth data during training, directly influencing how the model perceives classification boundaries.

4.1.5 Balanced Training

Next, let us assume a local training process at round $t+1$ in client C_i. The model is initialized with the global parameters of round t denoted by Θ^t. We can characterize

the parameters of C_i after local training, using gradient descent, as:

$$\theta_i^{t+1} = \theta^t - \eta \sum_{j=1}^{b\frac{|\mathcal{D}_i|}{B}c} \nabla L_i^j; \tag{4.5}$$

where η is the learning rate, B is the batch size, and L_i^j is the loss of the j^{th} batch of the local dataset D_i. By aggregating all the local models in the server using a simple technique such as averaging, the global model is obtained as:

$$\theta^{t+1} = \frac{1}{K} \sum_{i=1}^{K} (\theta_i^{t+1}) = \theta^t - \frac{\eta}{K} \sum_{i=1}^{K} \sum_{j=1}^{b\frac{|\mathcal{D}_i|}{B}c} \nabla L_i^j. \tag{4.6}$$

Accordingly, as

$$\mathbb{E}[\sum_{j=1}^{b\frac{|\mathcal{D}_i|}{B}c} \nabla L_i^j] = b\frac{|D_i|}{B}c\mathbb{E}[\nabla L_i]; \tag{4.7}$$

where $\mathbb{E}[\nabla L_i]$ represent the expected update of client i, the expected global model parameters can be described as:

$$\mathbb{E}[\theta^{t+1}] = \theta^t - \frac{\eta}{K} \sum_{i=1}^{K} b\frac{|D_i|}{B}c\mathbb{E}[\nabla L_i]. \tag{4.8}$$

According to Eq. 4.8, we can deduce that clients with larger datasets, i.e., larger $|D_i|$, have a greater influence in determining the expected value of the global model's update compared to clients with smaller datasets. This can degrade generalization, as the global model tends to gravitate toward the domain of the more influential clients. To address this issue, we apply a simple trick to ensure that each client operates under a fixed training budget regardless of the local dataset size. Specifically, we use a fixed budget for all clients denoted as S. If $|D_i| > S$, we randomly select S samples from

Algorithm 2 FedSB algorithm

Input: D_i, Batch size b, Number of iterations t
Output: Resampled dataset $D_k^{\emptyset}$
Step 1: Initialize
Compute number of batches: n_{batches} $\frac{n_k}{b}$
Initialize resampled dataset: $D_k^{\emptyset}$;
if n_{batches} t **then**
    Sample t b indices I_k $f1; 2; : : : ; n_k g$ without replacement
    Construct resampled dataset: $D_k^{\emptyset}$ $f(x_i; y_i) g_{i2 I_k}$
else
    Sample t b indices I_k $f1; 2; : : : ; n_k g$ with replacement
    Construct resampled dataset: $D_k^{\emptyset}$ $f(x_i; y_i) g_{i2 I_k}$
end if
Step 2: Return Resampled Dataset
$D_k^{\emptyset}$ rst t b samples from $D_k^{\emptyset}$ (if applicable)
return Resampled dataset $D_k^{\emptyset}$

D_i, whereas where $jD_i j < S$ we oversample D_i. By using the under/over-sampled dataset $\hat{D}_i$ for training each client, the expected global model is derived as:

$$\mathsf{E}[\Theta^{t+1}] = \Theta^t \quad \frac{S}{KB} \sum_{i=1}^{K} \mathsf{E}[\, r \hat{L}_i]; \tag{4.9}$$

were $\hat{L}$ denotes the loss over $\hat{D}_i$. As demonstrated in this equation, this straightforward and intuitive solution can effectively ensure equal contribution from all clients toward the global update and mitigate the impact of data heterogeneity, particularly with respect to dataset size.

4.2 Experimental Setup

In this section, we provide details on the experimental setup used to test our FedSB model and provide information regarding the choice of network architecture, datasets and evaluations methods.

4.2.1 Datasets

We evaluate our method on four datasets namely PACS [65], OfficeHome [66], VLCS [87], and TerraInc [67]. The PACS dataset comprises 9,991 images of the 'Photo', 'Art Painting', 'Cartoon', and 'Sketch' domains, each containing seven classes. OfficeHome also has four distinct domains: 'Art', 'Clipart', 'Product', and 'Real-world', comprising over 15,500 images belonging to 65 classes. Furthermore, Terra Incognita includes 24,788 pictures of animals from four distinct locations. Finally, VLCS comprises 10,729 images from four distinct object classification datasets: VOC, LabelMe, Caltech101, and SUN09 with five shared classes.

4.2.2 Image Augmentation

We follow the same augmentation strategy provided in [88] for all our baselines, which has been the standard in the literature. We resize the images to 224 224 and apply CenterCrop, RandomHorizontalFlip, ColorJitter, and RandomGrayscale. We normalize our inputs to have the same mean and standard deviation as those of the ImageNet dataset.

4.2.3 Network Architecture

We utilize backbones pre-trained on the ImageNet dataset and replace the final classification layer (classification head) with a MLP containing M output nodes, where M is the number of classes in each dataset.

4.2.4 Implementation Details

Following prior works, we use ResNet-18 for the PACS and VLCS datasets as the feature extractor, while for OfficeHome and TerraIncognita, we utilize ResNet-50. In addition to ResNet-based backbones used by prior works, we also report our results on PACS and OfficeHome using Vision Transformers (ViTs) [89]. Given that ViTs have not been used by the baselines, we re-implement the commonly used FedAVG baseline using two variants of ViTs, namely ViT-b/16 and ViT-b/32. Finally, to compare against CCST [45], we also evaluate our method on PACS with the ResNet-50 backbone. All models are trained with Adam and a learning rate of 1e-4, using a batch size of 64. We conduct the training for 100 global communication rounds for PACS and 50 rounds for all other datasets. Mixstyle [90] is also applied within each client for each experiment based on CNN backbones. We implement and train our method and baselines using the PyTorch framework on an Nvidia RTX 3090 GPU.

Table 4.1: Comparison of image recognition accuracy on the PACS dataset. The single-letter columns represent the unseen (test) domain in the PACS dataset: P (Photo), A (Art), C (Cartoon), and S (Sketch). FedSB outperforms all other baselines by great margins on the PACS dataset

	P	A	C	S	Ave.
FedAvg [1]	91.67	79.25	70.46	75.98	79.34
FedADG [44]	93.64	81.39	75.39	78.56	82.25
FedProx [33]	91.69	79.16	71.45	75.51	79.45
FedSR [43]	93.00	78.37	72.22	77.53	80.27
FedIIR [77]	**94.56**	80.06	75.20	79.63	82.36
FedSB (ours)	94.19	**81.80**	**75.28**	**83.52**	**83.81**

Table 4.2: Comparison of image recognition accuracy on the OfficeHome dataset. The single-letter columns represent the unseen (test) domain in the OfficeHome dataset: P (Product), A (Art), C (Clipart), and R (Real World). FedSB outperforms all other baselines by great margins on the OfficeHome dataset

	P	A	C	R	Ave.
FedAvg [1]	78.30	64.16	54.88	79.08	69.10
FedADG [44]	74.87	60.27	56.09	76.48	66.93
FedProx [33]	77.45	64.33	55.02	79.53	68.14
FedSR [43]	73.50	58.46	50.83	73.93	64.18
FedIIR [77]	75.68	63.57	54.53	78.16	68.08
FedSB (ours)	**78.33**	**65.88**	**60.05**	**79.42**	**70.92**

4.2.5 Evaluation

Following [88], we utilize the leave-one-domain-out setting. Each time, we select one domain D_i as a target and train the model on the rest of the domains. We repeat this for all domains and average the performance. The reason for choosing this method of evaluation is the lack of test data for each domain and the nature of FDG.

4.2.6 Baselines

We compare FedSB with FedAVG [1], FedADG [44], FedProx [33], FedSR [43], and FedIIR [77]. We re-run all the baselines and report the average performance over three different runs, except FedADG [44], where we report the results from the original paper.

4.3 Results

Tables 4.1, 4.2, 4.3, 4.4 reports the performance of FedSB in comparison to the baselines on PACS, OfficeHome, TerraINC, and VLCS datasets, respectively. As demonstrated, FedSB achieves state-of-the-art results on three out of four datasets. Our FedSB method improves the baseline results on PACS by 4.47%, additionally improving the accuracy on the OfficeHome dataset by 1.82%. Furthermore, we improve the image recognition accuracy on the TerraIncognita dataset by 1.73%. The only dataset, in which FedSB does not beat state-of-the-art is VLCS, where it gets second best performance.

Additionally, as shown in Fig. 4.2, our method shows better separability of classes compared to that of the FedAVG baseline, which makes it easier for a classifier to classify different classes. Similarly, comparisons against CCST in Table 4.5 demonstrate the superiority of our approach, without explicitly sharing information like style.

Table 4.3: Comparison of image recognition accuracy on the TerraIncognita dataset. The single-letter columns represent the unseen (test) domain in the TerraIncognita dataset: L36, L43, L48, and L100 represent different geographical locations.

	L36	L43	L48	L100	Ave.
FedAvg [1]	41.37	43.34	39.90	56.02	45.16
FedProx [33]	**43.13**	**45.79**	40.64	55.18	46.19
FedSR [43]	24.33	33.43	30.97	56.80	36.38
FedIIR [77]	41.10	47.79	39.47	47.63	44.01
FedSB (ours)	38.37	44.56	**43.60**	**61.02**	**46.89**

Table 4.4: Comparison of image recognition accuracy on the VLCS dataset. The single-letter columns represent the unseen (test) domain in the VLCS dataset: V (VOC2007), L (LabelMe), C (Caltech), and S (SUN).

	V	L	C	S	Ave.
FedAvg [1]	72.27	60.25	96.35	70.37	74.81
FedADG [44]	73.20	61.20	95.78	74.95	76.28
FedProx [33]	73.16	60.39	97.03	72.62	75.80
FedSR [43]	68.83	59.63	95.83	71.63	73.98
FedIIR [77]	**76.42**	61.84	96.79	74.69	**77.44**
FedSB (ours)	74.28	**62.78**	**97.35**	72.17	76.64

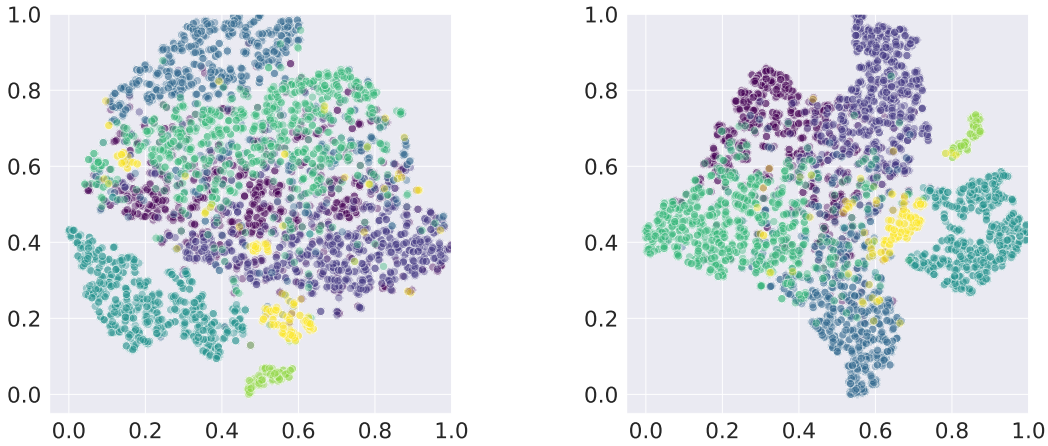

Figure 4.2: TSNE plot for S domain on PACS. The points are color-coded to represent different classes. FedSB better separates data points belonging to different classes, facilitating classification.

4.3.1 Ablation Studies

Table 4.6 presents the result of our ablation study where we systematically remove different components of FedSB and report the results on PACS. Table 4.6 presents the results of our ablation study, where we systematically remove different components

Table 4.5: Accuracy on PACS using a ResNet-50 backbone. FedSB achieves state-of-the-art performance even when compared to CCST, that explicitly shares data information among the clients.

Method	P	A	C	S	Ave.
FedAvg [1]	95.51	82.23	78.20	73.56	82.37
FedDG [5]	96.23	83.94	79.27	73.30	83.19
CCST [45]	**96.65**	**88.33**	78.20	82.90	86.52
FebSB (ours)	95.97	87.46	**80.86**	**85.22**	**87.33**

of FedSB and report the outcomes on the PACS dataset. While utilizing both components of FedSB achieves state-of-the-art accuracy on the PACS dataset (83.31%), removing the smoothing component results in a 2.3% drop in performance (second row of Table 4.6). Additionally, utilizing the budgeting technique results in a 2.17% gain in performance compared to the baseline. We observe that performance declines with the removal of each component, demonstrating the effectiveness of each component in our method.

Table 4.6: FedSB with ablations results on on PACS. Removal of each component in FedSB results in a drop in performance in the image recognition task

Smoothing	Budget	P	A	C	S	Ave.
–	–	91.67	79.25	70.46	75.98	79.34
–	✓	93.93	80.25	76.63	77.22	81.51
✓	–	93.51	**82.83**	75.14	82.77	83.31
✓	✓	**94.19**	81.80	**75.28**	**83.52**	**83.81**

4.3.2 Sensitivity Analysis

We also report the effect of varying , and S on the overall performance in Tables 4.7, and Table 4.8. Changing the value of from 0:1 to 0:3 results in minimal performance variations on the PACS dataset, with the maximum drop being 0.33% after an average of three runs. It can be understood that as long as any level of uncertainly is injected during training with the use of *epsilon*, the performance experiences marked improvements.

We conduct the same experiment with different values for S, setting it to 30, 45, and 60 batches on the PACS dataset. Table 4.8 demonstrates the robustness of using different budget values. It can be seen that our method does not exhibit high sensitivity to either hyper-parameter.

Table 4.7: Impact of varying on FedSB accuracy.

	P	A	C	S	Ave.
= 0:1	**93.51**	**82.83**	75.14	**82.77**	**83.31**
= 0:3	93.41	81.85	**75.73**	82.39	83.35
= 0:2	93.07	80.76	75.44	82.63	82.98

Table 4.8: Impact of varying S on FedSB accuracy.

S	P	A	C	S	Ave.
$S = 30B$	91.34	80.25	**76.63**	77.22	**81.51**
$S = 45B$	**92.40**	**81.12**	72.08	75.48	80.27
$S = 60B$	91.87	80.61	73.70	**78.02**	81.05

4.3.3 Performance Using Transformer-Based Backbones

We also evaluate FedSB while using transformer-based backbones. The evaluations using ViT-b/16 and Vit-b/32 [89] backbones are presented in Table 4.9 for PACS and Table 4.10 for OfficeHome, where we observe that FedSB continues to outperform the baseline on both backbones and datasets. For example, using the ViT-b/16 backbone, FedSB achieves an average accuracy of 88.56%, compared to 87.24% for FedAvg. Similarly, with the ViT-b/32 backbone, FedSB achieves an average accuracy of 85.97%, outperforming FedAvg's 84.47%. A similar pattern is observed on the OfficeHome dataset. Using the ViT-b/16 backbone, FedSB achieves an average accuracy of 75.73%, slightly outperforming FedAvg's 75.27%. With the ViT-b/32 backbone, FedSB continues to outperform FedAvg, achieving an average accuracy of 69.36%, compared to 68.64% for FedAvg. These results highlight the adaptability and robustness of the FedSB framework when using transformer-based architectures, further demonstrating its efficacy across diverse neural network backbones. As Vision Transformers continue to gain traction in the machine learning community, incorporating these backbones into federated learning setups like FedSB can lead to enhanced generalization performance, especially in scenarios involving complex, heterogeneous datasets. The number of global communication rounds while using transformer-based backbones for the PACS and OfficeHome datasets is 20, and 10, respectively.

Table 4.9: Accuracy of FedAvg and FedSB on the PACS dataset using ViT backbones.

		PACS				
Backbone	Method	P	A	C	S	Ave.
ViT-b/16	FedAvg [1]	**98.88**	90.65	76.56	82.89	87.24
	FebSB (ours)	98.84	**91.49**	**80.43**	**83.48**	**88.56**
ViT-b/32	FedAvg [1]	96.66	84.57	77.46	**79.01**	84.47
	FebSB (ours)	**96.96**	**87.89**	**78.07**	78.43	**85.97**

Table 4.10: Accuracy of FedAvg and FedSB on the OfficeHome dataset using ViT backbones.

		O ceHome				
		P	A	C	R	Ave.
ViT-b/16	FedAvg [1]	81.92	74.63	60.76	83.76	75.27
	FebSB (ours)	**82.46**	**75.17**	**60.80**	**84.52**	**75.73**
ViT-b/32	FedAvg [1]	**74.93**	64.82	57.47	77.33	68.64
	FebSB (ours)	74.57	**65.65**	**58.87**	**78.36**	**69.36**

Chapter 5

Conclusion

5.1 Summary

In this thesis, we proposed two novel approaches for Federated Domain Generalization (FDG), addressing both unsupervised and supervised learning settings. We evaluated our methods on various datasets and showed the effectiveness of our proposed methods through extensive experiments. Specifically, this work focuses on two main problems, gradient misalignment due to domain shift in unsupervised federated settings and the overconfidence of local clients, as well as the imbalanced training contribution of local clients throughout training in the supervised setup.

In Chapter 3, we first introduced Federated Unsupervised Domain Generalization using Global and Local Alignment of Gradients to handle domain shifts in federated learning without labeled data. By aligning gradients both locally at the client level and globally at the server level, FedGaLA enables the learning of domain-invariant features, leading to improved generalization across unseen domains. Extensive experiments on four commonly used multi-domain datasets showed that FedGaLA outperformed existing baselines, confirming its effectiveness. Ablation studies and sensitivity

analyses highlighted the importance of both local and global alignment components.

In Chapter 4, we proposed Federated Domain Generalization with Label Smoothing and Balanced Decentralized Training to address data heterogeneity in supervised federated learning. FedSB leverages label smoothing to prevent overfitting to domain-specific features and introduces a decentralized budgeting mechanism to ensure balanced training contributions from clients. Experiments across four datasets demonstrated that FedSB achieved state-of-the-art performance on three out of four datasets, validating its robustness in handling data heterogeneity.

Both approaches offer solutions to critical challenges in federated learning, improving generalization across diverse and unseen domains while preserving data privacy. The methods contribute to the advancement of the field by addressing key limitations in existing approaches, positioning them as state-of-the-art solutions in both unsupervised and supervised federated domain generalization.

5.2 Future Work

There are several directions for future research stemming from this work. For FedGaLA, one potential avenue is to improve the communication efficiency of the framework. Currently, the alignment of gradients at both the client and server levels involves some communication overhead, which could be optimized. Additionally, the application of FedGaLA in other federated learning scenarios involving different types of data and tasks, such as time-series data or more complex hierarchical models, can be explored. Additionally, this method can be extended to supervised and semi-supervised domain generalization and adaptation where there is available information about the target domain.

For FedSB, future work can focus on extending the approach to more complex and diverse federated learning settings. This includes integrating FedSB with advanced techniques such as differential privacy and model compression to make the approach even more scalable and robust. By incorporating privacy-preserving methods, the continued protection of client data can be ensured while maintaining high levels of model performance. Another avenue for improvement involves exploring more sophisticated mechanisms for balancing training contributions from clients with highly imbalanced datasets. Similarly, this work could be adapted to domain adaptation tasks.

Bibliography

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Arti cial Intelligence and Statistics*, 2017, pp. 1273–1282.

[2] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, 2021.

[3] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 19 586–19 597.

[4] Z. Charles, Z. Garrett, Z. Huo, S. Shmulyian, and V. Smith, "On large-cohort training for federated learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 20 461–20 475, 2021.

[5] Q. Liu, C. Chen, J. Qin, Q. Dou, and P.-A. Heng, "FedDG: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1013–1023.

[6] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[7] B. Liu, N. Lv, Y. Guo, and Y. Li, "Recent advances on federated learning: A systematic survey," *Neurocomputing*, p. 128019, 2024.

[8] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Learning multiple visual domains with residual adapters," *Advances in neural information processing systems*, vol. 30, 2017.

[9] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7765–7773.

[10] F. Lai, X. Zhu, H. V. Madhyastha, and M. Chowdhury, "Oort: Efficient federated learning via guided participant selection," in *15th fUSENIXg Symposium on Operating Systems Design and Implementation (fOSDIg 21)*, 2021, pp. 19–35.

[11] C. Li, X. Zeng, M. Zhang, and Z. Cao, "Pyramidfl: A fine-grained client selection framework for efficient federated learning," in *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, 2022, pp. 158–171.

[12] D. Li and J. Wang, "Fedmd: Heterogenous federated learning via model distillation," *arXiv preprint arXiv:1910.03581*, 2019.

[13] E. Diao, J. Ding, and V. Tarokh, "Heterofl: Computation and communication efficient federated learning for heterogeneous clients," *arXiv preprint arXiv:2010.01264*, 2020.

[14] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International conference on arti cial intelligence and statistics*. PMLR, 2020, pp. 2938–2948.

[15] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/60a6c4002cc7b29142def8871531281a-Paper.pdf

[16] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, "Data poisoning attacks against federated learning systems," in *Computer security{ESORICs 2020: 25th European symposium on research in computer security, ESORICs 2020, guildford, UK, September 14{18, 2020, proceedings, part i 25*. Springer, 2020, pp. 480–501.

[17] C. Xie, K. Huang, P.-Y. Chen, and B. Li, "Dba: Distributed backdoor attacks against federated learning," in *International conference on learning representations*, 2019.

[18] Z. Zhang, A. Panda, L. Song, Y. Yang, M. Mahoney, P. Mittal, R. Kannan, and J. Gonzalez, "Neurotoxin: Durable backdoors in federated learning," in *International Conference on Machine Learning*. PMLR, 2022, pp. 26 429–26 446.

[19] J. Zhu and M. Blaschko, "R-gap: Recursive gradient attack on privacy," *arXiv preprint arXiv:2010.07733*, 2020.

[20] C. Dwork, "Differential privacy: A survey of results," in *International conference on theory and applications of models of computation*. Springer, 2008, pp. 1–19.

[21] N. Agarwal, P. Kairouz, and Z. Liu, "The skellam mechanism for differentially private federated learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 5052–5064, 2021.

[22] P. Kairouz, Z. Liu, and T. Steinke, "The distributed discrete gaussian mechanism for federated learning with secure aggregation," in *International Conference on Machine Learning*. PMLR, 2021, pp. 5201–5212.

[23] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE transactions on information forensics and security*, vol. 15, pp. 3454–3469, 2020.

[24] R. Xu, N. Baracaldo, Y. Zhou, A. Anwar, and H. Ludwig, "Hybridalpha: An efficient approach for privacy-preserving federated learning," in *Proceedings of the 12th ACM workshop on arti cial intelligence and security*, 2019, pp. 13–23.

[25] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "*f*BatchCrypt*g*: Efficient homomorphic encryption for *f*Cross-Silo*g* federated learning," in *2020 USENIX annual technical conference (USENIX ATC 20)*, 2020, pp. 493–506.

[26] C. Liu, S. Chakraborty, and D. Verma, "Secure model fusion for distributed learning using partial homomorphic encryption," *Policy-Based Autonomic Data Governance*, pp. 154–179, 2019.

[27] Y. J. Cho, J. Wang, and G. Joshi, "Client selection in federated learning: Convergence analysis and power-of-choice selection strategies," *arXiv preprint arXiv:2010.01243*, 2020.

[28] F. Li, J. Liu, and B. Ji, "Combinatorial sleeping bandits with fairness constraints," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 3, pp. 1799–1813, 2019.

[29] Z. Song, H. Sun, H. H. Yang, X. Wang, Y. Zhang, and T. Q. Quek, "Reputation-based federated learning for secure wireless networks," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1212–1226, 2021.

[30] J. Zhang, C. Li, A. Robles-Kelly, and M. Kankanhalli, "Hierarchically fair federated learning," *arXiv preprint arXiv:2004.10386*, 2020.

[31] S. Gollapudi, K. Kollias, D. Panigrahi, and V. Pliatsika, "Profit sharing and efficiency in utility games," in *25th Annual European Symposium on Algorithms (ESA 2017)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2017.

[32] A. Ghorbani and J. Zou, "Data shapley: Equitable valuation of data for machine learning," in *International conference on machine learning*. PMLR, 2019, pp. 2242–2251.

[33] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[34] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 713–10 722.

[35] D. Liao, X. Gao, Y. Zhao, H. Dai, L. Li, K. Wang, K. Ye, Y. Wang, and C.-Z. Xu, "Feddrop: Trajectory-weighted dropout for efficient federated learning," 2021.

[36] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *Advances in neural information processing systems*, vol. 33, pp. 7611–7623, 2020.

[37] J. Nguyen, K. Malik, H. Zhan, A. Yousefpour, M. Rabbat, M. Malek, and D. Huba, "Federated learning with buffered asynchronous aggregation," in *International Conference on Artificial Intelligence and Statistics*, 2022, pp. 3581–3607.

[38] J. Zhang, Y. Hua, H. Wang, T. Song, Z. Xue, R. Ma, and H. Guan, "FedALA: Adaptive local aggregation for personalized federated learning," in *Proceedings of the Association for the Advancement of Artificial Intelligence*, vol. 37, 2023, pp. 11 237–11 244.

[39] X. Cao, M. Fang, J. Liu, and N. Z. Gong, "Fltrust: Byzantine-robust federated learning via trust bootstrapping," *arXiv:2012.13995*, 2020.

[40] J. Xu, S.-L. Huang, L. Song, and T. Lan, "Byzantine-robust federated learning through collaborative malicious gradient filtering," in *IEEE International Conference on Distributed Computing Systems*, 2022, pp. 1223–1235.

[41] Y. Li, X. Wang, R. Zeng, P. K. Donta, I. Murturi, M. Huang, and S. Dustdar, "Federated domain generalization: A survey," *arXiv:2306.01334*, 2023.

[42] R. Bai, S. Bagchi, and D. I. Inouye, "Benchmarking algorithms for federated domain generalization," *arXiv preprint arXiv:2307.04942*, 2023.

[43] A. T. Nguyen, P. Torr, and S. N. Lim, "FedSR: A simple and effective domain generalization method for federated learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 38 831–38 843, 2022.

[44] L. Zhang, X. Lei, Y. Shi, H. Huang, and C. Chen, "Federated learning with domain generalization," *arXiv preprint arXiv:2111.10487*, 2021.

[45] J. Chen, M. Jiang, Q. Dou, and Q. Chen, "Federated domain generalization for image recognition via cross-client style transfer," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 361–370.

[46] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1501–1510.

[47] G. Wu and S. Gong, "Collaborative optimization and aggregation for decentralized domain generalization and adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6484–6493.

[48] Y. Tan, G. Long, L. Liu, T. Zhou, Q. Lu, J. Jiang, and C. Zhang, "Fedproto: Federated prototype learning across heterogeneous clients," in *Proceedings of the AAAI Conference on Arti cial Intelligence*, vol. 36, no. 8, 2022, pp. 8432–8440.

[49] X. Yu, D. Wang, M. J. McKeown, and Z. J. Wang, "Contrastive-enhanced domain generalization with federated learning," *IEEE Transactions on Arti cial Intelligence*, vol. 5, no. 4, pp. 1525–1532, 2023.

[50] Y. Sun, N. Chong, and H. Ochiai, "Feature distribution matching for federated domain generalization," in *Asian Conference on Machine Learning*, 2023, pp. 942–957.

[51] J. Park, D.-J. Han, J. Kim, S. Wang, C. G. Brinton, and J. Moon, "StableFDG: Style and attention based learning for federated domain generalization," *Advances in Neural Information Processing Systems*, 2023.

[52] M. Bartholet, T. Kim, A. Beuret, S.-Y. Yun, and J. M. Buhmann, "Non-linear fusion in federated learning: A hypernetwork approach to federated domain generalization," *arXiv:2402.06974*, 2024.

[53] R. Zhang, Q. Xu, J. Yao, Y. Zhang, Q. Tian, and Y. Wang, "Federated domain generalization with generalization adjustment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 3954–3963.

[54] Y. Jin, X. Wei, Y. Liu, and Q. Yang, "Towards utilizing unlabeled data in federated learning: A survey and prospective," *arXiv:2002.11545*, 2020.

[55] B. van Berlo, A. Saeed, and T. Ozcelebi, "Towards federated unsupervised representation learning," in *Proceedings of the ACM International Workshop on Edge Systems, Analytics and Networking*, 2020, pp. 31–36.

[56] W. Zhuang, X. Gan, Y. Wen, S. Zhang, and S. Yi, "Collaborative unsupervised visual representation learning from decentralized data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4912–4921.

[57] W. Zhuang, Y. Wen, and S. Zhang, "Divergence-aware federated self-supervised learning," in *International Conference on Learning Representations*, 2022.

[58] S. Han, S. Park, F. Wu, S. Kim, C. Wu, X. Xie, and M. Cha, "FedX: Unsupervised federated learning with cross knowledge distillation," in *European Conference on Computer Vision*, 2022, pp. 691–707.

[59] R. Zhang, Q. Xu, J. Yao, Y. Zhang, Q. Tian, and Y. Wang, "Federated domain generalization with generalization adjustment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 3954–3963.

[60] F. Pourpanah, M. Molahasani, M. Soltany, M. Greenspan, and A. Etemad, "Federated unsupervised domain generalization using global and local alignment of gradients," *arXiv preprint arXiv:2405.16304*, 2024.

[61] R. Bai, S. Bagchi, and D. I. Inouye, "Benchmarking algorithms for federated domain generalization," in *The International Conference on Learning Representations*, 2024.

[62] L. Mansilla, R. Echeveste, D. H. Milone, and E. Ferrante, "Domain generalization via gradient surgery," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6630–6638.

[63] Y. Shi, J. Seely, P. Torr, S. N, A. Hannun, N. Usunier, and G. Synnaeve, "Gradient matching for domain generalization," in *International Conference on Learning Representations*, 2022.

[64] A. Rame, C. Dancette, and M. Cord, "Fishr: Invariant gradient variances for out-of-distribution generalization," in *International Conference on Learning Representations*, 2022.

[65] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5542–5550.

[66] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5018–5027.

[67] S. Beery, G. Van Horn, and P. Perona, "Recognition in terra incognita," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 456–473.

[68] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1406–1415.

[69] X. Zhang, L. Zhou, R. Xu, P. Cui, Z. Shen, and H. Liu, "Towards unsupervised domain generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4910–4920.

[70] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning*, 2020, pp. 1597–1607.

[71] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[72] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.

[73] I. Gulrajani and D. Lopez-Paz, "In search of lost domain generalization," in *International Conference on Learning Representations*, 2021.

[74] Z. Feng, C. Xu, and D. Tao, "Self-supervised representation learning from multi-domain data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3245–3255.

[75] R. Zhang, P. Isola, and A. A. Efros, "Split-brain autoencoders: Unsupervised learning by cross-channel prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1058–1067.

[76] A. Kolesnikov, X. Zhai, and L. Beyer, "Revisiting self-supervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1920–1929.

[77] Y. Guo, K. Guo, X. Cao, T. Wu, and Y. Chang, "Out-of-distribution generalization of federated learning via implicit invariant relationships," in *International Conference on Machine Learning*. PMLR, 2023, pp. 11 905–11 933.

[78] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.

[79] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 271–21 284, 2020.

[80] Q. Hu, X. Wang, W. Hu, and G.-J. Qi, "Adco: Adversarial contrast for efficient learning of unsupervised representations from self-trained negative adversaries," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1074–1083.

[81] S. T. Arasteh, C. Kuhl, M.-J. Saehn, P. Isfort, D. Truhn, and S. Nebelung, "Mind the gap: Federated learning broadens domain generalization in diagnostic ai models," *arXiv:2310.00757*, 2023.

[82] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[83] Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu *et al.*, "Gpipe: Efficient training of giant neural networks

using pipeline parallelism," *Advances in neural information processing systems*, vol. 32, 2019.

[84] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proceedings of the aaai conference on arti cial intelligence*, vol. 33, no. 01, 2019, pp. 4780–4789.

[85] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8697–8710.

[86] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," *arXiv preprint arXiv:1612.02695*, 2016.

[87] C. Fang, Y. Xu, and D. N. Rockmore, "Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1657–1664.

[88] I. Gulrajani and D. Lopez-Paz, "In search of lost domain generalization," *arXiv preprint arXiv:2007.01434*, 2020.

[89] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[90] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain generalization with mixstyle," *arXiv preprint arXiv:2104.02008*, 2021.

[91] "xkcd: Thesis defense," https://xkcd.com/1403/, (Accessed on 10/20/2017).

[92] "Big bang - warm kitty, soft kitty (sheldon's lullaby sick song) instrumental version lyrics — metrolyrics," http://www.metrolyrics.com/warm-kitty-soft-kitty-sheldons-lullaby-sick-song-instrumental-version-lyrics-big-bang.html?ModPagespeed=noscript, (Accessed on 10/20/2017).

[93] M. Shaw, "Writing good software engineering research papers," in *Software Engineering, 2003. Proceedings. 25th International Conference on*. IEEE, 2003, pp. 726–736.

[94] B. Paltridge, "Thesis and dissertation writing: an examination of published advice and actual practice," *English for Speci c Purposes*, vol. 21, no. 2, pp. 125–143, 2002.

[95] U. Eco, *How to write a thesis*. MIT Press, 2015.

[96] J. Wang, C. Lan, C. Liu, Y. Ouyang, T. Qin, W. Lu, Y. Chen, W. Zeng, and P. S. Yu, "Generalizing to unseen domains: A survey on domain generalization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 8, pp. 8052–8072, 2023.

[97] L. Alzubaidi, J. Bai, A. Al-Sabaawi, J. Santamaría, A. Albahri, B. S. N. Al-dabbagh, M. A. Fadhel, M. Manoufali, J. Zhang, A. H. Al-Timemy *et al.*, "A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications," *Journal of Big Data*, vol. 10, no. 1, p. 46, 2023.

[98] X. Liang, Y. Lin, H. Fu, L. Zhu, and X. Li, "Rscfed: Random sampling consensus federated semi-supervised learning," in *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, 2022, pp. 10 154–10 163.

[99] W. Jeong, J. Yoon, E. Yang, and S. J. Hwang, "Federated semi-supervised learning with inter-client consistency & disjoint learning," *arXiv:2006.12097*, 2020.

[100] S. Yu, P. Wu, P. P. Liang, R. Salakhutdinov, and L.-P. Morency, "PACS: A dataset for physical audiovisual commonsense reasoning," in *European Conference on Computer Vision*, 2022, pp. 292–309.

[101] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *European conference on computer vision*, 2016, pp. 69–84.

[102] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *European conference on computer vision*, 2016, pp. 649–666.

[103] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang, "Self-supervised learning: Generative or contrastive," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 1, pp. 857–876, 2021.

[104] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International Conference on Machine Learning*, 2015, pp. 1180–1189.

[105] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *International Conference on Machine Learning*, 2019, pp. 4615–4625.

[106] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1142–1154, 2022.

[107] M. Mendieta, T. Yang, P. Wang, M. Lee, Z. Ding, and C. Chen, "Local learning matters: Rethinking data heterogeneity in federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8397–8406.

[108] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021.

[109] I. Tenison, S. A. Sreeramadas, V. Mugunthan, E. Oyallon, I. Rish, and E. Belilovsky, "Gradient masked averaging for federated learning," *Transactions on Machine Learning Research*, 2023.

[110] R. Taori, A. Dave, V. Shankar, N. Carlini, B. Recht, and L. Schmidt, "Measuring robustness to natural distribution shifts in image classification," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18 583–18 599, 2020.

[111] F. Pourpanah and A. Etemad, "Exploring the landscape of ubiquitous in-home health monitoring: A comprehensive survey," *arXiv:2306.12660*, 2023.

[112] D. Kim, H. Bian, C. K. Chang, L. Dong, J. Margrett *et al.*, "In-home monitoring technology for aging in place: scoping review," *Interactive Journal of Medical Research*, vol. 11, no. 2, p. e39005, 2022.

[113] Y. Almalioglu, M. Turan, N. Trigoni, and A. Markham, "Deep learning-based robust positioning for all-weather autonomous driving," *Nature Machine Intelligence*, vol. 4, no. 9, pp. 749–760, 2022.

[114] S. Jabeen, X. Li, M. S. Amin, O. Bourahla, S. Li, and A. Jabbar, "A review on methods and applications in multimodal deep learning," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 19, no. 2s, pp. 1–41, 2023.

[115] D. Shenaj, E. Fanì, M. Toldo, D. Caldarola, A. Tavera, U. Michieli, M. Ciccone, P. Zanuttigh, and B. Caputo, "Learning across domains and devices: Style-driven source-free domain adaptation in clustered federated learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 444–454.

[116] G. Rizzoli, D. Shenaj, and P. Zanuttigh, "Source-free domain adaptation for rgb-d semantic segmentation with vision transformers," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 615–624.

[117] W. Zhuang, X. Gan, Y. Wen, and S. Zhang, "Easyfl: A low-code federated learning platform for dummies," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 13 740–13 754, 2022.

[118] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 750–15 758.

[119] W. Huang, M. Ye, Z. Shi, H. Li, and B. Du, "Rethinking federated learning with domain shift: A prototype view," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 312–16 322.

[120] J. Yu and K. Spiliopoulos, "Normalization effects on deep neural networks," *arXiv preprint arXiv:2209.01018*, 2022.

[121] H. Qi, J. Zhou, and H. Wang, "A note on factor normalization for deep neural network models," *Scienti c Reports*, vol. 12, no. 1, p. 5909, 2022.

[122] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, and L. Shao, "Normalization techniques in training dnns: Methodology, analysis and application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[123] Z. Goldfeld and K. Greenewald, "Sliced mutual information: A scalable measure of statistical dependence," *Advances in Neural Information Processing Systems*, vol. 34, pp. 17 567–17 578, 2021.

[124] J. Gao, Y. Hua, G. Hu, C. Wang, and N. M. Robertson, "Reducing distributional uncertainty by mutual information maximisation and transferable feature learning," in *European Conference on Computer Vision*, 2020, pp. 587–605.

[125] W. Menapace, S. Lathuilière, and E. Ricci, "Learning to cluster under domain shift," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 736–752.

[126] Z. Wang, Y. Luo, R. Qiu, Z. Huang, and M. Baktashmotlagh, "Learning to diversify for single domain generalization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 834–843.

[127] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," in *International Conference on Learning Representations*, 2020.

[128] E. Štrumbelj and I. Kononenko, "Explaining prediction models and individual predictions with feature contributions," *Knowledge and information systems*, vol. 41, pp. 647–665, 2014.

[129] ——, "A general method for visualizing and explaining black-box regression models," in *International Conference on Adaptive and Natural Computing Algorithms*. Springer, 2011, pp. 21–30.

[130] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning distance functions using equivalence relations," in *Proceedings of the International Conference on Machine Learning*, 2003, pp. 11–18.

[131] Z. Wang, C. Wu, Y. Yang, and Z. Li, "Learning transformation-predictive representations for detection and description of local features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 11 464–11 473.

[132] E. Jiang, Y. J. Zhang, and S. Koyejo, "Principled federated domain adaptation: Gradient projection and auto-weighting," in *The Twelfth International Conference on Learning Representations*, 2024.

[133] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient surgery for multi-task learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5824–5836, 2020.

[134] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[135] X. Zhang, L. Zhou, R. Xu, P. Cui, Z. Shen, and H. Liu, "Towards unsupervised domain generalization," *arXiv preprint arXiv:2107.06219*, 2021.

[136] K. Zhou, Y. Yang, T. Hospedales, and T. Xiang, "Learning to generate novel domains for domain generalization," in *Computer Vision{ECCV 2020: 16th European Conference, Glasgow, UK, August 23{28, 2020, Proceedings, Part XVI 16*. Springer, 2020, pp. 561–578.

[137] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, "Domain generalization: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4396–4415, 2022.

[138] S. Wang, L. Yu, C. Li, C.-W. Fu, and P.-A. Heng, "Learning from extrinsic and intrinsic supervisions for domain generalization," in *European Conference on Computer Vision*. Springer, 2020, pp. 159–176.

[139] Q. Xu, R. Zhang, Y. Zhang, Y. Wang, and Q. Tian, "A fourier-based framework for domain generalization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 14 383–14 392.

[140] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Fedavg with fine tuning: Local updates lead to representation learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 10 572–10 586, 2022.

[141] S. Seo, Y. Suh, D. Kim, G. Kim, J. Han, and B. Han, "Learning to optimize domain specific normalization for domain generalization," in *Computer Vision{ ECCV 2020: 16th European Conference, Glasgow, UK, August 23{28, 2020, Proceedings, Part XXII 16*. Springer, 2020, pp. 68–83.

[142] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," *Advances in neural information processing systems*, vol. 19, 2006.

[143] X. Peng, Z. Huang, Y. Zhu, and K. Saenko, "Federated adversarial domain adaptation," *arXiv preprint arXiv:1911.02054*, 2019.

[144] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[145] F. Zhang, Y. Zhang, S. Ji, and Z. Han, "Secure and decentralized federated learning framework with non-iid data based on blockchain," *Heliyon*, vol. 10, no. 5, 2024.

[146] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International conference on machine learning*. PMLR, 2017, pp. 1321–1330.

[147] C.-B. Zhang, P.-T. Jiang, Q. Hou, Y. Wei, Q. Han, Z. Li, and M.-M. Cheng, "Delving deep into label smoothing," *IEEE Transactions on Image Processing*, vol. 30, pp. 5984–5996, 2021.

[148] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[149] N. Shi, F. Lai, R. Al Kontar, and M. Chowdhury, "Fed-ensemble: Ensemble models in federated learning for improved generalization and uncertainty quantification," *IEEE Transactions on Automation Science and Engineering*, 2023.

[150] N. Koutsoubis, Y. Yilmaz, R. P. Ramachandran, M. Schabath, and G. Rasool, "Privacy preserving federated learning in medical imaging with uncertainty estimation," *arXiv preprint arXiv:2406.12815*, 2024.

[151] L. Qu, Y. Zhou, P. P. Liang, Y. Xia, F. Wang, E. Adeli, L. Fei-Fei, and D. Rubin, "Rethinking architecture design for tackling data heterogeneity in federated learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 061–10 071.

[152] Z. Yang, Y. Zhang, Y. Zheng, X. Tian, H. Peng, T. Liu, and B. Han, "Fedfed: Feature distillation against data heterogeneity in federated learning," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[153] Y. Wang, H. Fu, R. Kanagavelu, Q. Wei, Y. Liu, and R. S. M. Goh, "An aggregation-free federated learning for tackling data heterogeneity," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 26 233–26 242.

[154] A. Mora, A. Bujari, and P. Bellavista, "Enhancing generalization in federated learning with heterogeneous data: A comparative literature review," *Future Generation Computer Systems*, 2024.

[155] J. Zhang, Y. Liu, Y. Hua, and J. Cao, "Fedtgp: Trainable global prototypes with adaptive-margin-enhanced contrastive learning for data and model heterogeneity in federated learning," in *Proceedings of the AAAI Conference on Arti cial Intelligence*, vol. 38, no. 15, 2024, pp. 16 768–16 776.

[156] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" *Advances in neural information processing systems*, vol. 32, 2019.