2024

# Efficient and Effective Deep Learning Methods for Computer Vision in Centralized and Distributed Applications

Matias Mendieta
*University of Central Florida*

# EFFICIENT AND EFFECTIVE DEEP LEARNING METHODS FOR COMPUTER VISION IN CENTRALIZED AND DISTRIBUTED APPLICATIONS

by

MATÍAS ANDRÉS MENDIETA
B.S. University of North Carolina at Charlotte, 2019
M.S. University of North Carolina at Charlotte, 2020

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida

Summer Term
2024

Major Professor: Chen Chen

# ABSTRACT

In the rapidly advancing field of computer vision, deep learning has driven significant technological transformations. However, the widespread deployment of these technologies often encounters efficiency challenges, such as high memory usage, demanding computational resources, and extensive communication overhead. Efficiency has become crucial for both centralized and distributed applications of deep learning, ensuring scalability, real-world applicability, and broad accessibility. In distributed settings, federated learning (FL) enables collaborative model training across multiple clients while maintaining data privacy. Despite its promise, FL faces challenges due to clients' constraints in memory, computational power, and bandwidth. Centralized training systems also require high efficiency, where optimizing compute resources during training and inference, as well as label efficiency, can significantly impact the performance and practicality of such models. Addressing these efficiency challenges in both federated learning and centralized training systems promises to provide significant advancements, enabling more extensive and effective deployment of machine learning models across various domains.

To this end, this dissertation addresses many key challenges. First, in federated learning, a novel method is introduced to optimize local model performance while reducing memory and computational demands. Additionally, a novel approach is presented to reduce communication costs by minimizing model update frequency across clients through the use of generative models. In the centralized domain, this dissertation further develops a novel training paradigm for geospatial foundation models using a multi-objective continual pretraining strategy. This improves label efficiency and significantly reduces computational requirements for training large-scale models. Overall, this dissertation advances deep learning efficiency by improving memory utilization, computational demands, and communication efficiency, essential for scalable and effective application of deep learning in both distributed and centralized environments.

# ACKNOWLEDGMENTS

Praise be to my Lord and Savior, Jesus Christ. I would like to extend my sincere gratitude to my advisor, Dr. Chen Chen, for providing me the opportunity to conduct this research under his insightful supervision. Many thanks to my committee members, Dr. Mubarak Shah, Dr. Ser-Nam Lim, and Dr. Mingjie Lin, for their insightful guidance throughout the dissertation proposal and defense process. I am also grateful to my peers at the CRCV for the stimulating discussions, enjoyable collaborations, and friendship. Lastly, I am deeply thankful to my parents and siblings for their unwavering support and encouragement.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER 1: INTRODUCTION

In recent years, the application of deep learning in computer vision has brought about profound changes to the global technological landscape. The advent of AlexNet marked a turning point, catalyzing a rapid expansion in the adoption and utilization of computer vision technologies in a plethora of industries. However, as deep learning methodologies have become more prevalent and diversified in their applications, there has been a corresponding escalation in the complexity and resource requirements of these approaches. This includes the proliferation of larger model architectures, the demand for increased volumes of training data, and the necessity for heightened computational power to effectively train and deploy these models.

Therefore, efficiency has emerged as a crucial factor for both centralized and distributed applications of deep learning technologies. This is essential for ensuring scalability, real-world applicability, and ultimately facilitating its democratization and widespread accessibility. For example, in distributed settings, federated learning (FL) presents a powerful strategy by enabling collaborative model training across multiple clients while upholding the privacy of their data. This approach, while promising, underscores pivotal challenges due to each client's operation under significant constraints, including limited memory, computational power, and bandwidth. These constraints become particularly severe in large-scale FL systems where communication overhead escalates into a critical bottleneck, underscoring the necessity for enhanced efficiency in memory, computation, and communication to render distributed paradigms like federated learning both viable and effective. Conversely, centralized training systems also demand high efficiency, particularly focusing on optimizing compute resources during training or inference phases. In specialized fields such as remote sensing, where labels are often scarce for downstream tasks, label efficiency is critical for success. Enhancing both compute and label efficiency can significantly influence the performance and practicality of centralized training models. Addressing these efficiency challenges holistically

in both federated learning and centralized training systems can lead to significant advancements, supporting more extensive and effective application of machine learning models across varied domains.

In this dissertation, we endeavor to enhance efficiency across multiple dimensions. We begin by refining efficiency in distributed training, addressing inherent challenges by proposing effective solutions. Subsequently, we extend our investigation to the centralized paradigm, addressing training and labeling costs. The subsequent sections will introduce these settings and research motivations, leading to the presentation of novel methods for efficient deep learning in computer vision.

## 1.1 Challenges in Federated Learning

Federated learning (FL) is a distributed machine learning technique that enables multiple clients to participate in the training process in a privacy-preserving manner. In FL, each client trains a local model on its own data and sends the model to a central server. The server combines these updates to improve the global model, which is then sent back to the clients. This approach ensures that the clients' data is kept private while enabling the central server to learn from the collective knowledge of all participating users [45]. However, FL poses significant challenges in terms compute, memory, communication cost and optimization speed. We will discuss the factors that contribute to these challenges in the following subsections.

### *1.1.1 Efficient Solutions for Data Heterogeneity in Federated Learning*

In the FL setting, participating clients are typically deployed in a variety of environments or owned by a diverse set of users. Therefore, the distribution of each client's local data can vary considerably (i.e., data heterogeneity). This non-IID data distribution among participating devices in

FL makes optimization particularly challenging. As each client trains locally on their own data, they step towards their respective local minimum. However, this local convergence point may not be well aligned with the objective of the global model (that is, the model being learned though aggregation at the central server). *Therefore, the client model often drifts away from the ideal global optimization point and overfits to its local objective.* When such client drifting occurs, the performance of the central aggregated model is hindered [46, 56].

One straight-forward solution to this phenomenon is to simply limit the number of local training epochs performed between central aggregation steps. However, this severely hinders the convergence speed of the FL system, and many communication rounds are required to achieve adequate performance. The time to convergence and immense communication overhead incurred by such an approach are often not tolerable for real-world distributed systems. Therefore, effectively addressing data heterogeneity is of paramount concern in federated learning.

Many algorithmic solutions to this problem have been proposed in the literature [81, 58, 49, 3]. These strategies typically focus on mitigating the effects of data heterogeneity across clients by introducing a variety of *proximal terms* to restrain local updates with respect to the global model. *However, by restraining the drift, they also inherently limit the local convergence potential; less novel information is gathered per communication round.* Consequently, many current FL algorithms do not provide stable performance improvements across different non-IID settings in comparison to classic baselines [56, 58], especially on vision tasks beyond the difficulty of MNIST [54]. Furthermore, existing methods have paid little attention to the resource constraints of the client, typically scarce for deployed FL edge devices, and in some cases incur considerable compute and/or memory overheads on the client in their effort to alleviate client drift. For example, the state-of-the-art (SOTA) method MOON performs well on federated image tasks, but to do so incurs a *$\sim$3x overhead in both memory and compute* compared to the standard FedAvg baseline [67].

#### *1.1.1.1 Motivation for Efficient Solutions*

In the centralized training paradigm, network generalization capability has been well studied to combat overfitting. Even in standard settings where the training and test data are drawn from a similar distribution, models still overfit on the training data if no precautions are taken. This effect is further intensified when the training and test data are of different distributions. Various regularization techniques are introduced to enforce learning generality during training and preserve suitable test performance. Similarly, overfitting to the local training data of each device in FL is detrimental to overall network performance, as the client drifting effect creates conflicting objectives among local models. *Thus, a focus on improving model generality should be of primary concern in the presence of data heterogeneity.* Improving local learning generality during training would inherently position the objective of the clients closer to the overall global objective. However, despite its intuitive motivations, this perspective has been overlooked by the bulk of current FL literature.

Therefore, in this paper, we propose rethinking approaches to data heterogeneity in terms of **local learning generality** rather than proximal restriction. Specifically, we carefully analyze the effectiveness of various data and structural regularization methods at reducing client drift and improving FL performance (Section 3.1). Utilizing second-order information and insights from out-of-distribution generality literature [76, 72], we identify theoretical indicators for successful FL optimization, and evaluate across a variety of FL settings for empirical validation.

Although some of the regularization methods perform well at mitigating client drift, *significant resource overheads* are still incurred to achieve the best performance (see Section 3.2). Therefore, we propose **FedAlign**, a distillation-based regularization method that promotes local learning generality while maintaining excellent resource efficiency. Specifically, FedAlign focuses on regularizing the Lipschitz constants of the final block in a network with respect to its representations. By fo-

cusing solely on the last block, we effectively regularize the portion of the network most prone to overfitting and keep additional resource needs to a minimum. Therefore, FedAlign achieves state-of-the-art accuracy on multiple datasets across a variety of FL settings, while requiring significantly less computation and memory overhead in comparison to other state-of-the-art methods.

In this dissertation, we fundamentally approach one of the most troublesome FL challenges (i.e. client drift caused by data heterogeneity) from a unique angle than any other previous work. Particularly, we do not focus on reparameterization tricks to maintain closeness to the central model, or adjust the aggregation scheme to mitigate the effects of non-IID data distributions. *Rather, we propose the rethinking of this problem from fundamental machine learning training principles.* In this way, we analyze the performance of standard regularization methods on FL and their effectiveness against data heterogeneity. Not only do we empirically analyze the performance of regularization methods in FL, we also propose to take a deeper look. Specifically, we inform our analysis with theoretical indicators of learning generality to provide insight into which methods are best and why. We find that Hessian eigenvalue/trace measurements and Hessian matching across clients to be meaningful indicators for optimal FL methods. Additionally, we perform a thorough ablation study across a variety of FL settings to understand the empirical effects of different methods. Our aim is to provide this valuable knowledge to the FL community to inspire new, productive research directions. Informed by our analysis and examining the pitfalls of previous methods, we propose FedAlign, which achieves competitive state-of-the-art accuracy while maintaining memory and computational efficiency.

### *1.1.2 Efficient Solutions for Communication in Federated Learning*

Communication cost is a major bottleneck in FL systems, as clients need to communicate frequently with the server over multiple rounds during the training process [45, 81, 3]. This leads

to a high communication overhead, making the process slow or simply infeasible. To overcome this challenge, **one-shot federated learning** has recently gained traction in the research community [29, 109, 112, 82]. In this setting, clients only communicate once with the server during the training process, significantly reducing the communication requirements. This approach not only improves the efficiency of the training process but also provides a better framework for privacy and application. Specifically, one-shot FL provides better security against eavesdropping attacks, where adversaries attempt to steal or tamper with the information being sent between clients and the server [60]. By only requiring one round of communication, one-shot FL significantly reduces the likelihood of such attacks. Furthermore, traditional multi-round training may not be a practical option in some cases, such as that of model markets [57]. In these scenarios, models are trained to convergence by a participating user, and simply made available as a pretrained model to potential buyers, without any option for iterative communication.

However, the significant challenge in federated learning still remains, and that is, the data heterogeneity problem as discussed previously [68, 49, 58, 45]. In FL, clients often have very different data distributions, making optimization particularly challenging across the federated system. In the one-shot setting, this is especially detrimental to performance. Without the luxury of multiple communication rounds, the resulting models will be significantly biased towards their narrow data distribution and difficult to reconcile into a global model. Knowledge distillation-based approaches have been studied in the literature in an attempt to address these problems [29, 57, 109]. Nonetheless, these methods still struggle immensely under high heterogeneity, resulting in large drops in performance.

Yet, another class of model is potentially well-suited for such heterogeneous distributions at the clients. Rather than simply employing discriminative models to train on the clients, one could instead leverage generative models. These generative models can then be gathered from the clients and inferenced on the server to form a dataset for global model training, eliminating the need

for the challenging reconciliation process required for discriminative models. [37] conducted a preliminary study of such a framework with conditional variational autoencoders (CVAEs) [85] for one-shot FL, but there is still much to investigate in this paradigm. *Specifically, we consider two primary research questions (RQ) in this work.*

***RQ1***. First, we explore the utility of diffusion models in federated learning and their potential for improving the performance of the one-shot FL process. Diffusion models [38] have recently emerged as prominent approaches for image generation, inspiring our investigation. We suggest that specific traits of diffusion models could provide advantages for one-shot FL, as discussed in Section 4.1. We then validate this hypothesis through comprehensive experiments with our approach, FedDiff, across various settings.

***RQ2***. Second, we investigate one-shot FL methods under provable privacy budgets with differential privacy (DP), as this aspect is not addressed by existing state-of-the-art (SOTA) one-shot FL works. Safeguarding model privacy is critical in this setting, as the client models obtained in one-shot FL can be reused multiple times or even traded in a model market. Furthermore, in light of recent work [9], we examine the potential memorization of diffusion models within our FedDiff approach and the effectiveness of DP as a mitigation strategy.

After studying these research questions, we further explore a simple technique for improving the performance of our FedDiff method under DP settings. We observe that the quality of generated samples may deteriorate under DP constraints, rendering some samples counterproductive to the training of the global model. To improve the quality and consistency of the synthetic data, we propose a straightforward filtering approach, termed Fourier Magnitude Filtering (FMF). FMF leverages sample magnitudes derived from the Fourier transform to guide the selection of valuable samples. The resulting filtered dataset substantially improves the utility of the generated data, particularly in challenging conditions, as detailed in Section 4.3.3.

In this dissertation, we contribute to the FL literature with the first study exploring diffusion models in one-shot federated learning. Our comprehensive investigation unveils the unique advantages inherent to diffusion models, which enhances the overall performance of one-shot FL while also addressing the significant challenges of data heterogeneity. We therefore establish a novel approach, FedDiff, that not only ensures superior model performance but also aligns with the core requirements of one-shot FL. We further study the privacy and utility of both discriminative and generative-based SOTA one-shot FL methods with DP guarantees under heterogeneous settings. We find that our FedDiff approach outperforms all other methods by a significant margin (from $\sim$5% to $\sim$20% across many datasets and settings), even when differential privacy is employed. Furthermore, while FedDiff performs very well, we note that sample quality is affected under DP. Therefore, to improve performance in such conditions, we propose a simple Fourier Magnitude Filtering (FMF) approach, which improves the effectiveness of the generated data for global model training by removing low-quality samples.

## 1.2 Challenges in Centralized Settings

In the domain of centralized training systems and neural networks, computational efficiency is of paramount importance for the practical application of deep learning technologies. The ability to process vast amounts of data quickly and effectively without excessive computational cost is critical, as it directly influences the scalability, accessibility, and sustainability of machine learning solutions. Furthermore, label efficiency emerges as a crucial aspect in domains where acquiring labeled data is inherently difficult or expensive. In such scenarios, the ability to train models with fewer labels without compromising the performance is invaluable. Techniques that enhance label efficiency, such as self-supervised learning and continual pretraining, are therefore instrumental in maximizing the utility of available data, reducing the cost and effort involved in dataset curation,

and facilitating the rapid adaptation of models to new tasks or environments. These efficiencies not only accelerate the advancement of neural network capabilities but also significantly mitigate the barriers to their adoption in new areas, ensuring that deep learning can deliver its transformative potential across industries.

### *1.2.1 Geospatial Continual Pretraining*

In centralized systems, particularly within the geospatial and remote sensing sector, optimizing computational and label efficiency is paramount. Geospatial technologies play crucial roles in diverse fields such as agriculture, urban planning, and disaster management by enhancing our understanding and interaction with Earth's systems. Progress in this domain can substantially improve our ability to understand the earth and how we interact with it. With the rising popularity of foundation models in vision and natural language, researchers have begun to investigate applying such principles to the geospatial domain in order to enhance the suitability and label efficiency of deep learning models in downstream tasks [69, 66, 17, 7]. In the literature, various works have explored two prominent approaches for introducing pretrained foundation models in geospatial applications. The first obvious approach is to leverage existing foundation models from the natural image domain, like those trained on the large-scale ImageNet-22k dataset [19]. In practice, this is done by *directly finetuning publicly-available ImageNet pretrained models on the downstream tasks*. This approach has the advantage of being straight-forward, as ImageNet models can be simply downloaded from many open-source model zoos, and has been shown to be effective [69, 70]. However, due to the domain gap between natural images and remote sensing, this approach is not optimal for geospatial data, and still leaves performance gains on the table.

In recent years, a second approach has gained significant traction, where researchers aim to pretrained models specific to the geospatial domain [66, 7, 17, 89]. These methods typically *train a*

*network from scratch on a large corpus of remote sensing imagery* to learn in-domain representations transferable to downstream tasks. Unfortunately, this can require a significant amount of data and training time to achieve good performance, especially when employing large state-of-the-art (SOTA) transformer models. For instance, the current SOTA in geospatial foundation models, SatMAE [17], requires 768 hours on a V100 GPU for training a vision transformer [23]. This has substantial cost associated with producing the model, not just in terms of time and computation but also environmentally, with a total estimated carbon footprint of 109.44 kg $CO_2$ equivalent. Additionally, the final performance of such models are not consistently better across various tasks than simply utilizing publicly-available ImageNet pretrained models (Section 5.4), despite the high resource expense.

In this work, we propose to investigate a different paradigm for producing more effective geospatial foundation models with substantially less resource costs. First, we begin with a discussion on pretraining data selection, and ultimately construct a concise yet diverse collection of data from various sources to promote feature diversity and effective pretraining. Second, rather than following the aforementioned typical approaches, we investigate the potential of ***continual pretraining for the geospatial domain*** from readily-available ImageNet models. Continual pretraining has been practiced in the NLP domain with success in various works [30, 32, 63]. In this paradigm, existing foundation models are further improved for a specific domain or task through a secondary pretraining stage. This new single model can now be fine-tuned on the various downstream tasks in that domain. In principle, we reason that such a paradigm has the potential to boost performance by utilizing large-scale ImageNet representations as a base on which stronger geospatial foundation models can be built. Furthermore, such natural image models are constantly being improved and released by the general computer vision community, providing a consistent source of better baseline models. Therefore, an approach that could enable the geospatial domain to leverage these improvements with minimal resource needs and carbon footprint paves the way for continual,

sustainable benefits for the geospatial community.

However, when we initially experiment with the standard continual pretraining formulation, we find it provides only marginal benefits (Section 5.2). Instead, we discover that utilizing ImageNet representations as an auxiliary distillation objective during pretraining leads to a stronger geospatial foundation model. Building upon this principle, we propose a multi-objective continual pretraining paradigm that significantly enhances performance while requiring minimal resources. Our approach leverages ImageNet's powerful representations to facilitate and expedite learning, while also enabling the acquisition of valuable in-domain features via self-supervised learning on geospatial data. Furthermore, our proposed Geospatial Foundation Model (GFM) exhibits strong performance, surpassing previous state-of-the-art (SOTA) methods across a diverse range of downstream tasks (Section 5.4).

In this dissertation, we therefore investigate a novel paradigm for creating highly effective geospatial models with minimal resource costs. Our methodology begins with data selection and construction of a compact yet diverse dataset from multiple sources to promote feature diversity and enhance pretraining effectiveness, which we term GeoPile. We further explore the potential of continual pretraining from ImageNet models, but find it is not satisfactory in its standard formulation. To achieve better performance with minimal resource needs, we propose a multi-objective continual pretraining paradigm. Our design is surprisingly simple yet effective, constructed as a teacher-student strategy with both a distillation objective and self-supervised masked image modeling. This approach allows GFM to leverage the strong representations of ImageNet to guide and quicken learning, while simultaneously providing the freedom to learn valuable in-domain features. Furthremore, we evaluate our GFM approach, as well as several baseline and SOTA methods, on 7 datasets covering important geospatial applications such as change detection, classification, multi-label classification, semantic segmentation, and super-resolution. Overall, our GFM performs favorably over previous methods (as shown in Figure 5.3).

## 1.3 Overview

By fostering improvements in efficiency within these frameworks, both federated and centralized training models can be significantly refined, offering more sustainable, effective, and adaptable solutions for the application of machine learning across diverse environments. This strategic focus on efficiency not only aligns with technological advancement but also with environmental sustainability and economic viability, marking a critical step forward in the evolution of machine learning applications. In this dissertation, we provide a comprehensive literature review on related work and relevant background. In Chapter 3, we propose a novel method, FedAlign, for improving federated learning performance with memory and computational efficiency. In Chapter 4, we introduce a one-shot federated learning paradigm with diffusion models to achieve strong performance in FL settings with a single communication round, thereby significantly reducing communication overheads. In Chapter 5, we further investigate efficiency for centralized applications, and introduce a novel training paradigm for geospatial foundation models that minimized resource and label needs for effective downstream performance.

# CHAPTER 2: LITERATURE REVIEW

In this chapter, we provide a comprehensive literature review on related work and relevant background to this dissertation. We begin in the decentralized setting, with a formal description of federated learning (FL). We discuss fundamental FL strategies, such as Federated Averaging, and explore enhancements aimed at minimizing client drift through modifications like FedProx and MOON. Furthermore, the review covers one-shot federated learning, proposing solutions to reduce the number of communication rounds required for model convergence. It also discusses the integration of diffusion probabilistic models to address the challenges of model training under privacy constraints. Subsequent sections investigate the centralized setting, particularly for pretraining approaches such as masked image modeling, continual pretraining, and how pretraining is leveraged in the geospatial domain. Each section not only outlines the current methodologies and their advancements but also critically assesses their effectiveness and areas for improvement.

## 2.1 Federated Learning

In general, federated learning algorithms aim to obtain a collective model which minimizes the training loss across all clients. This objective can be expressed as

$$\min_{w} F(w) = \sum_{c=1}^{C} \alpha_c F_c(w), \tag{2.1}$$

where $F_c(w)$ is the local loss of device $c$, and $\alpha_c$ is an arbitrary weight parameter with $\sum_{c=1}^{C} \alpha_c = 1$. One of the earliest algorithms proposed in FL is Federated Averaging, or FedAvg [67]. This approach simply optimizes the local training loss with standard SGD training, and aggregates using a weighted average approach with $a_c = \frac{n_c}{n}$, where $n_c$ is equal to the number of training

samples on client $c$, with a total of $n$ training samples partitioned across all $C$ clients.

Recent works attempt to improve over this baseline with two distinct focuses: improvements to the local training at the client, or improvements to the global aggregation process at the server. In this work, we focus on local training and client drift, and therefore we will first discuss methods of this nature. To mitigate data heterogeneity complications, a common approach is to introduce proximal terms to the local training loss. For instance, FedProx [81] adds the proximal term $\frac{\mu}{2}\left\|w - w^t\right\|^2$, where $\mu$ is a hyperparameter, $w$ is the current local model weights, and $w^t$ is the global model weights from round $t$. The goal of this reparameterization is to minimize client drift by limiting the impact of local updates from becoming extreme. More recently, MOON [58] proposes a similar reparameterization idea inspired by contrastive learning. Specifically, the authors form a local model constrastive loss comparing representations of three models: the global model, the current local model, and a copy of the local model from the previous round. The goals of this term are similar to that of FedProx but in feature representation space; to push the current local representation closer to the global representation. At the same time, the current local model is being pushed away from the representations of the local model copy of the previous round. Other methods [3, 49] follow similar ideas; they aim to limit the impact of the local update or shift the update with a correction term.

*However, these approaches have two main downsides*. First, by restraining the drift, they also inherently limit the local convergence potential. With this, not as much new information is gathered per communication round. Second, many of these methods incur substantial overheads in memory and/or computation. For instance, because of its model constrastive loss, MOON [58] requires the storage of three full-size models in memory simultaneously during training, and forward passing through each of these every iteration. This requires a great deal of additional resources, which are often already scarce in FL client settings.

Other works focus on the server side of the system, aiming to improve the aggregation algorithm. [105] propose a Bayesian nonparametric method for matching neurons across local models at aggregation rather than naively averaging. However, the presented framework is limited in application to fully-connected networks, and therefore [90] extend it to CNNs and LSTMs. FedNova [91] presents a normalized averaging method as an alternative to the simple FedAvg update. As we focus on the local training, these works are orthogonal to our work. A few approaches [102, 71, 84] propose federated schemes inspired by the data augmentation method Mixup, using similar averaging techniques on the local data and sharing the augmented data with the global model or other devices. However, even though the data is augmented in some way prior to distribution, the sharing of private data from the client is less than ideal for privacy preservation. Furthermore, sharing additional data worsens the communication burden on the system, which is a principal concern in FL.

## 2.2 Learning Generality

In traditional centralized training, the practice of regularization of various forms is common practice for improving generality. Data-level regularization, including basic data augmentations and other more advanced techniques [108, 104], are known to be quite effective. Other methods introduce a level of noise to the training process via structural modification; for instance, random or deliberate modifications to the network connectivity [40, 27, 87]. [98] proposes a hybrid approach that introduces self-guided gradient perturbation to the training process through the use of subnetwork representations, knowledge distillation, and input transformations. As part of this work, we employ a variety of regularization methods in many FL settings and analyze their performance in comparison to state-of-the-art FL algorithms.

## 2.3 One-shot Federated Learning

Federated learning (FL) has emerged as a promising paradigm for collaborative machine learning across decentralized devices while preserving data privacy. The seminal work by McMahan et al. [67] introduced the concept of FL, where model updates are computed locally on user devices and aggregated on a central server. However, in the standard FL process, many iterative communication rounds are required for convergence. One-shot FL, therefore, studies how to effectively learn in this distributed setting in a single round, thereby mitigating the need for many communication rounds. Several approaches have been proposed to tackle the unique characteristics of one-shot FL. [29] introduce the one-shot federated learning framework and study several baseline approaches. In [29] and [57], distillation approaches are studied using the ensemble of client models to the global model, and assume a public dataset for this purpose. However, such an assumption is limited, as public data related to the domain of interest is often not available. A data-free method within the distillation methodology was proposed by [109], where a generative adversarial network (GAN) is trained at the server level to generate the data for distillation, and iteratively optimized between distilling to the server model and training the GAN with the ensemble of client models.

Nonetheless, these methods still struggle with heterogeneous environments, as we find in Section 4.1. Generative models on the client are well-suited for better undertaking in such settings, as they can focus on the narrow client distributions and simply generate data at the central location. [37] introduce the use of CVAEs in highly heterogeneous one-shot FL. However, CVAEs exhibit suboptimal sample quality, a limitation that becomes markedly exacerbated with more complex datasets and when subjected to the constraints of DP, which are not explicitly addressed in the study by [37]. In this work, we investigate diffusion models in one-shot FL and leverage their unique characteristics for the task, illustrating their potential in a variety of difficult FL settings and privacy guarantees.

## 2.4 Diffusion Probabilistic Models

Diffusion probabilistic models [38, 21], or simply diffusion models as they are now commonly referenced (DM), have gained traction for application in generative vision tasks. Simply put, DMs aim to learn the backward process that can iteratively denoise an image corrupted with Gaussian noise back to the original. Specifically, as detailed in [38], noise is introduced to a given sample via a Markovian chain forward process

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}), \tag{2.2}$$

where $T$ is the total number of iterations (or timesteps) applied, and $q(x_t|x_{t-1})$ is parameterized by $\mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$. $\beta$ is a value between (0,1), and increases with timestep $t$, essentially making the final $q(x_T|x_0)$ approximately a simple Gaussian $\mathcal{N}(0, I)$. This forward process is fixed, and the goal of the diffusion model is to learn the reverse process. During training, we simply optimize for predicting the noise $\boldsymbol{\rho}$ from an arbitrary step $t$ in the forward process, forming a loss function [38]

$$L = \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\rho}}\left[\|\boldsymbol{\rho} - \boldsymbol{\rho}_\theta\left(x_t, t\right)\|^2\right]. \tag{2.3}$$

The process can also be conditioned on another variable $y$ in $\boldsymbol{\rho}_\theta\left(x_t, y, t\right)$. For example, the diffusion model can be class conditioned [39], with $y$ being a variable representing the class of the sample from a classification dataset. We utilize the class-conditioning approach of [39] in our diffusion models for FL.

## 2.5 Differential Privacy

Differential privacy (DP) [25, 26, 24] is a framework for ensuring that the output of a computation, such as machine learning model training, does not reveal sensitive information about any individual data point in the training dataset. A computation is said to be differentially private if the probability of obtaining a particular output is roughly the same whether a particular individual's data sample is included in the computation or not. Formally [26],

$$Pr[A(D) \in S] \leq e^{\epsilon} \cdot Pr[A(D') \in S] + \delta, \tag{2.4}$$

where $A$ is a randomized algorithm, $D$ and $D'$ are a pair of datasets that differ in at most one record, and $S$ is any subset of the output space of $A$. $(\epsilon, \delta)$ control the level of privacy protection provided by the algorithm, essentially determining the maximum allowable amount of information that can be harnessed from the data. Larger values of $(\epsilon, \delta)$ correspond to weaker privacy guarantees, while smaller values of $(\epsilon, \delta)$ correspond to stronger guarantees.

To train deep learning models with such guarantees, differentially private stochastic gradient descent (DPSGD) is typically employed [1]. In DPSGD, two main mechanisms are used to protect the privacy of individual data points: per-sample gradient clipping and the addition of random noise to the clipped gradients. Per-sample gradient clipping involves setting a maximum threshold on the norm of the gradient computed for each data point, so that if the norm of a gradient exceeds the threshold, it is rescaled. This step is necessary to limit the sensitivity of the loss function, which measures how much the loss function changes when a single data point is removed from the training dataset. After the gradients have been clipped, random noise is added to them before they are used to update the model parameters. The amount of noise added is calibrated based on privacy budget parameters $(\epsilon, \delta)$.

## 2.6 Geospatial Pretraining

Various works have experimented with employing supervised or self-supervised pretraining paradigms in the geospatial domain. The classical work of [69], and more recent paper [89], investigate supervised pretraining on individual datasets of various sizes. Interestingly, these still often found the ImageNet pretrained models to perform very well, particularly with vision transformers [23, 62]. Other works have explored self-supervised learning paradigms for remote sensing, primarily focused on contrastive methods. [66] and [7] employ a MoCo [15] style objective using spatially aligned but temporally different images as the positive pairs. [48] and [42] also utilize a MoCo-inspired objective, but specify a cropping procedure to generate positives and negatives within and across images. [88] employs a colorization objective on Sentinel-2 imagery utilizing the various spectral bands. Most recently, SatMAE [17] explores the use of masked image modeling to train a large ViT model. This work is similar in some respect to ours, as we also train a transformer model with an MIM objective. However, we find that SatMAE often does not perform better than the off-the-shelf ImageNet-22k pretrained ViT (Section 5.4). This indicates both the difficulty of building strong geospatial pretrained models from scratch and highlights the potential usefulness of leveraging continual pretraining instead, as we investigate in this work.

## 2.7 Masked Image Modeling

Masked image modeling (MIM) has been proposed in various forms in recent years, and has recently been found to be particularly effective in the natural image domain, surpassing many contrastive works and being shown to be friendlier to downstream optimization [96, 35, 110, 8, 95] In general, the goal is to learn from data in a self-supervised manner by asking the model to generate pixel values for intentionally-withheld regions in an image. [74] is an early work with an aim of

learning strong visual representations through inpainting masked regions. In [13], Chen et. al train a large transformer to predict pixels autoregressively. After the introduction of vision transformers (ViT) [23], many works continued to improve various MIM variants. [8] and [110] take inspiration from BERT [20] in natural language processing, and tokenize the image patches with either a pretrained model or jointly trained online tokenizer, with the objective being to reconstruct at a token-level rather than raw pixels. Recently, [96] and [35] show that a masked image modeling task of simply regressing directly on the image pixels is sufficient and effective. In this work, we leverage the framework from [96], as it is compatible with hierarchical transformer architectures [62].

In this work, we develop our pretraining objective based on a masked image modeling approach like [96, 35]. Exploration of the masked image modeling framework in geospatial applications is still in its early stages, and could help alleviate some concerns with contrastive approaches in this domain. Particularly, the choice of augmentations with contrastive methods can be quite difficult, as common selections such as greyscale, color jitter and others that heavily affect the intensity of the image can instill undesirable invariances [69]. On the other hand, MIM objectives like [96, 35] rely only on simple spatial augmentations such as flipping and cropping. Furthermore, a common remote sensing application is that of change detection, which requires a model to detect changes in two images from the same location but at different times. In order to still be effective on this task, works that use contrastive approaches on temporal positives introduce various design choices. For instance, SeCo [66] creates multiple feature subspaces during pretraining, each one invariant to a separate form of augmentation. [6] also employs temporal positives, but instead chooses the sampling locations for the pretraining data to ensure that image pairs contain primarily natural illumination and viewing angle variant, without major changes such as new urban developments.

## 2.8 Continual Pretraining

Continual pretraining has been primarily introduced in the natural language domain [30, 32, 63], in order to improve large language models (LLM). [30] illustrates the viability of two additional stages of pretraining, using in-domain data (domain-adaptive), and then even further using task-specific data (task-adaptive). [32] proposes a continual training paradigm for enabling temporal reasoning abilities to pretrained language models. [63] focus on using continual pretraining to enable mixed language neural machine translation. In the vision domain, [47] employs a BYOL [28] style continual pretraining paradigm for 2D medical image segmentation. [77] explores a hierarchical pretraining approach for task adaptation. However, they primarily focus on adapting to a specific downstream task at a time, employing three training stages on top of an existing pretrained model for each task individually. In contrast, we employ one efficient in-domain pretraining setting that can generalize to many downstream tasks, as illustrated in Section 5.4. Furthermore, rather than directly loading the pretrained weights from existing models as initialization, we find instead that leveraging the representations as an auxiliary distillation objective during the pretraining process enables learning stronger representations.

# CHAPTER 3: RESOURCE EFFICIENT FEDERATED LEARNING

The work in this chapter has been published in the following paper:

*Local Learning Matters: Rethinking Data Heterogeneity in Federated Learning. Matias Mendieta, Taojiannan Yang, Pu Wang, Minwoo Lee, Zhengming Ding, Chen Chen. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022. Oral. Best Paper Finalist.*

---

Prior approaches addressing the challenge of data heterogeneity in federated learning have incurred notable memory and computational overhead. In contrast, our objective is to devise a simpler and more resource-efficient solution. To commence our investigation, we undertake an empirical study aimed at gaining deeper insights into the heterogeneity problem.

## 3.1 Empirical Study

**We wish to assess the data heterogeneity challenge of FL from a simple yet unique perspective of local learning generality.** Specifically, we first study the effectiveness of standard regularization techniques as solutions to this FL challenge in comparison to state-of-the-art methods.

### *3.1.1 Preliminaries*

We employ three FL algorithms, namely FedAvg, FedProx, and MOON. These works represent both classic baselines and current state-of-the-art, and are described in Section 2. For comparison, we employ three state-of-the-art regularization methods: Mixup [108], Stochastic Depth [40], and

GradAug [98]. Specifically, these regularization methods are applied to the local optimization within a standard FedAvg setup, and their operations are described as follows.

Mixup is a data-level augmentation technique that performs linear interpolation between two samples. Specifically, given two sample-label pairs $(x_i, y_i)$ and $(x_j, y_j)$, they are combined as $\tilde{x} = \beta x_i + (1-\beta)x_j$ and $\tilde{y} = \beta y_i + (1-\beta)y_j$, where $\beta \sim Beta(\gamma, \gamma)$.

Stochastic depth (StochDepth) is a structural-based method that drops layers during training, thereby creating an implicit network ensemble of different effective lengths. Specifically, the output of layer (or residual block) $\ell$ is given by $\zeta_\ell = \sigma\left(\lambda \mathcal{F}_{\theta_\ell}\left(\zeta_{\ell-1}\right) + \mathcal{I}\left(\zeta_{\ell-1}\right)\right)$, where $\lambda$ is a Bernoulli random variable, $\mathcal{F}_{\theta_\ell}$ is the operation within the network with parameter $\theta$ at layer $\ell$, $\mathcal{I}$ is the identity mapping operation of residual connections, and $\sigma$ is a non-linear activation function. The keep probability is defined as $\rho = P(\lambda = 1)$, where in practice each layer has its own keep probability set with a linear decay rule $\rho_\ell = 1 - \frac{\ell}{L}(1-\rho_L)$, with $L$ denoting the total number of layers (or blocks) in the network.

GradAug is a recent regularization approach that combines data-level and structural techniques in a distillation-based framework. Its training loss is defined as

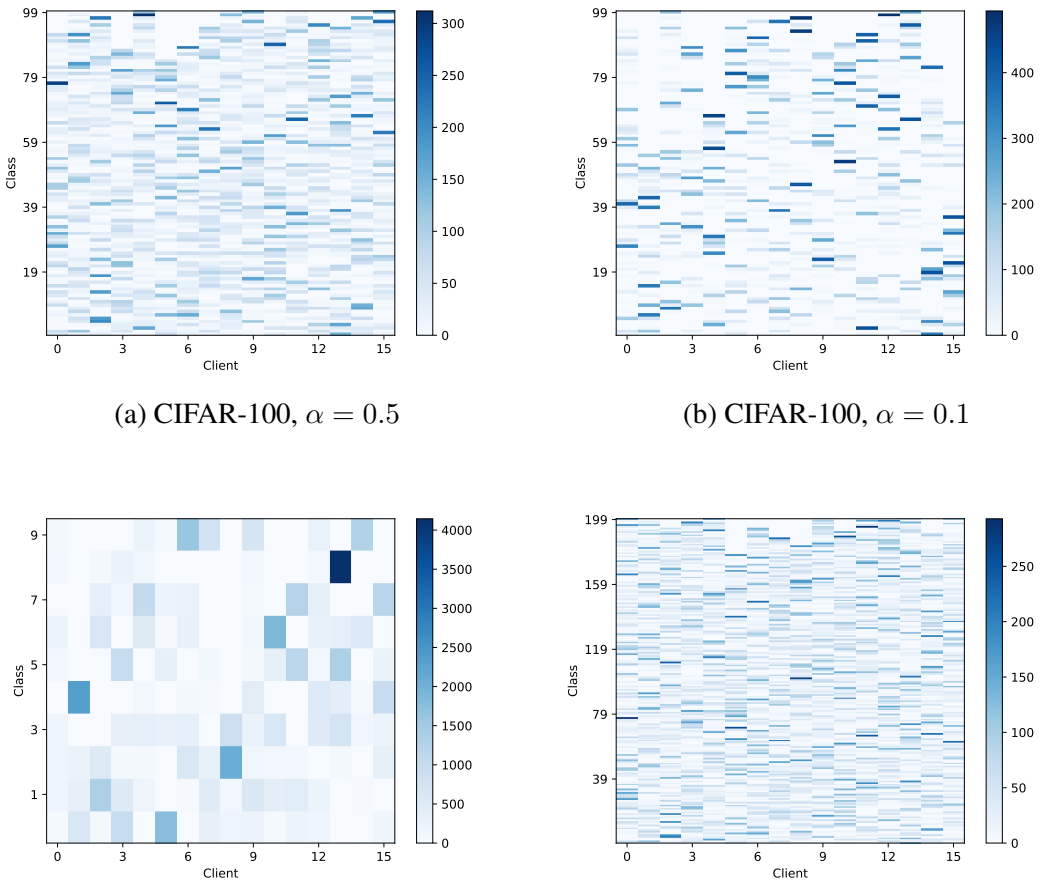
$$L_{GA} = L_{CE}(\mathcal{F}_\theta(x), y) + \mu \sum_{i=1}^{n} L_{KD}\left(\mathcal{F}_{\theta^{\omega_i}}\left(T^i(x)\right), \mathcal{F}_\theta(x)\right), \tag{3.1}$$

where $\mathcal{F}_{\theta^{\omega_i}}$ denotes a slimmed sub-network of fractional width $\omega_i$, $T^i$ is a transformation performed on the input (e.g. resolution scaling), and $\mu$ is a balancing parameter between the cross-entropy loss $L_{CE}$ and the summed Kullback–Leibler divergence ($L_{KD}$) loss on $n$ sub-networks. The $\omega_i$ fractional width for each sub-network is sampled from a uniform distribution between a lower bound $\omega^b$ and 1.0 (full-width).

### *3.1.2 Experimental Setup*

To begin our analysis, we test the accuracy of several state-of-the-art FL algorithms with several regularization methods in a common FL setting. We perform experiments using CIFAR-100 [52], an image recognition dataset with 50,000 training images across 100 categories, and employ ResNet56 [36] (as implemented in FedML [33] with PyTorch [73]) as the model. As common in the literature [58, 3, 33], the dataset is partitioned into $K$ unbalanced subsets using a Dirichlet distribution ($Dir(\alpha)$), with the default being $\alpha = 0.5$. With this data partitioning scheme, it is possible for a client to have no samples for one or multiple classes (see Figure 3.1). Therefore, many clients will only see a portion of the total class instances. This makes the setting more realistic and challenging. For all methods and experiments we use an SGD optimizer with momentum, and a fixed learning rate of 0.01. In our basic setting, training is conducted for 25 rounds, with 16 clients and 20 local epochs per round. Any modifications to this setting in subsequent results will be stated clearly.

We compare the previously described FL algorithms and regularization methods. FedProx, MOON, and GradAug all have a hyperparameter $\mu$ to balance their additional loss terms. We report all results with the optimal $\mu$ for all approaches, being 0.0001, 1.0, and 1.75 for FedProx, MOON, and GradAug respectively. For Mixup and Stochastic Depth, $\gamma$ and $\rho_L$ are set to 0.1 and 0.9 respectively. For GradAug specifically, the number of sub-networks $n = 2$, $\omega^b = 0.8$, and the applied transformation $T$ is random resolution scaling. A two-layer projection layer is added to the model for MOON and the default temperature parameter $\tau = 0.5$ as specified in the original paper. Basic data augmentations (random crop, horizontal flip, and normalization) are kept consistent across all methods.




(a) CIFAR-100, $\alpha = 0.5$

(b) CIFAR-100, $\alpha = 0.1$

(c) CIFAR-10, $\alpha = 0.5$

(d) ImageNet-200, $\alpha = 0.5$

Figure 3.1: Data distribution visualization for $Dir(\alpha)$ and $C = 16$ across multiple datasets. Each column shows the number of samples per class allocated to a client.

### *3.1.3 Results Comparison*

The accuracy results are shown in Table 3.1. Within the current state-of-the-art FL algorithms (upper portion of Table 3.1), MOON achieves the best accuracy. This is expected, as MOON is the most intricate of the FL methods, requiring the usage of three individual models for its contrastive learning technique. However, when we compare with standard regularization techniques (Mixup,

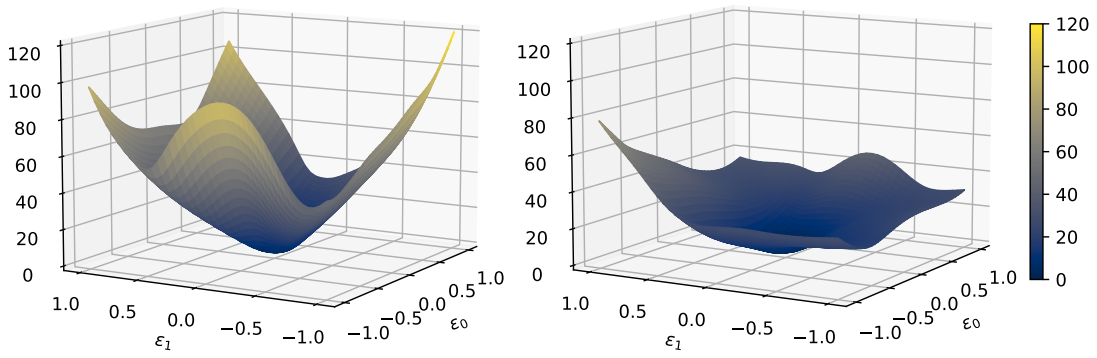
Table 3.1: Results for accuracy (%) on CIFAR-100 and second-order metrics indicating the smoothness of the loss space ($\lambda_{max}$, $H_T$) and cross-client consistency ($H_N$, $H_D$) for each method.

| Method | Acc. ↑ | $\lambda_{max}$↓ | $H_T$↓ | $H_N$↓ | $H_D$↑ |
|---|---|---|---|---|---|
| FedAvg | 52.9 | 297 | 6240 | 11360 | 0.98 |
| FedProx | 53.0 | 270 | 6132 | 6522 | 0.98 |
| MOON | 55.3 | 252 | 5520 | 5712 | 0.97 |
| Mixup | 54.0 | 216 | 5468 | 15434 | 0.99 |
| StochDepth | 55.5 | 215 | 3970 | 8267 | 0.97 |
| GradAug | **57.1** | **167** | **2597** | 2924 | 0.96 |

StochDepth and GradAug in the lower portion of Table 3.1), we see that these perform similarly or substantially better. GradAug particularly stands out, achieving an accuracy $\sim$2% higher than MOON and $\sim$4% higher than FedAvg and FedProx. StochDepth also achieves similar accuracy to MOON. Furthermore, these regularization methods bring the same or better performance than MOON, with less memory and/or compute requirements. *We find that regularization methods appear to have an advantage in this situation; however, we wish to further investigate why this could be the case.* Next, we present our in-depth analysis based on second-order information in Section 3.1.4.

### *3.1.4 Algorithm Analysis based on Second-order Information*

Recent works in the Neural Architecture Search domain [14, 106], as well as in network generalization [50, 101, 44], have noted the importance of the top Hessian eigenvalue ($\lambda_{max}$) and Hessian trace ($H_T$) as a predictor of performance and indicator of network generality. Having a lower $\lambda_{max}$ and $H_T$ typically yields a network that is less sensitive to small perturbations in the networks weights. This has the beneficial effects of smoothing the loss space during training, reaching a flatter minima, and easing convergence. These properties are particularly advantageous in federated learning, where extreme non-IID distributions and limited local data often make convergence




(a) FedAvg

(b) GradAug

Figure 3.2: Visualization of the parametric loss landscape with Hessian eigenvectors $\epsilon_0$ and $\epsilon_1$ for each resulting global model.

difficult.

Motivated by these insights, we analyze the top Hessian eigenvalue and Hessian trace of the global models trained with each FL scheme to provide insight into the effectiveness of each method. As described in [100], the top Hessian eigenvalues can be approximated with the Power Iteration [101] method using a simple inner product and standard backpropagation. Furthermore, [100] also find a similar approximation for the trace utilizing the Hutchinson method [41]. We conduct our analysis with the top Hessian eigenvalues and trace of the final averaged models using these methods.

In Table 3.1, we include the results of the Hessian analysis. First, we find that FedAvg has the highest $\lambda_{max}$ and $H_T$. FedProx and MOON each result in lower values, indicating some degree of improved generalization. However, interestingly, we find that regularization methods are most effective at reducing the $\lambda_{max}$ and $H_T$, with GradAug having by far the lowest in both values. We visualize the effect of this reduction in $\lambda_{max}$ and $H_T$ in Fig. 3.2, where it can be seen that GradAug is able to smooth out the loss landscape considerably in comparison to FedAvg.

In the separate field of out-of-distribution (O.O.D.) generalization for centralized training, second-order information is being found quite useful as a theoretical indicator. Recent works [72, 76] find that forming representations that are "hard to vary" seem to result in better O.O.D. performance. More specifically, they show that the resulting loss landscapes across domains for the learned model should be consistent with each other. In terms of theoretical indicators, this translates to matching domain-level Hessians, as the Hessian provides an approximation of local curvature. Similarly, in federated learning, each client is essentially a separate domain. Therefore, matching Hessians in norm and direction across clients reveals additional detail and reasoning behind the effectiveness of each method. In light of these findings in O.O.D. literature, we analyze the difference in Hessian norm ($H_N$) and the Hessian direction across clients ($H_D$), where

$$H_N^{k,j} = \left(\|\text{Diag}\left(\mathbf{H}_k\right)\|_F - \|\text{Diag}\left(\mathbf{H}_j\right)\|_F\right)^2 \text{ and} \tag{3.2}$$

$$H_D^{k,j} = \frac{\text{Diag}\left(\mathbf{H}_k\right) \odot \text{Diag}\left(\mathbf{H}_j\right)}{\|\text{Diag}\left(\mathbf{H}_k\right)\|_F \cdot \|\text{Diag}\left(\mathbf{H}_j\right)\|_F}. \tag{3.3}$$

Here, $\odot$ is the dot product, $\mathbf{H}_k$ and $\mathbf{H}_j$ are the Hessian matrices of clients $k$ and $j$, and $\|\cdot\|_F$ is the Frobenius norm. $H_N^{k,j}$ and $H_D^{k,j}$ are averaged across all pairs of clients and reported as simply $H_N$ and $H_D$ in Table 3.1. For these Hessian matching criteria, a lower $H_N$ (less difference) and a higher $H_D$ (essentially the cosine similarity) are desired.

As seen on the right side of Table 3.1, $H_D$ is fairly consistent across all methods. In terms of $\lambda_{max}$, $H_T$, and $H_D$, most methods seem to correlate decently well between these values and performance. However, there are a few cases which require more information. First, Mixup has a similar $H_T$ value as MOON, but lower accuracy. $H_N$ provides another detail; the Hessian norms of Mixup are not nearly as similar across clients as those of MOON. Between MOON and StochDepth, we see that MOON has both a higher $\lambda_{max}$ and $H_T$, but StochDepth has a higher $H_N$. In the end, MOON and StochDepth result in similar performance, with perhaps a slight edge towards the

latter.

**Key Insight.** It appears that both the eigenvalue/trace analysis and Hessian matching criteria can serve as a guiding indicator for optimal FL methods. Particularly, they provide insight into the facilitation of convergence and aggregation thorough landscape smoothness and consistency. To understand how these differences will play out empirically, we conduct a variety of ablations in Section 3.1.5.

### *3.1.5 Ablation Study under Various FL Settings*

#### *3.1.5.1 Data Heterogeneity*

Federated systems can be deployed with many different setups and diverse environments. We conduct further analysis across a variety of FL settings to ensure the generality of our findings. First, we examine the effect of varying the degree of heterogeneity in the client data distributions. The results are shown in Table 3.2. We report the mean accuracy $\pm$ the standard deviation across three runs. All other settings are maintained from Section 3.1.2; only the data distribution $Dir(\alpha)$ is varied. *A lower $\alpha$ value indicates a more heterogeneous distribution.*

Table 3.2: Ablation results for varying degrees of data heterogeneity.

| Method | $\alpha = 0.1$ | $\alpha = 0.5$ | $\alpha = 2.5$ | homog |
|---|---|---|---|---|
| FedAvg | 45.0±0.2 | 52.9±0.1 | 54.4±0.2 | 54.9±0.4 |
| FedProx | 45.2±0.3 | 53.1±0.3 | 54.5±0.3 | 54.8±0.5 |
| MOON | 46.5±0.5 | 55.0±0.5 | 56.3±0.6 | 56.3±0.5 |
| Mixup | 44.3±0.1 | 54.0±0.1 | 55.5±0.4 | 56.7±0.4 |
| StochDepth | 48.2±0.3 | 55.5±0.2 | 57.6±0.2 | 58.1±0.6 |
| GradAug | **48.6±0.4** | **57.0±0.1** | **59.6±0.2** | **60.5±0.2** |

As the degree of data heterogeneity decreases, the effect of client drift should become less signifi-

Table 3.3: Ablation results for number of local training epochs.

| Method | $E = 10$ | $E = 20$ | $E = 30$ |
|---|---|---|---|
| FedAvg | 50.6±0.1 | 52.9±0.1 | 53.2±0.3 |
| FedProx | 50.7±0.5 | 53.1±0.3 | 52.8±0.1 |
| MOON | 50.7±0.4 | 55.0±0.5 | 55.2±0.4 |
| Mixup | 50.5±0.4 | 54.0±0.1 | 54.4±0.3 |
| StochDepth | 50.9±0.6 | 55.5±0.2 | 56.4±0.3 |
| GradAug | **53.5±0.3** | **57.0±0.1** | **57.7±0.3** |

cant. Therefore, we expect that the accuracy for each method will increase, with peak performance in the homogeneous setting. All regularization methods, as well as FedAvg, perform as expected, and find consistent improvement across the degrees of data distribution. However, we see that the accuracy improvement of FedProx and MOON slows as the data approaches homogeneity, with accuracy in the purely homogeneous setting ("homog" in Table 3.2) remaining quite low. In their attempt to mitigate client drift and keep local updates close to the global model, it appears that they also hinder their ability to fully learn on minorly heterogeneous or even homogeneous data. This is not ideal for deployable FL systems, as the degree of heterogeneity is not known ahead of time. Moreover, even in the most heterogeneous cases, the structural regularization methods perform better than the standard FL algorithms. For instance, StochDepth achieves a $\sim$1.7% improvement over MOON at $\alpha$ = 0.1, while also having improvement in more homogeneous situations. In all settings, GradAug performs the best.

#### *3.1.5.2 Number of Local Training Epochs*

The main purpose for adequately handling data heterogeneity is to allow for more productive training on the client each round, therefore reducing the time to convergence and required communication cost. Therefore, to examine the training productivity of each method, we examine their

accuracy with various allotted local training epochs per round ($E$) in Table 3.3.

Ideally methods should continue to improve in accuracy with more allotted local training epochs. In Table 3.3, we see that all methods steadily improve from 10 epochs per round to 20. However, from 20 to 30, the trends vary considerably. As a baseline, FedAvg slightly improves by $\sim$0.3%. Surprisingly, FedProx and MOON stay relatively stagnant from 20 to 30 epochs. Meanwhile, the standard (particularly structural) regularization methods continue to increase in accuracy. Therefore, these methods illustrate the ability to maintain productive training, even across a wide range of allotted local epochs.

#### *3.1.5.3 Number of Clients*

In real-world FL settings, the number of participating clients can vary widely. Moreover, only a portion of clients are potentially sampled per round, whether for connectivity reasons or other capacity restrictions of the central system. Therefore, it is crucial that an FL method can converge under such conditions. We study the affect of client number and client sampling in Table 3.4. $C = 64 \times 0.25$ indicates that there are 64 total clients in the system, but only a fraction (0.25) are sampled each round. The rest of the presented results in Table 3.4 sample all $K$ clients each round. $C = 64 \times 0.25$ (100) is run for 100 rounds, and all other settings for the default 25 rounds.

The trends of most methods are similar with increasing clients. However, FedProx struggles to keep up with the FedAvg baseline, especially in the client sampling cases. These scenarios are particularly important; when a small percentage of clients are sampled, only a portion of the dataset is effectively trained on each round. Therefore, learning efficiency becomes paramount for maintaining suitable convergence. The standard regularization methods maintain better accuracy than FedAvg in all settings, often by a significant margin, and even in the client sampling scenario. Overall, GradAug performs the best in all cases. *Therefore, even though these regularization*

Table 3.4: Ablation results for varying number of clients $C$ in synchronous and client sampling cases.

| Method | $C = 16$ | $C = 32$ | $C = 64$ | $C = 64 \times 0.25$ | $C = 64 \times 0.25$ (100) |
|---|---|---|---|---|---|
| FedAvg | 52.9±0.1 | 44.5±0.3 | 34.6±0.2 | 32.7±0.5 | 46.5±0.6 |
| FedProx | 53.1±0.3 | 44.5±0.6 | 34.8±0.2 | 32.5±0.4 | 46.2±0.1 |
| MOON | 55.0±0.5 | 45.8±0.3 | 35.2±0.8 | 34.2±0.2 | 49.5±0.7 |
| Mixup | 54.0±0.1 | 46.0±0.1 | 36.0±0.2 | 33.6±0.6 | 49.1±0.2 |
| StochDepth | 55.5±0.2 | 47.5±0.2 | 35.5±0.6 | 34.6±0.1 | 51.4±0.1 |
| GradAug | **57.0±0.1** | **50.4±0.1** | **40.2±0.1** | **38.1±0.3** | **53.3±0.5** |

*methods were not designed for the FL setting and partial client sampling, they still perform on par with or improve over current state-of-the-art FL algorithms.*

## 3.2 Proposed Method – FedAlign

Overall, we find that GradAug is particularly effective in the FL setting, having the highest accuracy in all tested scenarios along with the lowest $\lambda_{max}$, $H_T$, and $H_N$. However, while this method is quite memory efficient in comparison to many FL methods (only requires a single stored model during training), it does incur a substantial increase in training time and local computation over the FedAvg baseline. This is because GradAug requires multiple forward passes through slimmed sub-networks for the distillation loss. It is possible to reduce the computation burden to some extent by using a smaller number of sub-networks during the knowledge distillation process, as seen in Table 3.5. Here, the $\mu$ in GradAug is adjusted to 2.0, 1.5, and 1.25 for $n$ = 1, 3, and 4, respectively. Nonetheless, a considerable gap still remains between GradAug and vanilla FedAvg in local compute requirements and subsequent wall-clock time. **Therefore, the question is, can we devise a method which provides similar effect and performance as GradAug in FL, but with substantially less computational overhead?** This is particularly important in the FL setting, where clients are typically deployed devices with minimal memory and computational resources.

Table 3.5: Analysis of local compute, stored parameters, and wall-clock time. FLOPs are calculated for the compute needs for the forward pass of the training process. Parameters include the total number of stored parameters needed for each method during training. Wall-clock time is measured as a per-round average on CIFAR-100 with $C$=16 and $E$=20 across 4 RTX-2080Ti GPUs.

| Method | Acc (%) ↑ | MFLOPs ↓ | Param (M) ↓ | Time (s) |
|---|---|---|---|---|
| FedAvg | 52.9±0.1 | 87.3 | **0.61** | 137.2 |
| FedProx | 53.1±0.3 | 87.3 | 1.21 | 161.9 |
| MOON | 55.0±0.5 | 262.2 | 2.21 | 414.2 |
| Mixup | 54.0±0.1 | 87.3 | **0.61** | 137.8 |
| StochDepth | 55.5±0.2 | **82.4** | **0.61** | **136.7** |
| GradAug ($n = 1$) | 56.7±0.3 | 133.9 | **0.61** | 229.2 |
| GradAug ($n = 2$) | **57.0±0.1** | 170.7 | **0.61** | 323.9 |
| GradAug ($n = 3$) | 56.8±0.3 | 217.4 | **0.61** | 417.7 |
| GradAug ($n = 4$) | 56.9±0.3 | 264.1 | **0.61** | 514.4 |
| **FedAlign** | 56.8±0.3 | 89.1 | **0.61** | 166.2 |

To do so, we first take note of the following insights gathered during our analysis: 1) Second-order information is insightful for understanding the learning generality of neural networks. Particularly, we find that flatness and consistency in this realm are desirable traits. 2) In practice, we find that structural regularization, and especially distillation-based like GradAug, is quite effective. Furthermore, the weight sharing mechanisms of such approaches are memory efficient compared to other methods that rely on global model or previous model storage. Therefore, we combine these insights into a novel algorithm to optimize for performance and resource needs in FL.

We propose **FedAlign**, a distillation-based regularization method that aligns the Lipschitz constants (i.e. top Hessian eigenvalues) of the most critical network components through the use of slimmed sub-blocks. Fig. 3.3 shows an overview of FedAlign, whose design is based on two key principles. First, motivated by the insights of Section 3.1.4, we internally regularize the Liptschitz constants of network blocks to *promote smooth optimization and consistency within the model.* Recent work [83] presents a quick approximation of the Lipschitz constants for neural network layers

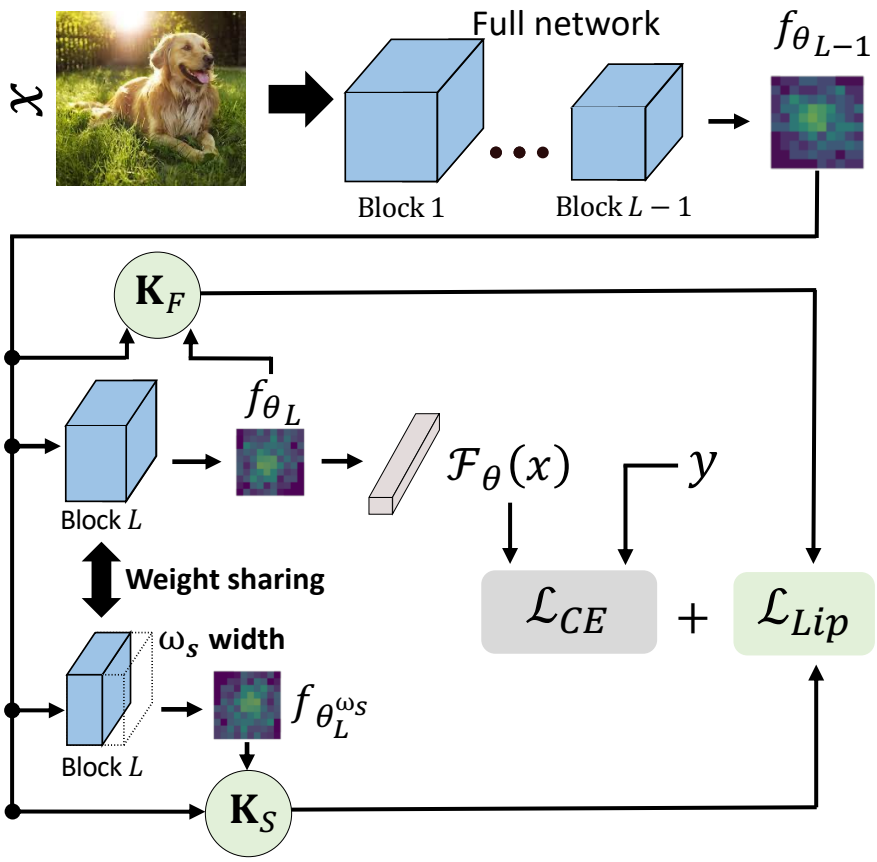



Figure 3.3: The proposed FedAlign for local client training in FL. Features $f_{\theta_{L-1}}$ are run through Block $L$ as normal. The only additional inference in FedAlign is through Block $L$ at a reduced width (i.e. sub-block), reusing features $f_{\theta_{L-1}}$ as input. The channels throughout the layers in the sub-block are a $\omega_S$ fraction of the original number. This is accomplished via temporary uniform pruning of Block $L$.

in a differentiable manner. This enables the use of second-order information in the distillation process, traditionally between a fully trained teacher and a learning student. We adapt this technique for distillation-based regularization with a *single untrained network* in place of the traditional logit-based loss. Second, in order to reduce computation in a purposeful manner, we take note of certain network properties. Particularly, it has been shown that the final layers of a neural network are most prone to overfit to the client distribution [65]. Therefore, we design FedAlign with a focus on these critical points in the network. *The question we raise is, when aiming to concentrate our*

*regularization efforts on the final layers, why should we run all sub-networks for distillation from start to finish?* Instead, we propose to reuse the intermediate features of the full network as input to just the final block at a reduced width, and therefore significantly reduce computation. In this way, we harness the benefits of distillation-based regularization in performance and memory footprint, while effectively mitigating computational overhead.

Combining these two key principles, we form the FedAlign local objective as

$$\mathcal{L}_{FA} = \mathcal{L}_{CE}(\mathcal{F}_\theta(x), y) + \mu \mathcal{L}_{Lip}\left(\mathbf{K}_S, \mathbf{K}_F\right), \tag{3.4}$$

where $\mu$ is a balancing constant, $\mathcal{L}_{CE}$ is the cross-entropy loss, and $\mathcal{L}_{Lip}$ is the mean squared error between the approximated Liptschitz constant vectors $\mathbf{K}_S$ and $\mathbf{K}_F$ for the reduced width (i.e. sub-block) and full width block $L$, respectively. Specifically, the Lipschitz approximations are calculated via the spectral norm of a transmitting matrix using feature maps as in [83], which bypasses the need for singular value decomposition. Therefore, we use the intermediate features for these transmitting matrices $\mathbf{X}_F$ and $\mathbf{X}_S$, where

$$\mathbf{X}_F = \left(f_{\theta_{L-1}}\right)^\top f_{\theta_L}, \text{and} \tag{3.5}$$

$$\mathbf{X}_S = \left(f_{\theta_{L-1}}\right)^\top f_{\theta_L^{\omega_S}}. \tag{3.6}$$

$f_{\theta_L}$ and $f_{\theta_{L-1}}$ are the feature maps outputted by the last and prior-to-last blocks of the full network $\mathcal{F}_\theta(x)$; $f_{\theta_L^{\omega_S}}$ is the output feature map of the final block $L$ at reduced width $\omega_S$ (see Fig. 3.3). Finally, the spectral norm ($SN$) of $\mathbf{X}_F$ and $\mathbf{X}_S$ are approximated using the Power Iteration method [101], and therefore $\mathbf{K}_F = \|\mathbf{X}_F\|_{SN}$ and $\mathbf{K}_S = \|\mathbf{X}_S\|_{SN}$. A pseudocode implementation of FedAlign in presented in Alg. 1. Looking back to Eq. 3.4, one could view $\mathcal{L}_{Lip}$ as a correction term; however, there is a key distinction between this form of regularization and that of traditional

**Algorithm 1** FedAlign

**SERVER OPERATIONS**
**Inputs:** Round number $R$, Set of clients $S$
**Output:** Final global model weights $\theta^R_{global}$
Initialize model weights $\theta^0_{global}$
**for** $r = 0, 1, \ldots, R-1$ **do**
  Sample available clients $C$ from $S$
  **for** client $c \in C$ **in parallel do**
    $\theta^r_c \leftarrow$ CLIENTOPERATIONS$(\theta^r_{global})$
  **end for**
  $\theta^{r+1}_{global} \leftarrow \sum_{c=1}^{C} \frac{n_c}{n} \theta^r_c$
**end for**
**CLIENT OPERATIONS**
**Input:** Model weights $\theta_{global}$
**Output:** Updated local model weights $\theta$
Load received weights $\theta_{global}$ to local model $\mathcal{F}_\theta$
**for** *epoch* $e = 0, 1 \ldots, E-1$ **do**
  **for** batch $\{x, y\} \in D$ **do** ▷ Local dataset $D$
    $f_{\theta_{L-1}}, f_{\theta_L}, pred = \mathcal{F}_\theta(x)$
    $f_{\theta^{\omega_S}_L} = \mathcal{F}_{\theta^{\omega_S}_L}(f_{\theta_{L-1}})$
    $\mathbf{X}_S, \mathbf{X}_F = TM(f_{\theta^{\omega_S}_L}, f_{\theta_{L-1}}, f_{\theta_L})$ ▷ Eqs. 3.5, 3.6
    $\mathbf{K}_S, \mathbf{K}_F = \|\mathbf{X}_S\|_{SN}, \|\mathbf{X}_F\|_{SN}$
    $\mathcal{L}_{FA} = \mathcal{L}_{CE}(pred, y) + \mu\mathcal{L}_{Lip}(\mathbf{K}_S, \mathbf{K}_F)$
    $\theta \leftarrow update(\theta, \mathcal{L}_{FA})$ ▷ Gradient descent
  **end for**
**end for**
Send updated local model weights $\theta$ to server

FL algorithms. *Our correction term promotes the local client models to learn well-generalized representations based on their own data, instead of forcing the local models to be close to the global model.*

Table 3.6: FedAlign ablation results on CIFAR-100.

| Method | $\alpha = 0.1$ | $\alpha = 2.5$ | homog | $E = 10$ | $E = 30$ | $C = 32$ | $C = 64$ | $C = 64 \times 0.25$ | $C = 64 \times 0.25$ (100) |
|---|---|---|---|---|---|---|---|---|---|
| FedAlign | 48.7±0.2 | 57.6±0.6 | 58.2±0.1 | 51.2±0.3 | 57.9±0.6 | 47.8±0.3 | 36.5±0.1 | 34.9±0.6 | 50.9±0.5 |

Table 3.7: CIFAR-10 and ImageNet-200 results for all methods.

| | CIFAR-10 | | | | ImageNet-200 | | | |
|---|---|---|---|---|---|---|---|---|
| Method | $C = 16$ | $C = 64 \times 0.25$ (100) | MFLOPs ↓ | Param (M) ↓ | $C = 16$ | $C = 32 \times 0.125$ (50) | GFLOPs ↓ | Param (M) ↓ |
| FedAvg | 81.9±0.6 | 78.9±0.3 | 87.3 | 0.61 | 60.7±0.4 | 52.7±0.2 | 18.1 | 11.22 |
| FedProx | 81.9±0.2 | 78.9±0.7 | 87.3 | 1.21 | 61.0±0.4 | 52.5±0.3 | 18.1 | 22.42 |
| MOON | 82.9±0.4 | 79.4±0.5 | 262.2 | 2.21 | 61.1±0.2 | 54.3±0.2 | 54.4 | 19.96 |
| Mixup | 80.3±0.4 | 80.5±0.5 | 87.3 | 0.61 | 61.0±0.3 | 52.3±0.3 | 18.1 | 11.22 |
| StochDepth | 82.2±0.2 | 80.8±0.7 | 82.4 | 0.61 | 60.5±0.2 | 52.9±0.2 | 17.3 | 11.22 |
| GradAug ($n = 2$) | 84.6±0.6 | 83.8±0.3 | 170.7 | 0.61 | 63.5±0.4 | 55.6±0.1 | 34.4 | 11.22 |
| GradAug ($n = 1$) | 84.0±0.2 | 82.3±0.5 | 133.9 | 0.61 | 62.8±0.3 | 54.4±0.4 | 25.3 | 11.22 |
| **FedAlign** | 82.3±0.3 | 82.3±0.3 | 89.1 | 0.61 | 62.0±0.1 | 55.1±0.5 | 19.3 | 11.22 |

As seen in Table 3.5, FedAlign achieves state-of-the-art accuracy in a resource-efficient manner. With just a 1.02x difference in FLOPs, FedAlign realizes a significant $\sim$3.9% accuracy improvement over the FedAvg baseline. For the FL algorithms FedProx and MOON, they not only have much lower accuracy than FedAlign, but also require substantially more compute and/or memory. Particularly, FedAlign achieves a $\sim$1.8% accuracy improvement over MOON, while reducing the local compute overhead by over 65% and the memory requirements by over 70%. Furthermore, FedAlign realizes a critical $\sim$47% and $\sim$33% reduction in compute needs compared to GradAug with ($n = 2$) and ($n = 1$), without sacrificing accuracy.

### *3.2.1 FedAlign Experiments*

We further verify the effectiveness of our method across various settings and datasets. In Table 3.6, we examine the performance of FedAlign with the same ablations as in Section 3.1.5, where FedAlign exhibits strong performance in many settings. We also investigate FedAlign and all other methods across two additional datasets: CIFAR-10 and ImageNet-200. For ImageNet-200, we randomly sample 200 classes from the classic ImageNet-1k [80] dataset. We employ ResNet56 and ResNet18 [36] as our models on CIFAR-10 and ImageNet-200, respectively. For FedAlign, $\omega_S = 0.25$ and $\mu = 0.45$ in all results. Hyperparameters for all other methods are those described in Section 3.1.2 (with $\mu = 2.0$ for GradAug ($n = 1$) as in Table 3.5).

For CIFAR-10, we ran a 16 client synchronous and 64 client case with sampling in Table 3.7. We note similar trends to CIFAR-100; regularization methods perform well, particularly in the more realistic client sampling case. On ImageNet-200, we also ran synchronous and sampling settings. Here, both GradAug and FedAlign maintain higher performance than other methods. FedAlign provides competitive accuracy with GradAug ($n = 1$) and even ($n = 2$) in the sampling case, while reducing computational needs by a significant margin. Interestingly, StochDepth does

not perform as well in the ImageNet-200 cases. As mentioned in the original paper [40], Stochastic Depth performs better with deeper networks. However, with ResNet18, the overall depth of the network is reduced compared to that in the CIFAR cases. Therefore, as most deployable networks favor width over depth, regularizing with respect to the width of a network is more applicable to the FL setting. *This highlights an additional benefit of FedAlign, which operates using width reduction in the final block and maintains relatively high accuracy despite low resource needs.*

## 3.3 Summary and Discussion

In this chapter, we study the data heterogeneity challenge of FL from a simple yet unique perspective of local learning generality. To this end, we present a thorough study of various methods in FL settings, and further propose FedAlign, which achieves competitive SOTA accuracy with excellent resource efficiency. One limitation of our study is that we only focused on image tasks and models for the experiments. Natural language processing applications of FL are also a common setting, and therefore could be explored in future work. Nonetheless, we note that FedAlign can easily be applied to language applications, as it operates in the feature space and does not have a fundamental reliance on the input type. On the other hand, GradAug is primarily designed for vision data, employing a random transformation and applying it to the input of sub-networks.

While no one presented regularization method is perfect in all respects, we emphasis that local learning is extremely important in federated settings. Furthermore, methods that particularly focus on promoting learning generality inherently improve global FL aggregation and optimization to a surprising degree. By introducing methods like GradAug in FL, we propose a rethinking of federated optimization and how to tackle its challenges. As a step further in this direction, FedAlign provides strong improvement over classic baselines and state-of-the-art FL methods while addressing the local computational restraints of an FL system.

# CHAPTER 4: COMMUNICATION EFFICIENT FEDERATED LEARNING

The work in this chapter has been submitted as a conference paper:

*Navigating Heterogeneity and Privacy in One-Shot Federated Learning with Diffusion Models, Matías Mendieta, Guangyu Sun, Chen Chen.*

---

A significant bottleneck in federated learning pertains to communication. Typically, communication with the server is necessitated in each round and across multiple rounds. In order to alleviate this bottleneck, we explore the one-shot federated learning paradigm, developing a framework with diffusion models to concurrently address the challenge of data heterogeneity while mitigating communication overheads. Within this investigation, we formulate two primary research questions:

***RQ1***. Initially, we inquire into the efficacy of diffusion models within the framework of federated learning and their potential for enhancing the performance of the one-shot federated learning process. The emergence of diffusion models [38] as prominent techniques for image generation inspires our inquiry. We posit that distinctive attributes of diffusion models may confer advantages for one-shot federated learning, as elucidated in Section 4.1. Subsequently, we substantiate this hypothesis through comprehensive experimentation with our proposed approach, denoted as FedDiff, across diverse experimental settings.

***RQ2***. Secondly, we dive into the realm of one-shot federated learning methodologies under provable privacy constraints, leveraging differential privacy (DP). This aspect remains largely unaddressed in existing state-of-the-art one-shot federated learning literature. Given the criticality of

safeguarding model privacy, particularly in scenarios where client models obtained through one-shot federated learning may be subject to multiple reuses or potential trading in a model marketplace, we examine the efficacy of DP as a privacy-preserving mechanism. Moreover, drawing upon recent advancements [9], we investigate potential memorization issues within the context of diffusion models utilized in our FedDiffapproach, and assess the effectiveness of DP as a mitigation strategy.

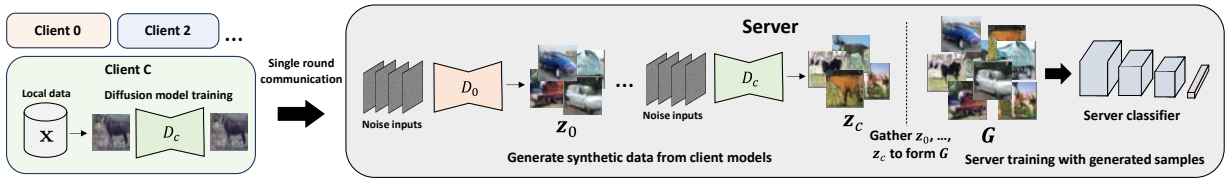



Figure 4.1: Our one-shot FL approach, FedDiff. We first train a class-conditioned diffusion model on local data $\mathbf{x}$ at the clients. After completing training, the local diffusion models $D_0$, $D_1$, ..., $D_c$ are gathered by the server, where they are used to generate data $\mathbf{z}_0$, $\mathbf{z}_0$, ..., $\mathbf{z}_c$, which are combined to form the global training data $\mathbf{G}$. The global model is then trained on this synthetic dataset $\mathbf{G}$.

## 4.1 Diffusion Models for Federated Learning

Before delving into the underlying motivation for our research questions ***RQ1*** and ***RQ2***, it is essential to provide a brief exposition of the one-shot FL process when integrating generative models. The core premise of this approach departs from the traditional method of client-side discriminative model training. Instead, it advocates for the training of generative models on the client devices. These client-side generative models are aggregated and used offline on the server side to synthesize data, which, in turn, facilitates the training of a global discriminative model. Within the scope of our study, we undertake an investigation into the viability of leveraging diffusion models in this paradigm.

**Why diffusion models?** In [94], a generative learning trilemma is shown with model types, trading off sample quality, diversity, and fast sampling. CVAEs (as employed in [37]) are typically identified to excel in diversity and fast sampling, but lacking in sample quality. However, for one-shot federated learning, fast sampling is not a concern, as the sampling can be done offline at the server (Figure 4.1). Therefore, ***high sample quality and diversity are more valuable properties in one-shot FL***, as these will positively impact the performance of the trained global model with the synthetic data. In this trilemma, ***diffusion models excel in sample quality and diversity*** [94], but are not as quick to sample. This motivated us to investigate the potential of DMs in this setting, as the inherit strengths of DMs align with the needs of one-shot FL.

Furthermore, while CVAEs and diffusion models share a common origin in terms of their objective, they differ in their approach to achieving this objective. The optimization task of the diffusion model is simplified to learning a Markov process to reverse a fixed forward process. The training is structured such that the model only needs to learn how to denoise a small step in the generation process, breaking down the problem. In contrast, CVAEs must simultaneously learn both the forward process to encode the image to a latent space, and the decoding process from that latent vector. We reason that the simplified objective of DMs helps achieve superior performance when dealing with complex data within the challenging FL environment (data heterogeneity, class imbalance, and limited sample sizes). Moreover, in the FL setting, privacy is of critical importance. To ensure privacy, training is done with DP, which introduces noise to the training process and increases the difficulty of optimization. In these settings, the simpler training paradigm of diffusion models is potentially advantageous.

**Overview.** To provide a contextual foundation for our research inquiries, we start by laying out the settings of our study and approach in Section 4.1.1. With this groundwork, we investigate ***RQ1*** in Section 4.2, where we dive into the effectiveness of diffusion models in one-shot FL with our FedDiff approach. In Section 4.3, we address ***RQ2*** through a systematic exploration of one-shot

FL methods within provable privacy budgets. Specifically, we evaluate FedDiff and other SOTA approaches under DP constraints, as well as investigate the viability of DP in mitigating memorization. In Section 4.3.3, we also introduce our Fourier Magnitude Filtering approach, aimed at enhancing the efficacy of generated data for global model training by selectively eliminating low-quality samples.

### *4.1.1 FedDiff and Experimental Setup*

The basis of our approach, FedDiff, is illustrated in Figure 4.1. We begin by training class-conditioned diffusion models using the local data $\mathbf{x}$ on the clients. After training, the server collects these local models, denoted as $D_0$, $D_1$, ..., $D_c$, which are then used to generate data $\mathbf{z}_0$, $\mathbf{z}_1$, ..., $\mathbf{z}_c$. The label distributions from the clients are used to condition the generative models during generation, as in [37]. The combination of these synthesized samples forms our global training dataset, $\mathbf{G}$. Subsequently, the global model is trained on the synthetic dataset $\mathbf{G}$ and evaluated in our experiments.

#### *4.1.1.1 Comparison Methods.*

We compare with key baselines and the most recent state-of-the-art one-shot FL methods throughout our investigation.

*FedAvg* [67] is a standard baseline, which simply trains discriminative classifiers at the clients and averages their parameters, typically weighted by the number of samples at each client, to form a single server model.

*DENSE* [109] is a one-shot FL approach that first trains the discriminative classifiers on the clients to convergence. Once the client models are collected, it performs two stages of training in an

interactive manner, switching between training a GAN-based network for generating synthetic data and using the synthetic data to distill the ensemble of client models to a single server model.

*OneShot-Ens*. We also include an idealized variant of DENSE, where rather than attempt to distill the ensemble of client models to a single server model, we simply employ the ensemble as the final model, as shown in [29] and similarly compared to in [37]. We term this approach OneShot-Ens throughout the paper.

*FedCVAE* [37]. This recently proposed method employs conditional variational autoencoders (CVAEs) for one-shot federated learning. Their approach has two variants, FedCVAE-KD and FedCVAE-Ens, which differ in how they operate at the server level. FedCVAE-KD distills all generative models from the clients to a single CVAE, and then generates data for training the global model. On the other hand, FedCVAE-Ens employs each client model to generate data, contributing to the final dataset for training the server model. The latter variant always shows significantly better performance than the other in their paper; therefore, we compare with this FedCVAE-Ens variant and refer to it as FedCVAE in the rest of the paper.

#### *4.1.1.2 Datasets.*

We employ three datasets, FashionMNIST [92], PathMNIST [97], and CIFAR-10 [51], which provide a range of domains and complexities. For our experiments, we divide the training set among $C$ clients with a Dirichlet distribution $Dir(\alpha)$, as commonly done in FL literature [68, 4, 34, 37]. This partitioning approach creates imbalanced subsets, where some clients may not have any samples for certain classes. As a result, a significant number of clients will only encounter a small subset (or potentially only one) of the available class instances. We visualize data distributions with $Dir(0.1)$ and $Dir(0.001)$ across 10 clients in Figure 4.2.

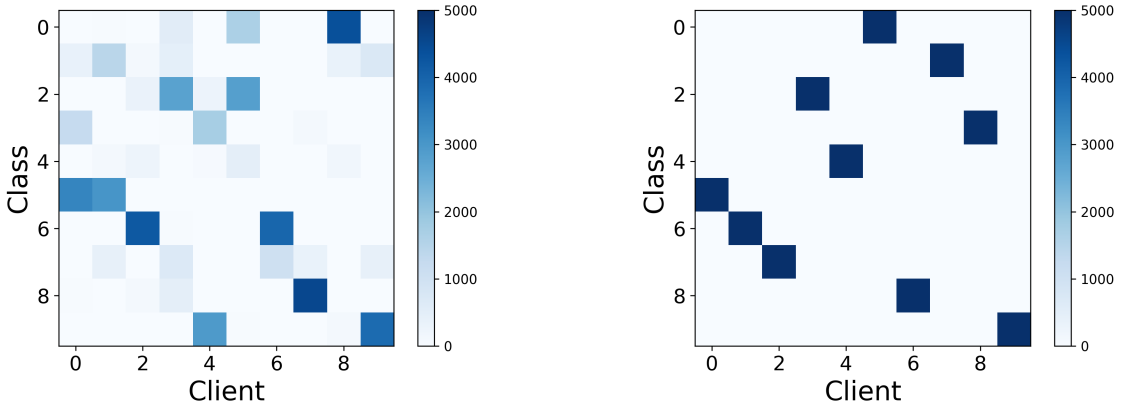
#### *4.1.1.3 Federated Learning Settings.*

We reproduce DENSE and FedCVAE for our settings with their respective official code repositories. For all experiments, we perform 3 independent runs with different seeds and report the mean and standard deviation. For all approaches, we train client models for 200 local epochs, as in [109]. For DENSE, FedCVAE, and FedDiff, we train the final global model for 50 epochs. For the generator of FedCVAE, we employ their CVAE variant with residual blocks, which has approximately 5.9M parameters. For our diffusion model, we employ a basic U-Net structure with residual blocks [38, 78] and class-conditioning, with similar parameters to FedCVAE ($\sim$5.8M). We employ a ResNet16 architecture for the discriminative models with approximately 6.4M parameters. For experiments with differential privacy, we employ the Opacus [103] library in PyTorch [73] to track privacy budgets.

#### *4.1.1.4 Additional Training Details*

The FashionMNIST dataset is an alternative to the original MNIST dataset, providing a more challenging task by replacing the handwritten digits with grayscale images of various fashion items. The dataset consists of 60,000 training images and 10,000 test images. The PathMNIST dataset is a medical dataset of colon pathology images in RGB, with a training set of 89,996 images and a test set containing 7,180 images with 9 classes. The CIFAR-10 dataset consists of 60,000 color images equally distributed into ten different classes. The dataset is composed of a training set containing 50,000 images and a test set comprising of 10,000 images. CIFAR-10 is natively sized at 32$\times$32 pixels. We upsample FashionMNIST and PathMNIST from 28$\times$28 to 32$\times$32.

We train with a batch size of 128 for all methods and use the AdamW optimizer. For local (and global training were applicable), we searched learning rates from [$3e^{-3}$, $1e^{-3}$, $3e^{-4}$, $1e^{-4}$] for




(a) $\alpha = 0.1$ (b) $\alpha = 0.001$

Figure 4.2: $Dir(\alpha)$ data partitioning for 10 clients on CIFAR-10. We show moderate ($\alpha = 0.1$) to severe ($\alpha = 0.01$) data heterogeneity levels. Data heterogeneity poses a significant challenge for many one-shot FL methods, as reconciling various models trained on widely different distributions is non-trivial. Our FedDiff approach rather trains diffusion models on the simple client distributions, which can then generate useful synthetic data for training global models.

each method using the CIFAR-10 dataset to find the optimal settings. For DP experiments, we set the max gradient norm clipping threshold to 1.0 for all experiments and methods. In accordance with the recommendations of the Opacus [103] library, we employ their Poisson batch sampling to ensure privacy guarantees.

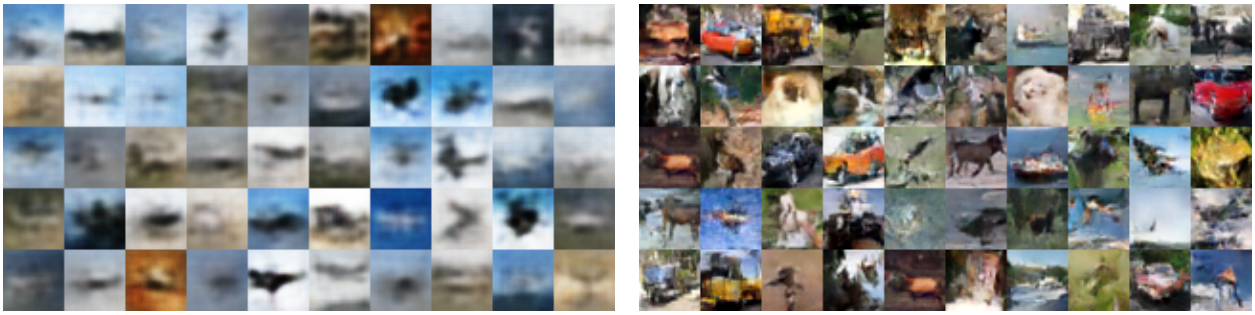
As mentioned in Section 3.1 of the main paper, our diffusion model is a basic U-Net structure with residual blocks [38, 78] and class-conditioning. For sampling at the server, we perform 1000 iterations as in [38] to generate each batch. The total number of generated samples is set equal to the size of the original dataset.

Table 4.1: Data heterogeneity results with various $Dir(\alpha)$ partitions. Smaller alpha values indicate higher levels of heterogeneity. Typical approaches leveraging discriminative models rapidly degrade in performance as heterogeneity increases. However, generative approaches are more robust to such conditions. **Our FedDiff shows superior performance to all, particularly in the most challenging scenarios (CIFAR-10, high heterogeneity)**.

| | Method | $\alpha = 0.1$ | $\alpha = 0.01$ | $\alpha = 0.001$ |
|---|---|---|---|---|
| FashionMNIST | FedAvg | 57.11±3.64 | 29.50±10.6 | 25.89±4.78 |
| | DENSE | 65.20±3.55 | 28.92±17.3 | 27.68±4.08 |
| | OneShot-Ens | 67.35±1.19 | 33.79±17.9 | 32.01±3.35 |
| | FedCVAE | 78.08±2.69 | 78.81±3.25 | 81.53±0.23 |
| | **FedDiff** | **87.21±0.74** | **86.81±0.54** | **86.59±0.69** |
| PathMNIST | FedAvg | 28.10±4.60 | 22.05±8.20 | 21.92±4.95 |
| | DENSE | 50.97±3.19 | 29.26±10.7 | 27.69±4.52 |
| | OneShot-Ens | 34.62±3.61 | 34.94±9.32 | 34.49±5.30 |
| | FedCVAE | 41.60±0.82 | 44.81±1.41 | 47.35±3.21 |
| | **FedDiff** | **74.58±1.02** | **70.61±1.37** | **69.43±1.30** |
| CIFAR-10 | FedAvg | 19.64±2.39 | 19.01±3.76 | 18.16±5.49 |
| | DENSE | 36.04±7.75 | 21.40±2.73 | 17.91±3.18 |
| | OneShot-Ens | 39.38±7.53 | 23.38±3.62 | 20.15±9.11 |
| | FedCVAE | 34.40±1.04 | 36.06±3.27 | 36.92±1.38 |
| | **FedDiff** | **57.69±2.07** | **56.57±2.42** | **55.75±1.55** |

## 4.2 ***RQ1***: FedDiff for One-Shot FL

We investigate ***RQ1*** by exploring the efficacy of our FedDiff approach and other SOTA one-shot FL methods across important FL scenarios, including different data heterogeneity levels, number of clients, and resource requirements.



(a) FedCVAE (b) FedDiff (ours)

Figure 4.3: Random sets of generated samples from FedCVAE and our FedDiff approach. By leveraging the intrinsic properties of diffusion models (DMs), which are well-aligned with the requirements of one-shot FL, we achieve substantial benefits in sample quality and subsequent global model performance.

### *4.2.1 Data Heterogeneity*

Data heterogeneity is a critical challenge in FL, particularly with one-shot settings. Even in the standard FL scenario of multiple communication rounds, client models often fit to very different distributions, and effectively reconciling their learnings is daunting. This is exacerbated in the one-shot setting, as we no longer have the luxury of getting many iterations to progressively steer the learning process towards an ideal encompassing representation.

In Table 4.1, we analyze the performance of all methods under moderate ($Dir(0.1)$) to extreme ($Dir(0.001)$ heterogeneity. Interestingly, FedDiff outperforms all other methods by a significant margin, from $\sim$5% to up to $\sim$20% in different scenarios. In the case of CIFAR-10, which is the most complex of the datasets, we find that FedDiff provides the most improvement. As discussed in our initial motivations (Section 4.1), we reason that the focus on sample quality and diversity that is provided by the DM objective enables much improved performance. Intuitively, this becomes increasingly evident in more complex settings. To verify this observation, we conduct a

Table 4.2: Results with varying number of clients $C$ with $Dir(0.01)$. As a fixed-size dataset is used in all experiments, increasing the number of clients also decreases the number of samples per client. We find that the SOTA discriminative approaches quickly degrade as the data is distributed across more clients. On the contrary, **our FedDiff maintains strong performance in all settings**.

| | Method | $C = 5$ | $C = 10$ | $C = 20$ |
|---|---|---|---|---|
| FashionMNIST | FedAvg | 43.73±2.37 | 29.50±10.6 | 28.38±3.17 |
| | DENSE | 48.24±6.25 | 28.92±17.3 | 20.72±9.82 |
| | OneShot-Ens | 49.53±6.08 | 33.79±17.9 | 31.36±8.01 |
| | FedCVAE | 78.45±2.44 | 78.81±3.25 | 78.33±2.45 |
| | **FedDiff** | **86.89±0.34** | **86.81±0.54** | **87.24±0.57** |
| PathAMNIST | FedAvg | 29.18±3.54 | 22.05±8.20 | 19.83±3.16 |
| | DENSE | 33.39±6.14 | 29.26±10.7 | 20.79±5.77 |
| | OneShot-Ens | 36.95±5.30 | 34.94±9.32 | 24.59±6.19 |
| | FedCVAE | 46.16±1.17 | 44.81±1.41 | 41.25±1.68 |
| | **FedDiff** | **72.74±0.63** | **70.61±1.37** | **69.11±0.99** |
| CIFAR-10 | FedAvg | 29.14±5.15 | 19.24±3.77 | 15.79±2.64 |
| | DENSE | 30.48±2.30 | 21.40±2.73 | 12.60±2.33 |
| | OneShot-Ens | 36.17±3.21 | 23.38±3.62 | 13.23±2.96 |
| | FedCVAE | 32.34±2.59 | 36.06±3.27 | 37.63±1.87 |
| | **FedDiff** | **57.68±1.86** | **56.57±2.42** | **58.45±0.73** |

comparative analysis of generated samples produced by FedCVAE and our FedDiff approach, illustrated in Figure 4.3. The discernible disparity is evident, with the samples generated by our method exhibiting significantly enhanced sharpness and overall quality.

### *4.2.2 Number of Clients*

Deepening our investigation, we also study the effect of the number of clients $C$ in Table 4.2. Note that, as we employ the same total number of samples in all experiments, the number of samples *per client* will increase with smaller $C$, and decrease with larger $C$. This allows us to observe the effect of increasing the distributed nature of the data across the client network.

One question arising from the adoption of generative models in FL settings pertains to their ability to maintain satisfactory performance when trained on a limited number of samples. Interestingly, when analyzing the results, we find that FedDiff is capable of handling a much smaller number of client training samples with little to no performance degradation. On the other hand, the discriminative model approaches quickly experience a collapse in performance when expanding to 20 clients. In the heterogeneous environment of federated learning, the local optimization of a discriminative model on a highly-imbalanced and small dataset proves challenging. Rather than being an overwelming burden, such a situation is handled well by FedDiff, as its sole focus is to capture the subsequently smaller distribution. Furthermore, we again find that FedDiff outperforms FedCVAE in all settings, further illustrating the potential for diffusion models in one-shot FL.

### *4.2.3 Resource Requirements*

To further explore the efficacy of our method, we also examine resource factors, including FLOPs and parameter count, for each method deployed on a single client. Notably, our FedDiff approach consistently delivers superior accuracy with comparable computational resources to other methods. We extend this assessment to a reduced model size (FedDiff$_s$ in Table 4.3), reaffirming its strong performance relative to alternative methods. This analysis underscores the effectiveness of FedDiff, even when deployed on hardware with modest computational capabilities.

It is pertinent to emphasize that training diffusion models within the FedDiff framework is *no more intricate than conventional methodologies and remains highly viable for FL.* The computational complexity aligns with training a conventional CNN model with a modest number of parameters, and we employ the same number of local epochs as previous work with CNNs [109]. Importantly, the training process entails selecting random steps in the diffusion process at any given training iteration, eliminating the necessity for sequential steps during training. During inference, the gen-

Table 4.3: Accuracy versus FLOPs and parameter count (Params) for each method on a single client. Our FedDiff approach consistently attains heightened accuracy levels while maintaining very reasonable resource demands on par with other methodologies. We also evaluate our method with a scaled-down model variant (FedDiff$_s$), further confirming its performance relative to alternative approaches. This analysis underscores the realistic feasibility of our FedDiff framework.

| | Resources | | Accuracy | | |
|---|---|---|---|---|---|
| Method | MFLOPs ↓ | Params ↓ | FashionMNIST | PathMNIST | CIFAR-10 |
| FedAvg | 479.92 | 6.44M | 29.50±10.6 | 22.05±8.20 | 19.24±3.77 |
| DENSE | 479.92 | 6.44M | 28.92±17.3 | 29.26±10.7 | 21.40±2.73 |
| OneShot-Ens | 479.92 | 6.44M | 33.79±17.9 | 34.94±9.32 | 23.38±3.62 |
| FedCVAE | 79.00 | 5.97M | 78.81±3.25 | 44.81±1.41 | 36.06±3.27 |
| **FedDiff** | 301.14 | 5.81M | **86.81±0.54** | **70.61±1.37** | **56.57±2.42** |
| **FedDiff$_s$** | **77.43** | **1.46M** | **85.90±0.92** | **70.53±5.61** | **50.08±1.87** |

eration process involves sequential denoising steps; however, this poses no issue for the clients, as generation occurs at the server in FedDiff. Therefore, FedDiff is an effective and practical approach for providing strong performance.

## 4.3 ***RQ2***: Privacy Considerations

Privacy holds paramount importance in one-shot FL. The trained client model may be repeatedly utilized, or even exchanged in a model market context, and therefore safeguarding the privacy of the model before it leaves the client is imperative. However, other SOTA works have not experimented with DP constraints, nor have they thoroughly explored this aspect, often leaving privacy discussions simply as a possibility for future work [109, 37]. In the subsequent sections, we meticulously investigate privacy from various perspectives and dive into our research question ***RQ2***.

### *4.3.1 Differential Privacy*

Differential privacy is the widely accepted standard for ensuring privacy of a model, as it offers a provable guarantee of privacy [1, 25, 26, 24, 31]. Utilizing $(\epsilon, \delta)$ differential privacy during model training guarantees comprehensive privacy protection, encompassing not only the model's parameters and activations, but also extending to all subsequent downstream operations such as inferences, fine-tuning, and distillation. It is important to note that a model trained under $(\epsilon, \delta)$ DP **safeguards the privacy of every training sample, regardless of its qualities or uniqueness** [103]. Specifically, we train all approaches under $(\epsilon, \delta)$ DP at the clients for various privacy levels of $\epsilon$ = 50, 25, and 10, with $\delta = 10e^{-5}$, $C = 10$, and $\alpha = 0.01$. Lower $\epsilon$ values correspond to a tighter privacy budget, and the stated budget is for the entire training of each local model. We employ the Opacus [103] library for implementing DP. We present the results for all approaches in Table 4.4.

As expected, all methods experience a drop in performance when trained under DP settings. Nonetheless, FedDiff still stands out, outperforming all other methods by a significant margin. Particularly for FashionMNIST, FedDiff experiences comparatively less accuracy drop under DP than FedCVAE. As articulated in our initial motivations outlined in Section 4.1, DP training introduces noise into the training process, exacerbating the complexity of optimization. In such scenarios, the simplicity of the training paradigm employed by diffusion models becomes notably advantageous. Overall, we show that FedDiff is a strong approach even when DP is employed.

### *4.3.2 Addressing Memorization*

In a recent study, [9] explored diffusion models and identified their ability to memorize samples under certain conditions. They acknowledge differential privacy as the gold standard defense strategy,

Table 4.4: Differential privacy (DP) results under various $\epsilon$ budgets. We set $C = 10$ and $\alpha = 0.01$ as the default setting. **Even under DP constraints, FedDiff is a particularly viable approach, outperforming all other SOTA one-shot FL methods**.

| | Method | $\epsilon = 50$ | $\epsilon = 25$ | $\epsilon = 10$ |
|---|---|---|---|---|
| FashionMNIST | FedAvg | 21.04±12.1 | 20.82±12.3 | 20.39±12.6 |
| | DENSE | 26.34±9.03 | 26.29±9.81 | 24.29±15.6 |
| | OneShot-Ens | 31.27±10.9 | 31.32±10.1 | 29.99±16.7 |
| | FedCVAE | 44.40±1.70 | 43.89±2.53 | 41.65±3.19 |
| | **FedDiff** | **75.92±1.86** | **75.08±2.13** | **73.43±1.50** |
| PathMNIST | FedAvg | 16.98±8.93 | 15.30±6.44 | 14.85±4.19 |
| | DENSE | 20.56±6.59 | 19.19±3.76 | 18.41±1.86 |
| | OneShot-Ens | 24.59±7.63 | 23.38±2.60 | 22.23±2.02 |
| | FedCVAE | 24.06±1.57 | 22.15±2.68 | 20.51±1.29 |
| | **FedDiff** | **54.98±2.04** | **51.51±1.85** | **47.85±3.68** |
| CIFAR-10 | FedAvg | 16.35±1.52 | 15.39±1.87 | 15.07±2.12 |
| | DENSE | 16.97±2.35 | 15.68±2.27 | 14.98±1.25 |
| | OneShot-Ens | 17.73±2.71 | 17.34±2.35 | 15.72±1.34 |
| | FedCVAE | 16.29±1.55 | 16.08±2.19 | 15.86±2.83 |
| | **FedDiff** | **32.93±1.93** | **31.76±2.68** | **27.78±1.66** |

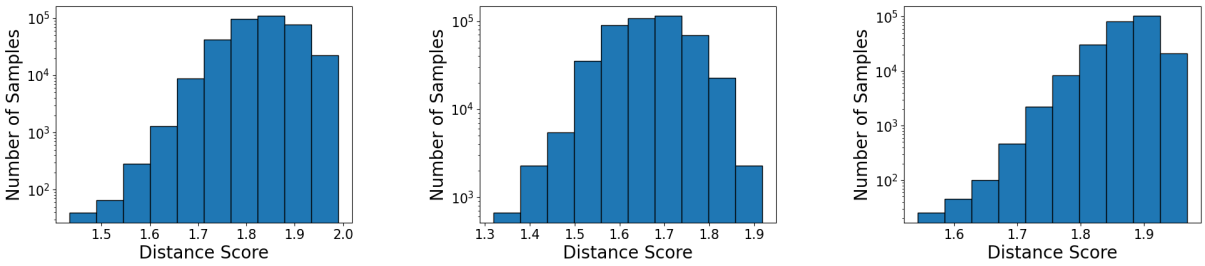
but did not provide completed experiments to this end. Therefore, we evaluate the effectiveness of DP to this end, assessing memorization within our DP-trained models to investigate whether inadvertent reproduction of the training data can be eliminated.

To conduct this study, we adopt the evaluation methodology established by [9] to scrutinize the occurrence of memorization. Specifically, from each DP-trained diffusion model, we generate a vast number of samples (five times the size of the training set). Subsequently, for each generated image, we assess potential memorization compared to the original training samples using the adaptive distance metric introduced by [9],

$$\ell\left(\hat{x}, x; S_{\hat{x}}\right) = \frac{\ell_2(\hat{x}, x)}{\alpha \cdot \mathbb{E}_{y \in S\hat{x}}\left[\ell_2(\hat{x}, y)\right]}. \tag{4.1}$$

Here, $S_{\hat{x}}$ denotes the set comprising the $n$ nearest elements from the training dataset to the example $\hat{x}$. The resulting distance metric yields a small value if the extracted image $x$ exhibits significantly closer proximity to the training image $\hat{x}$ compared to the $n$ closest neighbors of $\hat{x}$ within the training set. The idea is to find generated images that are unusually close to an original training image as indication of memorization. We set $\alpha = 0.5$ and $n = 50$ as in [9].

[9] did not define the specific threshold for Equation 4.1 for marking when a sample is considered memorized. Therefore, we consider the intuitive threshold to be less than 1, as this would indicate that the distance from the extracted image to the training image is less than half of the average distance to the closest $n$ neighbors. Upon conducting this assessment, we do not find any instances of memorized samples for all datasets under such definition, even at an elevated privacy parameter of $\epsilon = 50$, with the closest distance values being $\sim$1.3. We show the histogram of scores for all samples on each dataset in Figure 4.4.

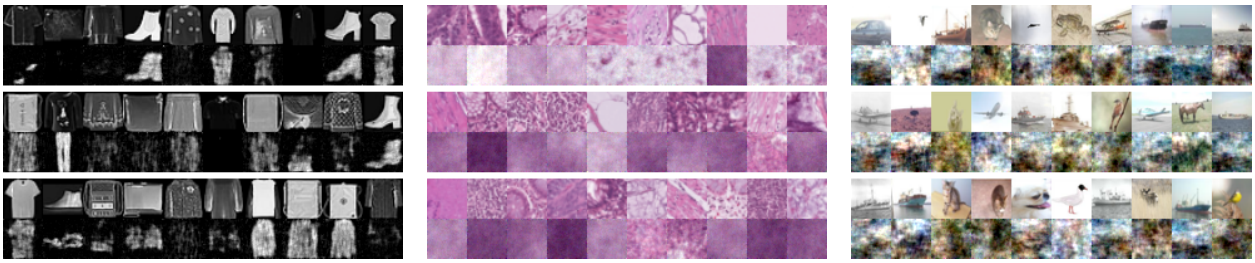



(a) FashionMNIST (b) PathMNIST (c) CIFAR-10

Figure 4.4: Histogram of distance scores for all generated samples at $\epsilon = 50$ to corresponding closest training image by Eq. 4.1 on each dataset. Note that the y-axis in in *log scale*, as there are very few samples with lower scores.

Because the threshold definition for memorization could vary, we also qualitatively show the samples with the lowest distances for all datasets at $\epsilon = 50$ in Figure 4.5. Notably, the training versus

the generated samples have discernible differences, in contrast to the nearly identical samples uncovered in [9] when training large diffusion models without DP. Also, given the nature of FL, the choice of diffusion model size will typically be small (for example, ours is $\sim$5.8M parameters), and therefore will be less likely to memorize compared to the larger DMs evaluated in [9]. As DP algorithms improve, we anticipate that even better final accuracy can be achieved while maintaining guaranteed privacy in the future with FedDiff.



(a) FashionMNIST (b) PathMNIST (c) CIFAR-10

Figure 4.5: Qualitative comparison of original training samples and generated samples at $\epsilon = 50$. We show the closest 30 samples via the similarity metric in Equation 4.1. In each stacked row, the original samples are on top, with the corresponding nearest generated image immediately below. Even under the loosest privacy guarantee of $\epsilon = 50$, we do not see blatant memorization.

### *4.3.3 Fourier Magnitude Filtering*

While FedDiff performs comparatively well against other SOTA one-shot FL methods under DP constraints, we further investigate a simple approach to improve our method, particularly for complex data most affected by DP. As shown in Figure 4.5, we note that the generated samples under DP can lack details, exhibiting reduced structure. Therefore, it may be advantageous to sort out and remove such poor quality samples from the final synthetic dataset prior to conducting the training of the global model.

In order to understand the impact of prioritizing data quality on performance, we conducted an

initial experiment. For the CIFAR-10 dataset, we leverage a centralized pretrained classifier as an oracle to discern high and low quality samples. Specifically, we selectively retain samples for which the oracle accurately classifies and discarded those it misclassifies. This provides a way to filter out samples that are likely irrelevant or misleading for training a model. We then train the global model exclusively on the curated dataset of accurate samples and evaluate. This investigation yields a discernible improvement ranging from approximately $2\%$ to $4\%$ in final global model accuracy compared to training with all generated data, verifying an importance for data quality. Therefore, a critical question arises from this observation: how can we conduct sample filtration in the absence of an oracle?

To do so, we look to the Fourier domain for a potential source of information. As inspiration, we note that the use of the magnitude of local client images has been utilized in FL to assist in domain generalization across clients by providing low-level "style" information without the high-level semantics encoded in the phase [61]. In our case, we propose to leverage the Fourier magnitude information as a potential referenceable indicator to guide the sample filtering process. Furthermore, we are able to do so under very tight DP guarantees.

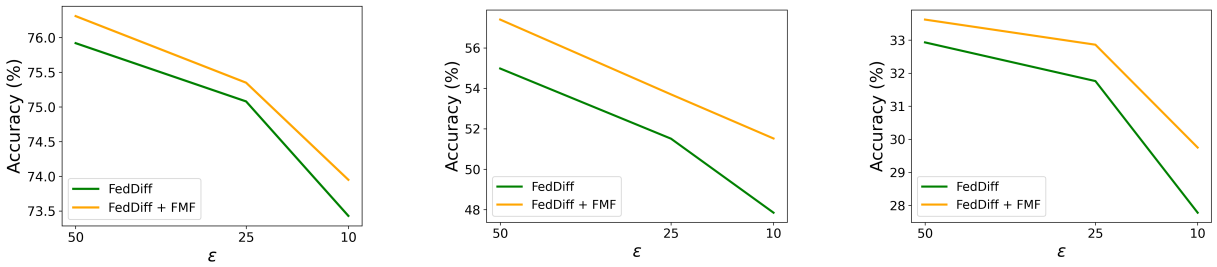
Specifically, on the client, we take the Fourier transform of the local samples and extract the magnitude information. For each client $c$, we gather the average sample magnitude with

$$\bar{M}_c = \frac{1}{n_c} \sum_{i=1}^{n_c} |\psi(\mathbf{x}^i)|, \tag{4.2}$$

where $\psi$ is the 2D Fourier transform operation, $\mathbf{x}^i$ is a sample, and $n_c$ is the total number of samples in client $c$. $\bar{M}$ is bundled with the model and transmitted by the client to the relevant global party.

As in our standard global training procedure, samples are generated with the client-trained diffusion models to form a synthetic set. Prior to conducting global training, we calculate a sample

score $s$ for the generated data $z$ from each diffusion model from the clients, $s_{\mathbf{z}_c^i} = \| |\psi(\mathbf{z}_c^i)| - \bar{M}_c \|_2$. We can then leverage this information to guide the removal of irrelevant samples, forming the final training set $\mathbf{G}$ by removing $\gamma$ percent of the generated data with the highest $s$ (larger magnitude difference). To continually ensure privacy guarantees, we apply DP in the FMF calculation. We do so by employing the DP bounded mean [55] from PyDP[1] to calculate the average magnitude $\bar{M}_c$ at each client. This allows us to precisely manage any degree of privacy leakage for $\bar{M}_c$ and include it in the overall privacy budget.

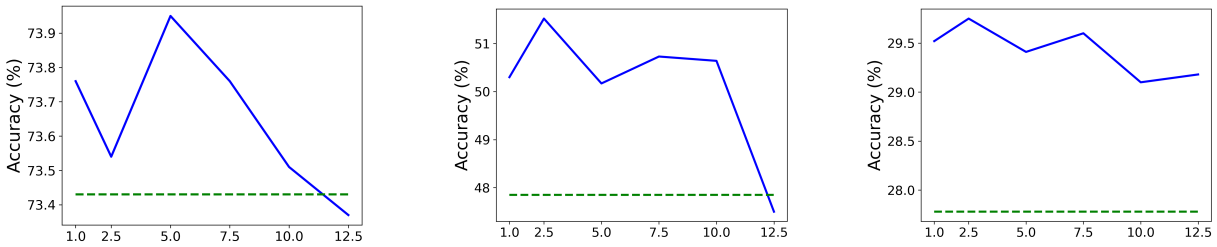



(a) FashionMNIST (b) PathMNSIT (c) CIFAR-10

Figure 4.6: Results with our Fourier Magnitude Filtering under DP. FedDiff is in **green** and FedDiff+FMF in **orange**. Our FMF approach provides a simple way to boost accuracy, especially in more challenging scenarios such as lower $\epsilon$ budgets and more complex datasets. We plot the mean across three runs with different seeds for each setting. Additional $\gamma$ ablations are provided in Figure 4.7

In Figure 4.6, we show the results of applying our FMF approach with FedDiff for the same overall DP budgets as Table 4.4. FMF is particularly effective in the most difficult scenarios, helping to mitigate the performance drop in harsh FL environments. For example, FMF provides over 3.5% and 2% improvements with PathMNIST and CIFAR-10 in the challenging $\epsilon = 10$ setting. Overall, FMF is a simple way to boost performance in one-shot FL under DP.

[1] https://github.com/OpenMined/PyDP




(a) FashionMNIST (b) PathMNSIT (c) CIFAR-10

Figure 4.7: Ablation study of $\gamma$ in FMF under the $\epsilon$ = 10 setting. The accuracy of FedDiff is in **green** and FedDiff+FMF for various $\gamma$ in **blue**. Generally, data filtering within the range of 1% to 10% produces positive outcomes, resulting in improved performance, with approximately 5% serving as an effective default choice. We plot the mean across three runs with different seeds for each setting.

### *4.3.4 FMF $\gamma$ Ablation*

In Figure 4.7, we present the outcomes obtained using FedDiff+FMF under $\epsilon = 10$ across a range of $\gamma$ values, encompassing data filtering percentages spanning from 1% to 12%. Our findings indicate that, in general, data filtering within the 1% to 10% range yields favorable results and leads to performance enhancements, with around 5% being a great default. Interestingly, the the degree of improvement provided by FMF becomes more pronounced and consistent as the dataset becomes more challenging. This phenomenon aligns with the anticipated trends, as more intricate datasets inherently pose a greater challenge, making it less likely for the generators to consistently produce high-quality samples. Consequently, the need for data filtering becomes more pronounced in such scenarios to enhance sample quality. This trend is also favorable since it addresses the specific need for improvement, especially in cases where performance is suboptimal and the challenges are more pronounced.

### 4.3.5 *Additional $\epsilon$ Experiment*

To demonstrate the feasibility of FedDiff under more stringent budget constraints, we conduct an experiment with an even tighter privacy budget of $\epsilon = 1$ in Table 4.5. Despite facing such stringent privacy constraints, FedDiff maintains a higher level of performance at $\epsilon$ = 1 than all other methods in Table 4 of the main paper at $\epsilon$ = 50.

Table 4.5: Differential privacy results under $\epsilon$ = 1.

| Privacy $\epsilon$ = 1 | **FedDiff** |
|---|---|
| FashionMNIST | **65.53±0.70** |
| PathMNIST | **44.38±3.35** |
| CIFAR-10 | **21.48±1.53** |

### 4.3.6 *Discussions, Limitations and Broader Impact*

**Model Heterogeneity**. In real FL systems, model heterogeneity may often occur [109, 37]. For instance, some clients may have architecture variations in their models or have smaller or larger models depending on their computing capabilities. Therefore, clients may have different architectures of similar generation capability, or even differing capabilities depending on the requirements of each client. Our approach allows for flexibility to accommodate such system diversity across clients. In FedDiff, we generate data from the client models and employ that synthetic data for global training, and therefore can leverage varying models without the worry of reconciling the weights themselves.

**Limitations and Broader Impact**. One downside of our method is that the generated data, particularly under DP constraints, still lacks in quality and effectiveness for global model training versus using true data. For instance, with DP on CIFAR-10 as shown in Figure 4 in the main paper, the data loses a substantial amount of structure. An interesting direction for future work would be to study how to further improve the quality of the generated data and its usefulness for global model training while maintaining privacy.

Looking at the broader impact of our work, FL depends on the diversity of data contributed by dif-

ferent participants. If biases exist in the local datasets, they can be propagated and amplified during the model training process. This could lead to unintended algorithmic biases and discrimination in the resulting models. Ensuring diversity and fairness in the data used for FL is an important research direction to mitigate this risk and promote equitable outcomes [2], particularly in the highly data heterogeneous environments explored in this work. Furthermore, as we have discussed throughout our paper, the privacy of client data is important in FL. To mitigate risks in this regard, we take many precautions to preserve privacy of the clients participated in the FL process though the use of DP, and operating within the one-shot setting to reduce the chance of eavesdropping.

## 4.4 Summary

In summary, this chapter addresses two valuable research questions in one-shot FL. Firstly, we investigate the potential of diffusion models for one-shot FL, and present the pioneering effort in this direction. In our investigation, we unveil the unique advantages that DMs offer, showcasing their potential to enhance the overall performance and tackle heterogeneity across diverse settings with our proposed approach, FedDiff. Secondly, we study privacy in SOTA one-shot FL and contribute a thorough investigation under provable privacy budgets, as well as address memorization concerns. Furthermore, to enhance performance under harsh DP conditions, we propose a novel and pragmatic solution, Fourier Magnitude Filtering, to boost the efficacy of generated data for global model training by eliminating low-quality samples. We hope our work will inspire the community and foster further research in this direction to improve communication-efficient one-shot FL with generative models.

# CHAPTER 5: RESOURCE EFFICIENT CONTINUAL PRETRAINING FOR GEOSPATIAL FOUNDATION MODELS

The work in this chapter has been published in the following paper:

*Towards Geospatial Foundation Models via Continual Pretraining. Matias Mendieta, Boran Han, Xingjian Shi, Yi Zhu, Chen Chen. International Conference on Computer Vision (ICCV), 2023*

---

Centralized training systems require high efficiency, particularly emphasizing the optimization of computational resources across training and inference phases. In specialized domains like remote sensing, where annotated data is often scarce for downstream tasks, labeling efficiency becomes crucial for achieving favorable outcomes. Enhancing both computational and labeling efficiency holds significant potential to enhance the effectiveness and practicality of centralized training models. Consequently, we propose to tackle resource and label efficiency in the centralized setting through a continuous pretraining approach for geospatial foundation models, a domain critical for advancing earth understanding and monitoring. In the following sections, we discuss the pretraining data selection (Sec, 5.1), investigate vanilla continual pretraining (Sec. 5.2), and present our GFM method (Sec. 5.3).

## 5.1 Pre-training Data Selection

A particularly common choice of source data among geospatial contrastive pretraining works is Sentinel-2 imagery [66, 6, 88] due to its large corpus of available data and ease of access. Therefore, to begin our study, we first gather a pretraining dataset of 1.3 million Sentinel-2 images using

the sampling technique from [66]. After gathering the Sentinel-2 data, we employ it to pretrain a Swin-B [62] model with the masked image modeling (MIM) objective from [96]. We then finetune and evaluate this model on a wide variety of downstream datasets to get a broad understanding of its performance potential in many tasks (see Section 5.4 for task details). For a comparison, we finetune the ImageNet-22k pretrained Swin-B from the official Swin Transformer repository [62] on all downstream tasks as a baseline. In order to compare these models across all tasks, we introduce an average relative performance metric (ARP) in which we take the relative difference on each task with respect to the ImageNet-22k baseline, and then average that difference:

$$\text{ARP}(M) = \frac{1}{N}\sum_{i=1}^{N}\frac{\text{score}(M, \text{task}_i) - \text{score}(\text{baseline}, \text{task}_i)}{\text{score}(\text{baseline}, \text{task}_i)}. \tag{5.1}$$

Here "baseline" is the Swin-B model pretrained on ImageNet-22k, as mentioned above. $M$ denotes the model for performance evaluation, and N is the number of tasks. There are 7 tasks used in Section 5.4 covering important geospatial applications such as classification, multi-label classification, semantic segmentation, change detection, and super-resolution. The reported ARP value is scaled by 100 to show as a percentage.

We compare these two models in Table 5.1. Interestingly, we find that the Sentinel-2 model performs poorly on downstream tasks compared to the ImageNet-22k baseline. To investigate further, we visualize multiple samples from Sentinel-2 in the left columns of Figure 5.1. Upon inspection, we note that the feature diversity within a single image and across images of Sentinel-2 is perceivably low. To further quantify this suspicion, we calculate the average image entropy over a randomly sampled set of 3000 images from the collected Sentinel-2 data as well as the typical ImageNet dataset as a baseline. Overall, the Sentinel images have an average entropy of 3.9 compared to 5.1 of ImageNet. Such an evaluation provides insights into the potential pitfalls of Sentinel-2 data in pretraining transformers. For MIM objectives, training data with a substantially

Figure 5.1: We visualize some example images from the pretraining datasets with Sentinel-2 (left) and GeoPile (right). Sentinel-2 has noticeably much lower feature diversity within a single image and across images than that of our GeoPile pretraining dataset.

lower entropy can make for an easier reconstruction task, since masked regions may be more similar to their neighbors. Therefore, the network does not have to work as hard to fill in the blanks, limiting the learning potential. Overall, these result indicate that the noticeably narrow scope of features and limited per-sample information in Sentinel-2 data may be limiting the potential of the pretrained model.

Therefore, we set out to collect a diverse geospatial pretraining dataset. Sourcing from both labeled and unlabelled data, we form a new pretraining dataset which we term GeoPile. The breakdown of GeoPile is shown in Table 5.2. For textural detail, we ensure a variety of ground sample distances (GSD), including images with much higher resolution than Sentinel-2 (which has a GSD of 10m). Furthermore, the selected labeled datasets encompass a wide variety of classes from general remote sensing scenes, ensuring visual diversity across samples. We calculate the average entropy of our

Table 5.1: Dataset Analysis. To evaluate each method, we finetune the pretrained model on seven different tasks, outlined in Section 5.4 and report the ARP metric defined in Equation 5.1. We also report the training time in hours on a V100 GPU, as well as the carbon impact estimations[1] in kg $CO_2$ equivalent [53]. Overall, our collected GeoPile pretraining dataset significantly improves downstream performance. † indicates the vanilla continual pretraining approach of initializing the model with ImageNet-22k weights prior to conducting MIM training on GeoPile. To further improve the performance in an efficient manner, we introduce our continuous pretraining paradigm GFM.

| Method | # Images | Epochs | ARP ↑ | Time ↓ | $CO_2$ ↓ |
|---|---|---|---|---|---|
| ImageNet-22k Sup. | 14M | - | 0.0 | - | - |
| Sentinel-2 [66] | 1.3M | 100 | -5.83 | 155.6 | 22.2 |
| GeoPile | 600k | 200 | 0.92 | 133.3 | 19.0 |
| GeoPile† | 600k | 200 | 1.24 | 133.3 | 19.0 |
| GeoPile† | 600k | 800 | 1.45 | 533.2 | 76.0 |
| GFM | 600k | 100 | 3.31 | 93.3 | 13.3 |

Table 5.2: Breakdown of datasets in the GeoPile. We gather approximately 600k samples from a combination of labeled and unlabeled satellite imagery with various ground sample distances and scenes.

| Dataset | # Images | GSD | # Classes |
|---|---|---|---|
| NAIP [5] | 300,000 | 1m | n/a |
| RSD46-WHU [64] | 116,893 | 0.5m - 2m | 46 |
| MLRSNet [75] | 109,161 | 0.1m - 10m | 60 |
| RESISC45 [16] | 31,500 | 0.2m - 30m | 45 |
| PatternNet [111] | 30,400 | 0.1m - 0.8m | 38 |

GeoPile dataset, and find it to be 4.6, much higher than that of Sentinel-2. Furthermore, the textural and visual diversity is qualitatively evident in Figure 5.1. In Table 5.1, the enhancing effect of the data selection is clearly shown by the substantial performance increase.

[1] $CO_2$ estimations were completed with mlco2.github.io from [53].

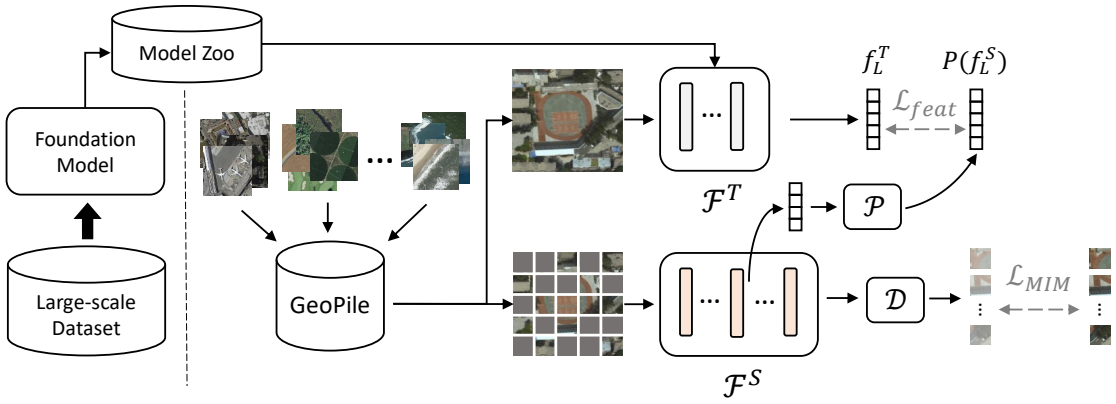



Figure 5.2: Our GFM continual pretraining pipeline, which leverages publicly-available large-scale models in concert with our compiled geospatial dataset and pretraining objective. First, we select a concise set of data from various sources, which we term GeoPile (Section 5.1). Next, we train GFM with our multi-objective continual pretraining approach. Our GFM framework is constructed as a teacher-student paradigm, with two parallel model branches. The teacher $\mathcal{F}^T$ is initialized with ImageNet-22k weights (top) and frozen during training. The student $\mathcal{F}^S$ is initialized from random initialization (bottom), and is trained to serve as the final geospatial foundation model. In a continual pretraining fashion, we leverage the intermediate features of an ImageNet-22k pretrained model to guide and quicken learning. Furthermore, we build in an MIM objective on the student branch to learn valuable in-domain features directly from the geospatial data.

## 5.2 Vanilla Continual Pretraining

Next, after establishing our pretraining data selection, we investigate an alternate pretraining paradigm that bridges the gap between the two common approaches mentioned in Section 1.2.1. Specifically, we investigate the potential of continual pretraining in the context of geospatial pretrained models. To do so, we first employ the vanilla continual pretraining approach; that is, using the ImageNet-22k weights as initialization prior to beginning the pretraining step with GeoPile. We find this to be helpful in improving performance over starting from scratch. This validates the possibility of continual pretraining as a beneficial paradigm to provide performance gain without additional resource costs. Nonetheless, the improvement is still limited, with $\sim$0.3% ARP increase over starting from scratch and $\sim$1.24% ARP over the baseline.

To further improve the performance of our pretrained model in comparison to the ImageNet-22k baseline, we increase the number of pretraining epochs in the next row of Table 5.1. While we are able to make improvements, this comes at the cost of substantially more computational cost and carbon footprint for marginal gain. Therefore, we ask the question: how can we significantly improve the performance further while maintaining minimal compute and carbon footprint overhead? To this end, we propose a simple and efficient approach for building geospatial pretrained models capable of strong downstream performance.

## 5.3 GFM Pretraining

A significant number of geospatial foundation model studies disregard the existing large-scale model representations. This is far from ideal, particularly for large transformer models known to require a vast amount of data and compute power to train. Instead, we reason that the valuable knowledge available in models like those trained on ImageNet-22k should be leveraged to produce strong performance with minimized overhead. To this end, we propose an unsupervised multi-objective training paradigm for effective and efficient pretraining of geospatial models, illustrated in Figure 5.2.

There are two main components in our framework. First, we randomly initialize an encoder $\mathcal{F}^S$ and decoder $\mathcal{D}$ set up for MIM as in [96]. During training, the input is randomly masked, and the network attempts to reconstruct the image at the output. This MIM objective is enforced with an L1 loss [96]:

$$\mathcal{L}_{MIM} = \frac{\|\mathbf{O}_\kappa - \mathbf{G}_\kappa\|_1}{N}, \tag{5.2}$$

where $\mathbf{O}_\kappa$ are the original pixel values from $\kappa$ masked regions, $\mathbf{G}_\kappa$ are the generated reconstructions for those regions, and $N$ is the total number of masked pixels.

For the continual pretraining of our framework, we initialize a second encoder branch $\mathcal{F}^T$ up to a chosen stage $L$ and load the ImageNet-22k pretrained weights. This branch behaves as a form of teacher during the training process to the student branch ($\mathcal{F}^S$), which will serve as our final model. For the ImageNet teacher, we freeze the weights, to both ensure that the structured representations are maintained during the training process, and also reduce the computation required during optimization.

Rather than using the masked input as in the student branch, the teacher receives the unmasked image as input, and provides a feature output $f_L^T$ at stage $L$. This feature has access to the full context of the input, enabling it to capture informative representations. We utilize this feature to guide the representations of the student, and form a secondary objective with the cosine similarity between branch features:

$$\mathcal{L}_{feat} = -\frac{P(f_L^S)}{\|P(f_L^S)\|_2} \cdot \frac{f_L^T}{\|f_L^T\|_2}, \tag{5.3}$$

where $f_L^S$ and $f_L^T$ are the intermediate features of the student and teacher branches at stage $L$, and $\mathcal{P}$ is an linear projection layer. Therefore, the final loss during training is simply the summation of these objectives:

$$\mathcal{L} = \mathcal{L}_{MIM} + \mathcal{L}_{feat}. \tag{5.4}$$

This training paradigm enables an ideal two-fold optimization. Distillation from the intermediate features of the teacher ensure that the student can benefit from the teacher's diverse knowledge, learning more in less time. Furthermore, the student is simultaneously given freedom to adapt to in-domain data through its own pretraining objective, gathering new features to improve performance.

We analyze the ARP and resource cost of this approach in Table 5.1. Notably, our GFM is able

to achieve better overall performance with substantially less computation and emissions impact compared to vanilla continual pretraining with the same dataset, illustrating that our multi-objective continual pretraining paradigm is an effective method for training these models. Comparatively, the SOTA geospatial pretrained method SatMAE [17] requires 768 hours on a V100 GPU and 109.44 kg equivalent $CO_2$ according to their reported results. Therefore, GFM enables more than 8× reduction in total training time and carbon impact. Moreover, we find that the performance of SatMAE is often not superior to the off-the-shelf ImageNet-22k pretrained ViT (Section 5.4). This implies that building powerful geospatial pretrained models from scratch is challenging and further underscores the benefits of utilizing continual pretraining instead. We show an overview of these results in Figure 5.3, and detail them in the following section.

## 5.4 Experiments

To verify the effectiveness of our model in detail, we conduct experiments on seven geospatial datasets of various tasks including change detection (Section 5.4.3), classification (Section 5.4.4), segmentation (Section 5.4.5), and super-resolution (Section 5.4.6).

For pretraining, we employ 8 NVIDIA V100 GPUs with a batch size of 2048 (128 per GPU) and the image size of 192×192. All pretraining settings are the same as in [96]. For downstream tasks, 4 NVIDIA A10G GPUs are employed. During the pretraining stage, we utilize RGB bands as they are most commonly available among data sources and tasks. For downstream tasks with additional band inputs, we initialize the RGB patch embeddings with the pretrained weights and randomly initialize the remaining channels. Potentially improving performance even further though the employment of additional data modalities will be an intriguing avenue for future research.

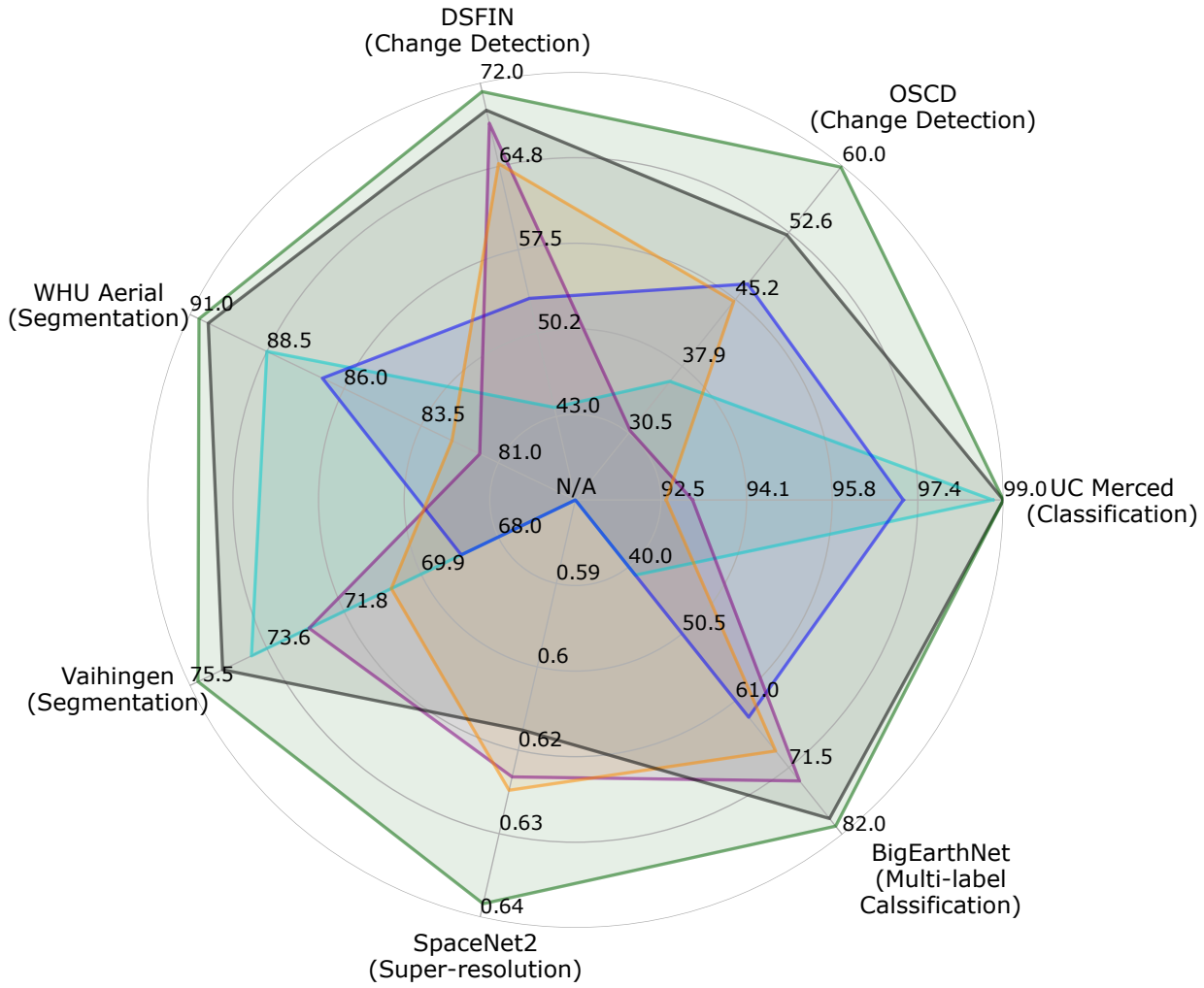



Figure 5.3: Our geospatial foundation model (GFM) achieves favorable performance on a broad set of tasks in comparison to other state-of-the-art geospatial pretraining methods (SeCo [66], SatMAE [17]) and ImageNet supervised pretraining baselines. Legend is as follows. Cyan: ImageNet-1k Supervised (ResNet50), Blue: SeCo [66], Purple: ImageNet-22k Supervised (ViT), Orange: SatMAE [17], Gray: ImageNet-22k Supervised (Swin), Green: GFM (ours).

### *5.4.1 Training Details*

We provide the training details for the various stages and tasks in our evaluation. Code, model weights, and GeoPile dataset are publicly available at `https://github.com/mmendiet/GFM`.

#### *5.4.1.1 Change Detection*

We modify the MMsegmentation [18] framework to conduct our change detection experiments. For OSCD, as the raw image size is large but the number of samples is very small, we tile the images into 192×192 pixels and train for 4000 iterations. We utilize the RGB bands for OSCD as in [66]. For DSFIN, we train for 10k iterations with image size 512×512. We employ an SGD optimizer with a learning rate of 0.01 and weight decay of 5.0e-4, and the default polynomial scheduler of [18].

#### *5.4.1.2 Classification*

On UC Merced, we train with a batch size of 1024 (128 per GPU) at image size 256×256. We train for 100 epochs with a base learning rate of 1.0e-4. We employ random flip, crop and standard Mixup [108] augmentation. Optimizer, weight decay, Mixup parameters, and other training settings are the same as in [96]. For BigEarthNet, we slightly upscale the original 120×120 images to 128×128 for ease of dimensional compatibility with the Swin transformer. We then employ the same training settings as with UC Merced.

#### *5.4.1.3 Segmentation*

We employ the MMsegmentation [18] framework to conduct our segmentation experiments. For both datasets, we train for 40k iterations with an image size of 512×512. All other training settings are the same as the default configuration in [18] for the respective backbones (Swin, ViT, ResNet50) and compatible decoders (UperNet [93] for transformers and Deeplabv3 [12] for ResNets).

#### *5.4.1.4 Super-resolution*

On the SpaceNet2 super-resolution tasks, we train with a batch size of 64 (16 per GPU) with input image size 160×160 and target size 640×640. We train for 100 epochs with a base learning rate of 1.25e-5. Optimizer, weight decay, and other training settings are the same as in [96], but with no random augmentations. We employ the standard decoder from [96] to produce the original input size from the encoder features, and then upscale using a convolution-based upsampling block based on the image reconstruction module for classic super-resolution employed in [59].

### *5.4.2 Training Time and Carbon Calculations*

To calculate the $CO_2$ impact of training various models, we employ the ML $CO_2$ Impact estimator at `https://mlco2.github.io/impact` from [53]. The total impact is dependent on the hardware type, GPU provider, region, and total time used. Our pretraining experiments were conducted in the AWS US East (Ohio) region, which has a carbon efficiency of 0.57 kg eq. CO2 per kWh. For our GFM, just 93.3 V100 GPU hours are needed for training, resulting in a total carbon impact of 13.3 kg eq. CO2. This is significantly lower than the previous state-of-the-art geospatial model, SatMAE [17]. According to the reported carbon impact in their paper [17], SatMAE requires 768 V100 GPU hours and 109.44 kg eq. CO2 on the Google Cloud Platform us-central1 region, which has a carbon efficiency of 0.57 kg eq. CO2 per kWh (same as AWS US East Ohio). Therefore, GFM enables more than 8× reduction in total training time and carbon impact in comparison to SatMAE.

Table 5.3: Onera Satellite Change Detection Results

| Method | Precision ↑ | Recall ↑ | F1 ↑ |
|---|---|---|---|
| ResNet50 (ImageNet-1k) [36] | **70.42** | 25.12 | 36.20 |
| SeCo [66] | 65.47 | 38.06 | 46.94 |
| MATTER [6] | 61.80 | 57.13 | 59.37 |
| ViT (ImageNet-22k) [23] | 48.34 | 22.52 | 30.73 |
| SatMAE [17] | 48.19 | 42.24 | 45.02 |
| Swin (random)[62] | 51.80 | 47.69 | 49.66 |
| Swin (ImageNet-22k)[62] | 46.88 | 59.28 | 52.35 |
| GFM | 58.07 | **61.67** | **59.82** |

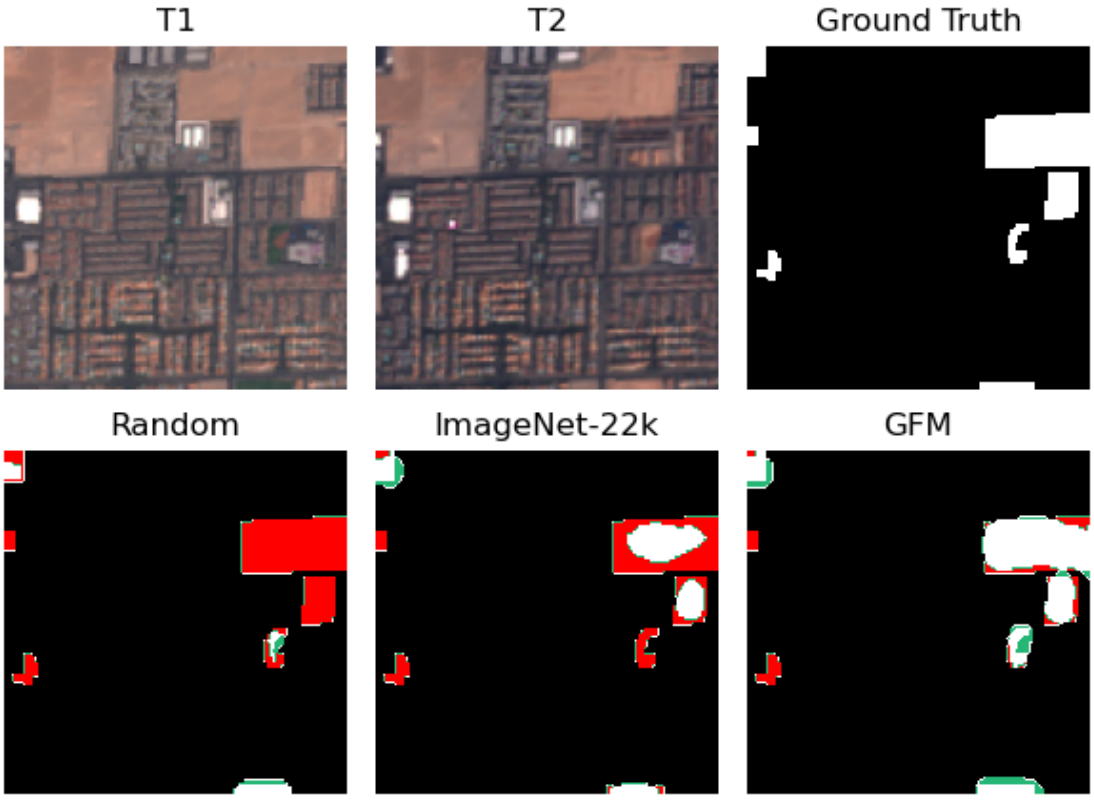



Figure 5.4: Qualitative results of downstream performance on OSCD comparing our GFM with ImageNet-22k and randomly initialized baselines. White, green, red colors show true positive, false positive, and false negative respectively.

Table 5.4: DSFIN Change Detection Results

| Method | Precision ↑ | Recall ↑ | F1 ↑ |
|---|---|---|---|
| ResNet50 (ImageNet-1k) [36] | 28.74 | **92.07** | 43.80 |
| SeCo [66] | 39.68 | 81.02 | 53.27 |
| ViT (ImageNet-22k) [23] | 70.77 | 66.34 | 68.49 |
| SatMAE [17] | 70.45 | 60.29 | 64.98 |
| Swin (random)[62] | 57.97 | 62.06 | 59.94 |
| Swin (ImageNet-22k)[62] | 67.11 | 72.33 | 69.62 |
| GFM | **74.83** | 67.98 | **71.24** |

### *5.4.3 Change Detection*

Change detection is a particularly important remote sensing task, helping us understand how humans interact with our planet over time, and natural phenomena that change our planet's landscape. We conduct experiments on both the Onera Satellite Change Detection (OSCD [10]) in Table 5.3 and DSIFN [107] in Table 5.4.

OSCD consists of 14 image pairs extracted from various regions around the world within a three year period of 2015 to 2018. The images are taken from Sentinel-2 with GSDs ranging from 10m to 60m, and split into 14 images for training and 10 for evaluation. The annotations indicate whether the change has occurred on a pixel level, and focus primarily on urban developments. Similarly, we also test our method on DSIFN dataset. This dataset contains high-resolution imagery, such as WorldView-3 and GeoEys-1 [107]. This dataset contains 3490 high resolution samples for training and 48 images for evaluation respectively. Every pair of images from a given location at two different timestamps will be fed into the swin encoder [62] for feature extraction. The difference between the features from each pair is computed and fed into an UPerNet [93] to generate the final binary segmentation masks [66, 11]. The encoder is initialized with the pretrained weights.

For both datasets, we report the precision, recall, and F1 score on the "change" class. As the results presented from OSCD (Table 5.3 and Figure 5.4) and DSIFN (Table 5.4), GFM shows a consistent improvement over the ImageNet-22k baseline across both datasets. Notably, SatMAE is able to improve over its ImageNet-22k baseline on OSCD, but lags behind on DSIFN. This further highlights the difficulty of training large vision transformers from scratch that can perform consistently across different GSDs.

### *5.4.4 Classification*

Another common remote sensing application is that of classification. We evaluate two datasets common in the literature [66, 6]: UC Merced Land Use Dataset [99] and BigEarthNet [86]. The UC Merced Land Use Dataset is a classic dataset in the remote sensing field. It contains 21 classes, each with 100 images at 256x256 pixels and an approximate GSD of 1 foot. We split the data into train and validation according to [22]. BigEarthNet [86] (BEN) is a large-scale remote sensing dataset for multi-label classification. The data consist of 12-band Sentinel-2 images with sizes of 120x120, 60x60, and 20x20 pixels for the bands at 10m, 20m, and 60m GSDs, respectively. We employ the data split and 19 class evaluation as common in the literature [69, 66, 17].

In Table 5.5, we report the classification accuracy on UC Merced (UCM) and mean average precision results on BigEarthNet (BEN) for all methods. On UC Merced, we note the SeCo [66] pretrained model performs significantly worse than its ImageNet-1k pretrained counterpart with ResNet-50. These two datasets are very different in both classes, satellite source, and GSDs, and therefore having a diverse feature knowledge is imperative to maintaining performance despite these distinctions. Our model can provide robust performance in both cases by leveraging ImageNet representations and remote sensing data in its learning. Furthermore, one key motivation for training a geospatial foundation model is to improve the sample efficiency for downstream tasks.

Table 5.5: UC Merced classification accuracy and BigEarthNet multi-label classification mean average precision results.

| Method | UCM | BEN 10% | BEN 1% |
|---|---|---|---|
| ResNet50 (ImageNet-1k) [36] | 98.8 | 80.0 | 41.3 |
| SeCo [66] | 97.1 | 82.6 | 63.6 |
| ViT (ImageNet-22k)[23] | 93.1 | 84.7 | 73.6 |
| SatMAE [17] | 92.6 | 81.8 | 68.9 |
| Swin (random)[62] | 66.9 | 80.6 | 65.7 |
| Swin (ImageNet-22k) [62] | **99.0** | 85.7 | 79.5 |
| GFM | **99.0** | **86.3** | **80.7** |

Notably, we find that our model maintains strong performance on BigEarthNet, even when only given 1% of the training data.

### *5.4.5 Segmentation*

Segmentation is a popular remote sensing application for enabling automated extraction of building footprints or land cover mappings over wide regions. We therefore conduct experiments on this task on two different datasets. Vaihingen [79] is an urban semantic segmentation dataset collected over Vaihingen, Germany at a GSD of 0.9m. We employ the data split implemented in the MMSegmentation library [18] for our experiments, with 344 training and 398 for validation, all with an image size of 512x512 pixels. The WHU Aerial building [43] dataset is sampled over Christchurch, New Zealand at a GSD of 0.3m. Image tiles are provided at $512 \times 512$ pixels, split into 4736 for training and 2416 for evaluation.

We report the intersect of union (IoU) segmentation results for all methods in Table 5.6. ImageNet pretrained models are notably strong performers in all cases. On both datasets, SeCo lags substantially behind its ImageNet counterpart. Interestingly, SatMAE is able to bring improvement over

Table 5.6: Results on the WHU Aerial and Vaihingen segmentation datasets. We finetune all methods for 40k iterations, and report the IoU for the building class on WHU and mean IoU (mIoU) across the 6 classes (impervious surface, building, low vegetation, tree, car, clutter) of Vaihingen.

| Method | WHU Aerial | Vaihingen |
|---|---|---|
| ResNet50 (ImageNet-1k) [36] | 88.5 | 74.0 |
| SeCo [66] | 86.7 | 68.9 |
| ViT (ImageNet-22k) [23] | 81.6 | 72.6 |
| SatMAE [17] | 82.5 | 70.6 |
| Swin (random) [62] | 88.2 | 67.0 |
| Swin (ImageNet-22k) [62] | 90.4 | 74.7 |
| GFM | **90.7** | **75.3** |

ImageNet-22k on WHU, but fails to do so to a larger degree on Vaihingen. However, our approach is able to leverage the already strong ImageNet-22k representations and guide them towards the geospatial domain, resulting in overall improvement.

### *5.4.6 Super-resolution*

In the previous experiments, we evaluated several common high-level tasks. Nonetheless, the low-level task of super-resolution is also important in the geospatial domain. For this task, we re-purpose the SpaceNet2 dataset, which contains 10,593 8-band images from four cities: Las Vegas, Paris, Shanghai, and Khartoum. The data is provided at both a GSD of 1.24m (multi-spectral, 162x162 pixels) and 0.3m (pan-sharpened multispectral, 650x650 pixels). We formulate a super-resolution task, taking as input the 1.24m multi-spectral images and generating the 0.3m pan-sharpened equivalent. We evaluate the super-resolution performance of our model and several baselines with the peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) in Table 5.7. The ViT-L ImageNet-22k model and our model are among the best in terms

Table 5.7: SpaceNet2 Super-resolution Results. Notably, while SatMAE fails to enhance its baseline (ViT ImageNet-22k), our method exhibits substantial improvement over its respective baseline (Swin ImageNet-22k) in both PSNR and SSIM.

| Method | PSNR ↑ | SSIM ↑ |
|---|---|---|
| ViT (ImageNet-22k)[23] | **23.279** | 0.619 |
| SatMAE [17] | 22.742 | 0.621 |
| Swin (random) [62] | 21.825 | 0.594 |
| Swin (ImageNet-22k) [62] | 21.655 | 0.612 |
| GFM | 22.599 | **0.638** |

of PSNR and SSIM, respectively. Interestingly, SatMAE is not able to improve over its baseline. On the other hand, our method improves considerably over its ImageNet-22k baseline.

## 5.5 Ablation Studies

We perform multiple ablation studies on the choice of distillation stage, student initialization, training objectives, the pretraining dataset components.

### *5.5.1 Distillation Stage*

When implementing our feature map distillation objective, a natural question is at which point should the mapping take place. We experiment with different locations by stage in the Swin transformer and calculate the corresponding ARP in Figure 5.5. Overall, performing the distillation after Stage 3 yields the highest ARP. Hence, we employ this scheme for all downstream experiments. This result is also intuitively expected; distilling at Stage 3 gives a large portion of the model the supervisory signal from the teacher, while still allowing for purely domain-specific feature learning in the final layers.

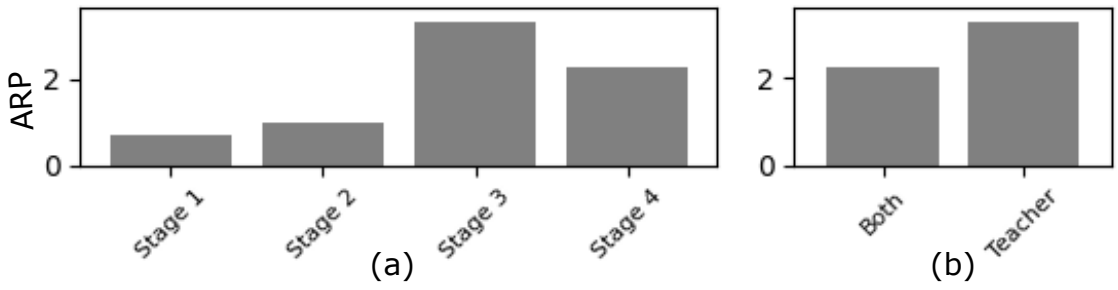



Figure 5.5: a) Distillation stage ablation results. b) Student initialization ablation results. "Both" indicates that the teacher and student branches are initialized with ImageNet weights prior to geospatial pretraining. "Teacher" indicates that just the teacher branch is initialized, as described in Section 5.3.

### *5.5.2 Student Initialization*

In our proposed framework, we maintain the teacher model frozen with ImageNet pretrained weights, and randomly initialize the student. Another alternative is to initialize the student also with ImageNet weights prior to beginning the geospatial pretraining process. However, as shown in Figure 5.5, this is not the most optimal option. Such initialization is unnecessary in our framework, since it already allows for seamless integration of ImageNet representations with valuable in-domain features. Forcibly doing so likely introduces too much bias towards the natural image representations. Therefore an unbiased student is most ideal and effective.

### *5.5.3 GeoPile Pretraining Dataset*

To ablate components of the GeoPile, we remove each dataset individually to see its relative importance. Also, we compare using just the labeled data portion and using just the unlabeled NAIP imagery portion. As expected, using just data from labeled datasets gives better performance with less images than using just images gathered from just NAIP. The human-curated samples in these

Table 5.8: GeoPile pretraining dataset ablation. We remove each dataset individually from GeoPile and report the number of images remaining and resulting ARP. The row "w/o curated datasets" removes all data other than NAIP imagery.

| Data | # Images | ARP ↑ |
|---|---|---|
| w/o WHU-RSD46 | 444,061 | 1.77 |
| w/o MLRSNet | 451,793 | 2.17 |
| w/o Resisc45 | 529,454 | 1.57 |
| w/o PatternNet | 557,554 | 1.79 |
| w/o curated datasets | 300,000 | 0.53 |
| w/o NAIP | 260,954 | 1.50 |

datasets are more likely to contain relevant objects and features, as they each correspond to a particular class of interest. Still, unlabeled data like NAIP can be sourced easily and with scale. Further scaling of both labeled and unlabeled portions could further improve performance; however, it will also increase the training time and sustainability impact. Therefore, we maintain GeoPile at approximately 600,000 images.

Table 5.9: Ablation results for the training objectives in GFM. For w/o teacher, we only conduct MIM with GeoPile. For w/o MIM, we simply perform the distillation objective from the ImageNet-22k model to our student model with GeoPile. We abbreviate the following for horizontal space: UC Merced (UCM), BigEarthNet (BEN), WHU Aerial (WHU), Vaihingen (Vai), SpaceNet2 (SN2).

| Method | OSCD (F1) | DSFIN (F1) | UCM | BEN 10% | BEN 1% | WHU | Vai. | SN2 (PSNR) | SN2 (SSIM) |
|---|---|---|---|---|---|---|---|---|---|
| w/o teacher | 57.3 | 67.65 | 98.8 | **86.5** | 80.0 | 90.5 | 74.0 | 22.509 | 0.631 |
| w/o MIM | 59.58 | **71.86** | 98.8 | 86.1 | 80.2 | 90.2 | 72.6 | 22.069 | 0.608 |
| GFM | **59.82** | 71.24 | **99.0** | 86.3 | **80.7** | **90.7** | **75.3** | **22.599** | **0.638** |

Table 5.10: Results for employing temporal pairs and datasets from SeCo [66] in our multi-objective pretraining framework. TP indicates that the teacher receives one image from a temporal pair, and the student receives the other. SI indicates that the same image is inputted to the teacher and student.

| Dataset | Inputs | OSCD (F1) | DSFIN (F1) | UCM | BEN 10% | BEN 1% | WHU | Vai. | SN2 (PSNR) | SN2 (SSIM) |
|---|---|---|---|---|---|---|---|---|---|---|
| SeCo 100k [66] | TP | 57.03 | 62.48 | 80.0 | 80.6 | 68.6 | 88.3 | 66.3 | 22.078 | 0.572 |
| SeCo 100k [66] | SI | 58.41 | 67.92 | 92.1 | 83.9 | 76.5 | 88.8 | 68.1 | 22.439 | 0.602 |
| SeCo 1M [66] | SI | 58.87 | 69.41 | 95.7 | 86.2 | 77.1 | 89.6 | 71.0 | 22.281 | 0.626 |
| GeoPile | SI | **59.82** | **71.24** | **99.0** | **86.3** | **80.7** | **90.7** | **75.3** | **22.599** | **0.638** |

### *5.5.4 Multi-objective Ablation.*

To dive deeper into the evaluation of GFM's performance, we extend our analysis by conducting experiments in which we exclude the teacher component and MIM component individually, as detailed in Table 5.9. We find that training with the multi-objective approach is the best performer overall. This shows that the integrated distillation and MIM objectives within the GFM framework both contribute to producing a well-balanced mode for downstream tasks, and are important aspects of efficient and effective geospatial learning.

### *5.5.5 Temporal Pairs Experiment*

Some works employ temporal pairs in the pretraining procedure [66, 7, 6], meaning two satellite images from the same spatial region but taken at different times. We also experiment with the use of temporal positives in our training paradigm using the dataset proposed in SeCo [66]. In this case, the teacher receives one image from a temporal pair, and the student receives the other. The temporal changes can possibly serve as a form of natural augmentation for the distillation objective. However, as shown in Table 5.10, we find that using temporal positives (TP) is worse than simply using the same image (SI) for both branches. Therefore, we simply use the same

image for both branches for other experiments. We further scale up the data by employing the 1M sample Sentinel-based dataset from SeCo. Nonetheless, GeoPile proves to be more effective as a pretraining data source for our GFM.

## 5.6 Summary and Discussion

In summary, this chapter investigates an alternative paradigm from previous work towards producing better geospatial foundation models with substantially less resource cost. To this end, we first construct a concise yet diverse collection of data from various remote sensing sources for pretraining. Second, we propose a surprisingly simply yet effective multi-objective continual pretraining paradigm, in which we leverage the strong representations of ImageNet-22k to guide and quicken learning, while simultaneously providing the freedom to learn valuable in-domain features through self-supervised learning on geospatial data. We hope that our GFM approach will serve as an example to inspire other works in investigating efficient and sustainable methods for developing geospatial foundation models.

**Broader Impact and Limitations.** As the geospatial community continues to innovate, the resulting impact promises to positively benefit both the earth and society. Automating the process of extracting useful information from geospatial data can aid scientists, engineers, and others to make data-informed decisions on infrastructure advancement, food supply improvements, and natural disaster response. A potential limitation of our GFM approach is that it may still be somewhat constrained by the performance of the ImageNet-22k model. If perhaps a model was trained from scratch on an extremely large corpus of remote sensing data, the performance may eventually also lead to improved performance over ImageNet baselines. However, this would incur a substantial amount of training time and $CO_2$ impact. Furthermore, as mentioned in Section 1.2.1, natural image models are constantly being improved and released by the general computer vision

community. Therefore, our approach enables the geospatial domain to effectively leverage these improvements for better in-domain performance with minimal carbon impact. We believe this is a sustainable way for the geospatial community to continually benefit from the most recent progress in computer vision, enabling a smarter, safer, and healthier planet.

# CHAPTER 6: CONCLUSION AND FUTURE WORK

## 6.1 Conclusion

This dissertation presents a comprehensive exploration of decentralized and centralized computer vision applications, introducing novel methodologies that enhance the efficiency and effectiveness of these systems.

In Chapter 3, we tackled the challenge of data heterogeneity in FL in an efficient and effective manner. Unlike previous studies that focus on reparameterization techniques or adjustments to aggregation schemes to counter non-IID data distributions, we propose a fundamental reevaluation of this problem through core machine learning training principles. We investigate the performance of standard regularization methods in FL and their efficacy in handling data heterogeneity. Our approach goes beyond empirical analysis, identifying Hessian eigenvalue/trace measurements and Hessian matching across clients as indicators for optimal FL methods. Through comprehensive ablation studies across diverse FL settings, we gain insights into the empirical impacts of various methods. Our findings aim to equip the FL community with valuable knowledge, fostering innovative and productive research directions. Based on our analysis and an examination of previous methods' shortcomings, we introduce FedAlign, which achieves competitive state-of-the-art accuracy while preserving memory and computational efficiency.

In Chapter 4, we continue to address efficiency by tackling the significant bottleneck of communication cost in FL systems. To this end, we investigate two important research questions. First, we explore the potential of diffusion models for one-shot FL, presenting a thorough effort in this area. Our research reveals the unique benefits of DMs, demonstrating their ability to improve performance and address heterogeneity across diverse environments with our proposed approach,

FedDiff. Second, we delve into privacy concerns in SOTA one-shot FL, offering a comprehensive analysis within provable privacy budgets and addressing issues related to data memorization. To further enhance performance under stringent DP conditions, we introduce a novel and practical solution, Fourier Magnitude Filtering. This technique improves the effectiveness of generated data for global model training by filtering out low-quality samples.

In Chapter 5, we further address compute and label efficiency in the centralized domain. Specifically, we propose a novel approach to developing superior geospatial foundation models while significantly reducing resource expenditure compared to prior efforts. Initially, we curate a streamlined yet varied dataset from multiple remote sensing sources for pretraining purposes. Next, we introduce an unexpectedly simple yet potent multi-objective continual pretraining method. This method harnesses the robust representations of ImageNet-22k to expedite the learning process, while concurrently allowing for the acquisition of valuable in-domain features through self-supervised learning on geospatial data.

## 6.2 Future Work

An interesting direction for future work would be to investigate ways to leverage second-order information to improve the global aggregation process of the client models in federated learning. For instance, the weight assigned to each model weight could be determined by the overall smoothness and/or similarity to other client models being aggregated to promote advantageous model aggregation. Furthermore, while our study and proposed FedAlign in Chapter 3 focus on vision experiments, neither the insights nor the method are inherently constrained to the vision domain. Therefore, future work could further adapt and test the effectiveness in other domains such as language or audio processing in FL settings.

In Chapter 4, we conduct an in-depth analysis of the privacy considerations for diffusion models in FL and introduce our FMF method to enhance performance under differential privacy. Despite these advancements, performance in the most challenging scenarios remains suboptimal. Future research could focus on developing methods that continue to improve performance while adhering to stringent privacy constraints.

The GeoPile dataset presented in Chapter 6 contains a variety of resolutions, locations, and objects, making it effective for geospatial pretraining. However, in regard to image spectrums, the dataset is composed of RGB images. A worthwhile avenue for future research would be to expand the GeoPile to include data paired with additional spectral bands, and apply such data in the GFM pretraining task.

# LIST OF REFERENCES


[1] Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. CCS'16, ACM (Oct 2016). https://doi.org/10.1145/2976749.2978318, http://dx.doi.org/10.1145/2976749.2978318

[2] Abay, A., Zhou, Y., Baracaldo, N., Rajamoni, S., Chuba, E., Ludwig, H.: Mitigating bias in federated learning. CoRR **abs/2012.02447** (2020), https://arxiv.org/abs/2012.02447

[3] Acar, D.A.E., Zhao, Y., Matas, R., Mattina, M., Whatmough, P., Saligrama, V.: Federated learning based on dynamic regularization. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=B7v4QMR6Z9w

[4] Acar, D.A.E., Zhao, Y., Navarro, R.M., Mattina, M., Whatmough, P.N., Saligrama, V.: Federated learning based on dynamic regularization. CoRR **abs/2111.04263** (2021), https://arxiv.org/abs/2111.04263

[5] of Agriculture, U.D.: National agriculture imagery program (naip)

[6] Akiva, P., Purri, M., Leotta, M.J.: Self-supervised material and texture representation learning for remote sensing tasks. CoRR **abs/2112.01715** (2021), https://arxiv.org/abs/2112.01715

[7] Ayush, K., Uzkent, B., Meng, C., Tanmay, K., Burke, M., Lobell, D.B., Ermon, S.: Geography-aware self-supervised learning. CoRR **abs/2011.09980** (2020), https://arxiv.org/abs/2011.09980

[8] Bao, H., Dong, L., Wei, F.: Beit: BERT pre-training of image transformers. CoRR **abs/2106.08254** (2021), https://arxiv.org/abs/2106.08254

[9] Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramer, F., Balle, B., Ippolito, D., Wallace, E.: Extracting training data from diffusion models. In: 32nd USENIX Security Symposium (USENIX Security 23). pp. 5253–5270 (2023)

[10] Caye Daudt, R., Le Saux, B., Boulch, A., Gousseau, Y.: Urban change detection for multispectral earth observation using convolutional neural networks. In: IEEE International Geoscience and Remote Sensing Symposium (IGARSS) (July 2018)

[11] Caye Daudt, R., Le Saux, B., Boulch, A.: Fully convolutional siamese networks for change detection. In: 2018 25th IEEE International Conference on Image Processing (ICIP). pp. 4063–4067 (2018). https://doi.org/10.1109/ICIP.2018.8451652

[12] Chen, L., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. CoRR **abs/1706.05587** (2017), http://arxiv.org/abs/1706.05587

[13] Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., Sutskever, I.: Generative pretraining from pixels. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 1691–1703. PMLR (13–18 Jul 2020), https://proceedings.mlr.press/v119/chen20s.html

[14] Chen, X., Hsieh, C.J.: Stabilizing differentiable architecture search via perturbation-based regularization. In: International Conference on Machine Learning. pp. 1554–1565. PMLR (2020)

[15] Chen, X., Fan, H., Girshick, R.B., He, K.: Improved baselines with momentum contrastive learning. CoRR **abs/2003.04297** (2020), https://arxiv.org/abs/2003.04297

[16] Cheng, G., Han, J., Lu, X.: Remote sensing image scene classification: Benchmark and state of the art. Proceedings of the IEEE **105**(10), 1865–1883 (Oct 2017). https://doi.org/10.1109/jproc.2017.2675998, http://dx.doi.org/10.1109/JPROC.2017.2675998

[17] Cong, Y., Khanna, S., Meng, C., Liu, P., Rozi, E., He, Y., Burke, M., Lobell, D.B., Ermon, S.: Satmae: Pre-training transformers for temporal and multi-spectral satellite imagery. arXiv preprint arXiv:2207.08051 (2022)

[18] Contributors, M.: MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation (2020)

[19] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)

[20] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

[21] Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. CoRR **abs/2105.05233** (2021), https://arxiv.org/abs/2105.05233

[22] Dimitrovski, I., Kitanovski, I., Kocev, D., Simidjievski, N.: Current trends in deep learning for earth observation: An open-source benchmark arena for image classification (2022). https://doi.org/10.48550/ARXIV.2207.07189, https://arxiv.org/abs/2207.07189

[23] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. CoRR **abs/2010.11929** (2020), https://arxiv.org/abs/2010.11929

[24] Dwork, C.: A firm foundation for private data analysis. Communications of the ACM **54**(1), 86–95 (2011)

[25] Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3. pp. 265–284. Springer (2006)

[26] Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science **9**(3–4), 211–407 (2014)

[27] Ghiasi, G., Lin, T.Y., Le, Q.V.: Dropblock: A regularization method for convolutional networks. Advances in Neural Information Processing Systems **31**, 10727–10737 (2018)

[28] Grill, J., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.Á., Guo, Z.D., Azar, M.G., Piot, B., Kavukcuoglu, K., Munos, R., Valko, M.: Bootstrap your own latent: A new approach to self-supervised learning. CoRR **abs/2006.07733** (2020), https://arxiv.org/abs/2006.07733

[29] Guha, N., Talwalkar, A., Smith, V.: One-shot federated learning. arXiv preprint arXiv:1902.11175 (2019)

[30] Gururangan, S., Marasovic, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., Smith, N.A.: Don't stop pretraining: Adapt language models to domains and tasks. CoRR **abs/2004.10964** (2020), https://arxiv.org/abs/2004.10964

[31] Ha, T., Dang, T.K., Dang, T.T., Truong, T.A., Nguyen, M.T.: Differential privacy in deep learning: An overview. In: 2019 International Conference on Advanced Computing and Applications (ACOMP). pp. 97–102 (2019). https://doi.org/10.1109/ACOMP.2019.00022

[32] Han, R., Ren, X., Peng, N.: DEER: A data efficient language model for event temporal reasoning. CoRR **abs/2012.15283** (2020), https://arxiv.org/abs/2012.15283

[33] He, C., Li, S., So, J., Zhang, M., Wang, H., Wang, X., Vepakomma, P., Singh, A., Qiu, H., Shen, L., Zhao, P., Kang, Y., Liu, Y., Raskar, R., Yang, Q., Annavaram, M., Avestimehr, S.: Fedml: A research library and benchmark for federated machine learning. arXiv preprint arXiv:2007.13518 (2020)

[34] He, C., Li, S., So, J., Zhang, M., Wang, H., Wang, X., Vepakomma, P., Singh, A., Qiu, H., Shen, L., Zhao, P., Kang, Y., Liu, Y., Raskar, R., Yang, Q., Annavaram, M., Avestimehr, S.: Fedml: A research library and benchmark for federated machine learning. CoRR **abs/2007.13518** (2020), https://arxiv.org/abs/2007.13518

[35] He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.B.: Masked autoencoders are scalable vision learners. CoRR **abs/2111.06377** (2021), https://arxiv.org/abs/2111.06377

[36] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

[37] Heinbaugh, C.E., Luz-Ricca, E., Shao, H.: Data-free one-shot federated learning under very high statistical heterogeneity. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=_hb4vM3jspB

[38] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. CoRR **abs/2006.11239** (2020), https://arxiv.org/abs/2006.11239

[39] Ho, J., Salimans, T.: Classifier-free diffusion guidance (2022)

[40] Huang, G., Sun, Y., Liu, Z., Sedra, D., Weinberger, K.Q.: Deep networks with stochastic depth. In: European conference on computer vision. pp. 646–661. Springer (2016)

[41] Hutchinson, M.: A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. Communication in Statistics- Simulation and Computation **18**, 1059–1076 (01 1989). https://doi.org/10.1080/03610919008812866

[42] Jean, N., Wang, S., Samar, A., Azzari, G., Lobell, D.B., Ermon, S.: Tile2vec: Unsupervised representation learning for spatially distributed data. CoRR **abs/1805.02855** (2018), http://arxiv.org/abs/1805.02855

[43] Ji, S., Wei, S., Lu, M.: Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set. IEEE Transactions on Geoscience and Remote Sensing **57**(1), 574–586 (2019). https://doi.org/10.1109/TGRS.2018.2858817

[44] Jiang*, Y., Neyshabur*, B., Mobahi, H., Krishnan, D., Bengio, S.: Fantastic generalization measures and where to find them. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=SJgIPJBFvH

[45] Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K.A., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R.G.L., Rouayheb, S.E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P.B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konečný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Raykova, M., Qi, H., Ramage, D., Raskar, R.,

Song, D., Song, W., Stich, S.U., Sun, Z., Suresh, A.T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F.X., Yu, H., Zhao, S.: Advances and open problems in federated learning. CoRR **abs/1912.04977** (2019), http://arxiv.org/abs/1912.04977

[46] Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., et al.: Advances and open problems in federated learning. Foundations and Trends® in Machine Learning **14**(1–2), 1–210 (2021). https://doi.org/10.1561/2200000083, http://dx.doi.org/10.1561/2200000083

[47] Kalapos, A., Gyires-Tóth, B.: Self-supervised pretraining for 2d medical image segmentation. arXiv preprint arXiv:2209.00314 (2022)

[48] Kang, J., Fernandez-Beltran, R., Duan, P., Liu, S., Plaza, A.J.: Deep unsupervised embedding for remotely sensed images based on spatially augmented momentum contrast. IEEE Transactions on Geoscience and Remote Sensing **59**(3), 2598–2610 (2021). https://doi.org/10.1109/TGRS.2020.3007029

[49] Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A.T.: Scaffold: Stochastic controlled averaging for federated learning. In: International Conference on Machine Learning. pp. 5132–5143. PMLR (2020)

[50] Keskar, N.S., Mudigere, D., Nocedal, J., Smelyanskiy, M., Tang, P.T.P.: On large-batch training for deep learning: Generalization gap and sharp minima (2016)

[51] Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)

[52] Krizhevsky, A., Nair, V., Hinton, G.: Cifar-100 (canadian institute for advanced research) http://www.cs.toronto.edu/~kriz/cifar.html

[53] Lacoste, A., Luccioni, A., Schmidt, V., Dandres, T.: Quantifying the carbon emissions of machine learning. arXiv preprint arXiv:1910.09700 (2019)

[54] LeCun, Y.: The mnist database of handwritten digits. http://yann. lecun. com/exdb/mnist/

[55] Li, N., Lyu, M., Su, D., Yang, W.: Differential privacy: From theory to practice. Synthesis Lectures on Information Security, Privacy, & Trust **8**(4), 1–138 (2016)

[56] Li, Q., Diao, Y., Chen, Q., He, B.: Federated learning on non-iid data silos: An experimental study. arXiv preprint arXiv:2102.02079 (2021)

[57] Li, Q., He, B., Song, D.: Model-agnostic round-optimal federated learning via knowledge transfer. CoRR **abs/2010.01017** (2020), https://arxiv.org/abs/2010.01017

[58] Li, Q., He, B., Song, D.: Model-contrastive federated learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10713–10722 (2021)

[59] Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., Timofte, R.: Swinir: Image restoration using swin transformer. arXiv preprint arXiv:2108.10257 (2021)

[60] Liu, P., Xu, X., Wang, W.: Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives. Cybersecurity **5**(1), 1–19 (2022)

[61] Liu, Q., Chen, C., Qin, J., Dou, Q., Heng, P.A.: Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)

[62] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. CoRR **abs/2103.14030** (2021), https://arxiv.org/abs/2103.14030

[63] Liu, Z., Winata, G.I., Fung, P.: Continual mixed-language pre-training for extremely low-resource neural machine translation. CoRR **abs/2105.03953** (2021), https://arxiv.org/abs/2105.03953

[64] Long, Y., Gong, Y., Xiao, Z., Liu, Q.: Accurate object localization in remote sensing images based on convolutional neural networks. IEEE Transactions on Geoscience and Remote Sensing **55**(5), 2486–2498 (2017). https://doi.org/10.1109/TGRS.2016.2645610

[65] Luo, M., Chen, F., Hu, D., Zhang, Y., Liang, J., Feng, J.: No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. arXiv e-prints pp. arXiv–2106 (2021)

[66] Mañas, O., Lacoste, A., Giró-i-Nieto, X., Vázquez, D., Rodríguez, P.: Seasonal contrast: Unsupervised pre-training from uncurated remote sensing data. CoRR **abs/2103.16607** (2021), https://arxiv.org/abs/2103.16607

[67] McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282. PMLR (2017)

[68] Mendieta, M., Yang, T., Wang, P., Lee, M., Ding, Z., Chen, C.: Local learning matters: Rethinking data heterogeneity in federated learning. CoRR **abs/2111.14213** (2021), https://arxiv.org/abs/2111.14213

[69] Neumann, M., Pinto, A.S., Zhai, X., Houlsby, N.: In-domain representation learning for remote sensing. CoRR **abs/1911.06721** (2019), http://arxiv.org/abs/1911.06721

[70] Nogueira, K., Penatti, O.A.B., dos Santos, J.A.: Towards better exploiting convolutional neural networks for remote sensing scene classification. CoRR **abs/1602.01517** (2016), http://arxiv.org/abs/1602.01517

[71] Oh, S., Park, J., Jeong, E., Kim, H., Bennis, M., Kim, S.L.: Mix2fld: Downlink federated learning after uplink federated distillation with two-way mixup. IEEE Communications Letters **24**(10), 2211–2215 (2020). https://doi.org/10.1109/LCOMM.2020.3003693

[72] Parascandolo, G., Neitz, A., Orvieto, A., Gresele, L., Schölkopf, B.: Learning explanations that are hard to vary. arXiv preprint arXiv:2009.00329 (2020)

[73] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019)

[74] Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. CoRR **abs/1604.07379** (2016), `http://arxiv.org/abs/1604.07379`

[75] Qi, X., Zhu, P., Wang, Y., Zhang, L., Peng, J., Wu, M., Chen, J., Zhao, X., Zang, N., Mathiopoulos, P.T.: Mlrsnet: A multi-label high spatial resolution remote sensing dataset for semantic scene understanding. ISPRS Journal of Photogrammetry and Remote Sensing **169**, 337–350 (2020). https://doi.org/https://doi.org/10.1016/j.isprsjprs.2020.09.020, `https://www.sciencedirect.com/science/article/pii/S0924271620302677`

[76] Rame, A., Dancette, C., Cord, M.: Fishr: Invariant gradient variances for out-of-distribution generalization. arXiv preprint arXiv:2109.02934 (2021)

[77] Reed, C.J., Yue, X., Nrusimha, A., Ebrahimi, S., Vijaykumar, V., Mao, R., Li, B., Zhang, S., Guillory, D., Metzger, S., et al.: Self-supervised pretraining improves self-supervised pre-

training. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2584–2594 (2022)

[78] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. pp. 234–241. Springer (2015)

[79] Rottensteiner, F., Sohn, G., Jung, J., Gerke, M., Baillard, C., Benitez, S., Breitkopf, U.: The isprs benchmark on urban object classification and 3d building reconstruction. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences I-3 (2012), Nr. 1 **1**(1), 293–298 (2012)

[80] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y

[81] Sahu, A.K., Li, T., Sanjabi, M., Zaheer, M., Talwalkar, A., Smith, V.: On the convergence of federated optimization in heterogeneous networks. CoRR **abs/1812.06127** (2018), `http://arxiv.org/abs/1812.06127`

[82] Salehkaleybar, S., Sharifnassab, A., Golestani, S.J.: One-shot federated learning: theoretical limits and algorithms to achieve them. The Journal of Machine Learning Research **22**(1), 8485–8531 (2021)

[83] Shang, Y., Duan, B., Zong, Z., Nie, L., Yan, Y.: Lipschitz continuity guided knowledge distillation (2021)

[84] Shin, M., Hwang, C., Kim, J., Park, J., Bennis, M., Kim, S.L.: Xor mixup: Privacy-preserving data augmentation for one-shot federated learning. arXiv preprint arXiv:2006.05148 (2020)

[85] Sohn, K., Lee, H., Yan, X.: Learning structured output representation using deep conditional generative models. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 28. Curran Associates, Inc. (2015), https://proceedings.neurips.cc/paper_files/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf

[86] Sumbul, G., Charfuelan, M., Demir, B., Markl, V.: Bigearthnet: A large-scale benchmark archive for remote sensing image understanding. In: IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium. pp. 5901–5904. IEEE (2019)

[87] Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C.: Efficient object localization using convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 648–656 (2015)

[88] Vincenzi, S., Porrello, A., Buzzega, P., Cipriano, M., Fronte, P., Cuccu, R., Ippoliti, C., Conte, A., Calderara, S.: The color out of space: learning self-supervised representations for earth observation imagery. CoRR **abs/2006.12119** (2020), https://arxiv.org/abs/2006.12119

[89] Wang, D., Zhang, J., Du, B., Xia, G.S., Tao, D.: An empirical study of remote sensing pretraining. IEEE Transactions on Geoscience and Remote Sensing (2022)

[90] Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., Khazaeni, Y.: Federated learning with matched averaging. In: International Conference on Learning Representations (2019)

[91] Wang, J., Liu, Q., Liang, H., Joshi, G., Poor, H.V.: Tackling the objective inconsistency problem in heterogeneous federated optimization. Advances in neural information processing systems (2020)

[92] Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. CoRR **abs/1708.07747** (2017), http://arxiv.org/abs/1708.07747

[93] Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: European Conference on Computer Vision. Springer (2018)

[94] Xiao, Z., Kreis, K., Vahdat, A.: Tackling the generative learning trilemma with denoising diffusion gans. CoRR **abs/2112.07804** (2021), https://arxiv.org/abs/2112.07804

[95] Xie, Z., Geng, Z., Hu, J., Zhang, Z., Hu, H., Cao, Y.: Revealing the dark secrets of masked image modeling. arXiv preprint arXiv:2205.13543 (2022)

[96] Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., Hu, H.: Simmim: A simple framework for masked image modeling. CoRR **abs/2111.09886** (2021), https://arxiv.org/abs/2111.09886

[97] Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., Pfister, H., Ni, B.: Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. Scientific Data **10**(1), 41 (2023)

[98] Yang, T., Zhu, S., Chen, C.: Gradaug: A new regularization method for deep neural networks. Advances in Neural Information Processing Systems **33** (2020)

[99] Yang, Y., Newsam, S.: Bag-of-visual-words and spatial extensions for land-use classification. In: Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems. pp. 270–279 (2010)

[100] Yao, Z., Gholami, A., Keutzer, K., Mahoney, M.W.: Pyhessian: Neural networks through the lens of the hessian. In: 2020 IEEE International Conference on Big Data (Big Data). pp. 581–590. IEEE (2020)

[101] Yao, Z., Gholami, A., Lei, Q., Keutzer, K., Mahoney, M.W.: Hessian-based analysis of large batch training and robustness to adversaries. Advances in Neural Information Processing Systems **31**, 4949–4959 (2018)

[102] Yoon, T., Shin, S., Hwang, S.J., Yang, E.: Fedmix: Approximation of mixup under mean augmented federated learning. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=Ogga20D2HO-

[103] Yousefpour, A., Shilov, I., Sablayrolles, A., Testuggine, D., Prasad, K., Malek, M., Nguyen, J., Ghosh, S., Bharadwaj, A., Zhao, J., Cormode, G., Mironov, I.: Opacus: User-friendly differential privacy library in PyTorch. arXiv preprint arXiv:2109.12298 (2021)

[104] Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6023–6032 (2019)

[105] Yurochkin, M., Agarwal, M., Ghosh, S., Greenewald, K., Hoang, N., Khazaeni, Y.: Bayesian nonparametric federated learning of neural networks. In: International Conference on Machine Learning. pp. 7252–7261. PMLR (2019)

[106] Zela, A., Elsken, T., Saikia, T., Marrakchi, Y., Brox, T., Hutter, F.: Understanding and robustifying differentiable architecture search. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=H1gDNyrKDS

[107] Zhang, C., Yue, P., Tapete, D., Jiang, L., Shangguan, B., Huang, L., Liu, G.: A deeply supervised image fusion network for change detection in high resolution bi-temporal remote sensing images. ISPRS Journal of Photogrammetry and Remote Sensing **166**, 183–200 (2020). https://doi.org/https://doi.org/10.1016/j.isprsjprs.2020.06.003, https://www.sciencedirect.com/science/article/pii/S0924271620301532

[108] Zhang, H., Cissé, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. CoRR **abs/1710.09412** (2017), http://arxiv.org/abs/1710.09412

[109] Zhang, J., Chen, C., Li, B., Lyu, L., Wu, S., Ding, S., Shen, C., Wu, C.: Dense: Data-free one-shot federated learning. Advances in Neural Information Processing Systems **35**, 21414–21428 (2022)

[110] Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A.L., Kong, T.: ibot: Image BERT pre-training with online tokenizer. CoRR **abs/2111.07832** (2021), https://arxiv.org/abs/2111.07832

[111] Zhou, W., Newsam, S., Li, C., Shao, Z.: Patternnet: A benchmark dataset for performance evaluation of remote sensing image retrieval. ISPRS Journal of Photogrammetry and Remote Sensing **145**, 197–209 (2018). https://doi.org/https://doi.org/10.1016/j.isprsjprs.2018.01.004, https://www.sciencedirect.com/science/article/pii/S0924271618300042, deep Learning RS Data

[112] Zhou, Y., Pu, G., Ma, X., Li, X., Wu, D.: Distilled one-shot federated learning. arXiv preprint arXiv:2009.07999 (2020)