A JOURNEY DOWN THE FEDERATED VALLEYS

A THESIS

submitted by

SOMYA TYAGI

for the award of the degree

of

MASTER OF TECHNOLOGY

(Research)

Computer Science and Engineering

INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY DELHI

NEW DELHI- 110020

June 2025

THESIS CERTIFICATE

This is to certify that the thesis titled **A JOURNEY DOWN THE FEDERATED VALLEYS**, submitted by **SOMYA TYAGI**, to the Indraprastha Institute of Information Technology Delhi, for the award of the degree of **Master of Technology (Research)**, is a bona fide record of the research work done by her under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute for the award of any degree or diploma.

Dr. Bapi Chatterjee
Thesis Supervisor
Assistant Professor
Dept. of Computer Science and Engineering
IIIT Delhi, 110020

Place: New Delhi

ACKNOWLEDGEMENTS

I am deeply grateful to Dr. Bapi Chatterjee for his invaluable guidance and mentorship throughout my research journey. His insightful advice and constant encouragement have been fundamental in shaping this thesis. Working under his supervision has been a privilege that has greatly enriched both my academic and professional development.

I would also like to thank all the members of the Distributed Computing and Learning Lab (DCLL) for their continuous support and stimulating discussions. Their collaborative environment fostered creativity and perseverance, making this journey both rewarding and enjoyable. The exchange of ideas and collective problem-solving within the lab greatly contributed to the success of my research.

I extend my sincere appreciation to all the faculty who provided assistance during my studies. Their dedication and professionalism created an environment conducive to learning and innovation.

Most importantly, I extend my heartfelt gratitude to my family for their unwavering love, patience, and confidence in me. Their constant support has been my pillar of strength during the most challenging times.

(SOMYA TYAGI)
MT23005

ABSTRACT

KEYWORDS: Federated Learning ; Optimization ; Min-Max Problem ; Instrumental Variable ; Generalized Method of Moments

Federated learning (FL) enables collaborative model training across distributed clients while preserving data privacy. However, the choice of optimizer on both the client and server sides significantly impacts training efficiency and model performance, especially under non-IID data distributions. Despite the existence of numerous optimizers, the absence of strong, consistent empirical evidence specific to federated environments makes it challenging to identify the most effective optimizer. Consequently, practitioners often rely on intuition and prior experience when choosing optimizers. This study provides comprehensive insights and practical guidelines for optimizer selection in federated learning frameworks.

Beyond standard empirical risk minimization, min-max optimization is a fundamental framework in machine learning to model adversarial and robust problems, and its utility extends beyond traditional ML applications into econometrics and causal inference. One notable application is the Generalized Method of Moments (GMM), a widely used technique for causal effect estimation via Instrumental Variables (IV) analysis, which finds practical applications in important areas such as healthcare and consumer economics. For IV analysis in high-dimensional settings, the Generalized Method of Moments (GMM) using deep neural networks offers an efficient approach. If the data is sourced from scattered, decentralized clients, federated learning readily fits for training the models while promising data privacy. However, to our knowledge, no federated algorithm for either GMM or IV analysis exists to date. This study also includes a method for federated instrumental variables analysis (FedIV) via the federated deep generalized method of moments (FedDeepGMM) for non-iid data. We characterize an equilibrium of a federated zero-sum game to show that it consistently estimates the local moment conditions of every participating client. The proposed algorithm is backed by extensive experiments to demonstrate the efficacy of our approach.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

FL	Federated Learning
FEDIV	Federated instrumental variables analysis
DEEPGMM	Deep Generalized Method of Moments
FEDDEEPGMM	Federtaed Deep Generalized Method of Moments
FEDGDA	Federated Gradient Descent-Ascent
NAG	Nesterov Accelerated Gradient
FEDAVG	Federated Averaging
SGD	Stochastic Gradient Descent
NSD	Negative Semi-definite
PSD	Positive Semi-definite

CHAPTER 1

Introduction

Federated Learning (FL) has emerged as a promising machine learning paradigm that enables collaborative model training across decentralized clients while preserving data privacy. It allows each client to keep their data local while participating in the global learning process by periodically sharing model updates with the central server. Despite its compelling advantages, FL introduces unique challenges, mainly due to data heterogeneity between clients and limited communication bandwidth.

This thesis explores two major directions in federated optimization. The first direction focuses on evaluating the impact of client-side and server-side optimizer choices on convergence and generalization performance under non-IID settings. The second investigates the application of Generalized Method of Moments (GMM), a widely used technique for causal effect estimation via Instrumental Variables (IV) analysis in federated setting, FEDDEEPGMM, which enables causal estimation in decentralized environments without compromising privacy.

1.1 Descending through a Federated Valley

In the first part of this work, we examine the interplay between client-side and server-side optimizers in federated settings characterized by non-IID data. While FEDAVG remains a widely used baseline, its performance degrades in the presence of heterogeneity, motivating the need to investigate alternative optimization strategies.

Motivated by this, we designed various FL optimization strategies described in 2.4.

Experiments are performed on four benchmark datasets which are CIFAR-10, CIFAR-100, FE-MNIST and SHAKESPEARE representing both vision and language tasks. We simulate realistic non-IID conditions using a Dirichlet based data partitioning strategy and assess performance in terms of convergence behavior and test accuracy.

Results demonstrate that optimizer selection plays a critical role in federated learning. Notably, adaptive and parameter-free optimizers outperform in most configurations, particularly when combined with extrapolated server updates. These findings offer practical insights for selecting optimizer combinations tailored to federated learning scenarios.

1.2 Federated Instrumental Variable Estimation via FedDeepGMM

The second part of the thesis introduces a novel approach of Federated Instrumental Variable (IV) analysis via Federated Generalized Method of Moments (FEDGMM), a method which can be used in econometrics and causal inference to estimate treatment effects in the presence of unobserved confounding factors.

This method is formulated as a federated minimax optimization problem, where each client contributes local moment conditions, and the server aggregates them to learn a shared causal model. We implement this using a Federated Gradient Descent-Ascent (FEDGDA) algorithm.

We establish the empirical validation for our method. FEDDEEPGMM consistently estimates causal effects across various datasets, validating its applicability in real-world scenarios such as healthcare and economics.

1.3 Contribution of the Thesis

Our main contributions through this thesis are as follows :

1. Design and evaluation of novel optimization methods in federated learning.
2. Proposal of FEDDEEPGMM, the first federated algorithm for instrumental variable analysis using DEEPGMM, preserving data privacy.
3. Empirical validation showing the efficacy of FEDDEEPGMM under non-IID and heterogeneous client settings.

CHAPTER 2

Descending through a Federated Valley

2.1 Introduction

Federated Learning (FL) is now an accepted paradigm for training machine learning models on distributed system of nodes McMahan *et al.* (2017) maintaining data-locality. Formally, for a model $\theta \in \mathbb{R}^d$ and the empirical risk functions $f_i(\theta)$ on the nodes $i \in [N]$, FL refers to solving

$$\min_{\theta \in \mathbb{R}^d} \left\{ f(\theta) := \frac{1}{N} \sum_{i=1}^{N} f_i(\theta) \right\}, \tag{2.1}$$

Commonly, a node designated as a *server* orchestrates the nodes $i \in [N]$, also called *clients*, to collaboratively solve (2.1). This data-decentralized setting for FL is also known as the *horizontal* FL.

On a data-centralized setting, arguably the most popular approach for the empirical risk minimization (ERM) problem: $\min_{\theta \in \mathbb{R}^d} f(\theta)$, is the iterative stochastic gradient descent (SGD) Robbins and Monro (1951) algorithm: $\theta_{t+1} = \theta_t - \alpha_t \tilde{\nabla} f(\theta_t)$, for $t \in [T]$, where $\theta_t \in \mathbb{R}^d$ is the state of the model, α_t is a dampening factor, often called the *learning rate* or *step-size*, and $\tilde{\nabla} f(\theta_t)$ is an estimator of the gradient $\nabla f(\theta_t)$ after t iterations. The convex optimization literature has been enriched by the numerous (more than a hundred and counting) variants of SGD Schmidt *et al.* (2021), which unceasingly grow with today's accelerated growth in deep learning. A framework to represent the variants of SGD can be given as

$$\theta_{t+1} = Optimizer(\theta_t, \tilde{\nabla} f\left(\theta_t\right), \alpha_t, \{\mathbf{B}_t\}), \tag{2.2}$$

where $\{\mathbf{B}_t\}$ is a set of hyper-parameters, in addition to α_t after iteration t.

ERM powers today's large-scale non-convex deep model training tasks. The right set of hyperparameters, in addition to the selection of the $Optimizer$ method, influences not only the computational cost of the training process but can qualitatively alter

the solutions in terms of an entirely different local minima achieved at the end of the training process Pascanu *et al.* (2025).

Federated Learning (FL), introduced by McMahan *et al.* (2017), is a learning paradigm that enables training machine learning models, such as deep neural networks, across multiple decentralized datasets located on client devices. FL is based on the principle that data remain on each client, and only model updates are transmitted to a central server. The foundational algorithm in Federated Learning is Federated Averaging (FEDAVG). It operates by performing k local steps of Stochastic Gradient Descent (SGD) on a randomly selected subset of client devices. Periodically, the locally updated models are communicated to a central server, where they are aggregated, typically by averaging, to update the global model.

Synchronization Challenges in Federated Learning FL primarily depends on achieving synchronization between the learning process of individual clients and the trajectory of the global model. As data is distributed across multiple clients, the issue of data heterogeneity arises. To address heterogeneity issues, FEDPROX Li *et al.* (2020) incorporates a regularization term, $\frac{\mu}{2}\|\theta^i - \theta\|^2$, into the local objective functions. Here θ^i and θ represent the client and server models, respectively. This modification makes client-side optimization proximal to the global model. FEDDYN Durmus *et al.* (2021) proposes an additional regularization term for clients' objectives similar to FEDPROX. In contrast, SCAFFOLD Karimireddy *et al.* (2020) introduces control variates at both the server and client sides to correct for *client drift*, thereby improving convergence under heterogeneous data distributions. Heterogeneity in clients local data distributions and computational capabilities leads to significant variation in the number of local updates performed by each client during a given communication round. To tackle this, FEDNOVA Wang *et al.* (2020) is introduced. The core idea of FedNova is to avoid directly averaging the cumulative local updates, $\theta^{(i)}_{t,k_i} - \theta_{t,0}$, where θ_t denotes the server's model after t synchronization rounds, also called the *global model*. With θ_t communicated to clients, $\theta^i_{t,k}$ is the model state at client $i \in [N]$ after k local gradient updates. Instead, the server aggregates the normalized local updates, given by $\theta^{(i)}_{t,k_i} - \theta_{t,0}/k_i$. This normalization ensures objective consistency across clients while maintaining efficient convergence. While these methods improve synchronization and robustness in the presence of heterogeneity, they often retain the standard averaging-based global update and

keep the server's learning rate η_g constant; often $\eta_g = 1$. Surely, it leaves a scope to tune the hyperparameters, very importantly, η_g.

Problem Formulation To formalize the federated learning objective, we consider the following minimization problem:

where $F(\theta)$ is the global objective function, aggregating local losses $f_i(\theta)$ from N clients. A widely adopted method for solving this optimization problem is Federated Averaging (FEDAVG) McMahan *et al.* (2017). In FEDAVG, clients perform several local SGD updates and periodically synchronize with a central server. The process can be described as follows:

$$\theta_{t,k}^{(i)} = \theta_{t,k-1}^{(i)} - \eta_{t,k}^{(i)} g(\theta_{t,k-1}^{(i)}) \tag{2.3}$$

$$\text{for } k \in [K], \text{ with } \theta_{t,k}^{(0)} = \theta_t$$

$$\Delta_t = \frac{1}{|P_t|} \sum_{i=1}^{|P_t|} \left\{ \Delta_t^i := \theta_t - \theta_{t,k}^i \right\}, \quad \theta_{t+1} = \theta_t + \eta_{g_t} \Delta_t \tag{2.4}$$

where θ_t denotes the server's model after t synchronization rounds, also called the *global model*. With θ_t communicated to clients, $\theta_{t,k}^i$ is the model state at client $i \in [N]$ after k local gradient updates. $P_t \subseteq [N]$ is a subset of participating clients after t rounds. $\Delta_t^i := \theta_t - \theta_{t,K}^i$ denotes the model update at client i due to K local gradient update steps, whereby, Δ_t represents the synchronized update to be applied to the model after t rounds; η_{g_t} is the learning rate at the server.

Data across clients can be heterogeneous (non-IID), which poses significant challenges for learning the global model. The standard federated learning protocol proceeds by having a subset of clients perform local optimization steps on their respective f_i's, followed by aggregation of their model updates at a central server. This decentralized setup gives rise to both statistical and system-level heterogeneity, which can hinder convergence and model performance if not properly addressed. To address these issues, modern federated learning approaches employ optimization not only on the client side, where local model updates are computed, but also on the server side, where these

updates are aggregated and used to refine the global model. The choice of optimization algorithms at both ends plays a critical role in balancing convergence of the global model, stability, and communication efficiency.

In this study, we conduct a series of experiments to assess the effectiveness of various optimization algorithms within a federated learning framework. Specifically, we investigate the impact of different combinations of client-side and server-side optimizers on overall performance. Given the decentralized nature of FL and the challenges posed by non-IID data and system heterogeneity, understanding how optimizer choices affect convergence and stability is crucial.

2.2 Related Work

Previous studies have shown that FedAvg introduced in McMahan *et al.* (2017) performs competitively with centralized machine learning training when data is distributed independently and identically (i.i.d.) among clients. However, its performance degrades in the presence of non-i.i.d. data, which reflects real-world scenarios. Despite this, FEDAVG remains one of the most widely used baseline algorithms due to its simplicity and efficiency.

2.2.1 Aggregation of Client Models

Building upon the foundational FEDAVG algorithm, several studies have explored alternative aggregation strategies that aim to improve the robustness and efficiency of model updates, particularly under non-IID data distributions. These aggregation methods, commonly employed in traditional multi-round federated learning settings, operate without relying on public datasets or requiring additional training procedures. Another direction in aggregation research focuses on the structural alignment of neural network parameters across client models. Yurochkin *et al.* (2019) identify the permutation invariance of neurons in multilayer perceptrons and propose PFNM, a Bayesian nonparametric framework for federated learning. Notably, this method constructs a global model by aligning neurons across local models without requiring server-side training or access to centralized data. Another direction is about aggregating the local updates

without server-side computation, Su *et al.* (2023) discusses one-shot federated learning as a method to reduce communication costs between clients and the server. A recent theoretical study Malinovsky *et al.* (2022) has shown that scaling client updates and tuning server-side stepsizes can significantly improve convergence in Federated Averaging (FEDAVG), especially when using Random Reshuffle during local training. Specifically, for convex, strongly convex and nonconvex problems this paper demonstrate that when local step sizes are small and the update direction is determined using FEDAVG combined with Random Reshuffling across all clients, it is possible to take a larger step along this direction, leading to improved convergence rates.

2.2.2 Hyperparameter tuning in Federated Learning

Recently, FEDEX Khodak *et al.* (2021) addressed the challenge of hyperparameter tuning in federated learning by leveraging weight-sharing techniques from neural architecture search. It efficiently tunes parameters such as learning rates and shows improved performance. This approach extends the vanilla FEDAVG algorithm by treating client updates as pseudo-gradients and applying a server step size, effectively transforming the aggregation into a generalized gradient descent step. Recent work Wu *et al.* (2025) models federated learning as a multi-criterion problem and highlights personalization as the key in high-dimensional settings. It shows that simple methods such as Finetuned FEDAVG (FTFA) and Ridge-tuned FedAvg (RTFA) can match or outperform more complex approaches while being more efficient.

HAFed Jiang and Tang (2023) is a personalized federated learning algorithm that dynamically adjusts hyperparameters during training, addressing the limitations of using static values. By applying a log-based function to adapt hyperparameters each round based on the gap between client models, HAFed improves performance across updates.

2.2.3 The existing FL methods

This section presents an overview of widely adopted federated learning (FL) algorithms along with their respective citation counts, as illustrated in Figure 2.1. Among the earliest and most foundational approaches, FEDAVG (McMahan *et al.*, 2017) remains the most cited, introducing a simple yet effective strategy of averaging local model updates

to construct the global model. Building on this, FEDPROX Li *et al.* (2020) incorporates a proximal term to better handle heterogeneity in client data and computation capabilities, making it more robust in real-world FL scenarios.

To further mitigate the effects of client drift in non-IID settings, SCAFFOLD Karimireddy *et al.* (2020) introduced control variates at server and clients. FEDPER Arivazhagan *et al.* (2019), a federated learning approach that separates deep learning models into a shared base and personalized layers. The base layers are trained via federated averaging (FEDAVG), while the personalized layers are updated using local data with SGD, helping mitigate the effects of statistical heterogeneity. Robust aggregation techniques, such as RFA Pillutla *et al.* (2022), use geometric median or trimmed mean instead of simple averaging, enhancing resilience to malicious or noisy updates.

FEDNOVA Wang *et al.* (2020) addresses disparities in client participation by normalizing updates based on the number of local steps, ensuring fairer aggregation. Meanwhile, FEDOPT Reddi *et al.* (2021) generalizes the server-side update rule by incorporating adaptive optimization algorithms like Adam and Yogi.

MIME Karimireddy *et al.* (2021) is a federated learning framework designed to mitigate client drift and adapt centralized optimization algorithms like momentum and Adam to the cross-device setting. By combining control variates and server-level optimizer states, MIME ensures local updates mimic those of centralized methods, demonstrating provably faster convergence when combined with momentum-based variance reduction.

Privacy-preserving FL is represented by DP-FEDAVG Noble *et al.* (2023), which adds differential privacy noise to client updates, thereby protecting sensitive data. MOON Li *et al.* (2021), is a federated learning framework that improves local training by leveraging the similarity between model representations through contrastive learning at the model level. It effectively addresses data heterogeneity and outperforms existing methods on image classification tasks. Finally, FEDDYN Durmus *et al.* (2021) introduces dynamic regularization to each client's objective, effectively mitigating divergence caused by non-IID data distributions.

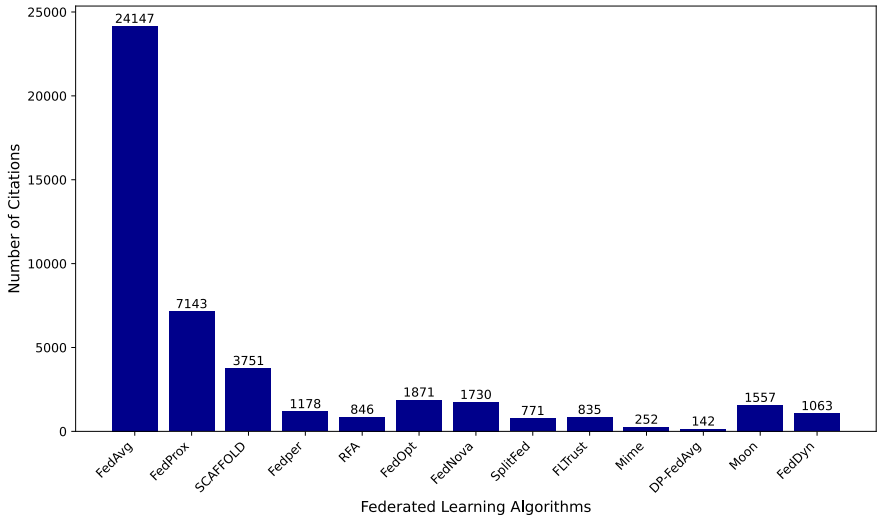


Figure 2.1: Citations of Federated Learning Algorithms

2.3 Methods

In recent years, there has been ongoing discussion about which methods perform best in federated learning settings and which optimizers are most suitable for local updates on the client side. However, a comprehensive comparison that evaluates these optimizers across both server and client roles under consistent experimental conditions has been largely missing. Therefore, our objective is to design and evaluate methods that can be used effectively in federated settings. In our study, we selected 11 optimizers, classified into three distinct groups, to perform global or local model updates. Table 2.2 lists the methods evaluated in our experiments. This comparison aims to provide practitioners and researchers with clearer guidance on selecting optimization strategies suited to various federated learning scenarios, especially when dealing with non-IID data or communication constraints and to better understand the impact of optimization algorithms and hyperparameter settings on training performance. We specifically investigate whether recently proposed optimizers offer improved performance for client-side and global-side model updates compared to well-established methods such as SGD and Adam.

Non-Adaptive Methods Non-adaptive optimization methods use a fixed learning rate that remains constant throughout all iterations. These methods are simple and computationally efficient, making them popular choices in a federated learning setting, especially where we have limited memory and communication constraints. This class

includes Stochastic Gradient descent(SGD), SGD with Momentum and Nesterov Accelerated Gradient (NAG). These optimizers perform well with minimal hyperparameter tuning, simple hand-tuning is often sufficient to achieve good results.

Adaptive Methods Adaptive optimization methods can be broadly classified into two main categories. The first category consists of optimizers that incorporate adaptive momentum mechanisms, which dynamically adjust the momentum term for faster convergence. The second category includes optimizers that focus on adaptive learning rates, where the step size is automatically modified based on historical gradients to improve optimization efficiency and robustness. Both approaches aim to improve the training process by adapting critical hyperparameters in response to the learning dynamics, thereby achieving faster convergence. The first category of optimizers is built upon ADAM as the underlying base optimizer, and the update rule is generally expressed as follows. Let g_t denote the gradient at time step t. The general form of adaptive optimizers is given by the following update rule:

$$
\begin{aligned}
m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t \\
v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \\
\hat{m}_t &= \frac{m_t}{1-\beta_1^t} \\
\hat{v}_t &= \frac{v_t}{1-\beta_2^t} \\
\theta_{t+1} &= \theta_t - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}
\end{aligned}
$$

where β_1 and β_2 are the decay rates for the first and second moments, ϵ is a small constant for numerical stability, and η is the learning rate. For the second category of optimizers, which are based on adaptive learning rates, ADAGRAD serves as the base optimizer. ADAGRAD modifies the general learning rate η at each time step t for every parameter θ_t, based on the accumulated historical gradients computed for θ_t. A major drawback of ADAGRAD is the continual accumulation of squared gradients in the denominator. Since each added term is positive, the accumulated sum grows over time, leading to a progressive reduction in the effective learning rate. Eventually, the learning rate can become so small that the algorithm effectively stops learning. To address this issue of ADAGRAD, ADADELTA, and RMSPROP are introduced which are somehow

similar.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t} + \epsilon} \odot g_t$$

ADADELTA extends ADAGRAD by addressing its overly aggressive, monotonically decreasing learning rate. Rather than accumulating all past squared gradients, it uses a decaying average over a fixed window. The running average $E[g^2]_t$ at time step t is updated recursively using a decay factor γ, similar to the momentum term, and incorporates only the previous average and the current gradient.

Parameter-Free Optimizers These optimizers perform well without extensive tuning, relying mainly on good initial parameter settings. Examples of optimizers are DOWG (Distance over Weighted Gradients), DOG (Distance over Gradients) and PRODIGY. Update rule for these optimizers is given in table 2.1

Optimizer	Update Rule	Parameter Initializations
SGD	$\theta_{t+1} = \theta_t - \eta\, g(\theta_t)$	$\eta = 0.001, \quad \theta \in \mathbb{R}_d$
ADAM	$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t,$ $v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2,$ $\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1-\beta_2^t}$ $\theta_{t+1} = \theta_t - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$	$m_0 = 0, \quad v_0 = 0,$ $\beta_1 = 0.9, \quad \beta_2 = 0.999,$ $\epsilon = 0.1, \quad \eta = 0.01,$ $\theta \in \mathbb{R}_d$
NADAM	$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \left(\beta_1 \hat{m}_t + \frac{(1-\beta_1) g_t}{1-\beta_1^t} \right)$	$m_0 = 0, \quad v_0 = 0,$ $\beta_1 = 0.9, \quad \beta_2 = 0.999,$ $\epsilon = 0.1, \quad \eta = 0.01,$ $\theta \in \mathbb{R}_d$
RADAM	$v_t = \frac{1}{\beta_2} v_{t-1} + (1-\beta_2) g_t^2,$ $m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t,$ $\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \quad \rho_t = \rho_\infty - \frac{2t\beta_2^t}{1-\beta_2^t},$ $\rho_\infty = \frac{2}{1-\beta_2} - 1 l_t = \frac{\sqrt{1-\beta_2^t}}{\sqrt{v_t} + \epsilon},$ $r_t = \sqrt{\frac{(\rho_t - 4)(\rho_t - 2)\rho_\infty}{(\rho_\infty - 4)(\rho_\infty - 2)\rho_t}}$ $\theta_t = \begin{cases} \theta_t - \gamma\, \hat{m}_t r_t l_t, & \text{if } \rho_t > 5, \\ \theta_t - \gamma\, \hat{m}_t, & \text{otherwise} \end{cases}$	$m_0 = 0, \quad v_0 = 0,$ $\beta_1 = 0.9, \quad \beta_2 = 0.999,$ $\epsilon = 0.1, \quad \eta = 0.01,$ $\theta \in \mathbb{R}_d$
AMSGRAD	$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t,$ $v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2,$ $\hat{v}_t = \max(\hat{v}_{t-1}, v_t)$ $\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} m_t$	$m_0 = 0, v_0 = 0, \hat{v}_0 = 0$ $\beta_1 = 0.9, \quad \beta_2 = 0.999,$ $\epsilon = 0.1, \quad \eta = 0.01,$ $\theta \in \mathbb{R}_d$
ADAMAX	$u_t = \beta_2^\infty v_{t-1} + (1-\beta_2^\infty) \lvert g_t \rvert^\infty$ $= \max(\beta_2 \cdot v_{t-1}, \lvert g_t \rvert)$ $\theta_{t+1} = \theta_t - \frac{\eta}{u_t} \hat{m}_t$	$v_0 = 0, u_0 = 0, \hat{m}_0 = 0$ $\beta_2 = 0.999, \eta = 0.01,$ $\theta \in \mathbb{R}_d$

Optimizer	Update Rule	Initializations
ADAGRAD	$g_{t,i} = \nabla_{\theta_t} f(\theta_t, i)$ $\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \cdot g_{t,i}$	$G_0 = 0, \eta = 0.001$ $\epsilon = 0.01, \theta \in \mathbb{R}_d$
ADADELTA	$\mathbb{E}[g^2]_t = \gamma \mathbb{E}[g^2]_{t-1} + (1-\gamma) g_t^2$ $\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\mathbb{E}[g^2]_t + \epsilon}} g_t$	$\mathbb{E}[g^2]_0 = 0, \eta = 0.01$ $\epsilon = 0.01, \gamma = 0.7,$ $\theta \in \mathbb{R}_d$
RMSPROP	$\mathbb{E}[g^2]_t = 0.9\, \mathbb{E}[g^2]_{t-1} + 0.1\, g_t^2,$ $\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\mathbb{E}[g^2]_t + \epsilon}} g_t$	$\mathbb{E}[g^2]_0 = 0, \eta = 0.01$ $\epsilon = 0.01, \theta \in \mathbb{R}_d$
DOWG	$\bar{r}_t = \max\left(\|x_t - x_0\|, \bar{r}_{t-1}\right),$ $v_t = v_{t-1} + \bar{r}_t^2 \, \|g(\theta_t)\|^2$ $\eta_t = \frac{\bar{r}_t^2}{\sqrt{v_t}} \quad , \theta_{t+1} = \Pi_{\mathcal{X}}\left(\theta_t - \eta_t g(\theta_t)\right)$	$\bar{r}_0, v_0 = 0$ $x_0 = 0, \theta \in \mathbb{R}_d$
DOG	$\eta_t = \frac{\max_{i \le t} \|x_i - x_0\|}{\sqrt{\sum_{i \le t} \|g_i\|^2}},$ $\theta_{t+1} = \theta_t - \eta_t \, g(\theta_t)$	$x_0 = 0, \theta \in \mathbb{R}_d$
PRODIGY	$\eta_t = \frac{d_t^2 \lambda_t}{\sqrt{d_t^2 G^2 + \sum_{i=0}^t d_i^2 \lambda_i^2 \|g_i\|^2}},$ $\theta_{t+1} = \theta_t - \eta_t g_t$ $\hat{d}_{t+1} = \frac{\sum_{i=0}^t \eta_i \langle g_i, x_0 - x_i \rangle}{\|x_{t+1} - x_0\|},$ $d_{t+1} = \max(d_t, \hat{d}_{t+1})$	$d_0 = 0.0001,$ $\lambda_i = 0.01, x_0 = 0,$ $\theta \in \mathbb{R}_d$

Table 2.1: Update rules and initializations of the optimization algorithms.

2.4 Federated Optimization

To evaluate the impact of different client-side and server-side optimization algorithms in a federated learning setting, we design a comprehensive set of experiments. Our primary goal is to analyze how various optimizer combinations on client-side and server-side influence training dynamics, model convergence, and final performance across non-IID data distributions. We conducted these experiments on multiple datasets to ensure generalizability of the observed patterns.

Specifically, we structured our evaluation around four distinct configurations to analyze the interaction between local (client-side) and global (server-side) updates:

1. **Symmetric Optimization:** Identical optimizers used on both the client side and the server side.
2. **Asymmetric Optimization I:** Stochastic Gradient Descent (SGD) applied on the client side with varying optimizers on the server side.
3. **Asymmetric Optimization II:** Different optimizers applied on the client side, while using FEDAVG as the server-side aggregation method.
4. **Asymmetric Optimization III:** Different client-side optimizers are applied while using Extrapolation with FEDAVG as the server-side aggregation method.

2.5 Experiments

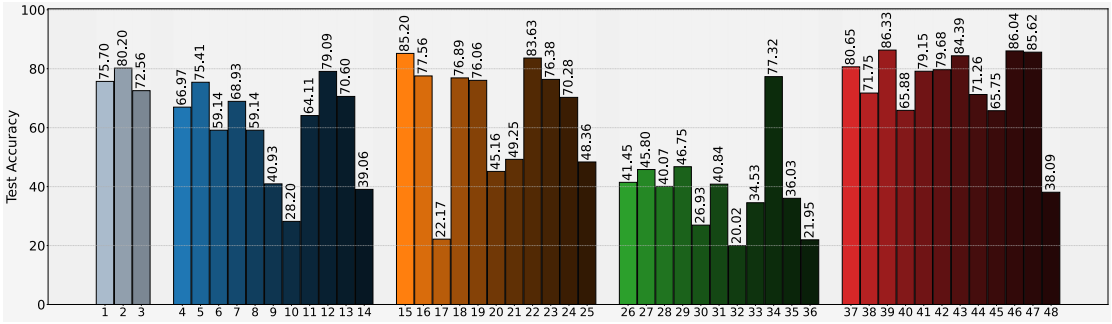
Datasets and Architecture: We evaluated the proposed algorithms on a diverse set of benchmarks that cover image classification and text prediction tasks. Our experiments involved four combination of datasets Caldas *et al.* (2018) and models: (a) **CIFAR-10** with ResNet-18, (b) **CIFAR-100** with ResNet-18, (c) **FEMNIST** with Convolutional Neural Network (CNN), and (d) **SHAKESPEARE** with Long Short-Term Memory (LSTM).

Experimental Setup For training across different algorithms, we distributed over 100 clients as in Jhunjhunwala *et al.* (2023). The number of clients for **FEMNIST** and

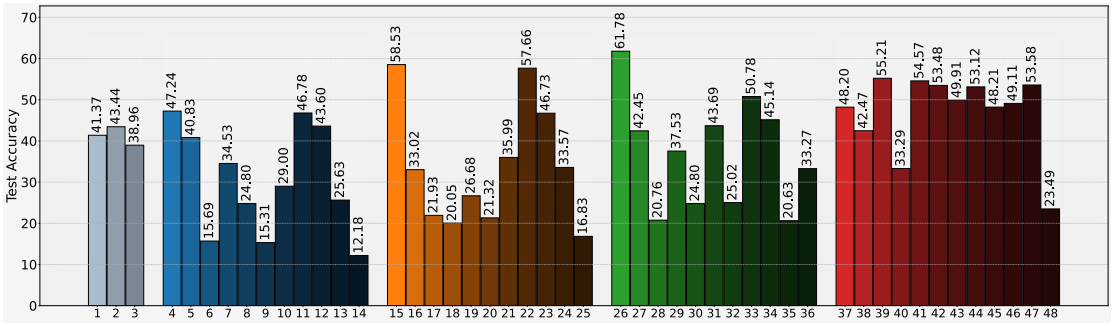
SHAKESPEARE are selected as in Caldas *et al.* (2018). In each training round, we uniformly sample 20 clients without replacement within a round, but with replacement across rounds. We compute mini-batch gradients on each client using a fixed batch size of 50. The number of local epochs is fixed at $K = 20$ for all experiments. To introduce heterogeneity in the data distribution across clients, we employ a Dirichlet distribution with a concentration parameter $\alpha = 0.3$ Caldas *et al.* (2018). All experiments were performed on NVIDIA A6000 GPUs with 48 GB onboard memory. Wherever required, we performed grid search for hyperparameter tuning.

2.5.1 Tuning

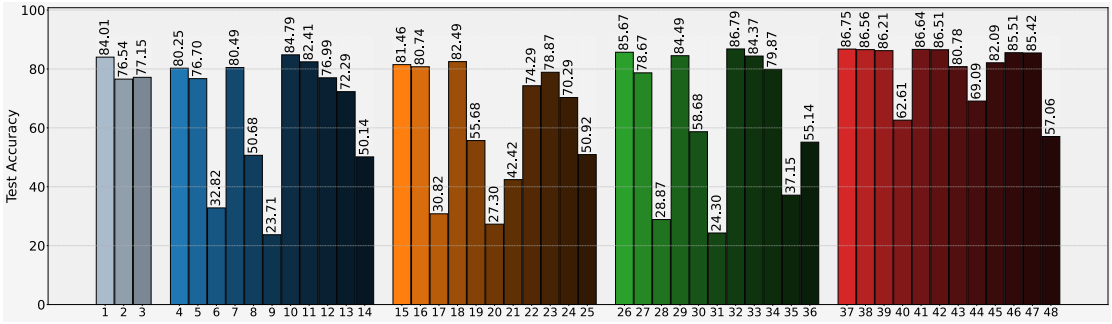
Hyperparameters always play a crucial role in the training performance of machine learning models. The learning rate (lr) is the most fundamental hyperparameter that must be tuned for every optimizer, and we have tuned it for our algorithms.Choi *et al.* (2020) demonstrates that certain optimizers can closely approximate the behavior of others by properly tuning their hyperparameters, and the epsilon parameter ϵ in adaptive methods is an important factor and is reinterpreted as a hyperparameter. We tuned the parameter ϵ in all adaptive methods for the update of client-side models, which improved both the training performance and the test accuracy. Careful tuning of these hyperparameters is essential to maximize the effectiveness of the optimization algorithms and achieve superior model performance.



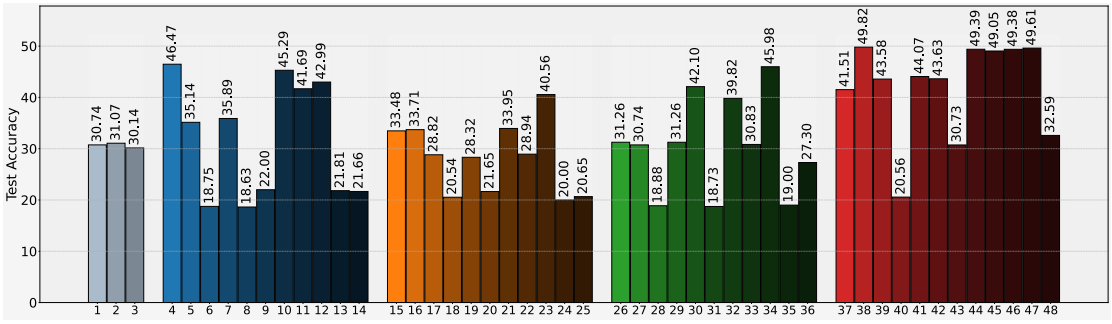
(a) CIFAR-10



(b) CIFAR-100



(c) FEMNIST



(d) SHAKESPEARE

Figure 2.2: Test Accuracy Results on Different Datasets

Table 2.2: Optimizer configurations evaluated in our experiments

Index	Server	Client	Color	Index	Server	Client	Color	Index	Server	Client	Color
1	SGD	SGD		17	RADAM	SGD		33	SGD	ADAGRAD	
2	SGD	PROX		18	AMSGRAD	SGD		34	SGD	DOWG	
3	SGD	SCAFFOLD		19	ADAMAX	SGD		35	SGD	DOG	
4	ADAM	ADAM		20	ADADELTA	SGD		36	SGD	PRODIGY	
5	NADAM	NADAM		21	RMSPROP	SGD		37	EXP	SGD	
6	RADAM	RADAM		22	ADAGRAD	SGD		38	EXP	ADAM	
7	AMSGRAD	AMSGRAD		23	DOWG	SGD		39	EXP	NADAM	
8	ADAMAX	ADAMAX		24	DOG	SGD		40	EXP	RADAM	
9	ADADELTA	ADADELTA		25	PRODIGY	SGD		41	EXP	AMSGRAD	
10	RMSPROP	RMSPROP		26	SGD	ADAM		42	EXP	ADAMAX	
11	ADAGRAD	ADAGRAD		27	SGD	NADAM		43	EXP	ADADELTA	
12	DOWG	DOWG		28	SGD	RADAM		44	EXP	RMSPROP	
13	DOG	DOG		29	SGD	AMSGRAD		45	EXP	ADAGRAD	
14	PRODIGY	PRODIGY		30	SGD	ADAMAX		46	EXP	DOWG	
15	ADAM	SGD		31	SGD	ADADELTA		47	EXP	DOG	
16	NADAM	SGD		32	SGD	RMSPROP		48	EXP	PRODIGY	

2.6 Results

The FEDEXP algorithm, proposed in Jhunjhunwala *et al.* (2023), adaptively adjusts the server step size in each federated learning round based on the pseudogradients computed during that round. The application of the extrapolation mechanism on the server with different optimizers on the client side offers better convergence and adaptability. Its consistent dominance across datasets underscores its potential as a strong baseline and motivates further research into exponential-style updates in federated optimization.

2.6.1 Which optimizer works better?

The effectiveness of an optimizer is dataset dependent, no single optimizer performs best across all tasks. However, based on our experiments, we can make the following observations:

1. DOWG consistently outperforms other parameter-free optimizers across all evaluated scenarios.

2. ADAM enhances RMSPROP by incorporating both momentum and bias-correction mechanisms, which stabilize the training process and accelerate convergence, especially in the initial training stages. Experimental results show that ADAM outperforms RMSPROP, ADADELTA, and ADAGRAD in nearly all cases.

3. Among the adaptive momentum methods, ADAM achieves the best performance in the majority of experiments. The only exception is CIFAR-10, where NADAM outperforms ADAM.

4. Among adaptive learning rate methods, ADAGRAD achieves the best performance in the majority of experiments. Unlike RMSPROP and ADADELTA, which use a decay rate (γ) to weigh historical gradients, ADAGRAD does not require tuning this parameter. This absence of decay rate tuning makes ADAGRAD more robust in scenarios where optimal hyperparameter selection is difficult, contributing to its superior performance.

CHAPTER 3

Federated Instrumental Variable Analysis via Federated Generalized Method of Moments

3.1 Introduction

Federated Learning (FL), introduced by McMahan *et al.* (2017), is a decentralized approach to training Machine Learning (ML) models where multiple clients collaboratively learn a shared model while keeping their individual data locally. Its emphasis on data privacy makes FL particularly appealing for various domains such as healthcare Nguyen *et al.* (2022); Antunes *et al.* (2022); Oh and Nadkarni (2023), finance and banking Byrd and Polychroniadou (2020); Long *et al.* (2020), smart cities and mobility Zheng *et al.* (2022); Gecer and Garbinato (2024), drug discovery Oldenhof *et al.* (2023), among others Ye *et al.* (2023). Despite its growing adoption, current FL research largely concentrates on supervised learning Kairouz *et al.* (2021), which often faces challenges in accurately predicting outcomes due to the presence of confounding variables that are not captured in training data. This issue can be effectively addressed by identifying and accounting for all latent confounding factors that affect the outcome.

For example, consider the Nature Medicine report by Dayan *et al.* (2021) on a global-scale FL to predict the effectiveness of oxygen administration (a treatment variable) to COVID-19 patients in the emergency rooms while maintaining their privacy. It is known that COVID-19 revival rates are highly influenced by lifestyle-related factors such as obesity and diabetes Wang *et al.* (2021), other co-morbidities Russell *et al.* (2023), and the patients' conditions at the emergency care admission time Izcovich *et al.* (2020). Arguably, the Dayan *et al.* (2021)'s approach may over- or under-estimate the effects of oxygen treatment.

One can address the above issue by observing and accommodating *every* confounding latent factor that may influence the outcome. Thus, it may require that obesity, diabetes,

overall health at the time of admission, and even genetic factors are accommodated; for example, using a technique such as matching Kallus (2020*b*,*a*). It may potentially render the treatment variable undergo a randomized controlled trial such as A/B testing Kohavi *et al.* (2013) on decentralized, scattered, and possibly private data. However, to our knowledge, these techniques are yet unexplored in the realms of FL.

Alternatively, one could assume conditional independence between unobserved confounders and the treatment variable, for example, the works by Shalit *et al.* (2017); Imai and Li (2023), etc. However, this may not be a fair approach for an application such as the federated estimation of effectiveness of oxygen therapy Dayan *et al.* (2021). To elaborate, Liang *et al.* (2023) suggests the hypoxia-inducible factors (HIF) – a protein that controls the rate of transcription of genetic information from DNA to messenger RNA by binding to a specific DNA sequence Latchman (1993) – plays a vital role in oxygen consumption at the cellular level. The machine learning model developed by FL implementation of Dayan *et al.* (2021) would miss the crucial counterfactual scenarios, such as HIF levels among patients undergoing oxygen therapy impacting morbidity outcomes, should it assume conditional independence between effects of oxygen treatment and every confounder. Such variables can be often traced in applications such as industry-scale federated drug discovery by AstraZeneca Oldenhof *et al.* (2023).

Instrumental variables (IV) provide a workaround to both the above issues under the assumption that the latent confounding factor influences only the treatment variable but does not directly affect the outcome. In the above example, the measure of HIF works as an instrumental variable that affects oxygen treatment as in its effective organ-level consumption but does not directly affect the mortality of the COVID-19 patient Dayan *et al.* (2021). IV can play an important role in a federated setting as the influence assumption between the confounders and the treatment variables will remain local to the clients.

IV analysis has been comprehensively explored in econometrics Angrist and Krueger (2001); Angrist and Pischke (2009) with several decades of history such as works of Wright (1928) and Reiersøl (1945). Its efficiency is now accepted for learning even high-dimensional complex causal relationships such as one in image datasets Hartford *et al.* (2017); Bennett *et al.* (2019). Naturally, the growing demand of FL entails designing methods for federated IV analysis, which, to our knowledge, is yet unexplored.

In the centralized deep learning setting, Hartford *et al.* (2017) introduced an IV analysis framework, namely DEEPIV, which uses two stages of neural networks training – first for the treatment prediction and the second with a loss function involving integration over the conditional treatment distribution. The two-stage process has precursors in applying least square regressions in the two phases Angrist and Pischke (2009)[4.1.1].

In the same setting, another approach for IV analysis applies the generalized method of moments (GMM) Wooldridge (2001). GMM is a celebrated estimation approach in social sciences and economics. It was introduced by Hansen (1982), for which he won a Nobel Prize in Economics Steif *et al.* (2014). Building on Wooldridge (2001), Bennett *et al.* (2019) introduced deep learning models to GMM estimation; they named their method DEEPGMM. Empirically, DEEPGMM outperformed DEEPIV. DEEPGMM is solved as a smooth zero-sum game formulated as a minimax optimization problem.

Prior to DEEPGMM, Lewis and Syrgkanis (2018) also employed neural networks for GMM estimation. Their method, called the adversarial generalized method of moments (AGMM), also formulated the problem as a minimax optimization to fit a GMM criterion function over a finite set of unconditional moments. DEEPGMM differs from AGMM in using a weighted norm to define the objective function. The experiments in Bennett *et al.* (2019) showed that DEEPGMM outperformed AGMM for IV analysis, and both won against DEEPIV. Nonetheless, to our knowledge, none of these methods have a federated counterpart.

Minimax optimization has been studied in federated settings Sharma *et al.* (2022); Wu *et al.* (2024), which potentially provides an underpinning for federated GMM. However, beyond the algorithm and its convergence results, there are a few key challenges:

(A) For non-i.i.d. client-local data, describing common federated GMM estimators is not immediate. It requires characterizing a synchronized model state that fit moment conditions of every client.

(B) To show that the dynamics of federated minimax optimization retrieves an equilibrium solution of the federated zero-sum game as a limit point. And,

(C) Under heterogeneity, to establish that the federated game equilibria also satisfies the equilibrium requirements of every client thereby consistently estimating the clients' local moments.

In this work, we address the above challenges. Our contributions are summarized as the following:

1. We introduce **FEDIV**: federated IV analysis. To our knowledge, **FEDIV** is the first work on IV analysis in a federated setting.
2. We present **FEDDEEPGMM**[1] – a federated adaptation of DEEPGMM of Bennett *et al.* (2019) to solve FEDIV. FEDDEEPGMM is implemented as a federated smooth zero-sum game.
3. We show that the limit points of a federated gradient descent ascent (FEDGDA) algorithm include the equilibria of the zero-sum game.
4. We show that an equilibrium solution of the federated game obtained at the server consistently estimates the moment conditions of every client.
5. We experimentally validate our algorithm. The experiments show that even for heterogenous data, FEDDEEPGMM has convergent dynamics analogous to the centralized DEEPGMM algorithm.

3.1.1 Deep Generalized Method of Moments

Consider a distributed system as a set of N clients $[N]$ with datasets

$$S^i = \{(x_j^i, y_j^i)\}_{j=1}^{n_i}, \quad \forall i \in [N].$$

We assume that for a client $i \in [N]$, the treatment and outcome variables x_j^i and y_j^i, respectively, are related by the process $Y^i = g_0^i(X^i) + \epsilon^i,\ i \in [N]$. We assume that each client-local residual ϵ^i has zero mean and finite variance, i.e. $\mathbb{E}[\epsilon^i] = 0, \mathbb{E}[(\epsilon^i)^2] < \infty$. Furthermore, we assume that the treatment variables X^i are endogenous on the clients, i.e. $\mathbb{E}[\epsilon^i|X^i] \neq 0,$ and therefore, $g_0^i(X^i) \neq \mathbb{E}[Y^i|X^i]$. We assume that the treatment variables are influenced by instrumental variables $Z^i, \forall i \in [N]$ so that

$$P(X^i|Z^i) \neq P(X^i). \tag{3.1}$$

Furthermore, the instrumental variables do not directly influence the outcome variables $Y^i, \forall i \in [N]$:

$$\mathbb{E}[\epsilon^i|Z^i] = 0. \tag{3.2}$$

[1] Wu *et al.* (2023) used FEDGMM as an acronym for federated Gaussian mixture models.

Note that, assumptions 3.1, 3.2 are local to the clients, thus, honour the data-privacy requirements of a federated learning task. In this setting, we aim to discover a common or global causal response function that would fit the data generation processes of each client without centralizing the data. More specifically, we learn a parametric function $g_0(.) \in G := \{g(., \theta) | \theta \in \Theta\}$ expressed as $g_0 := g(., \theta_0)$ for $\theta_0 \in \Theta$, defined by

$$g(., \theta_0) = \frac{1}{N} \sum_{i=1}^{N} g^i(., \theta_0). \tag{3.3}$$

The learning process essentially involves estimating the true parameter θ_0 by $\hat{\theta}$. To measure the performance of the learning procedure, we use the MSE of the estimate $\hat{g} := g(., \hat{\theta})$ against the true g_0 averaged over the clients. We adapt DEEPGMM Bennett *et al.* (2019) in the local setting of a client $i \in [N]$. For a self-contained reading, we include the description here.

3.1.2 Client-local Deep Generalized Method of Moments (DEEPGMM)

GMM estimates the parameters of the causal response function using a certain number of *moment conditions*. Define the *moment function* on a client $i \in [N]$ as a vector-valued function $f^i : \mathbb{R}^{|\mathcal{Z}|} \to \mathbb{R}^m$ with components $f^i_1, f^i_2, \ldots, f^i_m$. We consider the moment conditions as parametrized functions $\{f^i_j\}_{j=1}^m \; \forall i \in [N]$ with the assumption that their expectation is zero at the true parameter values. More specifically, using equation (3.2), we have

$$\mathbb{E}[f^i_j(Z^i)\epsilon^i] = 0, \forall j \in [m], \; \forall i \in [N], \tag{3.4}$$

We assume that m moment conditions $\{f^i_j\}_{j=1}^m$ at each client $i \in [N]$ are sufficient to identify a unique federated estimate $\hat{\theta}$ to θ_0. With (3.4), we define the moment conditions on a client $i \in [N]$ as

$$\psi(f^i_j; \theta) = 0, \; \forall j \in [m], \text{ where} \tag{3.5}$$

$$\psi(f^i; \theta) = \mathbb{E}[f^i(Z^i)\epsilon^i] = \mathbb{E}[f^i(Z^i)(Y^i - g^i(X^i; \theta)).$$

In empirical terms, the sample moments for the i-th client with n_i samples are given by

$$\psi_{n_i}(f^i;\theta) = \mathbb{E}_{n_i}[f^i(Z)\epsilon^i] = \frac{1}{n_i}\sum_{k=1}^{n_i} f^i(Z_k^i)(Y_k^i - g^i(X_k^i;\theta)), \tag{3.6}$$

where $\psi_{n_i}(f^i;\theta) = (\psi_{n_i}(f_1^i;\theta), \psi_{n_i}(f_2^i;\theta), \ldots, \psi_{n_i}(f_m^i;\theta))$ is the moment condition vector, and

$$\psi_{n_i}(f_j^i;\theta) = \frac{1}{n_i}\sum_{k=1}^{n_i} f_j^i(Z_k^i)(Y_k^i - g^i(X_k^i;\theta)). \tag{3.7}$$

Thus, for empirical estimation of the causal response function g_0^i at client $i \in [N]$, it needs to satisfy

$$\psi_{n_i}(f_j^i;\theta_0) = 0, \ \forall\, i \in [N] \text{ and } j \in [m] \tag{3.8}$$

at $\theta = \theta_0$. Equation (3.8) is reformulated as an optimization problem given by

$$\min_{\theta\in\Theta} \|\psi_{n_i}(f_1^i;\theta), \psi_{n_i}(f_2^i;\theta), \ldots, \psi_{n_i}(f_m^i;\theta)\|^2, \tag{3.9}$$

where we use the Euclidean norm $\|w\|^2 = w^T w$. Drawing inspiration from Hansen (1982), DEEPGMM used a weighted norm, which yields minimal asymptotic variance for a consistent estimator $\tilde{\theta}$, to cater to the cases of (finitely) large number of moment conditions. We adapt their weighted norm $\|w\|_{\tilde{\theta}}^2 = w^T \mathcal{C}_{\tilde{\theta}}^{-1} w$, to a client-local setting via the covariance matrix $\mathcal{C}_{\tilde{\theta}}$ defined by

$$[\mathcal{C}_{\tilde{\theta}}]_{jl} = \frac{1}{n_i}\sum_{k=1}^{n_i} f_j^i(Z_k^i) f_l^i(Z_k^i)(Y_k^i - g^i(X_k^i;\tilde{\theta}))^2. \tag{3.10}$$

Now considering the vector space $\mathcal{V}$ of real-valued functions,

$$\psi_{n_i}(f^i;\theta) = \left(\psi_{n_i}(f_1^i;\theta),\ \psi_{n_i}(f_2^i;\theta),\ \ldots,\ \psi_{n_i}(f_m^i;\theta)\right).$$

is a linear operator on $\mathcal{V}$ and

$$\mathcal{C}_{\tilde{\theta}}(f^i, h^i) = \frac{1}{n_i}\sum_{k=1}^{n_i} f^i(Z_k^i) h^i(Z_k^i)(Y_k^i - g^i(X_k^i;\tilde{\theta}))^2 \tag{3.11}$$

is a bilinear form. With that, for any subset $\mathcal{F}^i \subset \mathcal{V}$, we define a function

$$\Psi_{n_i}(\theta, \mathcal{F}^i, \tilde{\theta}) = \sup_{f^i\in\mathcal{F}^i} \psi_{n_i}(f^i;\theta) - \frac{1}{4}\mathcal{C}_{\tilde{\theta}}(f^i, f^i),$$

which leads to the following optimization problem.

Lemma 1 (Lemma 1 of Bennett *et al.* (2019))**.** *With the weighted norm defined by equation (3.10), and for* $\mathcal{F}^i = span(\{f_j^i\}_{j=1}^m)$

$$\|\psi_{n_i}(f_1^i;\theta), \psi_{n_i}(f_2^i;\theta), \ldots, \psi_{n_i}(f_m^i;\theta)\|_{\tilde{\theta}}^2 = \Psi_{n_i}(\theta, \mathcal{F}^i, \tilde{\theta}). \tag{3.12}$$

Thus, a weighted reformulation of (3.9) is given by

$$\theta^{GMM} \in \arg\min_{\theta\in\Theta} \Psi_{n_i}(\theta, \mathcal{F}^i, \tilde{\theta}). \tag{3.13}$$

As the data-dimension grows, the function class $\mathcal{F}^i$ is replaced with a class of neural networks of a certain architecture, i.e. $\mathcal{F}^i = \{f^i(z,\tau) : \tau \in \mathcal{T}\}$. Similarly, let $\mathcal{G}^i = \{g^i(x,\theta) : \theta \in \Theta\}$ be another class of neural networks with varying weights. With that, define

$$U_{\tilde{\theta}}^i(\theta,\tau) := \frac{1}{n_i}\sum_{k=1}^{n_i} f^i(Z_k^i,\tau)\left(Y_k^i - g^i(X_k^i;\theta)\right) - \frac{1}{4n_i}\sum_{k=1}^{n_i}\left(f^i(Z_k^i,\tau)\right)^2\left(Y_k^i - g^i(X_k^i;\theta)\right)^2 \tag{3.14}$$

Then (3.13) is reformulated as the following

$$\theta^{\text{DGMM}} \in \arg\min_{\theta\in\Theta}\sup_{\tau\in\mathcal{T}} U_{\tilde{\theta}}^i(\theta,\tau). \tag{3.15}$$

Equation (3.15) forms a zero-sum game, whose equilibrium solution is shown to be a true estimator to θ_0 under a set of standard assumptions; see Theorem 2 in Bennett *et al.* (2019).

3.1.3 Federated Deep GMM (FEDDEEPGMM)

The federated generalized method moment (FEDDEEPGMM) needs to find the global moment estimators for the causal response function to fit data on each client. Thus, the federated counterpart of equation (3.5) is given by

$$\psi(f;\theta) = \mathbb{E}_i[\mathbb{E}[f^i(Z^i)(Y_k^i - g^i(X^i;\theta)]] = 0, \tag{3.16}$$

where the expectation $\mathbb{E}_i$ is over the clients. In this work, we consider *full client participation*. Thus, for the empirical federated moment estimation, we formulate:

$$\psi_n(f;\theta) = \frac{1}{N}\sum_{i=1}^{N}\psi_{n_i}(f^i;\theta) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{n_i}\sum_{k=1}^{n_i} f^i(Z_k^i)(Y_k^i - g^i(X_k^i;\theta)) \tag{3.17}$$

With that, the federated moment estimation problem following (3.13) is formulated as:

$$\theta^{\text{FedDeepGMM}} \in \arg\min_{\theta\in\Theta} \|\psi_n(f;\theta)\|^2_{\tilde{\theta}}, \tag{3.18}$$

where

$$\|w\|_{-\tilde{\theta}} = w^\top g C^{-1}_{-\tilde{\theta}} x$$

is the previously defined weighted-norm with inverse covariance as weights.

In general cases, we do not have explicit knowledge of the moment conditions of various clients. We propose FEDDEEPGMM, a "deep" reformulation of the federated optimization problem based on the neural networks of a given architecture shared among clients and is shown to have the same solution as the federated GMM problem formulated earlier.

Lemma 2. *Let $\mathcal{F} = span\{f_j^i \mid i \in [N],\ j \in [m]\}$. An equivalent objective function for the federated moment estimation optimization problem (3.18) is given by:*

$$\|\psi_N(f;\theta)\|^2_{\tilde{\theta}} = \sup_{\substack{f^i\in\mathcal{F}\\ \forall i\in[N]}} \frac{1}{N}\sum_{i=1}^{N}\left(\psi_{n_i}(f^i;\theta) - \frac{1}{4}\mathcal{C}_{\tilde{\theta}}(f^i;f^i)\right), \text{ where} \tag{3.19}$$

$$\psi_{n_i}(f^i;\theta) := \frac{1}{n_i}\sum_{k=1}^{n_i} f^i(Z_k^i)(Y_k^i - g^i(X_k^i;\theta)), \text{ and } \mathcal{C}_{\tilde{\theta}}(f^i,f^i)$$
$$:= \frac{1}{n_i}\sum_{k=1}^{n_i}(f^i(Z_k^i))^2(Y_k^i - g^i(X_k^i;\tilde{\theta}))^2.$$

The federated zero-sum game is then defined by:

$$\hat{\theta}^{\text{FedDeepGMM}} \in \arg\min_{\theta\in\Theta}\sup_{\tau\in\mathcal{T}} U_{\tilde{\theta}}(\theta,\tau) := \frac{1}{N}\sum_{i=1}^{N} U^i_{\tilde{\theta}}(\theta,\tau), \tag{3.20}$$

where $U^i_{\tilde{\theta}}(\theta,\tau)$ is defined in equation (3.14). The federated GMM formulation by a zero-sum game defined by a federated minimax optimization problem (3.20) provides

the global estimator as its equilibrium solution. We solve (3.20) using the federated gradient descent ascent (FEDGDA) algorithm described next.

3.1.4 Federated Gradient Descent Ascent (FEDGDA) Algorithm

An adaptation of the standard gradient descent ascent algorithm to federated setting is well-explored: Deng and Mahdavi (2021); Sharma *et al.* (2022); Shen *et al.* (2024); Wu *et al.* (2024). The clients run the gradient descent ascent algorithm for several local updates and then the orchestrating server synchronizes them by collecting the model states, averaging them, and broadcasting it to the clients.

Similar to Bennett *et al.* (2019), we note that the federated minimax optimization problem (3.20) is not convex-concave on (θ, τ). The convergence results of variants of FEDGDA Sharma *et al.* (2022); Shen *et al.* (2024); Wu *et al.* (2024) assume that $U_{\tilde{\theta}}(\theta, \tau)$ is non-convex on θ and satisfies a $\mu-$Polyak Łojasiewicz (PL) inequality on τ, see assumption 4 in Sharma *et al.* (2022). PL condition is known to be satisfied by over-parametrized neural networks Charles and Papailiopoulos (2018); Liu *et al.* (2022). The convergence results of our method will follow Sharma *et al.* (2022). However, beyond convergence, we primarily aim to show that an optimal solution will consistently estimate the moment conditions of the clients, which we do next.

3.2 Algorithm

In this section, we present the algorithm federated generalized method moment (FED-DEEPGMM).

Algorithm 1 FEDGDA running on a federated learning server to solve the minimax problem (3.20)

Server Input: initial global estimate θ_1, τ_1; constant local learning rate α_1, α_2; total N clients

Output: global model states θ_{T+1}, τ_{T+1}

for synchronization round $t = 1, \ldots, T$ **do**
  server sends θ_t, τ_t to all clients
  for each $i \in [N]$ in parallel **do**
    $\theta^i_{t,1} \leftarrow \theta_t$, $\tau^i_{t,1} \leftarrow \tau_t$
    for $r = 1, 2, \ldots, R$ **do**
      $\theta^i_{t,r+1} = \theta^i_{t,r} - \alpha_1 \nabla_\theta f_i(\theta^i_{t,r}, \tau^i_{t,r})$
      $\tau^i_{t,r+1} = \tau^i_{t,r} + \alpha_2 \nabla_\tau f_i(\theta^i_{t,r}, \tau^i_{t,r})$
    end for
    $(\Delta\theta^i_t, \Delta\tau_t) \leftarrow (\theta^i_{t,R+1} - \theta_t, \tau^i_{t,R+1} - \tau_t)$
  end for
  $(\Delta\theta_t, \Delta\tau_t) \leftarrow \frac{1}{N}\sum_{i\in[N]}(\Delta\theta^i_t, \Delta\tau^i_t)$
  $\theta_{t+1} \leftarrow (\theta_t + \Delta\theta_t)$, $\tau_{t+1} \leftarrow (\tau_t + \Delta\tau_t)$
end for
return $\theta_{T+1}; \tau_{T+1}$

3.3 Assumptions and Theorems

As minimax is not equal to maximin in general for a non-convex-non-concave problem, it is important to model the federated game as a sequential game Jin *et al.* (2020) whose outcome would depend on what move – maximization or minimization – is taken first. We use some results from Jin *et al.* (2020), which we include here for a self-contained reading. We start with the following assumptions:

Assumption 1. *Client-local objective $U^i_{\tilde{\theta}}(\theta, \tau)$ $\forall i \in [N]$ is twice continuously differentiable for both θ and τ. Thus, the global objective $U_{\tilde{\theta}}(\theta, \tau)$ is also a twice continuously differentiable function.*

Assumption 2 (Smoothness). *The gradient of each client's local objective, $\nabla U^i_{\tilde{\theta}}(\theta, \tau)$, is Lipschitz continuous with respect to both θ and τ. For all $i \in [N]$, there exist constants $L > 0$ such that:*

$$\|\nabla_\theta U^i_{\tilde{\theta}}(\theta_1, \tau_1) - \nabla_\theta U^i_{\tilde{\theta}}(\theta_2, \tau_2)\| \leq L\|(\theta_1, \tau_1) - (\theta_2, \tau_2)\|, \text{ and}$$
$$\|\nabla_\tau U^i_{\tilde{\theta}}(\theta_1, \tau_1) - \nabla_\tau U^i_{\tilde{\theta}}(\theta_2, \tau_2)\| \leq L\|(\theta_1, \tau_1) - (\theta_2, \tau_2)\|,$$

$\forall(\theta_1, \tau_1), (\theta_2, \tau_2)$. *Thus, $U_{\tilde{\theta}}(\theta, \tau)$ is L-Lipschitz smooth.*

Assumption 3 (**Gradient Dissimilarity**). *The heterogeneity of the local gradients with respect to (w.r.t.) θ and τ is bounded as follows:*

$$\|\nabla_\theta U^i_{\tilde{\theta}}(\theta,\tau) - \nabla_\theta U_{\tilde{\theta}}(\theta,\tau)\| \leq \zeta^i_\theta \qquad \|\nabla_\tau U^i_{\tilde{\theta}}(\theta,\tau) - \nabla_\tau U_{\tilde{\theta}}(\theta,\tau)\| \leq \zeta^i_\tau,$$

where $\zeta^i_\theta,\ \zeta^i_\tau \geq 0$ are the bounds that quantify the degree of gradient dissimilarity at client $i \in [N]$.

Assumption 4 (**Hessian Dissimilarity**). *The heterogeneity in terms of hessian w.r.t. θ and τ is bounded as follows:*

$$\|\nabla^2_{\theta\theta} U^i_{\tilde{\theta}}(\theta,\tau) - \nabla^2_{\theta\theta} U_{\tilde{\theta}}(\theta,\tau)\|_\sigma \leq \rho^i_\theta, \qquad \|\nabla^2_{\tau\tau} U^i_{\tilde{\theta}}(\theta,\tau) - \nabla^2_{\tau\tau} U_{\tilde{\theta}}(\theta,\tau)\|_\sigma \leq \rho^i_\tau,$$
$$\|\nabla^2_{\theta\tau} U^i_{\tilde{\theta}}(\theta,\tau) - \nabla^2_{\theta\tau} U_{\tilde{\theta}}(\theta,\tau)\|_\sigma \leq \rho^i_{\theta\tau}, \qquad \|\nabla^2_{\tau\theta} U^i_{\tilde{\theta}}(\theta,\tau) - \nabla^2_{\tau\theta} U_{\tilde{\theta}}(\theta,\tau)\|_\sigma \leq \rho^i_{\tau\theta},$$

where $\rho^i_\theta,\ \rho^i_\tau,\ \rho^i_{\theta\tau},$ and $\rho^i_{\tau\theta} \geq 0$ quantify the degree of hessian dissimilarity at client $i \in [N]$ by spectral norm $\|.\|_\sigma$.

Assumptions 3 and 4 provide a measure of data heterogeneity across clients in a federated setting. We assume that $\zeta's$ and $\rho's$ are bounded. In the special case, when ζ and ρ's are all 0, then the data is homogeneous across clients.

We adopt the notion of Stackelberg equilibrium for pure strategies, as discussed in Jin *et al.* (2020), to characterize the solution of the minimax federated optimization problem for a non-convex non-concave function $U_{\tilde{\theta}}(\theta,\tau)$ for the sequential game where min-player goes first and the max-player goes second.

A local minimax point satisfies the Stackelberg equilibrium solution. Under Assumption 1, the local minimax points satisfy the first-order necessary condition of being a stationary point and the second-order necessary condition of $\nabla_{\tau\tau} U_{\tilde{\theta}}(\theta,\tau)$ to be NSD and Schur's Complement to be PSD when $\nabla_{\tau\tau} U_{\tilde{\theta}}(\theta,\tau)$ is ND. A sufficient condition for a point to be a strict local minimax is $\nabla_{\tau\tau} U_{\tilde{\theta}}(\theta,\tau)$ to be ND and Schur's complement be PD.

Extending the definition of Stackelberg equilibrium, we define an ϵ- approximate local minimax point and defined analogous first-order and second-order necessary and sufficient conditions for ϵ- approximate equilibrium for each federated client.

We now state the theoretical results informally.

1. We characterize that if the solution of a federated optimization problem satisfies equilibrium condition for a local minimax point in non-convex, non-concave setting then the solution will satisfy an approximate equilibrium condition with error ϵ^i for each client i. The error is dependent on heterogeneity bounds on gradient and hessian dissimilarity and eigenvalues of the problem.

2. The above result is then used to show that under specific conditions on error ϵ^i, the solution of the federated optimization problem will be a consistent solution for each client's objective.

3. We then verify that the solution of federated optimization problem obtain using FEDGDA algorithm satisfies the equilibrium condition for local minimax points for non-convex, non-concave federated problem upto some degenerate cases.

3.4 Experiments

In the experiments, we extend the experimental evaluations of Bennett *et al.* (2019) to a federated setting. More specifically, we evaluate the ability of FEDGMM to fit low and high dimensional data to demonstrate that it converges analogous to the centralized algorithm DEEPGMM. Similar to Bennett *et al.* (2019), we assess two scenarios in regards to $((X, Y), Z)$:

(a) **The instrumental and treatment variables Z and X are both low-dimensional.** In this case, we use 1-dimensional synthetic datasets corresponding to the following functions: (a) **Absolute**: $g_0(x) = |x|$, (b) **Step**: $g_0(x) = 1_{\{x \geq 0\}}$, (c) **Linear**: $g_0(x) = x$.

To generate the synthetic data, similar to Bennett *et al.* (2019); Lewis and Syrgkanis (2018) we apply the following generation process:

$$Y = g_0(X) + e + \delta \qquad \text{and } X = Z^{(1)} + Z^{(2)} + e + \gamma \tag{3.21}$$

$$(Z^{(1)}, Z^{(2)}) \sim \text{Uniform}([-3, 3]^2) \quad \text{and } e \sim \mathcal{N}(0, 1), \quad \gamma, \delta \sim \mathcal{N}(0, 0.1) \tag{3.22}$$

(b) **Z and X are low-dimensional or high-dimensional or both.** First, Z and X are generated as in (3.21,3.22). Then for high-dimensional data, we map Z and X to

an image using the mapping:

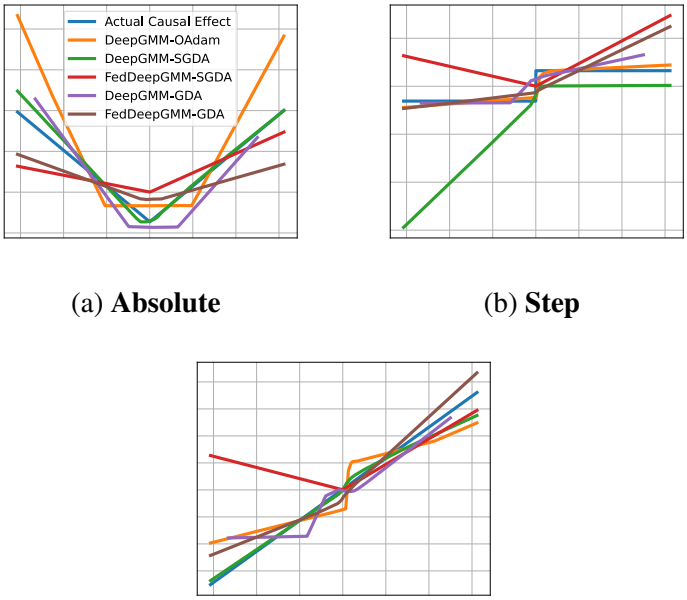
$$\text{Image}(x) = \text{Dataset}\left(\text{round}\left(\min\left(\max(1.5x + 5, 0), 9\right)\right)\right),$$

where $(\text{round}(\min(\max(1.5x + 5, 0), 9)))$ returns an integer between 0 and 9. Essentially, the function Dataset $(.)$ randomly selects an image following its index. We use datasets FEMNIST (Federated Extended MNIST) and CIFAR10 Caldas *et al.* (2018) for images of size 28×28 and $3 \times 32 \times 32$, respectively. Thus, we have the following cases: (a) $\mathbf{Dataset_z}$: $X = X^{\text{low}}, Z = \text{Image}(Z^{\text{low}})$, (b) $\mathbf{Dataset_x}$: $Z = Z^{\text{low}}, X = \text{Image}(X^{\text{low}})$, and (c) $\mathbf{Dataset_{x,z}}$: $Z = \text{Image}(Z^{\text{low}}), X = \text{Image}(X^{\text{low}})$, where $\mathbf{Dataset}$ takes values $\mathbf{FEMNIST}$ and $\mathbf{CIFAR10}$.

Bennett *et al.* (2019) used Optimistic Adam (OADAM), a variant of Adam Kingma (2015) based stochastic gradient descent ascent algorithm Daskalakis *et al.* (2018), which applies mirror descent based gradient updates. It guarantees the last iteration convergence of a GAN Goodfellow *et al.* (2014) training problem. It is known that a well-tuned SGD outperforms Adam in over-parametrized settings Wilson *et al.* (2017), closely resembling our FEDGMM implementation, where the size of neural networks often exceeds the data available on the clients. Considering that, we explored the comparative performance of GDA and SGDA against OADAM for a centralized DEEPGMM implementation. We then implemented the federated versions of each of these methods and benchmarked them for solving the federated minimax optimization problem for the FEDDEEPGMM algorithm.

For high-dimensional scenarios, we implement a convolutional neural network (CNN) architecture to process images, while for low-dimensional scenarios, we use a multilayer perceptron (MLP).

Non-i.i.d. data. We sample the train, test and validation sets similar to Bennett *et al.* (2019). For the low-dimensional scenario, we sample $n = 20000$ points for each train, validation, and test set, while, for the high-dimensional scenario, we have $n = 20000$ for the train set and $n = 10000$ for the validation and test set. To set up a non-i.i.d. distribution of data between clients, samples were divided amongst the clients using a Dirichlet distribution $Dir_S(\alpha)$ Wang *et al.* (2019), where α determines the degree of heterogeneity across S clients. We used $Dir_S(\alpha) = 0.3$ for each train, test, and validation samples.



(a) **Absolute** (b) **Step**

(c) **Linear**

Figure 3.1: Estimated $\hat{g}$ compared to true g in low-dimensional scenarios

Estimations	**DEEPGMM-OAdam**	**DEEPGMM-GDA**	**FDEEPGMM-GDA**	**DEEPGMM-SGDA**	**FDEEPGMM-SGDA**
Absolute	0.03 ± 0.01	$0.013 \pm .01$	0.4 ± 0.01	0.009 ± 0.01	0.2 ± 0.00
Step	0.3 ± 0.00	0.03 ± 0.00	0.04 ± 0.01	0.112 ± 0.00	0.23 ± 0.01
Linear	0.01 ± 0.00	0.02 ± 0.00	0.01 ± 0.00	0.03 ± 0.00	0.04 ± 0.00
$\mathbf{FEMNIST_x}$	0.50 ± 0.00	1.11 ± 0.01	0.21 ± 0.02	0.40 ± 0.01	0.19 ± 0.01
$\mathbf{FEMNIST_{x,z}}$	0.24 ± 0.00	0.46 ± 0.09	0.19 ± 0.03	0.14 ± 0.02	0.20 ± 0.00
$\mathbf{FEMNIST_z}$	0.10 ± 0.00	0.42 ± 0.01	0.24 ± 0.01	0.11 ± 0.02	0.23 ± 0.01
$\mathbf{CIFAR10_x}$	0.55 ± 0.30	0.19 ± 0.01	0.25 ± 0.03	0.20 ± 0.08	0.22 ± 0.08
$\mathbf{CIFAR10_{x,z}}$	0.40 ± 0.11	0.24 ± 0.00	0.24 ± 0.03	0.19 ± 0.03	0.22 ± 0.02
$\mathbf{CIFAR10_z}$	0.13 ± 0.03	0.13 ± 0.01	1.70 ± 2.60	0.24 ± 0.01	0.52 ± 0.60

Table 3.1: The averaged Test MSE with standard deviation on the low- and high-dimensional scenarios.

Hyperparameters. We perform extensive grid-search to tune the learning rate. For FEDSGDA, we use a minibatch-size of 256. To avoid numerical instability, we standardize the observed Y values by removing the mean and scaling to unit variance. We perform five runs of each experiment and present the mean and standard deviation of the results.

Observations and Discussion. In figure (3.1), we first observe that SGDA and GDA algorithms perform at par with OADAM to fit the DEEPGMM estimator. It establishes that hyperparameter tuning is effective. With that, we further observe that the federated algorithms efficiently fit the estimated function to the true data-generating process competitive to the centralized algorithms even though the data is decentralized and non-i.i.d.. Thus, it shows that the federated algorithm converges effectively. In Table 3.1 we present the test mean squared error (MSE) values. The MSE values indicate that the federated implementation achieves competitive convergence to their centralized counterpart. These experiments establish the efficacy of our method.

CHAPTER 4

Conclusion and Future Work

4.1 Conclusion

This thesis explored critical challenges in federated learning, focusing on optimizing client-server interactions through a comprehensive evaluation of optimization algorithms on client and server side, and extending federated learning frameworks to handle causal inference via federated instrumental variable analysis.

Our extensive experiments demonstrated that optimizer choice significantly affects federated learning performance, particularly under heterogeneous data distributions. Parameter-free optimizers like DOWG consistently outperformed others across diverse datasets, and adaptive methods such as Adam showed superior convergence properties. Furthermore, varying optimizer combinations at client and server sides revealed important trade-offs between convergence speed and stability.

We introduced FEDIV, the first federated instrumental variable analysis framework, and developed FEDDEEPGMM, a federated adaptation of deep generalized method of moments tailored for decentralized causal inference. Our theoretical analysis established the existence and characterization of equilibrium solutions in this federated zero-sum game setting.

4.2 Future Work

This thesis has laid a solid foundation by empirically analyzing optimizer choices on both client and server sides within federated learning under non-IID settings, as well as introducing FEDDEEPGMM for federated instrumental variable analysis. Nevertheless, several important directions remain open for further exploration.

First, while this work evaluated a diverse range of optimizer combinations, many promising configurations have yet to be systematically explored. For example, inte-

grating proximal optimization techniques on clients, which are known to handle heterogeneity effectively, combined with advanced adaptive exploration on the server, could yield more robust convergence and improved generalization. Investigating such hybrid optimizer strategies, especially under varying degrees of data heterogeneity and communication constraints, would deepen understanding of practical optimization in federated environments.

Second, this work introduced FEDDEEPGMM as the first federated adaptation of deep generalized method of moments for causal inference. We characterized the equilibrium solutions of federated zero-sum games in consideration of local minimax solutions for non-convex non-concave minimax optimization problems. Regardless of the analytical assumptions over the objective, the mixed strategy solutions for zero-sum games exist. However, unlike the pure strategy solutions, where the standard heterogeneity considerations over gradients and Hessians across clients, translates a local minimax solution for the federated objective to approximate local solutions for the clients, it is not immediate how a mixed strategy solution as a probability measure can be translated to that for clients. It leaves an interesting open problem to characterize the mixed startegy solutions for federated zero-sum games.

REFERENCES

1. **Angrist, J. D.** and **A. B. Krueger** (2001). Instrumental variables and the search for identification: From supply and demand to natural experiments. *Journal of Economic perspectives*, **15**(4), 69–85.

2. **Angrist, J. D.** and **J.-S. Pischke**, *Mostly harmless econometrics: An empiricist's companion*. Princeton university press, 2009.

3. **Antunes, R. S.**, **C. André da Costa**, **A. Küderle**, **I. A. Yari**, and **B. Eskofier** (2022). Federated learning for healthcare: Systematic review and architecture proposal. *ACM Transactions on Intelligent Systems and Technology (TIST)*, **13**(4), 1–23.

4. **Arivazhagan, M. G.**, **V. Aggarwal**, **A. K. Singh**, and **S. Choudhary** (2019). Federated learning with personalization layers. URL `https://arxiv.org/abs/1912.00818`.

5. **Bennett, A.**, **N. Kallus**, and **T. Schnabel** (2019). Deep generalized method of moments for instrumental variable analysis. *Advances in neural information processing systems*, **32**.

6. **Byrd, D.** and **A. Polychroniadou**, Differentially private secure multi-party computation for federated learning in financial applications. *In Proceedings of the First ACM International Conference on AI in Finance*. 2020.

7. **Caldas, S.**, **S. M. K. Duddu**, **P. Wu**, **T. Li**, **J. Konečnỳ**, **H. B. McMahan**, **V. Smith**, and **A. Talwalkar** (2018). Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*.

8. **Charles, Z.** and **D. Papailiopoulos**, Stability and generalization of learning algorithms that converge to global optima. *In International conference on machine learning*. PMLR, 2018.

9. **Choi, D.**, **C. J. Shallue**, **Z. Nado**, **J. Lee**, **C. J. Maddison**, and **G. E. Dahl** (2020). On empirical comparisons of optimizers for deep learning. URL `https://arxiv.org/abs/1910.05446`.

10. **Daskalakis, C.**, **A. Ilyas**, **V. Syrgkanis**, and **H. Zeng**, Training gans with optimism. *In International Conference on Learning Representations*. 2018.

11. **Dayan, I.**, **H. R. Roth**, **A. Zhong**, **A. Harouni**, **A. Gentili**, **A. Z. Abidin**, **A. Liu**, **A. B. Costa**, **B. J. Wood**, **C.-S. Tsai**, *et al.* (2021). Federated learning for predicting clinical outcomes in patients with covid-19. *Nature medicine*, **27**(10), 1735–1743.

12. **Deng, Y.** and **M. Mahdavi**, Local stochastic gradient descent ascent: Convergence analysis and communication efficiency. *In International Conference on Artificial Intelligence and Statistics*. PMLR, 2021.

13. **Durmus, A. E.**, **Z. Yue**, **M. Ramon**, **M. Matthew**, **W. Paul**, and **S. Venkatesh**, Federated learning based on dynamic regularization. *In International Conference on Learning Representations*. 2021.

14. **Gecer, M.** and **B. Garbinato** (2024). Federated learning for mobility applications. *ACM Computing Surveys*, **56**(5), 1–28.

15. **Goodfellow, I.**, **J. Pouget-Abadie**, **M. Mirza**, **B. Xu**, **D. Warde-Farley**, **S. Ozair**, **A. Courville**, and **Y. Bengio** (2014). Generative adversarial nets. *Advances in neural information processing systems*, **27**.

16. **Hansen, L. P.** (1982). Large sample properties of generalized method of moments estimators. *Econometrica: Journal of the econometric society*, 1029–1054.

17. **Hartford, J.**, **G. Lewis**, **K. Leyton-Brown**, and **M. Taddy**, Deep iv: A flexible approach for counterfactual prediction. *In International Conference on Machine Learning*. PMLR, 2017.

18. **Imai, K.** and **M. L. Li** (2023). Experimental evaluation of individualized treatment rules. *Journal of the American Statistical Association*, **118**(541), 242–256.

19. **Izcovich, A.**, **M. A. Ragusa**, **F. Tortosa**, **M. A. Lavena Marzio**, **C. Agnoletti**, **A. Bengolea**, **A. Ceirano**, **F. Espinosa**, **E. Saavedra**, **V. Sanguine**, *et al.* (2020). Prognostic factors for severity and mortality in patients infected with covid-19: A systematic review. *PloS one*, **15**(11), e0241955.

20. **Jhunjhunwala, D.**, **S. Wang**, and **G. Joshi** (2023). FedExP: Speeding Up Federated Averaging via Extrapolation. ArXiv:2301.09604 [cs].

21. **Jiang, M.** and **R. Tang**, Study on hyperparameter adaptive federated learning. *In 2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI)*. 2023.

22. **Jin, C.**, **P. Netrapalli**, and **M. Jordan**, What is local optimality in nonconvex-nonconcave minimax optimization? *In* **H. D. III** and **A. Singh** (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*. PMLR, 2020. URL `https://proceedings.mlr.press/v119/jin20e.html`.

23. **Kairouz, P.**, **H. B. McMahan**, **B. Avent**, **A. Bellet**, **M. Bennis**, **A. N. Bhagoji**, **K. Bonawitz**, **Z. Charles**, **G. Cormode**, **R. Cummings**, *et al.* (2021). Advances and open problems in federated learning. *Foundations and trends® in machine learning*, **14**(1–2), 1–210.

24. **Kallus, N.**, Deepmatch: Balancing deep covariate representations for causal inference using adversarial training. *In International Conference on Machine Learning*. PMLR, 2020*a*.

25. **Kallus, N.** (2020*b*). Generalized optimal matching methods for causal inference. *Journal of Machine Learning Research*, **21**(62), 1–54.

26. **Karimireddy, S. P.**, **M. Jaggi**, **S. Kale**, **M. Mohri**, **S. J. Reddi**, **S. U. Stich**, and **A. T. Suresh** (2021). Mime: Mimicking centralized stochastic algorithms in federated learning. URL `https://arxiv.org/abs/2008.03606`.

27. **Karimireddy, S. P.**, **S. Kale**, **M. Mohri**, **S. Reddi**, **S. Stich**, and **A. T. Suresh**, Scaffold: Stochastic controlled averaging for federated learning. *In International conference on machine learning*. PMLR, 2020.

28. **Khodak, M.**, **R. Tu**, **T. Li**, **L. Li**, **M.-F. Balcan**, **V. Smith**, and **A. Talwalkar** (2021). Federated hyperparameter tuning: Challenges, baselines, and connections to weight-sharing. URL `https://arxiv.org/abs/2106.04502`.

29. **Kingma, D. P.** (2015). Adam: A method for stochastic optimization. *ICLR*.

30. **Kohavi, R.**, **A. Deng**, **B. Frasca**, **T. Walker**, **Y. Xu**, and **N. Pohlmann**, Online controlled experiments at large scale. *In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2013.

31. **Latchman, D. S.** (1993). Transcription factors: an overview. *International journal of experimental pathology*, **74**(5), 417.

32. **Lewis, G.** and **V. Syrgkanis** (2018). Adversarial generalized method of moments. URL `https://arxiv.org/abs/1803.07164`.

33. **Li, Q.**, **B. He**, and **D. Song** (2021). Model-contrastive federated learning. URL `https://arxiv.org/abs/2103.16257`.

34. **Li, T.**, **A. K. Sahu**, **M. Zaheer**, **M. Sanjabi**, **A. Talwalkar**, and **V. Smith** (2020). Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, **2**, 429–450.

35. **Liang, Y.**, **W. Ruan**, **Y. Jiang**, **R. Smalling**, **X. Yuan**, and **H. K. Eltzschig** (2023). Interplay of hypoxia-inducible factors and oxygen therapy in cardiovascular medicine. *Nature Reviews Cardiology*, **20**(11), 723–737.

36. **Liu, C.**, **L. Zhu**, and **M. Belkin** (2022). Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, **59**, 85–116.

37. **Long, G.**, **Y. Tan**, **J. Jiang**, and **C. Zhang**, Federated learning for open banking. *In Federated learning: privacy and incentive*. Springer, 2020, 240–254.

38. **Malinovsky, G.**, **K. Mishchenko**, and **P. Richtárik** (2022). Server-side stepsizes and sampling without replacement provably help in federated optimization. URL `https://arxiv.org/abs/2201.11066`.

39. **McMahan, B.**, **E. Moore**, **D. Ramage**, **S. Hampson**, and **B. A. y Arcas**, Communication-efficient learning of deep networks from decentralized data. *In Artificial intelligence and statistics*. PMLR, 2017.

40. **Nguyen, D. C.**, **Q.-V. Pham**, **P. N. Pathirana**, **M. Ding**, **A. Seneviratne**, **Z. Lin**, **O. Dobre**, and **W.-J. Hwang** (2022). Federated learning for smart healthcare: A survey. *ACM Computing Surveys (Csur)*, **55**(3), 1–37.

41. **Noble, M.**, **A. Bellet**, and **A. Dieuleveut** (2023). Differentially private federated learning on heterogeneous data. URL `https://arxiv.org/abs/2111.09278`.

42. **Oh, W.** and **G. N. Nadkarni** (2023). Federated learning in health care using structured medical data. *Advances in kidney disease and health*, **30**(1), 4–16.

43. **Oldenhof, M.**, **G. Ács**, **B. Pejó**, **A. Schuffenhauer**, **N. Holway**, **N. Sturm**, **A. Dieckmann**, **O. Fortmeier**, **E. Boniface**, **C. Mayer**, *et al.*, Industry-scale orchestrated federated learning for drug discovery. *In Proceedings of the aaai conference on artificial intelligence*, volume 37. 2023.

44. **Pascanu, R.**, **C. Lyle**, **I.-V. Modoranu**, **N. E. Borras**, **D. Alistarh**, **P. Velickovic**, **S. Chandar**, **S. De**, and **J. Martens** (2025). Optimizers qualitatively alter solutions and we should leverage this. *arXiv preprint arXiv:2507.12224*.

45. **Pillutla, K.**, **S. M. Kakade**, and **Z. Harchaoui** (2022). Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, **70**, 1142–1154. ISSN 1941-0476. URL `http://dx.doi.org/10.1109/TSP.2022.3153135`.

46. **Reddi, S.**, **Z. Charles**, **M. Zaheer**, **Z. Garrett**, **K. Rush**, **J. Konečný**, **S. Kumar**, and **H. B. McMahan** (2021). Adaptive federated optimization. URL `https://arxiv.org/abs/2003.00295`.

47. **Reiersøl, O.** (1945). *Confluence analysis by means of instrumental sets of variables*. Ph.D. thesis, Almqvist & Wiksell.

48. **Robbins, H.** and **S. Monro** (1951). A stochastic approximation method. *The annals of mathematical statistics*, 400–407.

49. **Russell, C. D.**, **N. I. Lone**, and **J. K. Baillie** (2023). Comorbidities, multimorbidity and covid-19. *Nature medicine*, **29**(2), 334–343.

50. **Schmidt, R. M.**, **F. Schneider**, and **P. Hennig**, Descending through a crowded valley-benchmarking deep learning optimizers. *In International Conference on Machine Learning*. PMLR, 2021.

51. **Shalit, U.**, **F. D. Johansson**, and **D. Sontag**, Estimating individual treatment effect: generalization bounds and algorithms. *In International conference on machine learning*. PMLR, 2017.

52. **Sharma, P.**, **R. Panda**, **G. Joshi**, and **P. Varshney**, Federated minimax optimization: Improved convergence analyses and algorithms. *In International Conference on Machine Learning*. PMLR, 2022.

53. **Shen, W.**, **M. Huang**, **J. Zhang**, and **C. Shen**, Stochastic smoothed gradient descent ascent for federated minimax optimization. *In International Conference on Artificial Intelligence and Statistics*. PMLR, 2024.

54. **Steif, A. E.**, **J. Fan**, **X.-L. Meng**, **B. Yu**, **D. Madigan**, and **J. R. Manteiga** (2014). Nobel prize in economics. *IMS Bulletin*, **43**(1).

55. **Su, S.**, **B. Li**, and **X. Xue** (2023). One-shot federated learning without server-side training. *Neural Networks*, **164**, 203–215.

56. **Wang, H.**, **M. Yurochkin**, **Y. Sun**, **D. Papailiopoulos**, and **Y. Khazaeni**, Federated learning with matched averaging. *In International Conference on Learning Representations*. 2019.

57. **Wang, J.**, **Q. Liu**, **H. Liang**, **G. Joshi**, and **H. V. Poor** (2020). Tackling the objective inconsistency problem in heterogeneous federated optimization. URL `https://arxiv.org/abs/2007.07481`.

58. **Wang, J.**, **T. Sato**, and **A. Sakuraba** (2021). Worldwide association of lifestyle-related factors and covid-19 mortality. *Annals of medicine*, **53**(1), 1531–1536.

59. **Wilson, A. C.**, **R. Roelofs**, **M. Stern**, **N. Srebro**, and **B. Recht** (2017). The marginal value of adaptive gradient methods in machine learning. *Advances in neural information processing systems*, **30**.

60. **Wooldridge, J. M.** (2001). Applications of generalized method of moments estimation. *Journal of Economic perspectives*, **15**(4), 87–100.

61. **Wright, P. G.**, *The tariff on animal and vegetable oils*. 26. Macmillan, 1928.

62. **Wu, X.**, **J. Sun**, **Z. Hu**, **A. Zhang**, and **H. Huang** (2024). Solving a class of non-convex minimax optimization in federated learning. *Advances in Neural Information Processing Systems*, **36**.

63. **Wu, Y.**, **C. Tian**, **J. Li**, **H. Sun**, **K. Tam**, **L. Li**, and **C. Xu** (2025). A survey on federated fine-tuning of large language models. URL `https://arxiv.org/abs/2503.12016`.

64. **Wu, Y.**, **S. Zhang**, **W. Yu**, **Y. Liu**, **Q. Gu**, **D. Zhou**, **H. Chen**, and **W. Cheng**, Personalized federated learning under mixture of distributions. *In International Conference on Machine Learning*. PMLR, 2023.

65. **Ye, M.**, **X. Fang**, **B. Du**, **P. C. Yuen**, and **D. Tao** (2023). Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Computing Surveys*, **56**(3), 1–44.

66. **Yurochkin, M.**, **M. Agarwal**, **S. Ghosh**, **K. Greenewald**, **T. N. Hoang**, and **Y. Khazaeni** (2019). Bayesian nonparametric federated learning of neural networks. URL `https://arxiv.org/abs/1905.12022`.

67. **Zheng, Z.**, **Y. Zhou**, **Y. Sun**, **Z. Wang**, **B. Liu**, and **K. Li** (2022). Applications of federated learning in smart cities: recent advances, taxonomy, and open challenges. *Connection Science*, **34**(1), 1–28.