Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

UNIVERSITY OF PADUA
DEPARTMENT OF INFORMATION ENGINEERING

MASTER DEGREE IN ICT FOR INTERNET AND MULTIMEDIA

ADAPTIVE LOCAL STEPS AND SERVER-SIDE OPTIMIZATION FOR ASYNCHRONOUS FEDERETED CONTINUAL LEARNING

Supervisor:
Prof. Pietro Zanuttigh

Master Candidate:
Davide Antonelli
2157547

ACADEMIC YEAR 2025/2026

Graduation date 09/04/2026

Abstract

Asynchronous Federated Continual Learning (AFCL) represents a realistic yet challenging scenario where heterogeneous clients learn distinct tasks over time without central coordination. While prototype-based methods like FedSpace effectively address catastrophic forgetting, they often suffer from high computational costs and performance degradation in non-IID settings. This thesis proposes a framework designed to reconcile efficiency and robustness. First, we introduce FedAlt (Federated Adaptive Local Training), a strategy that accelerates convergence by dynamically adjusting local training steps, significantly reducing communication overhead. To recover the accuracy loss induced by accelerated training, we propose FedSpo (Federated Server-Side Prototype Optimization), a privacy-preserving module where the server refines the global model using synthetic data generated from global prototypes. We investigate advanced tuning strategies, including Quality Filtering for medium fragmentation and Soft Mixup for extreme fragmentation. Experiments on CIFAR-100 (with 50, 100, and 500 clients) demonstrate that our approach establishes a highly effective trade-off between computational efficiency and classification accuracy compared to state of the art baselines.

Contents

List of Figures

Chapter 1

Introduction

1.1 Context and Motivation

Over the past decade, the proliferation of interconnected devices has significantly altered the technological landscape. The expansion of the Internet of Things (IoT), alongside the widespread use of smartphones, autonomous vehicles, and smart infrastructures, generates large volumes of data directly at the "edge" of the network. These distributed devices produce multi-modal information, including high-resolution images, video streams, and sensor logs. Historically, training Deep Learning models on this data relied on a centralized paradigm, where raw data from peripheral devices was transmitted to cloud servers or centralized data centers. In these environments, GPU clusters processed static datasets to train neural network architectures, which were subsequently deployed back to the edge for inference.

However, this cloud-centric approach faces several practical limitations. The continuous transmission of high-dimensional data often saturates network bandwidth and introduces latency, which is prohibitive for real-time applications. Additionally, centralizing data raises significant privacy and security concerns. Legal frameworks, such as the General Data Protection Regulation (GDPR) in the European Union and the California Consumer Privacy Act (CCPA), impose strict limitations on the collection, transmission, and centralized storage of Personally Identifiable Information. Furthermore, managing and computing gradients over the aggregated data of millions of active devices creates computational and storage bottlenecks on centralized servers, limiting the scalability of the traditional cloud paradigm.

To address these challenges, *Federated Learning* (FL) has emerged as a decentralized alternative. This framework inverts the traditional methodology: rather than migrating raw data to a centralized model, the model is distributed to the data sources. In a standard federated ecosystem, participating client devices collaboratively train a shared global model while keeping their datasets on-device. A central server broadcasts the current global model to a subset of available clients. These clients execute local training on their private data and transmit only the computed model updates (e.g., gradient vectors or weight differentials) back to the server. The server aggregates these updates to refine the global model, learning the underlying data distribution without directly accessing the raw samples.

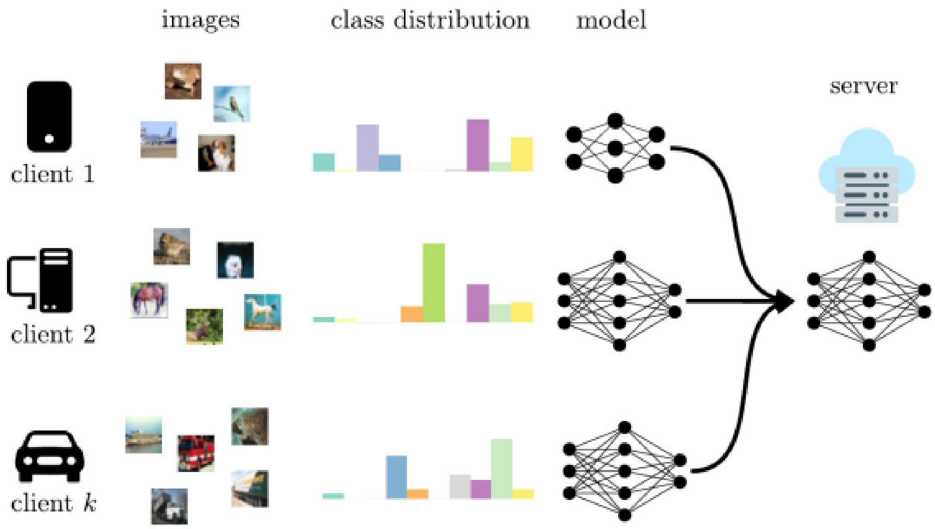

Figure 1.1: Standard Federated Learning Architecture. *Source: Federated Learning in Computer Vision*

While standard Federated Learning addresses privacy concerns, its deployment in edge environments introduces practical complexities. Edge devices typically operate in dynamic settings where data arrives as continuous, non-stationary streams. Consequently, the system must incorporate *Continual Learning* (CL), defined as the ability to sequentially acquire new knowledge or adapt to shifting data distributions over time without losing previously acquired capabilities [4].

This requirement exposes the network to the plasticity-stability dilemma. Without architectural countermeasures, neural networks suffer from *Catastrophic Forgetting*, a phenomenon where the parameter updates required to learn a new task overwrite the weights encoding the knowledge of previously learned tasks.

The operational context of this thesis lies at the intersection of these domains, defined as *Asynchronous Federated Continual Learning* (AFCL). As introduced by Shenaj et al. [7], AFCL is characterized by system asynchrony arising from heterogeneity in hardware capabilities, battery levels, and network bandwidth across clients. Devices cannot participate in synchronized communication rounds; instead, they independently fetch the model, train locally, and push updates at uncoordinated intervals, introducing update staleness. This asynchrony is compounded by

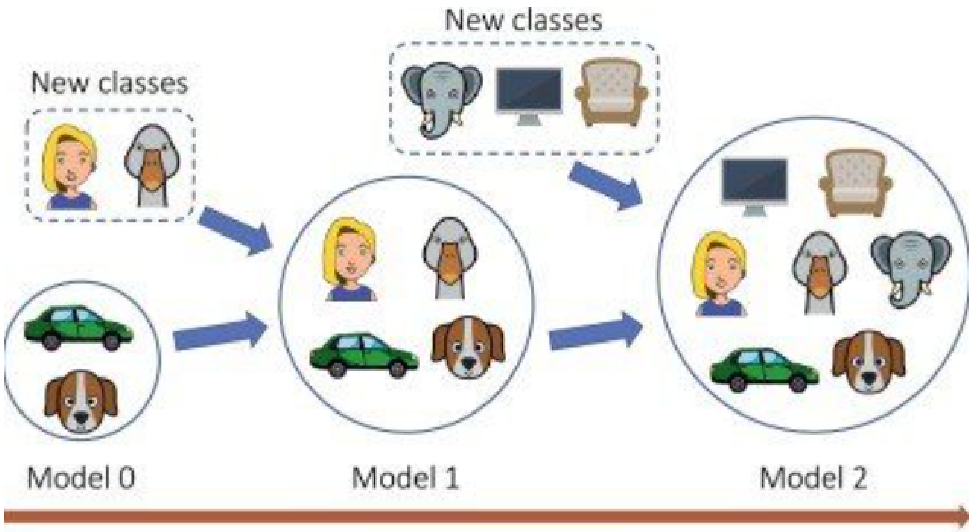

Figure 1.2: Continual Learning Paradigm. *Source: Advanced Methods and Deep Learning in Computer Vision*

statistical heterogeneity, as local datasets are typically Non-Independent and Identically Distributed (Non-IID). Operating in this environment requires algorithms capable of mitigating catastrophic forgetting and converging efficiently despite the variance introduced by Non-IID data distributions and delayed, asynchronous updates.

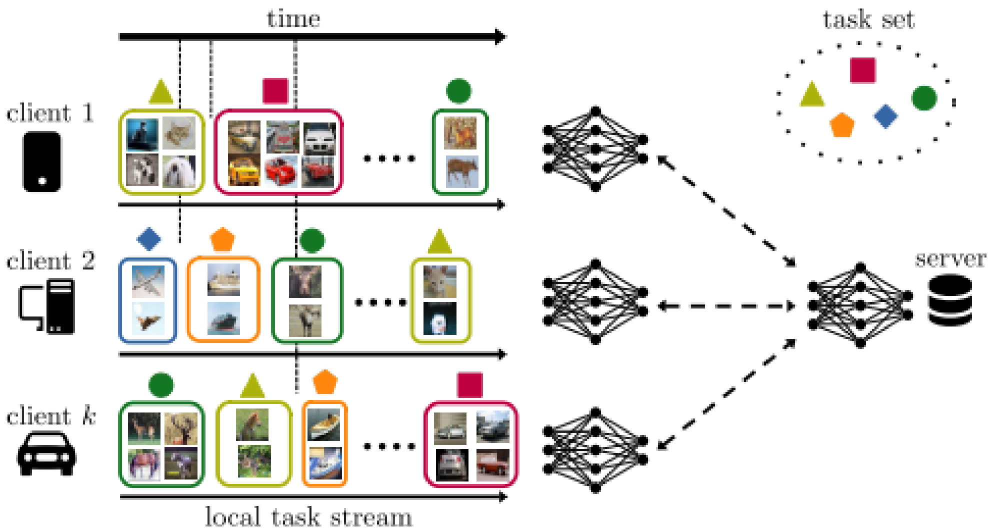

Figure 1.3: Asynchronous Federated Continual Learning (AFCL) Environment. *Source: Asynchronous Federated Continual Learning*

1.2 Problem Statement

In the context of Asynchronous Federated Continual Learning (AFCL), the FedSpace framework represents a key baseline for addressing Catastrophic Forgetting. To mitigate the loss of historical knowledge, FedSpace employs a regularization mechanism based on prototype learning. In this architecture, the central server maintains an aggregated representation (prototype) of each semantic class within the latent feature space. During asynchronous local training, clients align their local feature embeddings with these global anchors by minimizing the spatial distance between them. This prototype-driven regularization preserves the geometric structure of consolidated classes, preventing local model parameters from diverging significantly from the global consensus, thereby retaining past knowledge while adhering to data privacy constraints.

Despite its effectiveness in preserving accuracy, FedSpace and similar prototype-based approaches exhibits computational limitations that hinder its deployment in resource-constrained edge environments. The geometric regularization imposed by the prototype loss increases the computational overhead of the spatial alignment process. To reach convergence under these prototype constraints, clients are typically required to execute a large, statically fixed number of local training epochs during each communication round. This extensive local training protocol leads to increased energy consumption, accelerated battery depletion, and higher communication latency, which conflict with the efficiency requirements of edge computing.

Furthermore, the static optimization schedule employed by standard federated algorithms exacerbates this inefficiency. In conventional setups, the number of local optimization steps is determined *a priori* as a fixed hyperparameter applied uniformly across all participating devices, regardless of their convergence state or data distribution. This homogeneous approach is suboptimal in strictly Non-IID settings. Clients with local data distributions already aligned with the global model expend computational resources on redundant training steps, increasing the risk of local overfitting. Conversely, clients with highly divergent data distributions may lack sufficient training steps to overcome the prototype regularization, resulting in an under-representation of their local knowledge during global aggregation.

In addition to computational costs, the robustness of current prototype-based methods degrades as network fragmentation increases. When scaling the number of participating clients (e.g., from 50 to 500 nodes), the global dataset becomes increasingly partitioned, resulting in sparse local data silos. Under extreme fragmentation, the volume of local data available to a client for a specific class is often insufficient to compute a reliable, low-variance local prototype. Aggregating these high-variance local prototypes at the central server yields noisy global prototypes. Rather than functioning as stable geometric anchors, these corrupted prototypes introduce noise into the local training trajectories of subsequent clients, destabilizing the optimization process and degrading global classification accuracy.

The primary objective of this thesis is to decouple the dependency between computational cost and global model accuracy in AFCL. Specifically, this work investigates the viability of accelerating client-side training by dynamically adapting local execution steps, while simultaneously leveraging privacy-preserving, server-side generative techniques to recover overall global model performance across varying degrees of Non-IID data fragmentation.

1.3 Proposed Approach and Main Contributions

To address the computational limitations of existing prototype-based architectures, such as FedSpace [7], this thesis proposes an algorithmic framework for Asynchronous Federated Continual Learning (AFCL). The core objective of this approach is to decouple computational overhead from the preservation of global model accuracy. Rather than requiring resource-constrained edge devices to execute a fixed number of local training epochs, we distribute the computational load across the network. This is achieved through the integration of two complementary modules: a client-side efficiency controller adapted from Federated Adaptive Local Training (FedAlt) [5], and a server-side accuracy recovery module proposed in this work, denoted as Federated Server-Side Prototype Optimization (FedSpo).

The main contributions of this thesis can be summarized as follows:

1. **Adaptation of FedAlt for AFCL:** The first component of our framework integrates the FedAlt module [5]. We adapt this dynamic early-stopping mechanism to operate within an asynchronous continual learning environment. Instead of executing a predefined number of local steps, clients determine their training termination point by calculating the cosine similarity between local and global representations directly in the latent feature space. We demonstrate that applying this self-regulation logic to AFCL reduces energy consumption and communication latency while mitigating client drift.

2. **Introduction of the FedSpo Module:** To recover the global accuracy sacrificed by the truncated local training introduced by FedAlt, we propose the FedSpo module. Utilizing aggregated global class prototypes and their spatial variances, FedSpo implements a privacy-preserving generative replay

strategy on the central server. The server synthesizes Gaussian-distributed pseudo-samples in the latent feature space to fine-tune the global classification boundaries, reconstructing the target data distribution without accessing raw client data.

3. **Analysis of Network Fragmentation and Adaptive Regularization:** Finally, this work provides an empirical analysis demonstrating that server-side generative strategies must dynamically adapt to the scale of network fragmentation. Through experiments on a Dirichlet-partitioned CIFAR-100 dataset, we categorize the network dynamics into three operational scenarios and propose specific interventions for each:

   - *Medium Fragmentation (50 clients):* We demonstrate that accuracy recovery is maximized using a radius-based filtering mechanism that discards high-variance prototype outliers.
   - *High Fragmentation (100 clients):* We identify an intermediate statistical regime where disabling the variance-based filtering is necessary to prevent the elimination of minority classes.
   - *Extreme Fragmentation (500 clients):* To mitigate the severe variance inherent to this regime, we design a regularization strategy combining strict variance-based filtering with latent space interpolation, inspired by *Mixup* [12] and *Manifold Mixup* [8]. This adaptive pipeline stabilizes the optimization process at scale.

1.4 Thesis Structure

The remainder of this manuscript is organized as follows. The exposition builds the theoretical context of distributed machine learning before detailing the proposed algorithmic contributions and their quantitative evaluation.

Chapter 2 establishes the theoretical background for this research. It introduces the foundations of Federated Learning, Continual Learning, and Catastrophic Forgetting. It also outlines the core concepts of asynchronous optimization, prototype-based latent space regularization, and adaptive training mechanisms.

Chapter 3 provides a comprehensive review of the related work. It analyzes existing literature on distributed optimization under Non-IID constraints and Continual Learning strategies, culminating in a critical review of prototype-based Federated Learning frameworks and the specific limitations of FedSpace.

Chapter 4 presents the proposed methodology. It formally defines the Asynchronous Federated Continual Learning (AFCL) system model and introduces the architecture of the dual-module solution. This includes the client-side FedAlt algorithm for adaptive local training, and the server-side FedSpo refinement module, detailing the fragmentation-aware tuning and regularization strategies.

Chapter 5 details the experimental results. It outlines the dataset partitioning and the baseline evaluation metrics, followed by a rigorous empirical evaluation of the framework across medium, high, and extreme fragmentation regimes. The chapter concludes with a summary analyzing the trade-off between computational efficiency and classification accuracy.

Finally, Chapter 6 summarizes the methodological contributions and the experimental findings of this study. It acknowledges the structural limitations of the proposed framework and outlines potential directions for future research in adaptive federated edge ecosystems.

Chapter 2

Background

2.1 Foundations of Federated Learning

Federated Learning (FL) addresses the need to train machine learning models across distributed networks while strictly preserving data privacy. FL formulates a global optimization problem where the training data remains decentralized across multiple local nodes (clients) and is never explicitly shared with a central server.

As detailed by Shenaj et al. [6], the primary objective of FL is to find a set of global model parameters that minimizes a weighted average of the local loss functions. This decentralized approach introduces optimization challenges, particularly regarding convergence stability when local datasets are Non-Independent and Identically Distributed (Non-IID) and fail to represent the global data distribution.

The foundational algorithm for model aggregation in this setting is Federated Averaging (`FedAvg`). `FedAvg` operates through a series of communication rounds where, in each round, a subset of clients downloads the current global model and performs a fixed number of local Stochastic Gradient Descent (SGD) steps. However, the convergence of `FedAvg` is highly sensitive to statistical heterogeneity. When local data is severely skewed toward specific classes a standard characteristic of partitioned datasets like CIFAR-100 local updates diverge toward disjoint local optima. This parametric divergence from the global objective, formally known as *client drift*, is a fundamental challenge in FL and motivates the adaptive control mechanisms proposed in this work.

2.2 The Challenge of Continual Learning and Catastrophic Forgetting

While Federated Learning addresses the spatial distribution of data, Continual Learning (CL) focuses on the temporal dimension of model training. In dynamic environments, data is not static but arrives sequentially as distinct tasks or classes over time. The primary challenge in this domain is *Catastrophic Forgetting*, a phenomenon where artificial neural networks lose information acquired from previous tasks when optimized for new ones.

As detailed by Wang et al. [9], this occurs because the backpropagation algorithm updates synaptic weights to minimize the loss on the current data, often overwriting the parameter configurations that were optimal for past objectives. The impact of Catastrophic Forgetting is exacerbated in Federated Continual Learning (FCL). In this setting, the optimization process must bridge the distribution shifts across different clients (spatial heterogeneity) and across different time steps (temporal non-stationarity).

Systems operating in this environment must navigate the *plasticity-stability dilemma*: the trade-off between the capacity to integrate new knowledge (*plasticity*) and the ability to retain existing representations (*stability*). Standard CL approaches, such as rehearsal-based strategies that store a buffer of historical data, conflict with the privacy constraints of Federated Learning, as storing or sharing raw client data is prohibited. Consequently, stability mechanisms based on feature-space prototypes have emerged as a viable alternative.

While sharing class centroids avoids the transmission of raw data and adheres to data minimization principles, prototypes do not guarantee absolute privacy. Literature in adversarial machine learning demonstrates that latent representations remain vulnerable to information leakage, as feature inversion attacks can potentially reconstruct sensitive attributes from shared prototypes. Nevertheless, within the scope of Asynchronous Federated Continual Learning (AFCL), prototype-based regularization represents a practical compromise. It mitigates Catastrophic Forgetting while maintaining a higher degree of privacy compared to centralized rehearsal buffers.

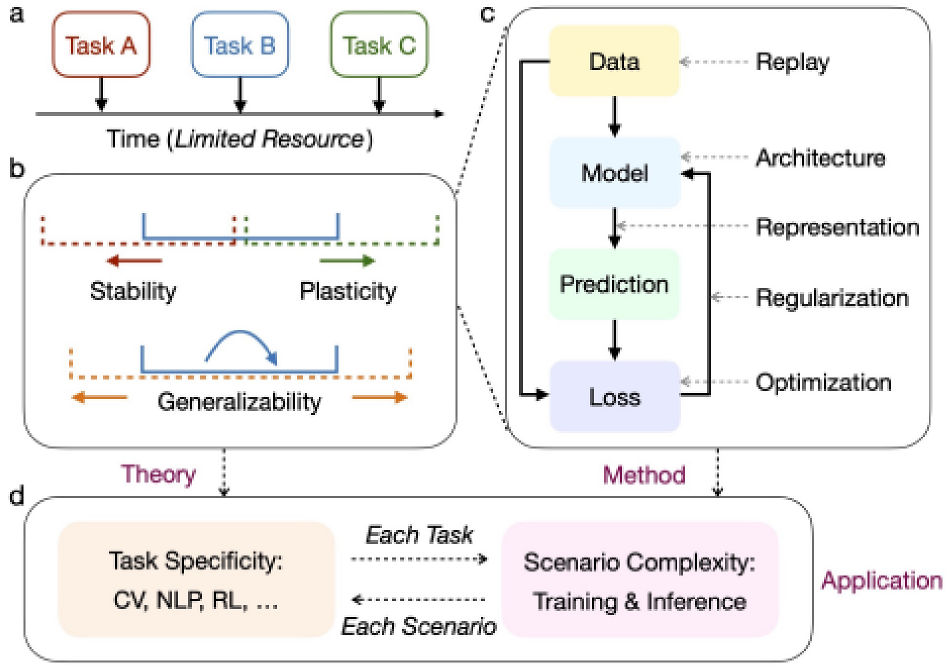

Figure 2.1: A conceptual framework of Continual Learning. *Source: A Comprehensive Survey of Continual Learning: Theory, Method and Application*

2.3 Asynchronous Optimization in Distributed Systems

While many Federated Learning frameworks assume a synchronous execution model where the server waits for a selected subset of clients to complete local training before aggregation this assumption is often impractical in edge computing environments. Edge devices exhibit high heterogeneity in computational power, memory capacity, energy availability, and network bandwidth. Relying on synchronous update cycles introduces the straggler problem, where the federation's overall progress is bottlenecked by the slowest participant, resulting in inefficient resource utilization and increased system latency.

Asynchronous Federated Learning (AFL) mitigates these bottlenecks by allowing the server to update the global model immediately upon receiving a local update from any individual client. However, this asynchronous protocol introduces the challenge of *update staleness*. By the time a slower client's update reaches the server, the global model may have already been advanced by faster clients. Consequently, stale updates are computed with respect to an obsolete global state, which can destabilize the optimization trajectory.

In the context of Asynchronous Federated Continual Learning (AFCL), as formalized by Shenaj et al. [7], the interaction between system asynchrony, Non-IID data distributions, and sequential task learning compounds these optimization challenges. Navigating this environment necessitates specialized regularization techniques, such as the prototype-based alignment utilized in FedSpace, to prevent delayed updates from inducing model divergence or exacerbating Catastrophic Forgetting.

2.4 Prototype-based Learning and Latent Space Regularization

Integrating prototype-based methodologies into Federated Learning provides a robust alternative to architectures relying exclusively on parameter averaging. In prototype-based learning, each class is represented by a mean vector (centroid) computed within the latent feature space, typically extracted from the penultimate layer of a neural network.

As demonstrated in FedSpace [7], global prototypes serve as geometric anchors that stabilize the distributed optimization process. Instead of relying solely on empirical risk minimization which can exhibit high variance in Non-IID scenarios the system imposes a structural constraint. Specifically, local feature embeddings for a given class are regularized to align spatially with the corresponding global prototype.

This paradigm shifts the optimization objective to include geometric alignment. The distance between a local sample embedding and the global prototype is minimized using a regularization loss function, typically based on Euclidean distance or cosine similarity. This regularization mitigates client drift by preventing local models from overfitting to skewed private data distributions, thereby maintaining global alignment.

Furthermore, in Continual Learning, prototypes function as a privacy-preserving representation of past distributions. By maintaining the spatial coordinates of class centroids from previous tasks, the network can acquire new classes without disrupting the latent space regions dedicated to prior knowledge. This approach addresses Catastrophic Forgetting without requiring the storage or rehearsal of raw data samples, thus adhering to federated privacy constraints.

2.5 Adaptive Mechanisms for Computational Efficiency

A critical area of research in distributed computing focuses on optimizing computational resources and maximizing energy efficiency across resource-constrained edge devices. Standard Federated Learning algorithms typically enforce a fixed number of local training epochs per communication round. However, this homogeneous approach does not account for the variability in local data distributions and the disparate computational capabilities of participating nodes.

Adaptive mechanisms address this limitation by dynamically modulating the computational effort based on the optimization progress. The framework of *Adaptive Local Training* (ALT), introduced by Shenaj et al. [5], replaces static epoch hyperparameters with a feedback mechanism where the number of local steps is regulated by continuously monitoring the learning trajectory on the client.

This adaptive approach mitigates the adverse effects of excessive local training on highly skewed Non-IID data partitions, which often leads to client drift. By tracking the representational similarity between the evolving local model and the frozen global model downloaded at the start of the round using metrics such as cosine similarity or Centered Kernel Alignment (CKA) the system can identify the point of diminishing returns.

When the computed divergence exceeds a dynamic threshold, the local training phase is terminated early. This early-stopping mechanism prevents severe client drift, reduces battery consumption, and minimizes bandwidth usage. Implementing this logic within asynchronous environments requires balancing the faster update cycles of early-stopping clients with the need to maintain an informative update stream for continual learning.

2.6 Data Augmentation and Generative Regularization

The final theoretical component of the proposed framework involves data augmentation and synthetic data generation for regularization. In highly fragmented federated scenarios such as networks comprising hundreds of clients with limited samples per class aligning local embeddings to global prototypes may be insufficient to construct robust decision boundaries.

To address this data sparsity, regularization techniques such as *Mixup* [12] and

Manifold Mixup [8] train neural networks on continuous convex combinations of data samples and their corresponding labels. Interpolating the model's behavior across intermediate latent spaces encourages smoother decision boundaries, improving generalization and resilience against statistical outliers.

Within Federated Learning, this regularization strategy can be extended to the server side to recover predictive performance compromised by sparse data or truncated local training. Through *Generative Replay*, the central server leverages the aggregated statistical properties of the global prototypes specifically their mean vectors and dynamically computed variance radii to synthesize new data points directly within the latent feature space.

This generative process simulates a broader dataset without violating client privacy constraints. By applying interpolation techniques like *Soft Mixup* to these synthetically generated latent representations, the server can refine the global model, smooth decision boundaries, and compensate for the informational sparsity inherent to extreme Non-IID data fragmentation.

Chapter 3

Related Work

This chapter reviews the literature across the intersecting domains that form the theoretical foundation of the proposed framework. To contextualize the contributions of the FedAlt and FedSpo modules, we examine the limitations of current state-of-the-art approaches. The review is structured around three primary areas: distributed optimization over statistically heterogeneous data, strategies to mitigate Catastrophic Forgetting in continual learning, and prototype-based regularization in asynchronous federated networks. Analyzing these works highlights the specific technological gaps namely, computational overhead and vulnerability to extreme data fragmentation that this thesis addresses.

3.1 Distributed Optimization and the Non-IID Challenge

Federated Learning as a decentralized optimization paradigm originates with the Federated Averaging (FedAvg) algorithm [3]. FedAvg demonstrated that deep neural networks can be trained across distributed nodes by periodically aggregating local Stochastic Gradient Descent (SGD) updates instead of centralizing raw data. However, as analyzed by Shenaj et al. [6], the theoretical convergence guarantees of FedAvg rely on the assumption that the data distributions across clients are Independent and Identically Distributed (IID). When applied to Non-IID data a standard condition in practical edge computing deployments the convergence of the global model degrades.

In a Non-IID regime, the local objective function $F_k(\mathbf{w})$ of client k diverges from the global objective function $F(\mathbf{w})$. Consequently, during local training, the model parameters converge toward local optima determined by the specific distribution of the client's private dataset. This phenomenon, known as *client drift* or weight divergence, leads the central server to aggregate divergent model updates. Aggregating these disparate vectors produces a global model with poor generalization across local distributions, undermining the collaborative efficacy of the federated process.

To mitigate client drift, several algorithmic interventions modify the local optimization process. Proximal regularization techniques, such as FedProx [2], introduce a penalty term into the local loss function to restrict the Euclidean distance between the evolving local weights and the global weights downloaded at the start of the round. Similarly, Scaffold [1] utilizes control variates to estimate and correct the variance introduced by heterogeneous client gradients during local training.

While these methods provide convergence guarantees and empirical success in synchronous federated environments, their underlying assumptions do not hold in Asynchronous Federated Continual Learning (AFCL). These algorithms presume a synchronous communication protocol and assume that local datasets, while heterogeneous, remain statistically stationary over time. In an asynchronous environment where client participation is variable and data arrives as a sequence of distinct tasks, historical gradient corrections and static proximal constraints are insufficient. Applying static weight-space regularization to a model continually adapting to new tasks can introduce optimization conflicts and destabilize the global model. The limitations of traditional weight-space penalization under asynchronous and continual constraints motivate the use of prototype-based latent space regularization.

3.2 Continual Learning Strategies in Centralized and Distributed Environments

Continual Learning (CL) involves designing systems capable of accumulating knowledge over a continuous stream of experiences. As documented by Wang et al. [9], the primary impediment to robust continual learning in deep neural networks is Catastrophic Forgetting. When a neural network is trained on a novel sequence of data representing a new task, the backpropagation algorithm updates the synaptic weights to minimize the empirical risk on the current distribution. This process frequently overwrites the parameter configurations that encoded the representational knowledge of previously learned tasks, leading to performance degradation on historical data.

Navigating this conflict requires algorithms to balance the plasticity-stability dilemma: the system must exhibit sufficient plasticity to integrate novel information while maintaining enough stability to preserve the decision boundaries of previously consolidated classes.

Within centralized machine learning literature, approaches to address Catastrophic Forgetting are broadly classified into three primary categories: regularization-based methods, architecture-based methods, and memory-based rehearsal strategies.

Regularization approaches, such as Elastic Weight Consolidation (EWC) or Learning without Forgetting (LWF) [10, 11], attempt to preserve historical knowledge by introducing auxiliary penalty terms into the loss function. These terms constrain the updating of specific neural weights identified as crucial for past tasks, or enforce similarity between the output probabilities of the historical and current models via knowledge distillation. Architecture-based methods, conversely, address forgetting by dynamically expanding network capacity, allocating isolated sub-networks or pathways for each incoming task. While effective in certain scenarios, both regularization and architectural methods struggle to scale efficiently with a large number of tasks, often encountering rigid parameter spaces or significant memory overhead.

Consequently, memory-based strategies commonly referred to as rehearsal or replay methods have consistently demonstrated high empirical efficacy in centralized CL benchmarks. These methodologies maintain a bounded episodic memory buffer containing a representative "coreset" of historical data samples. During the training of a new task, these historical samples are retrieved and interleaved with the incoming data stream, anchoring the network's decision boundaries and guiding the optimization process to satisfy past and present objectives simultaneously.

However, deploying these memory-based paradigms in a Federated Learning ecosystem encounters a strict theoretical and practical constraint: data privacy. Federated architectures explicitly prohibit the centralized storage, transmission, or cross-client sharing of raw data samples. Therefore, maintaining a rehearsal buffer containing data from multiple clients constitutes a violation of the privacy protocols (such as GDPR) that federated learning is designed to uphold.

To circumvent the prohibition on raw data sharing, recent literature has explored local *Generative Replay*. In these frameworks, clients train generative models, such as Generative Adversarial Networks (GANs) or Variational Autoencoders (VAEs), alongside their primary classification networks. The objective is to synthesize pseudo-samples of historical tasks locally, enabling rehearsal without retaining original data. Unfortunately, this approach introduces significant computational overhead. Edge devices, characterized by limitations in processing power, memory, and battery life, are ill-equipped to handle the continuous optimization of generative architectures. Furthermore, training GANs on the highly skewed, sparse local partitions typical of Non-IID federated environments frequently results in mode collapse, yielding synthetic data that is ineffective for regularization.

These limitations of traditional and generative continual learning techniques in distributed environments motivate an alternative approach. This gap in the literature informs the architectural design of our hybrid framework, wherein the computational burden of generative replay is offloaded to the central server and executed exclusively within the privacy-preserving latent feature space.

3.3 Prototype-based Federated Learning and the Limitations of FedSpace

Introducing prototype-based methodologies into Federated Learning provides a structural alternative to the standard aggregation of synaptic weights. Rather than constraining the parameter space directly, these approaches shift the focus of regularization to the latent feature space. Early works, such as FedProto [**tan˙fedproto**], demonstrated the effectiveness of representing each semantic class as a global prototype, defined as the centroid derived from the compressed representations of all samples belonging to a specific category.

Building upon this foundation, the FedSpace framework [7] extended prototype-based regularization to address the specific challenges of Asynchronous Federated Continual Learning (AFCL), where independent clients learn disjointed tasks during uncoordinated temporal windows. In FedSpace, during local training, active nodes compute the geometric distance between the latent embeddings of their private data and the corresponding global prototypes broadcasted by the server. By applying a distance-based loss function, the local network aligns its internal representations with these global anchors. This geometric regularization mitigates Catastrophic Forgetting: by anchoring the spatial coordinates of centroids related to previous tasks, the network preserves its historical discriminative capabilities without retaining or sharing raw data.

Despite its effectiveness in preserving accuracy, FedSpace exhibits structural limitations that hinder its applicability in large-scale edge deployments.

The first limitation is computational overhead. Achieving convergence with the prototype-based loss function requires a slow alignment process. To satisfy the constraints imposed by the global centroids, FedSpace requires participating clients to execute a fixed number of local training epochs during each asynchronous interaction. This static training protocol increases energy consumption and communication latency, critical constraints for battery-powered edge devices. Furthermore, the fixed optimization schedule fails to account for the convergence already achieved by individual local models, leading to redundant computations.

The second limitation emerges under high network scalability, specifically in scenarios with severe statistical fragmentation. As the federated network expands (e.g., to 500 independent clients), data partitioning becomes increasingly sparse. Consequently, individual nodes often possess an insufficient number of samples to compute statistically reliable local prototypes. Under these regimes, the generated local prototypes exhibit high variance and fail to accurately reflect the true latent distribution of the class. When the central server aggregates these representations, the resulting global prototypes become noisy anchors. Rather than guiding the subsequent training of other clients, these high-variance centroids destabilize the optimization process and degrade global classification accuracy.

These structural limitations define the operational focus of our research. The need to reduce the computational burden on edge clients motivates the integration of the adaptive early-stopping logic (FedAlt). Concurrently, the vulnerability of prototype aggregation to statistical noise necessitates the design of the server-side tuning module (FedSpo). By delegating the computational responsibility of filtering high-variance prototypes and reconstructing decision boundaries via latent interpolation (Soft Mixup [8, 12]) to the server, the proposed framework addresses both the efficiency constraints and the accuracy limitations of current methodologies.

Chapter 4

Proposed Method

4.1 Introduction to the Proposed Framework

Addressing computational inefficiencies and data fragmentation in Asynchronous Federated Continual Learning (AFCL) requires moving beyond static optimization paradigms. In this chapter, we present a hybrid algorithmic framework designed to decouple the computational load from the preservation of global model accuracy.

The proposed methodology addresses the limitations of baseline frameworks such as FedSpace, which rely on a fixed, predefined number of local training epochs (E) across all clients. While prototype-based geometric regularization mitigates Catastrophic Forgetting, applying a static local training schedule to clients with skewed Non-IID data exacerbates local overfitting, increases client drift, and generates unnecessary computational overhead. To resolve this inefficiency, we introduce an architecture composed of two complementary modules. On the client side, the Federated Adaptive Local Training (FedAlt) module dynamically terminates the local optimization process to prevent parametric divergence. On the server side, the Federated Server-Side Prototype Optimization (FedSpo) module utilizes generative replay to refine decision boundaries and recover global model accuracy.

By redistributing computational complexity across the network topology, this architecture allows resource-constrained edge devices to execute only the necessary gradient updates, reducing energy consumption and communication latency. Concurrently, the central server assumes an active optimization role rather than acting solely as a parameter aggregator. Utilizing the FedSpo module, the server leverages aggregated global prototypes to approximate the underlying data distribution via synthetic latent generation, performing a targeted refinement phase to enhance global decision boundaries.

The remainder of this chapter is structured as follows. Section 4.2 provides the mathematical formulation of the asynchronous system model. Section 4.3 details the similarity-based control mechanisms implemented on the edge devices, while Section 4.4 describes the latent synthesis strategies executed by the central server to compensate for the truncated client-side training. Finally, Section 4.5 outlines the fragmentation-aware tuning and regularization strategies across varying degrees of statistical variance.

4.2 Mathematical Formulation and System Model

We model the federated ecosystem as an asynchronous star-graph topology comprising a central aggregation server $\mathcal{S}$ and a set of heterogeneous peripheral nodes. Let $\mathcal{K} = \{1, 2, \ldots, K\}$ denote the set of participating edge devices. In our experimental evaluation, the total number of clients is set to $K \in \{50, 100, 500\}$ to simulate varying levels of data fragmentation.

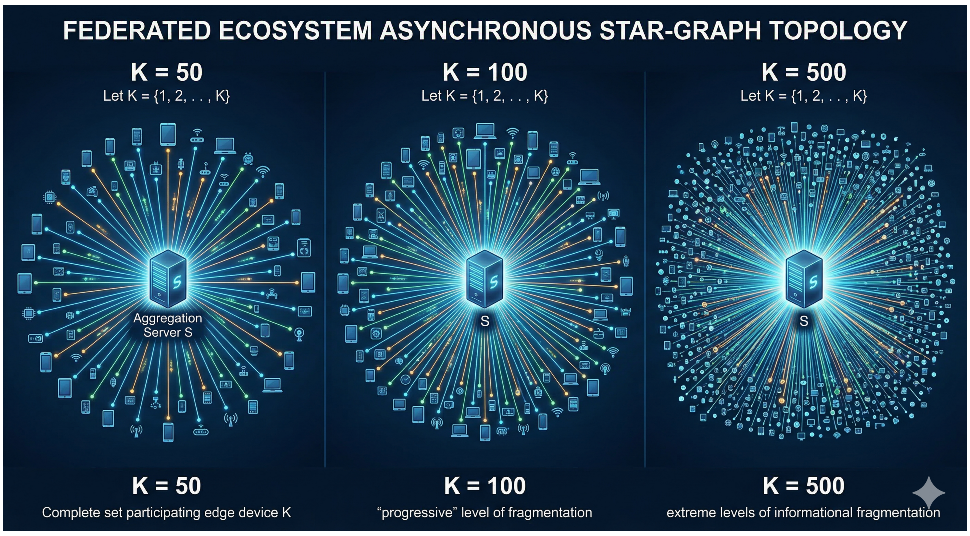

Figure 4.1: System Model Topology. *Source: AI-generated*

Each client $k \in \mathcal{K}$ holds a private dataset $\mathcal{D}_k = \{(x_i, y_i)\}_{i=1}^{N_k}$, consisting of N_k raw input samples x_i and their corresponding ground-truth labels y_i. This local dataset follows a statistical distribution $P_k(x, y)$ that is skewed relative to the global distribution $P(x, y)$, establishing a Non-Independent and Identically Distributed (Non-IID) scenario characterized by label distribution skew.

Unlike static federated learning approaches where the entire local dataset is available before optimization begins, our system model adheres to the Continual Learning paradigm. The data on the peripheral clients arrives as a continuous stream of distinct learning tasks $\mathcal{T}_k = \{T_{k,1}, T_{k,2}, \ldots\}$. Each new task introduces data samples from semantic classes previously unseen by the local model. The objective of each node is to collaboratively train a neural network, parameterized by weights $\mathbf{w} \in R^d$, to classify novel samples while preventing the parameters associated with historical classes from being overwritten by new gradient updates (Catastrophic Forgetting).

The communication protocol between the clients and the server operates asynchronously, avoiding synchronized global rounds. Based on resource availability, an active client k initiates a downlink transmission to retrieve the current global model parameters $\mathbf{w}_g$, the global class prototypes $\mathcal{P}_g$, and their corresponding global radii $\mathcal{R}_g$ from the central server. These components are required to compute the prototype-based regularization penalties during local training. The global prototypes $\mathcal{P}_g$ serve as spatial anchors to align the local feature extractor, while the radii $\mathcal{R}_g$ define the variance used to augment these prototypes and represent past data distributions in the latent space.

The client then initiates local training to minimize a composite objective function. This objective combines the standard Cross-Entropy loss $\mathcal{L}_{CE}$ for newly observed classes with prototype-based regularization penalties. Following the FedSpace architecture [7], the local optimization employs a representation loss $\mathcal{L}_{repr}$ and a prototype augmentation loss $\mathcal{L}_{aug}$. These terms constrain the latent feature embeddings of local samples to align with their global centroids. To balance the acquisition of new features with the retention of historical representations, the weighting coefficients for both prototype-based loss terms are set to $\lambda_{repr\_loss} = 0.01$ and $\lambda_{proto\_aug} = 0.01$.

Upon the dynamic termination of the local optimization phase governed by the FedAlt mechanism the client transmits its updated local weights $\mathbf{w}_k$, along with its recalculated local prototypes and variance metrics, back to the server. This asynchronous uplink exposes the global aggregation algorithm to the standard optimization challenges of delayed updates.

4.3 Client-Side Optimization via Federated Adaptive Local Training (FedAlt)

Standard Federated Learning algorithms, including the FedSpace baseline, rely on a fixed hyperparameter E, representing the exact number of local training epochs each selected client must perform before transmitting updates to the central server. While this homogeneous approach simplifies convergence analysis in controlled environments, it introduces significant computational inefficiencies in Asynchronous

Federated Continual Learning (AFCL) ecosystems.

Edge devices possess varying computational capabilities and highly heterogeneous Non-IID data distributions. Forcing a client with a skewed local dataset to execute a predefined number of optimization steps pushes its parameter space toward a local optimum. This divergence from the global consensus (client drift) degrades the quality of server-side aggregation, increases energy consumption, and exacerbates communication latency.

To address this computational inefficiency and mitigate parametric divergence, our framework integrates the Federated Adaptive Local Training (FedAlt) mechanism [5]. The specific methodological contribution of this section consists of adapting this dynamic early-stopping logic to operate within an asynchronous, prototype-based continual learning ecosystem. Like the original FedAlt formulation, our module evaluates representational divergence directly within the latent feature space. However, we specifically repurpose this feature-level similarity metric to serve the geometric objectives of FedSpace. By continuously monitoring the divergence of the extracted features, the module halts local training before the client-specific updates can disrupt the spatial alignment of the global prototypes, effectively preserving the structural knowledge of previous tasks without requiring a predefined epoch limit.

This architectural design reduces computational overhead and aligns the stopping criterion with the core objective of the prototype framework: preserving the spatial coordinates of previously learned classes to mitigate Catastrophic Forgetting. By transforming the local optimization phase into a feature-aware, self-regulating process, the system explicitly controls the trade-off between local task adaptation and global prototype stability.

The adapted FedAlt module evaluates the optimization trajectory in real time during the local training loop. Upon initialization, the client retains a frozen snapshot of the global model, serving as a geometric anchor. As the local model iteratively evolves via backpropagation to minimize the composite loss, the FedAlt controller calculates representational divergence dynamically at the batch level, rather than waiting for the conclusion of a full epoch.

Specifically, for a given batch of input data x, let $F(\mathbf{w}_k, x)$ represent the high-dimensional feature embeddings extracted by the active local encoder, and let $F(\mathbf{w}_g, x)$ represent the embeddings extracted by the frozen global encoder. Representational alignment is quantified using Cosine Similarity computed across the feature dimension:

$$\mathrm{Sim}(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{F(\mathbf{w}_k, x_i) \cdot F(\mathbf{w}_g, x_i)}{\|F(\mathbf{w}_k, x_i)\| \|F(\mathbf{w}_g, x_i)\|} \tag{4.1}$$

As local training progresses, the model adapts to the local Non-IID data, causing the latent feature embeddings to shift away from the global spatial anchors. This results in a monotonic decrease in Cosine Similarity. The client evaluates this metric after every optimization step. Local optimization terminates breaking both the batch and epoch loops when this similarity falls below a dynamically calculated tolerance threshold.

The tolerance threshold governing this early-stopping mechanism is designed to balance exploration and exploitation. Let R represent the total anticipated duration of the federated learning process in communication rounds, and let r denote the current round index. The dynamic threshold $T_h(r)$ is formulated as a strictly increasing linear function:

$$T_h(r) = a + b \cdot \frac{r}{R} \tag{4.2}$$

Based on empirical validation, the intercept parameter is set to $a = 0.1$, and the slope coefficient is defined as $b = 0.8$. This parameterization yields a permissive threshold (0.1) during the early stages of global training, linearly tightening to a restrictive value (0.9) as the system approaches its final rounds.

This linear progression aligns with the optimization dynamics of sequential learning. During the initial rounds ($r \ll R$), the global model possesses limited consolidated knowledge. The threshold remains low to encourage local exploration, allowing clients to execute more steps to construct foundational representations of their initial tasks. Conversely, during advanced stages ($r \approx R$), the global model has consolidated a robust prototype manifold. The threshold is elevated to enforce strict geometric compliance. Clients are permitted to execute only constrained gradient updates, ensuring that the assimilation of new data does not perturb established global decision boundaries. Integrating this latent-space formulation minimizes redundant computations while maintaining the geometric fidelity required for the subsequent FedSpo refinement phase.

4.4 Server-Side Accuracy Recovery via Federated Server-Side Prototype Optimization (FedSpo)

While the AFCL environment inherently struggles to form robust decision boundaries due to highly skewed Non-IID data distributions, the integration of the FedAlt module introduces a specific computational trade-off. Dynamically truncating the local stochastic gradient descent process effectively minimizes computational overhead and mitigates severe client drift; however, as demonstrated in our empirical evaluation, it also prevents the peripheral models from fully assimilating their localized tasks. Consequently, the aggregated global model exhibits a further degradation in classification accuracy compared to exhaustive local training baselines. To recover this performance and overcome the inherent data sparsity without violating privacy constraints, our framework introduces the Federated Server-Side Prototype Optimization (FedSpo) module.

FedSpo allows the central server to actively participate in the optimization process. Utilizing the aggregated statistical properties of the global class prototypes, the server executes a privacy-preserving generative replay strategy confined entirely within the latent feature space.

The functional mechanism of FedSpo assumes that global class prototypes and their corresponding variance metrics adequately approximate the underlying topological structure of the data distribution. During the asynchronous uplink phase, alongside updated model weights, active clients transmit their locally computed class centroids and the spatial dispersion of their local latent embeddings (local class radius). The server aggregates these metrics to establish a global prototype vector, denoted as $\mu_c \in R^d$, and a corresponding global scalar radius, $r_c \in R$, for every observed semantic class $c \in \mathcal{C}$.

These aggregated parameters define a high-dimensional bounding sphere for each category. To construct a balanced surrogate dataset on the server, FedSpo employs Multivariate Gaussian sampling to synthesize pseudo-representations. The formulation for generating a synthetic latent feature vector f_{syn} is defined as:

$$f_{syn} = \mu_c + z \cdot r_c \tag{4.3}$$

where $z \sim \mathcal{N}(0, I)$ is a multidimensional Gaussian noise vector sampled from a standard normal distribution. This stochastic perturbation scales the normalized noise z by the global radius r_c, anchoring it to the global centroid μ_c.

To execute server-side refinement, the generative engine populates a balanced, Independent and Identically Distributed (IID) synthetic dataset. Based on empirical evaluation, FedSpo synthesizes 20 distinct latent samples per class during each server-side tuning epoch. This provides a statistically sufficient approximation of the latent manifold without incurring excessive computational overhead.

During this refinement phase, the feature extractor (encoder) of the global model is strictly frozen. The synthetic pseudo-features pass exclusively through the fully connected classification head, which acts as a linear decoder mapping the latent representations to the semantic label space. By calculating the standard Cross-

Entropy loss over these balanced representations and executing backpropagation solely on the classifier's weights utilizing Stochastic Gradient Descent (SGD) with a Cosine Annealing learning rate scheduler the server recalibrates the decision boundaries. This architectural intervention corrects the class imbalances introduced by asynchronous Non-IID updates and recovers the classification accuracy sacrificed by client-side acceleration, effectively decoupling local efficiency from global predictive performance.

4.5 Fragmentation-Aware Tuning and Regularization Strategies

The effectiveness of the Generative Replay mechanism executed by the FedSpo module depends on the reliability of the aggregated global prototypes. Synthesizing latent representations from high-variance class centroids can introduce noise into the global model, degrading the decision boundaries.

In federated scenarios with a moderate number of participants, aggregating local extractions generally produces a reliable estimation of the latent manifold. However, as the network scales (e.g., from 50 to 500 nodes), local data partitions become increasingly sparse. Under these Non-IID conditions, clients frequently lack sufficient samples to compute robust local centroids, resulting in the transmission of high-variance prototype estimations. To stabilize the FedSpo generative engine against this variance, our framework introduces a fragmentation-aware optimization pipeline comprising Radius-Based Quality Filtering and Latent Soft Mixup regularization.

To mitigate prototype noise, we implement a dynamic, threshold-based filtering algorithm that evaluates the spatial reliability of aggregated class representations. During server-side aggregation, the system calculates the global mean radius, $\bar{R}$, across all valid classes. This scalar estimates the average intra-class variance within the global latent space. Subsequently, a dynamic threshold R_{th} is formulated as:

$$R_{th} = \beta \bar{R} \tag{4.4}$$

where β is a multiplier hyperparameter dependent on the fragmentation level. Any semantic class c with an aggregated global radius r_c exceeding this threshold ($r_c > R_{th}$) is classified as an outlier and excluded from the generative replay phase.

The calibration of β adapts to the network scale. In a medium fragmentation scenario (50 clients), the data remains sufficiently representative; thus, we set $\beta = 1.2$. This establishes a 20% tolerance margin above the global average, excluding extreme outliers while preserving the majority of the latent structural information.

Conversely, in an extreme fragmentation scenario (500 clients), high variance becomes a significant factor. To stabilize the system, we enforce a strict filtering regime by setting $\beta = 1.0$. This ensures that only prototypes with an intra-class variance less than or equal to the global average generate synthetic data. To further suppress oscillations caused by the asynchronous uplink of sparse clients, this filtering is coupled with a low Exponential Moving Average (EMA) momentum coefficient (0.05), ensuring stable geometric evolution of the global prototypes.

Applying strict filtering in the 500-client scenario introduces a secondary challenge: discarding a large percentage of high-variance prototypes creates sparse regions in the represented class space. To bridge these gaps and construct generalizable hyperplanes, we extend the FedSpo module by integrating a Soft Mixup regularization strategy directly within the generative latent space.

Following the principles of Manifold Mixup [8], the central server interpolates between synthetic samples generated from the retained prototypes. Let f_i and f_j represent two distinct synthetic latent features generated during the FedSpo replay phase. The server synthesizes a mixed latent representation $\tilde{f}$ through the convex combination:

$$\tilde{f} = \lambda f_i + (1 - \lambda) f_j \tag{4.5}$$

where the interpolation coefficient λ is sampled from a Beta distribution, $\lambda \sim \text{Beta}(\alpha, \alpha)$.

Based on empirical tuning, robustness in the extreme fragmentation scenario is optimized by setting the concentration parameter to $\alpha = 0.4$. This sub-unity parameterization yields a U-shaped probability density function, ensuring that the synthetic representations preserve the primary semantic identity of one class while injecting a regularizing perturbation from the secondary class. By optimizing the global classifier across these interpolated synthetic latent spaces, the framework mitigates the high variance inherent to large-scale AFCL environments, recovering classification accuracy under severe data fragmentation.

Chapter 5

Experimental Results

5.1 Experimental Setup and Dataset Partitioning

To evaluate the proposed hybrid architecture, we utilize a simulation environment designed to replicate an Asynchronous Federated Continual Learning (AFCL) deployment. The evaluation is conducted on the CIFAR-100 dataset, comprising 100 distinct semantic classes. While CIFAR-100 is a standard baseline within the Continual Learning literature, its structure provides a controlled environment to evaluate Catastrophic Forgetting and representational stability, allowing for the computationally feasible simulation of large-scale federated networks (up to 500 nodes). To adapt the dataset for sequential learning, the classes are temporally divided into a sequence of disjoint tasks. Each task introduces novel categories, requiring the global model to expand its decision boundaries without overwriting parameters associated with prior knowledge.

To generate heterogeneous data partitions across edge devices and enforce a Non-Independent and Identically Distributed (Non-IID) scenario, local datasets are sampled using a Dirichlet distribution. The Dirichlet concentration parameter controls the degree of label imbalance and data sparsity experienced by each node. A lower concentration parameter yields a skewed distribution, where clients possess abundant examples for a small subset of classes while lacking data for the majority.

This partitioning ensures that local gradients computed during asynchronous training are highly divergent, providing a rigorous evaluation environment for the FedAlt early-stopping logic and the FedSpo generative refinement module. This Dirichlet partitioning scheme is applied across three network scales to simulate varying levels of data sparsity: medium fragmentation (50 clients), high fragmentation (100 clients), and extreme fragmentation (500 independent nodes).

5.2 Baseline Architecture and Evaluation Metrics

The proposed hybrid framework is benchmarked against FedSpace [7], the state-of-the-art baseline for prototype-based regularization in AFCL environments. While FedSpace enforces geometric alignment between local features and global centroids, it relies on a static local optimization schedule.

To evaluate the dynamic early-stopping logic of the FedAlt module, the upper bound for local training epochs must be sufficiently large. If the maximum number of local epochs were restricted to one, the adaptive similarity thresholding would not trigger. Therefore, across all experiments for both the baseline and the proposed framework, the maximum local training duration is set to $E = 5$ epochs. This configuration isolates the impact of FedAlt's early termination and the accuracy recovery provided by FedSpo.

The primary evaluation metric is the Final Average Accuracy, computed after the model has processed all tasks. It represents the mean classification accuracy across all learned tasks on a globally balanced, held-out test set. Additionally, Total Execution Time is measured to assess the computational efficiency of the system. The specific server-side tuning hyperparameters applied by the FedSpo module across the varying fragmentation regimes are summarized in Table 5.1.

Regime	**Tuning Epochs**	**Samples / Class**	**Learning Rate**	**Filter β**	**Mixup α**
50 Clients	5	20	0.01	1.2 (Quality)	0.0 (None)
100 Clients	5	20	0.001	None (Disabled)	0.0 (None)
500 Clients	5	20	0.01	1.0 (Strict)	0.4

Table 5.1: Optimal FedSpo server-side tuning hyperparameters applied across varying fragmentation regimes.

5.3 Empirical Evaluation in Medium and High Fragmentation Regimes

The evaluation begins with the medium fragmentation scenario (50 clients). In this configuration, the degree of data sparsity is moderate; clients generally possess sufficient samples to compute stable local class centroids.

Running the FedSpace baseline for $E = 5$ local epochs required a total execution time of 45h 30m 23s, yielding a final average accuracy of 35.30%. Deploying only the FedAlt module on the clients reduced the execution time to 12h 54m 32s a computational reduction of approximately 72% but resulted in an accuracy drop to 34.37%.

To recover this precision, the complete FedAlt + FedSpo architecture was deployed. The server-side tuning was configured with a Quality Filter multiplier $\beta = 1.2$, an optimization learning rate of 0.01, 5 tuning epochs, 20 synthetic samples per class, and no Mixup ($\alpha = 0$). This combined framework required 13h 30m 55s to complete, achieving a final average accuracy of 36.51%. This result demonstrates that dynamically truncating the local optimization trajectory, coupled with variance-filtered server-side generative tuning, yields significant computational savings and produces decision boundaries more robust than static local training.

Figure 5.1: Performance evaluation for the 50-client medium fragmentation regime.

In the high fragmentation regime, the network is scaled to 100 independent nodes. As the CIFAR-100 dataset is partitioned across double the edge devices, local data partitions become significantly sparser. In this environment, the FedSpace baseline (5 epochs) completed in 35h 25m 00s, but its accuracy degraded to 29.60%, indicating susceptibility to local overfitting caused by the static epoch schedule.

Applying the FedAlt module independently reduced the execution time to 09h 49m 06s, while improving accuracy to 31.50%. This demonstrates that early-stopping intrinsically acts as a regularizer against high-variance data partitions. Deploying the complete FEDALT + FEDSPO framework required a specific adaptation of the filtering strategy. Empirical observations revealed that applying a rigid variance threshold at 100 clients systematically eliminated minority class information due to the specific statistical skew of this partition. Consequently, radius-based filtering was explicitly disabled (β = None). Configured with 20 synthetic samples, a learning rate of 0.001, 5 tuning epochs, and $\alpha = 0$, the full system completed in 08h 39m 34s, elevating the final accuracy to 32.41%.

Figure 5.2: Performance evaluation for the 100-client high fragmentation regime.

5.4 Performance in Extreme Fragmentation Regimes and the Computational Trade-off

The extreme fragmentation regime scales the topology to 500 peripheral clients. Within this environment, the Dirichlet partitioning results in highly sparse data silos. The limited samples available per class yield local prototypes that exhibit high variance across communication rounds.

Under these conditions, the FEDSPACE baseline (5 epochs) required 25h 16m 33s, achieving a final accuracy of 34.80%. Deploying solely the FEDALT module reduced the execution time to 07h 39m 03s; however, due to the extreme prototype noise, the unrefined global model suffered an accuracy drop, falling to 30.90%.

Navigating this environment requires a strict configuration of the FEDSPO module. The server enforced a Strict Filtering regime ($\beta = 1.0$) to exclude any prototype exhibiting above-average variance. To bridge the resulting sparse regions in the latent space, the Soft Mixup strategy was activated with a concentration parameter of $\alpha = 0.4$. Configured with a learning rate of 0.01, 20 samples, and 5 tuning epochs, the complete FedAlt + FedSpo architecture finished the federated process in 07h 10m 17s, recovering the final average accuracy to 34.04%.

The empirical results obtained under this configuration illustrate the computational trade-off. While the proposed framework exhibits a marginal accuracy degradation of approximately 0.76% relative to the static FedSpace baseline in the 500-client scenario (34.04% vs. 34.80%), this minor concession in predictive precision is offset by significant system-wide efficiency gains.

By allowing 500 clients to dynamically truncate their local optimization based on cosine similarity, the overall computational execution time was reduced by over 70% (from over 25 hours to just over 7 hours). This methodology demonstrates that intelligently shifting the regularization burden from resource-constrained edge devices to the central server's generative engine can sustain highly competitive continual learning performance while preserving the operational viability and energy efficiency of the federated edge ecosystem.

Figure 5.3: Performance evaluation for the 500-client extreme fragmentation regime.

5.5 Summary of Experimental Results

Table 5.2 summarizes the experimental results across all fragmentation regimes, highlighting the impact of the proposed modules on execution time and classification accuracy. The data demonstrates that in the 50 and 100 client scenarios, the complete FedAlt + FedSpo architecture outperforms the FedSpace baseline in both predictive accuracy and computational efficiency. Furthermore, under extreme fragmentation (500 clients), the hybrid framework maintains highly competitive accuracy exhibiting only a marginal 0.76% degradation while achieving a substantial 70% reduction in total execution time.

Fragmentation Level	Algorithm	Total Exec. Time	Accuracy (%)
	FedSpace (Baseline)	45h 30m 23s	35.30
Medium (50 Clients)	FedAlt (Adaptive Only)	12h 54m 32s	34.37
	FedAlt + FedSpo	13h 30m 55s	**36.51**
	FedSpace (Baseline)	35h 25m 00s	29.60
High (100 Clients)	FedAlt (Adaptive Only)	09h 49m 06s	31.50
	FedAlt + FedSpo	08h 39m 34s	**32.41**
	FedSpace (Baseline)	25h 16m 33s	**34.80**
Extreme (500 Clients)	FedAlt (Adaptive Only)	07h 39m 03s	30.90
	FedAlt + FedSpo	07h 10m 17s	34.04

Table 5.2: Comparative analysis of Total Execution Time and Final Average Accuracy across different fragmentation regimes on the CIFAR-100 dataset. All experiments strictly enforce a maximum local training bound of $E = 5$ epochs. The best accuracy per scenario is highlighted in bold.

Chapter 6

Conclusions and Future Work

6.1 Summary of Methodological Contributions

The objective of this thesis was to address the computational bottlenecks and structural limitations of prototype-based regularization in Asynchronous Federated Continual Learning (AFCL) environments. Training deep neural networks collaboratively on continuous, non-stationary data streams without centralizing raw data remains a primary challenge in edge computing.

While frameworks such as FedSpace mitigate Catastrophic Forgetting through geometric latent space alignment, their reliance on static local training protocols makes them computationally demanding for resource-constrained edge ecosystems. To overcome these limitations, this research introduced a dual-module architecture designed to decouple the computational burden on peripheral devices from the preservation of global model accuracy.

The first methodological contribution is the adaptation and integration of the Federated Adaptive Local Training (FedAlt) module. By establishing a dynamic control mechanism based on the cosine similarity between the evolving local feature embeddings and the frozen global representations at the batch level, the framework enables feature-aware early stopping. The implementation of a dynamically adjusted similarity threshold allows participating nodes to terminate local training when client drift begins to degrade global alignment. This adaptive logic reduces redundant computations, lowering energy consumption and communication latency.

Complementing this client-side efficiency, the second contribution is the Federated Server-Side Prototype Optimization (FedSpo) module. Because truncating local training can degrade classification accuracy, FedSpo was designed to recover this performance by utilizing the central server for generative replay. By aggregating local class centroids and their spatial variance metrics, the server approximates the underlying data distribution using Multivariate Gaussian sampling within the latent feature space. This privacy-preserving strategy synthesizes a balanced set of pseudo-representations. Fine-tuning the global classification head on these synthetic latent vectors refines the decision boundaries, balancing local computational efficiency with global predictive performance.

6.2 Concluding Remarks on Experimental Findings

The empirical evaluation of the proposed hybrid methodology, conducted on a Dirichlet-partitioned CIFAR-100 dataset, confirmed the framework's effectiveness across varying degrees of network fragmentation.

In the medium fragmentation scenario (50 clients), where local datasets retained sufficient sample density, the framework achieved improved performance metrics. By deploying a calibrated Quality Filtering algorithm with a 20% tolerance margin above the global average radius ($\beta = 1.2$), the FedSpo module filtered high-variance outliers and synthesized a robust pseudo-dataset. This configuration resulted in a final average accuracy of 36.51%, outperforming the static FedSpace baseline.

In the extreme fragmentation scenario (500 nodes), localized data partitions yielded high-variance prototypes. To mitigate this statistical variance, the central server applied a strict filtering regime ($\beta = 1.0$) combined with a low Exponential Moving Average momentum, ensuring that only stable centroids were retained during the aggregation process.

To address the resulting sparsity in the latent space, the framework integrated a Soft Mixup regularization strategy, utilizing a Beta distribution concentration parameter of $\alpha = 0.4$ to interpolate between the synthetic representations. Under these extreme conditions, the architecture achieved a final average accuracy of 34.04%. While this represents a marginal accuracy decrease of 0.76% relative to the 34.80% achieved by the static baseline, this minor concession is offset by a significant reduction in computational overhead. Specifically, the proposed framework reduced the total execution time by over 70%, from more than 25 hours to approximately 7 hours.

These results demonstrate that by delegating the regularization burden to the server's generative engine, it is possible to sustain competitive continual learning performance in sparse, Non-IID environments while preserving the computational viability and energy efficiency of the federated edge ecosystem.

6.3 Limitations and Future Research Directions

While the proposed architecture effectively balances computational efficiency and continual learning performance, the current framework exhibits specific structural limitations that provide clear directions for future research.

The primary constraint of our methodology resides in the statically defined nature of the hyperparameters governing the FedSpo server-side refinement phase. As demonstrated in the experimental evaluation, optimal performance relies on the manual calibration of the Quality Filtering multiplier (β) and the Soft Mixup concentration parameter (α). In our setup, these parameters were assigned based on prior knowledge of the network's scale, deploying $\beta = 1.2$ for the 50-client scenario, and enforcing $\beta = 1.0$ alongside an $\alpha = 0.4$ Mixup interpolation for the 500-client topology. However, in a real-world asynchronous edge ecosystem, the total number of participating nodes and the exact severity of the Non-IID data distribution are rarely known *a priori* and can fluctuate dynamically.

To address this limitation, future research should focus on developing adaptive hyperparameter tuning mechanisms directly within the server's aggregation logic. One approach could leverage purely statistical heuristics; by continuously monitoring the statistical distribution of the uploaded prototype radii and the asynchronous arrival rate of client updates, the server could dynamically adjust the filtering threshold β and the Mixup coefficient α. Alternatively, the integration of Reinforcement Learning (RL) or meta-learning agents to autonomously govern these parameters will also be considered. These self-calibrating approaches would allow the system to autonomously adapt to varying degrees of data fragmentation without requiring manual intervention or violating privacy constraints.

Furthermore, future investigations should seek to expand the generative capabilities of the central server. While the Multivariate Gaussian sampling utilized in this work is computationally lightweight, it may struggle to accurately capture the highly complex, non-linear latent manifolds associated with more intricate data structures. Research should evaluate the integration of more advanced generative architectures, such as Latent Diffusion Models (LDMs) or Flow-based networks, optimized to execute entirely within the compressed feature space.

Finally, to validate the generalizability of the proposed framework, subsequent empirical studies must scale beyond the CIFAR-100 benchmark. Testing the dynamic interplay of FedAlt and FedSpo across diverse, large-scale Continual Learning datasets including sequential natural language processing tasks or continuous time-series sensor data from industrial IoT deployments will be essential to confirm the robustness of this asynchronous federated architecture.

Bibliographic references

[1] Sai Praneeth Karimireddy et al. “SCAFFOLD: Stochastic Controlled Averaging for Federated Learning”. In: *Proceedings of the 37th International Conference on Machine Learning (ICML)*. 2020.

[2] Tian Li et al. “Federated Optimization in Heterogeneous Networks”. In: *Proceedings of Machine Learning and Systems (MLSys)*. 2020.

[3] Brendan McMahan et al. “Communication-Efficient Learning of Deep Networks from Decentralized Data”. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2017.

[4] Umberto Michieli, Marco Toldo, and Pietro Zanuttigh. “Domain adaptation and continual learning in semantic segmentation”. In: *Advanced Methods and Deep Learning in Computer Vision*. Ed. by E. R. Davies and Matthew Turk. Academic Press, 2022, pp. 275–303. ISBN: 978-0-12-822109-9. DOI: `10.1016/B978-0-12-822109-9.00017-5`. URL: `https://doi.org/10.1016/B978-0-12-822109-9.00017-5`.

[5] Donald Shenaj, Eugene Belilovsky, and Pietro Zanuttigh. “Adaptive Local Training in Federated Learning”. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2025.

[6] Donald Shenaj, Giulia Rizzoli, and Pietro Zanuttigh. “Federated Learning in Computer Vision”. In: *IEEE Access* 11 (2023).

[7] Donald Shenaj et al. “Asynchronous Federated Continual Learning”. In: *arXiv preprint arXiv:2304.03626* (2023).

[8] Vikas Verma et al. “Manifold Mixup: Better Representations by Interpolating Hidden States”. In: *Proceedings of the International Conference on Machine Learning (ICML)*. 2019.

[9] Liyuan Wang et al. “A Comprehensive Survey of Continual Learning: Theory, Method and Application”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[10] Qiao Wuerkaixi et al. "AF-FCL: Cross-Task Calibration for Asynchronous Federated Continual Learning". In: *OpenReview* (2024).

[11] Jaehong Yoon et al. "Federated Continual Learning with Weighted Inter-client Transfer". In: *Proceedings of the International Conference on Machine Learning (ICML)*. 2021.

[12] Hongyi Zhang et al. "mixup: Beyond Empirical Risk Minimization". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2018.