Self-Supervised Visual Representation Learning in Distributed Systems - Generalisation Analysis and Training Framework Design

Xuanyu Chen

Lead Supervisor: A.Prof Dong Yuan

Associate Supervisor: A.Prof Wei Bao, Dr. Nan Yang

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

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

2026

STATEMENT OF ORIGINALITY

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

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

Your Name: Xuanyu Chen

Your Signature: _____

ABSTRACT

This thesis investigates distributed self-supervised learning as a paradigm for training visual representation models directly from distributed and unlabelled data. While large-scale supervised datasets have fuelled advances in visual artificial intelligence, their centralised collection and annotation are prohibitively expensive, and real-world data is inherently fragmented across edge devices, institutions, and sensors. Distributed self-supervised learning aims to harness distributed data without labels or central coordination. Achieving this goal demands solutions to several open challenges, including robustness under heterogeneous client distributions, feasibility on resource-limited devices, tolerance for the absence of a central server, and resolution of the fundamental question of whether distributed training can approach the performance of centralised training.

The thesis makes four main contributions. First, it establishes how decentralisation of training data reshapes scaling laws, proving that the compute-optimal model size decreases as data becomes more distributed, thereby explaining why lightweight models are more effective on edge devices. Second, it shows that distributed training inevitably suffers a generalisation gap compared to centralised training under equal compute, and that this gap can only be reduced by expanding data through more clients or larger local datasets. Third, it provides the first systematic theoretical analysis of distributed self-supervised learning under heterogeneous data, showing that methods based on Masked Image Modelling (MIM) are more robust than contrastive approaches, with robustness increasing alongside network connectivity. It also introduces MAR loss, a refinement of MIM loss with alignment regularisation to further improve robustness. Finally, it proposes DeNAV, a decentralised self-supervised learning framework that eliminates server dependence, integrates

a navigator algorithm for informed client selection, staleness-aware aggregation for asynchronous updates, and lightweight masked autoencoder pre-training for communication efficiency. Theoretical analysis proves its convergence and consensus guarantees, while extensive experiments confirm its superiority over existing federated self-supervised and decentralised baselines.

Through these contributions, the thesis advances distributed self-supervised visual representation learning as a feasible and effective approach for distributed systems. It shows how theoretical insights on optimal model size estimation, generalisation study, and robustness analysis can be translated into practical algorithms and frameworks, thus fulfilling the central aim of enabling large-scale learning directly from distributed visual data without requiring expert supervision.

BIOGRAPHICAL SKETCH

Author: Xuanyu Chen

Degree: Doctor of Philosophy

Date: 2026

[Redacted] [redacted]

[Redacted] [redacted]

Undergraduate and Graduate Education:

- Master of Information Technology,
  The University of Melbourne, Australia, 2021
- Bachelor of Advanced Computing (Honours),
  The Australian National University, Australia, 2019

Major: Information Technology, Artificial Intelligence

Research: Machine Learning, Self-Supervised Learning, Distributed Learning

Publications:

Authorship is in alphabetical order

1. **Chen, Xuanyu** and Yang, Nan and Wang, Shuai and Yuan, Dong. Understanding the Robustness of Distributed Self-Supervised Learning Frameworks against Non-IID Data. In *International Conference on Learning Representations (ICLR), 2026. Accepted.*

2. **Chen, Xuanyu** and Wang, Shuai and Yang, Nan and Yuan, Dong. Generalization Performance Gap Analysis between Centralized and Federated Learning: How to Bridge this Gap? In *International Conference on Artificial Intelligence and Statistics (AISTATS), 2026. Accepted.*

3. **Chen, Xuanyu** and Yang, Nan and Wang, Shuai and Yuan, Dong. Scaling Law Analysis in Federated Learning: How to Select the Optimal Model Size? In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 2026. Accepted.*

4. **Chen, Xuanyu** and Yang, Nan and Liu, Charles Z and Yuan, Dong. DeNAV: Decentralized Self-Supervised Learning with a Training Navigator. In *Machine Learning, Asian Conference on Machine Learning (ACML Journal Track), 2025. Accepted.*

5. Yang, Nan and **Chen, Xuanyu** and Liu, Charles Z, Yuan, Dong, Bao Wei, and Cui, Lizhen. FedMAE: Federated Self-Supervised Learning with One-Block Masked Auto-Encoder. In *arXiv preprint arXiv:2303.11339*, 2023.

To my parents and family, for their unwavering love and encouragement that sustained me through every stage of this journey;

To my supervisor, for his invaluable guidance and mentorship;

To my research group, for their collaboration and friendship that made this path both meaningful and rewarding.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to the University of Sydney for providing the University of Sydney International Scholarship, which enabled me to pursue my doctoral research with financial stability. I also appreciate the Faculty of Engineering for offering numerous workshops, seminars, and tutoring opportunities, which enriched my academic development and expanded my teaching experience during the course of my candidature.

I am deeply grateful to my supervisor, Associate Professor Dong Yuan, whose consistent guidance, encouragement, and critical insights shaped the direction of my research. His high standards and patient mentorship taught me how to think rigorously, write clearly, and pursue research with independence and perseverance.

I would also like to thank my collaborators and colleagues. In particular, I am grateful to Dr. Nan Yang, whose mentorship and frequent collaboration as a senior colleague significantly advanced my research, and to Shuai Wang, whose contributions to theoretical derivations were invaluable. I also thank my peers in the research group for creating a supportive environment and for the stimulating discussions that expanded my perspectives.

Beyond academia, I am fortunate to have the support of friends both in Australia and in China. Their encouragement, companionship, and understanding helped me navigate the challenges of the PhD journey and maintain balance between research and life.

Finally, I owe my deepest gratitude to my family, especially my parents, whose unconditional love and unwavering belief in me have sustained me throughout this long journey. This thesis is dedicated to them.

AUTHORSHIP ATTRIBUTION STATEMENT

Any of my own material included as part of this thesis is clearly identified in the statements below. Each statement specifies the paper status, contribution details, and co-authorship attribution.

Chapter 3 of this thesis is based on a manuscript submitted and accepted at AAAI. I designed the study, carried out all theoretical derivations and experiments, and wrote the manuscript. The derivations were conducted jointly with Shuai Wang. The co-authors of this paper are Nan Yang, Shuai Wang, and Dong Yuan, who contributed to discussions and manuscript refinement.

Chapter 4 of this thesis is based on a manuscript submitted and accepted at AISTATS 2026. I designed the study, performed all theoretical derivations and experiments, and wrote the manuscript. The derivations were conducted jointly with Shuai Wang. The co-authors of this paper are Shuai Wang, Nan Yang, and Dong Yuan, who contributed to discussions and manuscript refinement.

Chapter 5 of this thesis is based on a manuscript submitted and accepted at ICLR 2026. I designed the study, implemented the algorithm, conducted all theoretical derivations and experiments, and wrote the manuscript. The derivations were conducted jointly with Shuai Wang. The co-authors of this paper are Nan Yang, Shuai Wang, and Dong Yuan, who contributed to discussions and manuscript refinement.

Chapter 6 of this thesis is based on a manuscript submitted and accepted at ACML Journal Track. I designed the study, implemented the framework, performed all theoretical derivations and experiments, and wrote the manuscript. The co-authors of this paper are Nan Yang, Charles Z. Liu, and Dong Yuan, who contributed to discussions and manuscript refinement.

In addition to the authorship attribution statements above, in cases where I am not the corresponding author of a published item, permission to include the published material has been granted by the corresponding author.

Name: Xuanyu Chen

Signature: _____

Date: 30/09/2025

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

Name: Associate Professor Dong Yuan

Signature: _____

Date: 30/09/2025

Use of Generative AI Statement

During the preparation of this thesis, the author used ChatGPT (OpenAI GPT-5) for the purposes of enhancing writing and formatting support. This included paraphrasing, improving sentence structure, refining logical flow between sections, and generating LaTeX templates for the required preliminary pages that are outside the original thesis template (e.g., originality statement, authorship attribution, and use of generative AI statement). The author confirms that where text or formatting suggestions were provided by the tool, the content was carefully reviewed for accuracy, clarity, and appropriateness. All research ideas, mathematical derivations, experimental design, and final conclusions are entirely the author's own. The author takes full responsibility for the submitted thesis and affirms that generative AI was used only within the parameters permitted by the University of Sydney generative AI guide for researchers.

TABLE OF CONTENTS

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

LIST OF TABLES
(Continued)

Table **Page**

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

Thesis Introduction

Chapter Overview: This chapter introduces the motivation and context for this thesis. It begins by highlighting the growing importance of artificial intelligence in real-world visual applications and the central role of large-scale labelled datasets in enabling recent advances. Against this background, it discusses why directly leveraging distributed, unlabelled data is both highly valuable and practically necessary, motivating the study of distributed self-supervised learning (D-SSL). The chapter then outlines the key challenges faced by D-SSL: heterogeneity across client data distributions, limited computational and communication resources at the edge, lack of reliable central coordination, and the fundamental concern of whether distributed training can approach the performance of centralised training. In response to these challenges, the main contributions of the thesis are presented, spanning theoretical analyses of scaling laws, generalisation gaps, and robustness under heterogeneity, as well as the development of new algorithmic components and frameworks such as MAR loss and DeNAV. Finally, the chapter provides an overview of the thesis structure, which proceeds from the literature review to the four main studies and concludes with a synthesis of findings.

1.1 Background and Motivation

Artificial intelligence (AI) has become a defining technology of modern society, with applications permeating many aspects of daily life. Common examples include facial recognition on smartphones, video surveillance in public spaces, and autonomous driving assistance systems in vehicles. These applications increasingly shape how people interact with both digital services and the physical world. However, the success of such systems in real-world deployment relies heavily on access to massive, well-labelled datasets. For example, in natural language processing, large language models used by GPT-style chatbots are typically trained on hundreds of billions of tokens [7]. Similarly, in computer vision, benchmark datasets such as ImageNet contain over 14 million annotated images spanning more than 20,000 object categories.

Despite their effectiveness, constructing large-scale labelled datasets is extremely costly. The creation of ImageNet alone required more than two years of effort and an estimated financial cost of several million U.S. dollars [19]. This challenge is particularly pronounced in real-world settings, where vast amounts of raw visual data are continuously generated by devices such as mobile phones and CCTV cameras. Such data is inherently distributed across different sources, uncurated, and largely unlabelled. Converting this distributed raw data into a centralised dataset like ImageNet requires not only extensive infrastructure for data collection, storage, and maintenance, but also substantial human labour for data cleaning and expert annotation [36].

In this context, the ability to train models directly from distributed and unlabelled data is of significant practical importance. This demand has motivated the development of distributed self-supervised learning (D-SSL). By designing pseudo-tasks that automatically generate supervisory signals from raw data, D-SSL enables models to learn meaningful visual representations without relying on human annotations, while simultaneously leveraging data distributed across multiple devices and institutions [93, 174]. As a result, D-SSL offers a promising alternative to

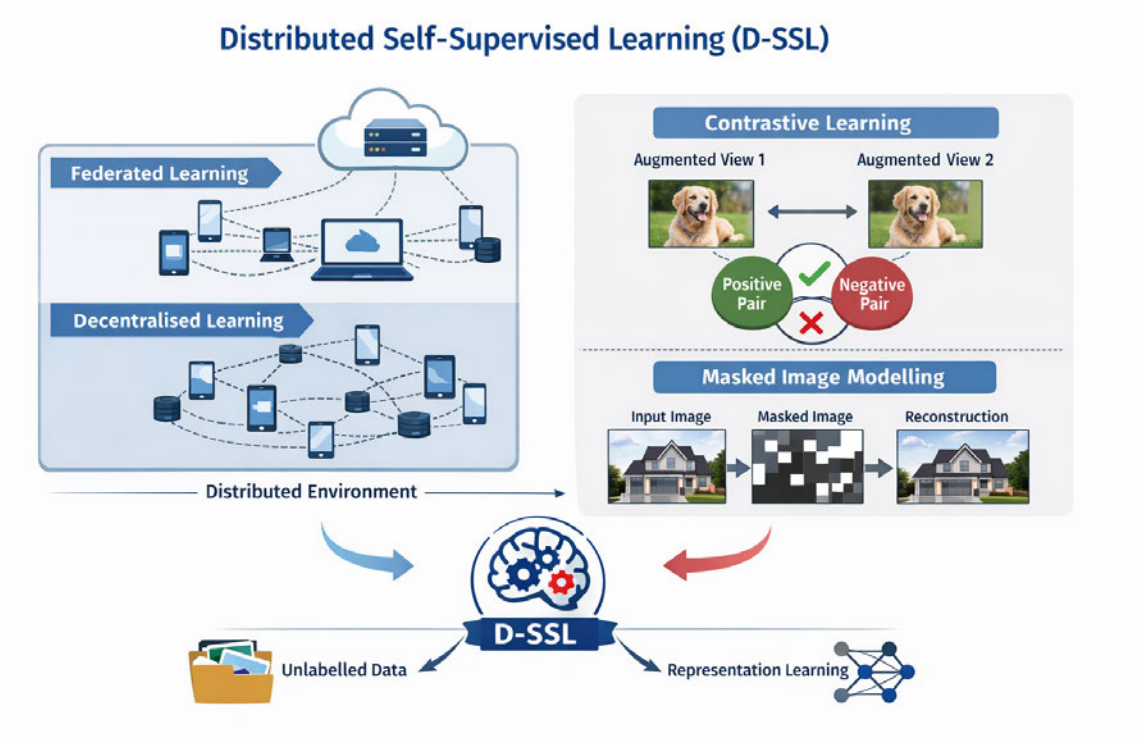


Figure 1.1: An overview of distributed self-supervised learning (D-SSL), combining distributed learning settings with self-supervised objectives.

conventional dataset construction pipelines, while also better respecting data privacy constraints and the inherently distributed nature of real-world data. Figure 1.1 provides the system-level overview of D-SSL for better illustrating this paradigm.

While D-SSL offers a compelling paradigm for learning from distributed, unlabelled data, its practical realisation remains at an early stage. In contrast to conventional self-supervised learning in centralised settings, effective D-SSL cannot be achieved by naively combining existing self-supervised objectives with distributed training frameworks. The shift from centralised to distributed learning fundamentally alters the learning dynamics, introducing new interactions between data, models, and communication. Understanding these changes, and the challenges they give rise to, is essential for assessing the robustness and scalability of D-SSL in real-world systems. These considerations motivate a closer examination of the key challenges faced by D-SSL, which are discussed in the following section.

1.2 Research Challenges

As discussed in the previous section, the transition from centralised to distributed self-supervised learning fundamentally reshapes the learning process. In practice, distributed self-supervised learning may be implemented under different architectural paradigms, most notably server-based federated frameworks [100] and fully decentralised peer-to-peer settings [131]. These environments introduce system-level and additional statistical factors that interact with self-supervised objectives in non-trivial ways, giving rise to a range of challenges hindering the development of D-SSL.

The first challenge is data heterogeneity, commonly referred to as the non-independent and identically distributed (non-IID) problem [62]. In real-world distributed systems, each client device collects data from its own environment, leading to systematic differences between local datasets. For example, in personal photo collections, one user's phone may contain mostly outdoor scenery, while another user's phone contains mostly indoor family photos. Unlike centralised datasets that are carefully curated and balanced, distributed data is inherently fragmented and biased. In self-supervised settings, where representations are learned without explicit label alignment across clients, distributional mismatch can significantly influence the consistency of learned features and lead to severe performance degradation in transferability [147]. Understanding how representation learning behaves under heterogeneous data distributions is therefore a central challenge in D-SSL.

The second challenge concerns resource limitations on edge devices. Distributed training typically relies on mobile phones, IoT cameras, or other lightweight clients with constrained computational power, memory, and communication bandwidth. These constraints limit feasible model sizes and training complexity. While modern deep learning often benefits from large-scale models and extensive computation, such scaling is not always compatible with distributed environments. This tension arises in both federated and fully decentralised settings, where local computation

and communication overhead directly determine system feasibility. Determining how model design and training strategies should adapt to distributed resource constraints remains an important open question [79, 152].

The third challenge relates to coordination under different distributed architectures [130, 162]. Federated learning assumes the presence of a central server to aggregate and redistribute model updates, providing a global synchronisation mechanism. Fully decentralised approaches, in contrast, rely on peer-to-peer communication without central coordination. These architectural differences lead to distinct trade-offs in stability, convergence behaviour, and communication efficiency. Understanding how learning dynamics differ across these distributed frameworks, and how architectural choices influence robustness under heterogeneous data, is thus necessary for designing more effective D-SSL methods.

Furthermore, beyond these technical considerations, a fundamental concern remains unresolved: can distributed self-supervised training ever reach the same level of performance as centralised training? This issue is of great importance for practitioners and organisations. Although the cost of building and maintaining centralised datasets is prohibitively high, decision-makers still need to understand whether distributed learning has the potential to achieve comparable generalisation performance. If the gap is inevitable, then distributed approaches may be viewed as secondary alternatives; whereas if the gap can be narrowed, distributed learning becomes a strong and viable choice. Establishing the theoretical nature of this performance gap and exploring strategies to mitigate it are therefore also critical to the credibility and long-term adoption of D-SSL.

1.3 Thesis Contributions

To address the above challenges, this thesis makes four major contributions that collectively advance the theoretical understanding of D-SSL and the practical development of distributed self-supervised learning in the visual domain.

- First, for the challenge of limited edge resources, the thesis establishes a theoretical foundation for how distributed environments reshape scaling laws. By deriving closed-form solutions for compute-optimal model size, it demonstrates that data decentralisation favours smaller models. This finding explains why lightweight architectures are better suited for edge-constrained clients and provides guidance for designing models that operate effectively under distributed constraints.

- Second, to clarify the fundamental disparity between centralised and distributed training, the thesis formally characterises the generalisation gap between the two. Through PAC-Bayesian analysis, it proves that this gap inevitably exists under equal compute budgets and cannot be eliminated by scaling up model size or communication rounds. The analysis shows that only allowing data advantage to distributed scenarios, either by adding more clients or enlarging local datasets, can narrow the gap, offering theoretical clarity and practical strategies for improving distributed training.

- Third, in response to the challenge of heterogeneity, the thesis provides the first systematic theoretical analysis of D-SSL under non-IID client data. It shows that self-supervised learning (SSL) based on masked image modelling (MIM) is inherently more robust to heterogeneity than SSL based on contrastive learning (CL) and that robustness improves with stronger network connectivity. Building on this insight, the thesis introduces MAR loss, a refinement of MIM loss with alignment regularisation, which enhances robustness and can be readily integrated into existing distributed MIM frameworks.

- Finally, to overcome the challenge of server-free coordination, the thesis proposes DeNAV, a decentralised self-supervised framework that combines a navigator algorithm for client selection, staleness-aware aggregation for asynchronous updates, and lightweight masked autoencoder pre-training for communication efficiency. DeNAV also supports parallel training with weight sharing for

scalability, and its theoretical analysis proves convergence, consensus, and justifies the training design. Extensive experiments confirm its superiority over prior distributed baselines. This contribution not only addresses the coordination challenge but also fulfils the thesis's central aim of establishing D-SSL as a feasible and effective paradigm.

1.4 Overview of Thesis Structure

The remainder of this thesis is structured as follows. Chapter 2 reviews the related literature on self-supervised learning, distributed training, and their intersection in D-SSL. Chapter 3 studies scaling laws under distributed settings, while Chapter 4 investigates the generalisation performance gap between centralised and distributed training. Chapter 5 extends the analysis to D-SSL under heterogeneous data and introduces MAR loss. Chapter 6 presents DeNAV with both theoretical guarantees and extensive empirical validation. Finally, Chapter 7 concludes the thesis and discusses future directions, while Chapter 8 provides the full proofs of the theoretical analyses mentioned in the main studies. The overall thesis structure is illustrated in Figure 1.2.

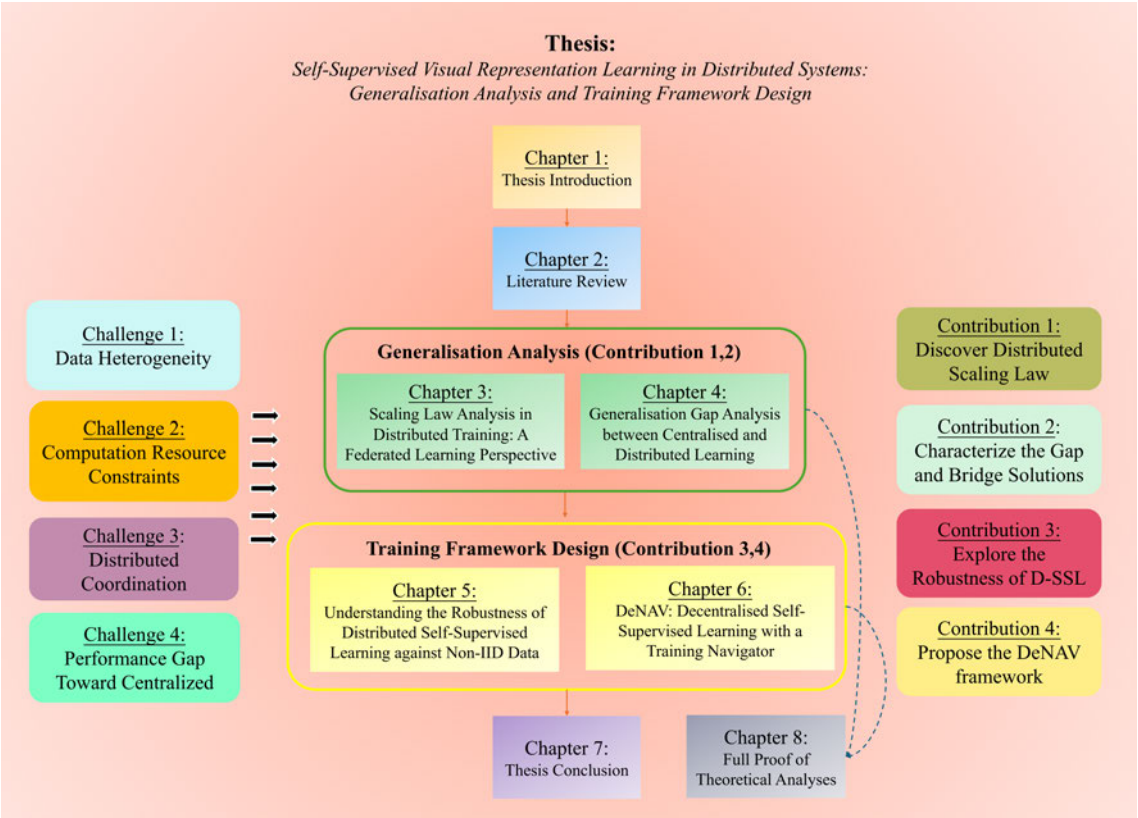


Figure 1.2: **The structure overview of this thesis.**

CHAPTER 2

Literature Review

Chapter Overview: This chapter reviews the foundations and recent advances relevant to distributed self-supervised learning. It begins by introducing self-supervised learning (SSL), covering its motivation, general training principles, commonly used model architectures such as convolutional neural networks and vision transformers, and two dominant paradigms: contrastive learning and masked image modelling. The discussion highlights how SSL leverages large amounts of unlabelled data to learn transferable representations and why it is well-suited for domains where labelled data is scarce.

The chapter then surveys distributed learning, contrasting it with traditional centralised training and outlining two main paradigms: federated learning, which relies on a central server, and decentralised learning, which is server-free. Representative algorithms are reviewed, together with theoretical analyses on convergence and generalisation that provide formal insights into their performance.

Finally, the chapter examines distributed self-supervised learning (D-SSL) as the intersection of SSL and distributed training. It introduces the motivations for combining these approaches, reviews algorithmic innovations that address challenges of heterogeneity and limited client resources, and summarises recent theoretical work that explains the robustness of SSL objectives in distributed settings. The survey emphasises both the progress achieved and the open problems that remain, laying the groundwork for the new analyses and contributions in the subsequent chapters.

2.1 Self-Supervised Learning (SSL)

2.1.1 Background and Motivation

Artificial intelligence (AI) has become deeply integrated into modern society, supporting a wide range of applications that influence daily life and essential services [92]. Smartphones can be unlocked through facial recognition systems that provide both convenience and security. Autonomous vehicles rely on computer vision models to accurately detect pedestrians, traffic lights, and road signs under diverse conditions. In healthcare, AI assists radiologists in interpreting X-rays, CT scans, and MRI images, helping to reduce workloads while improving diagnostic accuracy. Online platforms such as YouTube and TikTok analyse billions of user-uploaded videos and photos to deliver personalised recommendations. These examples demonstrate that artificial intelligence extends far beyond laboratory research and now plays a crucial role in shaping the way people live, communicate, and access information.

The effectiveness of these systems relies heavily on large amounts of training data. Deep neural networks, which serve as the foundation of modern AI, are data-intensive, and their performance often improves as the scale of training data increases. The progress of visual recognition models has been accelerated by large-scale datasets such as ImageNet [19], which contains millions of training samples, while extensive driving video collections have enabled advances in autonomous navigation. Without such abundant training resources, models would struggle to achieve the level of accuracy and robustness demanded by real-world applications.

To achieve this level of performance, most AI systems rely on labelled data [60]. A labelled dataset contains raw inputs, such as images or videos, labelled with meaningful information. For example, if we want an AI tool to be able to distinguish between images of different animals, then the provided labels should specify the category of the object, such as "dog", "cat", or "bird". Similarly, if the task of this tool is to recognise the location of objects, then we need to provide detailed bounding boxes that specify the location of the objects in the image. These annotations can

be used as supervisory signals to guide the model during training to ensure that the learned representations match the expected semantics.

In practice, however, the vast majority of available data remains unlabelled in modern society. People generate countless photos and videos in their daily lives, yet these rarely include annotations describing the content. Surveillance cameras capture continuous streams of footage from public spaces, but the recordings provide no explicit information about what objects or events appear. Hospitals store large archives of radiology scans, yet without expert interpretation. These data cannot directly support supervised training [57].

Creating labels for such data is expensive and labour-intensive. The construction of benchmark training datasets such as ImageNet reportedly required thousands of workers dedicating tens of thousands of hours to annotate millions of images. In the medical field, labelling diagnostic scans requires the expertise of radiologists, often demanding weeks of careful review and carrying high financial costs [17]. Commercial datasets for autonomous driving can require investments reaching millions of dollars, as annotators must label not only object categories but also their precise positions and interactions in complex traffic scenarios. These examples make clear that relying solely on labelled data is unsustainable in the long run.

This tension between the abundance of unlabelled data and the scarcity of labelled data has inspired researchers to consider a new question: can data itself provide the supervision needed for learning? Instead of relying entirely on costly human annotations, the idea is to construct learning objectives in which the data generates its own training signals. This is the essence of self-supervised learning, where the inherent structure and relationships within data guide the model in acquiring useful representations [36]. It is similar to how a person learning Japanese might rely on their knowledge of Chinese characters to guess the meaning of unfamiliar words, using patterns and context to fill in what they do not yet know. By following this principle, self-supervised learning reduces dependence on labelled datasets, makes large-scale training more feasible, and produces representations that can be transferred effectively

to a variety of downstream tasks. The following section discusses the general training principles that underpin this approach.

2.1.2 General Training Idea

Self-Supervised learning is a specialised form of unsupervised learning in which the learning objective is derived directly from the data itself [55]. Formally, let $x \sim \mathcal{D}$ denote an input sample drawn from a data distribution $\mathcal{D}$ without labels. The goal is to learn a representation function $f_\theta : \mathcal{X} \to \mathbb{R}^d$, parameterised by θ, that maps raw inputs to a feature space useful for downstream tasks.

In traditional unsupervised learning, the aim is often to discover global structure in the data, for example, by clustering samples into groups [118] or learning a compressed representation through dimensionality reduction. While such approaches can reveal statistical patterns, they usually lack explicit guidance for capturing semantic relationships that are important for downstream tasks. Self-Supervised learning addresses this gap by introducing an intermediate prediction problem whose solution encourages the model to extract features that are both discriminative and transferable. In this way, it retains the advantage of unsupervised learning in not requiring labelled data, while adding a structured training signal that directs the learning process toward representations that are useful beyond the initial task.

The intermediate prediction problem in self-supervised learning is known as a pretext task [21]. A pretext task is deliberately designed so that solving it requires the model to understand meaningful aspects of the input data. Specifically, given an input x, a transformation or masking operation $t(\cdot)$ is applied to construct a modified view $\tilde{x} = t(x)$, and the model is trained to minimise a self-supervised objective of the form

$$\min_\theta \mathbb{E}_{x\sim\mathcal{D}} \left[\mathcal{L}_{\text{ssl}}\big(f_\theta(\tilde{x}),\, y(x)\big) \right],$$

where $y(x)$ denotes a target derived from the original data, such as another view of x or a partially observed version of it. The design of $\mathcal{L}_{\text{ssl}}$ determines what information the model is encouraged to capture.

This process is similar to someone who already speaks Chinese learning Japanese on their own, using their knowledge of Chinese characters and the similarity between these two languages to deduce the meaning of unfamiliar words or sentences without formal instruction. In this analogy, practising with such sentences serves as the pretext task, while the ultimate goal of using Japanese fluently in real conversations corresponds to success on downstream applications. In self-supervised learning, the pretext task defines the pre-training stage, during which the model learns to solve the auxiliary objective using large amounts of unlabelled data, with the expectation that the learned representations will generalise to other tasks.

Common pretext tasks in visual representation learning can be broadly categorised into three groups, each imposing a different inductive bias and influencing the nature of the learned representations. The first group focuses on invariance to transformations [14, 15, 35], where the model must recognise that differently transformed versions of the same input correspond to the same underlying content. Such transformations may involve geometric changes such as cropping, rotation, translation, scaling, or flipping, as well as appearance changes such as colour adjustment or contrast modification. Tasks in this category encourage the model to learn features that are stable under a range of visual variations, improving robustness to changes in viewpoint, illumination, or image quality. The second group emphasises context prediction [5, 41], where the model uses visible portions of the input to infer missing or hidden parts, for example, by reconstructing masked regions or predicting the relative arrangement of shuffled patches. These tasks promote the learning of fine-grained semantic and structural information, as the model must understand detailed relationships between different parts of the scene. The third group targets cross-view or cross-modal agreement [107], where the model aligns representations from different perspectives of the same instance or from different data modalities, such as matching an image with its textual description. This category encourages the model to learn representations that capture correspondences across diverse input types, which is especially valuable for multimodal or retrieval-based applications.

Furthermore, considering that the aim of learning Japanese is not to perform well only on exercises but to communicate effectively in real situations, the goal of self-supervised learning is not to excel solely at the pretext task but to acquire representations that can be applied to tasks of practical interest. This requires a transfer learning stage, in which the knowledge gained during pre-training is adapted to the target application. Formally, given labelled data $(x, y) \sim \mathcal{D}_{\text{lab}}$, a predictor g_ϕ is trained on top of the learned representation:

$$\min_{\phi} \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{lab}}} \left[\mathcal{L}_{\text{task}}\big(g_\phi(f_\theta(x)),\, y\big) \right].$$

Transfer can be performed in different ways. One approach, known as fine-tuning [69], adapts the entire pre-trained system to the new task, allowing it to refine its knowledge for the target domain. Another approach, known as linear probing [42], keeps the learned representation fixed and trains only a simple prediction layer on top, providing a direct way to measure how well the representation captures generalisable information. The choice of strategy depends on factors such as the amount of available labelled data, the similarity between pre-training and target domains, and the computational budget. In practice, fine-tuning is often preferred when sufficient labelled data and resources are available, as it allows the model to specialise more closely to the target problem, while linear probing is advantageous for rapid evaluation and for scenarios where labelled data is scarce.

In summary, self-supervised learning transforms large-scale unlabelled data into a source of effective supervision through the careful design of pretext tasks, followed by transfer learning to adapt the learned representations to specific applications. While the general principles of self-supervised learning apply across different data modalities, their implementation in the visual domain depends heavily on the choice of backbone architecture. The following section discusses the model architectures most commonly used for visual self-supervised learning, highlighting their design characteristics and how these characteristics influence the learning of visual representations.

2.1.3 Model Architectures for Visual SSL

The effectiveness of self-supervised learning in computer vision is influenced not only by the design of the pretext task, but also by the choice of backbone architecture. The backbone serves as the feature extractor that processes raw visual inputs and generates representations for downstream tasks [36]. Different architectures introduce distinct inductive biases, computational characteristics, and representational capacities, all of which affect how well self-supervised learning objectives can be optimised and how transferable the learned features will be. In recent years, convolutional neural networks (CNNs) and vision transformers (ViTs) have emerged as the two dominant families of architectures for visual representation learning, each offering unique strengths and facing specific challenges in the self-supervised setting.

Convolutional Neural Networks CNNs are built upon the convolution operation, which applies learnable filters to local regions of an input image to extract spatially localised features. This local connectivity drastically reduces the number of parameters compared to fully connected networks and introduces translation invariance, meaning that a learned feature can be recognised regardless of its position in the image. Convolutions are typically combined with pooling layers, which further increase invariance by summarising local regions, and with nonlinear activation functions to enable the learning of complex mappings. By stacking multiple convolutional layers, CNNs learn a hierarchy of features, from edges and textures at lower layers to object parts and semantic concepts at higher layers.

Several CNN architectures have played pivotal roles in advancing visual representation learning:

- **LeNet (1998)** [74]: One of the earliest CNNs, designed for handwritten digit recognition. LeNet demonstrated that convolutional and pooling layers could drastically improve performance on image classification tasks compared to fully connected architectures, paving the way for deep convolutional models.

- **VGG (2014)** [117]: Known for its simplicity and uniform architecture, VGG uses small 3×3 convolutions stacked in depth, showing that deeper networks can achieve significantly better accuracy. Its straightforward design made it widely adopted as a baseline, although its large parameter count and computational cost limited its efficiency.

- **GoogLeNet / Inception Net (2014)** [127]: Introduced the Inception module, which processes input at multiple receptive field sizes in parallel before concatenating the results. This architecture improved efficiency by allowing deeper networks with fewer parameters, and it demonstrated the value of multi-scale feature extraction.

- **ResNet (2015)** [43]: Proposed residual connections that allow gradients to flow more easily through deep networks, enabling successful training of architectures with hundreds of layers. ResNet became the default backbone for many vision tasks, including self-supervised learning methods.

- **MobileNet (2017)** [50]: Designed for efficient inference on mobile and embedded devices, MobileNet employs depthwise separable convolutions to drastically reduce computation and model size, making it a popular choice for scenarios with limited computational resources.

Each of these CNN architectures represents a distinct balance between network depth, parameter efficiency, and the richness of extracted features. Early designs such as LeNet demonstrated the feasibility of convolutional processing for pattern recognition, while architectures like VGG favoured deeper but uniform stacks of convolutions to improve representational capacity. Inception networks introduced multi-scale feature extraction through parallel convolutional paths, and ResNet's residual connections made training very deep models practical without suffering from vanishing gradients. More recent lightweight models such as MobileNet optimise for low computational cost and reduced memory usage, enabling deployment on mobile and embedded devices. In visual representation learning, the choice of CNN

backbone influences not only the diversity and granularity of learned features but also the scalability of training to large datasets or resource-limited environments. The variety of available CNN designs allows researchers to select a backbone that aligns with the constraints and objectives of their specific application.

Vision Transformers The Vision Transformer family builds on the Transformer architecture originally developed for natural language processing (NLP). The core innovation of Transformers is the self-attention mechanism [141], which allows the model to compute relationships between all elements in a sequence, enabling global context modelling from the earliest layers. In NLP, self-attention replaced recurrent architectures by capturing long-range dependencies more efficiently, leading to breakthroughs such as BERT [20].

Subsequently, the Transformer architecture and self-attention mechanism were adapted to the vision domain, giving rise to the original Vision Transformers (ViTs) [22]. During the training process of ViTs, an image is split into fixed-size patches, each treated as a token in a sequence. These tokens are projected into embedding vectors and processed by a standard Transformer encoder. Without the locality constraints of convolution, ViTs can model relationships between distant regions of an image, making them particularly effective for tasks that require reasoning over global context, such as masked image modelling. However, the lack of strong spatial inductive biases means ViTs typically require larger training datasets and careful regularisation to match the data efficiency of CNNs.

Several transformer-based vision architectures have since been proposed:

- **DeiT (2021)** [135]: Data-efficient image transformers that achieve strong performance with smaller datasets through knowledge distillation and optimised training strategies.
- **Swin Transformer (2021)** [90]: Introduces a hierarchical structure with shifted windows, combining the benefits of local attention for efficiency and

global attention for context modelling. Swin has become a popular backbone for detection and segmentation due to its scalability and strong accuracy.

- **PVT (2021)** [151]: Employs a pyramid structure to process multi-scale features, making it suitable for dense prediction tasks.

Recently, ViTs have evolved from being an alternative to convolutional backbones into a widely adopted choice across a diverse range of vision tasks [63]. Their ability to model long-range dependencies and capture global context from the earliest stages of processing has proven beneficial not only in image classification but also in dense prediction tasks such as object detection, semantic segmentation, and instance segmentation, where understanding spatial relationships across the entire image is essential. ViTs have also been successfully applied to fine-grained recognition, scene understanding, and image retrieval, as the self-attention mechanism allows them to adaptively focus on the most informative regions of an image depending on the task requirements. In video understanding, their capacity to integrate information across both spatial and temporal dimensions has enabled competitive performance in action recognition and video object tracking. Beyond purely visual applications, vision transformers serve as the backbone in many multimodal frameworks, where visual features are aligned with text, audio, or other sensory inputs. The versatility of ViTs in handling both global reasoning and cross-domain alignment has made them a foundation for many modern visual systems. Hybrid architectures that incorporate convolutional stems into transformer pipelines further extend these advantages, combining efficient low-level feature extraction with the transformer's high-level global modelling capability.

2.1.4 Representative Methods in Visual SSL

Self-Supervised learning in the visual domain has benefited greatly from advances in model architectures such as convolutional neural networks and vision transformers, which provide strong backbones for representation learning. Building on these architectures, a variety of training strategies have emerged, aiming to extract

informative and transferable features from large collections of unlabelled images. Among these, two paradigms have become dominant: contrastive learning and masked image modelling. Contrastive learning focuses on learning representations by pulling semantically similar samples closer in the feature space while pushing apart dissimilar ones [55], whereas masked image modelling learns by reconstructing missing or corrupted parts of the input [165], encouraging the model to capture both local details and global context. In the following subsections, we review representative methods from each category, outlining their core principles and contributions to the development of modern visual self-supervised learning.

Contrastive Learning Contrastive learning builds on the idea that a good visual representation should bring semantically similar samples closer together in the embedding space while pushing apart dissimilar ones. This learning paradigm is analogous to how a baby learns to recognise a ball. The baby might see the same ball from different angles, distances, and lighting conditions, sometimes even partially occluded. Through repeated exposure, the baby learns that these varied appearances all represent the same object, while recognising that other objects, like a cup or a chair, are different. In the machine learning formulation, these different views of the same image are treated as a positive pair, meaning they should be embedded close together, whereas views from different images are treated as a negative pair, meaning they should be far apart in the learned feature space.

In practice, the typical process of contrastive learning starts by taking an image and producing two correlated views via random augmentations such as cropping, resizing, colour jittering, and blurring [55]. These augmented images are passed through an encoder network (which can be either a convolutional neural network or a vision transformer) to obtain feature vectors, which are then transformed by a projection head into a latent embedding space. A contrastive loss function measures the similarity between features of positive pairs and encourages dissimilarity between features of negative pairs [14]. Depending on the method, negative examples may

be taken from the same batch or from a larger memory bank to provide a richer set of comparisons. The outcome is an embedding space where semantically related inputs are close together even without labels, providing a foundation for effective downstream learning.

Over the past few years, several representative methods have shaped the evolution of contrastive learning.

- **SimCLR (2020)** [14]: Uses a minimal design with a large batch size to provide abundant negative pairs. Its contribution lies in showing that, with strong augmentations and a projection head, simple architectures can achieve high-quality representations without additional components.

- **MoCo (2020)** [42]: Introduces a momentum-updated encoder and a dynamic queue to store a large, consistent set of negative samples. This design avoids the impractical requirement for extremely large batches, making contrastive learning more memory-efficient.

- **BYOL (2020)** [35]: Removes negative pairs entirely by using two networks, one updated online and the other via momentum, and trains them to predict each other's representations. This shows that contrastive-like learning can succeed without explicit negatives.

- **Simsiam (2021)** [15]: Simplifies BYOL's architecture further by removing the momentum encoder, relying instead on a predictor head and a stop-gradient operation to prevent representational collapse, making the method lightweight and easy to train.

- **Barlow Twins (2021)** [163]: Moves beyond pairwise discrimination by minimizing the redundancy between feature dimensions while maximising the similarity between positive pairs, improving robustness to augmentation choices and batch size.

From its origins in methods that relied heavily on large numbers of negative examples, contrastive learning has progressed to approaches that require fewer resources and are more stable to train. Its strength lies in producing semantically rich, discriminative features without labels, often rivalling supervised learning in downstream tasks. However, it is sensitive to the choice and strength of augmentations, and methods relying on instance discrimination may lose fine-grained spatial details, which can limit performance in tasks that require pixel-level precision. These strengths and weaknesses have influenced the parallel development of other SSL paradigms such as masked image modelling, which we will discuss next.

Masked Image modelling Learning to understand an image when parts of it are missing is a skill that also appears in everyday human activities. For example, when interpreting a damaged photograph where regions have faded, a person can often guess the missing details based on context and prior knowledge. In a similar way, masked image modelling trains a model to reconstruct missing parts of an image from the visible regions. This reconstruction process forces the model to learn meaningful visual representations because it must infer semantic relationships between observed and hidden content rather than memorising surface patterns [49].

In masked image modelling, an image is first divided into small patches, and a certain proportion of them are masked [167]. The model then receives the remaining patches and attempts to predict the content of the masked ones. The choice of masking ratio, masking strategy, and reconstruction target can all significantly affect performance. Some approaches mask patches at random, while others use structured patterns or semantic cues to select which regions to hide. The reconstruction target can range from raw pixels to more abstract representations such as discrete tokens derived from a visual tokeniser. Regardless of these variations, the core idea remains that forcing a model to fill in missing information encourages it to capture both local structures and global context.

The well-known masked image modelling methods introduced in recent years include:

- **BeiT(2021)** [5]: Introduces the idea of using a pre-trained discrete visual tokeniser to convert image patches into token IDs, similar to subwords in natural language processing. The model is trained to predict the correct token IDs for masked patches, encouraging it to learn semantic-level features rather than focusing on pixel-level reconstruction.

- **iBOT (2021)** [171]: Combines masked image modelling with self-distillation, where the model learns not only to reconstruct missing content but also to align its representations with those of a momentum-updated teacher network.

- **SimMIM (2022)** [155]: A simpler baseline that predicts raw pixels for masked patches directly, demonstrating that even without a discrete tokeniser, masked image modelling can be effective when paired with strong architectures.

- **MAE (Masked Autoencoder, 2022)** [41]: Uses a high masking ratio and an asymmetric encoder–decoder architecture. The encoder processes only visible patches, while the lightweight decoder reconstructs the missing ones. This design drastically reduces computation during pre-training and has shown strong results across many vision benchmarks.

Over time, masked image modelling has evolved from methods inspired by masked language modelling in natural language processing to approaches that optimise masking strategies, reconstruction targets, and architectural efficiency. The main strength of masked image modelling lies in its ability to leverage large quantities of unlabelled data to learn rich contextual features without requiring negative samples, making it robust to dataset composition. This has led to strong performance in tasks where global context understanding is crucial, such as semantic segmentation and dense prediction. However, masked image modelling can be computationally expensive when reconstruction targets are high-dimensional, and its performance may be sensitive to masking ratios and tokenisation quality.

2.1.5 Self-Supervised Learning Beyond Vision

The development of masked image modelling, as discussed above, is closely related to earlier advances in natural language processing, where self-supervised learning has achieved remarkable success. The general formulation of self-supervised learning described above is not restricted to a specific data modality. Instead, the design of the pretext task and the corresponding objective function can be adapted to the structural properties of different types of data.

In the language domain, the sequential structure of text provides a rich source of self-supervision. One representative approach is masked language modelling, as used in **BERT (2019)** [20], where a subset of tokens in a sentence is masked and the model is trained to predict the missing tokens based on their surrounding context. This formulation closely parallels masked image modelling, with the key difference being that masking is applied over discrete tokens rather than image patches.

Another important paradigm is autoregressive modelling, as adopted in **GPT (2018)** models [108], where the model learns to predict each token conditioned on its preceding context. This approach does not rely on explicit masking but instead uses the natural ordering of sequences to construct the learning objective. Extensions such as **RoBERTa (2019)** [89] further improve the effectiveness of these methods through optimised training strategies and large-scale data.

These approaches demonstrate that self-supervised learning can effectively exploit the inherent structure of different data types to generate supervision signals. While language models rely on sequential dependencies, visual models rely on spatial structures and contextual relationships between image regions. Despite these differences, both paradigms share a common objective of learning representations that capture meaningful patterns in the data without requiring manual annotations.

In this thesis, the focus is placed on the visual domain, where contrastive learning and masked image modelling have emerged as the dominant paradigms. While these methods differ in their formulations, they can be viewed as specific instantiations of this general principle under visual data structures. In the following sections, we build

on these visual self-supervised methods and investigate how they can be adapted and analysed in distributed settings.

2.2 Distributed Learning

2.2.1 Background and Motivation

In many real-world scenarios, data is inherently distributed across a wide variety of sources [59]. For instance, billions of smartphone users generate photos, videos, and sensor readings every day, yet these data remain stored locally on their devices. In the healthcare domain, electronic medical records are held by different hospitals, where strict regulations prevent centralisation. In industrial settings, enterprises often maintain proprietary datasets that cannot be directly shared due to competitive or legal concerns. At the same time, the training of modern deep learning models requires massive amounts of data. The gap between the distributed nature of real-world data and the centralised requirements of conventional deep learning motivates the development of distributed learning paradigms.

Distributed learning differs fundamentally from the classical way of model training, which is often referred to as centralised learning. In centralised learning, data from all sources is pooled into a single repository where the model is trained using powerful servers and high-bandwidth storage systems. This setup ensures uniform access to data, unlimited computational scalability, and relatively low communication costs since all resources are colocated. By contrast, distributed learning must adhere to a no-data-sharing constraint: participants keep raw data locally and only exchange model updates or derived statistics.

Formally, considering a set of N clients, where each client $i \in [N]$ holds a local dataset $\mathcal{D}_i$ drawn from a client-specific distribution $\mathcal{P}_i$. Under this setting, the goal of distributed learning is to jointly learn a global model f_θ by minimising an objective of the form

$$\min_{\theta} \sum_{i=1}^{N} \frac{|\mathcal{D}_i|}{\sum_{j=1}^{N} |\mathcal{D}_j|} \mathbb{E}_{x \sim \mathcal{P}_i} \left[\mathcal{L}(f_\theta(x)) \right],$$

where each term reflects the contribution of a local dataset. While this ensures privacy, it introduces new challenges [100]. For instance, edge devices usually have weaker and heterogeneous computational power compared to centralised servers, leading to uneven training capability. Besides, communication across the network is limited by bandwidth and latency, making frequent synchronisation expensive. Moreover, local datasets are often non-IID, as each participant's data reflects different environments, behaviours, or labelling distributions, which significantly complicates training stability and convergence.

On the other hand, distributed learning can take different forms depending on the application scenario. In environments where a central coordinating entity exists and can be trusted, a server is available to orchestrate the training process by interacting with multiple clients [80]. This configuration is particularly common in consumer applications such as mobile devices, where updates from clients are collected and processed centrally. The presence of a server provides organisational structure and simplifies management, but it also raises concerns about dependence on a single coordinating party.

In contrast, other environments operate without any central authority, where only the clients exist and must collaborate directly [6]. Such settings appear in peer-to-peer systems, multi-institutional collaborations without a trusted hub, or ad-hoc networks where communication occurs only among neighbouring participants. Here, robustness and fault tolerance are enhanced by removing the central node, but the absence of a global coordinator makes the training process more sensitive to network topology and communication efficiency.

The presence or absence of a central server thus divides distributed learning into two primary paradigms, each suited to different application needs. The next subsections examine these paradigms in turn, beginning with federated learning as the representative server–client approach, followed by decentralised learning as the client-only approach.

2.2.2 Federated Learning (for Server-Client Scenario)

Federated learning (FL) emerged as a response to the growing demand for privacy-preserving machine learning in domains where sensitive data cannot be directly centralised [59, 80]. Typical examples include hospitals that manage confidential medical records, government agencies that handle personal identification information, and banks that store detailed financial transactions. In these settings, while the volume of data is large and valuable for training robust models, strict privacy regulations and security considerations make traditional centralised training infeasible. Federated learning offers a practical solution by enabling collaborative model training without requiring raw data to leave local institutions.

The basic workflow of federated learning involves a central server and multiple clients [100]. Each client holds a private dataset and trains a local model using this data. The server does not access the raw data but instead collects model parameters or updates from clients. Through an aggregation process, often implemented as a weighted average of the received updates, the server produces a global model that captures knowledge across clients. This global model is then redistributed to clients for the next round of local training. The cycle of local update, aggregation, and redistribution repeats until the global model converges. Aggregation thus serves as the critical mechanism that combines decentralised knowledge into a unified representation.

Federated learning can be classified in two complementary ways. The first classification is based on the structure of the data [80]. Horizontal federated learning refers to the case where clients share the same feature space but contain different subsets of users or samples, as is common in mobile applications. Vertical federated learning, by contrast, applies when clients hold different feature sets for the same user population, such as when hospitals and insurance companies collaborate using patient data with distinct attributes.

The second classification is based on the system architecture [161]. In centralised federated learning, clients only communicate with the central server, which handles

the global aggregation of updates. In hybrid or hierarchical federated learning, clients communicate both with the server and with one another, often through peer-to-peer or clustered structures. This approach aims to reduce communication bottlenecks and improve robustness by enabling local coordination among clients while still leveraging server aggregation.

Building on these foundations, several representative methods have shaped the development of FL:

- **FedAvg (2017)** [100]: Introduced as the baseline algorithm, it established the principle of averaging local updates weighted by client data size. Its simplicity made FL feasible at scale but it struggles when client data is highly non-IID.

- **FedProx (2020)** [81]: Proposed to address instability in heterogeneous data environments, it adds a proximal term to constrain local updates so that they do not drift too far from the global model. This design improves convergence guarantees under non-IID distributions.

- **SCAFFOLD (2020)** [62]: Addresses client drift caused by non-IID data by introducing control variates that correct local updates. By maintaining variance-reducing correction terms on both server and clients, it achieves faster convergence and improved stability compared to FedAvg, especially in heterogeneous settings.

- **FedNova (2020)** [147]: Tackles the problem of objective inconsistency caused by varying local training steps across clients. By normalising updates according to local computation, FedNova ensures that aggregation aligns with the intended global objective.

- **FedMA (2020)** [146]: Shifts focus from simple parameter averaging to layer-wise model matching, enabling meaningful aggregation when model architectures differ across clients. This expands the flexibility of FL beyond the classical settings with homogeneous client models.

- **FedPer (2019)** [3]: Pioneers personalisation in FL by separating model layers into shared and private components. Clients learn a common representation while fine-tuning personalised layers for their specific data distributions.

Together, these methods illustrate the trajectory of FL research: starting from the vanilla global aggregation (FedAvg), then strengthening robustness against heterogeneous data (FedProx, SCAFFOLD, FedNova), and more recently enabling flexible model architectures (FedMA) and personalised training strategies (FedPer).

The overall development of FL reflects a progressive response to real-world challenges. At the algorithmic level, the central concern is data heterogeneity, which continues to inspire solutions ranging from proximal updates to personalisation. At the system level, communication efficiency and scalability have become priorities as FL moves onto resource-limited edge devices. At the architectural level, hybrid and hierarchical variants have emerged to alleviate the bottlenecks of single-server frameworks. In recent years, these trends have naturally connected FL with decentralised learning, where reliance on a central server is relaxed, and with personalised learning, where client-specific needs are explicitly addressed. This convergence highlights that FL is not an isolated paradigm but a stepping stone toward broader distributed learning systems.

2.2.3 Decentralised Learning (for Client-Only Scenario)

While federated learning has achieved remarkable progress in enabling collaborative training under privacy constraints, its reliance on a central server remains a potential bottleneck. The server must handle aggregation, coordination, and communication, which can introduce single points of failure and limit scalability. In large-scale or highly dynamic environments, such as peer-to-peer networks of mobile devices or edge systems deployed across diverse organisations, depending on a central server is either impractical or undesirable. These concerns have motivated the shift toward decentralised learning, where clients themselves take over the responsibilities of communication and model aggregation [64].

In decentralised settings, no single node holds privileged authority. Instead, clients form a network in which information is exchanged directly among peers. This design naturally eliminates the server bottleneck, improves fault tolerance, and can scale to larger and more dynamic systems [124]. Currently, there are three dominant types of decentralised learning frameworks, which are All-Reduce, Gossip Learning, and Random Walk. A key way to distinguish them lies in their communication patterns. All-Reduce requires each client to exchange information with every other client at each round, ensuring global synchronisation but at the cost of high communication overhead. Gossip Learning relaxes this requirement by limiting communication to direct neighbours in the network topology, significantly reducing overhead but introducing slower information mixing. Random Walk takes another perspective by allowing the model itself to "travel" across the network, being sequentially updated by clients it visits. Each of these frameworks represents a distinct trade-off between communication cost, convergence behaviour, and robustness to heterogeneous data.

All-Reduce All-Reduce is one of the earliest and most widely adopted frameworks for decentralised training [130]. In this setup, each client maintains its own model and participates in synchronous communication rounds where local updates are exchanged with all other clients. The term "All-Reduce" originates from parallel computing, where a reduction operation such as summation is performed across multiple workers and the result is broadcast back so that every participant holds the same updated state. When applied to decentralised learning, this procedure effectively removes the need for a central server, as model averaging is achieved directly among peers. In doing so, it guarantees that all clients remain synchronised after each round of communication.

The practicality of this design has been validated in several representative studies. An early influential work [16] demonstrated that hardware-aware implementations leveraging InfiniBand multicast and GPU-direct technologies can achieve low-latency,

high-throughput synchronisation across large GPU clusters, showing that full decentralisation is possible without a parameter server.

More recently, FlexReduce [76] introduced a flexible variant of All-Reduce tailored to irregular and asymmetric network topologies, a common characteristic of shared cloud environments. Its key innovation lies in distributing gradient portions unevenly across GPUs according to network conditions, thereby maintaining high efficiency even when connectivity is unbalanced. Together, these studies illustrate both the feasibility and adaptability of All-Reduce across different system infrastructures.

The advantages of All-Reduce are straightforward yet powerful. Its simplicity makes it easy to implement, and its exact synchronisation ensures stable convergence without the risk of clients diverging toward different models. Such properties make it well-suited for high-performance computing environments like data centres, where devices are homogeneous and network links are both fast and reliable. On the other hand, these strengths reveal critical weaknesses when applied to edge or heterogeneous scenarios. The requirement for every client to communicate with all others in each round incurs quadratic communication cost, which quickly becomes prohibitive as the network grows. Moreover, synchronous updates mean that the slowest client dictates the speed of training, exposing the framework to severe straggler effects.

In summary, All-Reduce provides a robust solution for decentralised synchronisation in homogeneous clusters, where its full-consistency property can be fully exploited. However, its heavy communication demands and lack of tolerance to device heterogeneity limit its applicability in more practical distributed learning settings, motivating the search for alternative frameworks such as Gossip Learning and Random Walk.

Gossip Learning (Decentralised Federated Learning) Gossip Learning is a decentralised training framework where clients exchange model updates only with their direct neighbours instead of communicating with all peers simultaneously [130]. In each round, a client sends its current model to one or more neighbours, receives

models from others, and aggregates them locally before continuing training. This iterative peer-to-peer exchange resembles the way information spreads in social networks, where rumours or "gossip" propagate gradually through local interactions rather than a single broadcast. The term "gossip" thus captures both the decentralised and incremental nature of this communication scheme. In recent literature, Gossip Learning is also referred to as Decentralised Federated Learning (DFL), since every client not only trains a local model but also performs the aggregation step, effectively assuming the role traditionally played by the server in federated learning.

Several notable works have advanced the design of Gossip Learning and demonstrated its potential in decentralised training:

- **Lian et al.** [82] proposed Decentralised Parallel SGD (DP-SGD), showing that under certain conditions decentralised algorithms can even outperform centralised ones. Their work analysed convergence properties in depth and demonstrated that restricting communication to local neighbours still yields competitive or superior training efficiency.

- **Hegedűs et al.** [44] first studied Gossip Learning as a fully decentralised alternative to federated learning. By letting every client perform both training and aggregation, they effectively remove heavy dependence of FL on the central server. Their experiments on real-world data highlighted strong scalability and robustness compared to centralised FL.

- **Koloskova et al.** [66] developed a novel gossip-based decentralised learning method named CHOCO-SGD, which integrates gradient compression into gossip updates. By reducing the communication payload while maintaining statistical efficiency, their method significantly lowered bandwidth requirements without sacrificing convergence

- **Tang et al.** [131] presented GossipFL, a decentralised federated learning framework that combines gossip communication with sparsified and adaptive

update rules. This design accelerates information propagation while maintaining low communication cost, especially in large-scale and heterogeneous networks.

In recent years, the terminology of decentralised federated learning has been adopted to emphasise the serverless yet aggregation-responsible nature of gossip-based methods and to connect with the popularity of federated learning in distributed research. Two representative contributions include:

- **Shi et al.** [116] proposed two DFL algorithms named DFedSAM and DFedSAM-MGS, which improve model consistency in decentralised federated learning by designing update rules that mitigate divergence across clients, especially under highly non-IID data.

- **Liao et al.** [85] introduced an efficient DFL method, termed FedHP, which adaptively configures communication frequency and model aggregation for heterogeneous participants, ensuring fairness and improving convergence in heterogeneous edge environments.

As the above literature shows, Gossip Learning has gradually evolved from a simple peer-to-peer averaging mechanism into a family of sophisticated decentralised training algorithms. Its main advantage lies in its lightweight communication pattern: each client only interacts with a few neighbours, which avoids the quadratic cost of All-Reduce and makes it naturally scalable to large and dynamic networks. In addition, the absence of a central server improves robustness by eliminating single points of failure, and the peer-to-peer aggregation process has shown resilience to partial client dropouts and network instability. These characteristics make Gossip Learning particularly suitable for open and heterogeneous environments, such as mobile edge devices or sensor networks, where communication is constrained and infrastructure is unreliable.

At the same time, Gossip Learning faces several challenges. Because information spreads gradually through local exchanges, model synchronisation across the entire network is slower than in globally synchronised approaches, which may delay

convergence. Non-IID data exacerbates this issue, since localised aggregation can amplify statistical heterogeneity before information is fully mixed. Moreover, although recent works introduce compression, sparsification, or adaptive aggregation to reduce overhead, the communication cost and consistency gap compared to centralised baselines remain important open problems.

These limitations have motivated researchers to explore alternative ways of structuring peer-to-peer communication. Instead of all clients exchanging information simultaneously, one promising approach is to let the model itself circulate through the network, being updated sequentially by the clients it visits. This design, known as Random Walk, eliminates the need for synchronised communication rounds and provides a fundamentally different perspective on decentralised learning. It not only reduces per-round communication cost but also offers unique opportunities to balance efficiency and robustness in heterogeneous systems. The next subsection will focus on the decentralised studies on this framework.

Random Work Random Walk offers a distinct approach to decentralised training by allowing models to travel across the network instead of requiring all clients to conduct peer-to-peer communication in each round [18]. In a typical setup, one or more models are passed randomly or according to a predefined rule from client to client. Each visited client updates the model with its local data before forwarding it to the next participant. After a sufficient number of passes, the resulting models are eventually aggregated and redistributed to all clients, producing a final consensus. The name "random walk" comes from the stochastic nature of the model transmission, which mirrors the mathematical process of a random walk on a graph. This paradigm can be further divided into two categories: classical single-walk methods, where only one model traverses the network, and multi-walk methods, where several models travel in parallel. While the latter accelerates information mixing and convergence, it also requires an additional aggregation step to reconcile multiple walks, adding complexity to the framework.

Building on this general idea, several representative studies have refined and extended the Random Walk framework in different directions.

- **Ayache et al.** [4] proposed a Random Walk SGD approach that employs weighted random walks to sample nodes based on loss function smoothness. Their analysis shows that weighted sampling via random walks achieves faster convergence than uniform sampling, particularly in heterogeneous graphs.

- **Triastcyn et al.** [137] developed a decentralised learning method that couples random walk with adaptive optimisation techniques like Adam, while also incorporating compression and multiple local steps to minimise communication cost. Their approach demonstrates performance comparable to centralised federated learning in both theory and multi-domain benchmarks.

- **Sun et al.** [121] introduced Adaptive Random Walk Gradient Descent with momentum and adaptive step sizes, offering provable convergence rates in both convex and nonconvex settings. Their method achieves acceleration when stochastic gradients are sparse and is applicable even in zero-order optimisation scenarios.

These advancements collectively highlight how Random Walk has evolved from a simple model-passing mechanism into a versatile family of algorithms that integrate adaptive optimisation, compression, and theoretical guarantees.

Random Walk methods bring several advantages. Since only a limited number of models are transmitted at any time, the communication burden is significantly lighter than in All-Reduce, and the asynchronous, sequential updates reduce sensitivity to slow or unreliable clients, which is often a weakness of synchronous frameworks. Compared to Gossip Learning, Random Walk avoids repeated peer-to-peer averaging and instead relies on model propagation to naturally disseminate information across the network, improving robustness in dynamic or resource-constrained environments. However, information diffusion can be slow in the single-walk setting, while multi-walk approaches, though faster, require careful coordination to maintain consistency

among parallel walks. In summary, Random Walk introduces a unique model-centric communication paradigm that offers clear benefits in efficiency and robustness, while leaving open challenges in balancing convergence speed with system complexity.

2.2.4 Theoretical Analysis on Convergence

Theoretical analysis plays a vital role in understanding distributed learning, as it provides provable statements that go beyond empirical observations. Among the different directions, convergence analysis has been particularly valuable because it quantifies how efficiently distributed algorithms approach an optimal solution and under what conditions such guarantees hold. There is a vast amount of theoretical analysis work on convergence in distributed scenarios, which can be summarised into three main objectives.

First, it examines how various factors and training hyperparameters (including network topology, communication frequency, data heterogeneity, and local update schedules) directly influence convergence speed. For instance, recent work introduced the concept of neighbourhood heterogeneity [73], demonstrating how data distribution differences across a node's local neighbourhood, combined with connectivity, critically determine the convergence behaviour of decentralised SGD under both convex and non-convex objectives. Similarly, unified theoretical frameworks have been developed that establish convergence rates for local and gossip-based decentralised SGD under changing topology [65], with rates that smoothly interpolate between IID and highly non-IID data regimes.

Second, convergence analysis is often used to prove formal convergence guarantees for new distributed algorithms. This not only covers federated algorithms [58] but also includes settings that go beyond centralised server assumptions [131]. One notable result shows that asynchronous SGD can outperform mini-batch SGD by adopting a delay-adaptive learning rate scheme [67], with convergence rates depending on average rather than worst-case delays. Others analyse convergence under constrained

synchronisation or network asynchrony [134], providing bounds that hold without stringent topology assumptions.

Third, theoretical analysis enables rigorous comparison between methods, highlighting whether and how newly designed algorithms offer faster convergence. For example, adaptive decentralised methods such as AdaMDOS and AdaMDOF achieve near-optimal sample complexity bounds for non-convex stochastic and finite-sum optimisation [53], outperforming prior baselines with convergence guarantees. Other works, like MATCHA [148], improve convergence in practical settings by optimising the trade-off between error and runtime, achieving significantly faster convergence in wall-clock time through topology decomposition and prioritised communication over critical links.

In summary, convergence analysis in distributed learning primarily serves three purposes: (i) discovering the impact of scenario and training characteristics on optimisation speed, (ii) establishing formal guarantees for newly designed algorithms, and (iii) rigorously comparing methods to demonstrate accelerated convergence. These roles make convergence analysis not only a tool for theoretical validation but also a driver of algorithmic innovation, as it provides a principled foundation to ensure that advances in distributed learning are both practically effective and provably sound.

2.2.5 Theoretical Analysis on Generalisation

While convergence analysis focuses on how quickly distributed algorithms approach optimal solutions, another central question in theoretical research is how well these algorithms generalise to unseen data. The notion of generalisation is typically formalised through the generalisation error, defined as the gap between the expected risk of a learned model on the underlying data distribution and its empirical risk on the observed training dataset. Understanding and bounding this error is essential for distributed learning, since practical systems often operate with heterogeneous and incomplete data.

Two primary tools have been adopted to derive generalisation guarantees: PAC-Bayes (i.e., Probably Approximately Correct Bayesian) theory and algorithmic stability analysis. PAC-Bayes provides probabilistic upper bounds on the generalisation error by relating the posterior distribution over hypotheses (obtained after training) to a prior distribution defined before training. Its strength lies in offering direct estimation of generalisation performance with data-dependent guarantees. Classical works such as McAllester et al. (1998, 1999) [97, 98] and Seeger et al. (2002) [113] laid the foundation for PAC-Bayes bounds, while more recent studies have extended these results to distributed learning contexts. For example, Zhao et al. provided PAC-Bayes analysis for federated optimisation under client heterogeneity [170].

Stability analysis, on the other hand, measures how sensitive a learning algorithm is to perturbations in the training data, and then relates stability bounds to generalisation bounds. This approach generally requires weaker assumptions than PAC-Bayes, making it more suitable for distributed settings, and has gained broader acceptance in the academic community in recent years. Pioneering works include Bousquet et al. (2002) [10], who first established uniform stability as a sufficient condition for generalisation, and Hardt et al. (2016) [39], who analysed the stability of stochastic gradient descent (SGD). Recent research has extended stability-based generalisation analysis from centralised training to federated learning and decentralised learning. For instance, Sun et al. studied the stability and generalisation gap between federated and decentralised learning [123].

The broader goals of generalisation analysis in distributed learning can be grouped into two directions. First, it aims to identify how system factors (including network topology and data heterogeneity) [173] and training settings (such as the number of clients, the training rounds, and the local updates) [114] influence the generalisation behaviour of distributed algorithms. Second, it is used to provide theoretical justification that new algorithms generalise better than classical baselines. Representative works include Sun et al. (2024) [125], who highlighted the advantages

of new algorithms in handling non-IID client data by comparing the generalisation bounds of different federated learning algorithms.

Despite these advances, most existing studies remain restricted to a specific training scenario or isolated algorithmic settings, leading to a lack of unified theoretical understanding across multiple training settings, such as centralised, federated, and decentralised training. This motivates further research into developing a deeper theoretical understanding that can better explain and bridge the generalisation gaps between different training paradigms.

2.3 Distributed Self-Supervised Learning

2.3.1 Motivation and Challenges

The success of modern machine learning has been closely tied to the availability of massive labelled datasets, yet collecting high-quality labels is often expensive and infeasible in many domains. Self-Supervised learning (SSL) has emerged as a powerful alternative by leveraging raw, unlabelled data to learn generalisable representations, achieving state-of-the-art performance in vision [14, 41], language [72, 141], and multimodal tasks [107]. At the same time, real-world data is increasingly generated and stored in a distributed manner across mobile devices, sensors, and organisations. This decentralised availability of unlabelled data makes it natural and appealing to combine SSL with distributed training frameworks, giving rise to Distributed Self-Supervised Learning (D-SSL) [174].

Formally, let $\{\mathcal{D}_i\}_{i=1}^N$ denote distributed local datasets, where each $\mathcal{D}_i \sim \mathcal{P}_i$. D-SSL aims to learn a shared representation model f_θ by optimising a self-supervised objective across clients:

$$\min_\theta \sum_{i=1}^{N} \frac{|\mathcal{D}_i|}{\sum_j |\mathcal{D}_j|} \mathbb{E}_{x\sim\mathcal{P}_i} \left[\mathcal{L}_{\text{ssl}}(f_\theta, x)\right],$$

where $\mathcal{L}_{\text{ssl}}$ denotes a self-supervised loss such as contrastive or reconstruction objectives. The goal of D-SSL is to unlock the representation learning potential of

SSL while respecting the constraints of distributed environments, thereby enabling scalable, privacy-preserving, and resource-efficient learning at a global scale.

However, deploying SSL in distributed settings introduces unique challenges that go beyond those faced in centralised training. From the perspective of self-supervised learning, the main difficulty lies in the substantial computational overhead. SSL often requires large model architectures and complex training pipelines, including costly pseudo-label generation, contrastive pair sampling, or masked reconstruction. These requirements demand significant compute and memory resources, which are often unavailable on edge devices with limited hardware capacity. This mismatch between the heavy computation of SSL and the lightweight nature of distributed participants significantly complicates practical deployment.

From the perspective of distributed training, D-SSL inherits several long-standing challenges. First, statistical heterogeneity of non-IID local data can severely degrade training stability and convergence [62, 81], and its effect may be magnified in SSL, considering no explicit labels are available. Second, communication constraints arise because clients must frequently exchange model updates or representations [59], yet bandwidth and latency are limited in realistic networks. Third, system heterogeneity, including variable device capacities [79] and unreliable connectivity, complicates the synchronisation of SSL training pipelines that are already computationally demanding. Finally, privacy and security become even more critical in SSL, since raw data may remain on local devices and representation sharing risks the exposure of sensitive information [33].

Overall, while D-SSL promises to leverage the abundance of unlabelled distributed data for powerful representation learning, it must confront challenges stemming from both SSL's reliance on computationally heavy training signals and distributed training's inherent system and statistical constraints. Addressing these challenges is essential for making D-SSL a viable foundation for next-generation large-scale, privacy-aware, and decentralised AI systems.

2.3.2 Algorithm Innovation

While distributed self-supervised learning (D-SSL) faces unique challenges, recent years have witnessed significant progress in algorithmic innovations that enhance training effectiveness and model performance. These methods propose new mechanisms for aggregation, synchronisation, or representation alignment, enabling models to learn more robust features from decentralised unlabelled data. Below, we review several representative approaches and highlight their contributions.

FedU (2021) – Collaborative Unsupervised Visual Representation Learning from Decentralised Data. FedU [174] was one of the first frameworks to adapt self-supervised learning to federated environments with unlabelled, non-IID data. Its main contribution is a selective model update mechanism: rather than blindly aggregating all client updates, FedU decides whether a client should synchronise based on the divergence between its local model and the global model. This strategy prevents harmful updates from drifting the global model away from optimal representations. By introducing this divergence-aware synchronisation, FedU laid the foundation for subsequent methods targeting robustness to heterogeneity.

FedEMA (2022) – Divergence-Aware Federated Self-Supervised Learning. Extending FedU's ideas, FedEMA [175] proposes an exponential moving average (EMA)-based aggregation rule. Instead of hard thresholds for client participation, it assigns adaptive weights to client updates according to their divergence, effectively smoothing the aggregation process. This innovation makes the model less sensitive to outlier clients with highly skewed distributions, offering more stable training under extreme heterogeneity. FedEMA thus demonstrates that carefully designed weighting schemes can significantly reduce the performance gap with centralised SSL.

Orchestra (2022) – Unsupervised Federated Learning via Globally Consistent Clustering. Unlike FedU and FedEMA, which focus on divergence at the model level, Orchestra [93] tackles representation alignment through a

clustering-based approach. Clients locally learn cluster assignments, and a global clustering procedure enforces consistency across the federation. This avoids parameter-level drift and instead aligns feature spaces directly. Orchestra highlights that clustering can serve as a powerful intermediate representation, enabling better cross-client consistency and robustness.

FeatARC (2022) – Does Learning from Decentralised Non-IID Unlabelled Data Benefit from Self-Supervision. FeatARC [149] provides a diagnostic study on the fundamental benefits of SSL under decentralised conditions. It systematically evaluates whether self-supervised pre-training indeed improves generalisation when data is non-IID. The study finds that decentralised SSL is robust to heterogeneous data, and introducing feature alignment and representation correction mechanisms yields substantial improvements. The contribution is less about proposing a single algorithm, and more about offering insights into the limits and potential of SSL in non-IID distributed environments.

FedX (2022) – Unsupervised Federated Learning with Cross Knowledge Distillation. FedX [38] tackles the problem of inconsistent representations across clients by introducing a novel two-sided knowledge distillation mechanism. In addition to local self-distillation within each client, it introduces global knowledge distillation that aligns client models through a central relational signal. The key technical contribution is the use of a relational loss, formulated as the Jensen–Shannon Divergence (JSD), to measure and minimise discrepancies between pairwise feature similarities across clients. This design allows FedX to enforce both local representation compactness and global relational consistency, leading to improved performance in federated SSL under heterogeneous data.

L-DAWA (2023) – Layer-wise Divergence-Aware Weight Aggregation in Federated Self-Supervised Visual Representation Learning. L-DAWA [110] introduces layer-wise divergence weighting. Rather than treating the model as a whole,

it evaluates divergence at each layer across clients and aggregates accordingly. Layers that are more consistent across clients are given higher weights, while divergent layers are downweighted. This innovation provides fine-grained control over aggregation, which is especially beneficial when heterogeneity impacts different parts of the model unequally. The problem it addresses is the granularity of divergence handling in federated SSL.

FedU^2 (2023) – Re-thinking the Representation in Federated Unsupervised Learning with Non-IID Data. FedU^2 [83] addresses the challenge of representation collapse in federated self-supervised learning with highly non-IID data. It introduces two key modules designed to both stabilise and unify the representation space across clients. The first is the Flexible Uniform Regulariser (FUR), which constrains local representations by aligning them with a spherical Gaussian distribution. This is achieved through minimising the unbalanced optimal transport (UOT) distance between client features and a set of uniform reference samples, thereby mitigating local collapse and reducing its propagation during global aggregation. The second is the Efficient Unified Aggregator (EUA), which formulates aggregation as a multi-objective optimisation problem, ensuring that global updates remain consistent with the optimisation directions of heterogeneous clients. Together, these mechanisms prevent collapse and harmonise feature spaces, yielding improved robustness and stronger performance across both cross-device and cross-silo settings.

Overall, these works illustrate the rapid progress in addressing heterogeneity in D-SSL. Innovations have spanned from selective synchronisation (FedU), divergence-aware weighting (FedEMA, L-DAWA), and clustering-based consistency (Orchestra), to feature alignment studies (FeatARC), knowledge distillation approaches (FedX), and representation-level solutions (FedU^2). Despite these advances, existing methods remain largely centred on the non-IID problem, leaving other critical challenges in distributed learning underexplored. In particular, the following questions remain challenging to solve:

- How to efficiently train large models in resource-limited edge devices?
- How to handle heterogeneous compute power in clients and potential non-uniform model architectures?
- How to move beyond contrastive learning to incorporate Masked Image modelling (MIM) into D-SSL?

Addressing these issues will be essential for making D-SSL scalable, versatile, and closer to real-world deployment.

2.3.3 Theoretical Foundation

The theoretical analyses of convergence and generalisation discussed in the distributed learning literature are also relevant to distributed self-supervised learning (D-SSL). They provide useful guarantees about optimisation efficiency and generalisation behaviour in fragmented data environments. However, self-supervised learning differs fundamentally from supervised learning: rather than relying on labels, it extracts feature structure directly from the raw data. This distinction suggests that the robustness properties of SSL under heterogeneous data may not be fully explained by existing distributed learning theory, which thus calls for distinct theoretical analyses crafted specifically for SSL under distributed, heterogeneous scenarios.

A first study provides a theoretical explanation for why self-supervised learning is inherently more robust to dataset imbalance than supervised learning. In a toy setting with a three-class imbalanced dataset, Liu et al. [87] show that supervised objectives tend to overfit frequent classes at the expense of rare ones, while SSL objectives recover both informative directions and additional label-agnostic features. Mathematically, the authors analyse a simplified SSL objective under a matrix factorisation framework. They demonstrate that the SSL solution aligns with the top eigenvectors of the input covariance, which capture features beyond class labels and thus generalise better to rare classes. This insight implies that under local label skew, self-supervised models remain more stable and transferable. The study also

derives a bound showing that the performance gap between imbalanced and balanced pre-training is upper-bounded by terms involving the class imbalance ratio, and these bounds are tighter for SSL than for supervised learning.

A second recent work extends this perspective to decentralised learning under non-IID conditions. Wang et al. [149] follow the theoretical analysis of Liu et al. to introduce the notion of a representability vector, which measures how well the learned feature subspace captures canonical directions present across clients. They prove that, under simplified contrastive SSL objectives (such as SimCLR or SimSiam) combined with FedAvg-style coordination, the representability remains approximately invariant across clients even when data partitions are highly heterogeneous. In contrast, federated supervised training produces representations that diverge across clients when the local data is skewed. The theoretical result quantifies the deviation in representability as a function of data heterogeneity, showing that SSL remains robust under reasonable assumptions. Empirical experiments on non-IID splits of public datasets and an industrial warehouse dataset validate the theoretical findings.

Together, these contributions establish a foundational understanding unique to D-SSL. They show that SSL objectives inherently mitigate the negative impact of class imbalance, and that this robustness extends to client heterogeneity in distributed settings. However, both studies analyse simplified problem settings and fall short of covering different SSL pre-training, such as masked image modelling. They also do not explore decentralised topologies beyond server–client federation. Extending this theory to MIM objectives and to truly decentralised communication protocols remains an important open direction for understanding and improving D-SSL in real-world distributed systems.

CHAPTER 3

Scaling Law Analysis in Distributed Training: A Federated Learning Perspective

Chapter Overview: This chapter presents the first research contribution, which investigates how scaling laws manifest in distributed systems. The success of large language models (LLMs) has highlighted the importance of scaling laws, showing that larger models often yield better performance, but the model size must be carefully determined to be compute-optimal. However, their applicability in decentralised environments remains unclear. Federated Learning (FL), a widely adopted distributed training framework, raises questions about whether principles observed in centralised setups still hold and how to determine the optimal model size under distributed data.

To address this gap, we develop a theoretical framework based on PAC-Bayesian analysis that characterises the generalisation error of models trained with federated stochastic algorithms. By deriving an analytic solution for the model size that minimises this bound, the study reveals that the optimal size decreases with the number of clients when the total training compute is fixed. The results further show that switching from centralised to distributed training under the same compute inevitably reduces the upper bound of achievable generalisation performance, and that estimating optimal size should rely on the average compute across clients. Experiments across models, datasets, and network configurations confirm these theoretical predictions. This chapter provides the first principled understanding of scaling behaviour in FL and offers practical guidance on selecting model sizes for real-world distributed applications.

3.1 Introduction

The rapid progress of artificial intelligence in recent years has been closely tied to the emergence of increasingly large models. From natural language processing to computer vision and multimodal applications, models with billions of parameters have demonstrated unprecedented performance gains [7, 12, 86, 133]. These successes, however, have not been purely the result of trial and error. A major breakthrough enabling the efficient training of such models is the discovery of scaling laws [60]. Scaling laws describe the relationship between model performance and the primary training resources, namely model size, dataset size, and available computation. Formally, let $\mathcal{G}(d, n, c)$ denote the expected generalisation error of a model with size d, trained on n samples using training compute c. A scaling law characterises how $\mathcal{G}$ varies as these resources change, and in particular determines the compute-optimal model size

$$d^*(n, c) = \arg\min_d \mathcal{E}(d, n, c).$$

Such relationships provide guidance for selecting model configurations without exhaustive empirical search and enable principled design of large-scale learning systems. For example, the Chinchilla model, with 70 billion parameters [48], outperformed the much larger Gopher model with 280 billion parameters [109] by being trained on significantly more data in accordance with the predictions of scaling laws. Such results underscore the importance of scaling law theory as a guiding principle for building large-scale models efficiently.

Despite these advances, the practical applicability of scaling laws has been studied almost exclusively in centralised learning scenarios, where training data is assumed to be collected in a single place and models are trained with uniform access to this dataset. Instead, as highlighted in previous chapters, real-world data is rarely centralised and is generally generated and stored across a multitude of distributed devices, organisations, and environments. This distributed nature of modern data has led to the development of Federated Learning (FL), which enables clients to

collaboratively train models without sharing raw data. FL has become a key paradigm for privacy-preserving machine learning in applications ranging from healthcare and finance to mobile and IoT systems. However, decentralisation also introduces new challenges: statistical heterogeneity among clients, system heterogeneity in compute resources, and communication constraints. These challenges inevitably interact with scaling dynamics, raising questions about whether scaling laws derived in centralised training remain valid in federated environments.

Existing theoretical studies in distributed training, as reviewed in the literature review of this thesis, have largely focused on convergence rates and generalisation bounds of specific algorithms. While these analyses provide valuable guarantees, they are typically confined to a single learning paradigm and do not attempt to compare or unify centralised and federated settings. This leaves an important research question unsolved: ***how will the estimation of the optimal model size be affected when training large-scale models with distributed data?*** Given that a decentralised environment fundamentally alters both data distribution and computation allocation, there is no reason to assume that the optimal model size derived under centralised laws will transfer directly to these scenarios. Bridging this research gap is essential not only for theoretical completeness but also for guiding the design of scalable distributed applications in practice.

The goal of this chapter is to address this gap by providing a theoretical analysis of scaling laws in distributed systems, with particular focus on the problem of selecting the optimal model size. Specifically, we model distributed training as stochastic gradient descent over distributed data following the classical federated learning process and employ the PAC-Bayesian framework to derive an upper bound on the generalisation error. Based on this bound, we obtain an analytic solution for the model size that minimises the error, thereby quantifying how decentralisation influences compute-optimal scaling. Our analysis reveals several key findings:

1. The optimal model size has a negative power law relationship with the number of clients when the total training compute is fixed, meaning that distributed training should be allocated with smaller models.

2. Switching from centralised to distributed training leads to an inevitable increase in the upper bound of generalisation error, reflecting a fundamental limitation imposed by data decentralisation.

3. The average compute available per client is the critical factor for estimating the optimal model size in distributed scenarios, providing a practical rule for system designers.

Beyond theoretical contributions, this chapter also presents extensive empirical validation. We pre-train models with varying parameter sizes using transformer-based architectures under both centralised and federated setups, and evaluate them through downstream tasks on datasets such as CIFAR-100 and ImageNet. The results consistently align with our theoretical predictions, demonstrating that scaling behaviour in distributed systems follows distinct dynamics from centralised scenarios.

Notably, the theoretical framework developed in this chapter is not tied to the presence of labels. Although the motivation of this contribution originated from the challenge of training large supervised models, the analysis applies equally to supervised and self-supervised training in distributed environments. This generality ensures that the insights presented here extend beyond one paradigm, providing a foundation that connects with the subsequent chapters of this thesis.

3.2 Related Work

Scaling Law of LLMs. Given the infeasibility of repeatedly training large language models (LLMs) with billions of parameters [132, 164], researchers have developed various scaling laws to predict the relationship between the optimal model size and available training resources. Kaplan et al. were the first to observe a power-law

relationship between model performance and model size [60], laying the foundation for subsequent works. Hoffman et al. revisited the problem under computational constraints and proposed the Chinchilla scaling law, which recommends equally scaling both model size and dataset size [48]. Recent studies have noted that the Chinchilla scaling law could deplete available training data, leading to the development of new scaling laws for data-constrained scenarios [30, 103]. However, these scaling laws are derived from empirical results in centralised training and may not directly apply to scenarios where training data is distributed. This chapter addresses this gap by providing a theoretical analysis of the generalisation bound and empirically validating the impact of data decentralisation on scaling laws.

Federated Training of Large-scale Models. Federated Learning (FL) has garnered significant attention for enabling collaborative model training while preserving data privacy by keeping local data on clients [1, 100, 131]. Specifically, clients receive the global model from the central server, compute updates using their local data, and send these updates back to the server. The server aggregates the updates to refine the global model. This process is repeated until the model achieves the desired performance. With the rise of large-scale models like LLMs, there has been a growing interest in applying FL to train these large-scale models [13]. Since clients generally have fewer computational resources than the server, most of these works follow the intuition to reduce the model size by tailoring the architecture of language models or freezing part of the model parameters during training [37, 154, 156, 157]. However, few studies have explored the modified scaling behaviour of language models in federated scenarios [46, 111, 115], and those that have primarily offered observational insights based on empirical evidence. To address this theoretical gap, we model federated training as an SGD optimisation problem over distributed data and quantify the impact on scaling by deriving an analytic solution for the optimal model size.

Generalisation Bound for Stochastic Algorithms. Stochastic gradient descent (SGD) [9,126] is a widely used optimisation method in machine learning [32,47,75,100, 131]. Previous research has shown that the generalisation performance of stochastic algorithms can be quantified using a PAC-Bayes upper bound [40,91,102,105], which is applied to explore various aspects, including algorithm convergence [102,105], training stability [173], or strategy of tuning hyper-parameters [40]. The generalisation bound also provides treatment for federated learning, helping several studies to propose new training frameworks to address the non-IID problem [170] or model personalisation [2, 8, 142], and other studies to figure out the impact of common parameters in federated scenarios on training results [114]. In contrast, instead of proving similar generalisation bounds for one of the two training regimes, this chapter focuses on comparing the generalisation bounds of the stochastic algorithms in the federated settings with those in the centralised settings. The comparison results show us the impact of changing the training scenario on the optimal model size.

3.3 Preliminaries

3.3.1 Generalisation Error

Formally, considering the hypothesis class of a model is $\Theta \subset \mathbb{R}^d$, machine learning algorithms aim to find the vector of model parameters $\theta \in \Theta$ that minimises the expected risk $\mathcal{R}(\theta) = \mathbb{E}_{\zeta \sim \mathcal{D}} F(\theta; \zeta)$ where d is the dimension of the parameter θ, F is the loss function, and $\mathcal{D}$ is the latent distribution of testing data. Suppose the output parameter θ follows a distribution Q, the expected risk in terms of Q can be formulated as:

$$\mathcal{R}(Q) = \mathbb{E}_{\theta \sim Q} \mathbb{E}_{\zeta \sim \mathcal{D}} F(\theta; \zeta). \tag{3.1}$$

In practice, since $\mathcal{D}$ is not known in advance, the expected risk $\mathcal{R}$ is estimated by the empirical risk $\hat{\mathcal{R}}$ in terms of the latent distribution $\hat{\mathcal{D}}$ of the training data and is defined as:

$$\hat{\mathcal{R}}(Q) = \mathbb{E}_{\theta \sim Q} \mathbb{E}_{\hat{\zeta} \sim \hat{\mathcal{D}}} F(\theta; \hat{\zeta}). \tag{3.2}$$

The difference between $\mathcal{R}$ and $\hat{\mathcal{R}}$ is known as the generalisation error, and the upper bound of the generalisation error is usually used as a critical index to demonstrate the generalisation ability of the training algorithm.

3.3.2 SGD Optimisation

Stochastic Gradient Descent (SGD) is typically used to optimise the empirical risk $\hat{R}$. Consider a training dataset of size m, the mini-batch $\mathcal{S}$ of the training samples is equivalent to a subset of S random indices that are independently and identically (i.i.d.) drawn from the index set $\{1, \ldots, m\}$. The SGD iteration can be formally defined as:

$$\begin{aligned} \theta(t+1) &= \theta(t) - \eta \nabla_{\theta(t)} \hat{\mathcal{R}}(\theta(t)) \\ &= \theta(t) - \eta \frac{1}{S} \sum_{s \in \mathcal{S}} \nabla_{\theta(t)} F_s(\theta(t)). \end{aligned} \tag{3.3}$$

where $\nabla_{\theta_t} \hat{\mathcal{R}}(\theta_t)$ is the estimated gradient of empirical risk on mini-batch and η is the learning rate.

3.4 Theoretical Analysis

In this section, we theoretically explore the impact of distributed data in federated learning on the optimal model size using a PAC-Bayes generalisation bound for stochastic algorithms. In particular, we establish the analytic solution of the optimal model size based on the derived bound, and compare the solutions between different training scenarios to demonstrate several important insights. The analysis is organised as follows. We start with Section 3.4.1 to introduce the rigorous and fair problem setups for both federated and centralized SGD. In Section 3.4.2, we derive the generalisation bound for federated SGD under the established formulations. Section 3.4.3 then compares the optimal model sizes obtained under federated and centralised training using this bound. Based on this comparison, Section 3.4.4 establishes the resulting generalisation performance gap between the two training paradigms. Finally, Section 3.4.5 studies the optimal model size at the client level and shows how it

relates to the global scaling behaviour in distributed systems. The detailed proofs are omitted from this section and provided in Section 8.1 of this chapter.

3.4.1 Problem Setup

To analyse the scaling behaviour under distributed training, we introduce an analytical model that captures the computational structure of federated optimisation [100] while remaining tractable. We consider a distributed training system consisting of n clients and a central server that connects all clients. Each client $i \in \{1, \ldots, n\}$ possesses a local dataset $\mathcal{D}_i$, with the average dataset size denoted as $m = \frac{1}{n}\sum_{i=1}^{n} |\mathcal{D}_i|$. Thus, the total amount of data across all clients is nm. Suppose that training will be repeated for T rounds, following the classical FL algorithm FedAvg [100], the training process at round $j \in \{1, \ldots, T\}$ can be expressed as:

$$\theta_i(j+1) = \bar{\theta}(j) - \eta \nabla_{\bar{\theta}(j)} \mathbb{E}_{\zeta_i \sim \mathcal{D}_i} F(\bar{\theta}(j); \zeta_i), \tag{3.4}$$

$$\begin{aligned} \bar{\theta}(j+1) &= \frac{1}{n}\sum_{i=1}^{n} \theta_i(j+1) \\ &= \frac{1}{n}\sum_{i=1}^{n} (\bar{\theta}(j) - \eta \nabla_{\bar{\theta}(j)} \mathbb{E}_{\zeta_i \sim \mathcal{D}_i} F(\bar{\theta}(j); \zeta_i)). \end{aligned} \tag{3.5}$$

Eq.(3.4) shows the training of the global model $\bar{\theta}(j)$ on client i using its local dataset $\mathcal{D}_i$, and Eq.(3.5) demonstrates the formal update of FL in each round by combining Eq.(3.4) with the model aggregation operation on the central server. Besides, since the training optimisation is performed through SGD, we also define the batch size for local training as $S_{Fed} = k_{Fed} m \in \{1, \ldots, m\}$ where $\frac{1}{m} \leq k_{Fed} \leq 1$ and the number of local training epochs is t. Correspondingly, the baseline centralised scenario holds a dataset $\mathcal{D} = \bigcup_{i=1}^{n} \mathcal{D}_i$ of size nm, and the weights of the initial model in this scenario are the same as that in the federated scenario, i.e, $\{\theta(0) = \theta_i(0) | i \in n\}$. The training of θ follows the SGD optimisation described in Eq.(3.3), denoted as:

$$\theta(j+1) = \theta(j) - \eta \nabla_{\theta(j)} \mathbb{E}_{\zeta \sim \mathcal{D}} F(\theta(j); \zeta)), \tag{3.6}$$

and is iterated for $\frac{T}{n}$ rounds to match the total training compute, which we define following the previous scaling law studies [60, 103] as the total number of samples processed through training (i.e., dataset size times the number of training rounds). In each round, the model θ is trained using data from $\mathcal{D}$ for t epochs with the batch size $S_{Cen} = k_{Cen}nm \in \{1, \ldots, nm\}$ where $\frac{1}{nm} \leq k_{Cen} \leq 1$. Furthermore, we have $k_{Fed}m \leq k_{Cen}nm$ due to more training data allocated to centralised settings in practice, leading to a generally larger batch size in use.

3.4.2 A Generalisation Bound for Federated SGD

To prove a PAC-Bayes generalisation bound for the stochastic algorithms under federated settings, we first present some common assumptions aligned with the previous research [40, 120].

Assumption 1. *Considering that the stochastic gradient $\hat{g}_s(\theta) = \nabla_{\theta(t)}\hat{\mathcal{R}}(\theta(t))$ is computed as the sum of S independent gradients uniformly sampled from the training dataset, we assume that the gradient noise is Gaussian with covariance $\frac{1}{S}C(\theta)$, so $\hat{g}_s(\theta)$ can be approximated as*

$$\hat{g}_s(\theta) \approx g(\theta) + \frac{1}{\sqrt{S}}\Delta g(\theta), \quad \Delta g(\theta) \sim \mathcal{N}(0, C(\theta)), \tag{3.7}$$

where $g(\theta)$ denotes the full gradient of the expected loss. We further assume that $C(\theta)$ remains approximately constant with respect to θ and can be factorised into:

$$C(\theta) \approx C = BB^{\top}, \tag{3.8}$$

where $C \in \mathbb{R}^{d\times d}$ is symmetric and (semi) positive-definite.

We justify Assumption 1 by the central limit theorem when the training data size is substantially larger than the batch size. Since deep neural networks are typically trained on large-scale datasets in realistic cases, the Gaussian assumption about gradient noise is generally valid [24, 120]. Also, the constant matrix C can be justified

when SGD iterates are confined to a small enough region around a local optimum of the loss, where the noise covariance does not vary significantly.

Assumption 2. *Assuming the loss function $F(\theta)$ is smooth, when the stationary distribution of the iterates is confined to a local region near a minimum θ^*, the loss gradient satisfies:*

$$\nabla F(\theta) \approx A(\theta - \theta^*), \tag{3.9}$$

where $A \in \mathbb{R}^{d\times d}$ is a constant (semi) positive-definite matrix representing the local Jacobian of the gradient field.

Assumption 2 is generally valid when SGD converges to a low-variance quasi-stationary distribution near a deep local minimum, where the gradient noise is small compared to the average gradient. According to the fact that the exit time of a stochastic process is typically exponential in the height of the barriers between minima [120], local optima are very stable even in the presence of noise. Thus, SGD follows a relatively directed path toward the optimum. This assumption is also supported by empirical evidence (see p.1, Figures 1(a) and 1(b) and p.6, Figures 4(a) and 4(b) in [78]). Moreover, this assumption can be extended to general cases through translation operations, which would not modify the geometry of the objective function and its associated generalisation ability.

Besides the above assumptions, we also need the formal definition of the PAC-Bayes upper bound to bound the generalisation error. Following previous research [97, 98], we have:

Lemma 1. *For any positive real $\delta \in (0, 1)$, with probability at least $1 - \delta$ over a sample of size N, we have the following inequality for the distribution of the output hypothesis Q and the prior P:*

$$R(Q) \leq \hat{R}(Q) + \sqrt{\frac{\mathcal{D}(Q||P) + \log(\frac{1}{\delta}) + \log(N) + 2}{2N - 1}}, \tag{3.10}$$

where $\mathcal{D}(Q||P)$ is the KL divergence between the distributions Q and P and is defined as:

$$\mathcal{D}(Q||P) = \mathbb{E}_{\theta\sim Q} \log(\frac{Q(\theta)}{P(\theta)}). \tag{3.11}$$

Based on the two assumptions and Lemma 1, we can prove the following helpful lemmas and generalisation bound for federated SGD.

Lemma 2. *Under the above assumptions, if learning rate η and batch size $S_{Fed} = k_{Fed}m$ are fixed, we can derive the following analytic solution for the output parameter $\theta_{Fed}(T)$ of federated SGD:*

$$\theta_{Fed}(T) = \frac{1}{n}\sum_{i=1}^{n}\theta_i(T) = \theta_i(0)e^{-T\bar{A}t} + T\sqrt{\frac{\eta}{k_{Fed}m}}\int_0^t e^{-T\bar{A}(t-t')}\bar{B}dw(t'). \tag{3.12}$$

where A_i is the Jacobian matrix and B_i is the covariance matrix for local training on client i, respectively. Besides, we have $\bar{A} = \frac{1}{n}\sum\limits_{i=1}^{n} A_i$ and $\bar{B} = \frac{1}{n}\sum\limits_{i=1}^{n} B_i$.

Lemma 3. *Under the Assumption 2, the stationary distribution of the Ornstein-Uhlenbeck process for the federated SGD,*

$$q(\theta_{Fed}) = M\exp\left\{-\frac{1}{2}\theta_{Fed}^{\intercal}\Sigma_{Fed}^{-1}\theta\right\}, \tag{3.13}$$

has the following property,

$$T\bar{A}\Sigma_{Fed} + \Sigma_{Fed}T\bar{A} = \frac{T^2\eta}{k_{Fed}m}\bar{C}. \tag{3.14}$$

where M is the normaliser and Σ_{Fed} is the covariance matrix of the stationary distribution.

Theorem 1. *For any positive real $\delta \in (0,1)$, with probability at least $1-\delta$ over a distributed training data set with a total size nm across all clients, we have the following inequality for the distribution Q_{Fed} of the output hypothesis function of*

federated SGD:

$$R(Q_{Fed}) - \hat{R}(Q_{Fed}) \leq \sqrt{\frac{H_1 + H_2 - d + 2\log(\frac{1}{\delta}) + 2\log(nm) + 4}{4nm - 2}}, \tag{3.15}$$

where

$$H_1 = -\log(\det(\Sigma_{Fed})), H_2 = \frac{T\eta}{2k_{Fed}m} tr(\bar{C}\bar{A}^{-1}), \tag{3.16}$$

d is the dimension of the parameter (the model size) and $tr(\bar{C}\bar{A}^{-1})$ is the trace of the product matrix $\bar{C}\bar{A}^{-1}$.

Apparently, it is hard to quantify the above generalisation bound since the covariance matrix Σ_{Fed} for the stationary distribution is not available for the training data. In order to be able to estimate the optimal model size using the generalisation bound, we introduce the following assumption and study a special case of the generalisation bound as in other papers [40, 56].

Assumption 3. *We assume that A and Σ are symmetric matrices, which satisfies $A\Sigma = \Sigma A$.*

Assumption 3 implies that the local geometry around the global minimum and the stationary distribution is homogeneous across all dimensions of the parameter space. A similar assumption has also been used in previous papers [40, 56]. When Assumption 3 also holds, we reformulate the property found in the proof of Theorem 1 and derive a new generalisation bound as follows.

Theorem 2. *Under all the Assumptions of Theorem 1 and with Assumption 3, we have the below generalisation bound for the stationary distribution of federated SGD:*

$$R(Q_{Fed}) - \hat{R}(Q_{Fed}) \leq \sqrt{\frac{H_{Fed} + H'_{Fed} - d + 2\log(\frac{1}{\delta}) + 2\log(nm) + 4}{4nm - 2}}, \tag{3.17}$$

where $H_{Fed} = d\log(\frac{2k_{Fed}m}{T\eta}) - \log(\det(\bar{C}\bar{A}^{-1}))$ and $H'_{Fed} = \frac{T\eta}{2k_{Fed}m} tr(\bar{C}\bar{A}^{-1})$.

3.4.3 Relationship between Two Optimal Model Sizes

The previous derivation established the generalisation bound for federated SGD. To understand how distributed training modifies the scaling behaviour, a reference point is required. We therefore analyse a centralised training process under the same total training compute. This subsection derives the corresponding bound for centralised SGD and compares it with the federated result to characterise the relationship between the two compute-optimal model sizes. To this end, we first establish the lemmas and generalisation bound for centralised SGD trained on the same data and equal amount of training compute.

Lemma 4. *Under all assumptions of Lemma 2, if learning rate η and batch size $S_{Cen} = k_{Cen}nm$ are fixed, we can derive the following analytic solution for the output parameter of centralised SGD trained on the same amount of training data:*

$$\theta_{Cen}(T) = \theta(0)e^{-\frac{T}{n}At} + \frac{T}{n}\sqrt{\frac{\eta}{k_{Cen}nm}}\int_0^t e^{-\frac{T}{n}A(t-t')}BdW(t')). \tag{3.18}$$

where A is the Jacobian matrix and B is the covariance matrix for global training on nm data.

Lemma 5. *When Assumption 2 holds, the Ornstein-Uhlenbeck process's stationary distribution for the baseline centralised SGD,*

$$q(\theta_{Cen}) = M\exp\left\{-\frac{1}{2}\theta^{\intercal}\Sigma_{Cen}^{-1}\theta\right\}, \tag{3.19}$$

has the following property,

$$\frac{T}{n}A\Sigma_{Cen} + \Sigma_{Cen}\frac{T}{n}A = \frac{T^2\eta}{k_{Cen}n^3m}C. \tag{3.20}$$

Lemma 6. *Under all the assumptions of Theorem 2, we have the following generalisation bound for the stationary distribution of centralised SGD trained on the*

same amount of training data:

$$R(Q_{Cen}) - \hat{R}(Q_{Cen}) \leq \sqrt{\frac{H_{Cen} + H'_{Cen} - d + 2\log(\frac{1}{\delta}) + 2\log(nm) + 4}{4nm - 2}}. \tag{3.21}$$

where $H_{Cen} = d\log(\frac{2k_{Cen}n^2m}{T\eta}) - \log(\det(CA^{-1}))$ *and* $H'_{Cen} = \frac{T\eta}{2k_{Cen}n^2m}tr(CA^{-1})$.

Generalisation bounds provide an upper limit on an algorithm's generalisation error, with smaller bounds indicating better generalisation performance. A natural criterion for selecting the optimal model size d^* is the value of d that minimises this bound. While convexity is not guaranteed in general, empirical studies on scaling laws [48, 60, 103] suggest an approximately convex relationship between model size and generalisation performance. Based on this, we assume a locally convex regime and derive a closed-form approximation of d^* using first-order conditions.

Lemma 7. *When all the above assumptions hold, the optimal model size under the output hypothesis function of federated SGD has the following analytic solution:*

$$d^*_{Fed} = \frac{H_1 + H_2 + 8n\log(\frac{1}{\delta}) + 8n\log(nm) - \frac{4}{m} + 8n}{8n - \frac{2}{m} - 4n\log(\frac{2k_{Fed}m}{T\eta})}. \tag{3.22}$$

where $H_1 = -4n\log((\det(\bar{C}\bar{A}^{-1}))$ *and* $H_2 = (\frac{4nT\eta}{k_{Fed}m} - \frac{T\eta}{k_{Fed}m^2})tr(\bar{C}\bar{A}^{-1}))$.

On the other hand, the optimal model size for centralised SGD has the following analytic solution:

$$d^*_{Cen} = \frac{\hat{H}_1 + \hat{H}_2 + 8n\log(\frac{1}{\delta}) + 8n\log(nm) - \frac{4}{m} + 8n}{8n - \frac{2}{m} - 4n\log(\frac{2k_{Cen}n^2m}{T\eta})}. \tag{3.23}$$

where $\hat{H}_1 = -4n\log((\det(CA^{-1}))$ *and* $\hat{H}_2 = (\frac{4T\eta}{k_{Cen}nm} - \frac{T\eta}{k_{Cen}n^2m^2})tr(CA^{-1}))$.

The comparison between d^*_{Fed} and d^*_{Cen} reveals their relationship. Although the two expressions share a similar structure, several components differ. In the numerator, H_1 and H_2 depend on $\bar{A}, \bar{C}$ for federated training but on A, C for centralised training, indicating that the curvature and gradient noise statistics are

estimated from local client distributions rather than the global dataset. In the denominator, the logarithmic term involves $k_{Fed}m$ instead of $k_{Cen}n^2m$, reflecting a change in the effective batch scale under decentralised optimisation. Consequently, the difference between the two optimal model sizes is governed by the relationship between the averaged local statistics and the global statistics, namely $\mathrm{tr}(\bar{C}\bar{A}^{-1})$ versus $\mathrm{tr}(CA^{-1})$. However, we cannot quantify this relationship without further assumptions. Hence, we introduce another assumption.

Assumption 4. *Under the fair comparison condition that the same training dataset is used for both training scenarios, we assume that the local data distributions $\mathcal{D}_1, \ldots, \mathcal{D}_n$ across n clients of size m form a heterogeneous partition of the global dataset $\mathcal{D}$ of size $D = nm$. Hence, we have the following approximate relationships:*

$$\bar{A} \approx A + \Delta_A, \quad \bar{C} \approx \frac{1}{n^\gamma}(C + \Delta_C), \tag{3.24}$$

where $\gamma > 1$, and Δ_A, Δ_C are deviation terms introduced by data heterogeneity. These deviations are assumed to be bounded in norm:

$$\|\Delta_A\| \leq \epsilon_A, \quad \|\Delta_C\| \leq \epsilon_C, \tag{3.25}$$

where ϵ_A, ϵ_C grow with the non-IID degree across clients.

Assumption 4 reflects the realistic cases where client datasets are drawn from heterogeneous (non-IID) distributions and could be justified by the central limit theorem when the average data size m across clients and the size of the global dataset D are both large enough. While the centralised quantities A and C characterise curvature and noise under the full dataset, their decentralised counterparts $\bar{A}$ and $\bar{C}$ may deviate due to non-IID sampling. The inclusion of bounded deviation terms Δ_A and Δ_C, whose magnitudes reflect the degree of data heterogeneity across clients, and the scaling variable γ captures this variability while retaining analytical traceability

for us to quantify the impact of non-IID distributions. Under this assumption, the relationship between the two optimal model sizes is shown below.

Theorem 3. *When all the above assumptions hold, by comparing the optimal model size between the federated and centralised scenarios, we find that:*

$$\lim_{T\to\infty} d^*_{Fed} = \frac{\rho}{n^{\gamma-1}} d^*_{Cen}, \tag{3.26}$$

where $\rho = \frac{S_{Cen}\left(tr(CA^{-1})+tr(\Delta_1)\right)}{S_{Fed}tr(CA^{-1})} > 0$ *and* $\Delta_1 = (CA^{-1}\Delta_A + \Delta_C(I + A^{-1}\Delta_A))A^{-1}$.

Remark 1. *Since we have* $\gamma > 1$*, Theorem 3 shows* $d^*_{Fed} < d^*_{Cen}$ *and suggests the* ***first theoretical insight:***

- *When transferring the training of large-scale models from centralised to distributed scenarios with the same training compute, the optimal model size should be decreased, and the reduction ratio has a negative power law relationship with the number of clients.*

3.4.4 Evidence for the Inferior Generalisation of Distributed Training

The above theoretical proofs demonstrate the generalisation bound and the optimal model size under this bound for each training. In this subsection, we show that these proofs can also serve as an important theoretical basis for an empirical finding observed in many previous works. Specifically, models trained with distributed data are generally found to be inferior to the models trained with centralised data in performance [82, 121]. According to the respective generalisation bound and optimal model size, we derive the below theorem.

Theorem 4. *When all the above assumptions hold, we find the following inequality between the optimal generalisation error of federated SGD and centralised SGD using the same training compute:*

$$\lim_{T\to\infty} (\mathcal{G}^*_{Fed} - \mathcal{G}^*_{Cen}) > 0 \tag{3.27}$$

when the number of clients n satisfies the property: $n > \sqrt[\gamma-1]{\rho}$. Here, $\mathcal{G}^*$ is the optimal generalisation error computed with the optimal model size d^*.

Remark 2. *The condition in Theorem 4 basically holds in practice, considering that realistic distributed scenarios generally scale to a sufficiently large number of clients [59] (e.g., phones with user data, edge sensors, etc.) Also, it is expected that the value of ρ would not be very large. In the ideal case where client data is i.i.d. and both training uses identical batch size, we have $\rho = 1$. Since $n \geq 2$ holds for any federated scenarios, the inequality will be trivially satisfied. Based on this result, we summarise our* ***second theoretical insight:***

- *If a federated scenario with a large number of clients is not allocated more data than the centralised scenario, data decentralisation will lead to a definite gap between the optimal generalisation performance achieved through FL and that under centralised settings, which underscores the challenges of FL.*

3.4.5 Estimating Optimal Model Size by the Average Training Compute Between Clients

Previous analyses have studied the optimal model size for federated SGD training using all local data from the clients. Notably, the total training compute for the federated SGD training is equal to the sum of the training compute allocated to each client. Therefore, we are also interested in the optimal model size at the local level and how it relates to the above results. By a similar proof, we derive the analytic solution for local SGD training, the generalisation bound, and the optimal model size formulation at the client level as follows.

Lemma 8. *Under all the assumptions of Lemma 2, if learning rate η and batch size $S = k_i m$ are fixed, we can derive the following analytic solution for the local output parameter $\acute{\theta}_i(T)$ on client i:*

$$\acute{\theta}_i(T) = \theta_i(0)e^{-TA_i t} + T\sqrt{\frac{\eta}{k_i m}}\int_0^t e^{-TA_i(t-t')}B_i dW(t')). \tag{3.28}$$

Lemma 9. *Under all the assumptions of Theorem 2, we have the following generalisation bound for the stationary distribution of SGD training with solely the local data on client i under the same training compute:*

$$R(Q_i) - \hat{R}(Q_i) \leq \sqrt{\frac{d_i \log(\frac{2k_i m}{T\eta}) - \log(\det(C_i A_i^{-1})) + \frac{T\eta}{2k_i m} tr(C_i A_i^{-1}) - d_i + 2\log(\frac{1}{\delta}) + 2\log(m) + 4}{4m-2}}. \tag{3.29}$$

Lemma 10. *When all the above assumptions hold, the optimal model size at the client level d_i^* has the following analytic solution:*

$$d_i^* = \frac{H_1^{(i)} + H_2^{(i)} + 8\log(m) - \frac{4}{m} + 8}{8 - \frac{2}{m} - 4\log(\frac{2k_i m}{T\eta})}, \tag{3.30}$$

where $H_1^{(i)} = -4\log((\det(C_i A_i^{-1}))$ *and* $H_2^{(i)} = (\frac{4T\eta}{k_i m} - \frac{T\eta}{k_i m^2}) + 8\log(\frac{1}{\delta}) - 4\log((\det(C_i A_i^{-1}))$.

Then, considering that the local data on clients is heterogeneous, we use $\xi_i^C = C_i - \bar{C}$ and $\xi_i^A = A_i - \bar{A}$ to denote client variance in non-IID settings. Comparing d_i^* and the optimal model size d_{Fed}^* in FL across n clients shows their relationship.

Theorem 5. *When all the above assumptions hold, considering the same batch size $\{k_{Fed} m = k_i m | i \in n\}$, the following relation holds between the optimal model size d_i^* on a single client and the optimal model size d_{Fed}^* of FL across n clients:*

$$\lim_{T\to\infty} d_{Fed}^* = \frac{\kappa}{n} \sum_{i=1}^{n} d_i^*, \tag{3.31}$$

where $\kappa = \frac{\left(4m - \frac{1}{n}\right) tr(\bar{C}\bar{A}^{-1})}{(4m-1)\left(tr(\bar{C}\bar{A}^{-1}) + tr(\bar{\xi})\right)} > 0$ *and* $\bar{\xi} = \frac{1}{n}\sum_{i=1}^{n}((\bar{C}\bar{A}^{-1}\xi_i^A + \xi_i^C(I + \bar{A}^{-1}\xi_i^A))\bar{A}^{-1})$.

Remark 3. *Since the existing scaling law suggests that the optimal model size relates to the data size [60], it is intuitive that the optimal model size in FL would be decided by the total data size across clients (i.e., $d_{Fed}^* \approx \sum_{i=1}^{n} d_i^*$). However, Theorem 5 demonstrates that this thought is incorrect. Eq.(3.31) implies $d_{Fed}^* \approx \frac{1}{n}\sum_{i=1}^{n} d_i^*$, as*

$\frac{4m-\frac{1}{n}}{4m-1} \approx 1$ *and the average bias term* $tr(\bar{\xi})$ *is much smaller than* $tr(\bar{C}\bar{A}^{-1})$*. This result highlights our* ***third theoretical insight:***

- *The optimal model size for training in distributed systems is primarily determined by the average training compute per client, rather than the total compute across all clients or the number of clients.*

3.5 Empirical Validation

3.5.1 Experiment Setup

We conduct extensive experiments based on a popular model architecture, Vision Transformer (ViT) [22]. This architecture represents a dominant type of model in deep learning: transformers [141] relying on the attention mechanism, which is frequently used for building large-scale models. Specifically, we build 10 different sizes of ViTs with parameters ranging from 11.62 to 75.41 million.

These models are pre-trained on the Mini-ImageNet dataset [145], which contains 60,000 images extracted from the ImageNet dataset [19]. We adopt the Masked Autoencoder (MAE) [41] approach to pre-train ViTs. To evaluate the effectiveness of pre-training, we conduct linear probing tests, which freeze the pre-trained weights in the backbone and only fine-tune the head layer [43]. The linear probing accuracies of these models on two standard datasets (CIFAR-100 [70] and ImageNet [19]) are collected for analysis. We select the size of the model with the highest linear probing accuracy as the optimal model size.

For all experiments, we strictly follow the problem setup defined in the theoretical analysis. All training resources are kept the same between the centralised and federated scenarios, including the model, total training compute, and dataset. To simulate federated scenarios with n clients and non-IID client data, we divide the training dataset into n partitions by sampling the class priors of the Dirichlet distribution [51]. A more heterogeneous division can be made by specifying a smaller Dirichlet parameter α during sampling. We use $\alpha = 0.1$ by default. Our code for

experiments was implemented using the PyTorch framework and executed on a server with 8 NVIDIA® RTX A5000 GPUs. The details about the experiment and server settings are provided in Tables 3.1 and 3.2.

Table 3.1: **Experiment Settings of Chapter 3.**

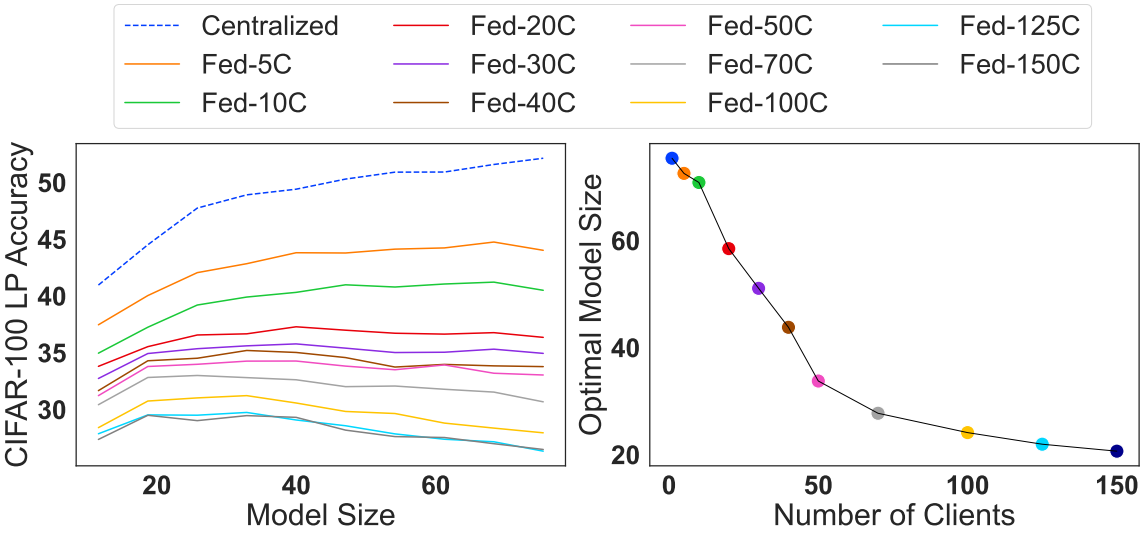
	Value
Model Architecture	Vision Transformer [22]
Pre-training Method	Masked Autoencoder [41]
Pre-training Dataset	Mini-ImageNet [145]
Downstream Dataset	CIFAR-100 [70], ImageNet [19]
Total Pre-Training Compute	5,000,000
Number of Clients in Federated Scenario	$\{5, 10, 20, 30, 40, 50, 70, 100, 125, 150\}$
Data Distribution on Clients	Non-IID ($\alpha = 0.1$ (default))
Model Size Options (Millions)	$\{11.62, 18.71, 25.80, 32.89, 39.97,$ $47.06, 54.15, 61.24, 68.33, 75.41\}$
Linear Probing Epochs	100 (CIFAR-100), 50 (ImageNet)
Pre-training Batch Size	128
Linear Probing Batch Size	512 (CIFAR-100), 1024 (ImageNet)
Base Learning Rate	1.5e-4

Table 3.2: **Server Settings of Chapter 3.**

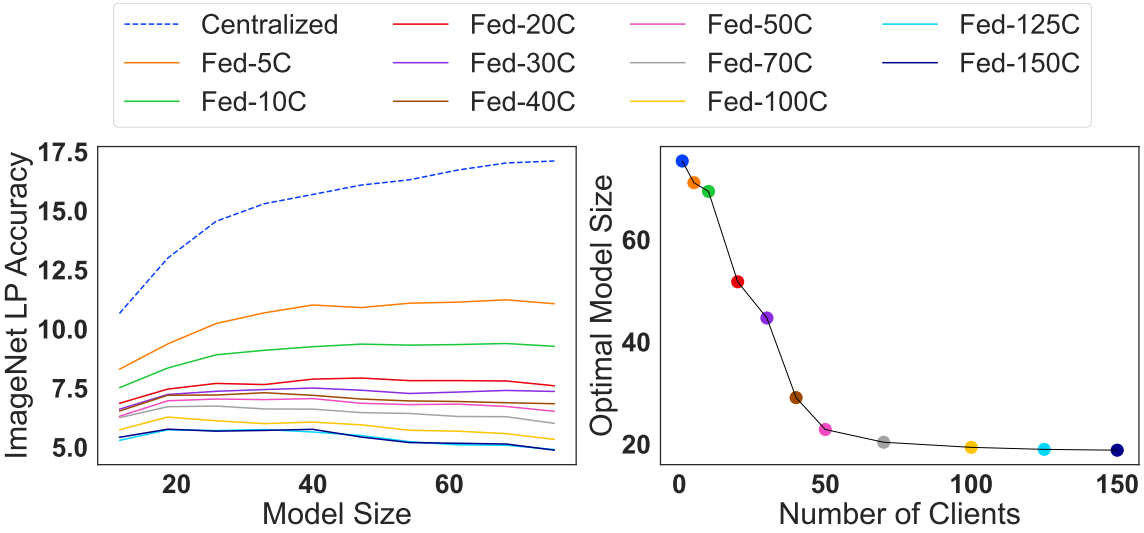
Config	Details
Server GPU Count	8
Server GPU Type	RTX A5000 (24GB)
Server CPU Type	AMD EPYC 7513 32-core
CUDA	11.3
Framework	PyTorch

3.5.2 Empirical Results

Empirical Evidence for the First Insight. We investigate the impact of distributed data on the optimal model size by training models with the same training compute in both the centralized scenario and federated scenarios with different numbers of clients. Figure 3.1(a, Left) shows the linear probing accuracies of ViTs with different sizes on CIFAR-100 in each scenario. Based on the highest accuracy, we find the optimal model size for each scenario and plot them in Figure 3.1(a, Right). The results clearly show a negative power-law relationship between the optimal model size and the number of clients, validating Theorem 3. Besides, we



(a) **CIFAR-100**

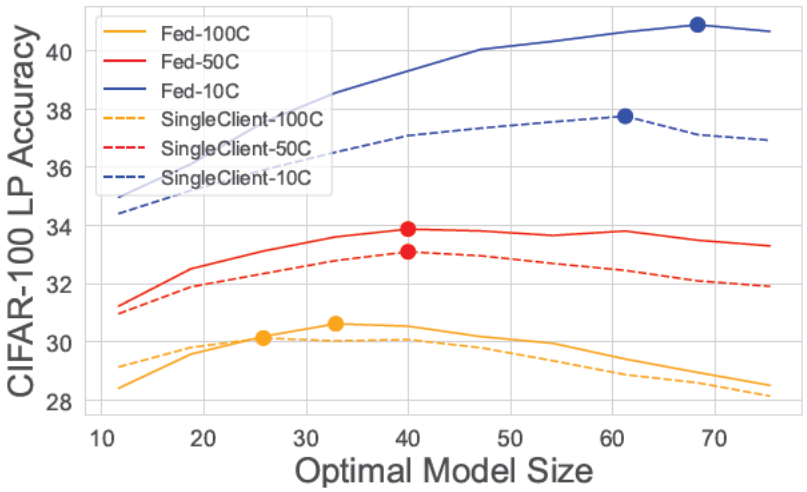


(b) **ImageNet**

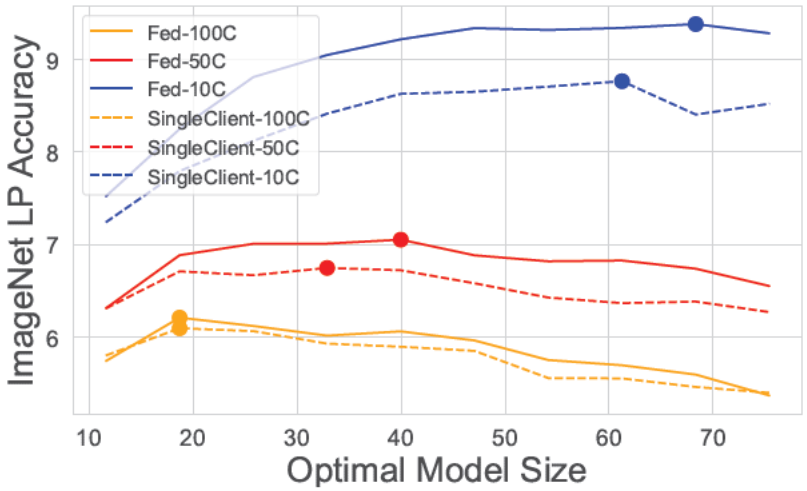
Figure 3.1: **Impact of distributed data on the optimal model size of ViT.** (Left) Curves of linear probing accuracy (%) versus model size. Different lines represent FL scenarios with a different number of clients. (Right) Curve of optimal model size versus the number of clients. Here, the centralised setting refers to the case $n = 1$. The dots represent the highest accuracy of each line in the top figure.

have also collected the linear probing results of ViTs on the ImageNet dataset [19], which has around 1.2 million images. We use 10% of the training samples for linear probing and still observe similar empirical results, as shown in Figure 3.1(b). To further validate the size relationship, we fit the log–log form of Theorem 3 (i.e., $\log d^*_{Fed} = (1-\gamma)\log n + \log \rho + \log d^*_{Cen}$) by the measured optimal sizes in FL.

This gives slopes in $[-0.28, -0.48]$ ($\gamma \in [1.3, 1.5], \rho \in [2.0, 2.4]$) with coefficient of determination $R^2 > 0.88$, confirming a strong negative power law in n.



(a) CIFAR-100



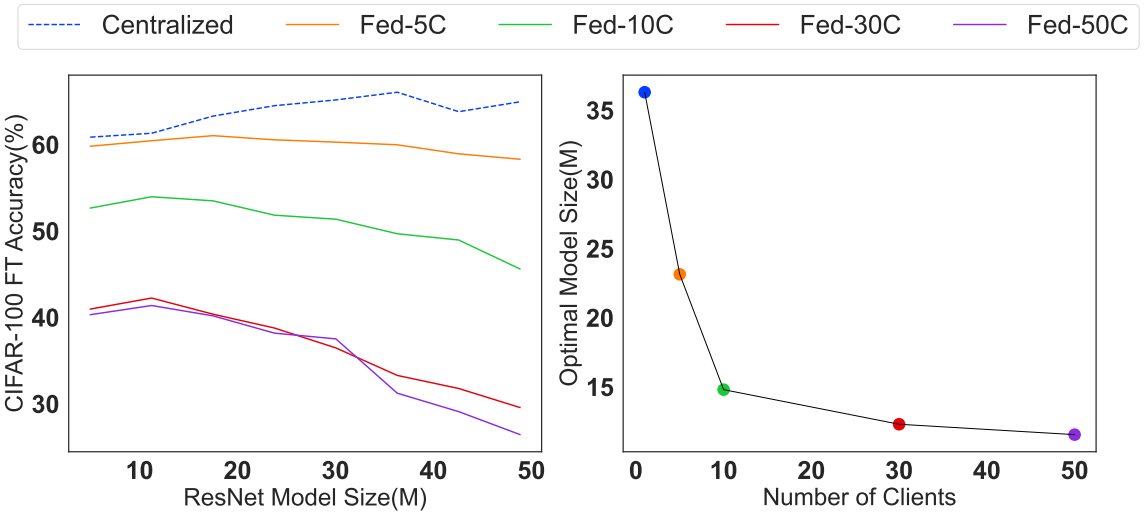
(b) ImageNet

Figure 3.2: **Comparison between the optimal model size across all clients and for a single client.** The dots represent the highest accuracy of each line.

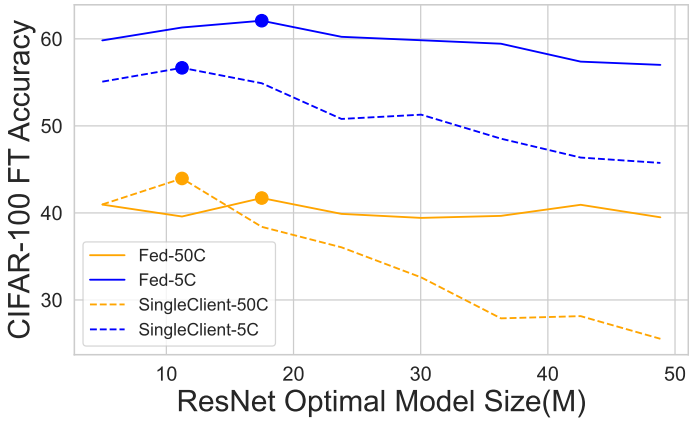
Empirical Evidence for the Second Insight. Figure 3.1(Left) also shows the impact of the number of clients on the linear probing accuracy of models. We observe that when the distributed scenario does not have an advantage in the total amount of training data (i.e., the aggregated data across clients equals the data used in

centralised training) and the total training compute is matched, models trained by federated learning are consistently inferior to those trained in the centralised regime even when the number of clients is small (e.g., $n = 5$). Moreover, the performance gap widens as the number of clients increases. At first glance, this behaviour may appear inconsistent with practical federated learning deployments, where performance can sometimes approach centralised training. However, in real-world systems the distributed setting often benefits from additional effective compute, such as repeated local epochs, prolonged optimisation, or larger cumulative data exposure across rounds. These factors implicitly increase the optimisation budget and partially compensate for the statistical inefficiency introduced by data partitioning. In contrast, our experimental protocol fixes both the total data and the total training compute, thereby isolating the intrinsic effect of decentralisation. Under this controlled regime, the empirical observation directly corresponds to Theorem 4, which states that a positive optimal generalisation gap is unavoidable when distributed training does not enjoy additional data or compute resources. Therefore, the gap in Figure 3.1 should be interpreted as a fundamental statistical limitation between centralized and FL algorithms.

Empirical Evidence for the Third Insight. FL indirectly uses all training data from clients to train a global model by an iterative process of having multiple models trained on different clients using their local data and aggregating the parameters of these models on the server. In Figure 3.2, we train models using only local data from a single client and compute the average of n sets of linear-probing accuracies from n clients. These results are then compared with those of FL. We observe that the optimal model size is actually very close between the two training cases, which matches our third insight shown by Theorem 5. Thus, if the optimal model size in a centralised scenario with the same amount of training data is not known in advance, the optimal model size in a distributed scenario can also be estimated based on the average training compute allocated to each client.



(a) Impact of distributed data on the optimal model size.



(b) Comparison between the optimal model size across all clients and the optimal size for a single client.

Figure 3.3: **Applicability Analysis on ResNets.**

Applicability Study. Beyond ViTs, we evaluate whether our theoretical insights hold for convolutional models. Figure 3.3 demonstrates that ResNets [43] exhibit similar behaviour on their optimal model size, reinforcing that the derived insights are not tied to a specific architecture and may thus serve as general guidelines for size selection in distributed training.

3.6 Chapter Conclusion

This chapter investigates the scaling behaviour of large-scale models in distributed systems, with a focus on how distributed data affects the estimation of the optimal model size. We derive a PAC-Bayes generalisation bound for federated SGD and analyse the global optimal model size under this bound. From this, we obtain three main insights. First, data decentralisation reduces the optimal model size, following an approximate negative power law with respect to the number of clients. Second, moving large-scale training to federated settings inevitably lowers the achievable generalisation performance. Third, the optimal model size should be estimated by the average training compute per client rather than the total compute or network size. Extensive experiments on transformer and convolutional models across multiple datasets confirm these findings. We further confirm these main insights with empirical study on different model backbones and datasets. It is worth noting that the empirical results in this chapter are designed to reveal scaling behaviour under a controlled training budget rather than to maximise predictive accuracy. The training compute is intentionally fixed across centralised and distributed settings so that performance differences reflect intrinsic statistical efficiency instead of optimisation advantages. Consequently, the absolute accuracy values are lower than commonly reported benchmarks, but they remain sufficient to consistently identify the scaling trends predicted by the theory. We expect our results to offer practical guidance for deploying the training of large-scale models in distributed environments.

3.7 Chapter Notations and Definitions

i	Client index
n	Number of clients in the network
m	Average number of local data across clients
θ, Θ	Model parameters
d	Dimension of model parameters / Model Size
$\mathcal{R}$	Risk
f, F	Loss function
$\mathcal{D}$	Dataset
ζ	Sampled data
η	Learning rate
T	Communication rounds
j	Round indices
$\mathcal{S}$	Training Batch
S	Batch size
k	Split ratio for batch
t	Local epochs
g	Stochastic gradient
B, C	Constant matrix about gradient noise
A	Jacobian matrix of the gradient field
Q	Output hypothesis distribution
P	Prior distribution of model parameters
N, D	Data size
δ	Probability
q	Stationary distribution of parameters
Σ	Covariance matrix of the stationary distribution of parameters
$\mathcal{M}$	Normaliser
γ	Value depending on data heterogeneity
$\mathcal{G}$	Generalisation error
$\mathcal{R}, \hat{\mathcal{R}}$	Expected risk and empirical risk
S_{Fed}, S_{Cen}	Batch sizes in federated and centralized training

CHAPTER 4

Generalisation Gap Analysis between Centralised and Distributed Learning

Chapter Overview: The previous chapter examined how scaling laws behave in distributed systems by deriving compute-optimal model sizes within the PAC-Bayesian framework. That analysis clarified how decentralisation alters training dynamics but also raises a more fundamental question: when resources are balanced, is distributed learning inherently disadvantaged in generalisation performance compared to centralised learning? Addressing this requires moving beyond model size and directly analysing the generalisation gap.

We begin with a theoretical analysis of decentralised learning using uniform stability. By constructing the stability bound, we show that generalisation improves with more simultaneously participating clients and stronger network connectivity. Since federated learning corresponds to full connectivity through a central server, its bound dominates that of decentralised learning. This observation justifies focusing on federated settings as the representative form of distributed learning when quantifying the gap with centralised training.

Building on this foundation, the chapter then develops a PAC-Bayesian formulation to explicitly characterise the generalisation gap. Distributed learning is modelled as federated stochastic gradient descent over distributed data, and the generalisation gap is defined as the discrepancy between the federated and centralised generalisation bounds. The results reveal that a gap necessarily exists under equal resources, showing that distributed learning cannot fully match centralised training in equal conditions,

and that the gap size depends on training and scenario settings. Furthermore, we identify the training advantages that can be provided with distributed learning to catch up with centralised learning. The only effective strategy is to enlarge the dataset, either by adding new clients or by expanding local datasets, with the latter proving to be more efficient. Scaling models or increasing communication rounds cannot close the gap. Extensive experiments across architectures and datasets validate these findings, illustrating how the predicted gap emerges in practice and offering guidance on how distributed learning can approach or surpass centralised performance in real-world applications.

4.1 Introduction

In the previous chapter, we investigated how scaling laws adapt under distributed training by deriving compute-optimal model sizes within a PAC-Bayesian framework. That study clarified how decentralisation reshapes the estimation of model scaling laws, yet it left open a more fundamental question. Beyond the choice of model size, it is still unclear whether distributed training can ever attain the same level of generalisation performance as centralised training when both operate under strictly balanced resources.

To understand why this issue matters, it is useful to recall how modern learning systems are trained in practice. Classical deep learning typically takes place in centralised environments, where massive datasets are aggregated on servers equipped with powerful computational resources. This setting has enabled remarkable progress, such as the training of large language models that perform impressively across diverse tasks [7, 60]. However, the assumption of data centralisation is not always realistic. Data in the real world is often dispersed across users, devices, and institutions, and aggregating it into a single location raises serious concerns about privacy and ownership. Federated and decentralised learning has been developed to address this difficulty by allowing multiple clients to collaborate without sharing their raw data [130]. While this design alleviates privacy risks, numerous empirical studies have shown that models trained in distributed systems typically underperform those trained centrally when both are trained with equal resources. The persistence of this observation, in the absence of a clear theoretical explanation, has left open the debate on whether the observed gap reflects limitations of current distributed algorithms or the inherent structural shortcomings.

This chapter addresses that question by using a two-step and rigorous theoretical analysis. We first conduct an analysis of decentralised learning through the lens of uniform stability [39, 173]. By deriving a stability-based generalisation bound, we show that the generalisation performance of decentralised learning improves with

both the number of clients participating simultaneously and the degree of network connectivity. Then, considering all edge devices can be indirectly connected to each other via a central server in FL, the generalisation bound of FL necessarily dominates that of decentralised learning due to the full connectivity. This establishes an ordering between the two paradigms and motivates our subsequent focus on federated settings as the representative form of distributed learning when quantifying the gap with centralised training.

Building on this stability foundation, we then develop a PAC-Bayesian formulation to explicitly characterise the generalisation gap. Distributed learning is modelled as federated stochastic gradient descent over decentralised data, and the generalisation gap is defined as the discrepancy between the federated and centralised bounds. The analysis yields non-vacuous upper and lower bounds on this gap, showing that distributed training cannot fully match centralised training in identical conditions, and that the size of this gap is affected by the training settings. Therefore, completely bridging this gap requires distributed scenarios to be provided with more training resources. Following this idea, we further prove that only incorporating new clients or adding data to existing clients can fully close the performance gap, while having an advantage in model size or communication rounds is not feasible.

To support these theoretical results, we conduct experiments across different model architectures, including ResNets and Vision Transformers, on benchmark datasets such as CIFAR-10 and Mini-ImageNet. The empirical outcomes align closely with the theoretical predictions, confirming the inevitability of the performance gap under equal resources and highlighting the central role of data resource advantage in narrowing it. Through this combination of theory and experiment, this chapter deepens the understanding of why distributed learning falls short of centralised training and offers principled guidance for practitioners seeking to reduce the divide. We thereby extend the previous chapter's theoretical exploration of scaling law variations into a broader investigation that identifies feasible solutions to help distributed training catch up with centralised training in generalisation.

4.2 Related Work

Federated Learning. Federated learning is a class of distributed learning methods proposed for collaborative model training without compromising privacy [1, 80]. The benchmark algorithm for federated learning is Federated Averaging (FedAvg) [100]. In recent years, as people have become aware of the importance of data privacy for security, many research works related to federated learning have emerged [136, 169, 174]. These works generally hold the impression that centralised learning must perform better than federated learning, and many of them focus on proposing advanced federated algorithms to catch up with the centralised baseline [174]. However, the correctness of this impression has not been fully explored from a theoretical aspect. This chapter fills this gap and identifies generic strategies that can bridge the gap between the two training setups.

Studies that Compare Federated Learning with Centralized Learning. Since FL was proposed, there have been studies focusing on the comparison between federated and centralized training. Some works aim to compare the performance of the models trained in each training scenario. These comparative evaluations report that models trained in a centralized setup generally outperform models trained in a federated setup across a variety of tasks and datasets, such as MNIST [96, 104], CIFAR-10 [169], and CICIDS2017 [27]. Similar experimental results are also found in the federated studies that adopt the centralized training results as one of the baselines [174]. In addition to performance comparison, there are comparisons on the convergence rate. Unlike the above studies, these studies show that federated algorithms can attain the same order or faster convergence rate than centralized algorithms [61]. Furthermore, a recent study by Drainakis et al. explores the differences between federated and centralized training from the perspectives of energy cost and bandwidth cost [23]. However, these existing comparisons mainly fall into two categories. First, empirical studies provide useful observations but do not offer theoretical explanations for why the gap arises. Second, some works attempted to

analyze the gap from the theoretical perspective, which they focused on optimization efficiency rather than generalization behavior. Consequently, the fundamental question of whether a generalization gap necessarily exists between federated and centralized training remains theoretically unclear. In this chapter, we address this question by providing a PAC-Bayesian characterization of the generalization gap and deriving analytic conditions under which the gap can be closed.

Generalisation Bound for Stochastic Algorithms. The generalisation of stochastic gradient algorithms has been extensively studied, with PAC-Bayesian theory providing one of the most powerful tools for deriving non-vacuous guarantees [40, 91, 102]. These bounds have clarified how SGD generalises under centralised settings, shedding light on convergence properties and the role of algorithmic hyperparameters. More recently, PAC-Bayesian analysis has also been extended to federated learning, where it has been applied to quantify the effect of non-IID data [170], guide personalisation [2], and understand communication topologies [123].

Complementary to PAC-Bayesian methods, uniform stability theory provides another rigorous avenue for analysing generalisation. Stability-based analyses explain how algorithmic perturbations, such as changes in a single training sample or updates across different clients, propagate to the final hypothesis [39, 125, 173]. This complements the PAC-Bayesian view by highlighting structural aspects of distributed training, especially in decentralised settings. Despite these advances, most prior studies remain tied to a single paradigm, either centralised or distributed, and rarely attempt a unified comparison. In particular, no existing work has derived explicit analytical expressions that directly capture the gap between centralised and distributed training.

This chapter addresses this missing piece by combining the two main tools for generalisation analysis: we first use uniform stability to establish an ordering between decentralised and federated generalisation, and then formulate both centralised and distributed training within the PAC-Bayesian framework to define the generalisation

gap as the difference between their respective bounds. This unified treatment not only explains the widely observed performance gap but also identifies feasible approaches to narrow it. A detailed comparison between this chapter and related studies is provided in Table 4.1 to highlight our contributions.

Table 4.1: **Generalisation Analysis Comparison to Related Works.**

Paper	**Theoretical Analysis**	**Analysis Framework**
London al. [91]	✓	PAC-Bayes
Mou et al. [102]	✓	PAC-Bayes / Stability
He et al. [40]	✓	PAC-Bayes
Yuan et al. [161]	✓	Independent Analysis
Peng et al. [104]	×	×
Mar'i et al. [96]	×	×
Zhao et al. [170]	✓	PAC-Bayes
Sefidgaran et al. [114]	✓	PAC-Bayes
Zhu et al. [173]	✓	Stability
Zhenyu et al. [125]	✓	Stability
Sun et al. [123]	✓	Stability
Ours	✓	Stability + PAC-Bayes

Paper	**Generalisation Bound**	**Gap Study**	**Gap-bridging Insights**
London al. [91]	Centralised	×	×
Mou et al. [102]	Centralised	×	×
He et al. [40]	Centralised	×	×
Yuan et al. [161]	×	✓	✓
Peng et al. [104]	×	✓	×
Mar'i et al. [96]	×	✓	×
Zhao et al. [170]	Federated	×	×
Sefidgaran et al. [114]	Federated	×	✓
Zhu et al. [173]	Federated	×	×
Zhenyu et al. [125]	Federated	×	×
Sun et al. [123]	Fed/Decentralised	✓	×
Ours	Fed/Centralised	✓	✓

4.3 Theoretical Analysis

In this section, we develop theoretical foundations for the generalisation gap between distributed and centralised settings and identify theoretically feasible approaches to close this gap. We first study the generalisation order between decentralised and federated learning through uniform stability analysis and find that the optimal generalisation of distributed learning is achieved by federated learning. Then, we rigorously investigate the generalisation gap. The main ingredient of our investigation is the expression of this gap in the view of the PAC-Bayesian framework. We derive non-vacuous bounds for this theoretical expression, showing that the performance gap necessarily exists under equal training resources and how this gap varies with the parameters. Further analysis suggests that only the strategy of introducing new clients or adding data to existing clients is possible to close this gap fully. The detailed proof is provided in Section 8.2.

4.3.1 Preliminaries and Problem Setup

The analysis of this chapter builds on the same preliminaries introduced in Chapter 3, including the definitions of generalisation error, PAC-Bayesian bounds, and the modelling of SGD optimisation under centralised and federated settings. The main difference here lies in the focus of analysis: rather than deriving compute-optimal model sizes under scaling laws, we investigate whether distributed training can match centralised training under strictly balanced resources (i.e., model size, training data, and total training compute). In this setting, both paradigms share the same training compute, defined as the total number of data samples consumed throughout training. Unless otherwise stated, notations and assumptions remain consistent with those in Chapter 3.

In addition, the analysis introduces the concept of uniform stability as a complementary theoretical tool. Uniform stability is a standard technique for analysing the generalisation behaviour of stochastic optimisation algorithms, defined as follows:

Definition 1. ***(Uniform Stability [125])*** *Consider a dataset $\mathcal{D}$ consisting of local datasets across all clients. Let $\tilde{\mathcal{D}}$ be a neighbouring dataset that differs from $\mathcal{D}$ in at most one data point within some client's local dataset $\mathcal{D}_i$. A learning algorithm $\mathcal{A}$ is said to be ϵ-uniformly stable if*

$$\sup z \sim \mathcal{D}_i, \mathbb{E}\Big[f(\mathcal{A}(\mathcal{D}), z) - f(\mathcal{A}(\tilde{\mathcal{D}}), z)\Big] \leq \epsilon, \tag{4.1}$$

where $f(\cdot, z)$ is the loss evaluated at sample z, and the expectation is taken over the randomness of $\mathcal{A}$.

Lemma 11. *([26, 39]) If a stochastic learning algorithm $\mathcal{A}$ is ϵ-uniformly stable, then its generalisation error satisfies $\epsilon_G \leq \epsilon$.*

The above definition and lemma show that bounding the stability ϵ of training algorithms directly provides a bound on the generalisation error.

4.3.2 Stability and Generalisation Bound of Decentralised and Federated Learning

Our theoretical analysis starts by examining how to formulate distributed learning in order to achieve optimal generalisation. In particular, we seek to establish which of the two prevalent frameworks (i.e., decentralised learning and federated learning) has the advantage in terms of generalisation performance. To find this answer, we introduce three new assumptions besides the ones introduced in Chapter 3:

1. (**Smoothness**) For local training in each client $i \in \{1, \ldots, n\}$, each loss $f_i(\cdot)$ is L-smooth.

2. **(Lipschitz continuity of the loss)** For all $\theta, \tilde{\theta} \in \Theta$ and any data sample z, the loss $f(\cdot; z)$ is G-Lipschitz continuous:

$$|f(\theta; z) - f(\tilde{\theta}; z)| \leq G \|\theta - \tilde{\theta}\|. \tag{4.2}$$

3. **(Bounded stochastic gradient)** For all $\theta \in \Theta$, clients $i \in \{1, \ldots, n\}$, and samples $z \sim \mathcal{D}_i$, the stochastic gradient is bounded:

$$\|\nabla f_i(\theta; z)\| \leq \mathcal{B}. \tag{4.3}$$

These assumptions are standard in prior uniform stability analyses [39, 123, 150] and enable us to establish the stability bound for decentralised learning. Noticeably, as stated in Section 2.2 of the literature review chapter, there are different kinds of decentralised learning frameworks. We adopt the pipeline of parallel random walk for mathematical formulation since it is not limited to full client participation per round, as All-reduce and Gossip learning.

Theorem 6. *Let the learning rate be constant $\eta_t \equiv \eta$ and define $\rho = 1+\eta L$. Consider a general decentralised learning process on a network of n clients with the maximum degree $deg_{\max}$ and the number of edges E, and there are $K \in \{1, \ldots, n\}$ clients participating in each communication round. Then, under the above assumptions, the expected stability and generalisation for this training can be formulated as:*

$$\begin{aligned} &\mathbb{E}\Big[\,\big|f(\theta^T; z) - f(\tilde{\theta}^T; z)\big|\,\Big] \\ &\leq \frac{G}{K}\,\eta\mathcal{B}\sum_{t=0}^{T-1}\rho^{T-1-t}\left(\sum_{j=1}^{K} nc_{\mathcal{P}}\lambda_2^t \;+\; 2\Big(1-\Big(1-\tfrac{deg_{\max}}{2E}\Big)^K\Big)\right), \end{aligned} \tag{4.4}$$

which simplifies to the closed form

$$\begin{aligned} &\mathbb{E}\big[\,\big|f(w^T; z) - f(\tilde{w}^T; z)\big|\,\big] \\ &\leq\; \eta G\mathcal{B}\left[\, nc_{\mathcal{P}}\,\rho^{T-1}\,\frac{1-\big(\lambda_2/\rho\big)^T}{1-\big(\lambda_2/\rho\big)} \;+\; \frac{\rho^T-1}{\rho-1}\cdot\frac{2\big(1-\big(1-\frac{deg_{\max}}{2E}\big)^K\big)}{K}\,\right]. \end{aligned} \tag{4.5}$$

Here, T is the total number of communication rounds, $\mathcal{P}$ is the transition matrix of the communication status, $c_{\mathcal{P}}$ is a constant depending on $\mathcal{P}$, and λ_2 is the second largest eigenvalue of $\mathcal{P}$.

Remark 4. *The bound in Theorem 6 highlights how the number of simultaneously participating clients k and the network connectivity influence the generalisation behaviour of decentralised learning. Although K appears in both the numerator and the denominator of Eq.(4.5), its asymptotic effect is favourable. As K increases, the $\frac{1}{K}$ term dominates, leading to a smaller bound and thus better stability and generalisation. Connectivity also has a decisive impact through the number of edges E and the second-largest eigenvalue λ_2. Denser networks increase E and reduce λ_2, both of which tighten the error bound. Importantly, this result further implies that the generalisation performance of decentralised learning is upper-bounded by that of FL under equal conditions. Specifically, this is because the central server assumed in FL enables indirect connectivity among all clients, corresponding to the case of full connectivity. Therefore, when the same number of clients K is sampled per round, federated learning achieves at least as strong generalisation as decentralised learning, making it the natural representative of distributed training when analysing the gap against centralised training.*

Proof Sketch. The proof relies on uniform stability analysis, which quantifies how the output of the algorithm changes when a single training example is perturbed. We couple two training trajectories $\{\theta^t\}$ and $\{\tilde{\theta}^t\}$ and track their distance $\Delta_t = \|\theta^t - \tilde{\theta}^t\|$. For the case where both trajectories sample the same client at step t, the L-smooth assumption implies

$$\|\theta^{t+1} - \tilde{\theta}^{t+1}\| \leq (1 + \eta_t L)\|\theta^t - \tilde{\theta}^t\|. \tag{4.6}$$

When different clients or different samples are selected, the recursion includes an additional perturbation term β_t, yielding

$$\Delta_{t+1} \leq (1 + \eta_t L)\Delta_t + \eta_t \beta_t, \tag{4.7}$$

which can be unrolled as

$$\Delta_T \leq \sum_{t=0}^{T-1} \left(\prod_{s=t+1}^{T-1} (1 + \eta_s L) \right) \eta_t \beta_t. \tag{4.8}$$

Taking expectations and using the bounded gradient assumption $\|\nabla f_i(w;z)\| \leq \mathcal{B}$, we obtain

$$\mathbb{E}[\Delta_T] \leq 2\mathcal{B}\sum_{t=0}^{T-1}\left(\prod_{s=t+1}^{T-1}(1+\eta_s L)\right)\eta_t\left(\tfrac{n}{2}c_{\mathcal{P}}\lambda_2^t + \tfrac{\deg_{\max}}{2E}\right), \tag{4.9}$$

where the two terms correspond to (i) the probability that a single communication process (often referred to as a walk) diverges to different clients in two decentralised scenarios, governed by the spectral gap $1-\lambda_2$, and (ii) the collision probability when both select the same client. By Lipschitz continuity of the loss with constant G, decentralised learning performed with a single communication walk holds the stability bound below

$$\mathbb{E}[|f(\theta^T;z) - f(\tilde{\theta}^T;z)|] \leq G\,\mathbb{E}[\Delta_T]. \tag{4.10}$$

Extending this result to the case of K participating clients and K parallel walks by averaging reduces variance and introduces interaction between trajectories. Formally,

$$\mathbb{E}\big[|f(\theta^T;z) - f(\tilde{\theta}^T;z)|\big] \leq \tfrac{G}{K}\sum_{j=1}^{K}\mathbb{E}\big[\|w_j^T - \tilde{w}_j^T\|\big]. \tag{4.11}$$

Compared to the single walk case, the key refinement is that the event probabilities now scale with K. This yields the following generalisation bound:

$$\mathbb{E}[|f(\theta^T;z)-f(\tilde{\theta}^T;z)|] \leq \tfrac{G}{K}\mathcal{B}\sum_{t=0}^{T-1}\left(\prod_{s=t+1}^{T-1}(1+\eta_s L)\right)\eta_t\Big(Knc_{\mathcal{P}}\lambda_2^t + 2\big(1-(1-\tfrac{\deg_{\max}}{E})^K\big)\Big), \tag{4.12}$$

which further simplifies into the generalisation inequalities in Theorem 6 by letting $\eta_t \equiv \eta$, and $\rho = 1+\eta L$.

4.3.3 PAC-Bayesian Generalisation Gap

Building on the above result, we study the generalisation performance of federated learning and compare it with centralised learning. To derive the PAC-Bayesian view of the performance gap between federated learning and centralised learning, we

first need to establish the PAC-Bayes upper bounds for the generalisation error of models trained in each scenario. By following the Assumptions 1 and 2 introduced in Chapter 3 and using a similar proof, we derive the following generalisation bounds:

Lemma 12. *For any positive real number $\delta \in (0,1)$, with probability at least $1-\delta$ over a distributed training dataset of total size nm across n clients, the following inequality holds for the distribution Q_{Fed} of the output hypothesis learned by federated SGD:*

$$R(Q_{Fed}) - \hat{R}(Q_{Fed}) \leq \sqrt{\frac{H_F + \frac{T\eta}{2k_{Fed}m} tr(\bar{C}\bar{A}^{-1}) - d + 2\log(\frac{1}{\delta}) + 2\log(nm) + 4}{4nm - 2}}. \tag{4.13}$$

where $H_F = -\log(\det(\Sigma_{Fed}))$, C_i is the covariance of the loss gradients and A_i is Jacobian matrix around the minimum of the loss function for local training on client i, $\bar{C} = \frac{1}{n}\sum_{i=1}^{n} C_i$, $\bar{A} = \frac{1}{n}\sum_{i=1}^{n} A_i$, d is the dimension of the model parameter θ (parameter size), T is the number of communication rounds, η is the learning rate, $tr(\bar{C}\bar{A}^{-1})$ is the trace of the product matrix $\bar{C}\bar{A}^{-1}$ and Σ_{Fed} denotes the covariance matrix for the stationary distribution of federated learning.

Corollary 1. *For any positive real number $\delta \in (0,1)$, with probability at least $1-\delta$ over a centralised training dataset of total size D on server, the following inequality holds for the distribution Q_{Cen} of the output hypothesis learned by centralised SGD:*

$$R(Q_{Cen}) - \hat{R}(Q_{Cen}) \leq \sqrt{\frac{H_C + \frac{T\eta}{2nk_{Cen}D} tr(CA^{-1}) - d + 2\log(\frac{1}{\delta}) + 2\log(D) + 4}{4D - 2}}. \tag{4.14}$$

where $H_C = -\log(\det(\Sigma_{Cen}))$, C and A are the covariance and Jacobian matrix around the minima for training with the centralised dataset, and Σ_{Cen} is the covariance matrix for the stationary distribution of centralised training.

Then, under the Assumptions 3 and 4 defined in Chapter 3, we can characterise the difference between federated and centralised generalisation behaviour and formally establish the theorem as follows.

Theorem 7. *When all the above assumptions hold and the training resources for federated and centralised learning are equal, the generalisation gap between the models trained through federated SGD optimisation and the models trained through centralised SGD optimisation has the following analytic solution:*

$$\begin{aligned}\mathcal{G}_{Fed} - \mathcal{G}_{Cen} = & \frac{d\log(\frac{n^{\gamma-1}k_{Fed}m}{k_{Cen}D}) + (\frac{T\eta}{2n^{\gamma}k_{Fed}m} - \frac{T\eta}{2nk_{Cen}D})tr(CA^{-1})}{4D-2} \\ & + \frac{\frac{T\eta}{2n^{\gamma}k_{Fed}m}tr(\Delta_1) + \log(\det(\Delta_2)^{-1})}{4D-2}\end{aligned} \tag{4.15}$$

where $\Delta_1 = (CA^{-1}\Delta_A + \Delta_C(I + A^{-1}\Delta_A))A^{-1}$, $\Delta_2 = (I + C^{-1}\Delta_C)(I + \Delta_A A^{-1})$, *and* $\mathcal{G}$ *is the generalisation bound of a learning algorithm.*

Proof Sketch. The first part of this proof is to re-formulate the generalisation bound derived for each training scenario. Based on Assumption 3, we re-arrange the properties found in the proofs of Lemma 12 and Corollary 1 to find an analytic solution for the constant matrix Σ. Substituting this solution to Eqs.(4.13) and (4.14) will yield new generalisation bounds. We then complete the proof by computing the distance between the two new PAC-Bayes upper bounds and applying Assumption 4 to rearrange this distance equation.

Theorem 7 shows the analytic solution of the generalisation performance gap in the PAC-Bayesian framework.

4.3.4 Non-Vacuous Bounds on Generalisation Gap

In this subsection, we continue to explore this theoretical expression to gain a deeper understanding of the gap. As pointed out at the beginning of the paper, our interest lies in these questions: 1) Does the generalisation gap always exist with equal training resources? 2) How is this gap affected by the environmental variables in the federated scenario? We answer these questions using the following theorem.

Theorem 8. *When all conditions of Theorem 7 hold, and assuming that the training resources are equal for both federated and centralised scenarios, the generalisation*

gap between models trained using federated SGD and those trained using centralised SGD satisfies the following inequalities:

$$\begin{aligned}
&\frac{d\log(3^{\gamma-1}) + T\left(\frac{\eta tr(CA^{-1})}{2*3^{\gamma}k_{Fed}m} - \frac{\eta tr(CA^{-1})}{6k_{Cen}D} + \frac{\eta tr(\tilde{\Delta}_1)}{2^{\gamma+1}k_{Fed}m}\right) + \log(\det(\Delta_2)^{-1})}{4D-2} \\
&\leq O\left(\mathcal{G}_{Fed} - \mathcal{G}_{Cen}\right) \\
&\leq \frac{d\log(\frac{D^{\gamma-1}k_{Fed}m}{k_{Cen}D}) + T\left(\frac{\eta(tr(CA^{-1}))}{2D^{\gamma}k_{Fed}m} - \frac{\eta(tr(CA^{-1})}{2D^{2}k_{Cen}}\right) + \frac{T\eta tr(\tilde{\Delta}_1)}{2^{\gamma+1}k_{Fed}m} + \log(\det(\Delta_2)^{-1})}{4D-2},
\end{aligned} \tag{4.16}$$

for $3 \leq n \leq D$, where $\tilde{\Delta}_1$ satisfies $(\tilde{\Delta}_1)_{i,j} = |(\Delta_1)_{i,j}|$, n represents the number of clients and $D = nm$ is the total data size across clients. Additionally, when $n = 2$, for any constant $\gamma \geq 2$, the generalisation gap between federated and centralised training satisfies the following inequality:

$$\begin{aligned}
&O\left(\mathcal{G}_{Fed} - \mathcal{G}_{Cen}\right) \geq \\
&\frac{d\log(2^{\gamma-1}) + T\left(\frac{\eta(tr(CA^{-1})+tr(\tilde{\Delta}_1))}{2^{\gamma+1}k_{Fed}m} - \frac{\eta tr(CA^{-1})}{4k_{Cen}D}\right) + \log(\det(\Delta_2)^{-1})}{4D-2}
\end{aligned} \tag{4.17}$$

Proof Sketch. We start by proving that the worst case of the generalisation gap monotonically increases with n if the condition $n \geq \sqrt[\gamma-1]{\gamma}$ holds and find that $\sqrt[\gamma-1]{\gamma} \geq e$. Therefore, this monotonic impact will always hold for $n \geq 3$. By substituting this range of n and adopting the fact $k_{Fed}m \leq k_{Cen}D$, we derive the bound of the generalisation gap for $n \geq 3$. Next, considering that the parameter n satisfies $\{2 \leq n \leq D | n \in \mathbb{Z}\}$, we solve $\sqrt[\gamma-1]{\gamma} = 2$ and derive that the lower bound for $n = 2$ can only be found with $\gamma \geq 2$.

Remark 5. *Theorem 8 establishes non-vacuous upper and lower bounds for the generalisation gap between federated and centralised training. These bounds allow us to analyse if the gap necessarily exists and how the gap is affected by various parameters:*

- ***Gap Existence:*** *Since C and A are (semi) positive-definite matrices, we can observe from Eqs.(4.16) and (4.17) that $\mathcal{G}_{Fed} - \mathcal{G}_{Cen} > 0$ requires satisfying*

$d > \frac{\frac{T\eta tr(CA^{-1})}{2nk_{Cen}D} + \log(\det(\Delta_2))}{\log(n^{\gamma-1})}$. *Considering that deep learning typically involves over-parameterised neural networks to perform well [48, 60] and federated scenarios often scale to a significant number of devices (i.e., leading to large n and D), this condition is readily satisfied in practice.*

- ***Number of clients*** n***:*** *As shown through the proved monotonicity, the gap increases with n.*

- ***Model dimensionality*** d***:*** *Both lower bounds scale with the term $d\log(n^{\gamma-1})$. Since $\gamma > 1$ and $n \geq 2$, the gap increases with d.*

- ***Communication rounds*** T***:*** *The impact of T appears in the form of a difference between two trace terms, which makes its sign unclear in general. In the special case where client data is i.i.d. and both training scenarios use identical batch size, the term can become positive, and the gap grows linearly with T. However, this assumption is rarely satisfied in realistic federated setups.*

- ***Non-IID degree:*** *The gap is explicitly affected by $\tilde{\Delta}_1$ and Δ_2, which quantify client heterogeneity. As the term $tr(\tilde{\Delta}_1)$ increases with T, and the term $\log(\det(\Delta_2)^{-1})$ remains fixed, the gap grows with the level of non-IIDness across clients.*

- ***Total dataset size*** D***:*** *Both lower bounds are inversely proportional to D, so increasing D consistently reduces the gap.*

With the above analysis, we can further summarise the following important insight:

- *Under equivalent training resources, a generalisation gap necessarily exists for deep learning between distributed and centralised settings. This gap is small if the training in a distributed scenario satisfies a small number of clients, with minimised data heterogeneity, and has a limited model size. Additionally, this gap is also mitigated if the total data size across clients is sufficiently large.*

4.3.5 Strategies for Completely Closing the Gap

The above theoretical results demonstrate that the gap cannot be eliminated completely as long as training resources are equal between the two scenarios. Therefore, if we still look forward to distributed training catching up with centralised training, the distributed scenario has to be allowed with an advantage in some training resources. Generally, increasing the data size and model size can result in an improvement in model performance. For example, researchers have concluded scaling laws indicating that the performance of large language models is related to these two parameters [48, 60]. Besides, previous federated studies have also empirically shown that increasing the number of communication rounds or the number of clients also leads to improved model performance [100, 174]. So, we study the related parameters n, m, d, and T in federated settings and derive the following theorems.

Theorem 9. *When all the above assumptions hold and assuming that the federated scenario is provided with an advantage in training conditions, the following inequalities hold for the generalisation gap between models trained through federated SGD and those trained through centralised SGD:*

$$\lim_{n\to\infty}(\tilde{\mathcal{G}}_{Fed}-\mathcal{G}_{Cen})=\lim_{n\to\infty}\left(O(\frac{(\gamma d+2)\log(n)}{n})+O(\frac{1}{n^{\gamma+1}})+O(\frac{1}{n})-O(1)\right)<0; \tag{4.18}$$

$$\lim_{m\to\infty}(\tilde{\mathcal{G}}_{Fed}-\mathcal{G}_{Cen})=\lim_{m\to\infty}\left(O(\frac{(d+2)\log(m)}{m})+O(\frac{1}{m^{2}})+O(\frac{1}{m})-O(1)\right)<0 \tag{4.19}$$

Here, $\gamma > 1$, $\tilde{\mathcal{G}}_{Fed}$ is the generalisation bound for federated scenarios having an advantage in training, and $\tilde{\mathcal{G}}_{Fed}-\mathcal{G}_{Cen}\leq 0$ implies that federated training catches up with or outperform centralised training in generalisation.

Theorem 10. *When all the above assumptions hold and assuming that the federated scenario is provided with an advantage in training conditions, the following inequality holds for the generalisation gap between models trained through federated SGD and*

those trained through centralised SGD:

$$\lim_{T\to\infty}(\tilde{\mathcal{G}}_{Fed}-\mathcal{G}_{Cen})=\infty. \tag{4.20}$$

Besides, if the federated scenario contains a large number of clients satisfying $n > \sqrt[\gamma]{\frac{T\eta e}{2k_{Fed}m}}$ *for any* $\gamma > 1$*, we also have:*

$$\lim_{d\to\infty}(\tilde{\mathcal{G}}_{Fed}-\mathcal{G}_{Cen})=\infty. \tag{4.21}$$

Proof Sketch. The proof is similar between Theorems 9 and 10. Each proof consists of two parts. In each part, we select a parameter and re-establish the theoretical representation of the gap by considering that the federated scenario has an advantage in this parameter. Then, we derive a bound for this new expression and compute the limits of this bound when the selected parameter approaches infinity.

Remark 6. *Theorems 9 and 10 show how the gap between distributed and centralised training behaves as key parameters approach infinity, represented by continually growing the advantage of federated training in these parameters. The condition in Theorem 10 basically holds in practice, considering that realistic distributed scenarios generally scale to a sufficiently large number of clients [59] (e.g., phones with user data, edge sensors, etc.) According to Eqs.(4.18), (4.19), (4.20) and (4.21), we find that simply increasing the number of communication rounds (*T*) or the model size (*d*) cannot close the generalisation gap unless more data is introduced. The two feasible approaches to do so are: (1) increasing the number of clients, or (2) increasing the average data per client. Among these, the latter is more efficient, as the gap decreases at a faster rate with respect to* m *than with respect to* n *(i.e.* $O(\frac{(d+2)\log(m)}{m})$ *vs* $O(\frac{(\gamma d+2)\log(n)}{n})$*). This suggests that, in reality, focusing on growing the local dataset in existing clients would be a more efficient way to make distributed training catch up with centralised training than introducing new clients.*

4.4 Empirical Validation

4.4.1 Experiment Setup

To empirically validate our theoretical findings and ensure that they can be applied to a broad range of learning scenarios, we conduct extensive experiments on different models and datasets. The model architectures we consider are ResNet-18 [43] and Vision Transformer (ViT) [22], which represent two dominant types of deep neural networks: Convolutional Neural Networks (CNNs) [75] and Transformers [141]. We build 10 models of different sizes for each architecture to study the impact of model size. We further use two standard datasets to evaluate training under different setups: CIFAR-10 [70], which contains 50,000 training images and 10,000 validation images across 10 classes, and Mini-ImageNet [145], which contains 60,000 images in 100 classes extracted from ImageNet [19].

Since the theoretical analysis in this chapter focuses on the optimisation and generalisation behaviour of learning algorithms rather than a particular training objective, the experiments are conducted under the standard supervised learning setting. Many self-supervised learning methods can be interpreted as supervised learning with automatically generated pseudo-labels [14, 41]. Therefore, the empirical observations obtained in this supervised setting are expected to extend to self-supervised representation learning scenarios as well.

For the Mini-ImageNet dataset, since it does not provide a predefined training split covering all classes, we randomly divide it into 48,000 training images and 12,000 validation images. The complete training set of each dataset is used for centralised training. To simulate federated scenarios with n clients and non-IID client data, we partition each training set into n subsets by sampling the class priors from a Dirichlet distribution [51]. A more heterogeneous partition can be obtained by specifying a smaller Dirichlet parameter α during sampling, and we use $\alpha = 0.1$ by default. The detailed experiment settings are provided in Table 4.2. All experiments are repeated

with three random seeds (i.e., 0, 10, and 100), and the reported results correspond to the average performance across these runs.

Table 4.2: **Experiment Settings of Chapter 4.**

System	Value
Model Architecture	Vision Transformer (ViT) [22] ResNet [43]
Dataset	Mini-ImageNet [145] CIFAR-10 [70]
Range on Communication Rounds	$25 \leq T \leq 100$
Range on Number of Clients	$2 \leq n \leq 100$
Data Distribution on Clients	Non-IID ($\alpha = 0.1$(default))
ViT Model Size Options (Millions)	$\{7.91, 15.00, 22.08, 29.17, 36.26,$ $43.35, 50.44, 57.52, 64.61, 71.70\}$
ResNet Model Size Options (Millions)	$\{4.91, 11.18, 17.45, 23.72, 29.99,$ $36.26, 42.54, 48.81, 55.08, 61.35\}$
Local Training Epochs	$t = 2$
Batch Size	256
Base Learning Rate	1.5e-4

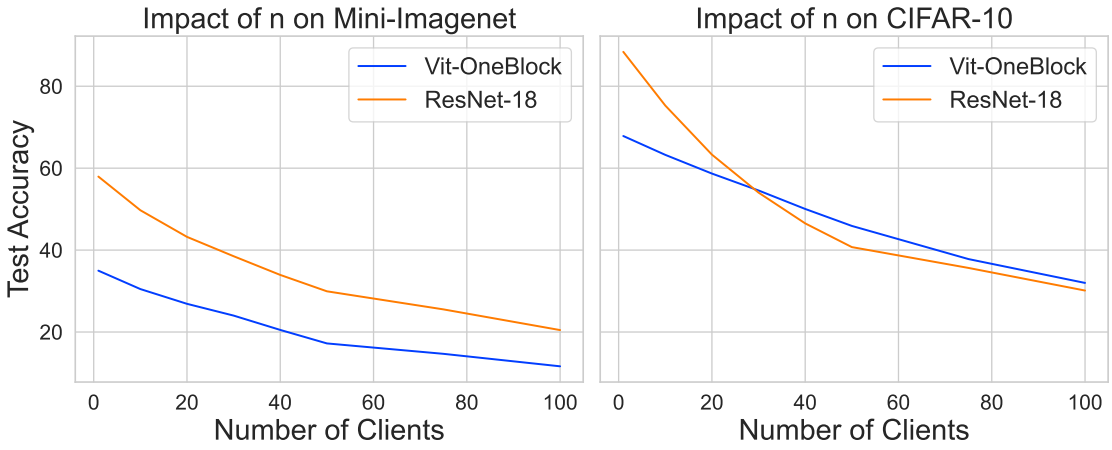


Figure 4.1: **Impact of the number of clients n on the generalisation performance.** Different colours represent different model architectures. **(Left)** Curves of Mini-ImageNet testing accuracy (%) versus the number of clients. **(Right)** Curves of CIFAR-10 accuracy (%) versus the number of clients. For the centralised scenario, we consider that it corresponds to the case $n = 1$.

4.4.2 Empirical Evidence

Generalisation Gap under Equal Training Resource We verify our non-vacuous bounds about the performance gap by first constructing federated and

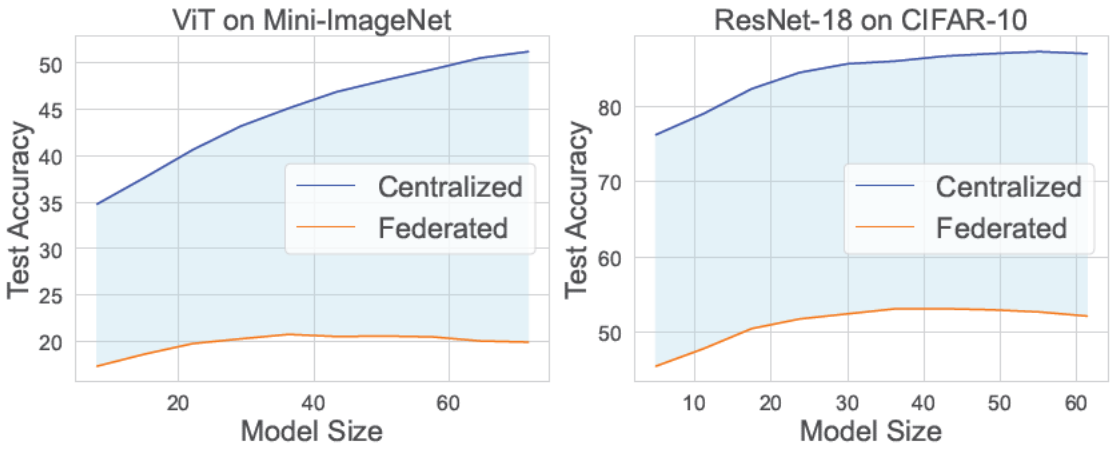


Figure 4.2: **Impact of the model size d (measured in M (millions parameters)) on the generalisation performance.** The generalisation gap between federated and centralised training is demonstrated by the light-blue area between the two lines.

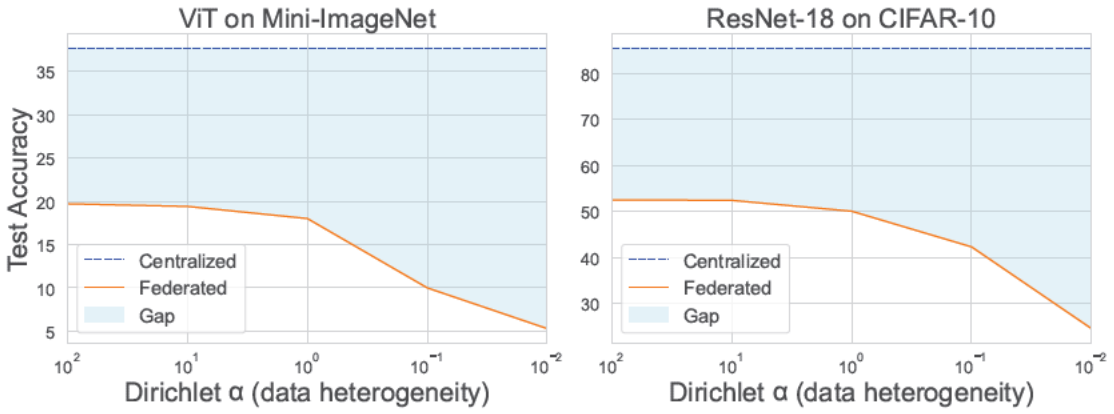


Figure 4.3: **Impact of the non-IID degree on the generalisation performance.** Smaller α implies greater date heterogeneity across clients (i.e., α decreases from left to right on the x-axis).

centralised scenarios with equivalent training resources based on our problem setup. According to Eqs.(4.16) and (4.17), we observe that the gap is affected by the number of clients, the model dimensionality, and the data heterogeneity across clients. Hence, we conduct three sets of experiments to validate their respective impact. Figure 4.1 shows that the testing accuracy of models decreases with the number of clients. Since the centralised scenario can be considered as containing only one client (which is the server), the impact of n on the performance gap is justified. Next, the changing

trend of the light blue area in Figure 4.2 demonstrates that the gap increases when we scale up the model size for both ViT and ResNet architectures. Finally, we also find from Figure 4.3 that the increasing non-IID level contributes to the enlargement of the gap between the two scenarios. This observation is also consistent with our theoretical analysis.

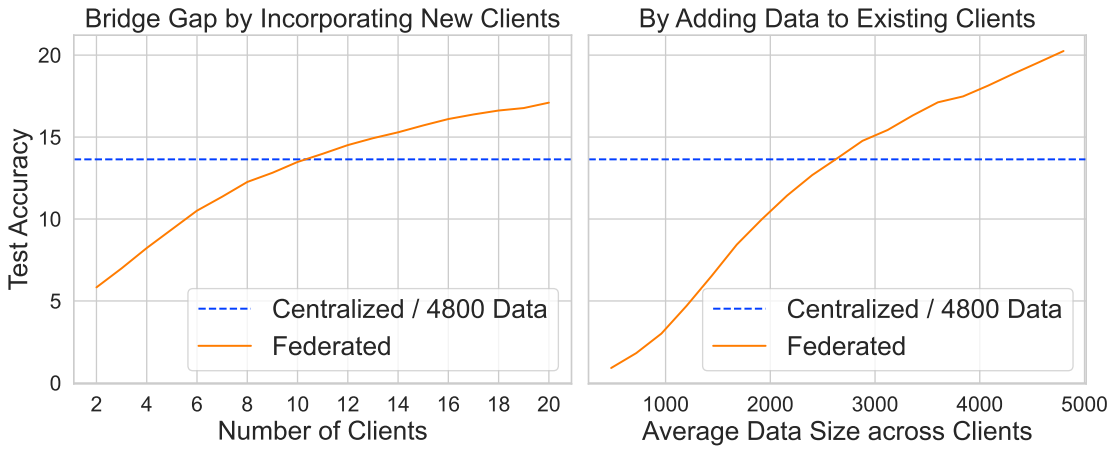


Figure 4.4: **Empirical evidence for fully closing the gap between federated and centralised training setups. (Left)** The strategy of incorporating new clients (increasing the number of clients n). **(Right)** The strategy of adding data to existing clients (increasing the average data amount m).

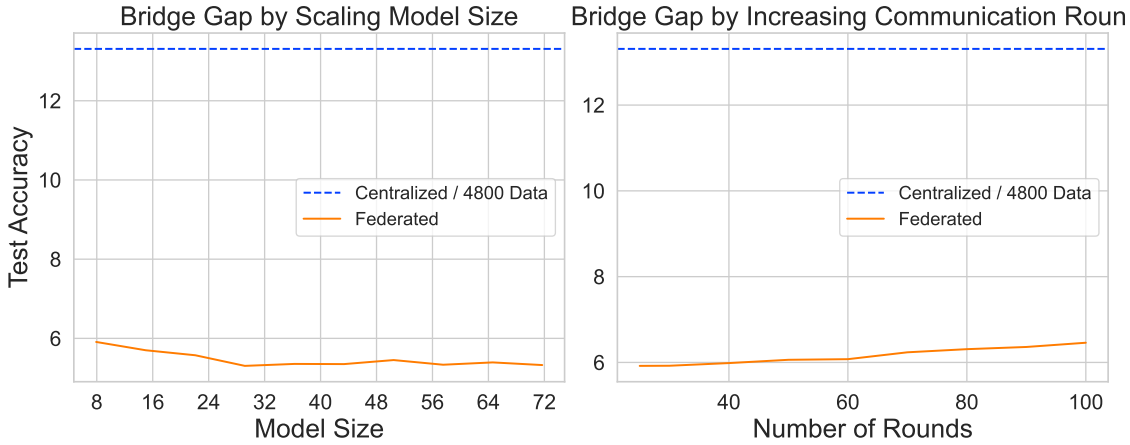


Figure 4.5: **Additional evidence for fully closing the gap.** The baseline centralised scenario contains 4800 data, aligned with the centralised scenario in the previous figure. **(Left)** The strategy of increasing d. **(Right)** The strategy of increasing communication rounds T).

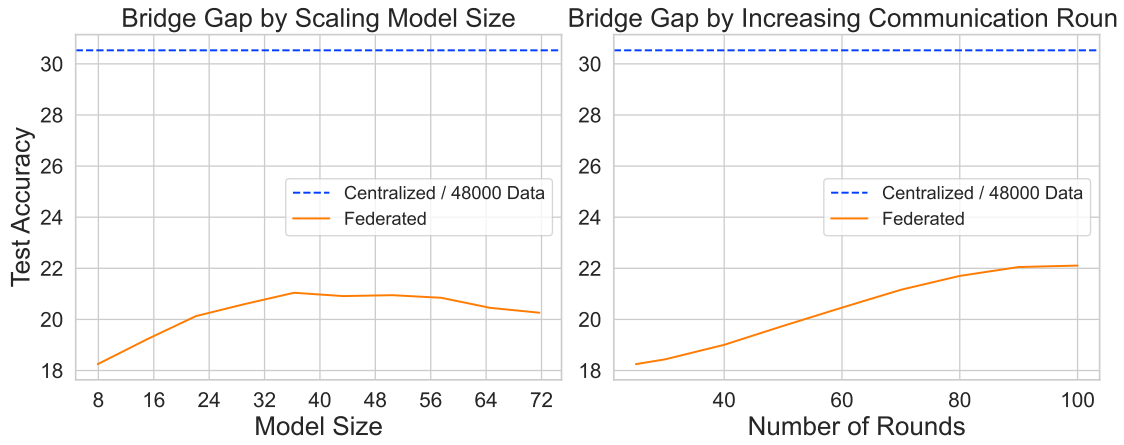


Figure 4.6: **Further evidence for fully closing the gap.** The baseline centralised scenario holds the complete training dataset containing 48000 data. **(Left)** The strategy of increasing the model size d). **(Right)** The strategy of increasing communication rounds T).

Bridge Gap by Increasing Training Resources To empirically investigate our theoretical insights about the complete elimination of the performance gap, we designed four sets of experiments for the four parameters involved in Theorems 9 and 10. In each experiment, a centralised scenario is compared with a federated scenario that holds an advantage in one kind of training resource. We start from the setting when this kind of training resource is equal between two scenarios and gradually amplify this focused parameter to check if the performance gap can be progressively closed. The results presented in Figures 4.4, 4.5 and 4.6 validate Theorems 9 and 10. Specifically, we can discover that the generalisation performance of models trained in federated setups catches up or surpasses those trained in centralised setups by either incorporating new clients or adding data to existing clients. Besides, Figure 4.4 also shows that the latter approach is more efficient in closing the gap by a steeper curve. In particular, for the same amount of increased data, scaling up the average model size by ten times (i,e., from $m = 480$ to $m = 4800$) resulted in a larger generalisation improvement than scaling up the number of clients by ten times (i,e., increasing from $n = 2$ to $n = 20$).

4.5 Chapter Conclusion

This chapter re-studies the problem that models trained in distributed setups do not perform as well as models trained in centralised setups, focusing on the theoretical exploration of this generalisation gap and valid strategies to bridge it. We began with a uniform stability analysis that established how decentralised generalisation improves with the number of clients and stronger network connectivity, but it will be upper-bounded by FL in identical conditions due to the full connectivity established through a central server. This finding justifies modelling federated learning as the representative form of distributed training when comparing against centralised training. Building on this foundation, we then derived a PAC-Bayesian formulation of the generalisation gap, expressed as the discrepancy between the generalisation error bounds of federated and centralised training. The results show that the gap provably persists under equal resources, and its magnitude depends on the training and scenario configuration. Crucially, the analysis also identified that the only effective way for distributed learning to catch up with centralised training is by gaining an advantage in the training data size, either through additional clients or by enlarging local datasets, with the latter proving more efficient. Extensive experiments across various model architectures and datasets further confirmed the correctness of these theoretical findings. Based on these theoretical results and empirical validations, the chapter not only explains why distributed training lags behind centralised training but also provides principled guidance for reducing the gap in practice.

4.6 Chapter Notations and Definitions

i, j	Client indices
n	Number of clients in the network
m	Average number of local data across clients
θ, Θ	Model parameters
d	Dimension of model parameters / Model Size
f, F	Loss function
$\mathcal{D}$	Dataset
z	Data sample
η	Learning rate
T	Communication rounds
s, t	Round indexes
$\mathcal{A}$	Algorithm
ϵ	Small constant
$L, G, \mathcal{B}$	Constants related to assumptions
K	Number of participating clients
$\mathcal{P}$	Transition matrix of communication
$c_{\mathcal{P}}$	Constant depending on $\mathcal{P}$
λ_2	Second largest eigenvalue of $\mathcal{P}$
deg	node degree in network
β	Training perturbation term
δ	probability
Q	distribution of the output hypothesis
C	Covariance matrix about gradient
A	Jacobian matrix of the gradient field
k	batch ratio
N, D	Data size
γ	Value depending on data heterogeneity
$\mathcal{G}$	Generalisation error bound
S_{Fed}, S_{Cen}	Batch sizes in federated and centralized training

CHAPTER 5

Understanding the Robustness of Distributed Self-Supervised Learning Frameworks against Non-IID Data

Chapter Overview: After establishing theoretical insights into optimal model sizes and generalisation in distributed learning, our attention now turns to distributed self-supervised learning (D-SSL), which introduces unique challenges and opportunities. D-SSL leverages large-scale unlabelled data across decentralised clients, offering great potential without costly annotation. However, its effectiveness is often undermined by heterogeneous data distributions, and there is still limited theoretical understanding of how different frameworks respond. In this chapter, we present a theoretical analysis of the robustness of D-SSL under non-IID settings. Our results show that Masked Image Modelling (MIM) exhibits stronger robustness than Contrastive Learning (CL), and that decentralised SSL becomes more robust with higher network connectivity. Moreover, federated learning, which aggregates updates globally through a server, is at least as robust as decentralised learning. Building on these insights, we propose MAR loss, a lightweight extension of the MIM loss that introduces alignment regularisation to further enhance robustness under heterogeneous data. Finally, extensive experiments across architectures and distributed scenarios validate both our theoretical findings and the effectiveness of MAR loss.

5.1 Introduction

In the previous chapters, we examined theoretical aspects of distributed learning more broadly. Building on that foundation, we now focus on distributed self-supervised learning (D-SSL), a setting that combines the promise of self-supervised representation learning with the practical reality of decentralised data. The increasing availability of large-scale unlabelled data across distributed sources, such as images collected from mobile devices or sensor networks, makes D-SSL an attractive paradigm for training models without costly annotation. At the same time, it raises fundamental challenges regarding how different D-SSL frameworks behave under heterogeneous data distributions.

Existing D-SSL frameworks can generally be distinguished in two aspects: differing by the adopted self-supervised learning (SSL) method or by the applied distributed framework. Self-supervised learning (SSL) is a widely used technique to learn representations without human-labelled annotations by solving pretext tasks that generate supervisory signals from raw data [36]. Depending on the approach used to generate supervisory signals, SSL methods are broadly categorised into Contrastive Learning (CL) and Masked Image Modelling (MIM) [87, 167], with representative methods like SimSiam [15] and MAE [41]. On the other hand, federated learning (FL) and decentralised learning (DecL) are two main frameworks in training models with distributed data [123, 143]. FL aggregates local models via a central server [100, 174], while DecL enables direct inter-client communications for aggregating models, enhancing privacy and avoiding the dependence on the central server [4, 131]..

The main obstacle facing D-SSL is data heterogeneity. In practice, the data across clients is often non-independent and non-identically distributed (non-IID), leading to degradation in both training and downstream performance. To tackle this challenge, previous works proposed advanced D-SSL algorithms with robustness to heterogeneous data. Notable examples include FedU [174], Orchestra [93], and L-DAWA [110]. However, despite continuous algorithmic innovation, there is still a

lack of theoretical understanding of this data heterogeneity problem. For example, FedU was designed within the FL framework, but how would its robustness to non-IID data change if deployed in a DecL framework without coordination from the server? Similarly, state-of-the-art D-SSL algorithms are primarily based on CL, while the adaptation of MIM methods to distributed settings remains under-explored. Could D-SSL based on MIM offer greater robustness to non-IID data than CL-based methods? These confusions converge into a fundamental research question affecting the advancement of D-SSL: ***How robust are different D-SSL frameworks against data heterogeneity?***

In this chapter, we address this question by developing mathematical models of D-SSL algorithms under a simplified non-IID Setting and analysing the representations they produce. Our analysis reveals that MIM-based approaches are inherently more robust than CL-based ones, regardless of whether the underlying framework is federated or decentralised, although robustness remains limited under severe heterogeneity. We also show that the robustness of decentralised SSL grows with network connectivity, and that federated SSL performs on par with decentralised SSL in the case of full connectivity. Motivated by these findings, we introduce a refined MIM objective, MAR loss, which improves robustness by encouraging local-to-global representation alignment. To empirically validate our theoretical insights and the proposed method, extensive experiments were conducted using ResNet [43] and Vision Transformer (ViT) [22]. We pre-trained these models with different D-SSL frameworks across varying levels of heterogeneity and evaluated their fine-tuning performance on multiple benchmark datasets. Through theoretical analysis and empirical validation, this chapter lays the foundation for understanding and improving the robustness of D-SSL algorithms.

5.2 Related Work

Self-supervised Learning. Self-supervised learning (SSL) leverages unlabelled data by generating pseudo labels from raw inputs to learn meaningful representations

[36]. Vision-based SSL methods are typically categorised into contrastive learning (CL) and masked image modelling (MIM) [87, 167]. CL learns representations by maximising the similarity between positive pairs (i.e., similar data points created by data augmentation) and minimising it between negative pairs (i.e., data pairs created by other data points) [14, 42]. Recent methods like SimSiam [15] and BYOL [35] advance the original contrastive loss by removing terms related to negative pairs, which improves stability and reduces batch size dependence. MIM, in contrast, randomly masks out patches of input images and predicts the missing parts, learning representations through a reconstruction loss [5, 41, 155, 171]. Although different in formulation, recent studies have shown that many MIM methods have close connections to CL (i.e., their objectives can be directly re-formulated as contrastive loss [68, 167]). In this chapter, we aim to figure out which SSL paradigm is inherently more robust against data heterogeneity.

Distributed Learning. Distributed learning enables collaborative model training across multiple clients without sharing data. Two dominant frameworks in this area are: federated learning (FL), which uses a central server to coordinate and aggregate models [100], and decentralised learning (DecL), where clients exchange models locally with neighbours [4, 131]. While FL is more widely adopted [166] for better convergence and training effectiveness, DecL offers benefits in scalability and privacy. Recent studies have started comparing these two frameworks [6, 45]. For example, Sun et al. explored which leads to better generalisation performance and the impact of network architecture on generalisation [123]. However, the relationship between network architecture and the robustness against heterogeneous data in distributed settings is still unclear. This chapter addresses this gap by providing both theoretical analysis and empirical findings to clarify this relationship.

Distributed SSL. Distributed SSL (D-SSL) integrates SSL with distributed frameworks to leverage unlabelled, decentralised data while preserving privacy [158, 174]. A core challenge is learning robust representations under data heterogeneity

[172]. Prior work has primarily focused on algorithmic solutions such as FedU [174] and L-DAWA [110]. Although some studies also provide theoretical analyses, their purpose is to demonstrate the validity of the proposed algorithms rather than to advance the understanding of the robustness variance between different D-SSL frameworks [58, 93]. The most relevant theoretical work is by Wang et al., who showed that SSL is more robust than supervised learning in distributed settings [149]. Unfortunately, their study only theoretically analysed a specific case of D-SSL where CL is combined with FL and did not extend it to other types of D-SSL frameworks. In contrast, we delve deeper into these differences, shedding light on the insensitivity of various D-SSL approaches under heterogeneous conditions.

5.3 Problem Setup

To provide theoretical insights on understanding this central question, we first introduce our problem setup about distributed training and D-SSL with heterogeneous client data.

5.3.1 Distributed Training

Distributed Setting. Consider a distributed scenario consisting of a connected network of N clients, represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of clients and $\mathcal{E}$ is the set of edges denoting direct communication links between clients. The connectivity of the graph is captured by a matrix $A \in \mathbb{R}^{N \times N}$, referred to as the adjacency matrix, where A_i denotes the set including client $i \in [N]$ itself and its neighbours shown by $\mathcal{E}$, $|A_i|$ represents the size of this neighbourhood set or the connectivity of client i, and $|\bar{A}| = \frac{1}{n}\sum_{i=1}^{n}(|A_i|)$ is the average connectivity between clients. Hence, distributed training conducted through the decentralised framework satisfies $\forall i \in [N], 2 \leq |A_i| \leq N$. In contrast, the federated learning framework relies on a central server that aggregates local models from all clients and broadcasts the global model back to them in each round, as in FedAvg [100]. This architecture effectively enables every client to communicate with all others

through the server, which corresponds to a fully connected decentralised topology where $\forall i \in [N], |A_i| = N$. From a graph perspective, federated learning can therefore be interpreted as a special case of the graph-based communication model, where the communication structure follows a star topology centred at the server. A more formal specification of the graph structure and the mixing-weight conditions for this distributed setting is provided in Section 8.3.1.

Objective of Distributed optimisation. To utilise different clients to learn useful representations, distributed training generally optimises the global objectives below:

$$W^*_{Dec} = \min_W \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|A_i|} \sum_{j \in A_i} \mathcal{L}_j(W_j); \quad W^*_{Fed} = \min_W \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_i(W_i) \tag{5.1}$$

where $\mathcal{L}_j$ is the objective of local SSL on client j, W^*_{Dec} and W^*_{Fed} denote the global objective of DecL and FL, respectively. In particular, at each iteration of DecL, each client conducts local updates using the local dataset and aggregates the updated local model with those from neighbours [131]. For generating the global model for downstream tasks, there will be an additional aggregation on all local models after all iterations. Differently, the optimisation of FL involves each round of model aggregation only on the central server [100]. Then, the server broadcasts the global model to all clients for the next round of training. Note that the FL framework does not need another aggregation between all local models since the updated global model on the server can be used directly for fine-tuning.

5.3.2 Rigorous Analysis of D-SSL on a Simplified Non-IID Setting

Non-IID Client Data. D-SSL involves all clients collaboratively training a global model by leveraging their local unlabelled datasets $\{D_i\}_{i=1}^N$ and communicating over the graph $\mathcal{G}$. Since sharing data is prohibited to protect privacy, the heterogeneity across these distributed data sources generally leads to a performance drop in many distributed applications [100, 174]. Two common types of data heterogeneity are:

feature heterogeneity and label heterogeneity [172]. Although D-SSL operates on unlabelled data, semantic heterogeneity across clients still arises from differences in the underlying data sources and object categories. Consequently, label distribution skew is commonly used as a convenient abstraction for modelling statistical heterogeneity in distributed datasets, since variations in class distributions typically correspond to shifts in the underlying feature distributions learned by self-supervised models. Following this intuition, in this paper, we refer to previous works [87, 149] to model a simplified but formal label non-IIDness between local datasets as follows. The global data distribution $D = \bigcup_{i=1}^{N} D_i$ across clients is assumed to contain unlabelled data from $2N$ classes. For the dataset on client i, the local data distribution D_i is constrained and imbalanced on three classes, with most samples belonging to classes $2i-1$ and $2i$, while the remaining very few samples come from the class $h_i \in [2N] \setminus \{2i-1, 2i\}$. Specifically, for a sufficiently large positive integer $d > 0$, let $x \in \mathbb{R}^d \sim D_i$ be the data points in the local dataset and $e_1, \ldots, e_d$ be the standard unit-norm vectors of the d-dimensional Euclidean space. For class $2i-1$, we set $x^{(2i-1)} = e_i - \Sigma_{k\neq i,k=1}^{N} q^{(2i-1,k)} \tau e_k + \mu \xi^{(2i-1)}$, where τ and μ are two positive hyperparameters, q is sampled uniformly from $\{0, 1\}$ and $\xi \sim \mathcal{N}(0, I)$ from Gaussian distribution. Likewise, for class $2i$, we define $x^{(2i)} = -e_i - \Sigma_{k\neq i,k=1}^{N} q^{(2i,k)} \tau e_k + \mu \xi^{(2i)}$. The size of the data from classes $2i-1$ and $2i$ are equal and both grow in polynomials of d. For infrequent class h_i, the samples are generated as: $x^{(h_i)} = e_{h_i} + \mu \xi^{(h_i)}$ and the amount of data is sublinear in d, denoted as $O(d^\alpha)$ with $\alpha \in (0, 1)$). Furthermore, we assume all N local datasets to have an equal total number of samples, i.e., $|D_1| = |D_2| = \ldots = |D_N|$. To facilitate understanding, we provide an overview of this non-IID data distribution in Figure 5.1. Next, we consider CL and MIM as two main paradigms of SSL and formulate CL and MIM, respectively.

CL Formulation. For CL, we adopt the more advanced SimSiam [15] which trains with only the positive pairs $(g_a(x), g_b(x))$, where $g_a(\cdot)$ and $g_b(\cdot)$ are random augmentations drawn from SimSiam's augmentation policy (e.g., Gaussian noise,

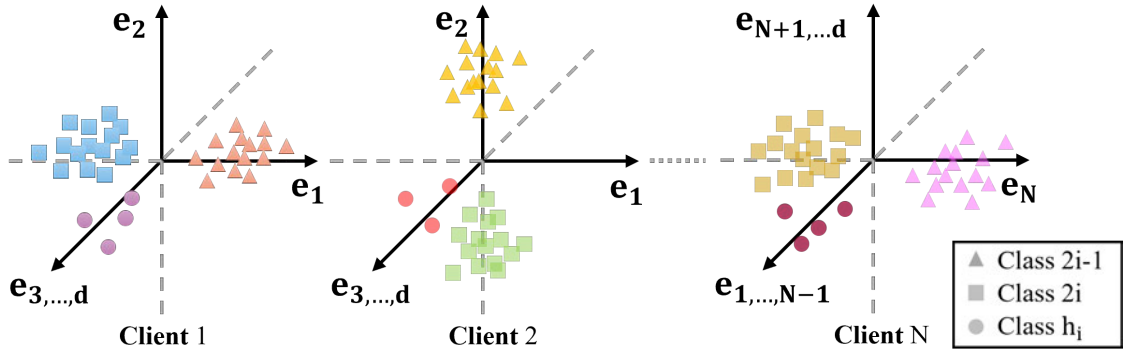


Figure 5.1: **Illustration of the constructed heterogeneous distribution for local data on clients.** Each client holds two unique data classes.

flipping). Consider a linear embedding function $f_W(x) = Wx$, where the weight matrix W satisfies $W \in \mathbb{R}^{c\times d}$ and $c \geq 2N$ according to the distributed settings, the local objective on client i is defined as below:

$$\mathcal{L}_{CL} = -\mathbb{E}_{x\sim D_i}||(W(g_a(x)))^\intercal(W(g_b(x)))||^2 + \frac{1}{2}||W^\intercal W||_F^2. \tag{5.2}$$

Eq.(5.2) captures the SimSiam loss by utilising the negative inner product $\langle a, b\rangle$ to measure the distance between the positive pairs. This objective also excludes a feature predictor for simplicity and includes a regularisation term $||W^\intercal W||_F^2$ for more mathematically tractable, similar to previous works [87, 149]. Note that Eq.(5.2) stands for a general form of SimSiam loss due to the wide class of augmentation functions [36]. For a detailed and tractable theoretical exploration, we consider the linear formulation of data augmentation and further differentiate CL by the similarity between $g_a(\cdot)$ and $g_b(\cdot)$. In particular, for the case where the positive pairs are generated by similar augmentations, the objective becomes:

$$\mathcal{L}_{CL} = -\mathbb{E}_{x\sim D_i}||(W(x+\xi))^\intercal(W(x+\xi'))||^2 + \frac{1}{2}||W^\intercal W||_F^2, \tag{5.3}$$

where $\xi, \xi' \sim \mathcal{N}(0, I)$ are random noise sampled IID from the Gaussian distribution. On the other hand, when $g_a(\cdot)$ and $g_b(\cdot)$ are different, we define the loss with the

following form:

$$\mathcal{L}'_{CL} = -\mathbb{E}_{x\sim D_i}||(W(x+\xi))^\intercal(W(Hx))||^2 + \frac{1}{2}||W^\intercal W||_F^2, \tag{5.4}$$

where $H \in \mathbb{R}^{d\times d}$ denotes a linear image transformation (e.g., rotation, translation, horizontal or vertical flip, etc.). The formal conditions on H are given in Section 8.3.1

MIM Formulation. For MIM, a random binary mask $m \in \{0,1\}^d$ (created by uniformly sampling 0 with probability p, i.e., mask ratio) is applied to partition the input x into two complementary views: the unmasked part $x_1 = x \odot m$ and the masked part $x_2 = x \odot (1-m)$ satisfying $x_1 + x_2 = x$. Then, we train an encoder-decoder model $f = f_d \circ f_e$, where the encoder f_e encodes the input x_1 to a latent representation $z = f_e(x_1)$, and the decoder f_d decodes z back to pixel space to reconstruct the masked part x_2. Hence, considering a linear encoder and decoder with embedding matrix $W_e \in \mathbb{R}^{c\times d}$ and $W_d \in \mathbb{R}^{d\times c}$, the local objective of MIM is given by

$$\mathcal{L}_{MIM} = \mathbb{E}_{x\sim D_i}\mathbb{E}_{x_1,x_2|x}||f_d(f_e(x_1)) - x_2||^2 = \mathbb{E}_{x\sim D_i}||W_dW_e(x\odot m) - (x\odot(1-m))||^2, \tag{5.5}$$

where the mean square error (MSE) loss is utilized to enforce the reconstructed image to be similar to the original image, and $\odot$ denotes the Hadamard product. Recent studies have focused on the connection between MIM and contrastive losses and found that the MIM reconstruction objective admits an alignment between the masked and unmasked parts [68, 167]. Based on these results, we adopt an alignment-style formulation of Eq.(5.5) with $W := W_e \in \mathbb{R}^{c\times d}$:

$$\mathcal{L}_{MIM} = -\mathbb{E}_{x\sim D_i}[(W(x\odot m))^\intercal(W(x\odot(1-m)))] + \frac{1}{2}||W^\intercal W||_F^2, \tag{5.6}$$

which implicitly aligns the masked and unmasked views in the embedding space. The regularization term $||W^\intercal W||_F^2$ is also introduced to ensure a well-posed quadratic form and improve the traceability.

5.4 Theoretical Insights

In this section, we use the above problem setup to model different D-SSL frameworks and compare their robustness to heterogeneous data. Our analysis is based on the observation that the robustness of distributed SSL to non-IID data can be reflected in the representations learned under different training objectives and network architectures. Under the non-IID data model introduced above, we first analyse how different SSL paradigms and distributed communication structures influence the learned representations. We then compare these representations to characterise their sensitivity to heterogeneous data and derive the corresponding theoretical insights on robustness. The complete proof of our analysis is provided in Section 8.3.

5.4.1 Analysis of Representations Learned by D-SSL

We begin our theoretical analysis with the following definition of the representability of the learned representation.

Definition 2. *(Representability Vector (RV)). Let $\{e_1, \ldots, e_d\}$ be the standard basis of $\mathbb{R}^d$. Let $W = [w_1, \ldots, w_c]^\intercal \in \mathbb{R}^{c \times d}$ be the feature matrix learned by the linear embedding function $f_W(x) = Wx$, where $c \leq d$. For row space $\mathcal{R} = \text{row}(W) \subseteq \mathbb{R}^d$, we denote the representability of $\mathcal{R}$ as a vector $r = [||\Pi_{\mathcal{R}}(e_1)||_2^2, \ldots, ||\Pi_{\mathcal{R}}(e_d)||_2^2]^\intercal$, where $\Pi_{\mathcal{R}}(e_k)$ is the projection of e_k onto $\mathcal{R}$ for $k \in [d]$. Hence, we have $||\Pi_{\mathcal{R}}(e_k)||_2^2 = \sum_{j=1}^{c} (e_k^\intercal v_j)^2$, where $\{v_1, \ldots, v_c\}$ is any orthonormal basis of $\mathcal{R}$.*

The intuition behind this definition is that for any input vectors $x \in \mathbb{R}^d$, the learned feature space should have a good representation of the standard basis vectors, $e_1, \ldots, e_d$, to perform well. In particular, these basis vectors should have large projections onto the feature space. The introduction of the representability vector

allows us to quantitatively assess the feature space learned by different D-SSL frameworks. Similar definitions and notations have also been used in previous works studying the feature space of SSL [87, 149]. Based on this definition and the above problem setup, we establish the following theorem for D-SSL frameworks that are based on MIM pre-training.

Theorem 11. *(Representability of Distributed MIM). Consider a distributed scenario consisting of* $N = \Theta(d^{\frac{1}{20}})$ *clients and following the above non-IID setup with* $\tau = d^{\frac{1}{5}}$ *and* $\mu = d^{-\frac{1}{5}}$*. For distributed SSL that utilises Masked Image Modelling (MIM) as the pre-training approach, with a high probability, the following statements hold:*

1. *Let* $r_i^M = [r_{i,1}^M, \dots, r_{i,c}^M]^\intercal$ *be the local RV learned on client* i*, then we have* $1 - \frac{O(d^{-\frac{2}{5}})}{2p(1-p)d^{\frac{2}{5}} + O(d^{-\frac{2}{5}})} \leq r_{i,k}^M \leq 1$*, where* $i \in [N] \backslash k$*.*
2. *Let* $\bar{r}_{Dec}^M = [\bar{r}_1^M, \dots, \bar{r}_c^M]^\intercal$ *be the RV learned through the global objective of DecL framework, then we have* $1 - \frac{O(d^{-\frac{2}{5}})}{2p(1-p)(1-1/|\bar{A}|)d^{\frac{2}{5}} + O(d^{-\frac{2}{5}})} \leq \bar{r}_{Dec}^M \leq 1$*; while for the RV* $\bar{r}_{Fed}^M = [\bar{\boldsymbol{r}}_1^M, \dots, \bar{\boldsymbol{r}}_c^M]^\intercal$ *learned through the FL framework, we have* $1 - \frac{O(d^{-\frac{2}{5}})}{2p(1-p)d^{\frac{2}{5}} - \Theta(d^{\frac{7}{20}}) + O(d^{-\frac{2}{5}})} \leq \bar{r}_{Fed}^M \leq 1$*.*

Theorem 11 shows the status of the feature space learned by distributed MIM with different objectives (i.e., local vs decentralised global vs federated global). Note that for each provided representability vector, we find a unique lower bound and a shared upper bound (considering $\sum_{j=1}^{d} (e_k^\intercal e_j)^2 = 1$). The distance between the lower and upper bound states how much the learned representation fluctuates in the c unit directions, $e_1, \dots, e_c$, associated with data generation. Therefore, the smaller the distance, the less sensitive the representation space is to the non-IID distribution of local datasets on clients. In other words, the corresponding D-SSL is more robust to data heterogeneity.

By a similar proof, we derive the representability vectors for D-SSL methods with CL pre-training as follows.

Theorem 12. *(Representability of Distributed CL). Consider the same distributed scenario in Theorem 11. For distributed SSL that utilises Contrastive Learning (CL) as the pre-training approach, with a high probability, the following statements hold:*

1. *Let* $r_i^C = [r_{i,1}^C, \ldots, r_{i,c}^C]^\intercal$ *be the local RV learned on client* i*. If positive pairs are generated by similar augmentations, we have* $1 - \frac{O(d^{-\frac{1}{5}})}{d^{\frac{2}{5}} + O(d^{-\frac{1}{5}})} \leq r_{i,k}^C \leq 1$*, where* $i \in [N] \backslash k$*. Otherwise, we have* $1 - \frac{O(d^{-\frac{1}{5}})}{tr(H) d^{\frac{2}{5}} + O(d^{-\frac{1}{5}})} \leq r_{i,k}^C \leq 1$ *for dissimilar augmentations, where* $tr(H)$ *denotes the trace of image transform matrix* H*.*

2. *Let* $\bar{r}_{Dec}^C = [\bar{r}_1^C, \ldots, \bar{r}_c^C]^\intercal$ *be the RV learned through the global objective of DecL framework, then we have* $1 - \frac{O(d^{-\frac{1}{5}})}{(1-1/|\bar{A}|) d^{\frac{2}{5}} + O(d^{-\frac{1}{5}})} \leq \bar{r}_{Dec}^C \leq 1$ *and* $1 - \frac{O(d^{-\frac{1}{5}})}{tr(H)(1-1/|\bar{A}|) d^{\frac{2}{5}} + O(d^{-\frac{1}{5}})} \leq \bar{r}_{Dec}^C \leq 1$ *for similar and dissimilar augmentations, respectively; while for the RV* $\bar{r}_{Fed}^C = [\bar{\boldsymbol{r}}_1^C, \ldots, \bar{\boldsymbol{r}}_c^C]^\intercal$ *learned through the FL framework, we have* $1 - \frac{O(d^{-\frac{1}{5}})}{d^{\frac{2}{5}} - \Theta(d^{\frac{7}{20}}) + O(d^{-\frac{1}{5}})} \leq \bar{r}_{Fed}^C \leq 1$ *and* $1 - \frac{O(d^{-\frac{1}{5}})}{tr(H) d^{\frac{2}{5}} - d^{\frac{2}{5}} + O(d^{-\frac{1}{5}})} \leq \bar{r}_{Fed}^C \leq 1$*.*

Theorem 12 demonstrates that the local and global feature spaces learned by distributed CL are distinct from those learned by distributed MIM. However, it is not obvious which feature spaces hold a smaller gap between the lower and upper bounds. To determine which type of pre-training is less sensitive to data heterogeneity, we further compare their global feature spaces learned in DecL and FL framework, respectively, and summarize the results in the following theorem.

5.4.2 MIM is Inherently More Robust than CL with Heterogeneous Data

Theorem 13. *Let* $s = \lceil \bar{r} \rceil - \lfloor \bar{r} \rfloor$ *be the sensitivity of distributed SSL to heterogeneous data* $x \in \mathbb{R}^d$*, with* $\lfloor \bar{r} \rfloor$ *and* $\lceil \bar{r} \rceil$ *to denote the lower and upper bound of the learned global representability vector* $\bar{r}$*. For any network architecture, distributed SSL satisfies the following property:* $\lim_{d \to \infty} [s^C > s^M]$*, where* s^C *and* s^M *represent the sensitivities of distributed SSL adopting contrastive learning and masked image modelling as the pre-training approach, respectively.*

The main intuition for the greater robustness (or smaller sensitivity) of distributed MIM is that CL learns representations from aligning features of the positive pair generated from the original data through data augmentation, whereas MIM aligns features of the reconstructed and the raw data to learn representations. Although the applied augmentation generally does not lead to a change in data labels [14, 15], the output is still a different image. In contrast, the masking operation splits the original image into the masked and unmasked parts, but a portion of the original data is retained in both parts. As a result, CL learns a local representation with greater randomness, and that additional randomness is also biased by local labels. Considering that data heterogeneity already exists among clients, the global representation learned by distributed CL is less uniform than that learned by distributed MIM.

5.4.3 Impact of the Average Connectivity on Non-IID Robustness

Next, we shift our focus to another dimension that distinguishes D-SSL algorithms and address the question: how does the network architecture affect the robustness of the feature space learned by D-SSL? The tool for solving this question is again the bounds of the representability vector. For the DecL setup where clients directly communicate with their direct neighbours, Theorem 11 and 12 have implicitly shown the answer.

Corollary 2. *For any SSL pre-training approaches, if the distributed scenario is fully decentralised (i.e., without a central server), the robustness of distributed SSL against heterogeneous local data improves with the average connectivity $|\bar{A}|$ between clients in the network.*

Corollary 2 also implies that the robustness of D-SSL conducted in a federated setup should be no worse than in a fully decentralised network. Consider the best case of the network topology, where each client can communicate with all other clients in the network. In this case, each client receives a model aggregated by the local models from all clients, which is exactly the global model distributed by the

server in the federated setup. We can continue exploring to verify that this intuition is correct. Theoretically, combining Theorem 11, Theorem 12, and Corollary 2, we arrive at another main theorem addressing the question introduced at the beginning of this section.

Theorem 14. *For any SSL pre-training paradigms, distributed SSL satisfies the following property:* $\lim_{d\to\infty}[s_{Dec} \geq s_{Fed}]$*, where* $s_{Dec} = \max_{k\in[c]} \bar{r}^{(k)}_{Dec} - \min_{k\in[c]} \bar{r}^{(k)}_{Dec}$ *denotes the sensitivity of distributed SSL performed in the DecL setup (i.e., clients directly communicate with neighbors), and* $s_{Fed} = \max_{k\in[c]} \bar{r}^{(k)}_{Fed} - \min_{k\in[c]} \bar{r}^{(k)}_{Fed}$ *represents the sensitivity of distributed SSL performed in the FL setup (i.e., all clients are indirectly connected through the central server).*

This theorem further demonstrates the robustness trade-off between applying SSL in federated and decentralised frameworks. For less concern about the impact of data heterogeneity, we should conduct distributed SSL in a federated setup (often also referred to as federated self-supervised learning [93, 110, 174, 175]). However, the decentralised case is more common in reality, as it is challenging to provide a central server that can be trusted by all clients and has stable communication with them. Then, we can consider increasing the average connectivity between clients in the network to minimise the negative impact of heterogeneous data on training (e.g., identifying under-connected clients and creating new direct communication links).

5.5 MAR Loss: Improving the Robustness of Distributed MIM to Data Heterogeneity with Local-to-Global Alignment Regularisation

The preceding analysis has addressed the main focus of this paper by establishing theoretical insights into the robustness of different D-SSL frameworks under heterogeneous data. As a further step, we illustrate how these insights can guide a more robust algorithmic design. In particular, our results show that although distributed MIM is fundamentally more robust than CL, its training dynamics

Algorithm 1 FedMAR Algorithm

Input: initial model W^0, number of local updates E, number of training rounds T, learning rate η, the upper bound of regularisation weight $\gamma_{\max}$, the lower bound $\gamma_{\min}$
Output: optimised global model W^T

for $t = 0, \ldots, T-1$ **do**
  if $t = 0$ **then**
    server broadcasts W^t to $\mathcal{C} \sim [N]$
  else
    computes $\gamma_t^{(i)}$ by $\gamma_{\max}$, $\gamma_{\min}$ on server (Eq.(5.9))
    server broadcasts $W^t, \bar{z}, \gamma_t^{(i)}$ to $\mathcal{C} \sim [N]$
  for client $i \in \mathcal{C}$ in parallel **do**
    $W_{i,0}^t \leftarrow W^t$
    if $t = 0$ **then**
      $W_{i,E}^t, z_i \leftarrow SGD(W_{i,0}^t, \eta, E, \mathcal{L}_{MIM})$
    else
      $W_{i,E}^t, z_i \leftarrow SGD(W_{i,0}^t, \eta, E, \mathcal{L}_{MAR}(\bar{z}, \gamma_t^{(i)}))$ (Eq.(5.7))
    sends $W_{i,E}^t, z_i$ to server
  $\bar{z} = \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} z_i$
  $W^{t+1} \leftarrow \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} W_{i,E}^t$

are dominated by the client-specific covariance, causing local encoders to drift toward different directions before aggregation gradually mitigates this effect. This observation motivates us to refine the MIM objective with an additional term that explicitly and dynamically promotes consistency between local and global masked representations, which we term MAR loss. The integration of MAR into both federated and decentralized frameworks is summarized in Algorithms 1 and 2.

Formally, MAR loss augments the MIM objective with an alignment regularization term:

$$\mathcal{L}_{MAR} = \mathbb{E}_{x \sim D_i} \mathbb{E}_{x_1, x_2 | x} \Big[\| f_d(f_e(x_1)) - x_2 \|^2 + \gamma_t^{(i)} \cdot \text{A-MMD}(z_i, \bar{z}) \Big], \tag{5.7}$$

where $z_i = f_e(x_1)$ and $\bar{z}$ denote the local masked and global representations, and $\gamma_t^{(i)} > 0$ is a dynamic weight for alignment. The alignment regularizer is based on *Maximum Mean Discrepancy (MMD)*, a widely used measure of distributional discrepancy in machine learning [31,34,77]. MMD compares whether two distributions

Algorithm 2 DecMAR Algorithm

Input: initial models $W_{i,E}^{-1}$, number of local updates E, number of training rounds T, learning rate η, the upper bound of regularisation weight $\gamma_{\max}$, the lower bound $\gamma_{\min}$
Output: optimised global model W^T

for $t = 0, \ldots, T-1$ **do**
  for client $i \in [N]$ in parallel **do**
    if $t = 0$ **then**
      send $W_{i,E}^{t-1}$ to its neighbours
    else
      computes $\gamma_t^{(i)}$ by $\gamma_{\max}$, $\gamma_{\min}$ for each neighbour (Eq.(5.9))
      send $W_{i,E}^{t-1}, z_i, \gamma_t^{(i)}$ to its neighbours
      $\bar{z} = \frac{1}{|A_i|} \sum_{j \in A_i} z_j$
    $W_{i,0}^t \leftarrow \frac{1}{|A_i|} \sum_{j \in A_i} W_{j,0}^{t-1}$
    if $t = 0$ **then**
      $W_{i,E}^t, z_i \leftarrow SGD(W_{i,0}^t, \eta, E, \mathcal{L}_{MIM})$
    else
      $W_{i,E}^t, z_i \leftarrow SGD(W_{i,0}^t, \eta, E, \mathcal{L}_{MAR}(\bar{z}, \gamma_t^{(i)}))$ (Eq.(5.7))
$W^T \leftarrow \frac{1}{N} \sum_{i \in [N]} W_{i,E}^{T-1}$

P and Q differ by mapping samples into a reproducing kernel Hilbert space (RKHS) and evaluating differences in their feature means. Typically, MMD adopts a Gaussian kernel $k(x, x') = \exp(-\|x - x'\|^2 / 2\sigma^2)$.

In MAR, we employ an adaptive version (A-MMD) to compare the feature spaces of local and global representations more robustly. Unlike prior FL works that use vanilla MMD [52, 84, 94], A-MMD selects the kernel bandwidth automatically rather than fixing it. Given batches of local and global embeddings of equal size B, A-MMD is computed as:

$$\text{A-MMD}(z_i, \bar{z}) = \frac{1}{B(B-1)} \left(\sum_{a \neq b} k(z_{i,a}, z_{i,b}) + \sum_{a \neq b} k(\bar{z}_a, \bar{z}_b) \right) - \frac{2}{B^2} \sum_{a=1}^{B} \sum_{b=1}^{B} k(z_{i,a}, \bar{z}_b), \tag{5.8}$$

with the adaptive kernel defined as $k(z, z') = exp(-\frac{||z - z'||}{2(\text{mean}_{a \neq b} ||z_a - z_b||)^2})$. This data-driven choice ensures stability across non-IID clients by scaling the kernel to the observed embedding distribution.

Finally, to balance early-stage consensus and late-stage efficiency, we design the regularization weight $\gamma_t^{(i)}$ to decay smoothly from $\gamma_{\max}$ to $\gamma_{\min}$. We adopt a cosine

schedule based on client participation:

$$\gamma_t^{(i)} = \gamma_{\min} + (\gamma_{\max} - \gamma_{\min}) \cdot \frac{1}{2}\left(1 + \cos \frac{\pi \cdot \omega_t^{(i)}}{\Omega}\right), \tag{5.9}$$

where $\omega_t^{(i)}$ counts the number of times client i has been selected up to round t, and Ω controls the decay horizon. In DecL, where all clients participate every round, one can simply set $\Omega = T$. In FL with partial participation, a practical choice is the expected number of selections per client, or T as a default. This schedule applies stronger alignment at the early training rounds when client divergence is most pronounced, and gradually relaxes toward $\gamma_{\min}$ as training progresses, ensuring dynamic robustness gains.

5.6 Experiments

In this section, we conduct extensive experiments to validate the correctness of our derived theoretical insights and evaluate the effectiveness of the MAR loss in improving the robustness of distributed MIM against data heterogeneity. We first introduce the experimental setup. Then we assess our results in different datasets, model backbones, and distributed settings.

5.6.1 Experimental Setup

Datasets and Distributed Simulation. We pre-train our models on the Mini-ImageNet dataset [145], which contains 60,000 images extracted from the ImageNet dataset [19]. To simulate a distributed scenario with label non-IIDness, the dataset is partitioned by sampling the class priors of the Dirichlet distribution [51]. A more heterogeneous division can be made with a smaller Dirichlet parameter α during sampling, while the IID case is simulated by setting a very large α. Besides, we follow prior works to simulate feature heterogeneity by uniformly dividing datasets and applying unique data augmentation for each client [149, 172]. Hence, the labels of local data are kept the same but features are skewed into different domains

before training. Furthermore, to simulate the DecL setup, we use the Erdős-Rényi model [28] to initialise a connected network with the number of clients and the average connectivity as inputs and return the adjacency matrix A. For FL, we additionally assume there exists a central server that can communicate with all the clients in this network. After pre-training, the models' backbones are fine-tuned on benchmark datasets, including CIFAR-10, CIFAR-100 [70], and ImageNet [19] dataset. We collect their fine-tuning accuracies for our analysis.

Table 5.1: **Experiment Settings of Chapter 5.**

	Details
Model Architecture	ResNet [43], Vision Transformer (ViT) [22]
Number of layers in ResNet	18
Number of blocks in ViT	5
Pre-training Method	MAE [41], SimSiam [15]
Pre-training Dataset	Mini-ImageNet [145]
Fine-tuning Dataset	CIFAR-10/100 [70], ImageNet [19]
Non-IID Options (i.e. the value of α)	$\{1e5$ (IID), $1, 0.1, 0.01, 0.001\}$
Options for the γ used in MAR loss	$\{1, 0.1, 0.01, 0.001\}$
For Federated Learning (FL):	
Number of clients	100
Number of sampled clients per round	5
Number of local training epochs	2
Number of total training rounds	100
For Decentralised Learning (DecL):	
Number of clients	20
Options for average connectivity	$3, 5, 10, 20$ (equals to FL)
Number of local training epochs	1
Number of total training rounds	25
Fine-tuning Epochs	50/100 (CIFAR-10/100), 20/100 (ImageNet)
Pre-train Batch Size	128
Fine-tune Batch Size	256 (CIFAR-10/100), 1024 (ImageNet)
Base Learning Rate	1.5e-4

Implementation Details. For our experiments, we use ResNet [43] and Vision Transformer (ViT) [22] as the model architecture. Following the problem setup in theoretical analysis, we select SimSiam [15] and MAE [41] as the representatives of CL and MIM pre-training, respectively. In original works, SimSiam is used to pre-train ResNet models, while MAE is used to pre-train ViTs. We implement

two new SSL baselines to show that our theoretical insights apply to any model architecture. One uses SimSiam to pre-train ViTs, and the other one pre-trains ResNet through MAE. Furthermore, we follow the classical distributed algorithms, D-PSGD [82] and FedAvg [100], to implement the DecL and FL frameworks, and then implement our FedMAR and DecMAR algorithms based on these frameworks. All our codes are implemented in Python using the PyTorch framework and executed on a server with 4 NVIDIA® RTX 3090 GPUs. The detailed training setup and server configuration can be found in Tables 5.1 and 5.2.

Table 5.2: **Server Settings of Chapter 5.**

Config	Details
Server GPU Count	4
Server GPU Type	RTX 3090 (24GB)
Server CPU Type	AMD EPYC 7282 16-Core
CUDA	12.4
Framework	PyTorch

5.6.2 Empirical Validation of Theory

Insensitivity Superiority of Distributed MIM. Table 5.3 compares the impact of data heterogeneity on the pre-training effectiveness between distributed MIM and CL. With highly heterogeneous data, the learned local feature space will be significantly different across clients, resulting in a greater divergence between local and global feature space and a larger drop in performance compared to the IID setup [93, 174, 175]. Across various datasets and backbone architectures, we observe that distributed MIM consistently exhibits a smaller performance gap between IID and non-IID settings compared to distributed CL. The experimental results align with Theorem 13, verifying that MIM is less sensitive than CL when handling heterogeneous data in distributed scenarios.

Besides Table 5.3 demonstrating the non-IID robustness of distributed CL and MIM by the gap in fine-tuning accuracy, we further explore the differences in their learned features empirically. Specifically, we simulate a heterogeneous setting with 100 clients using a Dirichlet sampling with $\alpha = 0.1$. For each D-SSL framework, we

Table 5.3: **Fine-tuning accuracy (%) of backbones pre-trained by different D-SSL algorithms.** All results are the mean of three trials (L/non-IID = Label Non-IID; F/non-IID = Feature Non-IID). The values in brackets denote the gap between IID and non-IID performance.

	CIFAR-10		
	IID	L/non-IID	F/non-IID
SimSiam + CNN	86.03	84.33 (↓1.70)	84.62 (↓1.41)
MAE + CNN	87.28	86.97 (↓**0.31**)	86.17 (↓**1.11**)
SimSiam + ViT	72.32	69.50 (↓2.82)	70.66 (↓1.66)
MAE + ViT	69.90	68.20 (↓**1.70**)	69.32 (↓**0.58**)

	CIFAR-100		
	IID	L/non-IID	F/non-IID
SimSiam + CNN	58.91	57.80 (↓1.11)	57.81 (↓1.10)
MAE + CNN	57.86	57.77 (↓**0.09**)	57.20 (↓**0.66**)
SimSiam + ViT	48.60	43.49 (↓5.11)	43.07 (↓5.53)
MAE + ViT	50.04	48.95 (↓**1.09**)	49.60 (↓**0.44**)

	ImageNet		
	IID	L/non-IID	F/non-IID
SimSiam + CNN	46.74	46.10 (↓0.64)	46.41 (↓0.33)
MAE + CNN	45.88	45.87 (↓**0.01**)	45.80 (↓**0.08**)
SimSiam + ViT	61.97	59.86 (↓2.11)	59.13 (↓2.84)
MAE + ViT	62.69	62.25 (↓**0.44**)	62.51 (↓**0.18**)

obtain three pre-trained ViT backbones: (1) a global model trained using federated learning across all clients; and (2) two local models trained solely on data from client 1 and client 100, respectively. To compare their learned feature spaces, we extract the encoder features of each model. These high-dimensional features are first projected to 20 dimensions using principal component analysis (PCA) and then embedded into 2D space using Umap [99] for visualisation.

Figure 5.2 presents the features of local and global models learned by each D-SSL method. Each column corresponds to one method, while each row shows features from a specific model (client 1, client 100, and the global model). We observe that for distributed MIM methods, the local features are more aligned with each other and also closer to the global features, suggesting more consistent representations

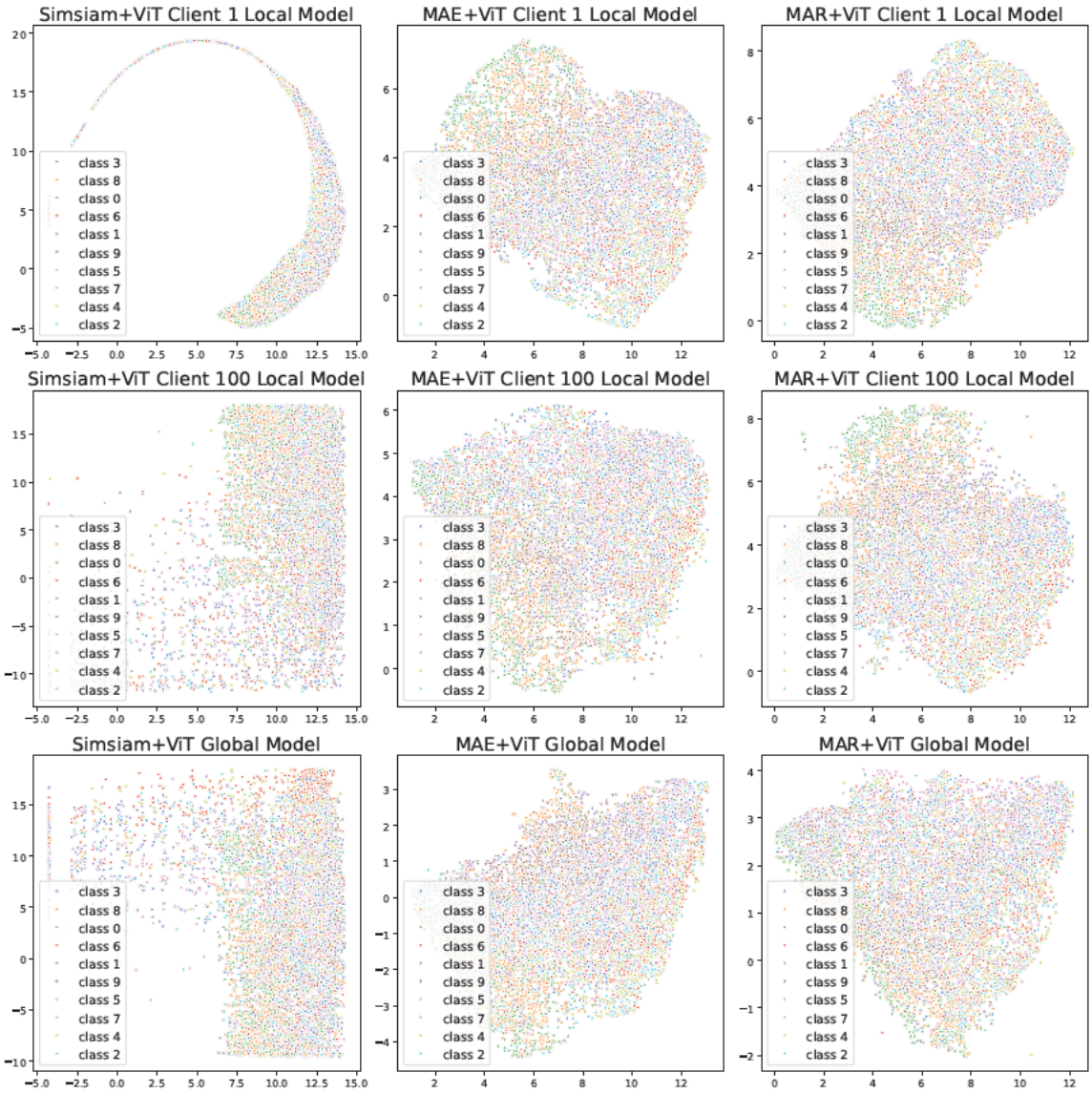


Figure 5.2: **Visualisation of the feature space of local and global models in non-IID setting.** Each column stands for a D-SSL framework (i.e., pre-training ViT by SimSiam, pre-training ViT by MAE, and pre-training ViT by MAR). The first row shows the local feature space from client 1, the second row shows the local feature space from client 100, and the last row shows the global feature space.

across heterogeneous clients. In contrast, distributed CL exhibits greater divergence between local and global features, indicating that it is inherently more sensitive to data heterogeneity.

To provide a more quantitative comparison, we also show the weight differences between local and global models in Table 5.4. In particular, we compute the layer-wise

Table 5.4: **Weight distance between local and global models learned from different D-SSL methods.**

l_2-Norm Difference	SimSiam + ViT	MAE + ViT	MAR + ViT
local 1 vs local 100	45.37	36.10	**35.75**
local 1 vs global	40.34	31.57	**31.38**
local 100 vs global	38.39	31.77	**31.25**

ℓ_2-norm difference between local and global model weights and report the sum across all layers. The results show that distributed MIM methods (MAE and MAR) yield significantly lower weight distances compared to distributed CL, reinforcing the observation that MIM leads to more stable and consistent model updates in the presence of non-IID data.

Furthermore, we observe that the difference between MAE and MAR is relatively moderate. This is consistent with their shared foundation in masked image modelling, where the core reconstruction objective already provides strong robustness to data heterogeneity. The additional alignment in MAR therefore acts as a refinement that further improves consistency across clients, rather than introducing a fundamentally different behaviour.

Impact of Average Connectivity on Non-IID Robustness. We verify our second insight by setting up decentralised networks with different average connectivity $|\bar{A}|$. For the same $|\bar{A}|$, we consider two cases: (1) a general case where the number of neighbours $|A_i|$ varies across clients, and (2) a uniform case where all clients have the same connectivity, i.e., $\forall i \in [N], |A_i| = |\bar{A}|$. Additionally, we set up a FL scenario with 20 clients training in parallel per round. Figure 5.3 shows that Corollary 2 is correct. We can observe that the fine-tuning accuracy of decentralised SSL increases with $|\bar{A}|$. Moreover, Figure 5.3 provides empirical evidence for Theorem 14. We find that pre-training in the federated framework is no less robust than in decentralised frameworks against heterogeneous data.

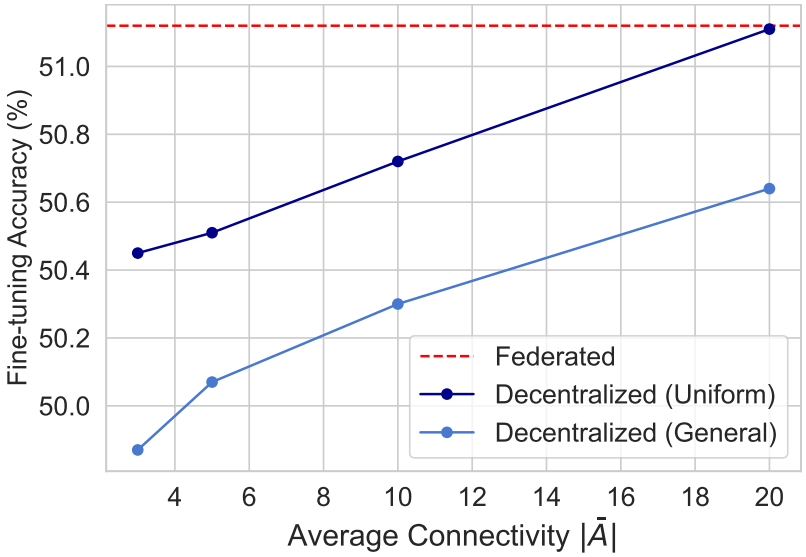


Figure 5.3: **Impact of the average connectivity between clients on the non-IID robustness.** Models are pre-trained in a network with 20 clients and then fine-tuned on CIFAR-100. The blue line shows the results of DecL, and the orange line shows FL results.

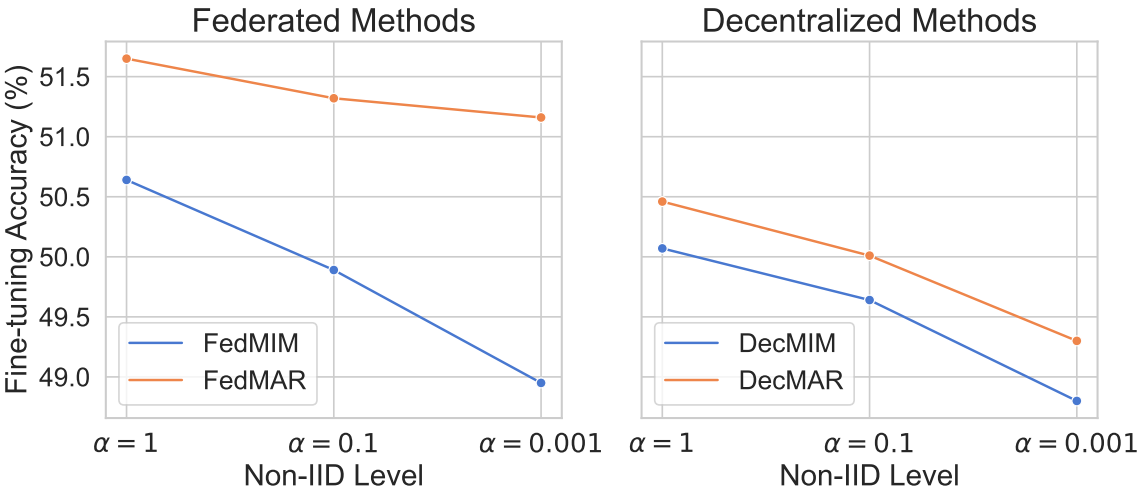


Figure 5.4: **Comparison of MAR and MIM loss on robustness to data heterogeneity in federated and decentralised settings.**

Effect of Different Consensus Matrices on Robustness Theory. This experiment examines whether our robustness findings remain valid under different choices of consensus matrices in decentralized learning. The theoretical bounds link robustness to the average connectivity of the network graph, while the average connectivity is closely related to how efficiently information mixes across clients. If various consensus rules produced qualitatively different mixing behavior, they could

Table 5.5: **CIFAR-100 Accuracy (%) of decentralized MIM under different consensus matrices.** Results are averaged over three test runs.

Method	Avg Connectivity ≈ 5	Avg Connectivity ≈ 10
FL (reference)	**52.72**	**52.72**
DecL with Data Size Weights	46.62	52.28
DecL with Degree Normalized	45.34	52.08
DecL with Doubly Stochastic	45.22	52.18
DecL with Push Sum	45.19	52.22

in principle affect robustness. To test this, we conduct two groups of experiments on decentralized networks with 20 clients under strong non-IID conditions using $\alpha = 0.1$. In the first group, the network has an average connectivity of around 5, while in the second group, connectivity is increased to about 10. Within each group, we pre-train distributed MIM using four commonly adopted consensus schemes, including data size weighting, degree normalized averaging, doubly stochastic matrices, and push sum, and compare all results against a federated learning baseline with the same number of clients. The results in Table 5.5 show a consistent pattern. When connectivity is low, all decentralized variants suffer a noticeable accuracy loss relative to federated learning, and the specific consensus rule makes only minor differences. When connectivity increases, all decentralized variants recover to a level that is close to the federated baseline yet never surpass it. These findings confirm that the qualitative ordering predicted by the theory persists. The choice of consensus matrix influences only constant factors in mixing but does not overturn the robustness relation that federated learning is at least as robust as decentralized learning.

5.6.3 Evaluation of the MAR Loss

Superior Performance over Vanilla MIM. To validate the effectiveness of the proposed MAR loss, we compare it against the standard MIM loss in both FL and DecL frameworks under varying degrees of data heterogeneity. Figure 5.4 illustrates that, as the non-IID level increases (i.e., the Dirichlet parameter α decreases from 1 to 0.001), fine-tuning accuracy declines across all methods. However, models

Table 5.6: **Comparison of FedMAR with SOTA F-SSL methods on Non-IID data ($\alpha = 0.1$) under cross-device ($n = 100$) settings.** Each method was pre-trained with Mini-ImageNet Dataset. The table shows the mean fine-tuning accuracy (%) of three trials.

Method	**Architecture**	**Params**	**GFLOPS**	CIFAR-10	CIFAR-100	ImageNet
FedU [174]	ResNet-18	38.47M	7.40	72.02	38.44	65.10
FedEMA [175]	ResNet-18	38.47M	7.40	70.73	40.78	65.24
Orchestra [93]	ResNet-18	11.84M	7.31	88.87	70.11	65.02
FeatARC [149]	ResNet-18	11.70M	1.83	89.60	64.11	68.17
LDAWA [110]	ResNet-18	15.39M	1.83	89.95	68.96	51.43
FedU^2 [83]	ResNet-18	15.39M	1.83	82.39	55.49	45.27
FedMAR(Ours)	ResNet-18	22.50M	3.64	**92.70**	**70.82**	65.36
FedMAR(Ours)	Tiny-ViT	11.60M	0.88	**90.03**	**71.28**	**75.99**

pre-trained with MAR loss consistently outperform those trained with MIM loss across all non-IID levels. This trend is evident in both FL and DecL frameworks, demonstrating that MAR loss effectively mitigates the sensitivity of distributed MIM to non-IID data.

Compare MAR to State-of-the-art Baselines. Furthermore, to more comprehensively evaluate the performance of our proposed loss, we compare the federated learning application, FedMAR, against several state-of-the-art (SOTA) federated self-supervised learning (F-SSL) baselines in a non-IID distributed setting. The SOTA baselines involve: **1) FedU** [174]: Using the divergence-aware predictor module for dynamic updates within the self-supervised BYOL network [35]; **4) FedEMA** [175]: Employing EMA of the global model to adaptively update online networks; **5) Orchestra** [93]: Combining clustering algorithms with Federated Learning for better model aggregation. **6) FeatARC** [149]: Combining clustering techniques with feature alignment; **7) LDAWA** [110]: Smartly aggregating models according to the angular divergence between local models; and **8) FedU^2** [83]: Optimising training with the flexible uniform regulariser and efficient unified aggregator. Following prior works [110,174], we simulate a highly heterogeneous scenario with 100 clients sampled from a Dirichlet distribution with $\alpha = 0.1$. In each round, 5 clients are randomly selected and each conducts 10 epochs of local training for 200 rounds in total.

Since most baselines employ ResNet-18 [43] as the backbone, we first implement FedMAR with ResNet-18 for a direct comparison. As shown in Table 5.6, FedMAR employed on ResNet-18 achieves higher accuracy on CIFAR-10 and CIFAR-100 while obtaining comparable results on ImageNet. This indicates that MAR loss can provide tangible improvements even when using the same CNN backbone as prior methods.

To further examine the generality of MAR, we also evaluate FedMAR with a lightweight Vision Transformer backbone (Tiny-ViT). Importantly, this model has a comparable number of parameters and GFLOPs to ResNet-18, ensuring fairness in comparison. In this setting, FedMAR employed on Tiny-ViT achieves superior performance on all three benchmarks, surpassing CNN-based baselines while maintaining lower computational cost. These results suggest that MAR loss is not limited to convolutional architectures and can be particularly effective when applied to transformer-based models in federated self-supervised learning.

Ablation Study on Alignment Metric. Our MAR loss (Eq.5.7) involves two key components: the dynamic regularisation weight γ_t and the A-MMD distributional penalty used to align local and global representations. To understand their impact, we perform ablation studies on each component. We first evaluate the contribution of the alignment metric.

For baselines, we consider two commonly used choices in prior work: cosine similarity, which has been widely adopted in federated SSL studies for enforcing alignment between local and global feature spaces [149], and vanilla MMD with a fixed kernel bandwidth, which has also been explored in recent federated learning works [52, 84, 94]. In addition, following standard practice in kernel methods [34], we include vanilla MMD with the bandwidth selected by the median heuristic, which adapts the kernel scale to the data distribution.

On top of these baselines, we evaluate our data-adaptive variant A-MMD, where the kernel bandwidth is estimated from pairwise distances between samples using either the median or mean statistics. As shown in Table 5.7, using the median heuristic

Table 5.7: **Evaluation of different alignment metrics for MAR loss on CIFAR-100.** We report accuracy (%) under three settings of fixed γ: 1e−1, 1e−2, and 0 (degenerate to vanilla MIM).

Metric	$\gamma = 1e{-}1$	$\gamma = 1e{-}2$	$\gamma = 0$
Cosine Similarity	51.71	52.47	51.45
Vanilla MMD ($\sigma = 1$)	51.79	52.12	51.45
Vanilla MMD (median σ)	52.15	53.09	51.45
A-MMD (median σ)	52.42	54.13	51.45
A-MMD (mean σ) [Ours]	**54.09**	**54.39**	51.45

Table 5.8: **Evaluation of regularisation weight γ for MAR loss.**

Weight Schedule	Acc(%)
$\gamma = 1$	51.50
$\gamma = 1e{-}1$	54.09
$\gamma = 1e{-}2$	54.39
$\gamma = 1e{-}3$	53.55
$\gamma : 1e{-}1 \rightarrow 1e{-}3$ (cosine decay)	**54.91**

already provides a clear improvement over fixed-bandwidth MMD, highlighting the importance of data-dependent kernel scaling. Building on this, A-MMD further improves performance across different γ values, consistently outperforming both cosine similarity and vanilla MMD with median bandwidth. Between the two variants, using the mean of pairwise distances achieves slightly better performance, and is adopted as our default design.

Ablation Study on Regularisation Weight. Next, we analyse the impact of the regularisation weight γ by fixing the alignment metric to A-MMD. Results in Table 5.8 show that using a large weight ($\gamma = 1$) degrades performance, as the alignment term overwhelms the reconstruction objective. Conversely, very small weights such as $\gamma = 1e{-}3$ reduce MAR to a near-vanilla MIM objective and fail to deliver sufficient robustness gains. Moderate fixed values such as $\gamma = 1e{-}2$ and $\gamma = 1e{-}1$ yield stronger results, but still remain below our proposed dynamic schedule.

Notably, the cosine decay schedule that smoothly decreases γ from 1e−1 to 1e−3 achieves the best performance (**54.91%**). This validates our intuition behind dynamic weighting: stronger alignment is most beneficial in the early stage when

client divergence is high, while gradual weight relaxation avoids excessive penalty in later stages. These findings highlight the importance of the dynamic design in MAR loss, which not only achieves higher accuracy but also improves training stability.

5.7 Further Discussions on Concerns of MAR

When deploying MAR loss in practice, natural concerns arise regarding the potential privacy risks and the additional communication associated with sharing local representations. We provide both quantitative and qualitative analyses below to show that these costs remain modest and manageable.

5.7.1 Privacy Considerations

The information communicated by MAR is limited to local representations $z_i = f_e(x_1)$ derived from the unmasked portion of the input. Because MIM typically adopts a high masking ratio (e.g., 75% in MAE [41]), most raw content remains hidden and the embedding dimensionality is substantially reduced, which mitigates potential leakage. For stronger guarantees, MAR can be further combined with standard Differential Privacy (DP) mechanisms [101, 153] by perturbing embeddings before transmission, e.g., $z_i \leftarrow f_e(x_1) + \mathcal{N}(0, \sigma^2 I)$ with σ calibrated to satisfy (ϵ, δ)-DP.

5.7.2 Communication Overhead

In addition to the standard model updates (e.g., gradients or weights), MAR transmits compact masked embeddings computed from the unmasked portion of each input. This is the sole extra payload introduced by MAR. For instance, in the MAE (ViT-B/16) setting on ImageNet with a 75% masking ratio, each image has 196 patches, of which 49 remain visible. With hidden size 768 and batch size 256, this yields about $49 \times 768 \times 256$ float values (≈ 36.8 MB in float32). By contrast, a full model with 86M parameters is ≈ 328 MB, so the additional cost from MAR is only $\sim 11\%$ under this configuration. Crucially, in cross-device settings where small batches are common, this extra cost decreases proportionally with the batch size:

at B=128 it is $\approx$ 18 MB ($\sim$ 5%), at B=64 it is $\approx$ 9 MB ($\sim$ 3%), and at B=32 it drops to around $\sim$1%. These calculations indicate that the MAR-induced overhead remains acceptable in realistic deployments. Moreover, MAR is optional: when minimal communication is the overriding priority, one can simply use the standard MIM objective, whose effectiveness is explained by our theory, at zero additional cost. When a small extra cost is acceptable, MAR offers corresponding robustness gains while keeping the overhead low.

5.8 Chapter Conclusion

This chapter examines the challenge of distributed self-supervised learning (D-SSL) when dealing with highly heterogeneous data, focusing on exploring the robustness differences across different D-SSL algorithms. Through rigorous theoretical analysis, we demonstrate that among the two dominant SSL paradigms, Masked Image Modelling (MIM) exhibits greater robustness to data heterogeneity compared to Contrastive Learning (CL). Moreover, we derive that the average connectivity of a network positively correlates with the insensitivity of D-SSL, which also implies the superior robustness of the federated learning framework over the decentralised learning framework. Building on these insights, we propose MAR loss, a novel approach to further enhance the robustness of distributed MIM by aligning local and global representations through regularisation. Extensive experiments validate our theoretical findings and confirm the effectiveness of MAR loss.

5.9 Chapter Notations and Definitions

i	Client index
N	Number of clients
$\mathcal{G}, \mathcal{V}, \mathcal{E}$	Client graph, Node set, and Edge set
A	Adjacency matrix of a graph
W	Model parameters/weights matrix
$f, \mathcal{L}$	Training loss
D	Dataset
h	Class index
x, x_1, x_2	Data points, Masked/unmasked part
d	Dimension of the feature space of data
j, k	Dimension indices
e	Standard unit-norm vector
τ, μ	Hyperparameters for data generation
q	Sampled constant from $\{0, 1\}$
ξ	Sampled Gaussian noise
$g(\cdot)$	Data augmentation operation
c	Weight dimension constant
H	Linear image transformation
m	Binary mask
p	Masking ratio
z	Feature latent/embedding
f_e, f_d	Encoding and decoding function
$\mathcal{R}$	Row span
r	Representability vector
s	Sensitivity to data heterogeneity
B	Batch size
$\mathcal{C}$	Sampled client set
$k(\cdot)$	Kernel function
a, b	Feature dimension indices

CHAPTER 6

DeNAV: Decentralised Self-Supervised Learning with a Training Navigator

Chapter Overview: In the preceding chapters, we established a list of connected theoretical foundations. Chapter 3 shows that edge devices are best suited to smaller models, as compute-optimal size decreases with stronger data decentralisation. Chapter 4 demonstrates that distributed training inevitably suffers a generalisation gap relative to centralised training, which can only be narrowed by involving more clients or enlarging local datasets. Chapter 5 finds that MIM-based self-supervised learning is particularly promising in distributed settings. These findings point toward an algorithmic direction: enabling decentralised training that scales across many clients, transmitting smaller models practical for edge devices, and exploiting MIM-based self-supervision to mitigate data heterogeneity.

Motivated by these insights, this chapter introduces DeNAV, a decentralised self-supervised learning framework for large-scale scenarios where clients hold only unlabelled data and communicate solely with neighbours. DeNAV builds on Masked Autoencoders (MAE) for local training, pre-trains multiple lightweight transformers across clients, employs a navigator algorithm to plan training routes, and adopts staleness-aware aggregation to handle asynchronous updates. Beyond the design, we provide theoretical analysis proving DeNAV's convergence and consensus guarantees, while also quantifying the effect of local data volume. Experiments further show that DeNAV matches state-of-the-art federated SSL and surpasses prior decentralised methods under equal communication budgets.

6.1 Introduction

The success of deep learning relies on the availability of vast training data, but in practice, most data is generated and stored in a decentralised manner across devices, institutions, and users. Centralising this data is often infeasible due to privacy, ownership, or communication constraints, and labelled data is particularly scarce. These realities highlight the demand for decentralised self-supervised learning, which enables clients to collaboratively train models without requiring labels or server coordination. Such an approach is especially relevant in environments like ad-hoc device networks, IoT systems or cross-institution collaborations without a trusted coordinator (e.g., financial institutions in blockchain service), where data is abundant but scattered and where reliable server access cannot be assumed. These scenarios also typically involve heterogeneous data distributions, dynamic connectivity, and limited communication resources, which make the direct adaptation of centralised self-supervised methods to distributed settings highly challenging.

The theoretical studies in the preceding chapters reinforce this motivation and also provide guidance on how a practical framework should be designed. Chapter 3 shows that the compute-optimal model size decreases as training becomes more decentralised, indicating that small backbones are best suited for edge devices. Chapter 4 establishes that federated training inevitably underperforms centralised learning unless additional clients or data are introduced, suggesting that scalability through broad client participation is essential. Chapter 5 demonstrates that self-supervised methods, particularly masked image modelling, are more robust to heterogeneous data than contrastive learning and that decentralised frameworks benefit significantly from higher connectivity. Together, these results not only confirm the necessity of decentralised self-supervised learning but also point to its design principles: it should prioritise small models for efficiency, exploit self-supervision for robustness, and support scaling across many clients without server dependence.

Motivated by these insights, this chapter introduces DeNAV, a decentralised self-supervised learning framework designed for large-scale scenarios where clients hold only unlabelled data and communicate solely with their neighbours. DeNAV incorporates several key innovations. First, it employs parallel routing and training of multiple lightweight masked autoencoder (MAE) models, enabling efficient utilisation of distributed resources while reducing communication costs. Second, it uses a navigator algorithm to dynamically determine training routes by evaluating each client's data volume, computational capacity, and past participation, ensuring balanced and effective exploration of the network. Third, staleness-aware aggregation is introduced to address the discrepancies that arise from asynchronous updates, thereby improving the stability of model integration across clients. Finally, DeNAV adopts an efficient masked pre-training strategy, starting with compact MAE models and later expanding them into larger backbones via weight sharing, which combines the advantages of lightweight communication with the ability to scale to stronger downstream models.

Beyond the design, this chapter provides a rigorous theoretical analysis of DeNAV. Specifically, we prove that DeNAV holds convergence and consensus guarantees, and justify the model routing policy employed in the navigator algorithm by showing that selecting clients with richer local datasets yields better model updates. These results ground DeNAV in solid theory. Furthermore, we empirically validate the effectiveness of DeNAV through comprehensive experiments. We compare its performance with state-of-the-art federated SSL baselines and existing decentralised methods, showing that DeNAV achieves competitive or even superior accuracy under the same communication budget. Additionally, we conduct ablation studies to elucidate the contribution of each design component, and hyperparameter sensitivity analysis confirms the robustness of the framework across various settings. These experiments provide strong empirical evidence for the practicality of DeNAV in challenging distributed environments.

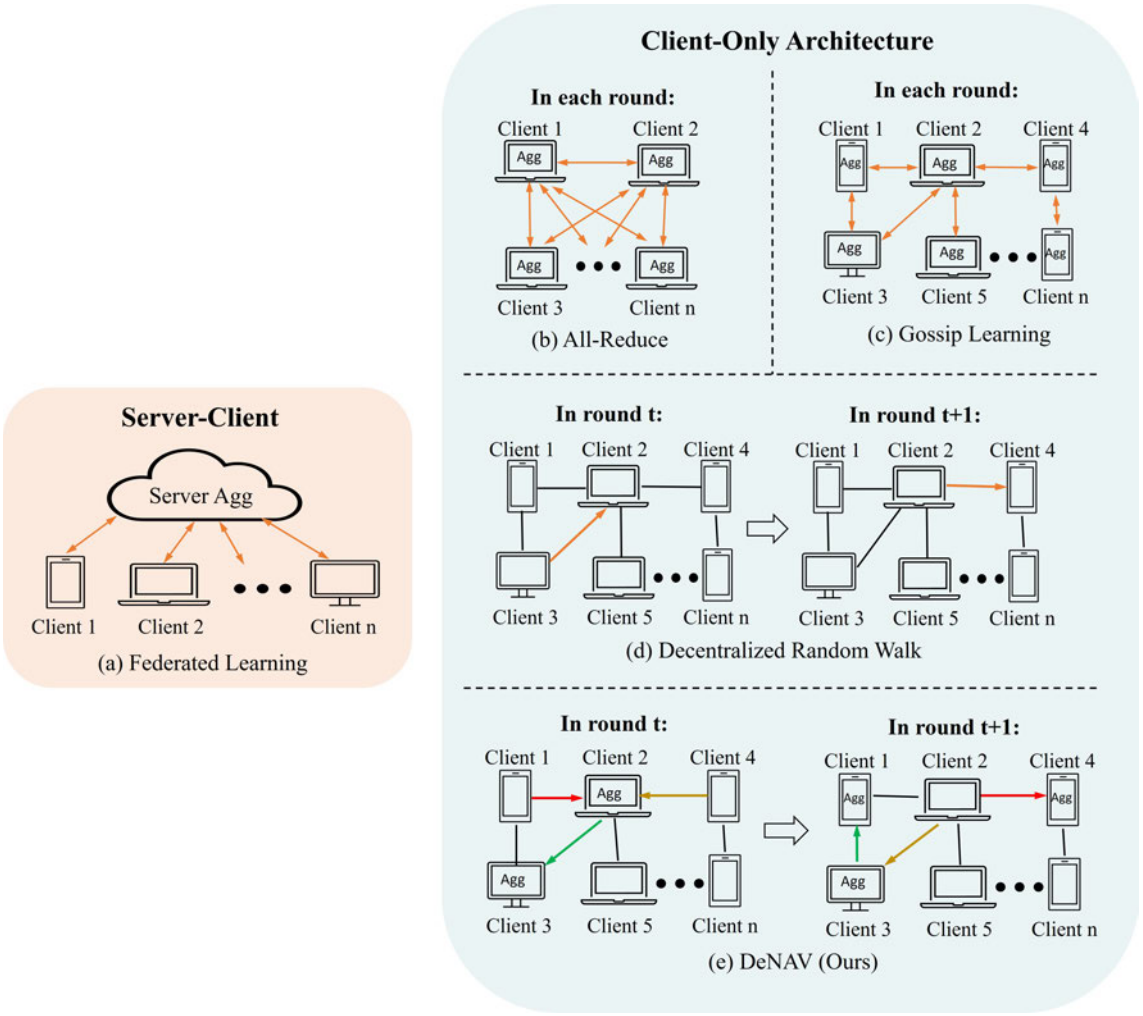


Figure 6.1: **Overview of Different Distributed Training Frameworks for Server-Client and Client-Only Architectures. (a) Federated Learning (FL)**: Multiple clients collaboratively train a global model under the coordination of a server. Beyond this classical form, FL also includes hybrid or hierarchical variants, where server coordination is augmented with peer-to-peer exchanges. **(b) All-Reduce**: Each client trains a model and communicates with all other clients to aggregate updates. **(c) Gossip Learning**: Each client trains a model and communicates with all neighbours. **(d) Decentralised Random Walk**: Train a global model by random walking among clients. **(e) DeNAV (Ours)**: Multiple models are smartly transmitted, aggregated, and trained among clients in parallel in the network.

6.2 Related Work

Transformer and Self-Supervised Learning. Transformer models [141], which are typically trained through SSL on large-scale training datasets, have received significant attention due to their powerful performance. When vision transformers

(ViTs) were proposed [22], corresponding SSL methods also emerged, such as Masked Autoencoder (MAE) [41], which randomly masks 75% of image patches and trains an autoencoder to reconstruct the masked portion. The significant advantage of MAE is the reduction of memory consumption during training. Similarly, ALBERT [72] drastically reduces the training overhead by sharing all parameters across transformer model layers. However, previous studies have focused on centralised scenarios. In reality, vast amounts of training data are generated and stored in a distributed manner. To significantly expand the training data size while avoiding the privacy issue resulting from data centralisation, we have proposed a new method, DeNAV, to implement decentralised self-supervised training of transformer models.

Federated Self-Supervised Learning. FL is a collaborative framework where models are trained on multiple clients and aggregated on the central server in each communication round [100]. Beyond the classical single-server form, hybrid or hierarchical FL introduces intermediate aggregation layers, such as edge servers or cluster heads, combining server coordination with peer-to-peer exchanges [88, 152]. These designs help reduce communication bottlenecks and improve robustness against client or server dropouts. Federated self-supervised learning (FSSL) combines SSL with these federated settings [174] to implement model pre-training with distributed unlabelled data. However, the state-of-the-art FSSL approaches rely heavily on centralised or hierarchical coordination to stabilise training [93, 175], limiting their applicability to server-free scenarios. Moreover, most FSSL methods are tailored for pre-training Convolutional Neural Networks (CNNs) [14, 15]. FedMAE recently explored federated pre-training of transformers [158], but without effective strategies for transformer aggregation. In contrast, DeNAV can be applied to fully decentralised scenarios, offering effective training and aggregation of transformer blocks across heterogeneous clients. It can also be adapted to hybrid or hierarchical FL, providing smart peer-to-peer exchanges to further enhance training efficiency.

Decentralised Learning. DecL allows model training in client-only distributed scenarios, but classical decentralised frameworks face expensive communication costs. All-Reduce requires each client to transmit their model updates to all other clients [16], while Gossip Learning requires them to transmit their model updates to all their neighbours [44, 82]. Recent studies propose new frameworks to improve communication efficiency, including Decentralised Random Walk [121], which sequentially trains a global model through random walking among clients, and Efficient Gossip Learning in which each client only communicates with a high-bandwidth neighbour in each communication round [131]. However, reducing communication traffic results in new problems. Since there are fewer models being aggregated and trained each round, these new algorithms are inferior to the previous ones in convergence and the generalisation capability of models. Our framework, DeNAV, overcomes this shortcoming by providing the flexibility to shift towards higher communication efficiency or better training results. As shown in Figure 6.1, unlike previous approaches, the number of models to be trained simultaneously can be freely specified. Besides, to further improve performance, we employ calibrated algorithms to smartly schedule the training routes for each model and aggregate model weights stored on clients.

6.3 Methodology

In this section, we define a realistic and challenging decentralised scenario and then introduce our proposed training framework, Decentralised Navigator (DeNAV), which realises Decentralised Self-Supervised Learning in this scenario.

6.3.1 Scenario Definition

Multiple parties aim to collaboratively learn a generic representation for various downstream tasks without sharing data. We represent these parties as n clients containing unlabelled data and assume that the training scenario is a connected network $\mathcal{G}(\mathcal{C}, \mathcal{E})$, where $\mathcal{C}$ denotes the set of clients, and $\mathcal{E}$ denotes the set of established

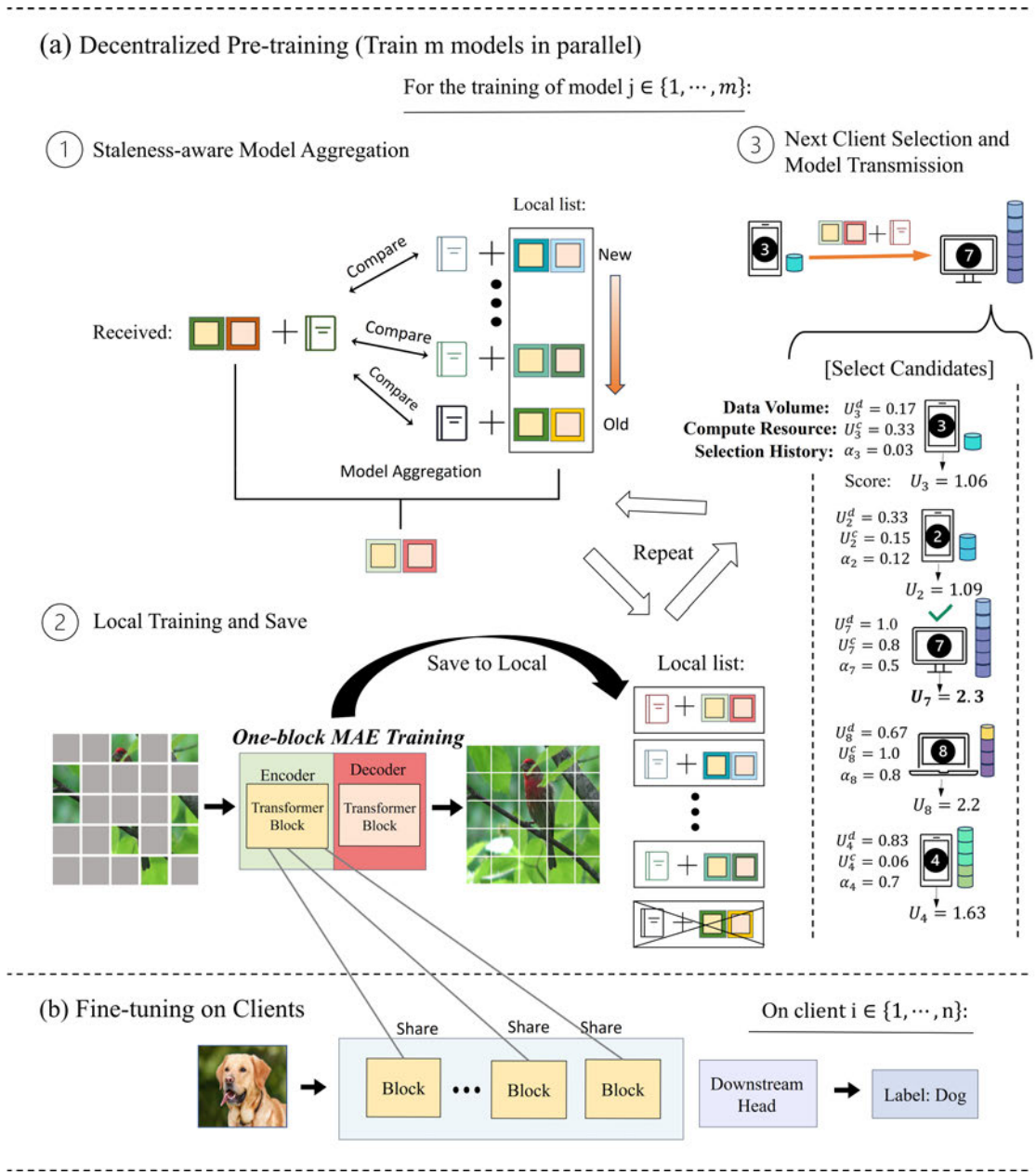


Figure 6.2: **Illustration of the training framework of DeNAV.** The training scenario is a client-only network where clients vary in terms of data classes, data volume, and computational resources. Our framework dynamically routes multiple models across clients through a training navigator, enabling adaptive coordination under heterogeneous data, computation, and communication conditions.

connections between clients. For each client $i \in \{1, \ldots, n\}$, there is a local dataset $\mathcal{D}_i = \{X_i\}$ and local computational resources q_i. The global training objective is to find the parameter $\theta := \arg\min_\theta \sum_{i=1}^{n} f_i(\theta)$, where $f_i(\theta) := \mathbb{E}_{\xi \sim \mathcal{D}_i} F_i(\theta; \xi)$ is the

expected loss over local data distribution $\mathcal{D}_i$, ξ is the unlabelled data sampled from $\mathcal{D}_i$ and $F_i(\theta;\xi)$ is the loss function. A key challenge about this scenario is that each client i can only exchange information with their peers j that satisfy $e_{i,j} \in \mathcal{E}$. Another challenge is that the clients are different from each other. Apart from the difference in data volume and computational resources, the data between these clients is most likely non-IID (not Independent and Identically Distributed), which means that each client does not hold the same classes of local data.

6.3.2 DeNAV Overview

The overview of our proposed training framework, DeNAV, is shown in Figure 6.2. DeNAV consists of two phases: (a) Decentralised Pre-training, and (b) Fine-tuning on Clients. At a high level, DeNAV enables multiple lightweight models to be dynamically routed across clients, thereby balancing communication efficiency and representation learning effectiveness. Such a design is particularly suitable for large-scale decentralised environments with heterogeneous data distributions and communication constraints, such as ad-hoc device networks, peer-to-peer systems, or cross-organisation collaborations without a trusted central coordinator.

Decentralised Pre-training Before starting the pre-training, practitioners can specify the number of simultaneously trained models and the total number of pre-training steps for each model. Here, we assume that there are $m \in [1, n]$ models and the pre-training will be iterated for T steps. Afterwards, m models along with m log files are initialised on m idle clients in the network. The pre-training model adopts a lightweight architecture called one-block masked autoencoder, which follows the architecture of masked autoencoder [41] but contains only one transformer block in both the encoder and decoder. The generated log file is used to record the latest training state of the model, containing the model ID, the transmission history of the model across clients, and the training history of the model. For each pre-training step $t \in \{1, \ldots, T\}$, the training pipeline of DeNAV consists of the following stages:

① ***Staleness-aware Model Aggregation***: Each training client $i \in \mathbb{C}_t$ receives the pre-training model θ_{t-1} and the state log ψ_{t-1} of this model from a neighbour or model initialisation. If there are some local models previously stored in the client, the client then compares the state log ψ_{t-1} with the state logs of local models Ψ_i. The models that satisfy the aggregation criteria will be aggregated with the received model based on the information stored in the state logs of both parties, generating the model for local training. ② ***Local Training and Save***: The client i then conducts a local training with local unlabelled data $\mathcal{D}_i$ of size $|\mathcal{D}_i|$ for K iterations to update the model from θ_{t-1} to θ_t. To reduce computation cost, we follow the training method of MAE to mask 75% of image patches out of the local data and train the model with the image reconstruction task. After the local training, the state log ψ_{t-1} is also updated to ψ_t. A copy of θ_t and ψ_t is saved to the local lists Θ_i and Ψ_i. Note that the length of these local lists depends on each client's storage space. If the space is full, then the new copy will replace the earliest saved model weights and state logs. ③ ***Next Client Selection and Model Transmission***: At the end of pre-training step, based on our training navigator algorithm, the client i computes the selection score for each client in the set of the candidate clients $\mathcal{C}_i$ (including the client i and the neighbours of client i), identifies the client j with the highest score, and sends the model weights θ_t and state log ψ_t to this client.

Fine-tuning on Clients. After T pre-training steps, each client has several local copies of the one-block masked autoencoder. By considering the latest local copy as the received one, these local models can be aggregated into a single model using our staleness-aware model aggregation algorithm. Then, all previously stored local models, as well as the decoder of the new model, are discarded. Only the single encoder transformer block of the new model is left for fine-tuning in the future. During fine-tuning, the client first initialises a large transformer backbone containing multiple blocks with random weights. Afterwards, similar to the parameter sharing trick used in ALBERT [72], the parameters of the pre-trained encoder block are

Algorithm 3 Decentralised Pre-training in DeNAV

Input: Network $\mathcal{G}$, datasets $\mathcal{D}$, number of models m, staleness bound λ, steps T, max selections per client Z, local epochs K
Output: Trained model weights on clients

Initialise models and state logs on sampled clients $\mathbb{C}_0$
for $t \leftarrow 0$ to $T-1$ **do**
    $\mathbb{C}_{t+1} \leftarrow \emptyset$
    for all client $i \in \mathbb{C}_t$ **in parallel do**
        Client i receives model θ_{t-1} and state log ψ_{t-1}
        if $m > 1$ and $\Theta_i \neq \emptyset$ and $\Psi_i \neq \emptyset$ **then**
            $\theta_{t-1}^{(0)} \leftarrow \textsc{StaleAwareModelAgg}(\theta_{t-1}, \psi_{t-1}, \Theta_i, \Psi_i, \lambda)$
        else
            $\theta_{t-1}^{(0)} \leftarrow \theta_{t-1}$
        for $k \leftarrow 0$ to $K-1$ **do**
            $\theta_{t-1}^{(k+1)} \leftarrow \text{MAE}(\theta_{t-1}^{(k)}, \mathcal{D}_i)$
        $\theta_t \leftarrow \theta_{t-1}^{(K)}$
        $\psi_t \leftarrow \textsc{UpdateStateLog}(\psi_{t-1}, i, t, |\mathcal{D}_i|)$
        $\Theta_i \leftarrow \Theta_i \cup \{\theta_t\}$, $\Psi_i \leftarrow \Psi_i \cup \{\psi_t\}$ ▷ save a local copy
        $j \leftarrow \textsc{TrainingNavigator}(\mathcal{G}, i, \psi_t, T, Z)$
        Client i sends θ_t and ψ_t to client j
        $\mathbb{C}_{t+1} \leftarrow \mathbb{C}_{t+1} \cup \{j\}$

shared with each block in the backbone. Lastly, the updated backbone is connected with a task head and fine-tuned with labelled data to build applications for various downstream tasks.

We summarise the pre-training of DeNAV in Algorithm 3. Next, we introduce the technical details of the key modules in DeNAV, which are staleness-aware aggregation and training navigator algorithms.

6.3.3 Staleness-aware Model Aggregation

At each pre-training step, each training client aggregates the received model with the local models to combine the strengths of each model. Traditional FL [100] generally employs synchronous model aggregation on the central server. However, in decentralised scenarios, there may be a long interval for the same clients to be re-selected for training. Therefore, the local models saved on the client are likely to be much more stale in the training status compared to the received model. Simply

aggregating the local and received models using average weights will not output a better model. To address this issue, we take the staleness of each model into account during model aggregation.

Our staleness-aware model aggregation starts by initialising the list of aggregated models Θ^{Agg}, the data volume weight list w^{Volume}, the staleness weight list w^{Stale}, and the aggregation weight list w. For each local model on client i, its state log ψ_i is compared with the state log of the received model ψ_t. If the local model $\theta^{(i)}$ is not a previous local copy of the model θ_t by checking ID and the step interval between the two models falls within the staleness bound λ, the local model $\theta^{(i)}$ is included in Θ^{Agg}, and two weight lists w^{Stale} and w^{Volume} are updated using the information recorded in the state log ψ_i. Subsequently, the softmax function normalises the weights in w^{Stale} and w^{Volume} into a probability distribution (summing to 1), which is described as:

$$w_i^{Stale} := \frac{\exp(\psi_i^{Steps})}{\sum\limits_{\theta^{(i)} \in \Theta_{Agg}} \exp(\psi_i^{Steps})}, \quad w_i^{Volume} := \frac{\exp(\psi_i^{Volume})}{\sum\limits_{\theta^{(i)} \in \Theta_{Agg}} \exp(\psi_i^{Volume})}. \tag{6.1}$$

The actual weights of the model aggregation w can be further calculated from w^{Stale} and w^{Volume} by applying the softmax function, with the following form:

$$w_i := \frac{\exp(w_i^{Stale} \times w_i^{Volume})}{\sum\limits_{\theta^{(i)} \in \Theta_{Agg}} \exp(w_i^{Stale} \times w_i^{Volume})}. \tag{6.2}$$

Finally, model aggregation is performed on the models within Θ^{Agg} using the weights w to output the model θ for the subsequent local training.

6.3.4 Training Navigator

Due to variations among clients, selecting which client for the next step of pre-training significantly affects the training effectiveness. We introduce the training navigator algorithm to optimise client selection.

Client Selection Score Formulation The objective of the training navigator is to find a sweet spot in the trade-off between training effectiveness and training efficiency by associating each client with its selection score. To accomplish this goal, we identify four critical challenges that must be addressed: **1.** How to determine which client's data would help improve the training effectiveness the most without compromising privacy? **2.** How to take into account the optimisation of training efficiency while optimising training effectiveness? **3.** How to balance between exploring new clients and continuing to exploit the clients that have been selected for maximum gain? **4.** How to ensure that all clients in the network have the opportunity to participate in the model pre-training? We handle these challenges by evaluating each candidate client in terms of training data, computational resources, and selection history.

(Data Volume Utility U^d) To address the first challenge, in DeNAV, we quantify the importance of the local data for each step of pre-training by the data volume $|\mathcal{D}_i|$. For many FL algorithms [100], the amount of data is often a good indicator of the effectiveness of local training and is widely used for aggregation weights. We adopt this insight and further theoretically find that data volume plays a dominant role in the training effectiveness of the one-block masked autoencoder (the theoretical justification on the data volume utility is provided in Section 6.4.2). Initially, this utility is formulated as $U_i^d = \frac{|\mathcal{D}_i|}{\sum_{i \in \mathcal{C}_j} |\mathcal{D}_i|}$, where $\mathcal{C}_j$ represents the set of communication candidates for the current training client j. However, a limitation of this formulation is that the sum of data volumes can be excessively large compared to individual volumes, resulting in an insignificant difference in the utilities between clients with less data. Thus, we define the utility U_i^d as $U_i^d = \frac{|\mathcal{D}_i|}{\max\{|\mathcal{D}_i||i \in \mathcal{C}_j\}}$, where the denominator is the maximum data volume of all clients.

(Computational Resource Utility U^c) In practice, clients with more computational resources can train models with a larger batch size and finish the training in less time than clients with fewer resources. Thus, for tackling the second challenge, we define computational resource utility as a measure of training efficiency:

$U_i^c = (\frac{\min\{\bar{t}_i | i \in \mathcal{C}_j\}}{\bar{t}_i})^{\mathbb{1}(L(i)>0)}$, where $\bar{t}_i$ is the time spent by the client i on its most recent local training, $L(i)$ denotes the last step it was selected, and $\mathbb{1}(x)$ is an indicator function that returns 1 if x is true and 0 otherwise. For never-selected clients, we allocate the maximum utility (i.e., $U_i^c = 1$) to encourage their participation and gather their training time for future evaluation.

(Selection History Factor α) Besides exploiting the already identified clients with excellent training effectiveness and efficiency, our algorithm should also explore new clients, as repeated selection of the same set of clients can cause the model to overfit their local data, leading to diminishing training rewards. Moreover, in practice, clients' data volume and computational resources may change at any time, making previously identified superior clients no longer advantageous. To handle the third challenge and harmonise the trade-off between exploration and exploitation, the selection history factor is introduced, expressed as $\alpha_i = (\frac{t-L(i)}{T})^{\mathbb{1}(L(i)>0)}$. The inclusion of α_i gradually increases the scores of clients that have not been selected for a long time, allowing well-qualified clients with sufficient scores to be re-selected.

(Total Selection Score U) The above three utilities are combined to derive the total client selection score. First, to yield good training performance while optimising the training efficiency, the data volume utility is associated with the computational resource utility as $U_i^d \times U_i^c$. Furthermore, since we prioritise the training performance over training efficiency, the association of α_i and two utilities is formulated as $U_i^d \times (\alpha_i + U_i^c)$ instead of $(\alpha_i + U_i^d) \times U_i^c$. Lastly, to tackle the fourth challenge and ensure that the model can be trained with all clients' data, the client selection score U_i is defined as:

$$U_i = (U_i^d \times (\alpha_i + U_i^c) + 1)^{\mathbb{1}(Z(i)<Z)}, Z \geq \frac{T}{n}, \quad \text{where}$$
$$U_i^d = \frac{|\mathcal{D}_i|}{\max\{|\mathcal{D}_i| \mid i \in \mathcal{C}_j\}}, U_i^c = (\frac{\min\{\bar{t}_i \mid i \in \mathcal{C}_j\}}{\bar{t}_i})^{\mathbb{1}(L(i)>0)}, \alpha_i = (\frac{t-L(i)}{T})^{\mathbb{1}(L(i)>0)}, \tag{6.3}$$

Algorithm 4 Training Navigator

Input: Network $\mathcal{G}$, client i, state log ψ_t, total steps T, max selections Z
Output: Selected client j for the next training step

$\mathcal{C}_i \leftarrow \textsc{FindNeighbours}(\mathcal{G}, i) \cup \{i\}$
$U_i \leftarrow \textsc{EvaluateValue}(\mathcal{C}_i, \psi_t, T, Z)$ ▷ see Eq. (6.3)
$\hat{\mathbb{C}}_i \leftarrow \textsc{DetermineClients}(\mathcal{C}_i, \max\{U_i\})$
upon $|\hat{\mathbb{C}}_i| > 1$ **do**
    $\hat{\mathcal{C}}_i \leftarrow \textsc{Findneighbours}(\mathcal{G}, \hat{\mathbb{C}}_i)$
    $\hat{U}_i \leftarrow \textsc{EvaluateValue}(\hat{\mathcal{C}}_i, \psi_t, T, Z)$
    $\tilde{\mathbb{C}}_i \leftarrow \hat{\mathbb{C}}_i$
    $\hat{\mathbb{C}}_i \leftarrow \textsc{DetermineClients}(\hat{\mathcal{C}}_i, \hat{U}_i)$
    if $|\hat{\mathbb{C}}_i| > |\tilde{\mathbb{C}}_i|$ **then**
        break
$j \leftarrow \hat{\mathbb{C}}_i[0]$

$Z(i)$ is the count of the previous selections of client i, and Z is the maximum allowed selections per client. If a client reaches the maximum selection count, its score will be adjusted to a minimum of 1, allowing other clients the chance for selection. Besides, the upper limit of client selection should depend on the ratio of the total training steps T to the number of clients n in the network. If n remains constant but T increases, each client can be selected more times while ensuring complete access to all local data.

Next Client Selection With the selection scores of candidates, our training navigator algorithm can identify the next training client. The client with the highest score is undoubtedly the best choice, but multiple clients may share the highest score. For instance, when the selection score of each candidate is 1 (i.e., the lowest bound) due to frequent selection, randomly selecting any client may not be optimal. To address this, we use different selection strategies depending on whether such cases occur. If there is no tie in the scores, the client candidate with the highest score will be selected. Otherwise, there will be an iterative comparison between the neighbours of the candidates sharing the highest score. The candidate connecting to neighbours with higher scores will be selected. The details of our training navigator algorithm are summarised in Algorithm 4.

Privacy and Robustness Considerations Privacy is a key concern in distributed training, as sharing model updates or auxiliary information may expose clients to inference attacks. While the training navigator is not explicitly designed to provide formal privacy guarantees, it inherently respects privacy constraints through its client selection design. Specifically, our method does not require clients to reveal sensitive information such as data representations or feature statistics. Instead, the selection relies only on coarse-grained signals, including local data volume, training time, and selection frequency, which are less sensitive and harder to exploit in privacy attacks. Moreover, the framework remains compatible with privacy-preserving techniques such as differential privacy (DP) [100, 153] during local training and communication, which can be further incorporated to strengthen both privacy protection and system robustness in future extensions.

Beyond privacy, these design choices also contribute to the robustness of the training process against potential manipulation. Although the selection signals are client-dependent, DeNAV mitigates such risks through its built-in mechanisms. First, the maximum selection constraint ensures that no client can dominate the training process, guaranteeing that all clients are eventually selected. Second, the selection history factor dynamically reduces the priority of frequently selected clients, encouraging exploration and preventing repeated selection of the same subset of clients. Third, the use of multiple utility factors reduces the reliance on any single signal, making it more difficult for a client to significantly influence its selection probability through isolated manipulation.

Overall, these properties of training navigator design enable DeNAV to maintain stable and balanced training behaviour in decentralised environments, even when client-side information may be imperfect or partially unreliable.

6.4 Theoretical Analysis

In this section, we develop the theoretical foundations of our proposed framework, DeNAV. The analysis is structured into three components. First, we show that

DeNAV provides both convergence and consensus guarantees, with a convergence rate comparable to existing gossip-based decentralised methods. Second, we extend the analysis to the generalisation perspective by employing uniform stability. This result highlights that as the number of simultaneously trained models increases, the algorithm enjoys improved generalisation performance. Third, we justify that guiding the training route toward clients with larger local datasets enhances the training effectiveness of the one-block masked autoencoder. The complete proofs are provided in Section 8.4.

6.4.1 Convergence and Consensus Guarantees

According to Algorithm 3, the model updating rule of DeNAV at the pre-training step t can be expressed as below:

$$\Theta_t = \Theta_{t-1}\mathbb{W}_{t-1} - \eta_{t-1}G(\Theta_{t-1};\xi_{t-1}) = \Theta_0\prod_{s=0}^{t-1}\mathbb{W}_s - \sum_{s=0}^{t-1}\eta_s G(\Theta_s;\xi_s)\prod_{r=s+1}^{t-1}\mathbb{W}_r \quad (6.4)$$

where η is the learning rate, $\theta^{(i)}$ is the latest model weight stored on client i, $\Theta = [\theta^{(1)}, \theta^{(2)}, ..., \theta^{(n)}] \in \mathbb{R}^{N\times n}$ is the local models on all clients, $\mathbb{W}_t \sim \mathbb{W} \in \mathbb{R}^{n\times n}$ is the communication topology, $\nabla F_i(\theta^{(i)};\xi^{(i)})$ is the loss gradient on the client i, and $G(\Theta;\xi) = [\nabla F_1(\theta^{(1)};\xi^{(1)}), ..., \nabla F_n(\theta^{(n)};\xi^{(n)})]$ is the loss gradients on all clients.

To analyze the convergence and consensus of DeNAV, we first impose several standard assumptions [29, 82, 131]:

1. (**Smoothness**) Each $f_i(\cdot)$ has L-Lipschitz continuous gradients.

2. (**Bounded variance**) The stochastic gradient variance is bounded, i,e.,

$$\mathbb{E}_{\xi\sim\Theta_i}||\nabla F_i(\theta;\xi) - \nabla f_i(\theta)|| \le \sigma^2; \quad \frac{1}{n}\sum_{i=1}^{n}||\nabla f_i(\theta) - \nabla f(\theta)|| \le \zeta^2; \quad \forall i, \forall\theta. \quad (6.5)$$

3. (**Markov communication chain**) For all t, $\mathbb{W}_t$ is doubly stochastic. Furthermore, considering that the training navigator employed by DeNAV

routes models based on the previous state recorded in the training state log ψ_t (e.g., based on past visits and client scores), which is similar to the idea of state augmentation in controlled Markov chains and Markov decision processes [71, 106], the sequence $\{\mathbb{W}_t\}$ forms a Markov chain on a finite state space $\mathcal{V} := \{S_t\}_{t=0}^{T}$ with a transition matrix $P \in \mathbb{R}^{\mathcal{V}\times\mathcal{V}}$. The chain is assumed to be irreducible and aperiodic, so that it admits a stationary distribution π. Moreover, under π, the second largest eigenvalue of $\mathbb{E}_\pi[\mathbb{W}^\top\mathbb{W}]$ is bounded by $\rho < 1$.

Then, we can prove DeNAV has convergence and consensus guarantees as follows.

Theorem 15. *Under the above assumptions, if η_t is fixed as η and $\eta = \frac{1}{4L\sqrt{D_2}+\frac{\sqrt{T}}{\sqrt{n}}}$, then*

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{|m|}||\nabla f(\bar{\Theta}_t)||^2 \lesssim \frac{n}{\sqrt{D_2}mT+\frac{mT^{\frac{3}{2}}}{\sqrt{n}}} + \frac{D_1}{mT}||\Theta_0-\bar{\Theta}_0\mathbf{1}_n^\intercal||_F^2 \\ + \frac{D_2n\sigma^2+D_2n\zeta^2}{D_2m+\frac{m\sqrt{D_2T}}{\sqrt{n}}+\frac{mT}{n}} + \frac{n\sigma^2}{\sqrt{D_2}mn+m\sqrt{nT}},\end{aligned} \tag{6.6}$$

where $D_1 = \frac{2}{1-(\rho^2+C)}$, $D_2 = \frac{2}{(1-\sqrt{\rho^2+C\lambda_2(P))^2}}$, $0 < C \leq 1$ is the contraction constant of doubly stochastic matrices and $\lambda_2(P)$ is the second largest eigenvalue of matrix P.

Remark 7. *Theorem 15 shows the asymptotic convergence bound of DeNAV and indicates that DeNAV has a convergence rate of $O(\frac{n}{m\sqrt{nT}})$ when T is large enough and if we make the initial models of all clients the same (i.e., $||\Theta_0-\bar{\Theta}_0\mathbf{1}_n^\intercal||_F^2 = 0$). If $m = n$, DeNAV has a convergence rate of $O(\frac{1}{\sqrt{nT}})$, which is in the same order as previous Gossip methods [82, 131]. If $m = 1$ (without parallel training), DeNAV converges with a rate of $O(\frac{n}{\sqrt{nT}})$.*

Theorem 16. *Under the above assumptions, if η_t is fixed as η and $\eta = \frac{1}{4L\sqrt{D_2}+\frac{\sqrt{T}}{\sqrt{n}}}$, then*

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n}\mathbb{E}||\theta_t^{(i)}-\bar{\Theta}_t||^2 \lesssim \frac{D_1}{T}||\Theta_0-\bar{\Theta}_0\mathbf{1}_n^\intercal||_F^2+\frac{D_2 n(\sigma^2+\zeta^2)}{D_2+\frac{\sqrt{D_2 T}}{\sqrt{n}}+\frac{T}{n}}+\frac{D_2 n}{D_2+\frac{\sqrt{D_2 T}}{\sqrt{n}}+\frac{T}{n}}$$
$$(\frac{n}{\sqrt{D_2}mT+\frac{mT^{\frac{3}{2}}}{\sqrt{n}}}+\frac{D_1}{mT}||\Theta_0-\bar{\Theta}_0\mathbf{1}_n^\intercal||_F^2+\frac{D_2 n\sigma^2+D_2 n\zeta^2}{D_2 m+\frac{m\sqrt{D_2 T}}{\sqrt{n}}+\frac{mT}{n}}+\frac{n\sigma^2}{\sqrt{D_2}mn+m\sqrt{nT}}). \tag{6.7}$$

Remark 8. *The definition of consensus among clients is that the local model on each client should be close to each other at the end of training. Theorem 16 shows the asymptotic consensus bound of DeNAV and indicates that if the number of pre-training steps T is large enough and if we make the initial models of all clients the same, DeNAV can attain consensus with a rate of $O(\frac{n^2}{T})$.*

Proof Sketch. We begin by proving that the communication matrix remains doubly stochastic even when only a subset of $m \in [1, n]$ clients communicate at each pre-training step and bound the averaging error $||\theta_t - \bar{\theta}_t\mathbf{1}_n^\intercal||^2$ for any vectors $\theta_t \in R^N$ after t communication iterations which follows the Markov chain property as follows:

$$\mathbb{E}_{s\ldots(t-1)}||\theta_t-\bar{\theta}_t\mathbf{1}_n^\intercal||^2 \leq (\rho^2+C\cdot\lambda_2(P)^s)^{(t-s)}||\theta_s-\bar{\theta}_s\mathbf{1}_n^\intercal||^2 \tag{6.8}$$

where s is a timestamp that $s < t$, $\bar{\theta}_t$ is the average of θ_t, and $0 \leq \lambda_2(P) < 1$ is the second largest eigenvalue of the transition matrix P. Eq.(6.8) shows that the error between θ_t and $\bar{\theta}_t\mathbf{1}_n^\intercal$ can converge to 0 and implies that the consensus in our scenario can be attained if $\rho^2 + C \cdot \lambda_2(P)^s < 1$. Then, with Eqs.(6.4) and (6.8), we bound the average error $\sum_{t=1}^{T}\sum_{i=1}^{n}\mathbb{E}||\theta_t^{(i)} - \bar{\Theta}_t\mathbf{1}_n^\intercal||^2$ in terms of the initial model parameters Θ_0 and the gradients $G(\Theta_t;\xi_t)$. Based on the assumption in Eq.(6.5), we also derive the asymptotic bound of $\sum_{t=1}^{T}\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2$ and the upper bound of $\mathbb{E}||G(\Theta_t;\xi_t)||$. Next, substituting this upper bound into the upper bound of $\sum_{t=1}^{T}\sum_{i=1}^{n}\mathbb{E}||\theta_t^{(i)} - \bar{\Theta}_t\mathbf{1}_n^\intercal||^2$ reveals the relationship between $\sum_{i=1}^{n}\mathbb{E}||\theta_t^{(i)} - \bar{\Theta}_t||^2$ and $\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2$. However, since $\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2$ is the gradient of the average model for

the case where all n clients perform local training and there are only m clients training in our scenario, we apply the central limit theorem to estimate: $\mathbb{E}_{|m|}||\nabla f(\bar{\Theta}_t)||^2 \approx \frac{n}{m}\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2$. Substituting this into the previous bound and rearranging yields the convergence result in Theorem 15. Finally, combining this with the equation between $\sum_{i=1}^{n}\mathbb{E}||\theta_t^{(i)} - \bar{\Theta}_t||^2$ and $\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2$, we prove Theorem 16.

6.4.2 Impact of Local Data Volume on DeNAV's Training

We next analyse how navigating the training route towards clients with larger local datasets affects the effectiveness of pre-training. Our goal is to justify the data volume utility used in the navigator algorithm by showing that training on clients with more data leads to better model updates for the one-block MAE.

We consider the local training of a one-block MAE learns features from the unlabelled dataset X by reconstruction, represented as a pairwise encoder-decoder relationship $\hat{x} = g(h(x))$, where the encoder $h(\cdot)$ and decoder $g(\cdot)$ are each composed of a single transformer block. Here, the sampled input x has been degraded to $\tilde{x}$ due to masking. Thus, the model parameters are optimised by solving the problem: $\theta^* = \arg\min_\theta \frac{1}{|X|}\sum_{x\in X} l(x, g(h(\tilde{x})); \theta)$, where the loss function l is the mean square error loss. To enable analysis, we adopt the standard assumptions below:

1. (**Smoothness**) $h(\cdot)$ and $g(\cdot)$ are L-smooth with bounded second-order derivatives $\|\nabla^2 h(\cdot)\| \leq \kappa_h$, $\|\nabla^2 g(\cdot)\| \leq \kappa_g$. Similar smoothness assumptions have been widely adopted in theoretical analyses of autoencoders [167].

2. (**Bounded inputs**) Inputs are normalised by rescaling image pixels to $[0, 1]$, ensuring $\|X_i\| \leq B$ for every client $i \in \{1, \ldots, n\}$. This condition is routinely satisfied in practice as part of the pre-processing in MAE pre-training [41].

3. (**Linear learning rate decay**) Training uses gradient descent with a linearly decaying step size $\eta = O(1/T)$, which is the default learning rate schedule used in MAE pre-training [41].

These assumptions reflect the realistic MAE configuration. Under these assumptions, we prove the following key theorem and corollary.

Theorem 17. *In each pre-training step of DeNAV, the approximate optimal solution for the one-block masked autoencoder over the m participating clients is obtained by* W_A^**, in which*

$$W_A^* = \mathbb{X}_t \tilde{\mathbb{X}}_t^T (\tilde{\mathbb{X}}_t \tilde{\mathbb{X}}_t^T)^{-1}, \tag{6.9}$$

where

$$\tilde{\mathbb{X}}_t = [\tilde{X}_i | i \in \mathbb{C}_t]^\intercal, \quad \mathbb{X}_t = [X_i | i \in \mathbb{C}_t]^\intercal, \tag{6.10}$$

$X_i \sim \mathcal{D}_i$ *is the local unlabelled data on client* i*, and* $\tilde{X}_i$ *denotes the masked input.*

Corollary 3. *In each pre-training step of DeNAV, the training effectiveness of the one-block masked autoencoder (MAE) depends on the amount of local data over the selected clients.*

Proof Sketch. We first analyse the architecture of a one-block MAE, where each transformer block in the encoder and decoder consists of a self-attention module and a two-layer feed-forward network with ReLU activation. Prior studies have shown that the outputs of the self-attention function can be represented by a linear combination of a set of basis vectors [95], and that a shallow ReLU network can be replaced by a deep network with linear activation [159]. Building on these results, we can expect the linear approximation error to be very small, and prove via a Taylor expansion around the zero reference point that both the encoder $h(\cdot)$ and decoder $g(\cdot)$ admit linear mappings (W_h, W_g) up to a higher-order Taylor residual. Substituting these linearised operators into the reconstruction loss of the one-block MAE yields Theorem 17, where the optimal solution W_A^* is expressed in terms of the aggregated matrices $\mathbb{X}_t$ and $\tilde{\mathbb{X}}_t$. Then, we examine the deviation of W_A^* and expand the stacked matrices $\tilde{\mathbb{X}}_t$ and $\mathbb{X}_t$. This expansion shows directly that when the selected clients hold more data, these matrices contain more non-zero rows, making them larger and less sparse. In turn, the linear system used to estimate

the optimal mapping has richer information and better conditioning. Therefore, the updates provide a more reliable approximation of the ground-truth mapping, which is exactly the statement of Corollary 3. Finally, although Theorem 17 and Corollary 3 explain a single pre-training step, training runs for many iterations. A natural concern is whether the higher-order Taylor residuals might accumulate and undermine the linear approximation across rounds. To address this, we further prove that under smoothness and bounded-input assumptions, and with a linear decaying learning rate $\eta = O(1/T)$, the cumulative higher-order residual ϵ_T satisfies $|\epsilon_T\| \leq O(1)$ when training with mean square error loss for T steps, ensuring that the linear approximation is not undermined across the entire pre-training process. Besides, we also provide a simple sanity check in Section 6.5.7 to empirically validate these key results.

6.5 Experiments

6.5.1 Experiment Setup

Dataset. In our experiments, we used the Mini-ImageNet dataset for pre-training. This dataset is a subset of the ImageNet [19] dataset, selected through the methodology detailed in [145]. Then, we tested the fine-tuning performance of methods on various benchmark datasets, including CIFAR-10 and CIFAR-100 [70], which are medium-scale datasets with small-sized images, and ImageNet and Mini-iNAT2021 [140], which are large-scale datasets with large-sized images.

Distributed Settings. Our experiments were conducted on a simulated Internet of Things (IoT) network, which is initialised using the Erdős-Rényi model [28]. For this network, we set the total number of clients to be 100 and the network connectivity to be 0.15. Moreover, each client is assigned a random number of data from the Mini-ImageNet dataset. If the local data is assumed to be IID (i.e., independent and identically distributed), then each client will hold images of all classes. Otherwise, if it is assumed that the data follows a non-IID distribution, the dataset will be partitioned

by sampling the class priors of the Dirichlet distribution [51] so that each client will have images of a few classes. Furthermore, to stay close to realistic scenarios, we also assume each client has a different amount of computational resources and set the local training time of a client to be affected by its resources. The impact is formulated as $\tilde{t}_i = \bar{t}_i/(1 + \frac{q_i - 1}{Q-1})$, where $\tilde{t}_i$ is the training time updated in the model state log, $\bar{t}_i$ is the actual time consumed by the client i to train the model, and h_i is the scale of computational resources on the client i (i.e. $\{1 \leq q_i \leq Q | q_i \in \mathbb{Z}\}$ where Q is the maximum scale). Finally, in the training scenario of DeNAV, we transmit and train 5 models in parallel following the standard federated settings, which sample 5 out of 100 clients per round [93, 149, 158, 174]. The details about our default experiment setups are provided in the tables below.

Table 6.1: **Decentralised System Settings of Chapter 6.**

Decentralised System	Value
Number of Clients n	100
Network Connectivity ω	0.15
Pre-training Steps T	200
Local Training Epochs K	5
Number of Participants per Step (Parallel Training) m	5
Fine-tuning Epochs	100
Client Selection Upper Limit Z	3
Staleness Bound (Parallel Training) λ	5
Depth of Downstream Model	5

Table 6.2: **Federated System of Chapter 6.**

Federated System	Value
Number of Clients	100
Pre-training Rounds	200
Local Training Epochs	10
Number of Participants per Round	5
Fine-tuning Epochs	100

6.5.2 Comparison with Federated Self-Supervised Learning

The following state-of-the-art FSSL benchmarks are compared with our proposed DeNAV framework: **1) FedU** [174]; **2) FedEMA** [175]; **3) Orchestra** [93]; **4) FeatARC** [149]; **5) LDAWA** [110]; **6) FedU^2** [83]; and **7) FedMAE** [158]. For all

baselines, the GFLOPs and parameter counts reported in Table 6.3(a) refer to the pre-training phase. Notably, similar to DeNAV, the ViT-based FedMAE adopts a lightweight backbone for pre-training and employs block cascading to expand the backbone during fine-tuning, whereas the ResNet-based baselines retain the same backbone across both stages. To ensure fairness, as shown by the additional entry in Table 6.3(a), the fine-tuning backbone chosen for DeNAV is comparable in size to the ResNet baselines and consistent with FedMAE, ensuring that its performance gains are not attributed to a larger model capacity. Although the parameter counts are similar, DeNAV achieves lower GFLOPs than most baselines during pre-training. This efficiency stems from two design choices: (i) the MAE framework, which masks 75% of image patches and trains only on the remaining 25%; and (ii) the use of the smallest viable MAE, with just one transformer block in both encoder and decoder. Table 6.3(b) further demonstrates that DeNAV delivers consistently higher and more stable fine-tuning accuracy across datasets. It surpasses Orchestra by about 10% on ImageNet and Mini-iNAT, FeatARC by 10% on CIFAR-100 and ImageNet, and FedMAE by 3% on Mini-iNAT. These results highlight DeNAV's ability to achieve strong performance while maintaining superior computational efficiency.

6.5.3 Comparison with Decentralised Training Frameworks

DeNAV is also compared to other DecL approaches. Since All-Reduce is only suitable for high-performance computing clusters, we compare DeNAV with Gossip Learning and Decentralised Random Walk and implement three related baselines: **1) Classical Gossip Learning (C_Gossip)** where each client trains a one-block masked autoencoder and communicates with all its neighbours for model aggregation [130]; **2) Efficient Gossip Learning (E_Gossip)** where each client trains a one-block masked autoencoder and communicates with one random neighbour [131]; and **3) Decentralised Random Walk (Dec_RW)** where a global one-block masked autoencoder is trained by uniform random walking among clients and distributed to each client after training [121]. Table 6.4 shows that under the same computation

cost, DeNAV holds equal communication efficiency as the communication-efficient decentralised methods such as **E_Gossip** and **Dec_RW**, and also outperforms all baselines in fine-tuning accuracy. Furthermore, since DeNAV's training involves additional communication costs for transmitting state logs, we also measure the size of these logs. The results show that the initial log size is 12KB, and after more than 200 training rounds, it can grow to a maximum of 16KB. In comparison to the megabytes of model weights that must be transmitted with each communication, we confirm that the extra communication costs are very small and do not significantly impact the overall communication overhead. Therefore, it appears to us that DeNAV has a better trade-off between high communication efficiency and advanced training effectiveness than existing decentralised approaches.

6.5.4 Ablation Studies

In this section, we conduct ablation studies to examine both the contributions of DeNAV's key components and its adaptability to different architectures. We first evaluate the staleness-aware aggregation and the navigator-based client selection to quantify their individual impact. We then extend DeNAV to CNN backbones to test whether its effectiveness generalises beyond Vision Transformers and to ensure that the observed gains are not solely tied to architectural choice.

Ablation on DeNAV Components. We conducted ablation studies to evaluate the contributions of the staleness-aware aggregation and the navigator-based client selection. As shown in Table 6.5, both modules bring accuracy gains on CIFAR-10 and CIFAR-100. The improvements are particularly evident on the more challenging CIFAR-100 dataset, where staleness-aware aggregation raises accuracy from 73.72% to 74.49%, and the navigator improves performance from 71.70% to 73.70%. When these two modules are integrated into the full DeNAV framework, the gains become substantially larger, as reflected in Table 6.3(b), where DeNAV consistently surpasses all baselines on complex datasets such as ImageNet and Mini-iNAT. Moreover, the

ablation on the navigator algorithm confirms that the effects of its utility components align with our design rationale: the data volume utility U_i^d provides the strongest improvement, the computational resource utility U_i^c enhances accuracy while reducing training time, and the selection history factor α_i offers smaller yet consistent benefits.

Analysis on CNN Adaptation. To further assess the adaptability of DeNAV beyond ViTs, we integrated its two main components, the staleness-aware aggregation and the navigator client selection, into CNN-based pre-training. Based on two simple baselines, Fed-SimSiam [15] and Fed-SimCLR [14], we created the DeNAV-SS and DeNAV-SC variants. As shown in Table 6.6, both achieve consistent gains: DeNAV-SS reaches 92.50% on CIFAR-10 and 71.46% on CIFAR-100, while DeNAV-SC achieves 92.22% and 71.27%, confirming the effectiveness of both modules under CNN backbones. Compared with the training navigator ablation in Table 6.5, the improvement margin is smaller, which aligns with our design rationale: MAE-based ViTs benefit strongly from selecting clients with larger data volumes, whereas CNNs are less sensitive to such factors, reducing the navigator's relative impact. Furthermore, we also note from Table 6.3(b) that the CNN-based DeNAV variants remain highly competitive with ResNet baselines. On CIFAR-10, both DeNAV-SS and DeNAV-SC surpass all ResNet-based methods, and on CIFAR-100 they outperform most baselines except Orchestra. This demonstrates that while DeNAV is tailored for ViTs, its mechanisms also transfer effectively to CNNs, and the performance advantage we observed previously is not attributed to architectural differences.

6.5.5 Hyperparameter Studies

Section 6.3 introduces the important hyperparameters for DeNAV's training. We evaluate the impact of some hyperparameters on the behaviour and performance of DeNAV with the following experiments.

Impact of Total Pre-training Steps T. The performance of DeNAV was evaluated for different numbers of pre-training steps. Figure 6.3(a) demonstrates

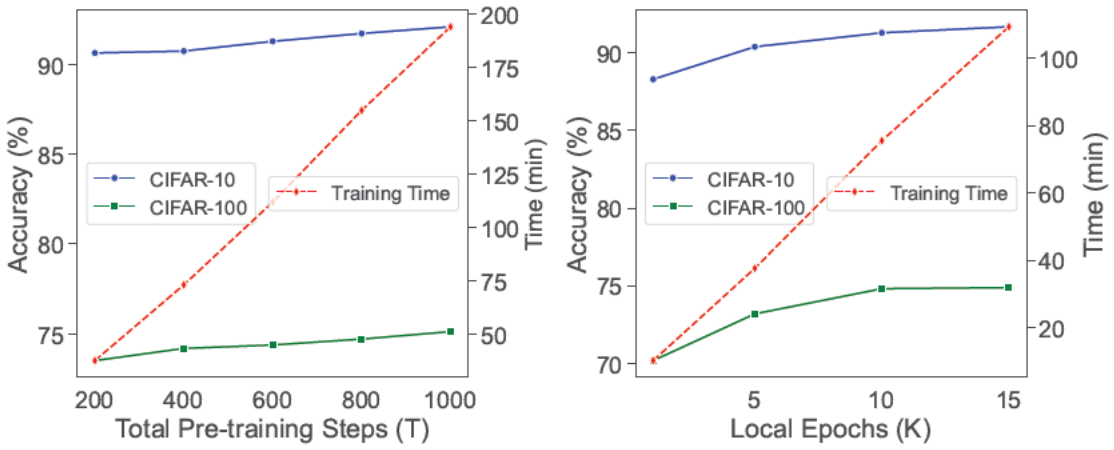


Figure 6.3: **Analysis on (a) the impact of T; and (b) the impact of K on the training of DeNAV.**

that as the number of pre-training steps T increases, DeNAV achieves better training performance. However, the total training time also increases in proportion to the increase in the number of steps. This effect is consistent with the number of training rounds on training performance in FL [174].

Impact of Local Epochs K. The parameter K is the number of iterations for each selected client to train the received model using local data. Similar to the impact of T on training, Figure 6.3(b) shows that increasing K improves the training performance of DeNAV, but the training time is proportional to the value of K. Besides, Figure 6.3(b) also shows that the performance improvement by increasing K converges to a certain value. For DeNAV training, if the extra time cost is affordable, setting the number of local epochs to a value between 10 and 15 will achieve optimal training performance.

Impact of Simultaneously Trained Models m. In FL, a number of clients are sampled to receive the model from the server and perform local training at each round. Similarly, there is a hyperparameter m in our training specifying the number of models that are simultaneously transmitted and trained in the network. Table 6.7(a) shows that increasing m results in an improvement in the training performance

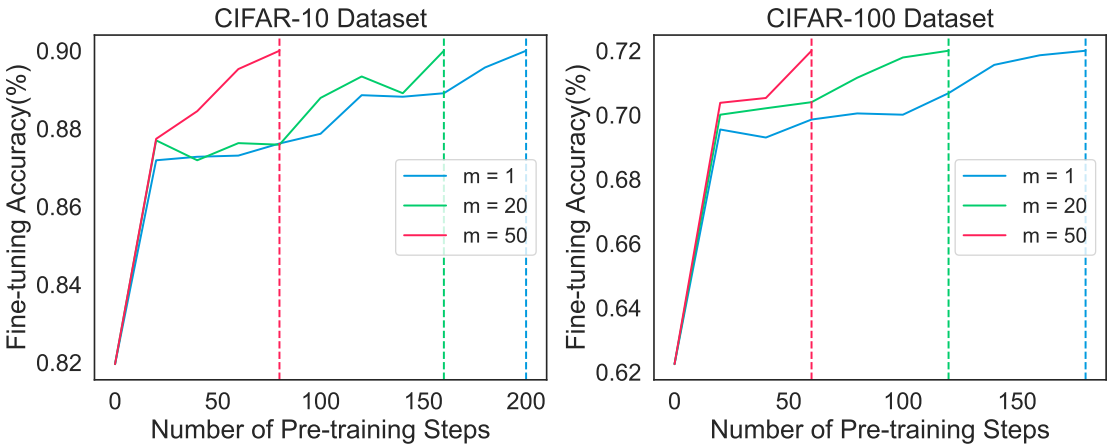


Figure 6.4: **Number of pre-training steps to reach the target fine-tuning accuracy.** We set the target to be 90% for CIFAR-10 and 72% for CIFAR-100, and the number of fine-tuning epochs to be 100. Each line refers to pre-training with a different m. Notably, step 0 represents fine-tuning with random weights.

of DeNAV for the same number of pre-training steps. Besides, setting a large m could also lead to a speedup in pre-training, as shown in Figure 6.4. DeNAV training with $m > 1$ takes fewer pre-training steps than the training with $m = 1$ to reach the target fine-tuning accuracy, and the amount of speedup depends on the value of m.

Impact of Network Connectivity ω. The client-only networks used in experiments were simulated using the Erdős-Rényi model. This model requires two parameters: the number of nodes in the network and the network connectivity ω. A lower value of ω corresponds to a lower probability that each node in the generated network is connected to other nodes. By varying ω, we evaluated the performance of DeNAV across different network structures. As shown in Table 6.7(b), the connectivity of the client network significantly affects DeNAV's training results. Specifically, for the same pre-training steps, the training performance of DeNAV deteriorates if the client network is sparse. Conversely, in dense networks, the performance of DeNAV improves.

Impact of Client Selection Upper Limit Z. The client evaluation in our training navigator relies on a key hyperparameter Z, which limits the maximum

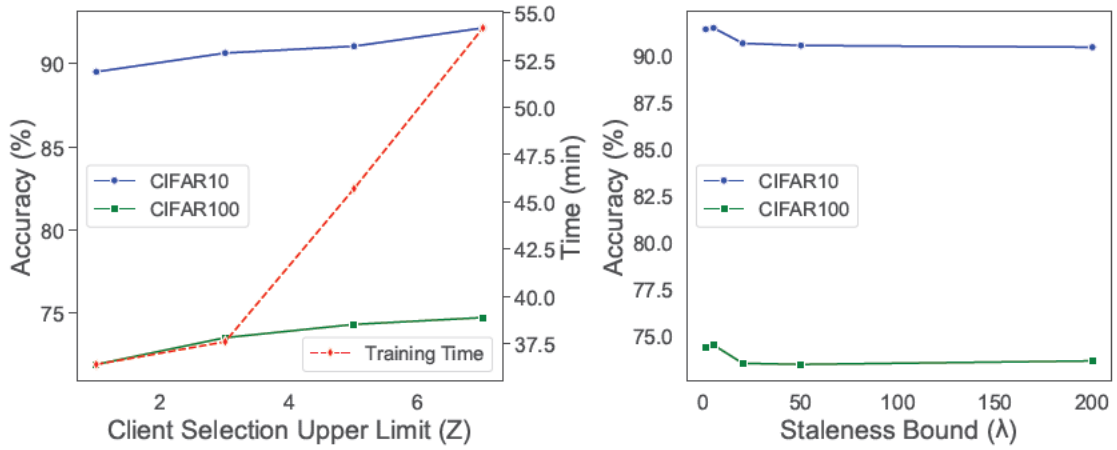


Figure 6.5: **Analysis on (a) impact of Client Selection Upper Limit Z and (b) impact of Staleness Bound λ on training of DeNAV.**

number of times a client can be selected. As shown in Figure 6.5(a), increasing Z enhances training effectiveness but also increases training time. Notably, the performance improvement from $Z = 1$ to $Z = 3$ is more significant than that from $Z = 3$ to $Z = 5$ or $Z = 5$ to $Z = 7$. This is because when $Z = 1$, the condition $Z \geq \frac{T}{n}$ is violated, causing premature termination of exploitation on high-quality clients. Therefore, Z should be chosen to satisfy this condition for effective training.

Impact of Staleness Bound λ. When multiple models are simultaneously pre-trained in DeNAV, the staleness bound λ is employed to restrict the aggregation of local models that have become excessively stale compared to the received model. The results in Figure 6.5(b) illustrate that relaxing λ makes the training of DeNAV worse, but overly tightening λ also results in limited or no aggregation between the received model and the local models on the same client during pre-training, thereby hindering optimal training performance. Therefore, tuning of λ is suggested for practitioners expecting optimal training results. In our experimental scenario, we find that setting $\lambda = 5$ provides the best training results for DeNAV.

6.5.6 Scalability Study

A potential concern for DeNAV is whether relying on a lightweight one-block masked autoencoder (MAE) for pre-training limits its applicability to larger models. To address this, we conducted a scalability study. We used DeNAV to pre-train 1-block, 2-block, and 3-block MAE models on Mini-ImageNet, and fine-tuned a 12-block ViT-Base model (100M parameters) on the large-scale Mini-iNAT dataset. To bridge the architecture differences between pre-training and fine-tuning, we employed a block-wise parameter sharing strategy: for example, with a pre-trained 2-block MAE, the weights of the two encoder blocks are copied to initialise the first two blocks of the 12-block ViT, while the remaining blocks replicate the weights of the first encoder block. As shown in Table 6.8, accuracy increases from 46.32% to 47.01% as the pre-training model grows, while training time and communication cost scale linearly (e.g., 221.65 MB and 54 minutes for 1-block vs. 612.25 MB and 158 minutes for 3-block). These findings confirm that DeNAV scales smoothly to larger models while justifying our default configuration: a 1-block MAE minimises communication in decentralised pre-training, and fine-tuning with parameter sharing fully exploits larger backbones within client resource limits, striking a practical balance between scalability and efficiency.

6.5.7 Sanity Check on Theory

Furthermore, to verify the plausibility of linear approximation analysis in Section 6.4.2, we conducted an empirical sanity check that compares the original one-block MAE with a linearised variant. The linearised model is constructed by removing all nonlinear components in the encoder and decoder blocks, such as ReLU activations and normalisation layers, and replacing them with linear mappings. Both versions were trained on the Mini-ImageNet dataset for 200 rounds in the same decentralised setting. During training, we recorded the reconstruction loss of each model together with the cumulative training sample volume stored in the training state log.

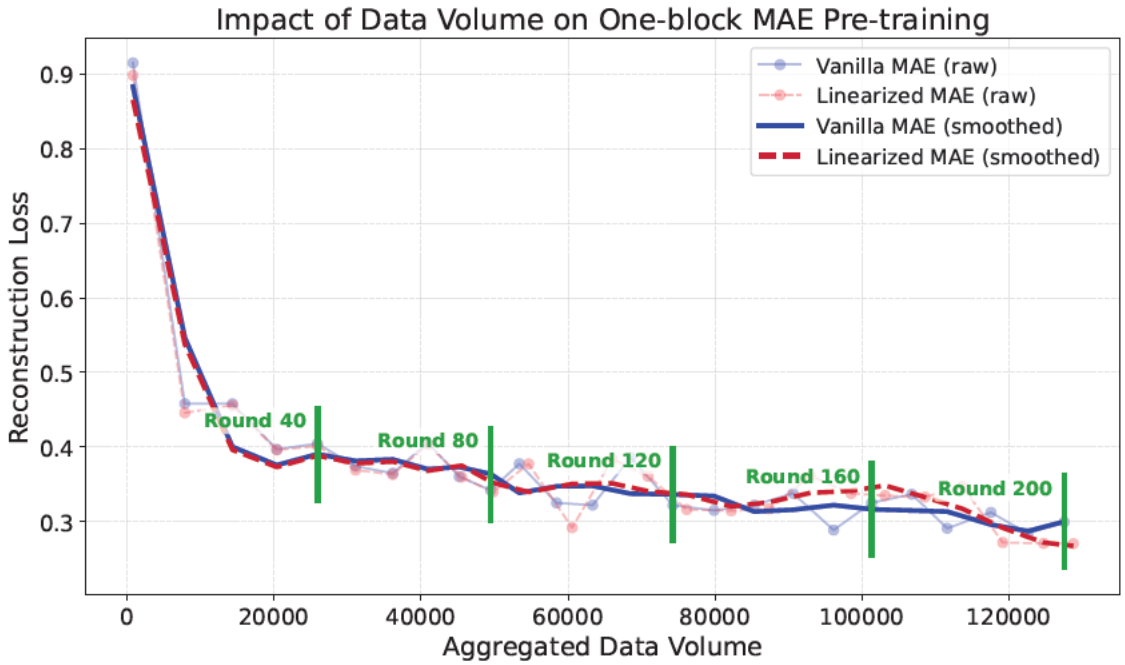


Figure 6.6: **Comparison of the pre-training behaviour between vanilla one-block MAE and linearised MAE.** The blue line shows the reconstruction loss of vanilla one-block MAE, and the red line shows the loss of linearised MAE.

Figure 6.6 reports the comparison between the two models. The blue curve corresponds to the vanilla one-block MAE, and the red curve corresponds to the linearised version. Both curves decrease consistently as the aggregated data volume increases, and the two trajectories remain closely aligned throughout the training process. This empirical evidence confirms that at this scale of single-block architecture, the impact of high-order nonlinear terms on the optimisation dynamics is very limited. Therefore, the Taylor residual ϵ_T can be safely considered bounded as training continues, and the encoder-decoder mappings can be approximated by linear operators. This directly supports the validity of Theorem 17, and Corollary 3.

6.6 Chapter Conclusion

While federated learning provides a way to enable privacy-preserving training across edge clients, its reliance on a central server limits its applicability in fully decentralised real-world settings. In this chapter, we introduced DeNAV,

a decentralised self-supervised learning framework that enables server-free training across clients with only unlabeled data. DeNAV pre-trains multiple lightweight Masked Autoencoders (MAE), aggregates models by taking training staleness into account, and uses a navigator algorithm to guide model transmission based on client-specific scores. Our theoretical analysis proves convergence and consistency guarantees, and further explains why selecting clients with larger datasets enhances DeNAV's training performance. Extensive experiments show that DeNAV achieves performance comparable to state-of-the-art federated SSL and surpasses prior decentralised baselines under equal communication budgets.

6.7 Chapter Notations and Definitions

i, j	Client indexes
$\mathcal{G}, \mathcal{C}, \mathcal{E}$	Client graph, Node set, Edge set
e	Edge
n	Number of clients in the network
$\mathcal{D}$	Dataset
x, X	Image data
q	Computational resource scale
θ, Θ	Model Parameters
f, F	Loss function
m	Number of simultaneously trained models
T	Number of pre-training steps
s, t	Step index
$\mathbb{C}$	Selected set of training clients
$\mathcal{C}_i$	Client selection candidates for client i
ψ, Ψ	Training state logs
K	Local training iterations
k	Iteration index
λ	Staleness bound for asynchronous model aggregation
Z	Maximum number of selections per client
ω	Network connectivity
U	Client selection utility
α	Selection history factor
w	Model aggregation weights
G	Set of loss gradients across all clients
$\mathbb{W}$	Communication topology
σ, ζ, κ	Bound constants defined in assumptions
P	Transition matrix of communication

$\mathcal{V}$	State space
S	State of communication process
π	Stationary distribution of communication process
ρ, C	Constants depending on communication transition
λ_2	Second largest eigenvalue of P
η	Learning rate
$h(\cdot), g(\cdot)$	Encoding/Decoding function
$\mathbb{X}$	Aggregated set of image data across clients
ϵ	Linear approximation error
σ, ζ, κ	Bound constants defined in assumptions
P	Transition matrix of communication
$\mathcal{V}$	State space
S	State of communication process
π	Stationary distribution of communication process
ρ, C	Constants depending on communication transition
λ_2	Second largest eigenvalue of P
η	Learning rate
$h(\cdot), g(\cdot)$	Encoding/Decoding function
$\mathbb{X}$	Aggregated set of image data across clients
ϵ	Linear approximation error
$\bar{t}$	Actual training clock time on client

Table 6.3: **Comparison of DeNAV with FSSL baselines.** (a) The size of the input image is 224x224. In our experiment settings, DeNAV pre-trains the one-block masked autoencoder, constructs a transformer backbone with 5 blocks by parameter sharing, and fine-tunes the backbone for downstream evaluation. (b) For pre-training, the local training epochs were set to 10. For the downstream evaluation, each model was fine-tuned for 100 epochs. The experimental results show the mean of three trials.

	FedU	FedEMA	Orchestra	FeatARC	LDAWA	FedU^2
Params(M)	38.47	38.47	11.84	11.70	15.39	15.39
GFLOPs	7.40	7.40	7.31	1.82	1.83	1.83

	FedMAE	DeNAV	FedMAE/DeNAV (Fine-tune)
Params(M)	**11.62**	**11.62**	39.97
GFLOPs	**1.23**	**1.23**	7.39

(a) Model parameters and GFLOPs.

Method	Architecture	CIFAR-10(%)		CIFAR-100(%)		ImageNet(%)		Mini-iNAT(%)	
		IID	non-IID	IID	non-IID	IID	non-IID	IID	non-IID
FedU	ResNet	77.43	72.02	40.40	38.44	65.34	65.34	37.88	37.61
FedEMA	ResNet	70.73	71.00	40.78	41.13	65.24	65.35	38.40	37.43
Orchestra	ResNet	88.87	90.66	72.11	72.27	65.02	66.50	38.74	39.23
FeatARC	ResNet	90.22	90.03	64.80	64.11	68.62	68.17	45.84	44.50
LDAWA	ResNet	90.52	89.95	69.94	68.96	52.36	51.43	37.70	37.60
FedU^2	ResNet	86.97	82.39	63.66	55.49	48.73	45.27	32.57	31.16
FedMAE	ViT	90.62	90.47	74.11	73.74	77.10	76.95	43.01	41.27
DeNAV	ViT	**91.12**	**91.00**	**74.50**	**73.89**	**77.49**	**77.62**	**46.38**	**44.98**

(b) Fine-tuning Accuracy on CIFAR-10, CIFAR-100, ImageNet, and Mini-iNAT.

Table 6.4: **Comparison between DeNAV and other decentralised methods.** "Computation (COMPU)" represents the total number of epochs for all training clients. "Communication (COMMU)" indicates the total number of model transmissions between all training clients and their neighbours.

	C_Gossip			E_Gossip			Dec_RW			DeNAV (Ours)		
COMPU	5000	15000	25000	5000	15000	25000	5000	15000	25000	5000	15000	25000
COMMU	14850	44550	74250	1000	3000	5000	1000	3000	5000	**1000**	**3000**	**5000**
CIFAR-10(%)	90.82	92.66	93.13	90.52	92.30	92.91	90.38	92.46	93.46	**92.80**	**93.44**	**93.88**
CIFAR-100(%)	73.52	75.48	76.33	72.34	72.52	75.71	74.27	76.59	76.60	**75.62**	**77.30**	**77.94**

Table 6.5: **Ablation study on the main components of DeNAV.** The left part shows results for the staleness-aware model aggregation, and the right part shows results for the training navigator algorithm.

(a) Analysis on Aggregation			**(b) Analysis on Client Selection**			
Aggregation	CIFAR-10	CIFAR-100	Method	CIFAR-10	CIFAR-100	Time(min)
Average	90.64	73.79	Random	90.28	71.70	33.7
Data Volume	91.08	73.72	w/o U_i^d	88.54	71.99	28.8
Staleness-aware	**91.46**	**74.49**	w/o U_i^c	90.65	73.46	37.6
			w/o α_i	90.23	73.51	38.9
			Our Formula	**90.96**	**73.70**	36.8

Table 6.6: **Ablation study on the adaptability of DeNAV on CNN pre-training.** We integrate Fed-SimSiam and Fed-SimCLR with the staleness-aware aggregation and training navigator in DeNAV, and still observe performance improvements.

	Fed-SimSiam	Dec-SS	DeNAV-SS	Fed-SimCLR	Dec-SC	DeNAV-SC
Architecture	ResNet	ResNet	ResNet	ResNet	ResNet	ResNet
Client Selection	Random	Random	Ours	Random	Random	Ours
Aggregation	Dataset Size	Ours	Ours	Dataset Size	Ours	Ours
CIFAR-10(%)	89.58	91.82	**92.50**	90.39	92.10	**92.22**
CIFAR-100(%)	70.95	70.71	**71.46**	71.24	70.73	**71.27**

Table 6.7: **Impact of m and ω on DeNAV.** The results report accuracy on CIFAR-10 and CIFAR-100 after 200 steps of pre-training.

(a) Impact of Number of Models m			**(b) Impact of Network Connectivity ω**		
	CIFAR-10(%)	CIFAR-100(%)		CIFAR-10(%)	CIFAR-100(%)
$m = 1$	90.90	73.81	$\omega = 0.03$	89.17	72.01
$m = 5$	91.48	73.44	$\omega = 0.15$	90.64	73.51
$m = 10$	91.81	74.04	$\omega = 0.75$	**91.16**	**73.65**
$m = 15$	**91.88**	**75.25**			

Table 6.8: **Scalability analysis of DeNAV.** We pre-train 1-block, 2-block, and 3-block MAE models and fine-tune a large 12-block ViT-Base. Communication cost is measured per round with 5 model updates being exchanged across clients and in float32 precision.

Pre-trained Model	**Params**	**COMMU Cost**	**Training Time**	**Mini-iNAT(%)**
1-Block MAE	11.62 M	221.65 MB	54m25s (per model)	46.32
2-Block MAE	21.86 M	416.95 MB	104m42s (per model)	46.63
3-Block MAE	32.10 M	612.25 MB	157m38s (per model)	47.01

CHAPTER 7

Thesis Conclusion

7.1 Summary of Thesis

This thesis is motivated by the observation that modern visual data, though abundant, is typically generated and stored in a distributed manner across mobile devices, cameras, and institutions. Centralising such visual data is often infeasible due to privacy concerns, communication bottlenecks, and ownership restrictions, while annotations remain scarce and expensive to obtain. At the same time, the success of large-scale deep learning has shown that scaling models on massive datasets is critical for learning powerful visual representations. Reconciling these realities highlights the importance of developing distributed self-supervised learning (D-SSL) in the visual domain to enable efficient model training using distributed data at low cost.

Building on this motivation, the thesis first situates its contributions within the broader research landscape through an extensive review of prior work. The literature review chapter surveys three key strands that underpin this research: self-supervised learning for representation learning without labels, distributed training paradigms such as federated and decentralised learning, and recent efforts that attempt to bridge these two areas. This review clarifies the state of knowledge before this thesis, highlighting both the empirical progress and the theoretical gaps that remain. In particular, while SSL methods such as contrastive learning and masked image modelling have achieved great success in centralised scenarios, their adaptation to distributed settings is far less understood. Similarly, while federated and decentralised

frameworks enable training across distributed data sources, most existing analyses rely on supervised tasks and lack a deep theoretical treatment of unlabelled data. These gaps provide the foundation upon which the thesis develops its four major studies, each addressing a different but connected dimension of D-SSL.

The first study, presented in Chapter 3, investigates how scaling laws adapt when training shifts from centralised to federated settings. By deriving PAC-Bayesian generalisation bounds for stochastic gradient descent under both regimes, it establishes closed-form solutions for compute-optimal model size. The analysis reveals that decentralisation fundamentally alters scaling behaviour: as data becomes more distributed across clients, the optimal model size decreases. This finding provides a theoretical explanation for why smaller models are more effective on edge devices with limited local data and compute. The study not only contributes to the theory of scaling laws but also gives concrete design guidelines for choosing model sizes in distributed environments.

Chapter 4 extends the investigation to a more fundamental question: can distributed training ever match centralised training under equal resources? Using the uniform stability analysis and the PAC-Bayesian framework, this chapter formally defines the performance gap as the distance between the centralised and federated generalisation bounds. It proves that such a gap inevitably exists when both operate under the same total compute, and that enlarging model size or increasing communication rounds cannot eliminate the discrepancy. Instead, the only way to close the gap is through giving distributed training advantages in training data, either by adding more clients or increasing the local dataset size of existing clients, with the latter being the more efficient option. These theoretical insights are validated by extensive experiments across different architectures and datasets, confirming the practical manifestation of the generalisation gap. This chapter thus settles a long-standing debate about whether learning in distributed systems can truly rival centralised learning under equal training resources.

Chapter 5 shifts the focus from supervised settings to self-supervised learning (SSL), which is particularly promising in distributed scenarios dominated by unlabelled data. It provides the first systematic theoretical analysis of D-SSL under heterogeneous client data. By constructing mathematical models of different D-SSL frameworks, the chapter shows that methods based on masked image modelling (MIM) are inherently more robust to data heterogeneity than contrastive learning (CL) methods. It also proves that robustness improves with greater network connectivity, while federated and decentralised frameworks achieve comparable robustness when the network is fully connected. Motivated by these results, the chapter introduces MAR loss, a refinement of MIM loss that incorporates alignment regularisation to enhance robustness. Experiments under varying levels of heterogeneity validate these theoretical findings and the effectiveness of MAR loss. In summary, this study establishes SSL as a promising paradigm for distributed representation learning and demonstrates that refined objectives such as MAR loss can effectively mitigate the challenges posed by heterogeneous data.

Finally, Chapter 6 proposes DeNAV, a decentralised self-supervised learning framework designed for realistic large-scale distributed systems where clients hold only unlabelled data and cannot rely on a central server. DeNAV introduces three technical innovations: a navigator algorithm that guides the training route of models to maximise exposure to diverse client data, staleness-aware aggregation to handle asynchronous updates, and lightweight MAE-based pre-training to reduce communication cost while preserving robustness. The framework allows multiple models to be trained in parallel and later scaled into larger backbones via weight sharing. On the theoretical side, the chapter also presents two complementary results. It proves that DeNAV achieves convergence and consensus guarantees, and justifies the navigator design by showing that selecting clients with larger local datasets improves training. Comprehensive experiments confirm these contributions: DeNAV performs on par with federated SSL baselines and surpasses prior decentralised approaches. Beyond baseline comparisons, ablation studies and hyperparameter

sensitivity analyses further demonstrate the effectiveness and robustness of the framework. This chapter illustrates how theoretical principles can be translated into practical algorithmic design for distributed self-supervised learning.

Taken together, the four studies form a coherent line of research that bridges theoretical analysis and practical algorithm design for distributed self-supervised learning. The first two studies establish fundamental theoretical principles: from the generalisation view, training in distributed systems should favour small models, and the only viable way to close the gap to centralised training is through data size advantage by involving more clients or enlarging their local datasets. The third study shifts the focus to self-supervised learning and shows, through rigorous theoretical analysis, that MIM is inherently more robust to heterogeneous data than CL, and that robustness improves with higher network connectivity. The fourth study then synthesises these insights into DeNAV, a decentralised self-supervised framework which: (1) utilises lightweight model architecture in pre-training to be compute-optimal; (2) employs a navigator algorithm to prioritise clients with richer data and broader coverage for better generalisation; and (3) adopts masked autoencoding as its core learning strategy to align with the result that MIM is less sensitive to the negative impact of heterogeneous data. Theoretical analysis further confirms DeNAV's convergence, consensus, and data-driven property, while comprehensive experiments validate its practical effectiveness. Beyond DeNAV, this thesis also introduces MAR loss, a flexible refinement of MIM loss that strengthens robustness by aligning local and global representations, which can be combined with DeNAV or future decentralised MIM algorithms to improve performance.

Overall, this thesis demonstrates how rigorous theoretical analysis and concrete algorithmic innovation can jointly advance self-supervised visual representation learning in distributed systems. By unifying scaling law analysis, generalisation theory, robustness study, and practical framework design, it provides a roadmap for developing scalable, communication-efficient, and effective methods that exploit the

vast and growing reservoir of unlabelled visual data at the network edge, laying the groundwork for future research on large-scale decentralised AI systems.

7.2 Discussions on Future Work

Building on these findings, several important directions emerge for further investigation. A primary challenge is how to extend distributed self-supervised learning to support large-scale model pre-training. While self-supervised learning has proven highly effective for training large models in centralised settings, achieving similar scalability in decentralised environments remains fundamentally challenging due to the absence of a central coordinator and the limited data and computational resources available on individual clients. Although the aggregated data volume across clients may be substantial, each client typically observes only a small and biased subset of the global distribution. As a result, effectively coordinating model capacity, communication frequency, and local training dynamics to approach the performance of centralised large-scale pre-training, while maintaining efficiency, remains an open and practically important problem.

Another promising direction lies in relaxing the common assumption of homogeneous model architectures across clients. In realistic distributed systems, clients often exhibit significant variability in computational capabilities, memory constraints, and energy budgets, which naturally leads to the use of models with different sizes or architectures [160]. However, most existing frameworks, including those studied in this thesis, assume a shared model structure for simplicity and tractability. Enabling heterogeneous models to be jointly trained, while allowing meaningful knowledge exchange across different architectures, raises new challenges in representation alignment, parameter compatibility, and aggregation design. Addressing these challenges could significantly broaden the applicability of decentralised learning in real-world systems.

In addition, as shown in Chapter 5, contrastive learning and masked image modelling are regarded as two representative paradigms of self-supervised learning.

While these approaches are often studied independently, recent advances in centralised settings suggest that combining them can lead to more robust and expressive representations [54, 168]. In decentralised environments, the presence of multiple clients introduces new opportunities for such hybrid strategies. For instance, clients may adopt combined objectives locally, or different subsets of clients may specialise in distinct objectives and collaboratively contribute to a shared representation. Understanding how to design, coordinate, and optimise such hybrid learning strategies across distributed clients remains an open question, with the potential to leverage the complementary strengths of different self-supervised paradigms.

Furthermore, most existing approaches aim to learn a single global representation model that is later adapted to individual clients through fine-tuning. However, in highly heterogeneous environments, a globally optimal model may not be optimal for every client due to variations in data distribution and downstream task requirements [112, 128]. This motivates the exploration of personalised distributed self-supervised learning, where representation learning is directly tailored to individual clients during the pre-training stage. Such an approach raises fundamental questions about how to balance global knowledge sharing with local specialisation, and how to design training mechanisms that enable efficient personalisation without sacrificing collaboration benefits.

Finally, the theoretical analysis in this thesis is primarily developed under simplified non-IID assumptions based on label distribution heterogeneity, which enables tractable analysis and clear insights. In practical scenarios, however, data heterogeneity is often significantly more complex, involving feature distribution shifts, domain discrepancies, and temporal variations across clients. Extending the current theoretical framework to capture these richer forms of heterogeneity, and systematically analysing their impact on generalisation, robustness, and optimisation, would further strengthen the connection between theory and real-world applications. Such developments could provide more precise and actionable guidance for designing distributed self-supervised learning systems in diverse and dynamic environments.

CHAPTER 8

Full Proofs of Theoretical Analyses

Chapter Overview: This chapter provides the complete theoretical proofs that form the rigorous foundations of the four main studies presented in Chapters 3–6. While earlier chapters summarised the key results and highlighted their implications, here we present the full derivations to ensure transparency, reproducibility, and mathematical rigour.

- Section 8.1 presents the proofs supporting Chapter 3. These include the derivation of the generalisation bound for federated SGD, the formal relationship between compute-optimal model sizes in centralised and federated settings, theoretical evidence for the inferior generalisation in distributed training, and the analytic method for estimating optimal model size from the average compute across clients.

- Section 8.2 details the proofs corresponding to Chapter 4. It studies the stability of decentralised and federated learning and includes formal derivations of Theorems 7–10, which characterise the inevitability of a PAC-Bayesian generalisation gap under equal resources and the training advantage conditions for fully closing it.

- Section 8.3 contains the proofs supporting Chapter 5. It first analyses representability for distributed masked image modelling (MIM) and contrastive learning, then rigorously proves the two central insights: (1) MIM's inherent

robustness compared to CL under heterogeneous data, and (2) the role of network connectivity in strengthening the non-IID robustness.

- Section 8.4 provides the proofs associated with Chapter 6. These include proofs of the convergence and consistency guarantees of the decentralised training process implemented in DeNAV, as well as an analysis of how the amount of local data affects the training effectiveness of DeNAV.

By presenting these proofs in full, this chapter consolidates the theoretical rigour underlying the thesis's contributions and ensures that the key results in previous chapters rest on a transparent and well-founded analytical basis.

8.1 Proofs of Chapter 3

8.1.1 Proof of Generalisation Bound for Federated SGD

This section provides the proof details for Lemma 2, Lemma 3, Theorem 1, and Theorem 2 of theoretical analysis in section 3.4.2.

We start our analysis of the generalisation with the proof of Lemma 2.

Proof. From the result of the Ornstein-Uhlenbeck process [139], the analytical solution for the SGD training with local data from client i in the first round $j = 1$ will be

$$\theta_i(1) = \theta_i(0)e^{-A_i t} + \sqrt{\frac{\eta}{k_{Fed}m}} \int_0^t e^{-A_i(t-t')} B_i dW(t'). \tag{8.1}$$

where $W(t')$ is a white noise and follows $\mathcal{N}(0, I)$. Since local models will be aggregated on the server at each round of federated learning, the analytic solution for local training on client i at the second round $j = 2$ should be

$$\theta_i(2) = \frac{1}{n}\sum_{i=1}^{n} \theta_i(1)e^{-A_i t} + \sqrt{\frac{\eta}{k_{Fed}m}} \int_0^t e^{-A_i(t-t')} B_i dW(t'). \tag{8.2}$$

Substituting Eq.(8.1) into Eq.(8.2), we have

$$\begin{aligned}
\theta_i(2) = &\frac{1}{n}\sum_{i=1}^{n} \left(\theta_i(0)e^{-A_i t} + \sqrt{\frac{\eta}{k_{Fed}m}} \int_0^t e^{-A_i(t-t')} B_i dW(t') \right) e^{-A_i t} \\
&+ \sqrt{\frac{\eta}{k_{Fed}m}} \int_0^t e^{-A_i(t-t')} B_i dW(t') \\
= &\theta_i(0)e^{-2\bar{A}t} + \sqrt{\frac{\eta}{k_{Fed}m}} \int_{-t}^{0} e^{-\bar{A}(t-t')} \bar{B} dW(t') + \sqrt{\frac{\eta}{k_{Fed}m}} \int_0^t e^{-A_i(t-t')} B_i dW(t').
\end{aligned} \tag{8.3}$$

In the same way, we formulate the analytic solution in the round $j = 3$ as follows:

$$\begin{aligned}
\theta_i(3) &= \frac{1}{n}\sum_{i=1}^{n}(\theta_i(0)e^{-2\bar{A}t} + \sqrt{\frac{\eta}{k_{Fed}m}}\int_{-t}^{0} e^{-\bar{A}(t-t')}\bar{B}dW(t') \\
&\quad + \sqrt{\frac{\eta}{k_{Fed}m}}\int_{0}^{t} e^{-A_i(t-t')}B_i dW(t'))e^{-A_i t} + \sqrt{\frac{\eta}{k_{Fed}m}}\int_{0}^{t} e^{-A_i(t-t')}B_i dW(t') \\
&= \theta_i(0)e^{-2\bar{A}t}\frac{1}{n}\sum_{i=1}^{n} e^{-A_i t} + \sqrt{\frac{\eta}{k_{Fed}m}}\int_{-t}^{0} e^{-\bar{A}(t-t')}\bar{B}dW(t')\frac{1}{n}\sum_{i=1}^{n} e^{-A_i t} \\
&\quad + \sqrt{\frac{\eta}{k_{Fed}m}}\frac{1}{n}\sum_{i=1}^{n}\int_{0}^{t} e^{-A_i(t-t')}e^{-A_i t}B_i dW(t') + \sqrt{\frac{\eta}{k_{Fed}m}}\int_{0}^{t} e^{-A_i(t-t')}B_i dW(t') \\
&= \theta_i(0)e^{-3\bar{A}t} + \sqrt{\frac{\eta}{k_{Fed}m}}\left(\int_{-2t}^{-t} e^{-\bar{A}(t-t')}\bar{B}dW(t') + \int_{-t}^{0} e^{-\bar{A}(t-t')}\bar{B}dW(t')\right) \\
&\quad + \sqrt{\frac{\eta}{k_{Fed}m}}\int_{0}^{t} e^{-A_i(t-t')}B_i dW(t') \\
&= \theta_i(0)e^{-3\bar{A}t} + \sqrt{\frac{\eta}{k_{Fed}m}}\int_{-2t}^{0} e^{-\bar{A}(t-t')}\bar{B}dW(t') + \sqrt{\frac{\eta}{k_{Fed}m}}\int_{0}^{t} e^{-A_i(t-t')}B_i dW(t').
\end{aligned} \tag{8.4}$$

Similarly, the analytic solution after T rounds of federated training can be derived as the following equation:

$$\begin{aligned}
&\theta_{Fed}(T) = \frac{1}{n}\sum_{i=1}^{n}\theta_i(T) \\
&= \theta_i(0)e^{-T\bar{A}t} + \sqrt{\frac{\eta}{k_{Fed}m}}\int_{(1-T)t}^{0} e^{-\bar{A}(t-t')}\bar{B}dW(t') + \sqrt{\frac{\eta}{k_{Fed}m}}\frac{1}{n}\sum_{i=1}^{n}\int_{0}^{t} e^{-A_i(t-t')}B_i dW(t') \\
&= \theta_i(0)e^{-T\bar{A}t} + \sqrt{\frac{\eta}{k_{Fed}m}}\int_{(1-T)t}^{0} e^{-\bar{A}(t-t')}\bar{B}dW(t') + \sqrt{\frac{\eta}{k_{Fed}m}}\int_{0}^{t} e^{-\bar{A}(t-t')}\bar{B}dW(t') \\
&= \theta_i(0)e^{-T\bar{A}t} + \sqrt{\frac{\eta}{k_{Fed}m}}\int_{(1-T)t}^{t} e^{-\bar{A}(t-t')}\bar{B}dW(t') \\
&= \theta_i(0)e^{-T\bar{A}t} + \sqrt{\frac{\eta}{k_{Fed}m}}\frac{1-e^{-T\bar{A}t}}{\bar{A}}\bar{B} \\
&= \theta_0 e^{-T\bar{A}t} + \sqrt{\frac{\eta}{k_{Fed}m}}\frac{T(1-e^{-T\bar{A}t})}{T\bar{A}}\bar{B} \\
&= \theta_0 e^{-T\bar{A}t} + T\sqrt{\frac{\eta}{k_{Fed}m}}\int_{0}^{t} e^{-T\bar{A}(t-t')}\bar{B}dW(t').
\end{aligned} \tag{8.5}$$

which completes the proof. □

Then, we use the results in Lemma 2 to prove Lemma 3.

Proof. From Eq.(3.13), we know that

$$\Sigma_{Fed} = \mathbb{E}_{\theta\sim Q}[\theta_{Fed}\theta_{Fed}^{\intercal}]. \tag{8.6}$$

Then, according to Eq.(8.5), we can derive the following equation:

$$\begin{aligned} T\bar{A}\Sigma_{Fed} + \Sigma_{Fed}T\bar{A} &= \frac{T^2\eta}{k_{Fed}m}\int_{-\infty}^{t} T\bar{A}e^{-T\bar{A}(t-t')}\bar{C}e^{-T\bar{A}(t-t')}dt' \\ &\quad + \frac{T^2\eta}{k_{Fed}m}\int_{-\infty}^{t} e^{-T\bar{A}(t-t')}\bar{C}e^{-T\bar{A}(t-t')}dt'T\bar{A} \\ &= \frac{T^2\eta}{k_{Fed}m}\int_{-\infty}^{t}\frac{d}{dt'}(e^{-T\bar{A}(t-t')}\bar{C}e^{-T\bar{A}(t-t')}) \\ &= \frac{T^2\eta}{k_{Fed}m}\bar{C}. \end{aligned} \tag{8.7}$$

which completes the proof. □

Based on the above two lemmas and Lemma 1, we can establish the PAC-Bayesian generalisation bound in Theorem 1 as follows.

Proof. Similarly to the classical PAC-Bayesian framework, we suppose the prior distribution over the parameter space θ is P and the distribution of the learned hypothesis from the federated SGD algorithm is Q. Then according to Eq.(3.13), the densities of the stationary distribution Q and the prior distribution P are respectively $q(\theta)$ and $p(\theta)$ in terms of the parameter θ and can be expressed as the following equations:

$$\begin{aligned} q(\theta) &= \frac{1}{\sqrt{2\pi\det(\Sigma_{Fed})}}\exp\left\{-\frac{1}{2}\theta^{\intercal}\Sigma_{Fed}^{-1}\theta\right\}, \\ p(\theta) &= \frac{1}{\sqrt{2\pi\det(I)}}\exp\left\{-\frac{1}{2}\theta^{\intercal}I\theta\right\}. \end{aligned} \tag{8.8}$$

Thus we have

$$\begin{aligned}\log\left(\frac{q(\theta)}{p(\theta)}\right) &= \log\left(\frac{\sqrt{2\pi\det(I)}}{\sqrt{2\pi\det(\Sigma_{Fed})}}\exp\left\{\frac{1}{2}\theta^{\intercal}I\theta - \frac{1}{2}\theta^{\intercal}\Sigma_{Fed}^{-1}\theta\right\}\right) \\ &= \frac{1}{2}\log\left(\frac{1}{\det(\Sigma_{Fed})}\right) + \frac{1}{2}\left(\theta^{\intercal}I\theta - \theta^{\intercal}\Sigma_{Fed}^{-1}\theta\right).\end{aligned} \tag{8.9}$$

Here we can calculate the KL divergence between the distribution Q and P by applying Eq.(3.11) in Lemma 1:

$$\begin{aligned}D(Q||P) &= \mathbb{E}_{\theta\sim Q}\left(\log\frac{Q(\theta)}{P(\theta)}\right) \\ &= \int_{\theta\in\Theta}\log\left(\frac{q(\theta)}{p(\theta)}\right)q(\theta)d\theta \\ &= \int_{\theta\in\Theta}\left[\frac{1}{2}\log\left(\frac{1}{\det(\Sigma_{Fed})}\right) + \frac{1}{2}\left(\theta^{\intercal}I\theta - \theta^{\intercal}\Sigma_{Fed}^{-1}\theta\right)\right]q(\theta)d\theta \\ &= \frac{1}{2}\log\left(\frac{1}{\sqrt{\det(\Sigma_{Fed})}}\right) + \frac{1}{2}\int_{\theta\in\Theta}\theta^{\intercal}I\theta q(\theta)d\theta - \frac{1}{2}\int_{\mathbb{R}^{|\mathcal{S}|}}\theta^{\intercal}\Sigma_{Fed}^{-1}q(\theta)d\theta \\ &= \frac{1}{2}\log\left(\frac{1}{\sqrt{\det(\Sigma_{Fed})}}\right) + \frac{1}{2}\mathbb{E}_{\theta\sim\mathcal{N}(0,\Sigma_{Fed})}\theta^{\intercal}I\theta - \frac{1}{2}\mathbb{E}_{\theta\sim\mathcal{N}(0,\Sigma_{Fed})}\theta^{\intercal}\Sigma_{Fed}^{-1}\theta \\ &= \frac{1}{2}\log\left(\frac{1}{\sqrt{\det(\Sigma_{Fed})}}\right) + \frac{1}{2}\mathrm{tr}(\Sigma_{Fed} - I).\end{aligned} \tag{8.10}$$

Since we have proved from Lemma 3 that $T\bar{A}\Sigma_{Fed} + \Sigma_{Fed}T\bar{A} = \frac{T^2\eta}{k_{Fed}m}\bar{C}$, we have

$$\begin{aligned}\bar{A}\Sigma_{Fed}\bar{A}^{-1} + \Sigma_{Fed} &= \frac{T^2\eta}{Tk_{Fed}m}\bar{C}\bar{A}^{-1} \\ \mathrm{tr}(\bar{A}\Sigma_{Fed}\bar{A}^{-1} + \Sigma_{Fed}) &= \mathrm{tr}(\frac{T\eta}{k_{Fed}m}\bar{C}\bar{A}^{-1}).\end{aligned} \tag{8.11}$$

For the left-hand side, we can change it to the following equation:

$$\begin{aligned} \text{LHS} &= \text{tr}(\bar{A}\Sigma_{Fed}\bar{A}^{-1} + \Sigma_{Fed}) \\ &= \text{tr}(\bar{A}\Sigma_{Fed}\bar{A}^{-1}) + \text{tr}(\Sigma_{Fed}) \\ &= \text{tr}(\bar{A}\bar{A}^{-1}\Sigma_{Fed}) + \text{tr}(\Sigma_{Fed}) \\ &= \text{tr}(\Sigma_{Fed}) + \text{tr}(\Sigma_{Fed}) \\ &= 2\text{tr}(\Sigma_{Fed}). \end{aligned} \tag{8.12}$$

Therefore,

$$\text{tr}(\Sigma_{Fed}) = \frac{1}{2}\text{tr}(\frac{T\eta}{k_{Fed}m}\bar{C}\bar{A}^{-1}) = \frac{T\eta}{2k_{Fed}m}\text{tr}(\bar{C}\bar{A}^{-1}). \tag{8.13}$$

On the other side, we can simply calculate that $\text{tr}(I) = d$, because $I \in \mathbb{R}^{d\times d}$, where d is the dimension of the parameter θ. Then we can have

$$\begin{aligned} D(Q_{Fed}||P) &= -\frac{1}{2}\log(\det(\Sigma_{Fed})) + \frac{1}{2}\text{tr}(\Sigma_{Fed}) - \frac{1}{2}\text{tr}(I) \\ &= -\frac{1}{2}\log(\det(\Sigma_{Fed})) + \frac{T\eta}{4k_{Fed}m}\text{tr}(\bar{C}\bar{A}^{-1}) - \frac{1}{2}d. \end{aligned} \tag{8.14}$$

By inserting Eq.(8.14) into Eq.(3.10), we can derive the following inequality for the global training sample set of size nm:

$$\begin{aligned} &R(Q_{Fed}) \leq \hat{R}(Q_{Fed}) \\ &+ \sqrt{\frac{-\log(\det(\Sigma_{Fed})) + \frac{T\eta}{2k_{Fed}m}\text{tr}(\bar{C}\bar{A}^{-1}) - d + 2\log(\frac{1}{\delta}) + 2\log(nm) + 4}{4nm - 2}}. \end{aligned} \tag{8.15}$$

which has completed the proof. □

By combining the results of Theorem 1 with Assumption 3, we can complete the following proof of Theorem 2.

Proof. Based on Assumption 3, we can reformulate Eq.(3.14) in Lemma 3 to

$$\begin{aligned} T\bar{A}\Sigma_{Fed} + \Sigma_{Fed}T\bar{A} &= \frac{T^2\eta}{k_{Fed}m}\bar{C} \\ 2T\Sigma_{Fed}\bar{A} &= \frac{T^2\eta}{k_{Fed}m}\bar{C} \\ \Sigma_{Fed} &= \frac{T\eta}{2k_{Fed}m}\bar{C}\bar{A}^{-1}. \end{aligned} \tag{8.16}$$

By substituting Eq.(8.16) into Eq.(3.15) and rearranging the equation, we have

$$\begin{aligned} & R(Q_{Fed}) - \hat{R}(Q_{Fed}) \\ & \leq \sqrt{\frac{-\log(\det(\frac{T\eta}{2k_{Fed}m}\bar{C}\bar{A}^{-1})) + \frac{T\eta}{2k_{Fed}m}\mathrm{tr}(\bar{C}\bar{A}^{-1}) - d + 2\log(\frac{1}{\delta}) + 2\log(nm) + 4}{4nm-2}} \\ & \leq \sqrt{\frac{-\log((\frac{T\eta}{2k_{Fed}m})^d \det(\bar{C}\bar{A}^{-1})) + \frac{T\eta}{2k_{Fed}m}\mathrm{tr}(\bar{C}\bar{A}^{-1}) - d + 2\log(\frac{1}{\delta}) + 2\log(nm) + 4}{4nm-2}} \\ & \leq \sqrt{\frac{d\log(\frac{2k_{Fed}m}{T\eta}) - \log(\det(\bar{C}\bar{A}^{-1})) + \frac{T\eta}{2k_{Fed}m}\mathrm{tr}(\bar{C}\bar{A}^{-1}) - d + 2\log(\frac{1}{\delta}) + 2\log(nm) + 4}{4nm-2}}. \end{aligned} \tag{8.17}$$

which completes the proof. □

8.1.2 Proof of First Insight: The Relationship Between Two Optimal Model Sizes

This section provides the proof details for Lemmas 4, 5, 6, 7 and Theorem 3 of the theoretical analysis in Section 3.4.3.

According to the proof of Lemma 2, we can also prove the analytic solution of centralised SGD shown in Lemma 4 as follows.

Proof. Based on the result of the Ornstein-Uhlenbeck process [139], we can simply derive the following analytic solution for the baseline centralised SGD:

$$\theta_{Cen}(T) = \theta(0)e^{-\frac{T}{n}At} + \frac{T}{n}\sqrt{\frac{\eta}{k_{Cen}nm}}\int_0^t e^{-\frac{T}{n}A(t-t')}BdW(t')). \tag{8.18}$$

thus completing the proof. □

The proof of Lemma 5 can also be completed in a similar way.

Proof. Based on Eq.(3.19), we know that

$$\Sigma_{Cen} = \mathbb{E}_{\theta\sim Q}[\theta_{Cen}\theta_{Cen}^{\intercal}]. \tag{8.19}$$

Then, by combining Eq.(3.18) and Eq.(8.19), we can derive the following equation:

$$\begin{aligned}
\frac{T}{n}A\Sigma_{Cen} + \Sigma_{Cen}\frac{T}{n}A &= \frac{T^2\eta}{k_{Cen}n^3m}\int_{-\infty}^{t}\frac{T}{n}Ae^{-\frac{T}{n}A(t-t')}Ce^{-\frac{T}{n}A(t-t')}dt' \\
&\quad + \frac{T^2\eta}{k_{Cen}n^3m}\int_{-\infty}^{t}e^{-\frac{T}{n}A(t-t')}Ce^{-\frac{T}{n}A(t-t')}dt'\frac{T}{n}A \\
&= \frac{T^2\eta}{k_{Cen}n^3m}\int_{-\infty}^{t}\frac{d}{dt'}(e^{-\frac{T}{n}A(t-t')}Ce^{-\frac{T}{n}A(t-t')}) \\
&= \frac{T^2\eta}{k_{Cen}n^3m}C.
\end{aligned} \tag{8.20}$$

which completes the proof. □

By combining Lemma 4 and Lemma 5, we further prove the generalisation bound of centralised SGD shown in Lemma 6.

Proof. Since we have proved from Lemma 5 that $\frac{T}{n}A\Sigma_{Cen} + \Sigma_{Cen}\frac{T}{n}A = \frac{T^2\eta}{k_{Cen}n^3m}C$, we have

$$\begin{aligned}
A\Sigma_{Cen} + \Sigma_{Cen}A &= \frac{T\eta}{k_{Cen}n^2m}C \\
A\Sigma_{Cen}A^{-1} + \Sigma_{Cen} &= \frac{T\eta}{k_{Cen}n^2m}CA^{-1} \\
\mathrm{tr}(A\Sigma_{Cen}A^{-1} + \Sigma_{Cen}) &= \mathrm{tr}(\frac{T\eta}{k_{Cen}n^2m}CA^{-1}) \\
2\mathrm{tr}(\Sigma_{Cen}) &= \mathrm{tr}(\frac{T\eta}{k_{Cen}n^2m}CA^{-1}) \\
\mathrm{tr}(\Sigma_{Cen}) &= \frac{T\eta}{2k_{Cen}n^2m}\mathrm{tr}(CA^{-1}).
\end{aligned} \tag{8.21}$$

Similarly to the proof of Theorem 1, by substituting Eq.(8.21) into Eq.(8.10), we can compute the KL divergence between the distribution of the output hypothesis

and the prior as below:

$$\begin{aligned} D(Q_{Cen}||P) &= -\frac{1}{2}\log(\det(\Sigma_{Cen})) + \frac{1}{2}\mathrm{tr}(\Sigma_{Cen}) - \frac{1}{2}\mathrm{tr}(I) \\ &= -\frac{1}{2}\log(\det(\Sigma_{Cen})) + \frac{T\eta}{4k_{Cen}n^2m}\mathrm{tr}(\bar{C}\bar{A}^{-1}) - \frac{1}{2}d. \end{aligned} \tag{8.22}$$

According to Lemma 1, we can derive the following inequality to bound the generalisation bound error of the baseline centralised SGD:

$$\begin{aligned} &R(Q_{Cen}) \leq \hat{R}(Q_{Cen}) + \\ &\sqrt{\frac{-\log(\det(\Sigma_{Cen})) + \frac{T\eta}{2k_{Cen}n^2m}\mathrm{tr}(CA^{-1}) - d + 2\log(\frac{1}{\delta}) + 2\log(nm) + 4}{4nm-2}}. \end{aligned} \tag{8.23}$$

Since we have assumed that $A\Sigma = \Sigma A$ from Assumption 3, we can reformulate Eq.(3.20) to

$$\begin{aligned} \frac{T}{n}A\Sigma_{Cen} + \Sigma_{Cen}\frac{T}{n}A &= \frac{T^2\eta}{k_{Cen}n^3m}C \\ 2\Sigma_{Cen}A &= \frac{T\eta}{k_{Cen}n^2m}C \\ \Sigma_{Cen} &= \frac{T\eta}{2k_{Cen}n^2m}CA^{-1}. \end{aligned} \tag{8.24}$$

By inserting Eq.(8.24) into Eq.(8.23) and rearranging the equation, we have

$$\begin{aligned} &R(Q_{Cen}) - \hat{R}(Q_{Cen}) \\ &\leq \sqrt{\frac{-\log(\det(\frac{T\eta}{2k_{Cen}n^2m}CA^{-1})) + \frac{T\eta}{2k_{Cen}n^2m}\mathrm{tr}(CA^{-1}) - d + 2\log(\frac{1}{\delta}) + 2\log(nm) + 4}{4nm-2}} \\ &\leq \sqrt{\frac{d\log(\frac{2k_{Cen}n^2m}{T\eta}) - \log(\det(CA^{-1})) + \frac{T\eta}{2k_{Cen}n^2m}\mathrm{tr}(CA^{-1}) - d + 2\log(\frac{1}{\delta}) + 2\log(nm) + 4}{4nm-2}}. \end{aligned} \tag{8.25}$$

The proof has been completed. □

Now, we have the PAC-Bayesian generalisation bounds for both federated and centralised SGD. We can start to prove their respective optimal model sizes shown in Lemma 7.

Proof. At the beginning, we define

$$L_{Fed} = \frac{d\log(\frac{2k_{Fed}m}{T\eta}) - \log(\det(\bar{C}\bar{A}^{-1})) + \frac{T\eta}{2k_{Fed}m}\mathrm{tr}(\bar{C}\bar{A}^{-1})) - d + 2\log(\frac{1}{\delta}) + 2\log(nm) + 4}{4nm - 2},$$

$$L_{Cen} = \frac{d\log(\frac{2k_{Cen}n^2m}{T\eta}) - \log(\det(CA^{-1})) + \frac{T\eta}{2k_{Cen}n^2m}\mathrm{tr}(CA^{-1}) - d + 2\log(\frac{1}{\delta}) + 2\log(nm) + 4}{4nm - 2}. \tag{8.26}$$

Then, Eq.(3.17) and Eq.(3.21) can be turned into

$$\begin{aligned} R(Q_{Fed}) &\leq \hat{R}(Q_{Fed}) + \sqrt{L_{Fed}}, \\ R(Q_{Cen}) &\leq \hat{R}(Q_{Cen}) + \sqrt{L_{Cen}}. \end{aligned} \tag{8.27}$$

To find the optimal model size that minimises the generalisation bound, we start by calculating the derivative of L_{Fed} with respect to the average amount of training data m on clients as follows:

$$\frac{\partial L_{Fed}}{\partial m} = \frac{G_1 - G_2}{(4nm-2)^2}. \tag{8.28}$$

where

$$\begin{aligned} G_1 &= (4nm-2)(\frac{d+2}{m} - \frac{T\eta}{2k_{Fed}m^2}\mathrm{tr}(\bar{C}\bar{A}^{-1})), \\ G_2 &= (4n)(d\log(\frac{2k_{Fed}m}{T\eta}) - \log(\det(\bar{C}\bar{A}^{-1})) + \frac{T\eta}{2k_{Fed}m}\mathrm{tr}(\bar{C}\bar{A}^{-1}) - d \\ &\quad + 2\log(\frac{1}{\sigma}) + 2\log(nm) + 4). \end{aligned} \tag{8.29}$$

By setting $\frac{\partial L_{Fed}}{\partial m} = 0$, we derive the following optimal model size:

$$
\begin{aligned}
&(4nm-2)\frac{d}{m} + (4nm-2)(\frac{2}{m} - \frac{T\eta}{2k_{Fed}m^2}\mathrm{tr}(\bar{C}\bar{A}^{-1})) \\
&= 4n(d\log(\frac{2k_{Fed}m}{T\eta}) - \log(\det(\bar{C}\bar{A}^{-1})) + \frac{T\eta}{2k_{Fed}m}\mathrm{tr}(\bar{C}\bar{A}^{-1}) - d \\
&\quad + 2\log(\frac{1}{\sigma}) + 2\log(nm) + 4)(4n - \frac{2}{m} - 4n\log(\frac{2k_{Fed}m}{T\eta}) + 4n)d \\
&= 4n(-\log(\det(\bar{C}\bar{A}^{-1})) + \frac{T\eta}{2k_{Fed}m}\mathrm{tr}(\bar{C}\bar{A}^{-1}) + 2\log(\frac{1}{\sigma}) + 2\log(nm) + 4) \\
&\quad - (4nm-2)(\frac{2}{m} - \frac{T\eta}{2k_{Fed}m^2}\mathrm{tr}(\bar{C}\bar{A}^{-1}))
\end{aligned}
$$

$$
d^*_{Fed} = \frac{-4n\log((\det(\bar{C}\bar{A}^{-1})) + (\frac{4nT\eta}{k_{Fed}m} - \frac{T\eta}{k_{Fed}m^2})\mathrm{tr}(\bar{C}\bar{A}^{-1})) + 8n\log(\frac{1}{\delta}) + 8n\log(nm) - \frac{4}{m} + 8n}{8n - \frac{2}{m} - 4n\log(\frac{2k_{Fed}m}{T\eta})}. \tag{8.30}
$$

In the same way, we obtain the below optimal model size d^*_{Cen} for the baseline centralised SGD:

$$
\begin{aligned}
d^*_{Cen} &= \\
&\frac{-4n\log\left(\det\left(CA^{-1}\right)\right) + \left(\frac{4T\eta}{k_{Cen}nm} - \frac{T\eta}{k_{Cen}n^2m^2}\right)\mathrm{tr}\left(CA^{-1}\right) + 8n\log\left(\frac{1}{\delta}\right)}{\left(8n - \frac{2}{m} - 4n\log\left(\frac{2k_{Cen}n^2m}{T\eta}\right)\right)} \\
&+ \frac{8n\log\left(nm\right) - \frac{4}{m} + 8n}{\left(8n - \frac{2}{m} - 4n\log\left(\frac{2k_{Cen}n^2m}{T\eta}\right)\right)}
\end{aligned} \tag{8.31}
$$

which completes the proof. □

Based on the Assumption 4 and results in Lemma 7, we can complete the proof of Theorem 3 below and derive the size relationship.

Proof. When Assumption 4 holds, we have:

$$
\begin{aligned}
\bar{C}\bar{A}^{-1} &= \frac{1}{n^\gamma}(C + \Delta_C)(A + \Delta_A)^{-1} \\
&\approx \frac{1}{n^\gamma}(C + \Delta_C)(A^{-1} + A^{-1}\Delta_A A^{-1}).
\end{aligned} \tag{8.32}
$$

Hence,

$$\begin{aligned}\operatorname{tr}(\bar{C}\bar{A}^{-1}) &= \operatorname{tr}(\frac{1}{n^{\gamma}}(C+\Delta_C)(A+\Delta_A)^{-1}) \\ &\approx \operatorname{tr}(\frac{1}{n^{\gamma}}(CA^{-1}+\underbrace{CA^{-1}\Delta_A A^{-1}+\Delta_C\left(A^{-1}+A^{-1}\Delta_A A^{-1}\right)}_{\Delta_1})) \\ &= \frac{1}{n^{\gamma}}(\operatorname{tr}(CA^{-1})+\operatorname{tr}(\Delta_1));\end{aligned} \tag{8.33}$$

$$\begin{aligned}-\log(\det(\bar{C}\bar{A}^{-1})) &= -\log(\det(\frac{1}{n^{\gamma}}(C+\Delta_C)(A+\Delta_A)^{-1})) \\ &\approx -\log(\det(\frac{1}{n^{\gamma}}CA^{-1}\underbrace{(I+C^{-1}\Delta_C)(I+\Delta_A A^{-1})}_{\Delta_2})) \\ &= -\log(\frac{1}{n^{\gamma d}}\det(CA^{-1}\Delta_2)) \\ &= -\log(\frac{1}{n^{\gamma d}}\det(CA^{-1})\det(\Delta_2)) \\ &= \gamma d\log(n)-\log(\det(CA^{-1}))+\log(\det(\Delta_2)^{-1}).\end{aligned} \tag{8.34}$$

Substituting Eqs.(8.33) and (8.34) into Eq.(3.22) derives:

$$\begin{aligned}&d^{*}_{Fed} = \\ &\frac{-4n\left(\log\left(\det\left(CA^{-1}\right)\right)+\log\left(\det\left(\Delta_2\right)\right)\right)+\left(\frac{4nT\eta}{n^{\gamma}k_{Fed}m}-\frac{T\eta}{n^{\gamma}k_{Fed}m^2}\right)\left(\operatorname{tr}\left(CA^{-1}\right)+\operatorname{tr}\left(\Delta_1\right)\right)}{8n-\frac{2}{m}-4n\log\left(\frac{2n^{\gamma}k_{Fed}m}{T\eta}\right)} \\ &+\frac{8n\log\left(\frac{1}{\delta}\right)+8n\log\left(nm\right)-\frac{4}{m}+8n}{8n-\frac{2}{m}-4n\log\left(\frac{2n^{\gamma}k_{Fed}m}{T\eta}\right)}.\end{aligned} \tag{8.35}$$

When $T \to \infty$, by comparing Eq.(8.35) and Eq.(3.23), we have

$$\begin{aligned}
\lim_{T\to\infty} \frac{d^*_{Fed}}{d^*_{Cen}} &= \frac{\left(\frac{4nT\eta}{n^\gamma k_{Fed} m} - \frac{T\eta}{n^\gamma k_{Fed} m^2}\right)\left(\mathrm{tr}\left(CA^{-1}\right) + \mathrm{tr}\left(\Delta_1\right)\right)}{-4n\log\left(\frac{2n^\gamma k_{Fed} m}{T\eta}\right)} \\
&\quad \times \frac{-4n\log\left(\frac{2k_{Cen}n^2 m}{T\eta}\right)}{\left(\frac{4T\eta}{k_{Cen}nm} - \frac{T\eta}{k_{Cen}n^2m^2}\right)\mathrm{tr}\left(CA^{-1}\right)} \\
&= \frac{\left(\frac{4n\eta}{n^\gamma k_{Fed} m} - \frac{\eta}{n^\gamma k_{Fed} m^2}\right)\left(\mathrm{tr}\left(CA^{-1}\right) + \mathrm{tr}\left(\Delta_1\right)\right)}{\left(\frac{4\eta}{k_{Cen}nm} - \frac{\eta}{k_{Cen}n^2m^2}\right)\mathrm{tr}\left(CA^{-1}\right)} \\
&= \frac{\left(\frac{4n\eta}{n^\gamma S_{Fed}} - \frac{\eta}{n^\gamma m S_{Fed}}\right)\left(\mathrm{tr}\left(CA^{-1}\right) + \mathrm{tr}\left(\Delta_1\right)\right)}{\left(\frac{4\eta}{S_{Cen}} - \frac{\eta}{nmS_{Cen}}\right)\mathrm{tr}\left(CA^{-1}\right)} \\
&= \frac{\left(\frac{(4nm-1)\eta}{n^\gamma m S_{Fed}}\right)\left(\mathrm{tr}\left(CA^{-1}\right) + \mathrm{tr}\left(\Delta_1\right)\right)}{\left(\frac{(4nm-1)\eta}{nmS_{Cen}}\right)\mathrm{tr}\left(CA^{-1}\right)} \\
&= \frac{S_{Cen}\left(\mathrm{tr}\left(CA^{-1}\right) + \mathrm{tr}\left(\Delta_1\right)\right)}{S_{Fed}\mathrm{tr}\left(CA^{-1}\right)} \frac{1}{n^{\gamma-1}}.
\end{aligned} \tag{8.36}$$

Let $\rho = \frac{S_{Cen}\left(\mathrm{tr}\left(CA^{-1}\right)+\mathrm{tr}(\Delta_1)\right)}{S_{Fed}\mathrm{tr}(CA^{-1})}$. Since C, $\bar{C}$, A and $\bar{A}$ are (semi) positive-definite matrices, we have $\mathrm{tr}(CA^{-1}) > 0$ and $\mathrm{tr}(\bar{C}\bar{A}^{-1}) > 0$, which further implies:

$$\mathrm{tr}\left(CA^{-1}\right) + \mathrm{tr}\left(\Delta_1\right) \approx n^\gamma \mathrm{tr}(\bar{C}\bar{A}^{-1}) > 0 \tag{8.37}$$

. Therefore, we find $\rho > 0$. The proof has been completed. □

8.1.3 Proof of Second Insight: Evidence for Inferior Generalisation of Distributed Training

This section provides the proof details for Theorem 4 of the theoretical analysis in Section 3.4.4.

Proof. When $T \to \infty$, we can observe that the optimal model sizes shown by Eq.(3.22) and Eq.(3.23) will turn to the following equations:

$$\begin{aligned}\lim_{T\to\infty} d^*_{Fed} &= \frac{\left(\frac{4nT\eta}{k_{Fed}m} - \frac{T\eta}{k_{Fed}m^2}\right)\mathrm{tr}\left(\bar{C}\bar{A}^{-1}\right)}{-4n\log\left(\frac{2k_{Fed}m}{T\eta}\right)} \\ &= \frac{\left(\frac{4nT\eta}{k_{Fed}m} - \frac{T\eta}{k_{Fed}m^2}\right)\mathrm{tr}\left(\bar{C}\bar{A}^{-1}\right)}{4n} \\ &= \frac{\left(\frac{(4nm-1)T\eta}{k_{Fed}m^2}\right)\mathrm{tr}\left(\bar{C}\bar{A}^{-1}\right)}{4n},\end{aligned} \tag{8.38}$$

$$\begin{aligned}\lim_{T\to\infty} d^*_{Cen} &= \frac{\left(\frac{4T\eta}{k_{Cen}nm} - \frac{T\eta}{k_{Cen}n^2m^2}\right)\mathrm{tr}\left(CA^{-1}\right)}{-4n\log\left(\frac{2k_{Cen}n^2m}{T\eta}\right)} \\ &= \frac{\left(\frac{4T\eta}{k_{Cen}nm} - \frac{T\eta}{k_{Cen}n^2m^2}\right)\mathrm{tr}\left(CA^{-1}\right)}{4n}.\end{aligned} \tag{8.39}$$

Therefore, the optimal generalisation bound for federated learning based on the optimal model size can be formulated as:

$$\begin{aligned}\lim_{T\to\infty}\left[R\left(Q_{Fed}\right) - \hat{R}\left(Q_{Fed}\right)\right] &\leq \sqrt{L_{Fed}} \\ &\leq \mathcal{G}^*_{Fed}\end{aligned} \tag{8.40}$$

where

$$\begin{aligned}\mathcal{G}^*_{Fed} &= \frac{-\log(\det(\Sigma_{Fed})) + \frac{T\eta}{2k_{Fed}m}\mathrm{tr}\left(\bar{C}\bar{A}^{-1}\right) - d^*_{Fed} + 2\log\left(\frac{1}{\delta}\right) + 2\log\left(nm\right) + 4}{4nm-2} \\ &= \frac{d^*_{Fed}\log\left(\frac{2k_{Fed}m}{T\eta}\right) - \log\left(\det\left(\bar{C}\bar{A}^{-1}\right)\right) + \frac{T\eta}{2k_{Fed}m}\left(\mathrm{tr}\left(\bar{C}\bar{A}^{-1}\right)\right) - d^*_{Fed}}{4nm-2} \\ &\quad + \frac{2\log\left(\frac{1}{\delta}\right) + 2\log\left(nm\right) + 4}{4nm-2}.\end{aligned} \tag{8.41}$$

Similarly, the optimal generalisation bound for centralised learning based on the optimal model size is defined as:

$$\begin{aligned}\mathcal{G}^*_{Cen} &= \frac{-\log(\det(\Sigma_{Cen})) + \frac{T\eta}{2k_{Cen}n^2m}\mathrm{tr}\left(CA^{-1}\right) - d^*_{Cen} + 2\log\left(\frac{1}{\delta}\right) + 2\log\left(nm\right) + 4}{4nm-2} \\ &= \frac{d^*_{Cen}\log\left(\frac{2k_{Cen}n^2m}{T\eta}\right) - \log\left(\det\left(CA^{-1}\right)\right) + \frac{T\eta}{2k_{Cen}n^2m}\mathrm{tr}\left(CA^{-1}\right) - d^*_{Cen} + 2\log\left(\frac{1}{\delta}\right)}{4nm-2} \\ &\quad + \frac{2\log\left(nm\right)+4}{4nm-2}.\end{aligned} \tag{8.42}$$

Then, we have

$$\begin{aligned}\mathcal{G}^*_{Fed} - \mathcal{G}^*_{Cen} &= \frac{-\log(\det(\Sigma_{Fed})) + \frac{T\eta}{2k_{Fed}m}\mathrm{tr}\left(\bar{C}\bar{A}^{-1}\right) - d^*_{Fed}}{4nm-2} \\ &\qquad - \frac{-\log(\det(\Sigma_{Cen})) + \frac{T\eta}{2k_{Cen}n^2m}\mathrm{tr}\left(CA^{-1}\right) - d^*_{Cen}}{4nm-2} \\ &= \frac{d^*_{Fed}\log\left(\frac{2k_{Fed}m}{T\eta}\right) - \log\left(\det\left(\bar{C}\bar{A}^{-1}\right)\right) + \frac{T\eta}{2k_{Fed}m}(\mathrm{tr}\left(\bar{C}\bar{A}^{-1}\right)) - d^*_{Fed}}{4nm-2} \\ &\quad - \frac{d^*_{Cen}\log\left(\frac{2k_{Cen}n^2m}{T\eta}\right) - \log\left(\det\left(CA^{-1}\right)\right) + \frac{T\eta}{2k_{Cen}n^2m}\mathrm{tr}\left(CA^{-1}\right) - d^*_{Cen}}{4nm-2} \\ &= \frac{d^*_{Fed}\log\left(\frac{2k_{Fed}m}{T\eta e}\right) - d^*_{Cen}\log\left(\frac{2k_{Cen}n^2m}{T\eta e}\right) + \log\left(\frac{\det\left(CA^{-1}\right)}{\det\left(\bar{C}\bar{A}^{-1}\right)}\right)}{4nm-2} \\ &\quad + \frac{T\eta\left(\frac{\mathrm{tr}\left(\bar{C}\bar{A}^{-1}\right)}{2k_{Fed}m} - \frac{\mathrm{tr}\left(CA^{-1}\right)}{2k_{Cen}n^2m}\right)}{4nm-2}.\end{aligned} \tag{8.43}$$

By inserting Eq.(8.38) and Eq.(8.39), Eq.(8.43) can be simplified into

$$
\begin{aligned}
&\lim_{T\to\infty}\left(\mathcal{G}_{Fed}^{*}-\mathcal{G}_{Cen}^{*}\right)\\
&=\frac{\frac{\left(\frac{(4nm-1)T\eta}{k_{Fed}m^2}\right)\mathrm{tr}\left(\bar{C}\bar{A}^{-1}\right)}{4n}\log\left(\frac{2k_{Fed}m}{T\eta e}\right)-\frac{\left(\frac{(4nm-1)T\eta}{k_{Cen}n^2m^2}\right)\mathrm{tr}\left(CA^{-1}\right)}{4n}\log\left(\frac{2k_{Cen}n^2m}{T\eta e}\right)}{4nm-2}\\
&\quad+\frac{T\eta\left(\frac{\mathrm{tr}\left(CA^{-1}\right)}{2k_{Fed}m}-\frac{\mathrm{tr}\left(\bar{C}\bar{A}^{-1}\right)}{2k_{Cen}n^2m}\right)}{4nm-2}\\
&=\frac{\left(\frac{T\eta(4nm-1)\log\left(\frac{2k_{Fed}m}{T\eta e}\right)}{4nk_{Fed}m^2}\right)\mathrm{tr}\left(\bar{C}\bar{A}^{-1}\right)-\left(\frac{T\eta(4nm-1)\log\left(\frac{2k_{Cen}n^2m}{T\eta e}\right)}{4k_{Cen}n^3m^2}\right)\mathrm{tr}\left(CA^{-1}\right)}{4nm-2}\\
&=\frac{\left(\frac{T\eta(4nm-1)\log(T)}{4k_{Cen}n^3m^2}\right)\mathrm{tr}\left(CA^{-1}\right)-\left(\frac{T\eta(4nm-1)\log(T)}{4nk_{Fed}m^2}\right)\mathrm{tr}\left(\bar{C}\bar{A}^{-1}\right)}{4nm-2}\\
&=\frac{\left(\frac{T\eta(4nm-1)\log(T)}{4S_{Cen}n^2m}\right)\mathrm{tr}\left(CA^{-1}\right)-\left(\frac{T\eta(4nm-1)\log(T)}{4S_{Fed}nm}\right)\mathrm{tr}\left(\bar{C}\bar{A}^{-1}\right)}{4nm-2}\\
&=\frac{T\eta\left(4nm-1\right)\log\left(T\right)\left(\frac{\mathrm{tr}\left(CA^{-1}\right)}{4S_{Cen}n^2m}-\frac{\mathrm{tr}\left(\bar{C}\bar{A}^{-1}\right)}{4S_{Fed}nm}\right)}{4nm-2}\\
&=T\eta\left(4nm-1\right)\log\left(T\right)\left(\frac{\mathrm{tr}\left(CA^{-1}\right)}{4S_{Cen}n^2m}-\frac{\left(\mathrm{tr}\left(CA^{-1}\right)+\mathrm{tr}\left(\Delta_1\right)\right)}{4S_{Fed}n^{\gamma+1}m}\right),
\end{aligned}
\tag{8.44}
$$

To satisfy $\lim_{T\to\infty}(\mathcal{G}_{Fed}^{*}-\mathcal{G}_{Cen}^{*})>0$, we solve

$$
\begin{aligned}
&T\eta\left(4nm-1\right)\log\left(T\right)\left(\frac{\mathrm{tr}\left(CA^{-1}\right)}{4S_{Cen}n^2m}-\frac{\left(\mathrm{tr}\left(CA^{-1}\right)+\mathrm{tr}\left(\Delta_1\right)\right)}{4S_{Fed}n^{\gamma+1}m}\right)>0\\
&\frac{\mathrm{tr}\left(CA^{-1}\right)}{4S_{Cen}n^2m}>\frac{\left(\mathrm{tr}\left(CA^{-1}\right)+\mathrm{tr}\left(\Delta_1\right)\right)}{4S_{Fed}n^{\gamma+1}m}\\
&n>\sqrt[\gamma-1]{\frac{S_{Cen}\left(\mathrm{tr}\left(CA^{-1}\right)+\mathrm{tr}\left(\Delta_1\right)\right)}{S_{Fed}\mathrm{tr}\left(CA^{-1}\right)}}.
\end{aligned}
\tag{8.45}
$$

Considering that we have defined $\rho=\frac{S_{Cen}\left(\mathrm{tr}\left(CA^{-1}\right)+\mathrm{tr}(\Delta_1)\right)}{S_{Fed}\mathrm{tr}(CA^{-1})}$ in Theorem 3. The above condition turns to $n>\sqrt[\gamma-1]{\rho}$, which completes the proof. □

8.1.4 Proof of Third Insight: Estimating Optimal Model Size by Average Training Compute Between Clients

This section provides the proof details for Lemmas 8, 9 10 and Theorem 5 of the theoretical analysis in Section 3.4.5.

To formulate the analytic solution of optimal model size at the client level, we first derive the analytic solution of SGD with data from a single client. The proof of Lemma 8 is shown below.

Proof. Based on Eq.(3.18), we can simply derive the following analytic solution for this baseline training that only uses the local data on client i and is also iterated for T rounds:

$$\acute{\theta}_i(T) = \theta_i(0)e^{-TA_i t} + T\sqrt{\frac{\eta}{k_i m}} \int_0^t e^{-TA_i(t-t')} B_i dW(t')). \tag{8.46}$$

which completes the proof. □

Starting from the above result, we can use a similar proof as that used to derive the generalisation bounds for federated and centralised SGD to obtain Lemma 9.

Proof. Similarly to the proof of Lemma 6, we first derive

$$\Sigma_i = \frac{T\eta}{2k_i m} C_i A_i^{-1}. \tag{8.47}$$

Then, we reformulate the Eq.(8.23) in terms of Eq.(8.47) and the size m of local data on client i to obtain the following generalisation bound:

$$R(Q_i) - \hat{R}(Q_i)$$
$$\leq \sqrt{\frac{d_i \log(\frac{2k_i m}{T\eta}) - \log(\det(C_i A_i^{-1})) + \frac{T\eta}{2k_i m}\mathrm{tr}(C_i A_i^{-1}) - d_i + 2\log(\frac{1}{\delta}) + 2\log(m) + 4}{4m - 2}}. \tag{8.48}$$

which completes the proof. □

By the bound result in Lemma 8, we formulate the optimal model size at the client level in Lemma 10. The following proof demonstrates our derivation.

Proof. We firstly define

$$L_i = \frac{d\log\left(\frac{2k_im}{T\eta}\right) - \log\left(\det\left(C_iA_i^{-1}\right)\right) + \frac{T\eta}{2k_im}\mathrm{tr}\left(C_iA_i^{-1}\right) - d + 2\log\left(\frac{1}{\delta}\right) + 2\log\left(m\right) + 4}{4m-2}. \tag{8.49}$$

Then, Eq.(3.29) becomes

$$R(Q_i) \leq \hat{R}(Q_i) + \sqrt{L_i}. \tag{8.50}$$

We calculate the derivative of L_i with respect to the amount of training data m on the client i as follow:

$$\frac{\partial L_i}{\partial m} = \frac{G_1 - G_2}{(4m-2)^2}. \tag{8.51}$$

where

$$\begin{aligned} G_1 &= (4m-2)\left(\frac{d+2}{m} - \frac{T\eta}{2k_im^2}\mathrm{tr}\left(C_iA_i^{-1}\right)\right), \\ G_2 &= 4(d\log\left(\frac{2k_im}{T\eta}\right) - \log\left(\det\left(C_iA_i^{-1}\right)\right) + \frac{T\eta}{2k_im}\mathrm{tr}\left(C_iA_i^{-1}\right) - d \\ &\quad + 2\log\left(\frac{1}{\delta}\right) + 2\log\left(m\right) + 4). \end{aligned} \tag{8.52}$$

Similarly, to find the optimal size, we set $\frac{\partial L_i}{\partial m} = 0$ and derive

$$\begin{aligned} &(4m-2)\left(\frac{d+2}{m} - \frac{T\eta}{2k_im^2}\mathrm{tr}\left(C_iA_i^{-1}\right)\right) = 4(d\log\left(\frac{2k_im}{T\eta}\right) - \log\left(\det\left(C_iA_i^{-1}\right)\right) \\ &+ \frac{T\eta}{2k_im}\mathrm{tr}\left(C_iA_i^{-1}\right) - d + 2\log\left(\frac{1}{\delta}\right) + 2\log\left(m\right) + 4) \\ &d_i^* = \frac{-4\log\left(\det\left(C_iA_i^{-1}\right)\right) + \left(\frac{4T\eta}{k_im} - \frac{T\eta}{k_im^2}\right)\mathrm{tr}\left(C_iA_i^{-1}\right) + 8\log\left(\frac{1}{\delta}\right) + 8\log\left(m\right) - \frac{4}{m} + 8}{8 - \frac{2}{m} - 4\log\left(\frac{2k_im}{T\eta}\right)}. \end{aligned} \tag{8.53}$$

The proof has been completed. □

Finally, based on the above result and the optimal model sizes in Lemma 7, the proof of Theorem 5 is completed as follows.

Proof. Based on Eqs.(3.22) and (8.53), we derive the below result when $T \to \infty$:

$$\lim_{T\to\infty} \frac{d_{Fed}^*}{\frac{1}{n}\sum_{i=1}^n d_i^*} = \frac{\left(\frac{4nT\eta}{k_{Fed}m} - \frac{T\eta}{k_{Fed}m^2}\right) \mathrm{tr}\left(\bar{C}\bar{A}^{-1}\right)}{-4n\log\left(\frac{2k_{Fed}m}{T\eta}\right)} \times \frac{-4\log\left(\frac{2k_im}{T\eta}\right)}{\left(\frac{4T\eta}{k_im} - \frac{T\eta}{k_im^2}\right)\frac{1}{n}\sum_{i=1}^n \mathrm{tr}\left(C_iA_i^{-1}\right)}. \tag{8.54}$$

Considering that we use $\xi_i^C = C_i - \bar{C}$ and $\xi_i^A = A_i - \bar{A}$ to denote client variance in non-IID settings, we have

$$C_iA_i^{-1} = (\bar{C} + \xi_i^C)(\bar{A} + \xi_i^A)^{-1} \approx (\bar{C} + \xi_i^C)(\bar{A}^{-1} + \bar{A}^{-1}\xi_i^A\bar{A}^{-1}), \tag{8.55}$$

which implies

$$\begin{aligned} \mathrm{tr}(C_iA_i^{-1}) &= \mathrm{tr}((\bar{C} + \xi_i^C)(\bar{A} + \xi_i^A)^{-1}) \\ &\approx \mathrm{tr}(\bar{C}\bar{A}^{-1} + \underbrace{\bar{C}\bar{A}^{-1}\xi_i^A\bar{A}^{-1} + \xi_i^C\left(\bar{A}^{-1} + \bar{A}^{-1}\xi_i^A\bar{A}^{-1}\right)}_{\xi_i}) \\ &= \mathrm{tr}(\bar{C}\bar{A}^{-1}) + \mathrm{tr}(\xi_i); \end{aligned} \tag{8.56}$$

With the above equation and $\{k_{Fed}m = k_im | i \in n\}$, Eq.(8.54) can be further simplified into

$$\begin{aligned} \lim_{T\to\infty} \frac{d_{Fed}^*}{\frac{1}{n}\sum_{i=1}^n d_i^*} &= \frac{\left(\frac{4n\eta}{k_{Fed}m} - \frac{\eta}{k_{Fed}m^2}\right) \mathrm{tr}\left(\bar{C}\bar{A}^{-1}\right)}{n\left(\frac{4\eta}{k_{Fed}m} - \frac{\eta}{k_{Fed}m^2}\right)\frac{1}{n}\sum_{i=1}^n \left(\mathrm{tr}\left(\bar{C}\bar{A}^{-1}\right) + \mathrm{tr}\left(\xi_i\right)\right)} \\ &= \frac{\left(\frac{4n}{k_{Fed}m} - \frac{1}{k_{Fed}m^2}\right) \mathrm{tr}\left(\bar{C}\bar{A}^{-1}\right)}{n\left(\frac{4}{k_{Fed}m} - \frac{1}{k_{Fed}m^2}\right)\left(\mathrm{tr}\left(\bar{C}\bar{A}^{-1}\right) + \mathrm{tr}\left(\bar{\xi}\right)\right)} \\ &= \left(\frac{4m - \frac{1}{n}}{4m - 1}\right) \frac{\mathrm{tr}\left(\bar{C}\bar{A}^{-1}\right)}{\left(\mathrm{tr}\left(\bar{C}\bar{A}^{-1}\right) + \mathrm{tr}\left(\bar{\xi}\right)\right)}, \end{aligned} \tag{8.57}$$

where $\bar{\xi} = \frac{1}{n}\sum_{i=1}^n \xi_i$. Let $\kappa = \left(\frac{4m-\frac{1}{n}}{4m-1}\right) \frac{\mathrm{tr}\left(\bar{C}\bar{A}^{-1}\right)}{\left(\mathrm{tr}\left(\bar{C}\bar{A}^{-1}\right)+\mathrm{tr}\left(\bar{\xi}\right)\right)}$. The proof has been completed. □

8.2 Proofs of Chapter 4

8.2.1 Proof of Stability and Generalisation Bound of Decentralised and Federated Learning

This section provides the proof details for Theorem 6 of the theoretical analysis in Section 4.3.2.

Proof. We couple two executions of decentralised learning on training datasets that differ in exactly one example. The two runs produce parameter sequences $\{\theta^t\}_{t=0}^T$ and $\{\tilde{\theta}^t\}_{t=0}^T$. We define the deviation between them by $\Delta_t := \|\theta^t - \tilde{\theta}^t\|$. At iteration t, let the two coupled runs visit clients i_t and $\tilde{i}_t$, and let the corresponding local samples be z_{i_t} and $z_{\tilde{i}_t}$. We consider the following two cases.

Firstly, if the same sample is selected for both training, we have ***Case A:***

$$\begin{aligned}\Delta_{t+1} &= \left\|\theta^t - \tilde{\theta}^t - \eta\big(\nabla f_{i_t}(\theta^t; z_{i_t}) - \nabla f_{i_t}(\tilde{\theta}^t; z_{i_t})\big)\right\| \\ &\leq \left\|\theta^t - \tilde{\theta}^t\right\| + \eta\left\|\nabla f_{i_t}(\theta^t; z_{i_t}) - \nabla f_{i_t}(\tilde{\theta}^t; z_{i_t})\right\| \\ &\leq (1+\eta L)\,\Delta_t.\end{aligned} \tag{8.58}$$

Otherwise, we have ***Case B:***

$$\begin{aligned}\Delta_{t+1} &= \left\|\theta^t - \tilde{\theta}^t - \eta\big(\nabla f_{i_t}(\theta^t; z_{i_t}) - \nabla f_{\tilde{i}_t}(\tilde{\theta}^t; z_{\tilde{i}_t})\big)\right\| \\ &\leq \left\|\theta^t - \tilde{\theta}^t\right\| + \eta\left\|\nabla f_{i_t}(w^t; z_{i_t}) - \nabla f_{\tilde{i}_t}(\tilde{\theta}^t; z_{i_t})\right\| + \eta\left\|\nabla f_{i_t}(\tilde{\theta}^t; z_{i_t}) - \nabla f_{\tilde{i}_t}(\tilde{\theta}^t; z_{\tilde{i}_t})\right\| \\ &\leq (1+\eta L)\,\Delta_t + \eta\,\beta_t,\end{aligned} \tag{8.59}$$

where we define $\beta_t = \left\|\nabla f_{i_t}(\tilde{\theta}^t; z_{i_t}) - \nabla f_{\tilde{i}_t}(\tilde{\theta}^t; z_{\tilde{i}_t})\right\|$. Combining the two cases yields

$$\Delta_{t+1} \leq (1+\eta L)\Delta_t + \eta\,\beta_t. \tag{8.60}$$

Extending this into the recursion of T steps further produces

$$\Delta_T \leq \sum_{t=0}^{T-1}\left(\prod_{s=t+1}^{T-1}(1+\eta_s L)\right)\eta_t\beta_t. \tag{8.61}$$

According to the bounded gradient assumption $\|\nabla f_i(\theta; z)\| \leq \mathcal{B}$, the following statement holds depending on the circumstance if the same client i is selected for both training:

$$\delta_t \leq \begin{cases} 2\mathcal{B}, & i_t \neq \tilde{i}_t, \\ 2\mathcal{B}, & i_t = \tilde{i}_t = i, \\ 0, & \text{otherwise.} \end{cases} \tag{8.62}$$

Then, by taking expectations in (8.61) and using linearity, we establish

$$\begin{aligned} E[\Delta_T] &\leq E[\sum_{t=0}^{T-1} \left(\prod_{s=t+1}^{T-1} (1+\eta_s L) \right) \eta_t \beta_t] \\ &= \sum_{t=0}^{T-1} \left(\prod_{s=t+1}^{T-1} (1+\eta_s L) \right) \eta_t E[\beta_t] \\ &\leq \sum_{t=0}^{T-1} \left(\prod_{s=t+1}^{T-1} (1+\eta_s L) \right) 2\eta_t \mathcal{B}(Pr(i_t \neq \tilde{i}_t) + Pr(i_t = \tilde{i}_t = i)). \end{aligned} \tag{8.63}$$

Let μ, ν be the initial distributions of the two coupled communication walks, and let P be the transition kernel of the communication process. Considering that each walk is determined by the selected training client in the last round, it thus follows a Markovian chain [150]. Standard mixing estimates give

$$\begin{aligned} \Pr(i_t \neq \tilde{i}_t) &\leq \|\mu \mathcal{P}^t - \nu \mathcal{P}^t\|_{\text{TV}} \\ &= \frac{1}{2} \|\mu \mathcal{P}^t - \nu \mathcal{P}^t\|_1 \\ &\leq \frac{n}{2} \|\mu \mathcal{P}^t - \nu \mathcal{P}^t\|_\infty \\ &\leq \frac{n}{2} c_{\mathcal{P}} \lambda_2^t, \end{aligned} \tag{8.64}$$

where $||\cdot||_{TV}$ denotes the total variation. For the collision probability $\Pr(i_t = \tilde{i}_t = i)$, using the stationary distribution $\pi(i) = \deg_i/(2E)$ and the upper bound $\sum_i \pi(i)^2 \leq \max_i \pi(i) \leq \deg_{\max}/(2E)$, we have

$$\Pr(i_t = \tilde{i}_t = i) \leq \frac{\deg_{\max}}{2E}. \tag{8.65}$$

Substituting (8.64) and (8.65) into (8.63) gives the single-walk estimate

$$E[\Delta_T] \leq 2\mathcal{B}\sum_{t=0}^{T-1}\left(\prod_{s=t+1}^{T-1}(1+\eta_s L)\right)\eta_t(\frac{n}{2}c_{\mathcal{P}}\lambda_2^t + \frac{\deg_{max}}{2E}). \tag{8.66}$$

By the assumption on Lipschitz continuity of the loss,

$$\begin{aligned}\mathbb{E}\left[|f\left(\theta^T;z\right) - f\left(\tilde{\theta}^T;z\right)|\right] &\leq G\mathbb{E}\left[||\theta^T - \tilde{\theta}^T||\right] \\ &\leq G\mathcal{B}\sum_{t=0}^{T-1}\left(\prod_{s=t+1}^{T-1}(1+\eta_s L)\right)\eta_t(nc_{\mathcal{P}}\lambda_2^t + \frac{\deg_{max}}{E})\end{aligned} \tag{8.67}$$

Next, we attempt to generalise the above result to k parallel communication walks. Considering that each work adopts the same initial learning rate and learning rate decay (i.e., $\forall_j : \eta_s^j \equiv \eta_s, \eta_t^j \equiv \eta_t$), we have

$$\begin{aligned}&\mathbb{E}\left[|f\left(\theta^T;z\right) - f\left(\tilde{\theta}^T;z\right)|\right] \\ &\leq GE\left[||\theta^T - \tilde{\theta}^T||\right] \\ &= GE\left[||\frac{1}{K}\sum_{j=1}^{K}\theta_j^T - \frac{1}{K}\sum_{j=1}^{K}\tilde{\theta}_j^T||\right] \\ &= GE\left[||\frac{1}{K}\sum_{j=1}^{K}(\theta_j^T - \tilde{\theta}_j^T)||\right] \\ &= \frac{G}{K}E\left[||\sum_{j=1}^{K}(\theta_j^T - \tilde{\theta_j^T})||\right] \\ &\leq \frac{G}{K}\sum_{j=1}^{K}\sum_{t=0}^{T-1}\left(\prod_{s=t+1}^{T-1}(1+\eta_s L)\right)2\eta_t\mathcal{B}(Pr(i_t^j \neq \tilde{i}_t^j) + Pr(i_t^j = \tilde{i}_t^j = i)) \\ &\leq \frac{G}{K}\sum_{t=0}^{T-1}\left(\prod_{s=t+1}^{T-1}(1+\eta_s L)\right)2\eta_t\mathcal{B}(Pr(\exists_j : i_t^j \neq \tilde{i}_t^j) + Pr(\exists_j : i_t^j = \tilde{i}_t^j = i)).\end{aligned} \tag{8.68}$$

Based on Eqs.(8.64) and (8.65) and by applying a union bound, we find

$$\Pr\left(\exists j :\ i_t^j \neq \tilde{i}_t^j\right) \leq \sum_{j=1}^{K}\frac{n}{2}c_{\mathcal{P}}\lambda_2^t = K\cdot\frac{n}{2}c_{\mathcal{P}}\lambda_2^t, \tag{8.69}$$

and

$$\Pr\left(\exists j:\ i_t^j = \tilde{i}_t^j = i\right) \leq 1 - \left(1 - \frac{\deg_{\max}}{2E}\right)^K. \tag{8.70}$$

Plugging Eqs.(8.69) and (8.70) into Eq.(8.68) produces

$$\begin{aligned}
&\mathbb{E}\Big[\big|f(\theta^T; z) - f(\tilde{\theta}^T; z)\big|\Big] \\
&\leq \frac{G}{K}\mathcal{B}\sum_{t=0}^{T-1}\left(\prod_{s=t+1}^{T-1}(1+\eta_s L)\right)\eta_t(Knc_{\mathcal{P}}\lambda_2^t + 2(1 - \left(1 - \frac{\deg_{\max}}{2E}\right)^K)).
\end{aligned} \tag{8.71}$$

Let $\eta_t \equiv \eta$, and $\rho = 1 + \eta L$. We can simplify the above equation into

$$\begin{aligned}
&\mathbb{E}\left[\left|f\left(\theta^T; z\right) - f\left(\tilde{\theta}^T; z\right)\right|\right] \\
&\leq G\eta\mathcal{B}\sum_{t=0}^{T-1}(1+\eta L)^{T-1-t}(nc_{\mathcal{P}}\lambda_2^t + \frac{2}{K}(1 - (1 - \frac{\deg_{max}}{2E})^K)).
\end{aligned} \tag{8.72}$$

This further simplifies into the following closed form:

$$\begin{aligned}
&\mathbb{E}\left[\left|f\left(\theta^T; z\right) - f\left(\tilde{\theta}^T; z\right)\right|\right] \\
&\leq \eta G\mathcal{B}\left[\ nc_{\mathcal{P}}\,\rho^{T-1}\,\frac{1-\left(\lambda_2/\rho\right)^T}{1-\left(\lambda_2/\rho\right)}\ +\ \frac{\rho^T-1}{\rho-1}\cdot\frac{2(1-\left(1-\frac{\deg_{\max}}{E}\right)^K)}{K}\ \right],
\end{aligned} \tag{8.73}$$

by the fact that

$$\begin{aligned}
&\sum_{t=0}^{T-1}\rho^{T-1-t}\lambda_2^t = \rho^{T-1}\sum_{t=0}^{T-1}(\frac{\lambda_2}{\rho})^t = \rho^{T-1}\cdot\frac{1-(\lambda_2/\rho)^T}{1-(\lambda_2/\rho)}, \\
&\sum_{t=0}^{T-1}\rho^{T-1-t} = \sum_{s=0}^{T-1}\rho^s = \frac{\rho^T-1}{\rho-1}.
\end{aligned} \tag{8.74}$$

This completes the proof. □

8.2.2 Proof of PAC-Bayesian Generalisation Gap

This section provides the proof details for Theorem 7 of the theoretical analysis in Section 4.3.3.

Proof. Based on Assumption 3, we can re-formulate Eq.(3.14) in Lemma 3 to

$$\begin{aligned} T\bar{A}\Sigma_{Fed} + \Sigma_{Fed}T\bar{A} &= \frac{T^2\eta}{k_{Fed}m}\bar{C} \\ 2T\Sigma_{Fed}\bar{A} &= \frac{T^2\eta}{k_{Fed}m}\bar{C} \\ \Sigma_{Fed} &= \frac{T\eta}{2k_{Fed}m}\bar{C}\bar{A}^{-1}. \end{aligned} \tag{8.75}$$

By substituting Eq.(8.75) into Eq.(4.13), we have

$$\begin{aligned} & R(Q_{Fed}) - \hat{R}(Q_{Fed}) \\ & \leq \sqrt{\frac{-\log(\det(\frac{T\eta}{2k_{Fed}m}\bar{C}\bar{A}^{-1})) + \frac{T\eta}{2k_{Fed}m}\mathrm{tr}(\bar{C}\bar{A}^{-1}) - d + 2\log(\frac{1}{\delta}) + 2\log(nm) + 4}{4nm-2}} \\ & \leq \sqrt{\frac{-\log((\frac{T\eta}{2k_{Fed}m})^d \det(\bar{C}\bar{A}^{-1})) + \frac{T\eta}{2k_{Fed}m}\mathrm{tr}(\bar{C}\bar{A}^{-1}) - d + 2\log(\frac{1}{\delta}) + 2\log(nm) + 4}{4nm-2}} \\ & \leq \sqrt{\frac{d\log(\frac{2k_{Fed}m}{T\eta}) - \log(\det(\bar{C}\bar{A}^{-1})) + \frac{T\eta}{2k_{Fed}m}\mathrm{tr}(\bar{C}\bar{A}^{-1}) - d + 2\log(\frac{1}{\delta}) + 2\log(nm) + 4}{4nm-2}} \\ & \leq \frac{d\log(\frac{2k_{Fed}m}{T\eta}) - \log(\det(\bar{C}\bar{A}^{-1})) + \frac{T\eta}{2k_{Fed}m}\mathrm{tr}(\bar{C}\bar{A}^{-1}) - d + 2\log(\frac{1}{\delta}) + 2\log(nm) + 4}{4nm-2}. \end{aligned} \tag{8.76}$$

Similarly, according to Assumption 3, we can re-formulate Eq.(3.20) to:

$$\begin{aligned} \frac{T}{n}A\Sigma_{Cen} + \Sigma_{Cen}\frac{T}{n}A &= \frac{T^2\eta}{n^2k_{Cen}D}C \\ 2\Sigma_{Cen}A &= \frac{T\eta}{nk_{Cen}D}C \\ \Sigma_{Cen} &= \frac{T\eta}{2nk_{Cen}D}CA^{-1}. \end{aligned} \tag{8.77}$$

By inserting the above equation into Eq.(4.14) and re-arranging the equation, we have

$$
\begin{aligned}
&R(Q_{Cen}) - \hat{R}(Q_{Cen}) \\
&\leq \sqrt{\frac{-\log(\det(\frac{T\eta}{2nk_{Cen}D}CA^{-1})) + \frac{T\eta}{2nk_{Cen}D}\mathrm{tr}(CA^{-1}) - d + 2\log(\frac{1}{\delta}) + 2\log(D) + 4}{4D-2}} \\
&\leq \sqrt{\frac{d\log(\frac{2nk_{Cen}D}{T\eta}) - \log(\det(CA^{-1})) + \frac{T\eta}{2nk_{Cen}D}\mathrm{tr}(CA^{-1}) - d + 2\log(\frac{1}{\delta}) + 2\log(D) + 4}{4D-2}} \\
&\leq \frac{d\log(\frac{2nk_{Cen}D}{T\eta}) - \log(\det(CA^{-1})) + \frac{T\eta}{2nk_{Cen}D}\mathrm{tr}(CA^{-1}) - d + 2\log(\frac{1}{\delta}) + 2\log(D) + 4}{4D-2}
\end{aligned}
\tag{8.78}
$$

For Eqs.(8.76) and (8.78), we define

$$
\begin{aligned}
\mathcal{G}_{Fed} &= \frac{d\log\left(\frac{2k_{Fed}m}{T\eta}\right) - \log\left(\det\left(\bar{C}\bar{A}^{-1}\right)\right) + \frac{T\eta}{2k_{Fed}m}\mathrm{tr}\left(\bar{C}\bar{A}^{-1}\right) - d + 2\log\left(\frac{1}{\delta}\right) + 2\log(nm) + 4}{4nm-2}, \\
\mathcal{G}_{Cen} &= \frac{d\log\left(\frac{2nk_{Cen}D}{T\eta}\right) - \log\left(\det\left(CA^{-1}\right)\right) + \frac{T\eta}{2nk_{Cen}D}\mathrm{tr}\left(CA^{-1}\right) - d + 2\log\left(\frac{1}{\delta}\right) + 2\log(D) + 4}{4D-2}.
\end{aligned}
\tag{8.79}
$$

The difference between $\mathcal{G}_{Fed}$ and $\mathcal{G}_{Cen}$, which is considered as the gap in the generalisation performance, can be derived with the following form:

$$
\begin{aligned}
&\mathcal{G}_{Fed} - \mathcal{G}_{Cen} \\
&= \frac{d\log(\frac{2k_{Fed}m}{T\eta}) - \log(\det(\bar{C}\bar{A}^{-1})) + \frac{T\eta}{2k_{Fed}m}\mathrm{tr}(\bar{C}\bar{A}^{-1}) - d + 2\log(\frac{1}{\delta}) + 2\log(nm) + 4}{4nm-2} \\
&- \frac{d\log(\frac{2nk_{Cen}D}{T\eta}) - \log(\det(CA^{-1})) + \frac{T\eta}{2nk_{Cen}D}\mathrm{tr}(CA^{-1}) - d + 2\log(\frac{1}{\delta}) + 2\log(D) + 4}{4D-2}.
\end{aligned}
\tag{8.80}
$$

When Assumption 4 hold, we have:

$$
\begin{aligned}
\bar{C}\bar{A}^{-1} &= \frac{1}{n^{\gamma}}(C + \Delta_C)(A + \Delta_A)^{-1} \\
&\approx \frac{1}{n^{\gamma}}(C + \Delta_C)(A^{-1} + A^{-1}\Delta_A A^{-1}).
\end{aligned}
\tag{8.81}
$$

Hence,

$$\begin{aligned}\operatorname{tr}(\bar{C}\bar{A}^{-1}) &= \operatorname{tr}(\frac{1}{n^{\gamma}}(C+\Delta_C)(A+\Delta_A)^{-1}) \\ &= \operatorname{tr}(\frac{1}{n^{\gamma}}(CA^{-1}+\underbrace{CA^{-1}\Delta_A A^{-1}+\Delta_C\left(A^{-1}+A^{-1}\Delta_A A^{-1}\right)}_{\Delta_1})) \\ &= \frac{1}{n^{\gamma}}(\operatorname{tr}(CA^{-1})+\operatorname{tr}(\Delta_1));\end{aligned} \tag{8.82}$$

$$\begin{aligned}-\log(\det(\bar{C}\bar{A}^{-1})) &= -\log(\det(\frac{1}{n^{\gamma}}(C+\Delta_C)(A+\Delta_A)^{-1})) \\ &= -\log(\det(\frac{1}{n^{\gamma}}CA^{-1}\underbrace{(I+C^{-1}\Delta_C)(I+\Delta_A A^{-1})}_{\Delta_2})) \\ &= -\log(\frac{1}{n^{\gamma d}}\det(CA^{-1}\Delta_2)) \\ &= -\log(\frac{1}{n^{\gamma d}}\det(CA^{-1})\det(\Delta_2)) \\ &= \gamma d\log(n)-\log(\det(CA^{-1}))+\log(\det(\Delta_2)^{-1}).\end{aligned} \tag{8.83}$$

Substituting the above results into Eq.(8.80) derives:

$$\begin{aligned}&\mathcal{G}_{Fed}-\mathcal{G}_{Cen} \\ &= \frac{d\log(\frac{2k_{Fed}m}{T\eta})-\log(\det(\bar{C}\bar{A}^{-1}))+\frac{T\eta}{2k_{Fed}m}\operatorname{tr}(\bar{C}\bar{A}^{-1})-d+2\log(\frac{1}{\delta})+2\log(nm)+4}{4nm-2} \\ &\quad -\frac{d\log(\frac{2nk_{Cen}D}{T\eta})-\log(\det(CA^{-1}))+\frac{T\eta}{2nk_{Cen}D}\operatorname{tr}(CA^{-1})-d+2\log(\frac{1}{\delta})+2\log(D)+4}{4D-2} \\ &= \frac{d\log(\frac{2n^{\gamma}k_{Fed}m}{T\eta})-\log(\det(CA^{-1}))+\frac{T\eta}{2n^{\gamma}k_{Fed}m}(\operatorname{tr}(CA^{-1})+\operatorname{tr}(\Delta_1))}{4D-2} \\ &\quad +\frac{\log(\det(\Delta_2)^{-1})-d+2\log(\frac{1}{\delta})+2\log(nm)+4}{4D-2} \\ &\quad -\frac{d\log(\frac{2nk_{Cen}D}{T\eta})-\log(\det(CA^{-1}))+\frac{T\eta}{2nk_{Cen}D}\operatorname{tr}(CA^{-1})-d+2\log(\frac{1}{\delta})+2\log(D)+4}{4D-2} \\ &= \frac{d\log(\frac{n^{\gamma-1}k_{Fed}m}{k_{Cen}D})+(\frac{T\eta}{2n^{\gamma}k_{Fed}m}-\frac{T\eta}{2nk_{Cen}D})\operatorname{tr}(CA^{-1})+\frac{T\eta}{2n^{\gamma}k_{Fed}m}\operatorname{tr}(\Delta_1)+\log(\det(\Delta_2)^{-1})}{4D-2}.\end{aligned} \tag{8.84}$$

The proof has been completed.

□

8.2.3 Proof of Non-Vacuous Bounds on Generalisation Gap

This section provides the proof details for Theorem 8 of the theoretical analysis in Section 4.3.4.

Proof. At the beginning, we construct the following helper function:

$$\begin{aligned} f\left(n\right) = d\log(\frac{n^{\gamma-1}k_{Fed}m}{k_{Cen}D}) + (\frac{T\eta}{2n^{\gamma}k_{Fed}m} - \frac{T\eta}{2nk_{Cen}D})\mathrm{tr}(CA^{-1}) \\ + \frac{T\eta}{2n^{\gamma}k_{Fed}m}\mathrm{tr}(\Delta_1) + \log(\det(\Delta_2)^{-1}). \end{aligned} \tag{8.85}$$

Let $S_{Fed} = k_{Fed}m$ and $S_{Cen} = k_{Cen}D$. By the fact that $n \geq 2$, we further derive

$$\begin{aligned} & f\left(n\right) \\ & = d\log(\frac{n^{\gamma-1}S_{Fed}}{S_{Cen}}) + (\frac{T\eta}{2n^{\gamma}S_{Fed}} - \frac{T\eta}{2nS_{Cen}})\mathrm{tr}(CA^{-1}) + \frac{T\eta}{2n^{\gamma}S_{Fed}}\mathrm{tr}(\Delta_1) + \log(\det(\Delta_2)^{-1}) \\ & \leq d\log(\frac{n^{\gamma-1}S_{Fed}}{S_{Cen}}) + (\frac{T\eta}{2n^{\gamma}S_{Fed}} - \frac{T\eta}{2nS_{Cen}})\mathrm{tr}(CA^{-1}) + \frac{T\eta}{2^{\gamma+1}S_{Fed}}\mathrm{tr}(\tilde{\Delta}_1) + \log(\det(\Delta_2)^{-1}), \end{aligned} \tag{8.86}$$

where $\tilde{\Delta}_1$ satisfies $(\tilde{\Delta}_1)_{i,j} = |(\Delta_1)_{i,j}|$. Then, we define

$$\begin{aligned} & g(n) = \\ & d\log(\frac{n^{\gamma-1}S_{Fed}}{S_{Cen}}) + (\frac{T\eta}{2n^{\gamma}S_{Fed}} - \frac{T\eta}{2nS_{Cen}})\mathrm{tr}(CA^{-1}) + \frac{T\eta}{2^{\gamma+1}S_{Fed}}\mathrm{tr}(\tilde{\Delta}_1) + \log(\det(\Delta_2)^{-1}). \end{aligned} \tag{8.87}$$

The derivative of $g(n)$ is:

$$g'\left(n\right) = \frac{(\gamma-1)d}{n} + \frac{T\eta}{2n^{\gamma+1}S_{Fed}S_{Cen}}(n^{\gamma-1}S_{Fed} - \gamma S_{Cen})\mathrm{tr}(CA^{-1}). \tag{8.88}$$

Since $\gamma > 1$, we know that $g'(n) > 0$ requires $n^{\gamma-1}S_{Fed} - \gamma S_{Cen} > 0$, this implies:

$$\begin{aligned} n^{\gamma-1}S_{Fed} &> \gamma S_{Cen} \\ n^{\gamma-1} &> \gamma \frac{S_{Cen}}{S_{Fed}} \\ n^{\gamma-1} &\geq \gamma \\ n &\geq \sqrt[\gamma-1]{\gamma}, \end{aligned} \tag{8.89}$$

where the third inequality adopts the fact that $S_{Cen} \geq S_{Fed}$. Since the constant γ satisfies $\gamma > 1$, we can prove $g'(n) > 0$ when $n \geq \sqrt[\gamma-1]{\gamma}$. Then, we construct another helper function and the derivative of this new helper function as follows:

$$\begin{aligned} h(x) &= x^{\frac{1}{x-1}} = e^{\frac{1}{x-1}\log(x)} \\ h'(x) &= e^{\frac{1}{x-1}\log(x)} \frac{1 - \frac{1}{x} - \log(x)}{(x-1)^2}. \end{aligned} \tag{8.90}$$

From Eq.(8.90), since $1 - \frac{1}{x} - \log(x) < 0$, it is clear that $h'(x) < 0$. Thus, we have $h(x) < h(1) = e$ and $\sqrt[\gamma-1]{\gamma} < e$. According to Eq.(8.85), the analytic solution of $O(\mathcal{G}_{Fed} - \mathcal{G}_{Cen})$ is monotonically increasing with n when $n \geq e$. Because of $n \in \mathbb{Z}^+$, substituting $n = 3$ and $n = D$ into Eq.(8.86) will derive the following inequalities for $3 \leq n \leq D$:

$$\begin{aligned} &\frac{d\log(\frac{3^{\gamma-1}k_{Fed}m}{k_{Cen}D}) + T\left(\frac{\eta\mathrm{tr}(CA^{-1})}{2*3^{\gamma}k_{Fed}m} - \frac{\eta\mathrm{tr}(CA^{-1})}{6k_{Cen}D}\right) + \frac{T\eta\mathrm{tr}(\tilde{\Delta}_1)}{2^{\gamma+1}k_{Fed}m} + \log(\det(\Delta_2)^{-1})}{4D-2} \\ &\leq O(\mathcal{G}_{Fed} - \mathcal{G}_{Cen}) \leq \\ &\frac{d\log(\frac{D^{\gamma-1}k_{Fed}m}{k_{Cen}D}) + T\left(\frac{\eta(\mathrm{tr}(CA^{-1}))}{2D^{\gamma}k_{Fed}m} - \frac{\eta(\mathrm{tr}(CA^{-1})}{2D^2k_{Cen}}\right) + \frac{T\eta\mathrm{tr}(\tilde{\Delta}_1)}{2^{\gamma+1}k_{Fed}m} + \log(\det(\Delta_2)^{-1})}{4D-2}. \end{aligned} \tag{8.91}$$

By again applying the condition $\frac{S_{Fed}}{S_{Cen}} \leq 1$, we can get a tighter lower bound as below:

$$\begin{aligned} &O(\mathcal{G}_{Fed} - \mathcal{G}_{Cen}) \\ &\geq \frac{d\log(3^{\gamma-1}) + T\left(\frac{\eta\mathrm{tr}(CA^{-1})}{2*3^{\gamma}k_{Fed}m} - \frac{\eta\mathrm{tr}(CA^{-1})}{6k_{Cen}D}\right) + \frac{T\eta\mathrm{tr}(\tilde{\Delta}_1)}{2^{\gamma+1}k_{Fed}m} + \log(\det(\Delta_2)^{-1})}{4D-2}. \end{aligned} \tag{8.92}$$

However, the lower bound of n is actually $n = 2$. To find the bound of $O(\mathcal{G}_{Fed} - \mathcal{G}_{Cen})$ covering the entire range $\{2 \leq n \leq D | n \in Z\}$, we solve:

$$\begin{aligned} \sqrt[\gamma-1]{\gamma} &= 2 \\ \gamma &= 2. \end{aligned} \tag{8.93}$$

Since $\sqrt[\gamma-1]{\gamma} < 2$ for any $\gamma > 2$, we know that when $\gamma \geq 2$ is satisfied, the following inequality holds for the case of $n = 2$:

$$O\left(\mathcal{G}_{Fed} - \mathcal{G}_{Cen}\right) \geq \frac{d\log(2^{\gamma-1}) + T\left(\frac{\eta(\mathrm{tr}(CA^{-1}) - \eta\mathrm{tr}(\tilde{\Delta}_1))}{2^{\gamma+1}k_{Fed}m} - \frac{\eta\mathrm{tr}(CA^{-1})}{4k_{Cen}D}\right) + \log(\det(\Delta_2)^{-1})}{4D - 2}, \tag{8.94}$$

which is derived by substituting $n = 2$ into the lower bound of Eq.(8.91). The proof has been completed. □

8.2.4 Proof of Valid Strategies in Closing the Gap

This section provides the proof details for Theorem 9 of the theoretical analysis in Section 4.3.5.

Proof. We define $\tilde{\mathcal{G}}_{Fed}$ for the generalisation bound of federated scenarios having an advantage in training resources and start with the case of n tends to infinity. The

generalisation performance gap $\tilde{\mathcal{G}}_{Fed} - \mathcal{G}_{Cen}$ for this case is formulated as follows:

$$
\begin{aligned}
&\tilde{\mathcal{G}}_{Fed} - \mathcal{G}_{Cen} \\
&= \frac{d\log(\frac{2k_{Fed}m}{T\eta}) - \log(\det(\bar{C}\bar{A}^{-1})) + \frac{T\eta}{2k_{Fed}m}\mathrm{tr}(\bar{C}\bar{A}^{-1}) - d + 2\log(\frac{1}{\delta}) + 2\log(nm) + 4}{4nm - 2} \\
&\quad - \frac{d\log(\frac{2nk_{Cen}D}{T\eta}) - \log(\det(CA^{-1})) + \frac{T\eta}{2nk_{Cen}D}\mathrm{tr}(CA^{-1}) - d + 2\log(\frac{1}{\delta}) + 2\log(D) + 4}{4D - 2} \\
&= \frac{d\log(\frac{2n^{\gamma}k_{Fed}m}{T\eta}) - \log(\det(CA^{-1})) + \frac{T\eta}{2n^{\gamma}k_{Fed}m}(\mathrm{tr}(CA^{-1}) + \mathrm{tr}(\Delta_1))}{4nm - 2} \\
&\quad + \frac{\log(\det(\Delta_2)^{-1}) - d + 2\log(\frac{1}{\delta}) + 2\log(nm) + 4}{4nm - 2} \\
&\quad - \frac{d\log(\frac{2nk_{Cen}D}{T\eta}) - \log(\det(CA^{-1})) + \frac{T\eta}{2nk_{Cen}D}\mathrm{tr}(CA^{-1}) - d + 2\log(\frac{1}{\delta}) + 2\log(D) + 4}{4D - 2} \\
&= \frac{d\log(\frac{2n^{\gamma}k_{Fed}m}{T\eta}) + \frac{T\eta}{2n^{\gamma}k_{Fed}m}(\mathrm{tr}(CA^{-1}) + \mathrm{tr}(\Delta_1)) + \log(\det(\Delta_2)^{-1}) - d + 2\log(nm)}{4nm - 2} \\
&\quad - \frac{d\log(\frac{2nk_{Cen}D}{T\eta}) + \frac{T\eta}{2nk_{Cen}D}\mathrm{tr}(CA^{-1}) - d + 2\log(D)}{4D - 2}.
\end{aligned}
\tag{8.95}
$$

According to the definition of PAC-Bayes bound in Lemma 1, we have $\mathcal{G}_{Cen} > 0$. Considering increasing n leads to $nm \geq D$, Eq.(8.95) turns to

$$
\begin{aligned}
&\tilde{\mathcal{G}}_{Fed} - \mathcal{G}_{Cen} \\
&= \frac{d\log(\frac{2n^{\gamma}k_{Fed}m}{T\eta}) + \frac{T\eta}{2n^{\gamma}k_{Fed}m}(\mathrm{tr}(CA^{-1}) + \mathrm{tr}(\Delta_1)) + \log(\det(\Delta_2)^{-1}) - d + 2\log(nm)}{4nm - 2} \\
&\quad - \frac{d\log(\frac{2\tilde{n}k_{Cen}D}{T\eta}) + \frac{T\eta}{2\tilde{n}k_{Cen}D}\mathrm{tr}(CA^{-1}) - d + 2\log(D)}{4D - 2},
\end{aligned}
\tag{8.96}
$$

where $n \geq \tilde{n}$. Then, we derive the limit of $\tilde{\mathcal{G}}_{Fed} - \mathcal{G}_{Cen}$ when n approaches infinity as follows:

$$
\begin{aligned}
&\lim_{n\to\infty}\left(\tilde{\mathcal{G}}_{Fed}-\mathcal{G}_{Cen}\right)= \\
&\lim_{n\to\infty}(\frac{d\log(\frac{2n^{\gamma}k_{Fed}m}{T\eta})+\frac{T\eta}{2n^{\gamma}k_{Fed}m}(\mathrm{tr}(CA^{-1})+\mathrm{tr}(\Delta_1))+\log(\det(\Delta_2)^{-1})-d+2\log(nm)}{4nm-2} \\
&\quad-\frac{d\log(\frac{2\tilde{n}k_{Cen}D}{T\eta})+\frac{T\eta}{2\tilde{n}k_{Cen}D}\mathrm{tr}(CA^{-1})-d+2\log(D)}{4D-2}) \\
&=\lim_{n\to\infty}\left(O(\frac{(\gamma d+2)\log(n)}{n})+O(\frac{1}{n^{\gamma+1}})+O(\frac{1}{n})-O(1)\right)<0.
\end{aligned}
\tag{8.97}
$$

Similarly, the limit of $\tilde{\mathcal{G}}_{Fed}-\mathcal{G}_{Cen}$ when m approaches infinity is established below:

$$
\begin{aligned}
&\lim_{m\to\infty}\left(\tilde{\mathcal{G}}_{Fed}-\mathcal{G}_{Cen}\right)= \\
&\lim_{m\to\infty}(\frac{d\log(\frac{2n^{\gamma}k_{Fed}m}{T\eta})+\frac{T\eta}{2n^{\gamma}k_{Fed}m}(\mathrm{tr}(CA^{-1})+\mathrm{tr}(\Delta_1))+\log(\det(\Delta_2)^{-1})-d+2\log(nm)}{4nm-2} \\
&\quad-\frac{d\log(\frac{2\tilde{n}k_{Cen}D}{T\eta})+\frac{T\eta}{2\tilde{n}k_{Cen}D}\mathrm{tr}(CA^{-1})-d+2\log(D)}{4D-2}) \\
&=\lim_{m\to\infty}\left(O(\frac{(d+2)\log(m)}{m})+O(\frac{1}{m^2})+O(\frac{1}{m})-O(1)\right)<0,
\end{aligned}
\tag{8.98}
$$

which completes the proof.

□

8.2.5 Proof of Invalid Strategies in Closing the Gap

This section provides the proof details for Theorem 10 of the theoretical analysis in Section 4.3.5.

Proof. Similar to the proof of Theorem 9, we study the case when T tends to positive infinity. Here, we represent the number of iterations for the centralised scenario as T_{Cen}. Increasing the number of communication rounds T in the federated scenario results in $T \geq T_{Cen}$. Thus, the performance gap $\tilde{\mathcal{G}}_{Fed}-\mathcal{G}_{Cen}$ denoted in Eq.(8.95) can be expressed as follows:

$$
\begin{aligned}
&\tilde{\mathcal{G}}_{Fed} - \mathcal{G}_{Cen} \\
&= \frac{d\log(\frac{2n^{\gamma}k_{Fed}m}{T\eta}) + \frac{T\eta}{2n^{\gamma}k_{Fed}m}(\mathrm{tr}(CA^{-1}) + \mathrm{tr}(\Delta_1)) + \log(\det(\Delta_2)^{-1}) - d + 2\log(nm)}{4nm - 2} \\
&\quad - \frac{d\log(\frac{2nk_{Cen}D}{T_{Cen}\eta}) + \frac{T_{Cen}\eta}{2nk_{Cen}D}\mathrm{tr}(CA^{-1}) - d + 2\log(D)}{4D - 2}.
\end{aligned}
\tag{8.99}
$$

It is easy to recognise that the value of $\tilde{\mathcal{G}}_{Fed} - \mathcal{G}_{Cen}$ depends on the first term in the right-hand side of Eq.(8.99) when T tends to infinity. To understand how this term changes as T increases, we need to compare the impact of $d\log\left(\frac{2n^{\gamma}k_{Fed}m}{T\eta}\right)$ and $\frac{T\eta}{2n^{\gamma}k_{Fed}m}(\mathrm{tr}(CA^{-1}) + \mathrm{tr}(\Delta_1))$, which is expressed as follows:

$$
\begin{aligned}
&\lim_{T\to\infty} \frac{d\log\left(\frac{2n^{\gamma}k_{Fed}m}{T\eta}\right)}{\frac{T\eta}{2n^{\gamma}k_{Fed}m}(\mathrm{tr}(CA^{-1}) + \mathrm{tr}(\Delta_1))} \\
&= \lim_{T\to\infty} \frac{\frac{d}{dT}\left(d\log\left(\frac{2n^{\gamma}k_{Fed}m}{T\eta}\right)\right)}{\frac{d}{dT}\left(\frac{T\eta}{2n^{\gamma}k_{Fed}m}(\mathrm{tr}(CA^{-1}) + \mathrm{tr}(\Delta_1))\right)} \\
&= \lim_{T\to\infty} \frac{-\frac{d}{T}}{\frac{\eta}{2n^{\gamma}k_{Fed}m}(\mathrm{tr}(CA^{-1}) + \mathrm{tr}(\Delta_1))} = 0.
\end{aligned}
\tag{8.100}
$$

From Eq.(8.100), we know that

$$
\lim_{T\to\infty}\left(d\log\left(\frac{2n^{\gamma}k_{Fed}m}{T\eta}\right) + \frac{T\eta}{2n^{\gamma}k_{Fed}m}(\mathrm{tr}(CA^{-1}) + \mathrm{tr}(\Delta_1))\right) = \infty. \tag{8.101}
$$

Hence, we have

$$
\lim_{T\to\infty}\left(\tilde{\mathcal{G}}_{Fed} - \mathcal{G}_{Cen}\right) = \infty. \tag{8.102}
$$

Then, we consider the case when d tends to positive infinity. Like above, we denote the model size in the centralised scenario as d_{Cen}. Since we attempt to increase the model size d in the federated scenario, we have $d \geq d_{Cen}$. With this condition, we

reformulate Eq.(8.95) with the following form:

$$\begin{aligned}
&\tilde{\mathcal{G}}_{Fed} - \mathcal{G}_{Cen} \\
&= \frac{d\log(\frac{2n^{\gamma}k_{Fed}m}{T\eta}) + \frac{T\eta}{2n^{\gamma}k_{Fed}m}(\mathrm{tr}(CA^{-1}) + \mathrm{tr}(\Delta_1)) + \log(\det(\Delta_2)^{-1}) - d + 2\log(nm)}{4nm-2} \\
&\quad - \frac{d_{Cen}\log(\frac{2nk_{Cen}D}{T\eta}) + \frac{T\eta}{2nk_{Cen}D}\mathrm{tr}(CA^{-1}) - d_{Cen} + 2\log(D)}{4D-2}.
\end{aligned} \tag{8.103}$$

When the number of clients is large enough to satisfy $n > \sqrt[\gamma]{\frac{T\eta e}{2k_{Fed}m}}$, we have

$$\begin{aligned}
n^{\gamma} &> \frac{T\eta e}{2k_{Fed}m} \\
\frac{2n^{\gamma}k_{Fed}m}{T\eta} &> e \\
\log(\frac{2n^{\gamma}k_{Fed}m}{T\eta}) &> \log(e) \\
\log(\frac{2n^{\gamma}k_{Fed}m}{T\eta}) - 1 &> 0.
\end{aligned} \tag{8.104}$$

Therefore,

$$\begin{aligned}
&\lim_{d\to\infty}\left(\tilde{\mathcal{G}}_{Fed} - \mathcal{G}_{Cen}\right) \\
&= \lim_{d\to\infty}\left(\frac{d(\log(\frac{2n^{\gamma}k_{Fed}m}{T\eta}) - 1)}{4nm-2}\right) = \infty.
\end{aligned} \tag{8.105}$$

The proof has been completed with Eqs.(8.102) and (8.105). □

8.3 Proofs of Chapter 5

8.3.1 Formal Assumptions

To make our analysis fully transparent and self-contained, we first summarize here all assumptions used in deriving the main results. These assumptions complement the problem setup in Section 5.3 and reflect the standard modeling choices commonly adopted in theoretical studies of distributed and self-supervised learning.

Assumption 5. *(Communication Graph). The distributed system is modeled as a fixed and connected communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $N = |\mathcal{V}|$ clients. Each client i communicates only with its neighborhood $A_i = \{j : (i,j) \in \mathcal{E}\} \cup \{i\}$. We assume $2 \leq |A_i| \leq N$ for all $i \in [N]$. The average neighborhood size $|\bar{A}| = \frac{1}{N}\sum_{i=1}^{N} |A_i|$ is used as a measure of connectivity.*

Remark 9. *This formulation encompasses decentralized learning on arbitrary connected topologies and federated learning as the fully connected special case (i.e., by the help of the central server, all clients can indirectly communicate with each other so there exists $|A_i| = N$ for all $i \in [N]$).*

Assumption 6. *(Consensus Weights). During decentralized aggregation, each client i forms a mixing vector $w_i = \{w_{ij}\}_{j \in A_i}$ satisfying:*

- *$w_{ij} > 0$ only if $j \in A_i$ (topology-respecting sparsity);*
- *$\sum_{j \in A_i} w_{ij} = 1$ (row-stochasticity).*

Remark 10. *The above conditions represent the standard requirements for decentralized model aggregation: each client averages only over its local neighborhood and the mixing vector is row-stochastic. This formulation covers commonly used consensus rules in decentralized optimization, including uniform averaging [100], degree-normalized weights [82], and symmetric doubly-stochastic schemes [131]. Our analysis relies only on these basic structural properties, while more general mixing operators could in principle be incorporated by extending the corresponding aggregation*

step. Exploring such extensions is an interesting direction for future work, but it is not required for the results presented here.

Assumption 7. *(Non-degenerate Embedding). Throughout the analysis, we focus on non-trivial stationary points of the regularized objectives, where the embedding matrix* $W \in \mathbb{R}^{c\times d}$ *satisfies* $W \neq 0$ *and* $\operatorname{rank}(W) = c$.

Remark 11. *The trivial solution* $W = 0$ *does not minimize the reconstruction or alignment terms in either the MIM or CL objectives, and corresponds to a representation carrying no information. Therefore, this assumption is generally satisfied in the theoretical analysis of self-supervised learning [87, 149].*

Assumption 8. *(Independence Local Sampling.) For each client* i*, the local dataset* D_i *consists of* $|D_i|$ *independent samples drawn from its local distribution* $\mathcal{D}_i$.

Remark 12. *The independence assumption is the minimal condition required for high-probability spectral norm bounds of empirical covariance matrices. It does not alter the established non-IID structure across clients, but ensures that the empirical covariance on each client concentrates around its population counterpart. This is a standard assumption in the theoretical analysis of distributed learning [131, 149].*

Assumption 9. *(Dissimilar Image Transformation for CL). When the two augmented views* $(g_a(x), g_b(x))$ *used in CL are generated from dissimilar transformations, we model* $g_b(x)$ *by a linear operator* $H \in \mathbb{R}^{d\times d}$ *acting on the input space. In the theoretical analysis,* H *enters only through the quadratic form* $x^\top Hx$*, and therefore only its symmetric component* $H_{\text{sym}} = (H + H^\top)/2$ *is relevant. Let* $\mathcal{S} = \operatorname{span}\{e_1, \ldots, e_c\}$ *denote the class-dependent subspace in the non-IID generative model, and let* P *be the orthogonal projection onto* $\mathcal{S}$*. We assume that*

$$\operatorname{tr}(PH_{\text{sym}}P) > 0.$$

Remark 13. *The above condition ensures that the transformation* H *preserves nontrivial energy on the class-dependent semantic subspace* $\mathcal{S}$*. It is a mild requirement*

and is common to hold in standard contrastive learning augmentations [14, 15], including rotations, flips, translations, crops, blurs, and color jittering. These transformations perturb the input in ways that do not cancel class-discriminative directions, so $\mathrm{tr}(PH_{\mathrm{sym}}P)$ *remains strictly positive in practice.*

8.3.2 Learned Representability for Distributed MIM

This section provides the full proof of Theorem 11 of the theoretical analysis in Section 5.4.1.

Proof. We begin by formulating the representability of local representation. Then, we derive the global representation based on the local feature. Since FL is different from decentralized learning in the updates, we establish the global representation for each distributed framework, respectively.

For local feature space. According to the alignment-style loss function of MIM shown in Eq.(5.6) and by the definition of Kronecker product, we have

$$\begin{aligned}\mathcal{L}_{MIM} &= -\mathbb{E}[|(W(x \odot m))^\intercal (W(x \odot (1-m)))] + \frac{1}{2}||W^\intercal W||_F^2 \\ &= -\mathbb{E}[(W(\mathrm{diag}(\mathrm{vec}(x)) \cdot \mathrm{vec}(m)))^\intercal (W(\mathrm{diag}(\mathrm{vec}(x)) \cdot \mathrm{vec}(1-m)))] + \frac{1}{2}||W^\intercal W||_F^2.\end{aligned} \tag{8.106}$$

Define

$$a = \mathrm{diag}(\mathrm{vec}(x)) \cdot \mathrm{vec}(m), \qquad b = \mathrm{diag}(\mathrm{vec}(x)) \cdot \mathrm{vec}(1-m), \tag{8.107}$$

so that the above loss becomes

$$\mathcal{L}_{MIM} = -\mathbb{E}[a^\intercal W^\intercal W b] + \frac{1}{2}||W^\intercal W||_F^2. \tag{8.108}$$

Using the fact that

$$a^\intercal W^\intercal W b = \mathrm{tr}(a^\intercal W^\intercal W b) = \mathrm{tr}(W^\intercal W b a^\intercal) = \mathrm{tr}(W b a^\intercal W^\intercal), \tag{8.109}$$

we obtain

$$\frac{\partial}{\partial W}\big(a^\intercal W^\intercal W b\big) = \frac{\partial}{\partial W}(\mathrm{tr}(W b a^\intercal W^\intercal)) = W(ba^\intercal + ab^\intercal). \tag{8.110}$$

Together with

$$\frac{\partial(\frac{1}{2}\|W^\top W\|_F^2)}{\partial W} = 2WW^\intercal W, \tag{8.111}$$

the gradient of the complete objective becomes

$$\frac{\partial \mathcal{L}_{\mathrm{MIM}}}{\partial W} = -W\mathbb{E}\big[ba^\intercal + ab^\intercal\big] + 2WW^\intercal W. \tag{8.112}$$

Setting the gradient to zero yields

$$W\mathbb{E}\left[ba^\intercal + ab^\intercal\right] = 2WW^\intercal W. \tag{8.113}$$

Under Assumption 7, multiplying both sides on the left by the Moore–Penrose pseudoinverse W^+ reduces Eq.(8.113) to the stationary condition

$$\frac{1}{2}\mathbb{E}\left[ba^\intercal + ab^\intercal\right] = W^\intercal W. \tag{8.114}$$

Let X_i^M represent the left-hand side of this equation. Consider the binary matrix m used for masking is sampled uniformly from the binomial distribution with a probability p, we establish

$$\begin{aligned} X_i^M = \frac{1}{2}\mathbb{E}[&\mathrm{diag}(\mathrm{vec}(x))\mathrm{vec}(1-m)\mathrm{vec}(m)^\intercal\mathrm{diag}(\mathrm{vec}(x))^\intercal \\ &+ \mathrm{diag}(\mathrm{vec}(x))\mathrm{vec}(m)\mathrm{vec}(1-m)^\intercal\mathrm{diag}(\mathrm{vec}(x))^\intercal] \\ = \frac{2p\,(1-p)}{|D_i|}&\sum_{j=1}^{|D_i|}\left(\mathrm{diag}\left(\mathrm{vec}\left(x_{i,j}\right)\right)\mathrm{diag}\left(\mathrm{vec}\left(x_{i,j}\right)\right)^\intercal\right), \end{aligned} \tag{8.115}$$

where $\mathbb{E}_{x\sim D_i}[xx^\intercal] = \frac{1}{|D_i|}\sum_{j=1}^{|D_i|}\left(\mathrm{diag}\left(\mathrm{vec}\left(x_{i,j}\right)\right)\mathrm{diag}\left(\mathrm{vec}\left(x_{i,j}\right)\right)^\intercal\right)$ denotes the empirical covariance matrix for the learning with local dataset on client i. Based on the setup of data generation in Section 5.3.2, we also derive the following expectation of X_i^M

with $\tau = d^{\frac{1}{5}}$ and $\mu = d^{-\frac{1}{5}}$:

$$\begin{aligned}
&\mathbb{E}\left[X_i^M\right] = \text{diag} \\
&\Bigg(\underbrace{2p\left(1-p\right)\tau^2 + O\left(d^{-\frac{2}{5}}\right), ..., \underbrace{2p\left(1-p\right) + O\left(d^{-\frac{2}{5}}\right)}_{i^{th}\text{ term}}, ..., \ 2p\left(1-p\right)\tau^2 + O\left(d^{-\frac{2}{5}}\right)}_{N\text{ terms}}, \\
&\underbrace{O\left(d^{-\frac{2}{5}}\right), ..., O\left(d^{-\frac{2}{5}}\right)}_{d-N\text{ terms}}\Bigg) \\
&= \text{diag}\Bigg(2p\left(1-p\right)d^{\frac{2}{5}} + O\left(d^{-\frac{2}{5}}\right), ..., 2p\left(1-p\right) + O\left(d^{-\frac{2}{5}}\right), \\
&\qquad ..., 2p\left(1-p\right)d^{\frac{2}{5}} + O\left(d^{-\frac{2}{5}}\right), ..., O\left(d^{-\frac{2}{5}}\right)\Bigg)
\end{aligned} \tag{8.116}$$

Next, consider the fact that up to a positive scaling and an additive constant, the regularized MIM objective can be rewritten as the Frobenius-norm objective $\mathcal{L}(W) = \|X_i^M - W^\intercal W\|_F^2$. Thus, minimizing $\mathcal{L}_{\text{MIM}}$ solves the Frobenius-norm best rank-c approximation problem for X_i^M. According to the Eckart-Young-Mirsky theorem [25], we notice that the row span of the optimal $W \in \mathbb{R}^{c\times d}$ is the span of the eigenvectors corresponding to the first c eigenvalues of X_i^M. Denoting the set of orthonormal eigenvectors of X_i^M as $\left\{v_{i,1}^M, ..., v_{i,d}^M\right\}$, we have $X_i^M = \sum_{j=1}^d \lambda_{i,j} v_{i,j}^M (v_{i,j}^M)^\intercal$, where $\lambda_{i,j} := \lambda_j(X_i^M)$ is the j-th largest eigenvalue of X_i^M. Therefore, the inequality below is satisfied:

$$\begin{aligned}
e_k^\intercal X_i^M e_k &= e_k^\intercal \left(\sum_{j=1}^d \lambda_{i,j} v_{i,j}^M (v_{i,j}^M)^\intercal\right) e_k \\
&= \sum_{j=1}^d \lambda_{i,j} (e_k^\intercal v_{i,j}^M)^2 \\
&\leq \lambda_{i,1}^M \sum_{j=1}^d (e_k^\intercal v_{i,j}^M)^2,
\end{aligned} \tag{8.117}$$

for any e_k with $k \in [N]\backslash\{i\}$. On the other hand, under the data construction described in Section 5.3, the number of samples on each client equals the sum of the samples from frequent classes and the rare class. Since each of the two frequent

classes grows in polynomials of d, while the amount of data from the rare class is $O(d^{\alpha})$ with $\alpha \in (0,1)$, the local sample size satisfies $|D_i| = \Theta(d^{\beta})$ with $\beta \geq 1$. Based on this sufficiently large sample size and Assumption 8, the matrix concentration bounds [144] implies that the spectral norm satisfies $\|X_i^M - \mathbb{E}\left[X_i^M\right]\|_2 \leq O\left(d^{-\frac{2}{5}}\right)$ with probability at least $1 - \frac{1}{2}e^{-d^{\frac{1}{10}}}$. Building on Weyl's inequality, we obtain that with high probability,

$$\left|\lambda_{i,k}^M - \lambda_k \mathbb{E}\left[X_i^M\right]\right| \leq \|X_i^M - \mathbb{E}\left[X_i^M\right]\|_2 \leq O\left(d^{-\frac{2}{5}}\right). \tag{8.118}$$

By combining Eqs.(8.116), (8.117) and (8.118), we can derive the below lower bound for $e_k^\intercal X_i^M e_k$:

$$\begin{aligned} e_k^\intercal X_i^M e_k &= e_k^\intercal \mathbb{E}\left[X_i^M\right] e_k + e_k^\intercal \left[X_i^M - \mathbb{E}\left[X_i^M\right]\right] e_k \\ &\geq 2p\left(1-p\right) d^{\frac{2}{5}} + O\left(d^{-\frac{2}{5}}\right) - \|X_i^M - \mathbb{E}\left[X_i^M\right]\| \\ &\geq 2p\left(1-p\right) d^{\frac{2}{5}} - O\left(d^{-\frac{2}{5}}\right), \end{aligned} \tag{8.119}$$

which is led by the fact that $\|X\|_{\max} \leq \|X\|$ for symmetric X. Likewise, we prove the upper bound as follows:

$$\begin{aligned} e_k^\intercal X_i^M e_k &= e_k^\intercal \mathbb{E}\left[X_i^M\right] e_k + e_k^\intercal \left[X_i^M - \mathbb{E}\left[X_i^M\right]\right] e_k \\ &\leq 2p\left(1-p\right) d^{\frac{2}{5}} + O\left(d^{-\frac{2}{5}}\right) + \|X_i^M - \mathbb{E}\left[X_i^M\right]\| \\ &\leq 2p\left(1-p\right) d^{\frac{2}{5}} + O\left(d^{-\frac{2}{5}}\right). \end{aligned} \tag{8.120}$$

Moreover, we notice from Eqs.(8.116) and (8.117) that the following statements hold for $\lambda_{i,1}^M$:

$$\begin{aligned} \lambda_{i,1}^M &\geq \lambda_1\left(\mathbb{E}\left[X_i^M\right]\right) - O\left(d^{-\frac{2}{5}}\right) \geq 2p\left(1-p\right) d^{\frac{2}{5}} - O\left(d^{-\frac{2}{5}}\right) \\ \lambda_{i,1}^M &\leq \lambda_1\left(\mathbb{E}\left[X_i^M\right]\right) + O\left(d^{-\frac{2}{5}}\right) = 2p\left(1-p\right) d^{\frac{2}{5}} + O\left(d^{-\frac{2}{5}}\right). \end{aligned} \tag{8.121}$$

With Eqs.(8.119) - (8.121), we further establish

$$\begin{aligned}\sum_{j=1}^{d}(e_k^\intercal v_j^M)^2 &\geq \frac{2p\,(1-p)\,d^{\frac{2}{5}}-O\left(d^{-\frac{2}{5}}\right)}{2p\,(1-p)\,d^{\frac{2}{5}}+O\left(d^{-\frac{2}{5}}\right)}\\ &=\frac{2p\,(1-p)\,d^{\frac{2}{5}}+O\left(d^{-\frac{2}{5}}\right)}{2p\,(1-p)\,d^{\frac{2}{5}}+O\left(d^{-\frac{2}{5}}\right)}-\frac{2O\left(d^{-\frac{2}{5}}\right)}{2p\,(1-p)\,d^{\frac{2}{5}}+O\left(d^{-\frac{2}{5}}\right)}\\ &=1-\frac{O\left(d^{-\frac{2}{5}}\right)}{2p\,(1-p)\,d^{\frac{2}{5}}+O\left(d^{-\frac{2}{5}}\right)}.\end{aligned} \tag{8.122}$$

This completes the proof of local representation.

For global feature space. Since the local goal can be equivalently re-formulated as $\|X_i^M - W^\intercal W\|_F^2$, we rewrite the global goal of D-SSL for the DecL framework (shown in Eq.(5.1)) as

$$\min_W \frac{1}{N}\sum_{i\in[N]}\frac{1}{|A_i|}\sum_{j\in A_i}\|X_j^M - W^\intercal W\|_F^2. \tag{8.123}$$

Note that the following function holds the same minimiser as Eq.(8.123):

$$\begin{aligned}&\min_W\|\frac{1}{N}\sum_{i\in[N]}\frac{1}{|A_i|}\sum_{j\in A_i}X_j^M - W^\intercal W\|_F^2\\ &=\min_W\|\frac{1}{N}\sum_{i\in[N]}\overline{X_i^M} - W^\intercal W\|_F^2\\ &=\min_W\|\overline{X^M} - W^\intercal W\|_F^2,\end{aligned} \tag{8.124}$$

where $\overline{X_i^M} = \sum_{j\in A_i}\frac{1}{|A_i|}X_j^M$ denotes the empirical covariance matrix for training with the local datasets across the local datasets on client i and its neighbours. So, finding the optimal W for DecL is equivalent to solving Eq.(8.124). Following the

derivation of Eq.(8.116) and linearity of expectation, we establish

$$\mathbb{E}\left(\overline{X_i^M}\right) = \text{diag}$$
$$\Bigg(\underbrace{...,\underbrace{2p\left(1-p\right)\left(\left(1-\frac{1}{|A_i|}\right)d^{\frac{2}{5}}+\frac{1}{|A_i|}\right)+O\left(d^{-\frac{2}{5}}\right)}_{j\in A_i\backslash i},...,\underbrace{2p\left(1-p\right)d^{\frac{2}{5}}+O\left(d^{-\frac{2}{5}}\right)}_{i^{th}\text{term}},...,}_{N\text{ terms}}$$
$$\underbrace{O\left(d^{-\frac{2}{5}}\right),...,O\left(d^{-\frac{2}{5}}\right)}_{d-N\text{ terms}}\Bigg), \tag{8.125}$$

where we prove with the fact that

$$\begin{aligned}
&\frac{\left(|A_i|-1\right)2p\left(1-p\right)d^{\frac{2}{5}}+2p\left(1-p\right)+|A_i|\,O\left(d^{-\frac{2}{5}}\right)}{|A_i|}\\
&=\frac{\left(|A_i|-1\right)2p\left(1-p\right)d^{\frac{2}{5}}+2p\left(1-p\right)}{|A_i|}+O\left(d^{-\frac{2}{5}}\right)\\
&=2p\left(1-p\right)\left(1-\frac{1}{|A_i|}\right)d^{\frac{2}{5}}+2p\left(1-p\right)\frac{1}{|A_i|}+O\left(d^{-\frac{2}{5}}\right)\\
&=2p\left(1-p\right)\left(\left(1-\frac{1}{|A_i|}\right)d^{\frac{2}{5}}+\frac{1}{|A_i|}\right)+O\left(d^{-\frac{2}{5}}\right).
\end{aligned} \tag{8.126}$$

With Eq.(8.125), we can also have

$$\mathbb{E}\left(\overline{X^M}\right) = \text{diag}$$
$$\Bigg(\underbrace{2p\left(1-p\right)\left(1-\frac{1}{|\bar{A}|}\right)d^{\frac{2}{5}}+O\left(d^{-\frac{9}{20}}\right),...,2p\left(1-p\right)\left(1-\frac{1}{|\bar{A}|}\right)d^{\frac{2}{5}}+O\left(d^{-\frac{9}{20}}\right),}_{N\ terms}$$
$$...,O\left(d^{-\frac{2}{5}}\right)\Bigg) \tag{8.127}$$

where we consider $\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|A_i|}=\left|\bar{A}\right|$ and the fact that

$$\begin{aligned}
&\frac{\sum_{i=1}^{N}\left(2p\left(1-p\right)\left(\left(1-\frac{1}{|A_i|}\right)d^{\frac{2}{5}}+\frac{1}{|A_i|}\right)+O\left(d^{-\frac{2}{5}}\right)\right)}{N}\\
&=2p\left(1-p\right)\left(\left(1-\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|A_i|}\right)d^{\frac{2}{5}}+\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|A_i|}\right)+O\left(d^{-\frac{2}{5}}\right)\\
&=2p\left(1-p\right)\left(\left(1-\frac{1}{\left|\bar{A}\right|}\right)d^{\frac{2}{5}}+\frac{1}{\left|\bar{A}\right|}\right)+O\left(d^{-\frac{2}{6}}\right)\\
&=2p\left(1-p\right)\left(1-\frac{1}{\left|\bar{A}\right|}\right)d^{\frac{2}{5}}+O\left(d^{-\frac{2}{5}}\right).
\end{aligned}\tag{8.128}$$

Through similar proof from Eq.(8.119) to Eq.(8.121), we prove that the following statements hold for all $i\in[N]$:

$$\begin{aligned}
e_k^{\intercal}\overline{X^M}e_k&\geq 2p\left(1-p\right)\left(1-\frac{1}{\left|\bar{A}\right|}\right)d^{\frac{2}{5}}-O\left(d^{-\frac{2}{5}}\right)\\
e_k^{\intercal}\overline{X^M}e_k&\leq 2p\left(1-p\right)\left(1-\frac{1}{\left|\bar{A}\right|}\right)d^{\frac{2}{5}}+O\left(d^{-\frac{2}{5}}\right)
\end{aligned}\tag{8.129}$$

$$\begin{aligned}
\lambda_{i,1}^{M}&\geq\lambda_1\left(\mathbb{E}\left[\overline{X^M}\right]\right)+O\left(d^{-\frac{2}{5}}\right)=2p\left(1-p\right)\left(1-\frac{1}{\left|\bar{A}\right|}\right)d^{\frac{2}{5}}-O\left(d^{-\frac{2}{5}}\right)\\
\lambda_{i,1}^{M}&\leq\lambda_1\left(\mathbb{E}\left[\overline{X^M}\right]\right)+O\left(d^{-\frac{2}{5}}\right)=2p\left(1-p\right)\left(1-\frac{1}{\left|\bar{A}\right|}\right)d^{\frac{2}{5}}+O\left(d^{-\frac{2}{5}}\right),
\end{aligned}\tag{8.130}$$

which then implies:

$$\begin{aligned}
\sum_{j=1}^{d}(e_k^{\intercal}\bar{v}_j^M)^2 &\geq \frac{2p\left(1-p\right)\left(1-\frac{1}{|\bar{A}|}\right)d^{\frac{2}{5}}-O\left(d^{-\frac{2}{5}}\right)}{2p\left(1-p\right)\left(1-\frac{1}{|\bar{A}|}\right)d^{\frac{2}{5}}+O\left(d^{-\frac{2}{5}}\right)} \\
&= \frac{2p\left(1-p\right)\left(1-\frac{1}{|\bar{A}|}\right)d^{\frac{2}{5}}+O\left(d^{-\frac{2}{5}}\right)}{2p\left(1-p\right)\left(1-\frac{1}{|\bar{A}|}\right)d^{\frac{2}{5}}+O\left(d^{-\frac{2}{5}}\right)}-\frac{2O\left(d^{-\frac{2}{5}}\right)}{2p\left(1-p\right)\left(1-\frac{1}{|\bar{A}|}\right)d^{\frac{2}{5}}+O\left(d^{-\frac{2}{5}}\right)} \\
&= 1-\frac{O\left(d^{-\frac{2}{5}}\right)}{2p\left(1-p\right)\left(1-\frac{1}{|\bar{A}|}\right)d^{\frac{2}{5}}+O\left(d^{-\frac{2}{5}}\right)}.
\end{aligned} \tag{8.131}$$

The proof of the global featured space learned in the decentralised learning framework has been completed. Next, consider federated learning (FL) as a special case of decentralised learning with $\forall i \in [N], |A_i| = N$. The global objective of FL is thus:

$$\min_{W}\frac{1}{N}\sum_{i\in[N]}\|X_i^M - W^{\intercal}W\|_F^2. \tag{8.132}$$

This is similar to solving

$$\min_{W}\|\overline{X^M} - W^{\intercal}W\|_F^2, \tag{8.133}$$

where $\overline{X^M} := \frac{1}{N}\sum_{i\in[N]}X_i^M$ denotes the empirical covariance matrix for learning with the global dataset. Then, we derive

$$\begin{aligned}
\mathbb{E}\left(\overline{X^M}\right) = \operatorname{diag}\Bigg(&2p\left(1-p\right)d^{\frac{2}{5}}-\Theta\left(d^{\frac{7}{20}}\right)+O\left(d^{-\frac{2}{5}}\right),...,\\
&2p\left(1-p\right)d^{\frac{2}{5}}-\Theta\left(d^{\frac{7}{20}}\right)+O\left(d^{-\frac{2}{5}}\right),...O\left(d^{-\frac{2}{5}}\right)\Bigg)
\end{aligned} \tag{8.134}$$

where we adopt $N = \Theta(d^{\frac{1}{20}})$ have used the fact that

$$\begin{aligned}
&\frac{(N-1)\,2p\,(1-p)\,d^{\frac{2}{5}} + 2p\,(1-p) + NO\left(d^{-\frac{2}{5}}\right)}{N} \\
&= \frac{\left(\Theta\left(d^{\frac{1}{20}}\right) - 1\right) 2p\,(1-p)\,d^{\frac{2}{5}} + 2p\,(1-p)}{\Theta\left(d^{\frac{1}{20}}\right)} + O\left(d^{-\frac{2}{5}}\right) \\
&= 2p\,(1-p)\left(1 - \Theta\left(d^{-\frac{1}{20}}\right)\right) d^{\frac{2}{5}} + \Theta\left(d^{-\frac{1}{20}}\right) + O\left(d^{-\frac{2}{5}}\right) \\
&= 2p\,(1-p)\,d^{\frac{2}{5}} - \Theta\left(d^{\frac{7}{20}}\right) + O\left(d^{-\frac{2}{5}}\right).
\end{aligned} \tag{8.135}$$

Again, by similar arguments from Eq.(8.119) to Eq.(8.121), we further prove

$$\begin{aligned}
&\sum_{j=1}^{d} (e_k^{\intercal} \bar{v}_j^M)^2 \geq \frac{2p\,(1-p)\,d^{\frac{2}{5}} - \Theta\left(d^{\frac{7}{20}}\right) - O\left(d^{-\frac{2}{5}}\right)}{2p\,(1-p)\,d^{\frac{2}{5}} - \Theta\left(d^{\frac{7}{20}}\right) + O\left(d^{-\frac{2}{5}}\right)} \\
&= \frac{2p\,(1-p)\,d^{\frac{2}{5}} - \Theta\left(d^{\frac{7}{20}}\right) + O\left(d^{-\frac{2}{5}}\right)}{2p\,(1-p)\,d^{\frac{2}{5}} - \Theta\left(d^{\frac{7}{20}}\right) + O\left(d^{-\frac{2}{5}}\right)} - \frac{2O\left(d^{-\frac{2}{5}}\right)}{p\,(1-p)\,d^{\frac{2}{5}} - \Theta\left(d^{\frac{7}{20}}\right) + O\left(d^{-\frac{2}{5}}\right)} \\
&= 1 - \frac{O\left(d^{-\frac{2}{5}}\right)}{2p\,(1-p)\,d^{\frac{2}{5}} - \Theta\left(d^{\frac{7}{20}}\right) + O\left(d^{-\frac{2}{5}}\right)},
\end{aligned} \tag{8.136}$$

which completes the proof of this theorem. □

8.3.3 Learned Representability for Distributed CL

This section provides the full proof of Theorem 12 of the theoretical analysis in Section 5.4.1.

Lemma 13. *(Representability of Distributed CL under Similar Augmentations). Consider the same distributed scenario in Theorem 11. For distributed SSL that utilises Contrastive Learning (CL) in pre-training and generates positive pairs through similar augmentations, with a high probability, the following statements hold:*

1. *Let* $r_i^C = [r_{i,1}^C, \ldots, r_{i,c}^C]^\intercal$ *be the local RV learned on client* i*. If positive pairs are generated by similar augmentations, we have* $1 - \frac{O(d^{-\frac{1}{5}})}{d^{\frac{2}{5}}+O(d^{-\frac{1}{5}})} \leq r_{i,k}^C \leq 1$*, where* $i \in [N] \backslash k$*.*

2. *Let* $\bar{r}_{Dec}^C = [\bar{r}_1^C, \ldots, \bar{r}_c^C]^\intercal$ *be the RV learned through the global objective of DecL framework, then we have* $1 - \frac{O(d^{-\frac{1}{5}})}{(1-\frac{1}{|A|})d^{\frac{2}{5}}+O(d^{-\frac{1}{5}})} \leq \bar{r}^C \leq 1$*.*

3. *Let* $\bar{r}_{Fed}^M = [\bar{\boldsymbol{r}}_1^M, \ldots, \bar{\boldsymbol{r}}_c^M]^\intercal$ *be the RV learned through the global objective of FL framework, we have* $1 - \frac{O(d^{-\frac{1}{5}})}{d^{\frac{2}{5}}-\Theta(d^{\frac{7}{20}})+O(d^{-\frac{1}{5}})} \leq \bar{r}_{Fed}^C \leq 1$*.*

Proof. Following the proof in 8.3.2, we first discuss local representability learned by distributed contrastive learning and then derive the global representation based on these local features. Since federated learning differs from decentralised learning in terms of updates, we construct separate global representations for each distributed framework.

For local feature space. Based on the loss function of contrastive learning (CL) as shown in Eq.(5.3), we obtain

$$\begin{aligned}
\mathcal{L}_{CL} &= -\mathbb{E}_{x\sim D_i}||(W(x+\xi))^\intercal(W(x+\xi'))||^2 + \frac{1}{2}||W^\intercal W||_F^2 \\
&= -\mathbb{E}||\left(x^\intercal W^\intercal + \xi^\intercal W^\intercal\right)\left(W\left(x+\xi'\right)\right)||^2 + \frac{1}{2}\|W^\intercal W\|_F^2 \\
&= -\mathbb{E}||\left(x^\intercal W^\intercal W x + x^\intercal W^\intercal W \xi' + \xi^\intercal W^\intercal W x + \xi^\intercal W^\intercal W \xi'\right)||^2 + \frac{1}{2}\|W^\intercal W\|_F^2.
\end{aligned} \tag{8.137}$$

To find the minimiser of this function, we solve for

$$\frac{\partial \mathcal{L}_{CL}}{\partial W} = -2W\mathbb{E}\left[(x^\intercal x + x^\intercal \xi' + \xi^\intercal x + \xi^\intercal \xi')\right] + 2WW^\intercal W = 0, \tag{8.138}$$

leading to

$$\mathbb{E}\left[(x^\intercal x + x^\intercal \xi' + \xi^\intercal x + \xi^\intercal \xi')\right] = W^\intercal W. \tag{8.139}$$

Similarly, let X_i^C represent the left-hand side of this equation. We can then establish

$$X_i^C = \frac{1}{|D_i|} \sum_{j=1}^{|D_i|} (x^\intercal x + x^\intercal \xi' + \xi^\intercal x + \xi^\intercal \xi'), \tag{8.140}$$

where X_i^C represents the empirical covariance matrix for the local feature learned by CL on client i. Considering that $\xi, \xi' \sim \mathcal{N}(0, I)$, we also derive the following expectation of X_i^C:

$$\begin{aligned}
&\mathbb{E}\left[X_i^C\right] = \text{diag} \\
&\Bigg(\underbrace{\tau^2 + O\left(d^{-\frac{2}{5}}\right) + 2O\left(d^{-\frac{1}{5}}\right), ..., \underbrace{1 + O\left(d^{-\frac{2}{5}}\right) + 2O\left(d^{-\frac{1}{5}}\right)}_{i^{\text{th}} \text{ term}}, ..., \tau^2 + O\left(d^{-\frac{2}{5}}\right) + 2O\left(d^{-\frac{1}{5}}\right)}_{N \text{ terms}}, \\
&\quad \ldots \underbrace{2O\left(d^{-\frac{1}{5}}\right) + O\left(d^{-\frac{2}{5}}\right), ..., 2O\left(d^{-\frac{1}{5}}\right) + O\left(d^{-\frac{2}{5}}\right)}_{d-N \text{ terms}} \Bigg) \\
&= \text{diag}\left(d^{\frac{2}{5}} + O\left(d^{-\frac{1}{5}}\right), ..., 1 + O\left(d^{-\frac{1}{5}}\right), ..., d^{\frac{2}{5}} + O\left(d^{-\frac{1}{5}}\right), ..., O\left(d^{-\frac{1}{5}}\right)\right)
\end{aligned} \tag{8.141}$$

Next, using similar arguments from Eqs. (8.117) to (8.121), we arrive at the results below:

$$\begin{aligned}
d^{\frac{2}{5}} - O\left(d^{-\frac{1}{5}}\right) \leq e_k^\intercal X_i^C e_k \leq d^{\frac{2}{5}} + O\left(d^{-\frac{1}{5}}\right) \\
d^{\frac{2}{5}} - O\left(d^{-\frac{1}{5}}\right) \leq \lambda_{i,1}^C \leq d^{\frac{2}{5}} + O\left(d^{-\frac{1}{5}}\right).
\end{aligned} \tag{8.142}$$

With these inequalities, we derive

$$\begin{aligned}
\sum_{j=1}^{d} (e_k^\intercal v_j^C)^2 &\geq \frac{d^{\frac{2}{5}} - O\left(d^{-\frac{1}{5}}\right)}{d^{\frac{2}{5}} + O\left(d^{-\frac{1}{5}}\right)} \\
&= 1 - \frac{O\left(d^{-\frac{1}{5}}\right)}{d^{\frac{2}{5}} + O\left(d^{-\frac{1}{5}}\right)},
\end{aligned} \tag{8.143}$$

which completes the proof of the local part.

For global feature space. Since the local goal can be equivalently reformulated as $\|X_i^C - W^\intercal W\|_F^2$, the global goal of distributed contrastive learning in the decentralised learning (DecL) framework is given by

$$\min_W \sum_{i\in[N]} \frac{1}{N} \sum_{j\in A_i} \frac{1}{|A_i|} \|X_j^C - W^\intercal W\|_F^2. \tag{8.144}$$

Furthermore, we find that this is equivalent to solving

$$\begin{aligned} &\min_W \| \frac{1}{N} \sum_{i\in[N]} \frac{1}{|A_i|} \sum_{j\in A_i} X_j^C - W^\intercal W\|_F^2 \\ =& \min_W \| \frac{1}{N} \sum_{i\in[N]} \overline{X_i^C} - W^\intercal W\|_F^2 \\ =& \min_W \| \overline{X^C} - W^\intercal W\|_F^2. \end{aligned} \tag{8.145}$$

Again, using similar arguments from Eq. (8.125) to Eq. (8.131), we further establish

$$\begin{aligned} \sum_{k=1}^{d} (e_k^\intercal \bar{v}_j^C)^2 &\geq \frac{\left(1 - \frac{1}{|\bar{A}|}\right) d^{\frac{2}{5}} - O\left(d^{-\frac{1}{5}}\right)}{\left(1 - \frac{1}{|\bar{A}|}\right) d^{\frac{2}{5}} + O\left(d^{-\frac{1}{5}}\right)} \\ &= 1 - \frac{O\left(d^{-\frac{1}{5}}\right)}{\left(1 - \frac{1}{|\bar{A}|}\right) d^{\frac{2}{5}} + O\left(d^{-\frac{1}{5}}\right)}. \end{aligned} \tag{8.146}$$

The proof of the global feature space learned in the DecL framework has been completed. Next, denote federated learning (FL) as a special case of decentralised learning with $\forall i, |A_i| = N$. The global objective of FL is expressed as

$$\min_W \| \overline{X^C} - W^\intercal W\|_F^2. \tag{8.147}$$

where we denote $\overline{X^C} := \frac{1}{N}\sum_{i\in[N]} X_i^C$. By similar arguments from Eq. (8.134) to Eq. (8.136), we have

$$\begin{aligned}
\sum_{j=1}^{d}(e_k^\intercal \bar{v}_j^C)^2 &\geq \frac{d^{\frac{2}{5}} - \Theta\left(d^{\frac{7}{20}}\right) - O\left(d^{-\frac{1}{5}}\right)}{d^{\frac{2}{5}} - \Theta\left(d^{\frac{7}{20}}\right) + O\left(d^{-\frac{1}{5}}\right)} \\
&= \frac{d^{\frac{2}{5}} - \Theta\left(d^{\frac{7}{20}}\right) + O\left(d^{-\frac{1}{5}}\right)}{d^{\frac{2}{5}} - \Theta\left(d^{\frac{7}{20}}\right) + O\left(d^{-\frac{1}{5}}\right)} - \frac{2O\left(d^{-\frac{1}{5}}\right)}{d^{\frac{2}{5}} - \Theta\left(d^{\frac{7}{20}}\right) + O\left(d^{-\frac{1}{5}}\right)} \\
&= 1 - \frac{O\left(d^{-\frac{1}{5}}\right)}{d^{\frac{2}{5}} - \Theta\left(d^{\frac{7}{20}}\right) + O\left(d^{-\frac{1}{5}}\right)},
\end{aligned} \tag{8.148}$$

which completes the proof of this lemma.

□

Then, we start to prove Theorem 12 as follows.

Proof. Lemma 13 demonstrates the learned local and global representations of distributed CL when positive pairs are generated by similar augmentations. For the other case using dissimilar augmentations, we adopt a similar process to derive the local and global representations.

For local feature space. According to the loss function of contrastive learning (CL) with dissimilar augmentations in Eq.(5.4), we have

$$\begin{aligned}
\mathcal{L}'_{CL} &= -\mathbb{E}_{x\sim D}\left\|\left(W(x+\xi)\right)^\top WHx\right\|^2 + \frac{1}{2}\left\|W^\top W\right\|_F^2 \\
&= -\mathbb{E}\left[\left(x^\top W^\top WHx + \xi^\top W^\top WHx\right)\right] + \frac{1}{2}\left\|W^\top W\right\|_F^2.
\end{aligned} \tag{8.149}$$

The minimiser of this loss function is

$$\frac{\partial \mathcal{L}'_{CL}}{\partial W} = -2W\mathbb{E}\left[x^\intercal Hx + \xi^\intercal Hx\right] + 2WW^\intercal W = 0. \tag{8.150}$$

Rearranging it derives

$$\mathbb{E}\left[(x+\xi)^\intercal Hx\right] = W^\intercal W. \tag{8.151}$$

Let $X_i^{C'}$ denote the left-hand side of the above equation. Hence,

$$X_i^{C'} = \mathbb{E}\left[(x+\xi)^\intercal Hx\right] = \frac{1}{|D_i|}\left(\sum_{j=1}^{|D_i|} x_{i,j}^\intercal H x_{i,j} + \sum_{j=1}^{|D_i|} \xi^\intercal H x_{i,j}\right). \tag{8.152}$$

Similarly, based on the formulation that $\xi \sim \mathcal{N}(0, I)$, $\tau = d^{\frac{1}{5}}$ and $\mu = d^{-\frac{1}{5}}$, the expectation of $X_i^{C'}$ can be written as

$$\begin{aligned}
&\mathbb{E}(X_i^{C'}) = \operatorname{diag} \\
&\underbrace{\left(\operatorname{tr}(H)\tau^2 + O\left(d^{-\frac{2}{5}}\right), \ldots, \underbrace{\operatorname{tr}(H) + O\left(d^{-\frac{2}{5}}\right)}_{i^{\text{th}}\text{ term}}, \ldots, \operatorname{tr}(H)\tau^2 + O\left(d^{-\frac{2}{5}}\right), \ldots, O\left(d^{-\frac{2}{5}}\right)\right)}_{N\text{ terms}} \\
&+ \operatorname{diag}\left(\underbrace{O\left(d^{-\frac{1}{5}}\right), \ldots, O\left(d^{-\frac{1}{5}}\right)}_{N\text{ terms}}, \ldots, O\left(d^{-\frac{1}{5}}\right)\right) \\
&= \operatorname{diag}\left(\operatorname{tr}(H)d^{\frac{2}{5}} + O\left(d^{-\frac{1}{5}}\right), \ldots, \operatorname{tr}(H) + O\left(d^{-\frac{1}{5}}\right), \ldots,\right. \\
&\quad \left.\operatorname{tr}(H)d^{\frac{2}{5}} + O\left(d^{-\frac{1}{5}}\right), \ldots, O\left(d^{-\frac{1}{5}}\right)\right).
\end{aligned} \tag{8.153}$$

Following the proof process from Eqs. (8.118) to (8.121), the following inequalities can be found

$$\begin{aligned}
&\left|\lambda_{i,k}^{C'} - \lambda_k \mathbb{E}\left[X_i^{C'}\right]\right| \leq \|X_i^{C'} - \mathbb{E}\left[X_i^{C'}\right]\|_2 \leq O\left(d^{-\frac{1}{5}}\right) \\
&\operatorname{tr}(H)d^{\frac{2}{5}} - O\left(d^{-\frac{1}{5}}\right) \leq e_k^\intercal X_i^{C'} e_k \leq \operatorname{tr}(H)d^{\frac{2}{5}} + O\left(d^{-\frac{1}{5}}\right) \\
&\operatorname{tr}(H)d^{\frac{2}{5}} - O\left(d^{-\frac{1}{5}}\right) \leq \lambda_{i,1}^{C'} \leq \operatorname{tr}(H)d^{\frac{2}{5}} + O\left(d^{-\frac{1}{5}}\right).
\end{aligned} \tag{8.154}$$

However, unlike the previous proof, there exists a potential issue that the image transformation matrix H may lead to the case that $X_i^{C'}$ is not a square matrix. Then we denote $X_i^{C'} = \sum_{j=1}^{d} \lambda_{i,j} u_{i,j}^{C'} v_{i,j}^{C'}$, where $u_{i,j}^{C'}$ and $v_{i,j}^{C'}$ are left and right singular

vectors produced by SVD decomposition. So, we have

$$\begin{aligned} e_k^\intercal X_i^{C'} e_k &= \sum_{j=1}^{d} \lambda_{i,j} (e_k^\intercal u_{i,j}^{C'} v_{i,j}^{C'} e_k) \\ &\leq \lambda_{i,1}^{C'} \sum_{j=1}^{d} |e_k^\intercal u_{i,j}^{C'} v_{i,j}^{C'} e_k|, \end{aligned} \tag{8.155}$$

which further leads to

$$\begin{aligned} \sum_{j=1}^{d} |e_k^\intercal u_{i,j}^{C'} v_{i,j}^{C'} e_k| &\geq \frac{\mathrm{tr}(H) d^{\frac{2}{5}} - O\left(d^{-\frac{1}{5}}\right)}{\mathrm{tr}(H) d^{\frac{2}{5}} + O\left(d^{-\frac{1}{5}}\right)} \\ &= 1 - \frac{O\left(d^{-\frac{1}{5}}\right)}{\mathrm{tr}(H) d^{\frac{2}{5}} + O\left(d^{-\frac{1}{5}}\right)}. \end{aligned} \tag{8.156}$$

For global feature space. By similar augments from Eq. (8.125) to Eq. (8.131) and based on Eq.(8.156), for the global representation learned through the decentralised learning framework, we establish

$$\begin{aligned} \sum_{k=1}^{d} |e_k^\intercal \bar{u}_j^{C'} \bar{v}_j^{C'} e_k| &\geq \frac{\mathrm{tr}(H) \left(1 - \frac{1}{|\bar{A}|}\right) d^{\frac{2}{5}} - O\left(d^{-\frac{1}{5}}\right)}{\mathrm{tr}(H) \left(1 - \frac{1}{|\bar{A}|}\right) d^{\frac{2}{5}} + O\left(d^{-\frac{1}{5}}\right)} \\ = 1 - \frac{O\left(d^{-\frac{1}{5}}\right)}{\mathrm{tr}(H) \left(1 - \frac{1}{|\bar{A}|}\right) d^{\frac{2}{5}} + O\left(d^{-\frac{1}{5}}\right)}. \end{aligned} \tag{8.157}$$

On the other hand, for the global objective of the federated learning framework, we follow the arguments from Eq. (8.134) to Eq. (8.136) to derive

$$\begin{aligned} \sum_{j=1}^{d} |e_k^\intercal \bar{u}_j^{C'} \bar{v}_j^{C'} e_k| &\geq \frac{\mathrm{tr}(H) d^{\frac{2}{5}} - \Theta\left(d^{\frac{7}{20}}\right) - O\left(d^{-\frac{1}{5}}\right)}{\mathrm{tr}(H) d^{\frac{2}{5}} - \Theta\left(d^{\frac{7}{20}}\right) + O\left(d^{-\frac{1}{5}}\right)} \\ = 1 - \frac{O\left(d^{-\frac{1}{5}}\right)}{\mathrm{tr}(H) d^{\frac{2}{5}} - \Theta\left(d^{\frac{7}{20}}\right) + O\left(d^{-\frac{1}{5}}\right)}. \end{aligned} \tag{8.158}$$

Combining Lemma 13, Eq.(8.156), Eq.(8.157) and Eq.(8.158) completes the proof.

□

8.3.4 Proof of First Theoretical Insight

This section provides the full proof of Theorem 13 of the theoretical analysis in Section 5.4.2.

Proof. According to Theorem 11 and Theorem 12, we can find that the main difference between the global representations lies in the lower bound. For the global feature learned in the decentralised learning (DecL) framework, we denote the sensitivity of D-SSL as below:

$$s_{Dec}^{M} = \frac{O\left(d^{-\frac{2}{5}}\right)}{2p\left(1-p\right)\left(1-\frac{1}{\left|\bar{A}\right|}\right)d^{\frac{2}{5}} + O\left(d^{-\frac{2}{5}}\right)}, \tag{8.159}$$

$$s_{Dec}^{C_1} = \frac{O\left(d^{-\frac{1}{5}}\right)}{\left(1-\frac{1}{\left|\bar{A}\right|}\right)d^{\frac{2}{5}} - O\left(d^{-\frac{1}{5}}\right)}, \tag{8.160}$$

$$s_{Dec}^{C_2} = \frac{O\left(d^{-\frac{1}{5}}\right)}{\mathrm{tr}(H)\left(1-\frac{1}{\left|\bar{A}\right|}\right)d^{\frac{2}{5}} - O\left(d^{-\frac{1}{5}}\right)}, \tag{8.161}$$

where s_{Dec}^{M} represents the sensitivity of MIM-based D-SSL to heterogeneous data, $s_{Dec}^{C_1}$ represents the sensitivity of CL-based SSL with similar augmentations, and $s_{Dec}^{C_2}$ represents the sensitivity of CL-based SSL with dissimilar augmentations. Then, we

compare the magnitude of s_{Dec}^{M} and $s_{Dec}^{C_1}$ by solving the following equation:

$$
\begin{aligned}
s_{Dec}^{M} - s_{Dec}^{C_1} &= \frac{O\left(d^{-\frac{2}{5}}\right)}{\left(1-\frac{1}{|\bar{A}|}\right)d^{\frac{2}{5}} + O\left(d^{-\frac{2}{5}}\right)} - \frac{O\left(d^{-\frac{1}{5}}\right)}{\left(1-\frac{1}{|\bar{A}|}\right)d^{\frac{2}{5}} - O\left(d^{-\frac{1}{5}}\right)} \\
&= \frac{O\left(d^{-\frac{2}{5}}\right)\left(\left(1-\frac{1}{|\bar{A}|}\right)d^{\frac{2}{5}} - O\left(d^{-\frac{1}{5}}\right)\right) - O\left(d^{-\frac{1}{5}}\right)\left(\left(1-\frac{1}{|\bar{A}|}\right)d^{\frac{2}{5}} + O\left(d^{-\frac{2}{5}}\right)\right)}{\left(\left(1-\frac{1}{|\bar{A}|}\right)d^{\frac{2}{5}} - O\left(d^{-\frac{1}{5}}\right)\right)\left(\left(1-\frac{1}{|\bar{A}|}\right)d^{\frac{2}{5}} + O\left(d^{-\frac{2}{5}}\right)\right)}.
\end{aligned}
\tag{8.162}
$$

Consider the dimension d of the Euclidean space is very large so that $d \to \infty$. Then, we have

$$
\begin{aligned}
&\lim_{d\to\infty}[s_{Dec}^{M} - s_{Dec}^{C_1}] = \\
&\lim_{d\to\infty} \frac{O\left(d^{-\frac{2}{5}}\right)\left(\left(1-\frac{1}{|\bar{A}|}\right)d^{\frac{2}{5}} - O\left(d^{-\frac{1}{5}}\right)\right) - O\left(d^{-\frac{1}{5}}\right)\left(\left(1-\frac{1}{|\bar{A}|}\right)d^{\frac{2}{5}} + O\left(d^{-\frac{2}{5}}\right)\right)}{\left(\left(1-\frac{1}{|\bar{A}|}\right)d^{\frac{2}{5}} - O\left(d^{-\frac{1}{5}}\right)\right)\left(\left(1-\frac{1}{|\bar{A}|}\right)d^{\frac{2}{5}} + O\left(d^{-\frac{2}{5}}\right)\right)} \\
&= \lim_{d\to\infty} \frac{-\left(1-\frac{1}{|\bar{A}|}\right)O\left(d^{\frac{1}{5}}\right)}{\left(1-\frac{1}{|\bar{A}|}\right)^2 \Theta\left(d^{\frac{4}{5}}\right)}.
\end{aligned}
\tag{8.163}
$$

Due to the fact that $2 \leq \left|\bar{A}\right| \leq N$, we prove

$$
\lim_{d\to\infty}[s_{Dec}^{M} - s_{Dec}^{C_1}] < 0. \tag{8.164}
$$

Similarly, we determine if s_{Dec}^{M} is less than $s_{Dec}^{C_1}$ as follows

$$
\lim_{d\to\infty}[\frac{s_{Dec}^{C_2}}{s_{Dec}^{M}}] = \lim_{d\to\infty} \frac{\dfrac{O\left(d^{-\frac{1}{5}}\right)}{\mathrm{tr}(H)\left(1-\frac{1}{|\bar{A}|}\right)d^{\frac{2}{5}} + O\left(d^{-\frac{1}{5}}\right)}}{\dfrac{O\left(d^{-\frac{2}{5}}\right)}{2p(1-p)\left(1-\frac{1}{|\bar{A}|}\right)d^{\frac{2}{5}} + O\left(d^{-\frac{2}{5}}\right)}} = \frac{d^{-\frac{3}{5}}}{d^{-\frac{4}{5}}} = \infty, \tag{8.165}
$$

which implies

$$\lim_{d\to\infty}[s_{Dec}^{M}-s_{Dec}^{C_2}]<0. \tag{8.166}$$

Combining Eqs.(8.164) and (8.166) arrives

$$\lim_{d\to\infty}[s_{Dec}^{M}-s_{Dec}^{C}]<0, \tag{8.167}$$

where s_{Dec}^{C} denotes the sensitivity of CL-based SSL to heterogeneous data. On the other hand, for the federated learning (FL) framework, we denote the following sensitivity of D-SSL:

$$s_{Fed}^{M}=\frac{O\left(d^{-\frac{2}{5}}\right)}{2p\left(1-p\right)d^{\frac{2}{5}}-\Theta\left(d^{\frac{7}{20}}\right)+O\left(d^{-\frac{2}{5}}\right)}, \tag{8.168}$$

$$s_{Fed}^{C_1}=\frac{O\left(d^{-\frac{1}{5}}\right)}{d^{\frac{2}{5}}-\Theta\left(d^{\frac{7}{20}}\right)+O\left(d^{-\frac{1}{5}}\right)}, \tag{8.169}$$

$$s_{Fed}^{C_2}=\frac{O\left(d^{-\frac{1}{5}}\right)}{\mathrm{tr}(H)d^{\frac{2}{5}}-\Theta\left(d^{\frac{7}{20}}\right)+O\left(d^{-\frac{1}{5}}\right)}. \tag{8.170}$$

The difference between s_{Fed}^{M} and $s_{Fed}^{C_1}$ is given by

$$\begin{aligned}
s_{Fed}^{M}-s_{Fed}^{C_1}&=\frac{O\left(d^{-\frac{4}{5}}\right)}{2p\left(1-p\right)-\Theta\left(d^{-\frac{1}{20}}\right)+O\left(d^{-\frac{4}{5}}\right)}-\frac{O\left(d^{-\frac{3}{5}}\right)}{1-\Theta\left(d^{-\frac{1}{20}}\right)+O\left(d^{-\frac{3}{5}}\right)}\\
&=\frac{O\left(d^{-\frac{4}{5}}\right)\left(1-\Theta\left(d^{-\frac{1}{20}}\right)+O\left(d^{-\frac{3}{5}}\right)\right)-O\left(d^{-\frac{3}{5}}\right)\left(1-\Theta\left(d^{-\frac{1}{20}}\right)+O\left(d^{-\frac{4}{5}}\right)\right)}{\left(1-\Theta\left(d^{-\frac{1}{20}}\right)+O\left(d^{-\frac{4}{5}}\right)\right)\left(1-\Theta\left(d^{-\frac{1}{20}}\right)+O\left(d^{-\frac{3}{5}}\right)\right)}\\
&=\frac{-O\left(d^{-\frac{3}{5}}\right)+\Theta\left(d^{-\frac{13}{20}}\right)}{d^{\frac{1}{5}}-\Theta\left(d^{\frac{3}{20}}\right)+O\left(d^{-\frac{3}{5}}\right)}.
\end{aligned} \tag{8.171}$$

For the above result, let $d \to \infty$, we can establish

$$\lim_{d\to\infty}[s_{Fed}^{M} - s_{Fed}^{C_1}] = \lim_{d\to\infty} \frac{-O\left(d^{-\frac{3}{5}}\right) + \Theta\left(d^{-\frac{13}{20}}\right)}{d^{\frac{1}{5}} - \Theta\left(d^{\frac{3}{20}}\right) + O\left(d^{-\frac{3}{5}}\right)} = \lim_{d\to\infty} \frac{-O\left(d^{-\frac{3}{5}}\right)}{d^{\frac{1}{5}}} < 0 \tag{8.172}$$

Then, for the comparison between s_{Fed}^{M} and $s_{Fed}^{C_2}$, we have

$$\lim_{d\to\infty}[\frac{s_{Fed}^{C_2}}{s_{Fed}^{M}}] = \lim_{d\to\infty} \frac{\frac{O\left(d^{-\frac{1}{5}}\right)}{\mathrm{tr}(H)\, d^{\frac{2}{5}} - \Theta\left(d^{\frac{7}{20}}\right) + O\left(d^{-\frac{1}{5}}\right)}}{\frac{O\left(d^{-\frac{1}{5}}\right)}{d^{\frac{2}{5}} - \Theta\left(d^{\frac{7}{20}}\right) + O\left(d^{-\frac{1}{5}}\right)}} = \frac{d^{-\frac{3}{5}}}{d^{-\frac{4}{5}}} = \infty. \tag{8.173}$$

With Eqs.(8.172) and (8.173), we find

$$\lim_{d\to\infty}[s_{Fed}^{M} - s_{Fed}^{C}] < 0. \tag{8.174}$$

Combining Eq.(8.167) and Eq.(8.174) completes the proof.

□

8.3.5 Proof of Second Theoretical Insight

This section provides the full proof of Corollary 2 and Theorem 14 of the theoretical analysis in Section 5.4.3.

Proof. For the decentralised learning (DecL) framework, we notice from Eqs.(8.159), (8.160), and (8.161) that their denominators both include the term $1 - \frac{1}{|\bar{A}|}$. Since $|\bar{A}|$ is proportional to $1 - \frac{1}{|\bar{A}|}$, we derive that $|\bar{A}|$ is inversely proportional to s_{Dec}^{M}, $s_{Dec}^{C_1}$ and $s_{Dec}^{C_2}$, which completes the proof of Corollary 2. Next, by a similar proof from Eq.(8.159) to Eq.(8.174), we compare the robustness of distributed MIM between

the DecL and FL framework by solving

$$s_{Dec}^{M}-s_{Fed}^{M}=\frac{O\left(d^{-\frac{2}{5}}\right)}{2p\left(1-p\right)\left(1-\frac{1}{\left|\bar{A}\right|}\right)d^{\frac{2}{5}}+O\left(d^{-\frac{2}{5}}\right)}-\frac{O\left(d^{-\frac{2}{5}}\right)}{2p\left(1-p\right)d^{\frac{2}{5}}-\Theta\left(d^{\frac{7}{20}}\right)+O\left(d^{-\frac{2}{5}}\right)}. \tag{8.175}$$

This is equivalent to solving

$$\begin{aligned}&2p\left(1-p\right)\left(1-\frac{1}{\left|\bar{A}\right|}\right)d^{\frac{2}{5}}-\left(2p\left(1-p\right)d^{\frac{2}{5}}-\Theta\left(d^{\frac{7}{20}}\right)\right)\\&=2p\left(1-p\right)d^{\frac{2}{5}}-\frac{2p\left(1-p\right)}{\left|\bar{A}\right|}d^{\frac{2}{5}}-2p\left(1-p\right)d^{\frac{2}{5}}+\Theta\left(d^{\frac{7}{20}}\right).\end{aligned} \tag{8.176}$$

Due to the fact that

$$\lim_{d\to\infty}\left[2p\left(1-p\right)d^{\frac{2}{5}}-\frac{2p\left(1-p\right)}{\left|\bar{A}\right|}d^{\frac{2}{5}}-2p\left(1-p\right)d^{\frac{2}{5}}+\Theta\left(d^{\frac{7}{20}}\right)\right]<0, \tag{8.177}$$

we have

$$\lim_{d\to\infty}[s_{Dec}^{M}-s_{Fed}^{M}]>0. \tag{8.178}$$

Similarly, for CL-based SSL, we have

$$s_{Dec}^{C_1}-s_{Fed}^{C_1}=\frac{O\left(d^{-\frac{1}{5}}\right)}{\left(1-\frac{1}{\left|\bar{A}\right|}\right)d^{\frac{2}{5}}-O\left(d^{-\frac{1}{5}}\right)}-\frac{O\left(d^{-\frac{1}{5}}\right)}{d^{\frac{2}{5}}-\Theta\left(d^{\frac{7}{20}}\right)+O\left(d^{-\frac{1}{5}}\right)}, \tag{8.179}$$

$$s_{Dec}^{C_2}-s_{Fed}^{C_2}=\frac{O\left(d^{-\frac{1}{5}}\right)}{\mathrm{tr}(H)\left(1-\frac{1}{\left|\bar{A}\right|}\right)d^{\frac{2}{5}}-O\left(d^{-\frac{1}{5}}\right)}-\frac{O\left(d^{-\frac{1}{5}}\right)}{\mathrm{tr}(H)d^{\frac{2}{5}}-\Theta\left(d^{\frac{7}{20}}\right)+O\left(d^{-\frac{1}{5}}\right)}, \tag{8.180}$$

implying that

$$\lim_{d\to\infty}[s_{Dec}^{C_1}-s_{Fed}^{C_1}]>0, \tag{8.181}$$

$$\lim_{d\to\infty}[s_{Dec}^{C_2}-s_{Fed}^{C_2}]>0. \tag{8.182}$$

With Eqs.(8.181) and (8.182), we find

$$\lim_{d\to\infty}[s^C_{Dec} - s^C_{Fed}] > 0. \tag{8.183}$$

Combining Eq.(8.178) with Eq.(8.183) derives

$$\lim_{d\to\infty}[s_{Dec} > s_{Fed}]. \tag{8.184}$$

Note that Eq.(8.184) holds for decentralised learning setups in which each client has an inconsistent number of neighbours. However, there exists an optimal case, denoted by $\forall i, |A_i| = N$. In this case, the global objective of decentralised learning can be re-formulated as follows:

$$\sum_{i\in[N]} \frac{1}{N} \sum_{j\in[N]} \frac{1}{N} \mathcal{L} = \sum_{i\in[N]} \frac{1}{N} \mathcal{L}. \tag{8.185}$$

This equation is exactly the same as the global objective of federated learning shown in Eq.(5.1). Therefore, we know the following statement holds:

$$\lim_{d\to\infty}[s_{Dec} = s_{Fed}], \tag{8.186}$$

when $\forall i \in [N], |A_i| = N$. Combining Eq.(8.184) and Eq.(8.186) completes the proof. □

8.4 Proofs of Chapter 6

8.4.1 Proof of Convergence and Consensus Guarantees

This section provides the full proof of Theorems 15 and 16 of the theoretical analysis in Section 6.4.1.

At the beginning, we recall several relevant definitions for a finite-state Markov chain to facilitate understanding:

- **Transition Matrix.** Let $P \in \mathbb{R}^{\mathcal{V}\times\mathcal{V}}$ be a stochastic matrix i.e.,

$$P_{v,w} \geq 0 \quad \text{for all } v, w \in \mathcal{V}, \qquad \sum_{w\in\mathcal{V}} P_{v,w} = 1 \quad \text{for all } v \in \mathcal{V}.$$

  A (time-homogeneous) Markov chain on the finite state space $\mathcal{V}$ with transition matrix P is a stochastic process $\{W_t\}_{t\geq 0}$ such that, for any $t \geq 0$ and $v, w \in \mathcal{V}$,

$$\mathbb{P}(W_{t+1} = w \mid W_t = v, W_{t-1} = v_{t-1}, \ldots, W_0 = v_0) = \mathbb{P}(W_{t+1} = w \mid W_t = v) = P_{v,w}. \tag{8.187}$$

- **Irreducibility.** A Markov chain is irreducible if for any $v, w \in \mathcal{V}$, there exists an integer $t \geq 1$ such that $(\mathbb{P}^t)_{v,w} > 0$, i.e., every state can be reached from every other state in a finite number of steps.

- **Aperiodicity.** A Markov chain is aperiodic if there exists $t_0 > 0$ such that for all $t \geq t_0$ and $i, j \in \mathcal{V}$, $(\mathbb{P}^t)_{v,w} > 0$.

- **Stationary distribution.** An irreducible and aperiodic Markov chain admits a stationary distribution π such that $\pi\mathbb{P} = \pi$.

- **Mixing time.** The ϵ-mixing time is defined as

$$\tau_{mix}(\epsilon) = \min\Big\{t \geq 0 \ : \ \max_{\mu} \|\mu\mathbb{P}^t - \pi\|_{TV} \leq \epsilon\Big\},$$

where μ ranges over all initial distributions on $\mathcal{V}$ and $\|\cdot\|_{TV}$ denotes the total variation distance. In other words, the mixing time represents the number of steps of the Markov chain required for the distribution of the current state to be close to the stationary distribution π.

Next, we start our proof of the convergence and consensus guarantees of DeNAV by deriving some necessary lemmas.

Lemma 14. *For a decentralised scenario with n clients, if each client communicates with only one of their neighbours and aggregates their respective models, then the status of communication can be expressed as a doubly stochastic matrix $\mathbb{W}_t \sim \mathbb{W} \in \mathbb{R}^{n\times n}$ where $\mathbb{W}_{t,ij} > 0$ and $\mathbb{W}_{t,ji} > 0$ indicate that client i communicates with client j at iteration t. This lemma has been proved in [11].*

Corollary 4. *For a decentralised scenario with n clients, if $m \in [1, n]$ clients communicate with only one of their neighbours and aggregate their respective models at iteration t, then the communication matrix $\mathbb{W}_t \sim \mathbb{W} \in \mathbb{R}^{n\times n}$ is still a doubly stochastic matrix.*

Proof. When there is only a client initialises communication (client i communicates with client j), we are aware that

$$\begin{aligned} \mathbb{W}_{t,ij} + \mathbb{W}_{t,ji} &= 1 \\ \forall \mathbb{I}, \forall \mathbb{J} \notin \{i, j\} \text{ and } \forall \mathbb{J} \neq \mathbb{I}, \mathbb{W}_{t,\mathbb{IJ}} = 0, \mathbb{W}_{t,\mathbb{JI}} &= 0, \mathbb{W}_{t,\mathbb{II}} = 1 \\ \forall u, \forall v, \sum_u \mathbb{W}_{t,uv} = 1, \sum_v \mathbb{W}_{t,uv} &= 1. \end{aligned} \tag{8.188}$$

So the statement is true when $m = 1$ at iteration t. Then, if we assume that the statement is true for some arbitrary m, we have

$$\begin{aligned} \forall i, \forall j \in \{(i_1, j_1), \ldots, (i_m, j_m)\}, \mathbb{W}_{t,ij} + \mathbb{W}_{t,ji} &= 1 \\ \forall \mathbb{I}, \forall \mathbb{J} \notin \{(i_1, j_1), \ldots, (i_m, j_m)\} \text{ and } \forall \mathbb{I} \neq \mathbb{J}, \mathbb{W}_{t,\mathbb{IJ}} = 0, \mathbb{W}_{t,\mathbb{JI}} &= 0, \mathbb{W}_{t,\mathbb{II}} = 1 \\ \forall u, \forall v, \sum_u \mathbb{W}_{t,uv} = 1, \sum_v \mathbb{W}_{t,uv} &= 1. \end{aligned} \tag{8.189}$$

By combining equation (8.188) and (8.189), we have

$$\begin{aligned} \forall i, \forall j \in \{(i,j),(i_1,j_1),\ldots,(i_m,j_m)\}, \mathbb{W}_{t,ij} + \mathbb{W}_{t,ji} = 1 \\ \forall \mathbb{I}, \forall \mathbb{J} \notin \{(i,j),(i_1,j_1),\ldots,(i_m,j_m)\} \text{ and } \forall \mathbb{I} \neq \mathbb{J}, \mathbb{W}_{t,\mathbb{IJ}} = 0, \mathbb{W}_{t,\mathbb{JI}} = 0, \mathbb{W}_{t,\mathbb{II}} = 1 \\ \forall u, \forall v, \sum_u \mathbb{W}_{t,uv} = 1, \sum_v \mathbb{W}_{t,uv} = 1. \end{aligned} \tag{8.190}$$

Equation (8.190) implies that the statement also holds for the case of $m+1$. By mathematical induction, we prove that the statement is true for all integers $m \in [1, n]$. □

Lemma 15. *Let the routing process for a single model $\{S_t\}_{t=0}^T$ be defined as a sequence of states where*

$$S_t := (i_t, \psi_t). \tag{8.191}$$

Here, $i_t \in \{1,\ldots,n\}$ is the index of the current client, and $\psi_t \in \mathcal{H}$ is the training state log transmitted along with the model across clients, defined as a map from each client index $i \in \{1,\ldots,n\}$ to a tuple:

$$\psi_t(i) = (Z_t(i), L_t(i), \bar{t}_i^{(t)}), \tag{8.192}$$

where

- *$Z_t(i) \in \mathbb{N}$: the number of times client i has been selected up to time t;*
- *$L_t(i) \in \{-1, 0, \ldots, t-1\}$: the latest round when client i was selected (with -1 indicating never selected);*
- *$\bar{t}_i^t \in \mathbb{R}^+$: the most recent training time observed from client i until time t.*

Then, under the routing policy defined in our training navigator algorithm, the process $\{S_t\}_{t=0}^T$ satisfies the Markov property:

$$\mathbb{P}(S_{t+1} \mid S_t, S_{t-1}, \ldots, S_0) = \mathbb{P}(S_{t+1} \mid S_t) \quad \forall_t \in \{0,\ldots,T-1\}. \tag{8.193}$$

Proof. Let $\mathcal{C}_t = \text{neighbours}(i_t) \cup \{i_t\}$ be the candidate set of the next training client at time t. For each $i \in \mathcal{C}_t$, we follow our algorithm to define the selection score:

$$U_i^t = (U_i^d \times (\alpha_i + U_i^c) + 1)^{\mathbb{1}(Z(i)<Z)} \tag{8.194}$$

where

- $U_i^d = \frac{|\mathcal{D}_i|}{\max\{|\mathcal{D}_i||i\in\mathcal{C}_t\}}$ represents the data volume utility;
- $U_i^c = (\frac{\min\{\bar{t}_i^{(t)}|i\in\mathcal{C}_t\}}{\bar{t}_i})^{\mathbb{1}(L(i)>0)}$ represents the computational resource utility;
- $\alpha_i = (\frac{t-L(i)}{T})^{\mathbb{1}(L(i)>0)}$ represents the selection history factor.

Since local data volume on each client is typically unchanged throughout the training and the upper limit of the selection times Z and the total number of training rounds are pre-set constants, Eq.(8.194) shows that U_i^t depends only on the values $Z_t, L_t, \bar{t}_i$ in ψ_t. Therefore, according to the selection criteria of the next training client in the navigator algorithm:

$$i_{t+1} = \begin{cases} \arg\max\limits_{i\in\mathcal{C}_t} U_i^t & \text{if } \exists_i \in \mathcal{C}_t \quad \text{s.t.} Z(i) < Z, \\ \text{Uniform}(\mathcal{C}_t) & \text{otherwise,} \end{cases} \tag{8.195}$$

we conclude:

$$\mathbb{P}(i_{t+1} \mid S_t, S_{t-1}, \ldots, S_0) = \mathbb{P}(i_{t+1} \mid (i_t, \psi_t)) = \mathbb{P}(i_{t+1} \mid S_t). \tag{8.196}$$

Next, once the next training client i_{t+1} is selected and the local training on this client is finished, the new state log ψ_{t+1} is updated as follows:

$$\psi_{t+1}(i) = \begin{cases} (Z_t(i)+1, t+1, \bar{t}_i^{(t+1)}) & \text{if } i = i_{t+1}, \\ \psi_t(i) & \text{otherwise,} \end{cases} \tag{8.197}$$

for each client $i \in \{1, \ldots, n\}$. Here, $\bar{t}_i^{(t+1)}$ is the training time of client i observed in round $t+1$ and is collected from the local training consequence at timestamp $t+1$. Hence, we have:

$$\mathbb{P}(\psi_{t+1} \mid S_t, S_{t-1}, \ldots, S_0) = \mathbb{P}(\psi_{t+1} \mid i_{t+1}, S_t). \tag{8.198}$$

By combining Eq.(8.196) with Eq.(8.198), we derive:

$$\begin{aligned}
\mathbb{P}(S_{t+1} \mid S_t, S_{t-1}, \ldots, S_0) &= \mathbb{P}(i_{t+1} \mid S_t, S_{t-1}, \ldots, S_0) \cdot \mathbb{P}(\psi_{t+1} \mid S_t, S_{t-1}, \ldots, S_0) \\
&= \mathbb{P}(i_{t+1} \mid S_t) \cdot \mathbb{P}(\psi_{t+1} \mid i_{t+1}, S_t) \\
&= \mathbb{P}(\psi_{t+1}, i_{t+1} \mid S_t) \qquad \text{(by the chain rule)} \\
&= \mathbb{P}(S_{t+1} \mid S_t).
\end{aligned} \tag{8.199}$$

Thus, $\{S_t\}_{t=0}^{T}$ is a Markov chain, and the proof has been completed. □

Remark 14. *As noted in Lemma 15, although the routing policy relies on historical information such as selection frequency and recent training time, this information can be retrieved from the training state log ψ_t, which is transmitted with the model and updated each round. Therefore, the next routing decision only depends on the current state $S_t = (i_t, \psi_t)$, without reference to the full past trajectory $\{S_0, \ldots, S_t\}$. This follows the standard idea of state augmentation in controlled Markov chains and Markov decision processes: by encoding sufficient statistics of the history into the state, the overall process regains the Markov property [106, 119]. Hence, the communication state space $\{S_t\}_{t=0}^{T}$ generated by our navigator algorithm forms a time-homogeneous Markov chain.*

Lemma 16. *For any row vector sequence $\theta_t \in R^N$ defined as:*

$$\theta_t = \theta_{t-1} \mathbb{W}_{t-1}, \tag{8.200}$$

we have

$$\mathbb{E}_{s\ldots(t-1)}||\theta_t - \bar{\theta}_t \boldsymbol{1}_n^\intercal||^2 \leq (\rho^2 + C \cdot \lambda_2(P)^s)^{(t-s)}||\theta_s - \bar{\theta}_s \boldsymbol{1}_n^\intercal||^2, \tag{8.201}$$

where $\mathbb{W}_t$ is a doubly stochastic matrix, s is a time stamp earlier than t, $\bar{\theta}_t$ is the average value of vector θ_t, $\boldsymbol{1}_n$ is a full-one column vector, $||\cdot||$ is the l_2 norm, $0 < C \leq 1$ is the contraction constant and $\lambda_2(P)$ is the second largest eigenvalue of the transition matrix P.

Proof. Let $\mathbf{1}_n = [11\ldots11]^\intercal \in R^n$ denotes a full-one column vector and $||\cdot||$ represent the l_2 norm. Based on the property of doubly stochastic matrix $\theta\mathbb{W}\frac{\mathbf{1}_n\mathbf{1}_n^\intercal}{n} = \theta\frac{\mathbf{1}_n\mathbf{1}_n^\intercal}{n} = \theta\frac{\mathbf{1}_n\mathbf{1}_n^\intercal}{n}\mathbb{W}$, we derive

$$\begin{aligned}
\bar{\theta}_t\mathbf{1}_n^\intercal &= \theta_t\frac{\mathbf{1}_n\mathbf{1}_n^\intercal}{n} \\
&= \theta_{t-1}\mathbb{W}_{t-1}\frac{\mathbf{1}_n\mathbf{1}_n^\intercal}{n} \\
&= \theta_{t-1}\frac{\mathbf{1}_n\mathbf{1}_n^\intercal}{n}\mathbb{W}_{t-1} \\
&= \bar{\theta}_{t-1}\mathbf{1}_n^\intercal\mathbb{W}_{t-1}.
\end{aligned} \tag{8.202}$$

Therefore,

$$\begin{aligned}
y_t &= \theta_t - \bar{\theta}_t\mathbf{1}_n^\intercal \\
&= \theta_{t-1}\mathbb{W}_{t-1} - \bar{\theta}_{t-1}\mathbf{1}_n^\intercal\mathbb{W}_{t-1} \\
&= (\theta_{t-1} - \bar{\theta}_{t-1}\mathbf{1}_n^\intercal)\mathbb{W}_{t-1} \\
&= y_{t-1}\mathbb{W}_{t-1}.
\end{aligned} \tag{8.203}$$

Since $\{\mathbb{W}_t\}$ forms a Markov chain, we define the distribution of $\mathbb{W}_{t-1}$ as $v_{t-1} \in \mathcal{V}$. Then, according to the assumption that the chain is assumed to be irreducible and aperiodic, we have

$$v_{t-1}(\mathbb{W}) = \pi(\mathbb{W}) + \delta_{t-1}(\mathbb{W}), \tag{8.204}$$

where $\delta_{t-1}(\mathbb{W})$ is the deviation between the distribution v_{t-1} and the stationary distribution π. Therefore,

$$\begin{aligned}\mathbb{E}_{t-1}||y_t||^2 &= \sum_{\mathbb{W}} v_{t-1}(\mathbb{W})||y_{t-1}\mathbb{W}||^2 \\ &= \sum_{\mathbb{W}} \pi(\mathbb{W})||y_{t-1}\mathbb{W}||^2 + \sum_{\mathbb{W}} \delta_{t-1}(\mathbb{W})||y_{t-1}\mathbb{W}||^2 \\ &= \mathbb{E}_{\mathbb{W}\sim\pi}||y_{t-1}\mathbb{W}||^2 + \sum_{\mathbb{W}} \delta_{t-1}(\mathbb{W})||y_{t-1}\mathbb{W}||^2.\end{aligned} \tag{8.205}$$

Considering the i.i.d. property of π, we have

$$\begin{aligned}\mathbb{E}_{\mathbb{W}\sim\pi}||y_{t-1}\mathbb{W}||^2 &= \mathbb{E}_{\mathbb{W}\sim\pi}[(y_{t-1}\mathbb{W})(y_{t-1}\mathbb{W})^\intercal] \\ &= y_{t-1}\mathbb{E}_{\mathbb{W}\sim\pi}[\mathbb{W}\mathbb{W}^\intercal]y_{t-1}^\intercal \\ &= y_{t-1}\mathbb{E}_{\mathbb{W}\sim\pi}[\mathbb{W}^\intercal\mathbb{W}]y_{t-1}^\intercal.\end{aligned} \tag{8.206}$$

According to [11], $y_{t-1}\mathbb{E}[\mathbb{W}^\intercal\mathbb{W}]y_{t-1}^\intercal \leq \rho^2||y_{t-1}||^2$, where ρ is the second largest eigenvalue of $\mathbb{E}[\mathbb{W}^\intercal\mathbb{W}]$. Thus,

$$\mathbb{E}_{\mathbb{W}\sim\pi}||y_{t-1}\mathbb{W}||^2 \leq \rho^2||y_{t-1}||^2. \tag{8.207}$$

Next, for the right term in Eq.(8.205), based on [122], the below statement holds

$$\sum_{\mathbb{W}} \delta_{t-1}(\mathbb{W})||y_{t-1}\mathbb{W}||^2 \leq ||\delta_{t-1}||_{TV} \cdot \sup_{W\in\mathbb{W}} ||y_{t-1}W||^2 \leq (\lambda_2(P))^{t-1} \cdot \sup_{W\in\mathbb{W}} ||y_{t-1}W||^2, \tag{8.208}$$

where $\lambda_2(P)$ is the second largest eigenvalue of the transition matrix P, which verifies $0 \leq \lambda_2(P) < 1$. For any $t \geq \tau_{mix}(\epsilon)$, $(\lambda_2(P))^t \leq \epsilon$ further holds. Then, due to the property of the doubly stochastic matrix, it implies

$$\sup_{W\in\mathbb{W}} ||y_{t-1}W||^2 \leq C||y_{t-1}||^2, \tag{8.209}$$

where C is the contraction constant of doubly stochastic matrices and satisfies $0 < C \leq 1$. Thus, Eq.(8.208) turns into

$$\sum_{\mathbb{W}} \delta_{t-1}(\mathbb{W})||y_{t-1}\mathbb{W}||^2 \leq C \cdot (\lambda_2(P))^{t-1} \cdot ||y_{t-1}||^2. \tag{8.210}$$

By combining Eqs.(8.207) and (8.210), we have

$$\mathbb{E}_{t-1}||y_t||^2 \leq (\rho^2 + C \cdot \lambda_2(P)^{t-1})||y_s||^2. \tag{8.211}$$

Repeating Eq.(8.211) from timestamp s to t derives

$$\begin{aligned} \mathbb{E}_{s\ldots(t-1)}||y_t||^2 &\leq \left(\prod_{k=s}^{t-1}(\rho^2 + C \cdot \lambda_2(P)^k)\right)||y_s||^2 \\ &\leq (\rho^2 + C \cdot \lambda_2(P)^s)^{t-s}||y_s||^2. \end{aligned} \tag{8.212}$$

Substituting $y_t = \theta_t - \bar{\theta}_t \mathbf{1}_n^\intercal$ into equation (8.212) completes the proof. □

Remark 15. *Lemma 16 indicates that the error between θ_t and $\bar{\theta}_t \mathbf{1}_n^\intercal$ can converge to 0 at a rate governed by $\rho^2 + C \cdot \lambda_2(P)^s$. So, for a decentralised scenario where $m \in [0, n]$ clients communicate with only one of their neighbours and aggregate their respective models at each iteration, clients can provably attain consensus if $\rho^2 + C \cdot \lambda_2(P)^s < 1$.*

Lemma 17. *Given two non-negative sequences $\{a_t\}_{t=1}^{\infty}$ and $\{b_t\}_{t=1}^{\infty}$ that satisfying*

$$a_t = \sum_{s=1}^{t} \rho^{(t-s)} b_s \tag{8.213}$$

with $\rho \in [0, 1)$, we have

$$\begin{aligned} S_k &:= \sum_{t=1}^{k} a_t \leq \sum_{s=1}^{k} a_t \frac{b_s}{1-\rho} \\ D_k &:= \sum_{t=1}^{k} a_t^2 \leq \frac{1}{(1-\rho)^2} \sum_{s=1}^{k} b_s^2, \end{aligned} \tag{8.214}$$

which has been proved in the appendix of [129].

Lemma 18. *Under the above assumptions, if Θ_t is iteratively updated by Eq.(6.4), then we have*

$$\begin{aligned}
&\sum_{t=1}^{T}\sum_{i=1}^{n}\mathbb{E}\|\theta_t^{(i)}-\bar{\Theta}_t\boldsymbol{1}_n^\intercal\|^2\\
&\leq\frac{2}{1-(\rho^2+C)}||\Theta_0-\bar{\Theta}_0\boldsymbol{1}_n^\intercal||_F^2+\frac{2}{(1-\sqrt{(\rho^2+C\lambda_2(P))})^2}\sum_{t=1}^{T}\mathbb{E}||\eta_t G(\Theta_t;\xi_t))||_F^2,
\end{aligned}\tag{8.215}$$

where $\bar{\Theta}=\Theta\frac{\boldsymbol{1}_n}{n}$.

Proof. From equation (6.4), we have

$$\begin{aligned}
&\sum_{i=1}^{n}\mathbb{E}||\theta_t^{(i)}-\bar{\Theta}_t\mathbf{1}_n^\intercal||^2\\
&=\sum_{i=1}^{n}\mathbb{E}||\Theta_t e_n^{(i)}-\bar{\Theta}_t\mathbf{1}_n^\intercal||^2\\
&=\mathbb{E}||\Theta_t-\Theta_t\frac{\mathbf{1}_n}{n}\mathbf{1}_n^\intercal||_F^2\\
&=\mathbb{E}||\Theta_t(I-\frac{\mathbf{1}_n}{n}\mathbf{1}_n^\intercal)||_F^2\\
&=\sum_{j=1}^{N}\mathbb{E}||e_N^{(j)}\Theta_t(I-\frac{\mathbf{1}_n}{n}\mathbf{1}_n^\intercal)||_F^2\\
&=\sum_{j=1}^{N}\mathbb{E}||e_N^{(j)}(\Theta_0\prod_{s=0}^{t-1}\mathbb{W}_s)(I-\frac{\mathbf{1}_n}{n}\mathbf{1}_n^\intercal)-e_N^{(j)}(\sum_{s=0}^{t-1}\eta_s G(\Theta_s;\xi_s)\prod_{r=s+1}^{t-1}\mathbb{W}_r)(I-\frac{\mathbf{1}_n}{n}\mathbf{1}_n^\intercal)||_F^2\\
&=\sum_{j=1}^{N}\mathbb{E}||\theta_0^{[j]}\prod_{s=0}^{t-1}\mathbb{W}_s(I-\frac{\mathbf{1}_n}{n}\mathbf{1}_n^\intercal)-\sum_{s=0}^{t-1}\eta_s G^{[j]}(\Theta_s;\xi_s)\prod_{r=s+1}^{t-1}\mathbb{W}_r(I-\frac{\mathbf{1}_n}{n}\mathbf{1}_n^\intercal)||^2\\
&\leq 2\sum_{j=1}^{N}(\underbrace{\mathbb{E}||\theta_0^{[j]}\prod_{s=0}^{t-1}\mathbb{W}_s(I-\frac{\mathbf{1}_n}{n}\mathbf{1}_n^\intercal)||^2}_{\mathbf{Q1}}+\underbrace{\mathbb{E}||\sum_{s=0}^{t-1}H_s^{[j]}||^2}_{\mathbf{Q2}}),\qquad\text{(Parallelogram Law)}
\end{aligned}\tag{8.216}$$

where $H_s^{[j]}=\eta_s G^{[j]}(\Theta_s;\xi_s)\prod_{r=s+1}^{t-1}\mathbb{W}_r(I-\frac{\mathbf{1}_n}{n}\mathbf{1}_n^\intercal)$. By using Lemma 16 to bound Q1, we have

$$\mathbb{E}\|\theta_0^{[j]}\prod_{s=0}^{t-1}\mathbb{W}_s(I-\tfrac{\mathbf{1}_n}{n}\mathbf{1}_n^\intercal)\|^2\leq(\rho^2+C)^t\|\theta_0^{[j]}-\theta_0^{[j]}\tfrac{\mathbf{1}_n}{n}\mathbf{1}_n^\intercal\|^2.\tag{8.217}$$

For Q2, we have

$$\begin{aligned}\mathbb{E}||\sum_{s=0}^{t-1} H_s^{[j]}||^2 &= \sum_{s=0}^{t-1}\mathbb{E}||H_s^{[j]}||^2 + 2\sum_{s<z}^{t-1}\langle H_s^{[j]}, H_z^{[j]}\rangle \\ &\leq \sum_{s=0}^{t-1}\mathbb{E}||H_s^{[j]}||^2 + 2\sum_{s<z}^{t-1}\mathbb{E}||H_s^{[j]}||||H_z^{[j]}||.\end{aligned} \tag{8.218}$$

We can again bound $\mathbb{E}||H_s^{[j]}||^2$ using Lemma 16. So, we have

$$\begin{aligned}\mathbb{E}\|H_s^{[j]}\|^2 &= \mathbb{E}||\eta_s G^{[j]}(\Theta_s;\xi_s)\prod_{r=s+1}^{t-1}\mathbb{W}_r(I-\frac{\mathbf{1}_n\mathbf{1}_n^\intercal}{n})||^2 \\ &= \mathbb{E}||\eta_s G^{[j]}(\Theta_s;\xi_s) - \eta_s G^{[j]}(\Theta_s;\xi_s)\frac{\mathbf{1}_n\mathbf{1}_n^\intercal}{n})||(\rho^2 + C\lambda_2(P)^{s+1})^{t-s-1} \\ &\leq (\rho^2 + C\lambda_2(P)^{s+1})^{t-s-1}\mathbb{E}\|\eta_s G^{[j]}(\Theta_s;\xi_s)\|^2.\end{aligned} \tag{8.219}$$

Then we bound $\mathbb{E}||H_s^{[j]}||||H_z^{[j]}||$, i.e.,

$$\begin{aligned}\mathbb{E}||H_s^{[j]}||||H_z^{[j]}|| &= \mathbb{E}||\eta_s G^{[j]}(\Theta_s;\xi_s)(I-\frac{\mathbf{1}_n\mathbf{1}_n^\intercal}{n})||||\eta_z G^{[j]}(\Theta_z;\xi_z)(I-\frac{\mathbf{1}_n\mathbf{1}_n^\intercal}{n})|| \\ &\leq (\rho^2 + C\lambda_2(P)^{s+1})^{(t-s-1)/2}\mathbb{E}||\eta_{s-1}G^{[j]}(\Theta_{s-1};\xi_{s-1})|| \\ &\quad\cdot(\rho^2 + C\lambda_2(P)^{z+1})^{(t-z-1)/2}||\eta_{z-1}G^{[j]}(\Theta_{z-1};\xi_{z-1})||.\end{aligned} \tag{8.220}$$

Combining (8.218), (8.219), and (8.220), we can bound Q2 as

$$\begin{aligned}&\mathbb{E}||\sum_{s=0}^{t-1}\eta_s G^{[j]}(\Theta_s;\xi_s)\prod_{r=s+1}^{t-1}\mathbb{W}_r(I-\frac{\mathbf{1}_n}{n}\mathbf{1}_n^\intercal)||^2 \\ \leq &\sum_{s=0}^{t-1}(\rho^2 + C\lambda_2(P)^{s+1})^{(t-s-1)}\mathbb{E}||\eta_{s-1}G^{[j]}(\Theta_{s-1};\xi_{s-1})||^2 \\ &+ 2\sum_{s<z}^{t-1}(\rho^2 + C\lambda_2(P)^{s+1})^{(t-s-1)/2}\mathbb{E}||\eta_{s-1}G^{[j]}(\Theta_{s-1};\xi_{s-1})|| \\ &\quad\cdot(\rho^2 + C\lambda_2(P)^{z+1})^{(t-z-1)/2}||\eta_{z-1}G^{[j]}(\Theta_{z-1};\xi_{z-1})|| \\ \leq &\sum_{s=0}^{t-1}(\rho^2 + C\lambda_2(P)^{s+1})^{(t-s-1)/2}\mathbb{E}||\eta_s G^{[j]}(\Theta_s;\xi_s)||^2.\end{aligned} \tag{8.221}$$

Combining (8.216), (8.217), and (8.221), we have

$$\begin{aligned}
&\sum_{i=1}^{n}\mathbb{E}||\theta_t^{(i)}-\bar{\Theta}_t\mathbf{1}_n^\intercal||^2\\
&\leq 2\sum_{j=1}^{N}\left((\rho^2+C)^t||\theta_0^{[j]}-\theta_0^{[j]}\frac{\mathbf{1}_n\mathbf{1}_n^\intercal}{n}||^2)\right)\\
&\quad+2\sum_{j=1}^{N}\left((\sum_{s=0}^{t-1}(\rho^2+C\lambda_2(P)^{s+1})^{(t-s-1)/2}\mathbb{E}||\eta_s G^{[j]}(\Theta_s;\xi_s)||)^2\right).
\end{aligned}\tag{8.222}$$

By using Lemma 17, the sum of (8.222) from $t=1$ to $t=T$ can be expressed as

$$\begin{aligned}
&\sum_{t=1}^{T}\sum_{i=1}^{n}\mathbb{E}||\theta_t^{(i)}-\bar{\Theta}_t\mathbf{1}_n^\intercal||^2\\
&\leq 2\sum_{j=1}^{N}\sum_{t=1}^{T}\left((\rho^2+C)^t||\theta_0^{[j]}-\theta_0^{[j]}\frac{\mathbf{1}_n\mathbf{1}_n^\intercal}{n}||^2\right)\\
&\quad+2\sum_{j=1}^{N}\sum_{t=1}^{T}\left(\sum_{s=0}^{t-1}(\rho^2+C\lambda_2(P)^{s-1})^{(t-s-1)/2}\mathbb{E}||\eta_s G^{[j]}(\Theta_s;\xi_s)||^2\right)\\
&\leq \frac{2}{1-(\rho^2+C)}\sum_{j=1}^{N}||\theta_0^{[j]}-\theta_0^{[j]}\frac{\mathbf{1}_n\mathbf{1}_n^\intercal}{n}||^2+\frac{2}{(1-\sqrt{\rho^2+C\lambda_2(P)})^2}\sum_{j=1}^{N}\sum_{t=1}^{T}\mathbb{E}||\eta_t G^{[j]}(\Theta_t;\xi_t)||^2\\
&\leq \frac{2}{1-(\rho^2+C)}||\Theta_0-\bar{\Theta}_0\mathbf{1}_n^\intercal||_F^2+\frac{2}{(1-\sqrt{\rho^2+C\lambda_2(P)})^2}\sum_{t=1}^{T}\mathbb{E}||\eta_t G(\Theta_t;\xi_t)||_F^2,
\end{aligned}\tag{8.223}$$

which completes the proof of Lemma 18. □

Remark 16. *Note that if we make the initial models of all clients the same (i.e., $||\Theta_0-\bar{\Theta}_0\mathbf{1}_n^\intercal||_F^2=0$), then the consensus is only impacted by the gradients (i.e., $||G(\Theta_t;\xi_t)||_F^2$), showing that our method DeNAV can attain consensus if the training converges.*

Then we start to prove the convergence of DeNAV.

Lemma 19. *Following the above assumptions, we have*

$$\frac{\eta_t}{2}\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2 + \frac{\eta_t - 2L\eta_t^2}{2}\mathbb{E}||\overline{\nabla f}(\Theta_t)||^2 \leq \mathbb{E}f(\bar{\Theta}_t) - \mathbb{E}f(\bar{\Theta}_{t+1}) + \frac{L^2\eta_t}{2n}\sum_{i=1}^{n}\mathbb{E}||\theta_t^{(i)} - \bar{\Theta}_t||^2 + \frac{L\eta_t^2\sigma^2}{n}, \tag{8.224}$$

where $\nabla f(\bar{\Theta}_t) = \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\frac{1}{n}\sum_{i=1}^{n}\theta^{(i)})$ *and* $\overline{\nabla f}(\Theta_t) = \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\theta^{(i)})$.

Proof. To help our proof, we first define:

$$\begin{aligned}\nabla f(\bar{\Theta}_t) &= \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\frac{1}{n}\sum_{i=1}^{n}\theta^{(i)}) \\ \overline{\nabla f}(\Theta) &= \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\theta^{(i)}).\end{aligned} \tag{8.225}$$

According to Eq.(6.4), we have

$$\begin{aligned}\bar{\Theta}_{t+1} &= \Theta_{t+1}\frac{\mathbf{1}^n}{n} \\ &= (\Theta_t\mathbb{W}_t - \eta_t G(\Theta_t;\xi_t))\frac{\mathbf{1}^n}{n} \\ &= \Theta_t\frac{\mathbf{1}^n}{n} - \eta_t G(\Theta_t;\xi_t)\frac{\mathbf{1}^n}{n} \\ &= \bar{\Theta}_t - \eta_t\bar{G}(\Theta_t;\xi_t).\end{aligned} \tag{8.226}$$

According to the Lipschitzian condition for the objective function f_i and f, we have

$$
\begin{aligned}
&\mathbb{E}f(\bar{\Theta}_{t+1}) \\
&\leq \mathbb{E}f(\bar{\Theta}_t) + \mathbb{E}\langle \nabla f(\bar{\Theta}_t), -\eta_t \bar{G}(\Theta_t;\xi_t)\rangle + \frac{L}{2}\mathbb{E}|| - \eta_t \bar{G}(\Theta_t;\xi_t))||^2 \\
&= \mathbb{E}f(\bar{\Theta}_t) - \eta_t \langle \mathbb{E}\nabla f(\bar{\Theta}_t), \mathbb{E}_{\xi_t}\bar{G}(\Theta_t;\xi_t)\rangle + \frac{L\eta_t^2}{2}\mathbb{E}||(\bar{G}(\Theta_t;\xi_t)) - \overline{\nabla f}(\Theta_t)) + \overline{\nabla f}(\Theta_t)||^2 \\
&\leq \mathbb{E}f(\bar{\Theta}_t) - \eta_t \mathbb{E}\langle \nabla f(\bar{\Theta}_t), \overline{\nabla f}(\Theta_t)\rangle + L\eta_t^2 \mathbb{E}||\bar{G}(\Theta_t;\xi_t) - \overline{\nabla f}(\Theta_t)||^2 + L\eta_t^2\mathbb{E}||\overline{\nabla f}(\Theta_t)||^2 \\
&= \mathbb{E}f(\bar{\Theta}_t) - \eta_t \mathbb{E}\langle \nabla f(\bar{\Theta}_t), \overline{\nabla f}(\Theta_t)\rangle \\
&\quad + \frac{L\eta_t^2}{n}\sum_{i=1}^{n}\mathbb{E}||\nabla F_i(\theta_t;\xi_t) - \overline{\nabla f_i}(\theta_t^{(i)})||^2 + L\eta_t^2\mathbb{E}||\overline{\nabla f}(\Theta_t)||^2 \\
&\leq \mathbb{E}f(\bar{\Theta}_t) + \frac{\eta_t}{2}\left(\mathbb{E}||\nabla f(\bar{\Theta}_t) - \overline{\nabla f}(\Theta_t)||^2 - \mathbb{E}||\nabla f(\bar{\Theta}_t)||^2 - \mathbb{E}||\overline{\nabla f}(\Theta_t||^2\right) \\
&\quad + \frac{L\eta_t^2\sigma^2}{n} + L\eta_t^2\mathbb{E}||\overline{\nabla f}(v_t)||^2 \\
&= \mathbb{E}f(\bar{\Theta}_t) + \frac{\eta_t}{2}\mathbb{E}||\nabla f(\bar{\Theta}_t) - \overline{\nabla f}(\Theta_t)||^2 - \frac{\eta_t}{2}\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2 \\
&\quad - \frac{\eta_t - 2L\eta_t^2}{2}\mathbb{E}||\overline{\nabla f}(\Theta_t||^2 + \frac{L\eta_t^2\sigma^2}{n}.
\end{aligned}
\tag{8.227}
$$

$\mathbb{E}||\nabla f(\bar{\Theta}_t) - \overline{\nabla f}(\Theta_t)||^2$ can be again bounded by Lipschitzian condition, which is described as

$$
\begin{aligned}
\mathbb{E}||\nabla f(\bar{\Theta}_t) - \overline{\nabla f}(\Theta_t)||^2 &= \frac{1}{n}\mathbb{E}||\sum_{i=1}^{n}\nabla f_i(\bar{\Theta}_t) - \overline{\nabla f_i}(\theta_t)||^2 \\
&\leq \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}||\nabla f_i(\bar{\Theta}_t) - \overline{\nabla f_i}(\theta_t^{(i)})||^2 \\
&\leq \frac{L^2}{n}\sum_{i=1}^{n}\mathbb{E}||\theta_t^{(i)} - \bar{\Theta}_t||^2.
\end{aligned}
\tag{8.228}
$$

Combining (8.227) and (8.228) and rearranging, we have

$$\begin{aligned}
&\mathbb{E}f(\bar{\Theta}_{t+1}) \leq \mathbb{E}f(\bar{\Theta}_t) + \frac{L^2\eta_t}{2n}\sum_{i=1}^{n}\mathbb{E}||\theta_t^{(i)} - \bar{\Theta}_t||^2 \\
&- \frac{\eta_t}{2}\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2 - \frac{\eta_t - 2L\eta_t^2}{2}\mathbb{E}||\overline{\nabla f}(\Theta_t)||^2 + \frac{L\eta_t^2\sigma^2}{n} \\
&\frac{\eta_t}{2}\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2 + \frac{\eta_t - 2L\eta_t^2}{2}\mathbb{E}||\overline{\nabla f}(\Theta_t)||^2 \leq \mathbb{E}f(\bar{\Theta}_t) - \mathbb{E}f(\bar{\Theta}_{t+1}) \\
&\quad + \frac{L^2\eta_t}{2n}\sum_{i=1}^{n}\mathbb{E}||\theta_t^{(i)} - \bar{\Theta}_t||^2 + \frac{L\eta_t^2\sigma^2}{n},
\end{aligned} \tag{8.229}$$

which completes the proof. □

Lemma 20. *Under the above assumptions, we can bound* $||G(\Theta_t;\xi_t)||_F^2$ *as follows*

$$\mathbb{E}||G(\Theta_t;\xi_t)||_F^2 \leq n\sigma^2 + 4L^2\sum_{i=1}^{n}\mathbb{E}||\theta_t^{(i)} - \bar{\Theta}_t||^2 + 8n\zeta^2 + 8n\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2. \tag{8.230}$$

Proof. By rearranging the equation, we have

$$\begin{aligned}
&\mathbb{E}||G(\Theta_t;\xi_t)||_F^2 \\
&= \sum_{i=1}^{n}\mathbb{E}||\nabla F_i(\theta_t;\xi_t)||_F^2 \\
&= \sum_{i=1}^{n}\mathbb{E}||\left(\nabla F_i(\theta_t;\xi_t) - \nabla f_i(\theta_t^{(i)})\right) + \nabla f_i(\theta_t^{(i)})||^2 \\
&\leq 2\sum_{i=1}^{n}\mathbb{E}||\nabla F_i(\theta_t;\xi_t) - \nabla f_i(\theta_t^{(i)})||^2 + 2\sum_{i=1}^{n}\mathbb{E}||\nabla f_i(\theta_t^{(i)})||^2 \\
&\leq n\sigma^2 + 2\sum_{i=1}^{n}\mathbb{E}||(\nabla f_i(\theta_t^{(i)}) - \nabla f_i(\bar{\Theta}_t)) + \nabla f_i(\bar{\Theta}_t)||^2 \\
&\leq n\sigma^2 + 4\sum_{i=1}^{n}\mathbb{E}||\nabla f_i(\theta_t^{(i)}) - \nabla f_i(\bar{\Theta}_t)||^2 + 4\sum_{i=1}^{n}\mathbb{E}||\nabla f_i(\bar{\Theta}_t)||^2 \\
&= n\sigma^2 + 4\sum_{i=1}^{n}\mathbb{E}||\nabla f_i(\theta_t^{(i)}) - \nabla f_i(\bar{\Theta}_t)||^2 + 4\sum_{i=1}^{n}\mathbb{E}||(\nabla f_i(\bar{\Theta}_t) - \nabla f(\bar{\Theta}_t)) + \nabla f(\bar{\Theta}_t)||^2 \\
&\leq n\sigma^2 + 4L^2\sum_{i=1}^{n}\mathbb{E}||\theta_t^{(i)} - \bar{\Theta}_t||^2 + 8n\zeta^2 + 8n\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2,
\end{aligned} \tag{8.231}$$

which completes the proof. □

Lemma 21. *Under the above assumptions, we have*

$$\begin{aligned}&\sum_{t=1}^{T}(1-4D_2L^2\eta_t^2)\sum_{i=1}^{n}\mathbb{E}||\theta_t^{(i)}-\bar{\Theta}_t||^2\\&\leq D_1||\Theta_0-\bar{\Theta}_0\boldsymbol{1}_n^\intercal||_F^2+D_2n(\sigma^2+8\zeta^2)\sum_{t=1}^{T}\eta_t^2+8D_2n\sum_{t=1}^{T}\eta_t^2\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2,\end{aligned}\tag{8.232}$$

where $D_1=\frac{2}{1-(\rho^2+C)}$, $D_2=\frac{2}{(1-\sqrt{\rho^2+C\lambda_2(P)})^2}$ *and* $0<C\leq 1$ *is the contraction constant of doubly stochastic matrices.*

Proof. Substituting Lemma 20 into Lemma 18, we have

$$\begin{aligned}\sum_{t=1}^{T}\sum_{i=1}^{n}\mathbb{E}||\theta_t^{(i)}-\bar{\Theta}_t||^2&\leq D_1||\Theta_0-\bar{\Theta}_0\mathbf{1}_n^\intercal||_F^2+D_2\sum_{t=1}^{T}\eta_t^2\mathbb{E}||G(\Theta_t;\xi_t)||_F^2\\&\leq D_1||\Theta_0-\bar{\Theta}_0\mathbf{1}_n^\intercal||_F^2+D_2\sum_{t=1}^{T}\eta_t^2(n\sigma^2\\&\quad+4L^2\sum_{i=1}^{n}\mathbb{E}||\theta_t^{(i)}-\bar{\Theta}_t||^2+8n\zeta^2+8n\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2)\\&=D_1||\Theta_0-\bar{\Theta}_0\mathbf{1}_n^\intercal||_F^2+D_2n(\sigma^2+8\zeta^2)\sum_{t=1}^{T}\eta_t^2\\&\quad+4D_2L^2\sum_{t=1}^{T}\eta_t^2\sum_{i=1}^{n}\mathbb{E}||\theta_t^{(i)}-\bar{\Theta}_t||^2+8D_2n\sum_{t=1}^{T}\eta_t^2\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2.\end{aligned}\tag{8.233}$$

Rearranging the above equation, we have

$$\begin{aligned}&\sum_{t=1}^{T}(1-4D_2L^2\eta_t^2)\sum_{i=1}^{n}\mathbb{E}||\theta_t^{(i)}-\bar{\Theta}_t||^2\\&\leq D_1||\Theta_0-\bar{\Theta}_0\mathbf{1}_n^\intercal||_F^2+D_2n(\sigma^2+8\zeta^2)\sum_{t=1}^{T}\eta_t^2+8D_2n\sum_{t=1}^{T}\eta_t^2\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2,\end{aligned}\tag{8.234}$$

which completes the proof. □

Remark 17. *According to Lemma 21, if* $1 - 4D_2L^2\eta_t^2 > 0$, *then* $\mathbb{E}||\theta_t^{(i)} - \bar{\Theta}_t||^2$ *is bounded.*

Lemma 22. *For a decentralised scenario with* n *clients, if each step of training only involves* $m \in [1, n]$ *clients, then*

$$\mathbb{E}_{|m|}||\nabla f(\bar{\Theta}_t)||^2 \approx \frac{n}{m}\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2 \tag{8.235}$$

where $\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2$ *is the expected loss gradient of the average model at pre-training step* t *for the case where all* n *clients perform local training at each step, and* $\mathbb{E}_{|m|}||\nabla f(\bar{\Theta}_t)||^2$ *is the expected loss gradient of the average model at pre-training step* t *for the case where a subset of clients perform local training at each step and the size of the subset is* m.

Proof. Following the definition, we have

$$\begin{aligned}
\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2 &= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}||\nabla f(\theta_t^{(i)})||^2 \\
\mathbb{E}_{|m|}||\nabla f(\bar{\Theta}_t)||^2 &= \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}||\nabla f(\theta_t^{(i)})||^2,
\end{aligned} \tag{8.236}$$

where $\mathbb{E}||\nabla f(\theta_t^{(i)})||^2$ is the expected loss gradients at client i. Assume these gradient estimates $\mathbb{E}||\nabla f(\theta_t^{(i)})||^2$ are independent and identically distributed (i.e., i.i.d) and let each with mean μ and finite variance σ^2. According to the central limit theorem and the law of large numbers, when m and n becomes large, both $\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}||\nabla f(\theta_t^{(i)})||^2$ and $\frac{1}{m}\sum_{i=1}^{m}\mathbb{E}||\nabla f(\theta_t^{(i)})||^2$ will approach a normal distribution $\mathcal{N}(\mu, \sigma^2)$. Therefore, the variance for the first case can be expressed as:

$$\begin{aligned}
\mathrm{Var}(\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2) &= \mathrm{Var}(\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}||\nabla f(\theta_t^{(i)})||^2) \\
&= \frac{1}{n^2}\sum_{i=1}^{n}\mathrm{Var}(\mathbb{E}||\nabla f(\theta_t^{(i)})||^2) = \frac{1}{n^2}\cdot n\sigma^2 = \frac{\sigma^2}{n}.
\end{aligned} \tag{8.237}$$

We can also formulate the variance for the second case as:

$$\begin{aligned}\mathrm{Var}(\mathbb{E}_{|m|}||\nabla f(\bar{\Theta}_t)||^2) &= \mathrm{Var}(\frac{1}{m}\sum_{i=1}^{m}\mathbb{E}||\nabla f(\theta_t^{(i)})||^2) \\ &= \frac{1}{m^2}\sum_{i=1}^{m}\mathrm{Var}(\mathbb{E}||\nabla f(\theta_t^{(i)})||^2) = \frac{1}{m^2}\cdot m\sigma^2 = \frac{\sigma^2}{m}\end{aligned} \tag{8.238}$$

Thus, we have

$$\frac{\mathrm{Var}(\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2)}{\mathrm{Var}(\mathbb{E}_{|m|}||\nabla f(\bar{\Theta}_t)||^2)} = \frac{\sigma^2}{n}\cdot\frac{m}{\sigma^2} = \frac{m}{n}. \tag{8.239}$$

The above equation shows the relationship between the variance of two cases. Therefore, we can further express the relationship between $\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2$ and $\mathbb{E}_{|m|}||\nabla f(\bar{\Theta}_t)||^2$ as

$$\mathbb{E}_{|m|}||\nabla f(\bar{\Theta}_t)||^2 \approx \frac{n}{m}\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2, \tag{8.240}$$

which completes the proof. □

Lemma 23. *Under the above assumptions, if η_t is fixed as η and satisfies $1 - 4D_2L^2\eta^2 > 0$ for DeNAV, then*

$$\begin{aligned}&\frac{1-12D_2L^2\eta^2}{1-4D_2L^2\eta^2}\sum_{t=1}^{T}\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2 + \sum_{t=1}^{T}(1-4L\eta)\mathbb{E}||\overline{\nabla f}(\Theta_t)||^2 \\ \leq\ &\frac{2}{\eta}(\mathbb{E}f(\bar{\Theta}_0) - f^*) + \frac{L^2D_1}{n(1-4D_2L^2\eta^2)}|\Theta_0 - \bar{\Theta}_0\boldsymbol{1}_n^\intercal||_F^2 \\ &+ \left(\frac{L^2D_2T\eta^2}{1-4D_2L^2\eta^2} + \frac{2LT\eta}{n}\right)\sigma^2 + \frac{8L^2D_2\zeta^2T\eta^2}{1-4D_2L^2\eta^2}.\end{aligned} \tag{8.241}$$

Proof. According to Lemma 19, we have

$$\begin{aligned}\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2 + (1-4L\eta_t)\mathbb{E}||\overline{\nabla f}(\Theta_t)||^2 \leq\ &\frac{2}{\eta_t}(\mathbb{E}f(\bar{\Theta}_{t-1}) - f^* - (\mathbb{E}f(\bar{\Theta}_t) - f^*)) \\ &+ \frac{L^2}{n}\sum_{i=1}^{n}\mathbb{E}||\theta_t^{(i)} - \bar{\Theta}_t||^2 + \frac{2L\eta_t\sigma^2}{n}.\end{aligned} \tag{8.242}$$

According to Lemma 21, if we satisfy $1 - 4D_2L^2\eta^2 > 0$ with fixing η_t as η and sum both sides of (8.242) from $t = 1$ to $t = T$, we obtain

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbb{E}||\nabla f(\bar{\Theta}_t)||^2 + \sum_{t=1}^{T}(1 - 4L\eta)\mathbb{E}||\overline{\nabla f}(\Theta_t)||^2 \\
&\leq \frac{2}{\eta}(\mathbb{E}f(\bar{\Theta}_0) - f^* - (\mathbb{E}f(\bar{\Theta}_T) - f^*)) + \frac{L^2}{n}\sum_{t=1}^{T}\sum_{i=1}^{n}\mathbb{E}||\theta_t^{(i)} - \bar{\Theta}_t||^2 + \frac{2LT\eta\sigma^2}{n} \\
&\leq \frac{2}{\eta}(\mathbb{E}f(\bar{\Theta}_0) - f^*) + \frac{L^2}{n}(\frac{D_1||\Theta_0 - \bar{\Theta}_0\mathbf{1}_n^\intercal||_F^2}{1 - 4D_2L^2\eta^2} + \frac{D_2n(\sigma^2 + 8\zeta^2)T\eta^2}{1 - 4D_2L^2\eta^2} \\
&\quad + \frac{8D_2n\eta^2}{1 - 4D_2L^2\eta^2}\sum_{t=1}^{T}\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2) + \frac{2LT\eta\sigma^2}{n}.
\end{aligned} \tag{8.243}$$

Rearranging the above equation, we have

$$\begin{aligned}
&\frac{1 - 12D_2L^2\eta^2}{1 - 4D_2L^2\eta^2}\sum_{t=1}^{T}\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2 + \sum_{t=1}^{T}(1 - 4L\eta)\mathbb{E}||\overline{\nabla f}(\Theta_t)||^2 \\
&\leq \frac{2}{\eta}(\mathbb{E}f(\bar{\Theta}_0) - f^*) + \frac{L^2D_1||\Theta_0 - \bar{\Theta}_0\mathbf{1}_n^\intercal||_F^2}{n(1 - 4D_2L^2\eta^2)} + \frac{L^2D_2\sigma^2T\eta^2}{1 - 4D_2L^2\eta^2} + \frac{8L^2D_2\zeta^2T\eta^2}{1 - 4D_2L^2\eta^2} + \frac{2LT\eta\sigma^2}{n} \\
&= \frac{2}{\eta}(\mathbb{E}f(\bar{\Theta}_0) - f^*) + \frac{L^2D_1}{n(1 - 4D_2L^2\eta^2)}||\Theta_0 - \bar{\Theta}_0\mathbf{1}_n^\intercal||_F^2 \\
&\quad + \left(\frac{L^2D_2T\eta^2}{1 - 4D_2L^2\eta^2} + \frac{2LT\eta}{n}\right)\sigma^2 + \frac{8L^2D_2\zeta^2T\eta^2}{1 - 4D_2L^2\eta^2},
\end{aligned} \tag{8.244}$$

which completes the proof. □

Remark 18. *This lemma provides DeNAV with a convergence guarantee.*

Based on the above results, we can complete the proof of Theorem 15 as follows.

Proof. For $\eta = \frac{1}{4L\sqrt{D_2} + \frac{\sqrt{T}}{\sqrt{n}}}$, it satisfies

$$1 - 4L\eta > 0 \tag{8.245}$$

$$4D_2L^2\eta^2 \leq \frac{1}{4} \tag{8.246}$$

$$\frac{1 - 12D_2L^2\eta^2}{1 - 4D_2L^2\eta^2} \geq \frac{1}{3}. \tag{8.247}$$

Then we can set $\frac{1-12D_2L^2\eta^2}{1-4D_2L^2\eta^2}=\frac{1}{3}$, and remove $(1-4L\eta)\mathbb{E}||\overline{\nabla f}(\Theta_t)||^2$ by substituting $\eta=\frac{1}{4L\sqrt{D_2}+\frac{\sqrt{T}}{\sqrt{n}}}$ into Eq.(8.244) because η is small enough. So we have

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2 &\leq (\frac{6}{T\eta}(\mathbb{E}f(\bar{\Theta}_0)-f^*)+\frac{3L^2D_1}{nT(1-4D_2L^2\eta^2)}||\Theta_0-\bar{\Theta}_0\mathbf{1}_n^\intercal||_F^2\\ &\quad+(\frac{3L^2D_2\eta^2}{1-4D_2L^2\eta^2}+\frac{6L\eta}{n})\sigma^2+\frac{24L^2D_2\zeta^2\eta^2}{1-4D_2L^2\eta^2}\\ \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}||\nabla f(\bar{\Theta}_t)||^2 &\leq (\frac{6}{4LT\sqrt{D_2}+\frac{T^{\frac{3}{2}}}{\sqrt{n}}})(\mathbb{E}f(\bar{\Theta}_0)-f^*)+\frac{4L^2D_1}{nT}||\Theta_0-\bar{\Theta}_0\mathbf{1}_n^\intercal||_F^2\\ &\quad+\frac{4L^2D_2\sigma^2+32L^2D_2\zeta^2}{16L^2D_2+\frac{8L\sqrt{D_2T}}{\sqrt{n}}+\frac{T}{n}}+\frac{6L\sigma^2}{4L\sqrt{D_2}n+\sqrt{nT}}.\end{aligned}\tag{8.248}$$

By combining Lemma 22 with Eq.(8.248), we further find

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{|m|}||\nabla f(\bar{\Theta}_t)||^2 &\leq (\frac{6n}{4mLT\sqrt{D_2}+\frac{mT^{\frac{3}{2}}}{\sqrt{n}}})(\mathbb{E}f(\bar{\Theta}_0)-f^*)+\frac{4L^2D_1}{mT}||\Theta_0-\bar{\Theta}_0\mathbf{1}_n^\intercal||_F^2\\ &\quad+\frac{4nL^2D_2\sigma^2+32nL^2D_2\zeta^2}{16mL^2D_2+\frac{8mL\sqrt{D_2T}}{\sqrt{n}}+\frac{mT}{n}}+\frac{6nL\sigma^2}{4L\sqrt{D_2}mn+m\sqrt{nT}},\end{aligned}\tag{8.249}$$

which means

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{|m|}||\nabla f(\bar{\Theta}_t)||^2 &\lesssim \frac{n}{\sqrt{D_2}mT+\frac{mT^{\frac{3}{2}}}{\sqrt{n}}}+\frac{D_1}{mT}||\Theta_0-\bar{\Theta}_0\mathbf{1}_n^\intercal||_F^2\\ &\quad+\frac{D_2n\sigma^2+D_2n\zeta^2}{D_2m+\frac{m\sqrt{D_2T}}{\sqrt{n}}+\frac{mT}{n}}+\frac{n\sigma^2}{\sqrt{D_2}mn+m\sqrt{nT}}.\end{aligned}\tag{8.250}$$

Thus completing the proof. □

Then, Theorem 16 can also be established.

Proof. Combining Lemma 21 with Eq.(8.250) and substituting $\eta = \frac{1}{4L\sqrt{D_2}+\frac{\sqrt{T}}{\sqrt{n}}}$, we obtain

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n}\mathbb{E}||\theta_t^{(i)}-\bar{\Theta}_t||^2 \\
&\leq \frac{4}{3T}D_1||\Theta_0-\bar{\Theta}_0\mathbf{1}_n^\intercal||_F^2+\frac{4D_2n(\sigma^2+8\zeta^2)\eta^2}{3}+\frac{32D_2n\eta^2}{3}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{|m|}||\nabla f(\bar{\Theta}_t)||^2 \\
&\lesssim \frac{D_1}{T}||\Theta_0-\bar{\Theta}_0\mathbf{1}_n^\intercal||_F^2+\frac{D_2n(\sigma^2+\zeta^2)}{D_2+\frac{\sqrt{D_2T}}{\sqrt{n}}+\frac{T}{n}}+\frac{D_2n}{D_2+\frac{\sqrt{D_2T}}{\sqrt{n}}+\frac{T}{n}}(\frac{n}{\sqrt{D_2}mT+\frac{mT^{\frac{3}{2}}}{\sqrt{n}}} \\
&\quad+\frac{D_1}{mT}||\Theta_0-\bar{\Theta}_0\mathbf{1}_n^\intercal||_F^2+\frac{D_2n\sigma^2+D_2n\zeta^2}{D_2m+\frac{m\sqrt{D_2T}}{\sqrt{n}}+\frac{mT}{n}}+\frac{n\sigma^2}{\sqrt{D_2}mn+m\sqrt{nT}}).
\end{aligned} \tag{8.251}$$

The proof has been completed. □

8.4.2 Proof of Impact of Local Data Volume on DeNAV Training

This section provides the full proof of Theorem 17 and Corollary 3 of the theoretical analysis in Section 6.4.2.

From the architecture perspective, in DeNAV, the encoder and decoder of a one-block masked autoencoder contain only a single vision transformer block, which is mainly comprised of the self-attention module and a two-layer feed-forward network with ReLU activation [22, 141]. The self-attention function can be formulated as follows:

$$\text{Attention}(Q,K,V)=\text{softmax}(\frac{QK^T}{\sqrt{d}})V. \tag{8.252}$$

Previous literature [95] has proved through Tucker decomposition [138] that the outputs of the self-attention function can be represented by a linear combination of a set of basis vectors, expressed as the following lemma:

Lemma 24. *(See [95], Theorem 3.1) Let $e_1,\ldots,e_n$ be basis vectors from the vector space S. Assume that these vectors $e_1,\ldots,e_n$ are linearly independent and Q,K,V can be linearly represented by this set of basis vectors. The output of the attention function in Eq.(8.252) can be represented by a linear combination of the set of these*

basis vectors.

$$Attention(Q, K, V) = (e_1, \ldots, e_n)M \tag{8.253}$$

where $M \in \mathbb{R}^{n\times d}$ *is a coefficient matrix, and* d *is the dimension of these matrices (i.e.* Q*,* K*, and* V*).*

On the other hand, there was also proof that a shallow ReLU network can be replaced by a deep network with linear activation [159], which is described as follows:

Lemma 25. *(See [159], Proposition 1) Let* $\rho : \mathbb{R} \to \mathbb{R}$ *be any continuous piece-wise linear function with* M *breakpoints, where* $1 \leq M < \infty$*.*
(a) Let ξ *be a network with the activation function* ρ*, having depth* L*,* W *weights, and* U *computation units. Then there exists a ReLU network* η *that has depth* L*, not more than* $(M+1)^2W$ *weights and not more than* $(M+1)U$ *units, and that computes the same function as* ξ*.*
(b) Conversely, let η *be a ReLU network of depth* L *with* W *weights and* U *computation units. Let* $\mathcal{D}$ *be a bounded subset of* $\mathbb{R}^n$*, where* n *is the input dimension of* η*. Then there exists a network with the activation function* ρ *that has depth* L*,* $4W$ *weights and* $2U$ *units, and that computes the same function as* η *on the set* $\mathcal{D}$*.*

Based on Lemma 24 and Lemma 25, we can expect the error bound between the actual computation of the one-block masked autoencoder and its linear approximation to be very small and derive the following proposition.

Proposition 1. *There exists a linear equivalent mapping with* W_h *to the approximate transformer encoder* $h(\cdot)$ *and a linear equivalent mapping with* W_g *to the approximate transformer decoder* $g(\cdot)$*.*

Proof. Expanding the nonlinear vector function $h(x)$ into a Taylor series at 0, we have

$$h(x) = h(0) + \nabla_x h(0)x + \epsilon, \tag{8.254}$$

where $\nabla_x h(0)$ denotes the gradient of operator $h(\cdot)$ at 0 in the direction of the vector x, and ϵ a higher order infinitesimal residual. According to Lemma 24 and Lemma

25, we recognise that the residual ϵ is expected to be very small in the current case, resulting in a limited effect on the output $h(x)$. Therefore, by neglecting the residual ϵ and letting $\nabla_x h(0)x = W_h x$, we get

$$h(x) \approx W_h x + h(0) \tag{8.255}$$

As $h(0) \to 0$, $h(x)$ can be represented by the mapping W_h. Likewise, by neglecting the residual and letting $\nabla_x g(0)x = W_g x$, we get

$$g(x) \approx W_g x + g(0). \tag{8.256}$$

Since $g(0) = 0$, $g(x)$ can be represented by the mapping W_g, thus completing the proof. □

With the above proposition, the local training on the client i can be formulated as $\hat{x} = g_i(h_i(\tilde{x})) \approx W_{g_i} W_{h_i} \tilde{x} = W_i x$. Then, we start to prove Theorem 17 below.

Proof. The local data on client i can be formulated as X_i, and the corrupted input to the model on client i can be formulated as $\tilde{X}_i$. Both X_i and $\tilde{X}_i$ have the same size. Then, the transformed loss function of the training with the aggregation can be formulated as

$$\begin{aligned} l(X_i, \hat{X}_i) &= l(X_i, g_i(h_i(\tilde{X}_i))) \\ &= \frac{1}{2}\|X_i - W_i\tilde{X}_i\|^2 \\ &= \frac{1}{2}\mathrm{tr}\left[(X_i - W_i\tilde{X}_i)(X_i - W_i\tilde{X}_i)^\intercal\right]. \end{aligned} \tag{8.257}$$

This loss function is a convex function that can reach a minimum value when its derivative is 0. Therefore, with $\nabla_{W_i} l(X_i, \hat{X}_i) = 0$, it yields

$$\begin{aligned} 2\nabla_{W_i} l(X_i, \hat{X}_i) &= \nabla_{W_i} \mathrm{tr}\left[(X_i - W_i\tilde{X}_i)(X_i - W_i\tilde{X}_i)^\intercal\right] \\ &= \nabla_{W_i} \mathrm{tr}(X_i^\intercal X_i - \tilde{X}_i^\intercal W_i^\intercal X_i - X_i^\intercal W_i \tilde{X}_i + \tilde{X}_i^\intercal W_i^\intercal W_i \tilde{X}_i) \\ &= \nabla_{W_i} \mathrm{tr}(W_i \tilde{X}_i \tilde{X}_i^\intercal W_i^\intercal) - 2\nabla_{W_i} \mathrm{tr}(W_i X_i \tilde{X}_i^\intercal) \\ &= 2W_i \tilde{X}_i \tilde{X}_i^\intercal - 2X_i \tilde{X}_i^\intercal = 0. \end{aligned} \tag{8.258}$$

Solving Eq.(8.258) yields $W_i^* = X_i \tilde{X}_i^\intercal (\tilde{X}_i \tilde{X}_i^\intercal)^{-1}$. Next, if we aggregate the input and the ground-truth data over the selected clients $\mathbb{C}_t$ for each step, we have

$$\tilde{\mathbb{X}}_t = [\tilde{X}_i | i \in \mathbb{C}_t]^\intercal, \quad \mathbb{X}_t = [X_i | i \in \mathbb{C}_t]^\intercal. \tag{8.259}$$

Since the data size varies from client to client, it is necessary to append 0 to the empty space of $\tilde{\mathbb{X}}_t$ and $\mathbb{X}_t$. Finally, by defining $W_A^* = [W_i^* | i \in \mathbb{C}_t]^\intercal$, the approximate optimal solution for the model W_A can be represented by $\tilde{\mathbb{X}}_t$ and $\mathbb{X}_t$, with the following form:

$$W_A^* = \mathbb{X}_t \tilde{\mathbb{X}}_t^T (\tilde{\mathbb{X}}_t \tilde{\mathbb{X}}_t^T)^{-1}. \tag{8.260}$$

The proof has been completed. □

According to the formulation in Theorem 17, the proof of Corollary 3 can be easily completed as follows.

Proof. We first define $\mathbf{x} = \max\{|X_i| \mid i \in \mathbb{C}_t\}$. Expanding $\tilde{\mathbb{X}}_t$ and $\mathbb{X}_t$ gives:

$$\tilde{\mathbb{X}}_t = \begin{bmatrix} \tilde{x}_{1,1} & \dots & \tilde{x}_{1,m} \\ \vdots & \tilde{x}_{u,v} & \vdots \\ \tilde{x}_{\mathbf{x},1} & \dots & \tilde{x}_{\mathbf{x},m} \end{bmatrix}_{\mathbf{x}\times m}, \quad \mathbb{X}_t = \begin{bmatrix} x_{1,1} & \dots & x_{1,m} \\ \vdots & x_{u,v} & \vdots \\ x_{\mathbf{x},1} & \dots & x_{\mathbf{x},m} \end{bmatrix}_{\mathbf{x}\times m} \tag{8.261}$$

where

$$\tilde{\mathbb{X}}_t^{u,v}, \mathbb{X}_t^{u,v} = \begin{cases} 0 & \text{if } u > |X_i| \\ \tilde{x}_i, x_i & \text{otherwise} \end{cases} \tag{8.262}$$

Eq.(8.261) shows that the size of $\tilde{\mathbb{X}}_t$ and $\mathbb{X}_t$ will increase if the selected clients have large data volumes. According to Theorem 17, we identify that our selection strategy will make the approximate optimal solution W_A^* have a larger size and be less sparse, leading to a more reliable estimation of the ground-truth mapping from input to output. Therefore, the proof has been completed. □

Theorem 17 and Corollary 3 establish the connection between the training effectiveness of the one-block MAE and the local data volume on the training clients under the view that the encoder and decoder can be well approximated by linear mappings. However, this justification primarily applies to a single pre-training step. Since pre-training proceeds over multiple iterations, it is crucial to examine whether the linear approximation remains valid across rounds. In particular, the key challenge lies in analysing the behaviour of the higher-order residual term ϵ introduced in Proposition 1. To ensure that these residuals do not accumulate and undermine the approximation, we provide the following additional lemma.

Lemma 26. *Let $h(\cdot)$ and $g(\cdot)$ denote the encoder and decoder of the one-block MAE, expanded at the zero reference point as in Proposition 1. Then, when training with mean square error loss for T steps, and using gradient descent with learning rate $\eta = O(\frac{1}{T})$ following a standard linear decay, the cumulative higher-order residual ϵ_T satisfies*

$$\|\epsilon_T\| \leq O(1). \tag{8.263}$$

Proof. Recall Proposition 1, the nonlinear encoding function $h(x)$ can be expanded into a Taylor series at 0 as follows

$$h(X_t) \approx h(0) + \nabla h(0) X_t + \epsilon_h(X_t), \tag{8.264}$$

where X_t is the input data at time stamp t, $\epsilon_h(X_t) = \frac{1}{2}X_t^\intercal \nabla^2 h(\sigma_t) X_t$ is the second-order Taylor residual and $\sigma_t \in (0, X_t)$. Here, for tractability, we represent the formulation of higher-order residuals by the second-order derivatives, which is the largest term among them. Assuming that there exists a bound for the norm of second-order derivatives, i.e., $\forall_\sigma, ||\nabla^2 h(\sigma)|| \leq \kappa_h$, we have

$$||\epsilon_h(X_t)|| \leq \frac{\kappa_h}{2}||X_t||^2. \tag{8.265}$$

Similarly, for the decoder part, the Taylor expansion of the non-linear decoding function at 0 is shown below

$$g(z) \approx g(0) + \nabla g(0)z + \epsilon_g(z), \tag{8.266}$$

and we have

$$||\epsilon_g(z)|| \leq \frac{\kappa_g}{2}||z||^2. \tag{8.267}$$

For the reconstruction at the pre-training step t, we define $\hat{X}_t = g(h(X_t))$. Substituting Eqs.(8.264) and (8.266) into it and considering $h(0) = 0, g(0) = 0$, we derive

$$\begin{aligned} \hat{X}_t &= W_g(W_h X_t + \epsilon_h(X_t)) + \epsilon_g(h(X_t)) \\ &= W_g W_h X_t + W_g \epsilon_h(X_t) + \epsilon_g(h(X_t)) \end{aligned} \tag{8.268}$$

where $W_h := \nabla h(0)$ and $W_g := \nabla g(0)$. Thus, the residual term R_t is denoted as

$$R_t = W_g \epsilon_h(X_t) + \epsilon_g(h(X_t)). \tag{8.269}$$

We start to bound this term step by step. For the first term, we have

$$||W_g \epsilon_h(X_t)|| \leq ||W_g|| \cdot ||\epsilon_h(X_t)|| \leq ||W_g|| \cdot \frac{\kappa_h}{2}||X_t||^2. \tag{8.270}$$

For the second term, note that $h(X_t) = W_h X_t + \epsilon_h(X_t)$, it implies

$$||h(X_t)|| \leq ||W_h|| \cdot ||X_t|| + ||\epsilon_h(X_t)|| \leq ||W_h|| \cdot ||X_t|| + \frac{\kappa_h}{2}||X_t||^2. \tag{8.271}$$

By substituting it into Eq.(8.267), we have

$$\begin{aligned} ||\epsilon_g(h(X_t))|| &\leq \frac{\kappa_g}{2}||h(X_t)||^2 \\ &\leq \frac{\kappa_g}{2}\left(||W_h|| \cdot ||X_t|| + \frac{\kappa_h}{2}||X_t||^2\right)^2. \end{aligned} \tag{8.272}$$

Combining Eq.(8.270) and Eq.(8.272) establishes the following bound for R_t:

$$\begin{aligned} ||R_t|| &\leq \frac{\kappa_h}{2}||W_g|| \cdot ||X_t||^2 + \frac{\kappa_g}{2}\left(||W_h|| \cdot ||X_t|| + \frac{\kappa_h}{2}||X_t||^2\right)^2 \\ &= (\frac{\kappa_h}{2}||W_g|| + \frac{\kappa_g}{2}||W_h||^2) \cdot ||X_t||^2 + \frac{\kappa_g\kappa_h}{2}||W_h|| \cdot ||X_t||^3 + \frac{\kappa_g\kappa_h}{8}||X_t||^4. \end{aligned} \tag{8.273}$$

A simplification of this bound is

$$|R_t|| \leq \alpha||X_t||^2 + \beta||X_t||^3 + \gamma||X_t||^4, \tag{8.274}$$

where $\alpha = \frac{\kappa_h}{2}||W_g|| + \frac{\kappa_g}{2}||W_h||^2$, $\beta = \frac{\kappa_g\kappa_h}{2}||W_h||$ and $\gamma = \frac{\kappa_g\kappa_h}{8}$. Then, let $W_{gh} = W_g W_h$, the mean square error (MSE) loss used for training step t is defined as

$$L_t = \frac{1}{2}||X_t - \hat{X}_t||^2 = \frac{1}{2}||X_t - W_{gh}X_t - R_t||, \tag{8.275}$$

and the gradient of L_t is

$$\begin{aligned} \nabla_\theta L_t &= -(X_t - W_{gh}X_t - R_t)X_t^\intercal \\ &= \underbrace{-(X_t - W_{gh}X_t)X_t^\intercal}_{G_t} + \underbrace{R_t X_t^\intercal}_{\Delta_t}, \end{aligned} \tag{8.276}$$

where G_t is the ideal loss gradient and Δ_t is the gradient deviation led by the higher-order Taylor residuals. Next, given the learning rate η, the training update at

step t follows

$$\theta_t = \theta_{t-1} - \eta \nabla_\theta L_t = \theta_{t-1} - \eta G_t - \eta \Delta_t. \tag{8.277}$$

By iterating it from $t = 1$ to the final training step T, we have

$$\theta_T = \theta_1 - \eta \sum_{t=1}^{T} G_t - \eta \sum_{t=1}^{T} \Delta_t = \theta_1 - \eta \sum_{t=1}^{T} G_t - \eta \sum_{t=1}^{T} (R_t X_t^\intercal). \tag{8.278}$$

Let $\epsilon_T = \eta \sum_{t=1}^{T} (R_t X_t^\intercal)$, the norm bound of the cumulative deviation term ϵ_T is

$$||\epsilon_T|| \leq \eta \sum_{t=1}^{T} (||R_t|| \cdot ||X_t||). \tag{8.279}$$

Substituting Eq.(8.274) into it derives

$$\begin{aligned} ||\epsilon_T|| &\leq \eta \sum_{t=1}^{T} \left(\alpha ||X_t||^2 + \beta ||X_t||^3 + \gamma ||X_t||^4\right) ||X_t|| \\ &= \eta(\alpha \sum_{t=1}^{T} ||X_t||^3 + \beta \sum_{t=1}^{T} ||X_t||^4 + \gamma \sum_{t=1}^{T} ||X_t||^5). \end{aligned} \tag{8.280}$$

Assuming that the inputs satisfy $\forall_t, |||X_t|| \leq B$ (which often holds if we employ normalisation to transform the input image into [0,1] range), we further have

$$||\epsilon_T|| \leq \eta T(\alpha B^3 + \beta B^4 + \gamma B^5). \tag{8.281}$$

Since the learning rate η follows a standard linear decay (which is also common in real training scenarios), denoted as $\eta = O(\frac{1}{T})$, then the above bound will turn into

$$||\epsilon_T|| \leq O(1), \tag{8.282}$$

showing that the higher-order Taylor residuals do not accumulate with the training steps and implying that the linear approximation assumed in Theorem 17 and Corollary 3 is not undermined. The proof has been completed. □

BIBLIOGRAPHY

[1] Sawsan AbdulRahman, Hanine Tout, Hakima Ould-Slimane, Azzam Mourad, Chamseddine Talhi, and Mohsen Guizani. A survey on federated learning: The journey from centralized to distributed on-site learning and beyond. *IEEE Internet of Things Journal*, 8(7):5476–5497, 2020.

[2] Idan Achituve, Aviv Shamsian, Aviv Navon, Gal Chechik, and Ethan Fetaya. Personalized federated learning with gaussian processes. In *Advances in Neural Information Processing Systems*, volume 34, pages 8392–8406, 2021.

[3] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

[4] Ghadir Ayache and Salim El Rouayheb. Random walk gradient descent for decentralized learning on graphs. In *2019 IEEE International Parallel and Distributed Processing Symposium Workshops*, pages 926–931. IEEE, 2019.

[5] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. In *International Conference on Learning Representations*, 2022.

[6] Enrique Tomás Martínez Beltrán, Mario Quiles Pérez, Pedro Miguel Sánchez Sánchez, Sergio López Bernal, Gérôme Bovet, Manuel Gil Pérez, Gregorio Martínez Pérez, and Alberto Huertas Celdrán. Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges. *IEEE Communications Surveys & Tutorials*, 2023.

[7] Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*, 2022.

[8] Mahrokh Ghoddousi Boroujeni, Andreas Krause, and Giancarlo Ferrari Trecate. Personalized federated learning of probabilistic models: A pac-bayesian approach. *arXiv preprint arXiv:2401.08351*, 2024.

[9] Léon Bottou. On-line learning and stochastic approximations. In David Saad, editor, *On-line Learning in Neural Networks*, pages 9–42. Cambridge University Press, 1998.

[10] Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

[11] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE transactions on Information Theory*, 52(6):2508–2530, 2006.

[12] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020.

[13] Chaochao Chen, Xiaohua Feng, Jun Zhou, Jianwei Yin, and Xiaolin Zheng. Federated large language model: A position paper. *arXiv preprint arXiv:2307.08925*, 2023.

[14] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, volume 119, pages 1597–1607. PMLR, 2020.

[15] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[16] Ching-Hsiang Chu, Xiaoyi Lu, Ammar A Awan, Hari Subramoni, Jahanzeb Hashmi, Bracy Elton, and Dhabaleswar K Panda. Efficient and scalable multi-source streaming broadcast on gpu clusters for deep learning. In *2017 46th International Conference on Parallel Processing*, pages 161–170. IEEE, 2017.

[17] Norah L Crossnohere, Mohamed Elsaid, Jonathan Paskett, Seuli Bose-Brill, and John FP Bridges. Guidelines for artificial intelligence in medicine: literature review and content analysis of frameworks. *Journal of Medical Internet Research*, 24(8):e36823, 2022.

[18] Atish Das Sarma, Danupon Nanongkai, Gopal Pandurangan, and Prasad Tetali. Distributed random walks. *Journal of the ACM*, 60(1):1–31, 2013.

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[21] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1422–1430, 2015.

[22] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[23] Georgios Drainakis, Panagiotis Pantazopoulos, Konstantinos V Katsaros, Vasilis Sourlas, Angelos Amditis, and Dimitra I Kaklamani. From centralized to federated learning: Exploring performance and end-to-end resource consumption. *Computer Networks*, 225:109657, 2023.

[24] Weinan E. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 5(1):1–11, 2017.

[25] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[26] Andre Elisseeff, Theodoros Evgeniou, Massimiliano Pontil, and Leslie Pack Kaelbing. Stability of randomized learning algorithms. *Journal of Machine Learning Research*, 6(1), 2005.

[27] Omar Elnakib, Eman Shaaban, Mohamed Mahmoud, and Karim Emara. Evaluation of centralized, distributed and federated learning for iot intrusion detection systems. In *2023 Eleventh International Conference on Intelligent Computing and Information Systems*, pages 315–320. IEEE, 2023.

[28] Paul Erdős and Alfréd Rényi. On random graphs. i. *Publicationes Mathematicae*, 6:290–297, 1959.

[29] Mathieu Even. Stochastic gradient descent under markovian sampling schemes. In *International Conference on Machine Learning*, volume 202, pages 9412–9439. PMLR, 2023.

[30] Samir Yitzhak Gadre, Georgios Smyrnis, Vaishaal Shankar, Suchin Gururangan, Mitchell Wortsman, Rulin Shao, Jean Mercat, Alex Fang, Jeffrey Li, Sedrick Keh, et al. Language models scale reliably with over-training and on downstream tasks. In *International Conference on Learning Representations*, 2025.

[31] Mingming Gong, Kun Zhang, Tongliang Liu, Dacheng Tao, Clark Glymour, and Bernhard Schölkopf. Domain adaptation with conditional transferable components. In *International Conference on Machine Learning*, volume 48, pages 2839–2848. PMLR, 2016.

[32] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, pages 2672–2680, 2014.

[33] Rémi Gosselin, Loïc Vieu, Faiza Loukil, and Alexandre Benoit. Privacy and security in federated learning: A survey. *Applied Sciences*, 12(19):9901, 2022.

[34] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[35] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284, 2020.

[36] Jie Gui, Tuo Chen, Jing Zhang, Qiong Cao, Zhenan Sun, Hao Luo, and Dacheng Tao. A survey on self-supervised learning: Algorithms, applications, and future trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9052–9071, 2024.

[37] Tao Guo, Song Guo, and Junxiao Wang. Pfedprompt: Learning personalized prompt for vision-language models in federated learning. In *Proceedings of the ACM Web Conference 2023*, volume 1, pages 1364–1374, 2023.

[38] Sungwon Han, Sungwon Park, Fangzhao Wu, Sundong Kim, Chuhan Wu, Xing Xie, and Meeyoung Cha. Fedx: Unsupervised federated learning with cross knowledge distillation. In *European Conference on Computer Vision*, pages 691–707. Springer, 2022.

[39] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, volume 48, pages 1225–1234. PMLR, 2016.

[40] Fengxiang He, Tongliang Liu, and Dacheng Tao. Control batch size and learning rate to generalize well: Theoretical and empirical evidence. In *Advances in Neural Information Processing Systems*, volume 32, pages 1141–1150, 2019.

[41] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[42] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[43] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[44] István Hegedűs, Gábor Danner, and Márk Jelasity. Gossip learning as a decentralized alternative to federated learning. In *IFIP International Conference on Distributed Applications and Interoperable Systems*, volume 11534 of *Lecture Notes in Computer Science*, pages 74–90. Springer, 2019.

[45] István Hegedűs, Gábor Danner, and Márk Jelasity. Decentralized learning works: An empirical comparison of gossip learning and federated learning. *Journal of Parallel and Distributed Computing*, 148:109–124, 2021.

[46] Agrin Hilmkil, Sebastian Callh, Matteo Barbieri, Leon René Sütfeld, Edvin Listo Zec, and Olof Mogren. Scaling federated learning for fine-tuning of large language models. In *International Conference on Applications of Natural Language to Information Systems*, volume 12738, pages 15–23. Springer, 2021.

[47] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[48] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[49] Vlad Hondru, Florinel Alin Croitoru, Shervin Minaee, Radu Tudor Ionescu, and Nicu Sebe. Masked image modeling: A survey. *International Journal of Computer Vision*, pages 1–47, 2025.

[50] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[51] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

[52] Kai Hu, Yaogen Li, Shuai Zhang, Jiasheng Wu, Sheng Gong, Shanshan Jiang, and Liguo Weng. Fedmmd: a federated weighting algorithm considering non-iid and local model deviation. *Expert Systems with Applications*, 237:121463, 2024.

[53] Feihu Huang and Jianyu Zhao. Faster adaptive decentralized learning algorithms. In *International Conference on Machine Learning*, volume 235, pages 20490–20525, 2024.

[54] Zhicheng Huang, Xiaojie Jin, Chengze Lu, Qibin Hou, Ming-Ming Cheng, Dongmei Fu, Xiaohui Shen, and Jiashi Feng. Contrastive masked autoencoders are stronger vision learners. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4):2506–2517, 2023.

[55] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

[56] Stanisław Jastrzkebski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017.

[57] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):4037–4058, 2020.

[58] Shusen Jing, Anlan Yu, Shuai Zhang, and Songyang Zhang. Fedsc: Provable federated self-supervised learning with spectral contrastive objective over non-iid data. In *International Conference on Machine Learning*, volume 235, pages 22304–22325. PMLR, 2024.

[59] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021.

[60] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[61] Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Breaking the centralized barrier for cross-device federated learning. *Advances in Neural Information Processing Systems*, 34:28663–28676, 2021.

[62] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, volume 119, pages 5132–5143. PMLR, 2020.

[63] Asifullah Khan, Anabia Sohail, Mustansar Fiaz, Mehdi Hassan, Tariq Habib Afridi, Sibghat Ullah Marwat, Farzeen Munir, Safdar Ali, Hannan Naseem, Muhammad Zaigham Zaheer, et al. A survey of the self supervised learning mechanisms for vision transformers. *arXiv preprint arXiv:2408.17059*, 2024.

[64] Qazi Waqas Khan, Anam Nawaz Khan, Atif Rizwan, Rashid Ahmad, Salabat Khan, and Do-Hyeun Kim. Decentralized machine learning training: a survey on synchronization, consolidation, and topologies. *IEEE Access*, 11:68031–68050, 2023.

[65] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian Stich. A unified theory of decentralized sgd with changing topology and local updates. In *International Conference on Machine Learning*, volume 119, pages 5381–5393. PMLR, 2020.

[66] Anastasia Koloskova, Sebastian Stich, and Martin Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. In *International Conference on Machine Learning*, volume 97, pages 3478–3487. PMLR, 2019.

[67] Anastasiia Koloskova, Sebastian U. Stich, and Martin Jaggi. Sharper convergence guarantees for asynchronous sgd for distributed and federated learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 17202–17215, 2022.

[68] Lingpeng Kong, Cyprien de Masson d'Autume, Wang Ling, Lei Yu, Zihang Dai, and Dani Yogatama. A mutual information maximization perspective of language representation learning. *arXiv preprint arXiv:1910.08350*, 2019.

[69] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 2661–2671, 2019.

[70] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical Report TR-2009, University of Toronto, Toronto, Canada, 2009.

[71] Pietari Laitinen and Matti Vihola. An invitation to adaptive markov chain monte carlo convergence theory. *arXiv preprint arXiv:2408.14903*, 2024.

[72] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2019.

[73] Batiste Le Bars, Aurélien Bellet, Marc Tommasi, Erick Lavoie, and Anne-Marie Kermarrec. Refined convergence and topology learning for decentralized sgd with heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, volume 206, pages 1672–1702. PMLR, 2023.

[74] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[75] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[76] Jinho Lee, Inseok Hwang, Soham Shah, and Minsik Cho. Flexreduce: Flexible all-reduce for distributed deep learning on asymmetric network topology. In *2020 57th ACM/IEEE Design Automation Conference*, pages 1–6. IEEE, 2020.

[77] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*, volume 30, pages 2200–2209, 2017.

[78] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, volume 31, pages 6391–6401, 2018.

[79] Mingyi Li, Xiao Zhang, Qi Wang, Tengfei Liu, Ruofan Wu, Weiqiang Wang, Fuzhen Zhuang, Hui Xiong, and Dongxiao Yu. Resource-aware federated self-supervised learning with global class representations. In *Advances in Neural Information Processing Systems*, volume 37, pages 10008–10035, 2024.

[80] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3347–3366, 2021.

[81] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

[82] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, volume 30, pages 5330–5340, 2017.

[83] Xinting Liao, Weiming Liu, Chaochao Chen, Pengyang Zhou, Fengyuan Yu, Huabin Zhu, Binhui Yao, Tao Wang, Xiaolin Zheng, and Yanchao Tan. Rethinking the representation in federated unsupervised learning with non-iid data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22841–22850, 2024.

[84] Xinting Liao, Weiming Liu, Pengyang Zhou, Fengyuan Yu, Jiahe Xu, Jun Wang, Wenjie Wang, Chaochao Chen, and Xiaolin Zheng. Foogd: Federated collaboration for both out-of-distribution generalization and detection. *Advances in Neural Information Processing Systems*, 37:132908–132945, 2024.

[85] Yunming Liao, Yang Xu, Hongli Xu, Lun Wang, and Chen Qian. Adaptive configuration for heterogeneous participants in decentralized federated learning. In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2023.

[86] Opher Lieber, Or Sharir, Barak Lenz, and Yoav Shoham. Jurassic-1: Technical details and evaluation. *White Paper. AI21 Labs*, 1(9), 2021.

[87] Hong Liu, Jeff Z HaoChen, Adrien Gaidon, and Tengyu Ma. Self-supervised learning is more robust to dataset imbalance. In *International Conference on Learning Representations*, 2021.

[88] Lumin Liu, Jun Zhang, SH Song, and Khaled B Letaief. Client-edge-cloud hierarchical federated learning. In *ICC 2020-2020 IEEE International Conference on Communications*, pages 1–6. IEEE, 2020.

[89] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[90] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[91] Ben London. A pac-bayesian analysis of randomized learning with application to stochastic gradient descent. In *Advances in Neural Information Processing Systems*, volume 30, pages 2931–2940, 2017.

[92] Yang Lu. Artificial intelligence: a survey on evolution, models, applications and future trends. *Journal of management analytics*, 6(1):1–29, 2019.

[93] Ekdeep Lubana, Chi Ian Tang, Fahim Kawsar, Robert Dick, and Akhil Mathur. Orchestra: Unsupervised federated learning via globally consistent clustering. In *International Conference on Machine Learning*, volume 162, pages 14461–14484. PMLR, 2022.

[94] Xiao Ma, Hong Shen, Wenqi Lyu, and Wei Ke. Enhancing federated learning robustness in non-iid data environments via mmd-based distribution alignment. In *International Conference on Parallel and Distributed Computing: Applications and Technologies*, volume 15502, pages 280–291. Springer, 2024.

[95] Xindian Ma, Peng Zhang, Shuai Zhang, Nan Duan, Yuexian Hou, Ming Zhou, and Dawei Song. A tensorized transformer for language modeling. In *Advances in Neural Information Processing Systems*, volume 32, pages 2227–2237, 2019.

[96] Farhanna Mar'i, Ahmad Afif Supianto, and Fitra Abdurrachman Bachtiar. Comparation of federated and centralized learning for image classification. *PIKSEL: Penelitian Ilmu Komputer Sistem Embedded and Logic*, 11(2):393–400, 2023.

[97] David A McAllester. Some pac-bayesian theorems. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, volume 144 of *COLT '98*, pages 230–234. ACM, 1998.

[98] David A McAllester. Pac-bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 164–170, 1999.

[99] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[100] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, volume 54, pages 1273–1282. PMLR, 2017.

[101] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017.

[102] Wenlong Mou, Liwei Wang, Xiyu Zhai, and Kai Zheng. Generalization bounds of sgld for non-convex learning: Two theoretical viewpoints. In *Conference on Learning Theory*, volume 75, pages 605–638. PMLR, 2018.

[103] Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling data-constrained language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[104] Sony Peng, Yixuan Yang, Makara Mao, and Doo-Soon Park. Centralized machine learning versus federated averaging: A comparison using mnist dataset. *KSII Transactions on Internet and Information Systems*, 16(2):742–756, 2022.

[105] Ankit Pensia, Varun Jog, and Po-Ling Loh. Generalization error bounds for noisy, iterative algorithms. In *2018 IEEE International Symposium on Information Theory*, pages 546–550. IEEE, 2018.

[106] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[107] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, volume 139, pages 8748–8763. PMLR, 2021.

[108] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. Technical report, OpenAI, 2018.

[109] Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

[110] Yasar Abbas Ur Rehman, Yan Gao, Pedro Porto Buarque De Gusmão, Mina Alibeigi, Jiajun Shen, and Nicholas D Lane. L-dawa: Layer-wise divergence aware weight aggregation in federated self-supervised visual representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16464–16473, 2023.

[111] Jae Hun Ro, Theresa Breiner, Lara McConnaughey, Mingqing Chen, Ananda Theertha Suresh, Shankar Kumar, and Rajiv Mathews. Scaling language model size in cross-device federated learning. *arXiv preprint arXiv:2204.09715*, 2022.

[112] Fahad Sabah, Yuwen Chen, Zhen Yang, Muhammad Azam, Nadeem Ahmad, and Raheem Sarwar. Model optimization techniques in personalized federated learning: A survey. *Expert Systems with Applications*, 243:122874, 2024.

[113] Matthias Seeger. Pac-bayesian generalisation error bounds for gaussian process classification. *Journal of machine learning research*, 3(Oct):233–269, 2002.

[114] Milad Sefidgaran, Romain Chor, Abdellatif Zaidi, and Yijun Wan. Lessons from generalization error analysis of federated learning: You may communicate less often! In *International Conference on Machine Learning*, volume 235, pages 44093–44135. PMLR, 2024.

[115] Tao Shen, Didi Zhu, Ziyu Zhao, Zexi Li, Chao Wu, and Fei Wu. Will llms scaling hit the wall? breaking barriers via distributed resources on massive edge devices. *arXiv preprint arXiv:2503.08223*, 2025.

[116] Yifan Shi, Li Shen, Kang Wei, Yan Sun, Bo Yuan, Xueqian Wang, and Dacheng Tao. Improving the model consistency of decentralized federated learning. In *International Conference on Machine Learning*, volume 202, pages 31269–31291. PMLR, 2023.

[117] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[118] Kristina P Sinaga and Miin-Shen Yang. Unsupervised k-means clustering algorithm. *IEEE access*, 8:80716–80727, 2020.

[119] Rahul Singh, Abhishek Gupta, and Ness B. Shroff. Learning in constrained markov decision processes. *IEEE Transactions on Control of Network Systems*, 10(1):441–453, 2023.

[120] Mandt Stephan, Matthew D Hoffman, David M Blei, et al. Stochastic gradient descent as approximate bayesian inference. *Journal of Machine Learning Research*, 18(134):1–35, 2017.

[121] Tao Sun, Dongsheng Li, and Bao Wang. Adaptive random walk gradient descent for decentralized optimization. In *International Conference on Machine Learning*, volume 162, pages 20790–20809. PMLR, 2022.

[122] Tao Sun, Yuejiao Sun, and Wotao Yin. On markov chain gradient descent. In *Advances in Neural Information Processing Systems*, volume 31, pages 9918–9927, 2018.

[123] Yan Sun, Li Shen, and Dacheng Tao. Towards understanding generalization and stability gaps between centralized and decentralized federated learning. *arXiv preprint arXiv:2310.03461*, 2023.

[124] Yuwei Sun, Hideya Ochiai, and Hiroshi Esaki. Decentralized deep learning for multi-access edge computing: A survey on communication efficiency and trustworthiness. *IEEE Transactions on Artificial Intelligence*, 3(6):963–972, 2021.

[125] Zhenyu Sun, Xiaochun Niu, and Ermin Wei. Understanding generalization of federated learning via stability: Heterogeneity matters. In *International Conference on Artificial Intelligence and Statistics*, volume 238, pages 676–684. PMLR, 2024.

[126] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International Conference on Machine Learning*, volume 28, pages 1139–1147. PMLR, 2013.

[127] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[128] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE transactions on neural networks and learning systems*, 34(12):9587–9603, 2022.

[129] Hanlin Tang, Shaoduo Gan, Ce Zhang, Tong Zhang, and Ji Liu. Communication compression for decentralized training. In *Advances in Neural Information Processing Systems*, volume 31, pages 765–775, 2018.

[130] Zhenheng Tang, Shaohuai Shi, Xiaowen Chu, et al. Communication-efficient distributed deep learning: A comprehensive survey. *ArXiv Preprint arXiv:2003.06307*, 2020.

[131] Zhenheng Tang, Shaohuai Shi, Bo Li, and Xiaowen Chu. Gossipfl: A decentralized federated learning framework with sparsified and adaptive communication. *IEEE Transactions on Parallel and Distributed Systems*, 34(3):909–922, 2022.

[132] Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. Scale efficiently: Insights from pre-training and fine-tuning transformers. *arXiv preprint arXiv:2109.10686*, 2021.

[133] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

[134] Mauro DL Tosi and Martin Theobald. Convergence analysis of decentralized asgd. *arXiv preprint arXiv:2309.03754*, 2023.

[135] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, volume 139, pages 10347–10357. PMLR, 2021.

[136] Nguyen H Tran, Wei Bao, Albert Zomaya, Minh NH Nguyen, and Choong Seon Hong. Federated learning over wireless networks: Optimization model design and analysis. In *IEEE INFOCOM 2019-IEEE conference on computer communications*, pages 1387–1395. IEEE, 2019.

[137] Aleksei Triastcyn, Matthias Reisser, and Christos Louizos. Decentralized learning with random walks and communication-efficient adaptive optimization. In *NeurIPS 2022 Workshop on Federated Learning: Recent Advances and New Challenges*, 2022.

[138] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[139] George E. Uhlenbeck and Leonard S. Ornstein. On the theory of the brownian motion. *Physical Review*, 36(5):823–841, 1930.

[140] Grant Van Horn et al. Mini inaturalist 2021 dataset. Available at: https://github.com/visipedia/inat_comp, 2021.

[141] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008, 2017.

[142] Elahe Vedadi, Joshua V Dillon, Philip Andrew Mansfield, Karan Singhal, Arash Afkanpour, and Warren Richard Morningstar. Federated variational inference: Towards improved personalization and generalization. In *Proceedings of the AAAI Symposium Series*, volume 3, pages 323–327, 2024.

[143] Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, Jeroen Kloppenburg, Tim Verbelen, and Jan S Rellermeyer. A survey on distributed machine learning. *Acm computing surveys*, 53(2):1–33, 2020.

[144] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[145] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, volume 29, pages 3630–3638, 2016.

[146] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2020.

[147] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020.

[148] Jianyu Wang, Anit Kumar Sahu, Zhouyi Yang, Gauri Joshi, and Soummya Kar. Matcha: Speeding up decentralized sgd via matching decomposition sampling. In *2019 Sixth Indian Control Conference*, pages 299–300. IEEE, 2019.

[149] Lirui Wang, Kaiqing Zhang, Yunzhu Li, Yonglong Tian, and Russ Tedrake. Does learning from decentralized non-iid unlabeled data benefit from self supervision? In *International Conference on Learning Representations*, 2022.

[150] Puyu Wang, Yunwen Lei, Yiming Ying, and Ding-Xuan Zhou. Stability and generalization for markov chain stochastic gradient methods. *Advances in Neural Information Processing Systems*, 35:37735–37748, 2022.

[151] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578, 2021.

[152] Zhiyuan Wang, Hongli Xu, Jianchun Liu, He Huang, Chunming Qiao, and Yangming Zhao. Resource-efficient federated learning with hierarchical aggregation in edge computing. In *IEEE INFOCOM 2021-IEEE conference on computer communications*, pages 1–10. IEEE, 2021.

[153] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE transactions on information forensics and security*, 15:3454–3469, 2020.

[154] Xing Wu, Zhaowang Liang, and Jianjia Wang. Fedmed: A federated learning framework for language modeling. *Sensors*, 20(14):4048, 2020.

[155] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 9653–9663, 2022.

[156] Mengwei Xu, Dongqi Cai, Yaozong Wu, Xiang Li, and Shangguang Wang. {FwdLLM}: Efficient federated finetuning of large language models with perturbed inferences. In *2024 USENIX Annual Technical Conference*, pages 579–596, 2024.

[157] Zheng Xu, Yanxiang Zhang, Galen Andrew, Christopher A. Choquette-Choo, Peter Kairouz, H. Brendan McMahan, Jesse Rosenstock, and Yuanbo Zhang. Federated learning of gboard language models with differential privacy. *arXiv preprint arXiv:2305.18465*, 2023.

[158] Nan Yang, Xuanyu Chen, Charles Z Liu, Dong Yuan, Wei Bao, and Lizhen Cui. Fedmae: Federated self-supervised learning with one-block masked auto-encoder. *arXiv preprint arXiv:2303.11339*, 2023.

[159] Dmitry Yarotsky. Error bounds for approximations with deep relu networks. *Neural Networks*, 94:103–114, 2017.

[160] Mang Ye, Xiuwen Fang, Bo Du, Pong C Yuen, and Dacheng Tao. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Computing Surveys*, 56(3):1–44, 2023.

[161] Honglin Yuan, Warren Morningstar, Lin Ning, and Karan Singhal. What do we mean by generalization in federated learning? *arXiv preprint arXiv:2110.14216*, 2021.

[162] Liangqi Yuan, Ziran Wang, Lichao Sun, Philip S Yu, and Christopher G Brinton. Decentralized federated learning: A survey and perspective. *IEEE Internet of Things Journal*, 11(21):34617–34638, 2024.

[163] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, volume 139, pages 12310–12320. PMLR, 2021.

[164] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 12104–12113, 2022.

[165] Chaoning Zhang, Chenshuang Zhang, Junha Song, John Seon Keun Yi, Kang Zhang, and In So Kweon. A survey on masked autoencoder for self-supervised learning in vision and beyond. *arXiv preprint arXiv:2208.00173*, 2022.

[166] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, 2021.

[167] Qi Zhang, Yifei Wang, and Yisen Wang. How mask matters: Towards theoretical understandings of masked autoencoders. *Advances in Neural Information Processing Systems*, 35:27127–27139, 2022.

[168] Yucheng Zhao, Guangting Wang, Chong Luo, Wenjun Zeng, and Zheng-Jun Zha. Self-supervised visual representations learning by contrastive mask prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10160–10169, 2021.

[169] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

[170] Zihao Zhao, Yang Liu, Wenbo Ding, and Xiao-Ping Zhang. Federated pac-bayesian learning on non-iid data. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5945–5949. IEEE, 2024.

[171] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. Image bert pre-training with online tokenizer. In *International Conference on Learning Representations*, 2022.

[172] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. Federated learning on non-iid data: A survey. *Neurocomputing*, 465:371–390, 2021.

[173] Miaoxi Zhu, Li Shen, Bo Du, and Dacheng Tao. Stability and generalization of the decentralized stochastic gradient descent ascent algorithm. In *Advances in Neural Information Processing Systems*, volume 36, pages 31717–31751, 2023.

[174] Weiming Zhuang, Xin Gan, Yonggang Wen, Shuai Zhang, and Shuai Yi. Collaborative unsupervised visual representation learning from decentralized data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4912–4921, 2021.

[175] Weiming Zhuang, Yonggang Wen, and Shuai Zhang. Divergence-aware federated self-supervised learning. In *International Conference on Learning Representations*, 2021.