IMPERIAL

Representation Learning for Collective Intelligence

Author
S.A. Frilingos
CID: 01754624

Supervised by
Dr S. Vlaski

A Thesis submitted in fulfillment of requirements for the degree of
Master of Science in Communications and Signal Processing

Department of Electrical and Electronic Engineering
Imperial College London
2025

Abstract

This thesis investigates collaborative Self-Supervised Learning (SSL) in fully decentralized, Peer-to-Peer (P2P) networks. It establishes that standard protocols based on naive model averaging fail when agents learn from visually distinct data domains. This failure is attributed to the emergence of inconsistent and misaligned latent representation spaces, which fundamentally hinders effective collaboration.

To overcome this challenge, we propose a novel distributed collaboration method. Our approach decouples local specialization from collaborative generalization. First, agents train specialized representation embeddings on their unique local data. Then, instead of averaging the entire models, their latent spaces are explicitly aligned using a small, shared public dataset as a common anchor. After the alignment stage, agents collaborate again to build a single, robust model by sharing the parameters of a single-layer linear classifier.

Systematic experiments validate that this alignment-first approach significantly outperforms standard decentralized, federated, and centralized methods in multi-domain settings. The primary contribution is a practical distributed framework that enables learning from heterogeneous domains by separating the process of local representation learning from that of global classifier consensus.

Declaration of Originality

I hereby declare that the work presented in this thesis is my own unless otherwise stated. To the best of my knowledge the work is original and ideas developed in collaboration with others have been appropriately referenced.

Copyright Declaration

Acknowledgments

I would like to sincerely thank my advisor, Dr. Stefan Vlaski, for giving me the opportunity to dive into this previously foreign and fascinating world of distributed optimization and representation learning. His insights, although hard to grasp at first, were pivotal in shaping the mission of this project and finding effective solutions that led to our new proposed method.

I would also like to thank my parents, Nikos and Eleni for sending me frozen steaks to enjoy on long nights of work. Last but not least, my roommate Angelos for his daily support, Philippos for breaking the tension with our weekly BBQ, Dimos for providing an inspiring workplace at Putney, the Katerinas for keeping me company at the library sessions and Guillermo for our coffee breaks.

Contents

List of Acronyms

FL Federated Learning

RL Representation Learning

SSL Self-Supervised Learning

P2P Peer-to-Peer

ML Machine Learning

NLP Natural Language Processing

IID Independent and Identically Distributed

FedAvg Federated Averaging

MTL Multi-Task Learning

MLP Multi-Layer Perceptron

t-SNE t-Distributed Stochastic Neighbor Embedding

KL Kullback-Leibler

PCA Principle Component Analysis

SGD Stochastic Gradient Descent

ATC Adapt-then-Combine

HPC High Performance Computing

1 Introduction

Contents

1.1 Introduction to Representation Learning

1.1.1 A Review of Representation Learning

The success of machine learning models in performing a task, whether that is classification, identification or regression, is significantly dependent on the data that the model is trained on. While the quantity and quality of training data are important, the choice of data representation is arguably just as critical to the performance of the model. Indeed, much of the effort in deploying machine learning algorithms has traditionally been dedicated to feature engineering—the design of data transformations and preprocessing pipelines that yield the optimal representation of the data for training.

Representation Learning (RL) offers a paradigm shift from this manual process. It encompasses a class of methods that seek to automatically discover the salient features of the data, thereby making it "easier to extract useful information when building classifiers or other predictors" as stated in the official review of RL [1]. The underlying hypothesis is that real-world data is generated by a complex interplay of various factors, and a good representation is one that successfully disentangles these factors of variation. By learning such representations, a model can become less sensitive to irrelevant variations in the data and more attuned to the underlying semantics, which is a crucial step towards building more robust and generalizable Machine Learning (ML) systems.

1.1.2 Self-Supervised Learning (SSL) for Images

Within the broader field of representation learning, Self-Supervised Learning (SSL) has emerged as a particularly powerful paradigm for leveraging vast amounts of unlabeled data. Famously described by Yann LeCun as the "dark matter of intelligence" [2], SSL methods generate their own supervisory signals directly from the data. This is achieved by defining a pretext task where the model learns to predict certain properties of the input from other parts of the same input, thereby producing descriptive and intelligible representations without the need for manual labels [3].

While SSL has been applied across various domains, including audio processing and Natural Language Processing (NLP), its application in computer vision is the focus of this research. The dominant approach involves a form of instance discrimination, where the objective is to learn representations that are invariant to data augmentations. Rather than grouping images of the same class, the model is trained to produce similar representations for different augmented "views" of the same image. This focus on instance-level rather than class-level invariance, is a defining characteristic of the SSL paradigm. These views are generated through a series of stochastic transformations, such as random cropping, filtering, and color distortion [4]. By learning to maximize the agreement between these augmented views, SSL approaches have shown great promise in developing representations that are robust to common data issues like label corruption, distortion, and misalignment [5]. Furthermore, the generic, task-agnostic nature of the learned representations makes them particularly valuable in domains where the final downstream task may not be known in advance [6].

1.2 Introduction to Distributed Optimization

The representation learning methods discussed thus far operate within a centralized framework, where a single agent processes the entire dataset. This research, however, explores the paradigm of distributed optimization, specifically focusing on fully decentralized, or P2P, learning systems. In this setting, a global learning objective is collaboratively solved by a network of interconnected agents without the coordination of a central server [7].

The structure of such a system is formally described as a graph, $G = (N, E)$, which consists of a set of nodes N and a set of edges E that define the connections between them. In our context, each node $n \in N$ represents an agent—an independent learner capable of acquiring, storing, and processing its own local data. The edges $(n_i, n_j) \in E$ represent the communication links that allow pairs of agents to exchange information, such as model parameters. [8]

This graph-based topology forms the foundation of our research. It enables a departure from the centralized model by allowing each agent to independently learn representations from its local, and potentially unique, dataset. The collaborative process facilitated by the network can serve two distinct but related purposes. Firstly, agents can pool their knowledge to construct a single, unified consensus representation for a global downstream task—a model that is potentially more robust than what any single agent could learn in isolation. Secondly, communication can act as a mechanism for knowledge transfer, allowing an agent to leverage the diverse perspectives of its peers to enhance its own specialized representation, thereby improving its performance on the global data domain beyond what could be achieved by training in isolation.

The precise nature of this collaborative learning process, however, is more nuanced than a simple workflow of local training followed by model averaging. The effectiveness of the final representation is highly dependent on the characteristics of the data distribution across the network (e.g. Independent and Identically Distributed (IID) vs. non-IID), the communication protocol, and the specific goals of the downstream task. These complexities, which motivate the core research questions of this thesis, are discussed in more detail below 1.3.1.

1.2.1 Federated Learning

As a distinct paradigm within distributed optimization, Federated Learning (FL) presents a centrally-coordinated approach to training models on decentralized data [9]. Unlike the P2P topology, FL

employs a hub-and-spoke architecture, where a central fusion center orchestrates the learning process across a group of agents. This preserves the core structure of local training followed by a global aggregation step, but with a clear hierarchical communication pattern.

The typical FL process is iterative. The central server distributes the current global model to a subset of agents. Each agent then performs several epochs of training locally on its private data partition. Subsequently, the agents communicate their model updates—not their raw data—back to the server. The server then aggregates these updates, commonly through an algorithm like Federated Averaging (FedAvg) [10], to produce an improved global model. This updated model is then sent back to the agents for the next round of training.

The performance of FL is highly contingent on the statistical properties of the data distribution. In scenarios where data is distributed in an IID fashion, and communication is frequent, the performance of the federated model can theoretically approach that of a centrally trained model. However, a significant challenge arises in the more realistic non-IID setting. When the local data distributions differ substantially across agents, their individual models can drift during local training. This phenomenon, known as "client drift," can degrade the performance of the aggregated global model, as a simple averaging of disparate models may not yield a fruitful consensus. These challenges are particularly acute in the context of federated unsupervised representation learning, where client drift can lead to inconsistent representation spaces and misaligned features across the network [11].

1.2.2 Multi-Task Learning

While FL typically assumes a single-task objective where all agents collaborate to learn one global model, the Multi-Task Learning (MTL) framework addresses the scenario where agents in a network aim to solve distinct, though related, tasks [12]. In this paradigm, the goal of collaboration is not necessarily to converge to a single consensus model, but rather to leverage the relationships between tasks to improve the performance of each agent individually.

In an MTL network, each agent k seeks to find an optimal parameter vector w_k^o for its own local cost function $J_k(w_k)$. The underlying assumption is that the optimal models w_k^o across the network are related. For example, the models of closely neighboring agents may be expected to be similar. Collaboration in this context act as a form of regularization, allowing agents to share knowledge and transfer inductive biases that help them learn their individual tasks more effectively than if they had trained in isolation. This framework is particularly relevant for the

research questions explored in this thesis, where we investigate if P2P communication can help agents learn specialized, local representations tailored to their unique data domains.

1.3 Multi-Agent Representation Learning

1.3.1 Motivation and Core Research Questions

The preceding concepts of SSL and distributed optimization converge to the central investigation of this thesis: how can a network of agents effectively learn visual representations from heterogeneous, decentralized data? This research approaches the problem from two distinct but complementary perspectives, each addressing a fundamental type of data heterogeneity.

The first research path investigates knowledge transfer under semantic heterogeneity, commonly referred to as label skew. Consider a network of autonomous cameras in an agricultural setting, where each agent's field of view is limited to specific crops. One agent may primarily observe tomatoes, while another observes apples, resulting in highly non-IID local datasets. While a conventional federated approach would rely on a central server to aggregate this disparate data, the question explored here is whether these specialist agents, by exchanging model parameters over a P2P network, can collaboratively construct a single consensus representation. The goal is to determine if a shared "visual grammar" can be learned from semantically disjoint information, leading to a unified model that is robust and generalizable across all classes.

The second research path explores the problem of domain adaptation within a MTL framework. In this scenario, all agents share the same high-level objective—for instance, identifying a common set of object classes—but each agent's local data is subject to a unique domain shift. This could manifest as a network of cameras with different sensor calibrations, color processing, or lighting conditions, leading to systematic variations in image properties like contrast or saturation. Here, the objective is not to learn a single consensus model, but for each agent to learn a specialized representation to compensate for its local domain shift. The challenge is whether these agents, through P2P communication, can collaboratively use their personal representations to train a simple, shared classifier that can capture the full set of domains and classes more effectively than agents trained in isolation. The domain shift is trivially portrayed in Figure 1.1

1

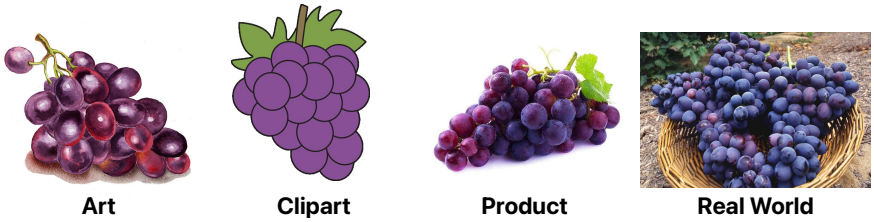


Figure 1.1: Four unique domains of grape images, as perceived by independent agents in Art, Clipart, Product, Real-World

These two research paths provide a comprehensive framework for evaluating the efficacy of decentralized SSL. The first path tests the limits of forming a general, unified model from semantically diverse data, while the second investigates the benefits of collaboration for building specialized, domain-robust local representation models. Motivated by these ideas, a more general question arises. Can effective collaboration be achieved not by averaging entire specialized models, but by first establishing a shared representational language and then collaboratively building a task-specific component? This approach will be explored further in this report.

1.3.2 Choosing The Core SSL Algorithm - VICReg

To investigate these research questions, this report employs the VICReg (**V**ariance-**I**nvariance-**C**ovariance **Reg**ularization) algorithm as the core SSL method [13]. VICReg was selected due to its specific properties that are highly advantageous for a multi-agent, decentralized learning environment. The algorithm trains joint embedding architectures by ”preserving the information content of the embeddings” [13] through a loss function composed of three key terms:

- **Invariance:** A standard mean-squared-error term that encourages the representations of two different augmented views of the same image to be similar.
- **Variance:** A hinge loss that maintains the standard deviation of each dimension of the embedding vector above a certain threshold. This term prevents a trivial solution where all embeddings collapse to a single point by forcing the representations of different samples within a batch to be distinct.
- **Covariance:** A regularization term that pushes the off-diagonal elements of the covariance matrix of the embeddings towards zero. This decorrelates the dimensions of the representation, preventing informational collapse where all dimensions carry redundant information.

1.3.3 Related Work

Addressing non-IID Challenges in Distributed Optimization

A significant challenge for distributed unsupervised learning, particularly under non-IID conditions, is the tendency for locally trained models to develop inconsistent and misaligned representation spaces. The work of Zhang et al [11] provides a foundational approach to this problem within a federated framework. They identify two key failure modes: *inconsistency* of the feature space due to label skew, and geometric *misalignment* of representations between clients.

To address this, they propose the Federated Contrastive Averaging (FedCA) algorithm. FedCA introduces two primary mechanisms: a server-managed *dictionary module* to aggregate sample representations and create a globally consistent feature space, and an *alignment module* that regularizes local models by forcing their outputs on a small public dataset to match those of a shared base model.

Our alignment strategy in this thesis is directly inspired by Zhang et al [11], who use a public dataset to correct latent space misalignment in a *federated setting*. However, their server-dependent FedCA algorithm introduces potential communication overhead via a shared dictionary module. Our work adapts this core alignment principle for a fully decentralized network, avoiding the need for a central server and focusing collaboration on a computationally lighter, shared linear classifier.

Contrastive Learning

Among the various approaches to SSL, contrastive methods have become a dominant paradigm. These methods are typically based on joint embedding architectures and operate on the principle of instance discrimination. The core objective is to learn an embedding space where representations of a ”positive pair” (e.g. two augmented views of the same image) are pulled closer together, while representations of ”negative pairs” (all other images) are pushed apart.

A prominent example, SimCLR, ”learns representations by maximizing agreement between differently augmented views of the same data example via a contrastive loss in the latent space” [4]. To source the requisite negative pairs, SimCLR relies on a large in-memory batch of samples. An alternative approach, Momentum Contrast (MoCo) [14], utilizes a dynamic dictionary, or memory bank, to maintain a queue of negative samples, thereby decoupling the batch size from the number

of negatives . The objective function for these methods is typically a variant of the InfoNCE loss [15].

While highly effective, the reliance of contrastive methods on a large number of negative pairs presents significant practical challenges, particularly for decentralized learning. The requirement for large batch sizes or external memory banks introduces substantial computational and memory overhead, which can be prohibitive for resource-constrained agents. Furthermore, the fundamental assumption of instance discrimination—where every other image is treated as a negative example—can be suboptimal, as it may inadvertently push apart representations of different instances that belong to the same semantic class. These limitations motivate the exploration of non-contrastive alternatives that can prevent representational collapse without the need for explicit negative sampling.

Clustering Methods

Clustering-based methods offer an alternative to instance-wise discrimination by instead grouping samples based on feature similarity. The goal is to learn representations that are semantically coherent at a group level. An early and influential approach, DeepCluster [16], "iteratively groups the features with a k-means clustering algorithm," using the resulting cluster assignments as pseudo-labels to supervise the network's weight updates. While effective, this process requires a computationally expensive and asynchronous clustering step, making it difficult to scale for large datasets and risking label error propagation through pseudo-labels.

More recent methods have focused on making the clustering process more efficient and integrated into the training loop. SwAV (Swapping Assignments between multiple Views) proposes an online clustering mechanism that avoids explicit pairwise comparisons [17]. It learns a set of prototype vectors, or "codes" and trains the model to predict the code of one augmented view from the representation of another. This assignment is performed efficiently online using the Sinkhorn-Knopp transform [18], eliminating the need for a large memory bank or a separate clustering phase. Although these methods can be viewed as performing contrastive learning at the cluster level, they still often rely on a global set of prototypes, which can be complex to maintain and synchronize across agents in a fully decentralized network without central coordination.

Distillation Methods

A third family of SSL methods avoids the need for explicit negative pairs by drawing inspiration from knowledge distillation [19]. Prominent examples such as BYOL (Bootstrap Your Own Latent) [20] and SimSiam [21] employ an asymmetric teacher-student architecture to prevent representational collapse. In this setup, two networks process different augmented views of an image, and one network (the student) is trained to predict the output of the other (the teacher).

Collapse is prevented through various architectural tricks that break the symmetry of the training. For instance, BYOL uses a momentum encoder, where the teacher network's weights are an exponential moving average of the student's weights. SimSiam, on the other hand, employs a stop-gradient operation, which prevents the gradient from flowing through the teacher network, thus avoiding a trivial solution where both networks converge to the same constant output. While these methods are highly effective, the precise theoretical reasons for their success in avoiding collapse are not yet fully understood. Other related techniques, such as OBoW (Online Bag-of-Words) [22], learn representations by having a student network reconstruct a bag-of-visual-words representation generated by a teacher. The inherent asymmetry and architectural complexity of these distillation-based methods, however, can complicate their application in a P2P setting, where model averaging and parameter exchange benefit from simpler, symmetric architectures.

Information Maximization Methods

This class of SSL methods prevents collapse not by using negative samples or architectural asymmetry, but by directly maximizing the information content of the learned embeddings. The central principle is redundancy reduction: the model is regularized to ensure that the different dimensions of the embedding vector are decorrelated, thus preventing an "informational collapse" where all dimensions carry the same information.

Barlow Twins, for example, feeds two distorted views of an image through two identical networks and computes the cross-correlation matrix between their outputs [23]. The loss function then pushes this matrix to be as close as possible to the identity matrix. This simultaneously encourages the representations of the same image to be similar (the diagonal elements of the matrix) while decorrelating the different feature dimensions (the off-diagonal elements). Similarly, W-MSE (Whitening Mean-Squared-Error) is based on whitening the latent features, which has a scattering effect that forces the embeddings to be decorrelated and spread out over the unit sphere, thereby

avoiding collapse [24]. A key advantage of these methods is their architectural simplicity; like VICReg, they do not require negative pairs, momentum encoders, or stop-gradients, making them well-suited for simpler training pipelines.

However, methods like Barlow Twins that operate on the cross-correlation matrix between network branches may implicitly favor scenarios where the data distributions are similar. In a highly heterogeneous multi-agent setting, where different agents learn from distinct data domains, this can be a limitation. This motivates the use of a method that regularizes each agent's representation space independently.

1.3.4 VICReg: The Core Piece of the Puzzle

The VICReg algorithm can be understood as a synthesis of the strongest principles from the aforementioned methods, adapted into a framework that is uniquely suited for decentralized, multi-agent learning. While it is based on a standard joint embedding architecture, its approach to preventing representational collapse is what sets it apart.

VICReg adopts the redundancy reduction principle from information maximization methods like Barlow Twins, using a covariance term to decorrelate embedding dimensions. However, it makes two critical improvements. First, it introduces an *explicit variance-preservation term*, which directly prevents the collapse of embeddings to a trivial solution and, in doing so, alleviates the need for the batch-wise normalization required by other methods. Second, and most importantly for this research, the variance and covariance regularizations are applied to each network branch *independently*. This is a key distinction from methods that regularize the cross-correlation between branches, as it makes the framework more robust to the statistical shifts inherent in the heterogeneous, multi-domain data found across different agents.

These properties are particularly advantageous for our two chosen benchmarks. On CIFAR-10 with label skew, VICReg avoids collapse despite each agent training on a narrow semantic slice, ensuring that collaboration can expand the representational space across disjoint classes. On OfficeHome with domain shifts, VICReg's branch-wise regularization allows each agent to stabilize its representation space despite systematic visual differences between domains.

Crucially, VICReg's design avoids several components that pose significant challenges in a decentralized setting. Unlike contrastive methods such as SimCLR, it does not require a large number of negative examples and is therefore not reliant on large batch sizes—a critical advantage

for agents training on small, local datasets. Furthermore, it eschews the need for the asymmetric architectural tricks of distillation methods, such as momentum encoders or stop-gradients, which simplifies the process of model aggregation. This combination of architectural simplicity and a robust, explicit mechanism for preventing collapse makes VICReg an ideal foundation for exploring representation learning in a distributed network.

1

2 Methods

Contents

2.1 Core Algorithm: VICReg

2.1.1 Network Architecture

As introduced in Section 1.3.4, the VICReg algorithm is based on a *joint embedding architecture*. While the framework does not strictly require shared parameters between its two processing branches, the experiments in this thesis employ a Siamese network implementation where both branches are identical and share weights.

The architecture of each branch, as depicted in Figure 2.1, consists of two main components. The first is an *encoder*, denoted as f_θ, which is a deep convolutional neural network, in our case a ResNet-18 that processes an input image and outputs a feature vector. This vector, y, serves as the final representation used for downstream tasks. The second component is an *expander*, denoted as h_ϕ, which is typically a Multi-Layer Perceptron (MLP). The expander takes the representation y as input and maps it to a higher-dimensional embedding space. The purpose of the expander is twofold: first, to project the representations into a space where the invariance loss can be effectively computed, and second, to facilitate the decorrelation of the embedding dimensions as enforced by the covariance regularization term. After the self-supervised training phase is complete, the expander is discarded.

The end-to-end data flow through the architecture begins with a stochastic data augmentation pipeline. For a given image i from the dataset I, two distinct transformation functions, t, t', are sampled from a predefined distribution of augmentations T. These functions are applied to the image to generate two augmented "views," $x = t(i)$ and $x' = t'(i)$. Both views are then passed through the shared-weight encoder f_θ to produce their respective representations, $y = f_\theta(x), y' = f_\theta(x')$. Finally, these representations are mapped by the expander h_ϕ into the embedding space, yielding the embeddings $z = h_\phi(y), z' = h_\phi(y')$, upon which the VICReg loss function is computed.

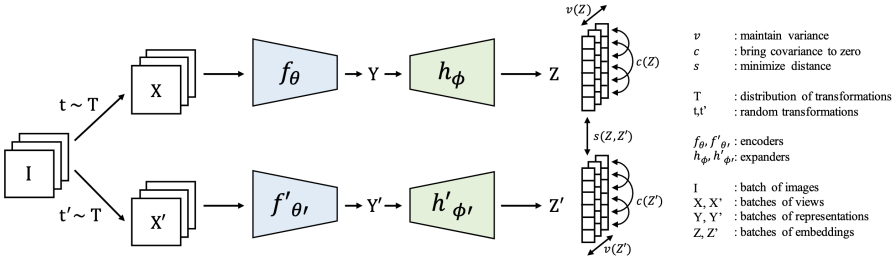


Figure 2.1: VICReg main design architecture. I represents a batch of images, which undergo certain transformations to become different views in X and X' before being encoded (encoder is a ResNet network) into Y and Y'. Finally, these representations are passed through the expander (Multi-Layer Perceptron) to produce the embeddings Z and Z' on which the VICReg loss function is minimized.

2.1.2 VICReg Loss Function

The VICReg loss function is computed on batches of embeddings and is composed of three distinct regularization terms: an *invariance* term to promote similarity between positive pairs, and *variance* and *covariance* terms that together prevent representational collapse.

During training, images are processed in mini-batches of size n, usually between 128 and 256.

This results in two corresponding batches of embeddings, denoted as $Z = [z_1, ..., z_n]$ and $Z' = [z'_1, ..., z'_n]$, where each z_i is a d-dimensional vector. For the subsequent definitions, we denote z^j the vector composed of the j-th element from all n embedding vectors within a batch. For example, if there the set with $n = d = 3$ where $Z = [z_1, z_2, z_3] = \begin{bmatrix} 7 & -2 & 5 \\ 3 & 0 & 9 \\ -1 & 4 & 8 \end{bmatrix}$ then $z^2 = [-1, 4, 8]$, in a zero-indexed format.

The first component is the *variance* regularization term, $v(Z)$, which prevents the embeddings from collapsing to a single point. It operates as a hinge loss on the batch-wise standard deviation of each embedding dimension, penalizing it for dropping below a target value γ, which is typically fixed to 1. As defined in [13], with $S(x, \epsilon) = \sqrt{\text{Var}(x) + \epsilon}$ representing the regularized standard deviation and ϵ being a small scalar for numerical stability, the term is:

$$v(Z) = \frac{1}{d} \sum_{j=1}^{d} \max(0, \gamma - S(z^j, \epsilon)) \tag{2.1}$$

This formulation forces the model to maintain variance in the embedding space through γ, ensuring that different inputs map to different representations. The use of standard deviation S, rather than variance directly, is critical for stable optimization, as it provides a more consistent gradient signal when the variance is close to the target value. If we had $S(x) = \text{Var}(x)$ in 2.1 and the gradient $\nabla_x S$ approaches 0, as x approaches $\bar{x}$, then the gradient of v would also trend close to 0, thus collapsing the embeddings.

The second component is the *covariance* regularization term, $c(Z)$, which prevents informational collapse by decorrelating the dimensions of the embedding. This is achieved by penalizing the off-diagonal elements of the batch-wise covariance matrix, $C(Z)$, which is defined as:

$$C(Z) = \frac{1}{n-1} \sum_{i=1}^{n} (z_i - \bar{z})(z_i - \bar{z})^T, \quad \text{where} \quad \bar{z} = \frac{1}{n} \sum_{i=1}^{n} z_i. \tag{2.2}$$

Having now defined the covariance matrix 2.2 and drawing inspiration from Barlow Twins [23], the covariance regularization term can be defined as the sum of the squared off-diagonal elements of this matrix:

$$c(Z) = \frac{1}{d} \sum_{i \neq j} [C(Z)]_{i,j}^2 \tag{2.3}$$

By driving these off-diagonal terms to zero, this regularizer encourages each dimension of the embedding to encode unique information, thus maximizing the novel information content of the representation. $C(Z)$ is scaled by the inverse of the representation dimension $1/d$, thus encouraging it towards an identity matrix.

The final component is the *invariance* term, $s(Z, Z')$, which enforces the primary SSL objective: that different views of the same image should have similar representations. This is measured as the mean squared Euclidean distance between the embedding vectors of each positive pair, without any normalization.

$$s(Z, Z') = \frac{1}{n} \sum_{i} \|z_i - z'_i\|_2^2 \tag{2.4}$$

In the simulations, these three terms (variance, invariance, covariance) were logged throughout training, to ensure that representational stability was established. Figure 2.2 validates that representational collapse was avoided and each term served its function.

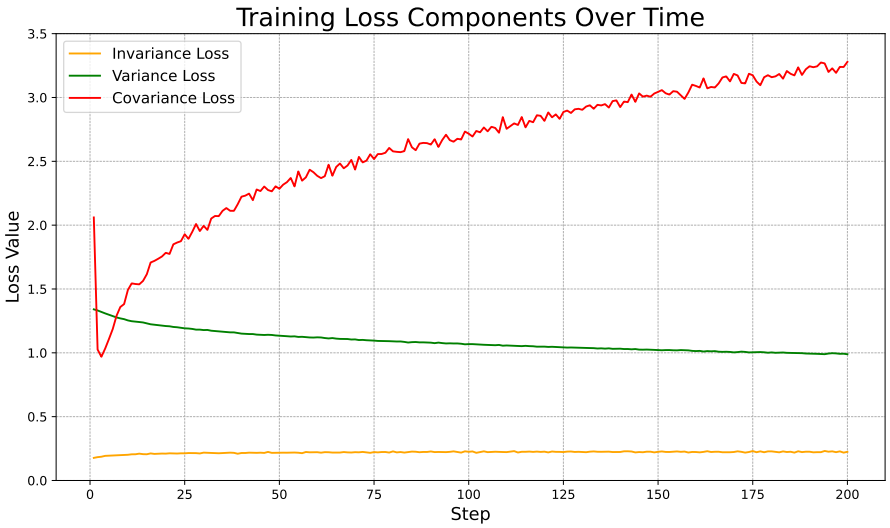


Figure 2.2: Loss terms variance, invariance and covariance during training. The variance is kept at a steady value around 1, ensuring representations do not collapse to zero. Covariance increases through training, ensuring decorrelation of the embedding dimensions. Invariance is kept at a steady, small positive value, ensuring same class embeddings are closely related in the representation space.

These three components are combined into a single, weighted loss function. The hyperparameters λ, μ, and ν govern the trade-off between these competing objectives. A higher λ value places more emphasis on the invariance criterion, forcing the model to learn representations that are insensitive to augmentations. Conversely, higher μ and ν values prioritize the structural properties of

the embedding space, focusing on collapse prevention and information maximization. The specific tuning of these weights, discussed in Section 2.2.5, is crucial for achieving optimal performance. The complete loss function is defined as:

$$\ell(Z, Z') = \lambda s(Z, Z') + \mu[v(Z) + v(Z')] + \nu[c(Z) + c(Z')] \tag{2.5}$$

The overall training objective is to minimize this loss over the entire unlabeled dataset D. This is achieved using stochastic gradient descent, where for each training step, a mini-batch of images $I \in D$ is sampled. For each image in the batch, two random augmentations are generated, and the loss $l(Z^I, Z'^I)$ is computed on the resulting embeddings. By iteratively updating the parameters of the encoder θ and the expander ϕ to minimize this objective, the model learns a mapping from the raw pixel space to a structured embedding space that is both informative and robust to nuisance variations. The full objective function 2.6 is:

$$\mathcal{L} = \sum_{I \in \mathcal{D}} \sum_{t,t' \sim \mathcal{T}} \ell(Z^I, Z'^I) \tag{2.6}$$

2.2 Experimental Setup

2.2.1 Datasets

The selection of appropriate datasets was guided by the need to balance computational feasibility with the requirements of the core research questions. While foundational SSL literature often utilizes large-scale datasets such as ImageNet [25], the significant computational resources required for training on such a scale were deemed prohibitive for the iterative development and analysis central to this project. Consequently, two smaller yet widely recognized benchmark datasets were selected, each serving a distinct purpose in the experimental design.

For the initial development, validation, and benchmarking of the centralized, federated, and decentralized simulation frameworks, the CIFAR-10 dataset was employed [26]. Comprising of 60,000 low-resolution (32x32 pixels) images across 10 classes, its manageable scale facilitated rapid prototyping and robust testing of the core algorithms, particularly for investigating the effects of semantic heterogeneity— *label skew*.

To address the research questions related to domain adaptation, the Office-Home dataset was selected [27]. This dataset is uniquely suited for this purpose, as it contains images of 65 object classes captured across four distinct visual domains: artistic renderings (Art), Clipart, product pictures, and real-world photographs. This multi-domain structure provides a natural and challenging testbed for evaluating how agents can learn specialized representations to counteract domain shift. By distributing data such that each agent trains on a different domain while sharing the same underlying classes, it becomes possible to directly measure the benefit of peer-to-peer communication in building a robust, domain-invariant classifier.

2.2.2 Data Partitioning for Heterogeneity

A central component of this research is the simulation of heterogeneous data environments. While the centralized training mode serves as an IID baseline using the full dataset, the multi-agent frameworks are evaluated under two primary forms of non-IID data partitioning: class and domain heterogeneity.

Class Heterogeneity (Label Skew)

To simulate scenarios where agents are specialists in different classes, a label skew partitioning strategy was implemented. This approach distributes the data such that each agent receives a biased, non-uniform sample of the available classes. The degree of this non-IID distribution is controlled using a Dirichlet distribution.

Formally, for a dataset with K classes and a network of N agents, we draw a vector of class proportions $q_i = (q_{i1}, \ldots, q_{iK})$ for each agent i from a Dirichlet distribution with a concentration parameter α: $q_i \sim \mathrm{Dir}(\alpha)$.

Here, q_{ik} represents the proportion of data from class k that is assigned to agent i. The parameter α directly controls the level of heterogeneity: a small value (e.g., $\alpha \to 0$) results in each agent receiving samples from only a few classes, creating a high degree of non-IID label skew. Conversely, a large value ($\alpha \to \infty$) results in a uniform distribution, where each agent receives a similar proportion of all classes, approaching the IID case. In the non-IID label setting, the primary goal is to evaluate the ability of communicating agents to form a single, robust representation model that can perform well on the global distribution.

Domain Heterogeneity (Domain Shift)

The second approach investigates the challenge of domain shift, where agents share a common set of classes but their local data is subject to different acquisition parameters. This was implemented in two ways:

1. **Artificial Domain Shift:** For datasets like CIFAR-10, distinct domains were simulated by applying a different, fixed data transformation (e.g. Gaussian blur, rotation, random colour jitter) to the local data of each agent.

2. **Natural Domain Shift:** The Office-Home dataset was used to provide realistic domain heterogeneity. Each agent was assigned all classes but from only one of the four domains: Art, Clipart, Product, or Real-World.

In this experimental setting, the objective is not to create a single global representation model, but to assess whether peer-to-peer communication enables each agent to learn a more robust personalized local model for its specific domain, compared to a baseline of training in complete isolation, while collaborating towards a robust, shared linear classifier.

2.2.3 Network Topologies

In the decentralized P2P framework, the communication protocol is governed by the underlying graph structure, as introduced in Section 1.2. The specific topology of this graph dictates which agents can exchange information and directly influences the dynamics of model aggregation and knowledge transfer across the network. The communication structure is formally represented by a symmetric adjacency matrix, where a non-zero entry indicates a communication link between two agents. For this research, several distinct network topologies were implemented to investigate their impact on learning performance.

In the scope of the project, four different network topologies were implemented for communication: fully-connected, ring, random and disconnected. The design of the topology is based on the graph matrix $G = (N, E)$ where N refers to the set of *nodes* and E denotes a set of directed *edges* linking pairs of nodes in N. Thus, the code for each topology simply generates the G graph matrix, signifying the connection design.

- **Fully-Connected:** In this topology, every agent is connected to every other agent in the network. This represents an ideal communication scenario where information from any agent can be directly shared with the entire network in a single communication round, allowing for the fastest possible consensus.

- **Ring:** Agents are arranged in a circular structure, where each agent is connected only to its two immediate neighbors (one preceding and one succeeding). While information can still propagate across the entire network, it must do so iteratively through multi-hop communication, resulting in a slower rate of information diffusion compared to a fully-connected graph.

- **Random:** A random graph is generated by connecting pairs of agents with a fixed probability, with the constraint that the final graph must be connected (i.e. no agent or subgroup of agents is completely isolated). This topology simulates more realistic, ad-hoc network structures that are neither perfectly ordered nor fully connected.

- **Disconnected:** This topology serves as a critical baseline where no communication occurs between agents. Each agent trains in complete isolation. This allows for a direct measurement of the performance gains attributable to collaboration by comparing the results of connected topologies against this non-communicative scenario.

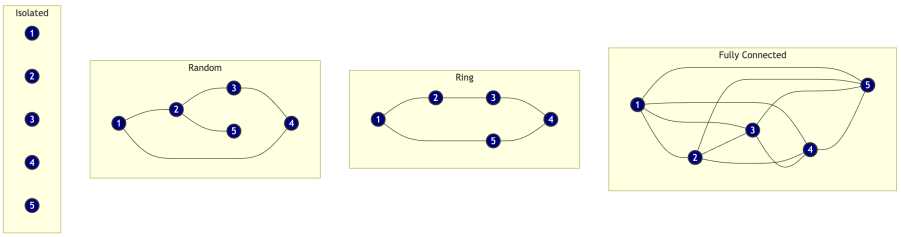


Figure 2.3: Example display of the different network topologies from left to right: Disconnected (Isolated), Random, Ring, Fully-Connected

2.2.4 Evaluation Protocol

To assess the quality of the representations learned during the self-supervised phase, two standard downstream evaluation protocols from popular SSL literature were employed: linear evaluation and k-Nearest Neighbors (k-NN) classification. The application of these protocols was adapted to suit the two primary research objectives: evaluating a single consensus model and evaluating multiple personalized local models.

For experiments focused on learning a consensus model (i.e. the centralized and federated modes), a single, final aggregated model is evaluated on the global test set that contains all classes. For the P2P experiments, a personalized evaluation is performed throughout training, using k-NN and the linear classifier. In this case, each agent's final, specialized model is evaluated independently on the global test set containing all classes and domains. The final performance is then reported as the average accuracy across all agents.

Linear Evaluation

The linear evaluation protocol measures the linear separability of the learned features, providing a standardized measure of representation quality. Due to its computational requirements, this method was used exclusively for the final evaluation after the completion of the self-supervised training phase in the centralized and federated training schemes. However, for our method, as will be explained in 2.3.4 a linear classifier is initialized and trained throughout the communication phase. The linear evaluation procedure is as follows:

1. The self-supervised encoder network f_θ, also referred to as the backbone, is frozen, and its weights are no longer updated.
2. A new, randomly initialized linear classifier (a single fully-connected neural network layer followed by a softmax activation function [28]) is attached to the frozen backbone.
3. This linear classifier is then trained in a fully supervised manner on the labeled training data of the downstream task, using a standard optimizer such as Adam [29].
4. Finally, the performance of the trained classifier is measured on the held-out test set. The resulting accuracy indicates how well the self-supervised representations can be linearly mapped to the target classes.

k-NN Evaluation

The k-Nearest Neighbors (k-NN) evaluation offers a computationally efficient alternative for assessing the quality of the feature space. It was used both for final model evaluation and to generate learning curves by monitoring performance at regular intervals during training. The protocol is as follows:

1. The encoder backbone is frozen at the time of evaluation.

2. A feature "memory bank" is created by passing the entire labeled training set through the frozen encoder and storing the resulting representation vectors.

3. For each image in the test set, its representation is computed using the same frozen encoder.

4. The cosine similarity between the test image's representation and all representations in the memory bank is calculated.

5. The k most similar training set representations—the nearest neighbors—are identified. The class of the test image is then predicted based on a majority vote of the labels of these k neighbors.

This method is significantly faster than linear evaluation as it does not require training a new network, making it ideal for frequent performance monitoring.

t-SNE Evaluation

In addition to the quantitative metrics, a qualitative assessment of the learned representation space was performed using t-Distributed Stochastic Neighbor Embedding (t-SNE) [30]. t-SNE is a non-linear dimensionality reduction technique designed specifically for visualizing high-dimensional datasets in a low-dimensional space, such as a two-dimensional plot.

The algorithm works by constructing a probability distribution over pairs of high-dimensional data points, where similar points have a high probability of being picked, and dissimilar points have a low probability. It then defines a similar probability distribution over the points in the low-dimensional map and minimizes the Kullback-Leibler (KL) [8] divergence between the two distributions. This process effectively arranges the data in the 2D space such that the neighborhood structure of the original high-dimensional representation space is preserved as closely as possible.

The algorithm works by first converting the high-dimensional Euclidean distances between data points into conditional probabilities that represent similarities. Let $p_{j|i}$ be the conditional probability that datapoint x_i would pick x_j as its neighbor. It then defines a similar probability distribution over the points in the low-dimensional map, which we can call $q_{j|i}$. The core of the t-SNE algorithm is to arrange the points in the low-dimensional embedding by minimizing the KL divergence between the joint probability distributions, P and Q, of the high-dimensional and

low-dimensional spaces, respectively. The cost function C that is minimized is given by:

$$C = \mathrm{KL}(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{2.7}$$

This process effectively arranges the data in the 2D space such that the neighborhood structure of the original high-dimensional representation space is preserved as closely as possible.

For this research, t-SNE is applied to the representations of the test set as generated by the final trained models. The resulting visualizations provide an intuitive visual assessment of the quality of the learned feature space. A successful representation will manifest as a 2D plot where data points belonging to the same class form consistently similar clusters throughout agents, offering qualitative evidence of the model's ability to learn semantically meaningful features.

Note: t-SNE is only used for qualitative illustration. The relative sizes of, and distances between, clusters in the 2D projection should not be interpreted quantitatively. The t-SNE figures are presented to visualize the abstract structure of the representation space under domain shift.

Principle Component Analysis (PCA) was also tested, which is a linear dimensionality reduction method that projects high-dimensional data onto a lower-dimensional space while preserving the maximum amount of variance. While it provides a perspective of the global variance structure, it does not capture non-linear relationships as well and thus was omitted from the report.

2.2.5 Implementation Details & Hyperparameters

The experimental framework was implemented in Python using the PyTorch machine learning library. All parameters were managed through a centralized configuration script to ensure consistency and reproducibility across simulations. The training was conducted on Imperial's high-performance computing cluster equipped with NVIDIA GPUs, utilizing the CUDA toolkit.

The hyper-parameters for the VICReg loss function were determined based on the recommendations from the original paper [13] and a subsequent grid search. The covariance weight, ν, was fixed at 1. A grid search was performed for the invariance and variance weights, λ and μ, over the set 5, 15, 25, 50, confirming the optimal performance was achieved with $\lambda = \mu = 25$.

The key parameters used across the different experimental setups are summarized in Table 2.1. The total number of training epochs in the centralized mode corresponds to the total number of communication rounds in the multi-agent modes.

Table 2.1: Key hyperparameters and implementation details for all experimental setups.

Parameter	Value
Model Architecture	
Encoder Backbone	ResNet-18
Projection Input Dimension	512
Projection Hidden Dimension	2048
Projection Output Dimension	2048
VICReg Loss Coefficients	
Invariance Weight (λ)	25.0
Variance Weight (μ)	25.0
Covariance Weight (ν)	1.0
General Training Parameters	
Optimizer	SGD (momentum=0.9, weight_decay=1e-4)
Batch Size	256
Learning Rate (CIFAR-10)	0.01
Learning Rate (Office-Home)	0.001
Total Epochs / Comm. Rounds	100–200 (varied by experiment)
Multi-Agent Parameters	
Number of Agents	4–15 (varied by experiment)
Local Epochs	1
Non-IID Label Skew (α)	0.5–100 (varied by experiment)
Alignment Coefficient	25
Evaluation Parameters	
Linear Evaluation Epochs	50
k-NN Neighbors (k)	20
k-NN Temperature (τ)	0.1
Evaluation Frequency	Every 5 rounds

2.3 Simulation Frameworks

2.3.1 Centralized Baseline

The centralized training framework serves as the primary performance benchmark for this research. In this mode, a single agent is trained on the entire dataset, effectively replicating the standard training protocol for the VICReg algorithm [13]. This setup provides an upper-bound performance metric, representing the ideal scenario with no data partitioning or communication constraints for the label-skew approach.

The implementation follows a standard self-supervised training loop. For each training epoch, the model iterates through the dataloader, processing one mini-batch at a time. For each batch, two augmented views are generated, passed through the encoder and expander, and the VICReg loss is computed on the resulting embeddings. The model's parameters are then updated via a single step of a SGD optimizer. To facilitate development, certain components, such as the VICReg-specific data augmentations and loss computation, were implemented using the lightly SSL [31] library for PyTorch.

To ensure the correctness of this baseline implementation, it was first validated on the CIFAR-10 dataset. The final performance of the trained model was benchmarked against established results from the literature, as in the lightly experiment benchmark results [31], confirming that the implementation correctly reproduces the expected behavior and performance of the VICReg algorithm. Throughout the multi-agent experiments, this validated centralized model served as the crucial point of comparison to quantify the impact of decentralization and data heterogeneity with CIFAR-10.

Centralized Validation

A critical first step in the experimental process was to validate the custom implementation of the VICReg algorithm and establish a reliable performance benchmark. The objective was to ensure that the centralized training could reproduce results consistent with those reported in the wider SSL literature. As the original VICReg paper did not report benchmarks on the CIFAR-10 dataset, an external reference point was sourced from the lightly SSL library's publicly available benchmarks, which show that high-performing models typically achieve accuracies of approximately 80% on this dataset [31].

Achieving a comparable level of performance required a systematic hyperparameter tuning process. The initial implementation, using default parameters, yielded suboptimal results, indicating that the learning dynamics were not yet stable. The tuning process focused primarily on two key areas: the learning rate and the VICReg loss coefficients. A grid search was conducted on the learning rate, testing values in the range of [0.1, 0.01, 0.001]. It was found that a balanced learning rate of 0.01 was crucial for preventing loss divergence and ensuring stable convergence on CIFAR-10. Subsequently, holding the learning rate fixed, a grid search was performed on the λ and μ coefficients of the VICReg loss, confirming that the values of $\lambda = \mu = 25$, were optimal for this setup.

After 200 epochs of training with these tuned parameters, the centralized model achieved a final linear evaluation accuracy of 74.3% and k-NN of 72.3% as depicted in Figure 2.4. This result was deemed a successful validation of the implementation. It is significantly higher than the 10% accuracy expected from a random-chance classifier, confirming that the model learned semantically meaningful representations. This quantitative result was further supported by qualitative analysis using t-SNE visualizations in Figure 2.5, which showed clear clustering of the classes in the learned embedding space, providing visual evidence of the representation's quality. This validated baseline serves as the benchmark against which all subsequent multi-agent experiments are compared for CIFAR-10.

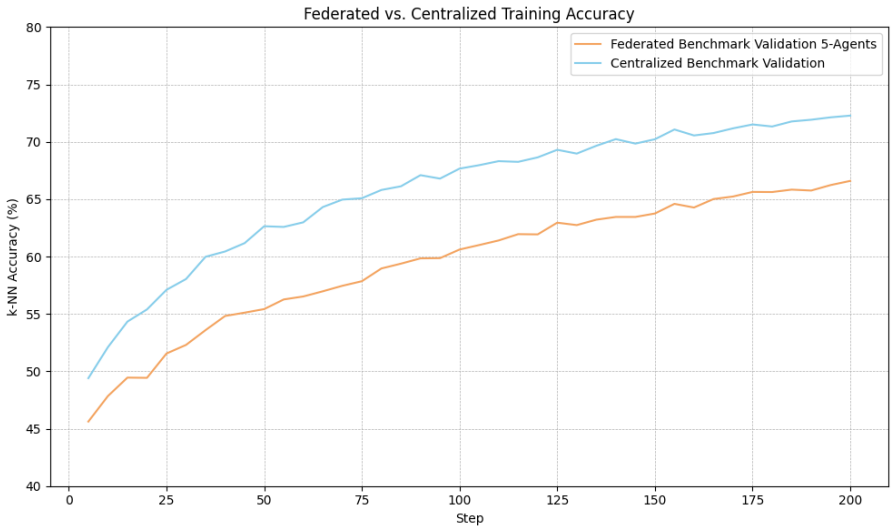


Figure 2.4: Centralized (blue) vs FL (orange) with 5 agents. The k-NN accuracy percentage is displayed on the vertical axis, with the number of epochs (steps) on the horizontal axis

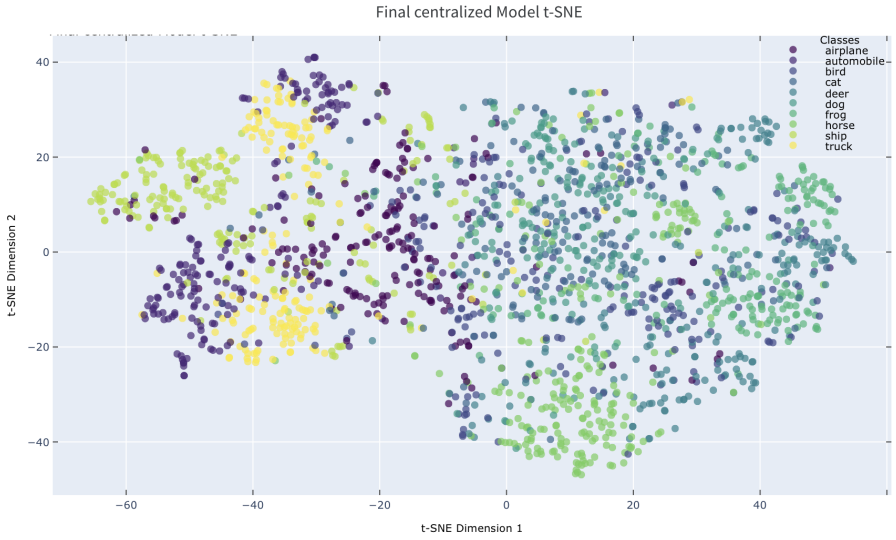


Figure 2.5: Two dimensional visualization of representations with t-SNE for centralized training with CIFAR-10

2.3.2 Federated Learning Baseline

The FL framework, as introduced in Section 1.2.1, serves as the baseline prototype for a centrally-coordinated distributed learning approach. In this mode, the training data is partitioned among the agents according to the heterogeneity scenarios described previously: *label skew* or *domain shift*.

Unlike the single loop of the centralized mode, the FL training process is structured around a series of communication rounds. Within each round, a set amount of local computation is performed by each agent before the results are aggregated. This introduces a critical trade-off between communication frequency and local computation, controlled by the number of local epochs. To maintain a fair comparison of total computational effort, the federated simulations are configured such that the total number of training passes over the data are equivalent to the centralized baseline. For instance, a 200-epoch centralized experiment can be compared to a FL experiment with 200 communication rounds and 1 local epoch per round, or one with 100 communication rounds and 2 local epochs per round.

The federated training process for a single communication round proceeds as follows:

1. **Distribution:** The central server sends the current global model state to all participating agents.

2. **Local Training:** Each agent creates a local copy of the global model and trains it for a specified number of local epochs on its private data partition. The training process for each agent is identical to a single-agent training step, utilizing the VICReg loss and an SGD optimizer.

3. **Aggregation:** After local training is complete, each agent sends its updated model weights back to the central server. The server then aggregates these models, typically by averaging their parameters with FedAvg [9], to produce a new, improved global model.

This cycle is repeated for the total number of communication rounds. To monitor progress, the performance of the current global model is evaluated on the global test set using the k-NN method for every n rounds as specified in the configuration, generating a learning curve. As the final output of the FL process is a single consensus model, the final evaluation is identical to the centralized baseline, employing linear evaluation, k-NN and t-SNE to assess its quality.

Federated Validation

To validate the FL framework, it was essential to demonstrate that under ideal conditions, its performance could closely approximate that of the centralized benchmark. The ideal condition for FL is a scenario with a nearly IID data distribution across agents, where the communication and aggregation process should theoretically introduce minimal performance degradation.

To this end, a specific validation experiment was conducted. A network of 5 agents was configured to train on the CIFAR-10 dataset for 200 communication rounds with one local epoch per round. The data was partitioned using a Dirichlet distribution with a very high concentration parameter ($\alpha = 100$), ensuring a nearly uniform IID split. All other hyperparameters were kept identical to the centralized baseline. The results validate correct implementation as seen in Figure 2.4 and further discussion of the results can be found in section 3.2.1.

2.3.3 Decentralized P2P Framework

The fully decentralized P2P framework represents the core experimental paradigm of this research, moving beyond the server-client architecture of FL. In this setting, agents collaborate directly with their neighbors as defined by a given network topology, without any central coordination. The training process is iterative, structured around communication rounds, and consists of two primary

stages per round: local training and P2P communication. The specific objective and evaluation protocol, however, are adapted to the two distinct heterogeneity scenarios under investigation.

Consensus Learning under Class Heterogeneity

This experimental path investigates the formation of a single, unified model from semantically diverse *label-skewed* data. The training process for each communication round is as follows:

1. **Local Training:** Each agent trains its local model for a specified number of epochs on its private, non-IID data partition, identical to the local training step in the federated baseline.
2. **Gossip Averaging:** Following local training, agents engage in a communication step. Using a gossip averaging protocol (see Section 2.3.3), each agent averages its model parameters with those of its immediate neighbors, as defined by the network's adjacency matrix. This iterative exchange allows information to propagate throughout the network over multiple rounds. The agents exchange all model parameters, which includes the weights of both the ResNet-18 backbone and the VICReg projection head.

For evaluation, each agent is periodically tested on a global test set using k-NN, to examine how exchanging model parameters can teach the network of agents to learn meaningful representations for classes they have not encountered directly. At the end of training, each agent is also tested on a global task through linear evaluation and t-SNE. Unlike our method for domain shift 2.3.4, here each agent learns its own personal linear classifier.

Personalized Learning for Domain Adaptation

This path investigates whether P2P communication can help agents learn specialized models that are robust to their local domain shifts. The data is partitioned such that all agents share the same classes, but each agent's data is subject to a unique domain shift (either artificially induced for CIFAR-10 or naturally as in Office-Home).

The training loop of local updates followed by gossip averaging remains the same. However, the objective and evaluation protocol are fundamentally different:

- **Objective:** The goal is not to converge to a single model, but for each agent to leverage the knowledge from its peers to improve its own local representations.

- **Evaluation:** Each agent's model is evaluated independently on a global test set with the trained linear classifier. A unique learning curve is thus generated for each agent and averaged. The final reported performance is the mean accuracy across all of these personalized evaluations.

This personalized evaluation framework allows for a direct measurement of the benefit of collaboration, by comparing the performance of agents in a communicating network against a baseline of agents training in complete isolation. For the Office-Home dataset, this protocol was used to test scenarios where agents were assigned unique domains to cleanly isolate the effects of domain adaptation.

While this protocol effectively evaluates personalization, it lacks a mechanism to explicitly reconcile the differing representations learned by each agent, a challenge that is directly addressed by the novel framework proposed in Section 2.3.4.

Gossip Averaging

The communication and aggregation in the decentralized P2P framework are performed using a gossip-based diffusion protocol. Diffusion strategies are a well-established class of algorithms for decentralized optimization that are structured around a two-step process known as Adapt-then-Combine (ATC) [8]. The key characteristic of this strategy is that agents first adapt their models locally using their private data before combining them with information from their neighbors [8].

The implementation in this research directly follows this ATC structure. For each communication round i, the process for an agent k can be formally described by the following two steps:

1. **Adaptation Step:** First, each agent performs a local training update. It takes its current model parameter vector, $w_{k,i-1}$, and computes an intermediate updated vector, $\psi_{k,i}$, by first descending along the objective function J_k:

$$\psi_{k,i} = w_{k,i-1} - \mu \nabla J_k(w_{k,i-1}) \tag{2.8}$$

   In the context of this project, this step corresponds to the agent update function, where each agent trains on its local data for a set number of epochs.

2. **Combination Step:** Next, the agent combines its intermediate model with those received from its neighbors. The final model for the round, $w_{k,i}$, is a weighted average of the inter-

mediate models from its neighborhood $\mathcal{N}_k$:

$$w_{k,i} = \sum_{l \in \mathcal{N}_k} a_{lk} \psi_{l,i} \tag{2.9}$$

This step is implemented by the gossip average function. For simplicity and robustness, a uniform averaging scheme is used, where the combination weights a_{lk} are set to $1/|\mathcal{N}_k|$ for all neighbors $l \in \mathcal{N}_k$. This process acts as a diffusion mechanism, allowing locally learned information to propagate across the network, enabling the agents to iteratively converge towards a consensus without a central server.

2.3.4 Our Novel Method

The preliminary simulation results, which will be detailed in Chapter 3, revealed a fundamental challenge in applying self-supervised learning to decentralized settings. A naive diffusion of the entire local agent models was not very effective in learning with skewed labels but even more so with domain shifts. This was an anticipated outcome, as each agent's SSL algorithm is incentivized to specialize its representations for its own unique data distribution. Consequently, averaging its model parameters with a peer that has learned from different data often dilutes this specialized knowledge, leading to performance that is no better, and sometimes worse, than agents training in complete isolation.

This observation led us to focus specifically on the domain shift scenario, a persistent challenge in real-world applications where a common task must be solved using data from varied acquisition sources. The central research question thus became: *how can specialist agents with different data domains collaboratively train a robust, shared model that can understand all agents' unique perspectives?* Intuitively, the solution required a mechanism to align the agents' latent representation spaces, ensuring that regardless of the input domain, the same semantic classes would be mapped to similar locations in the latent representation space.

This challenge inspired a new communication protocol. While VICReg [13] provides the powerful self-supervised foundation, the work on Federated Unsupervised Representation Learning by Zhang et al [11] provided the key motivation for overcoming the representation misalignment caused by non-IID data domains. The core insight is that agents' locally learned representations, while semantically meaningful on their own, are fundamentally misaligned with one another. Our method seeks to explicitly synchronize these latent spaces before attempting collaboration on a

downstream task.

It is worth noting that this approach fundamentally differs from prior work such as [11] which relies on constructing shared dictionaries. Instead our method uses a small, public anchor dataset to regularize alignment of latent spaces before the classifier collaboration.

The solution is a multi-faceted method that introduces a shared linear classifier trained concurrently with the self-supervised backbones. Contrary to the previous frameworks where linear evaluation was a post-training step, here the classifier is an integral part of the collaborative training loop. The methodology for each communication round can be broken down into the following steps:

- **Step 1: Local Self-Supervised and Supervised Training.** Each agent begins by performing local updates for each batch of data in a sequential two-step process. First, its personalized ResNet-18 backbone is trained using only the unsupervised VICReg loss ($\mathcal{L}_{\text{VICReg}}$) on its private, domain-specific data. Immediately after, the backbone is frozen and its newly learned representations are used to train a local linear classifier in a fully supervised manner with a Cross-Entropy loss ($\mathcal{L}_{\text{CE}}$).

- **Step 2: Representation Alignment via Public Anchoring.** To address the latent space misalignment, agents utilize a small, shared public dataset of 100 images, which acts as a common reference point. Each agent passes these public images through its encoder to generate embeddings. A round of gossip averaging is then performed on these embeddings, allowing each agent to compute a local consensus representation. A mean squared error (MSE) loss ($\mathcal{L}_{\text{MSE}}$) is used to apply a corrective update to the agent's backbone, nudging its latent space towards this shared consensus. This technique, inspired by the alignment module in [11], allows for a privacy-preserving alignment without sharing private data.

- **Step 3: Classifier Collaboration.** After the encoders are better aligned, agents perform a second gossip averaging step, this time on the parameters of their linear classifiers. By iteratively training locally and averaging, the classifiers across the network gradually converge towards a single, robust consensus model capable of interpreting features from any of the specialized domains.

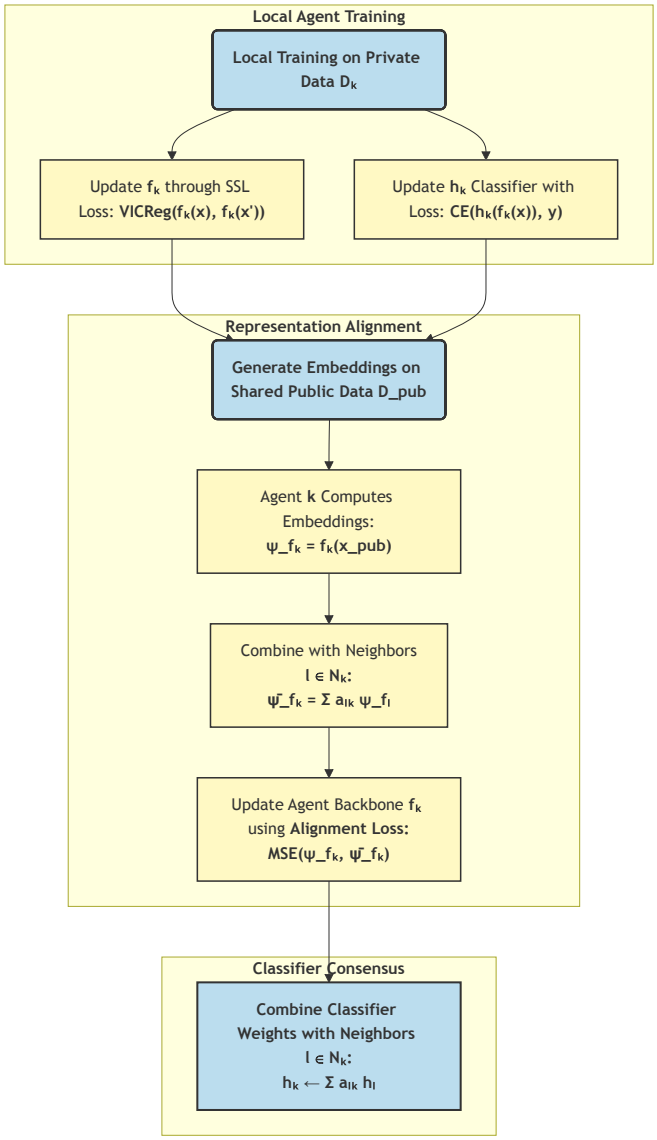


Figure 2.6: A flowchart illustrating the three key stages of a single communication round in our proposed alignment-regularized collaborative protocol: (1) Local Training on private data, (2) Representation Alignment using a shared public dataset, and (3) Classifier Consensus via gossip averaging.

The overall framework is presented visually through a simplified flowchart 2.6, that analyses the three main steps of our method in training. Starting from the local SSL and linear classifier training, before moving to the alignment module using the public dataset and finally communicating on the classifier weights.

In this framework, agents communicate two distinct sets of parameters: feature embeddings from the public dataset for encoder alignment, and the weights of the linear classifier for collaborative model building. The central hypothesis is that the public dataset provides a sufficient anchor

to align the latent spaces, which in turn enables the agents to effectively fuse their knowledge into a high-performing shared classifier. The effectiveness of this novel approach will be evaluated in the subsequent chapter 3.

3

Results & Analysis

Contents

3.1 Result Summary

Here we present the empirical results of the simulation frameworks detailed in the preceding chapter. The experiments are structured to tell a clear story: we first establish performance benchmarks with standard centralized and federated learning, then demonstrate the challenges of decentralized learning under statistical heterogeneity, and finally evaluate our novel collaborative protocol designed to overcome these challenges.

3.2 Benchmarks and Problem Formulation

To provide a clear context for the performance of our decentralized protocol, we first establish upper-bound benchmarks using standard centralized and federated learning models. These initial simulations serve a dual purpose: first, to validate that our implementation of VICReg can learn high-quality representations in idealized settings, and second, to precisely demonstrate the performance degradation caused by the data heterogeneity challenges inherent in multi-agent systems, as highlighted by [11]. This degradation, particularly the inconsistency and misalignment of representation spaces among agents, is the central problem our work aims to solve.

3.2.1 Centralized and Federated Learning on CIFAR-10

The initial set of experiments validates our implementation of the VICReg algorithm and the FedAvg communication protocol on the CIFAR-10 dataset under an IID data distribution. As shown in Figure 3.1, both the centralized and federated learning frameworks demonstrate successful learning, achieving high k-NN classification accuracy and confirming that the models are functioning as expected.

The centralized model, which trains on the entire dataset, serves as the theoretical performance ceiling. Its learning curve (in blue) shows a steady increase in accuracy, reaching a final k-NN accuracy of 72.3% and linear accuracy 74.3%. This result is consistent with established benchmarks for self-supervised learning on CIFAR-10 and validates the quality of the representations learned by VICReg in an ideal setting.

The federated framework, simulated with five agents each receiving an IID partition of the data, also shows robust learning, albeit with a noticeable performance gap compared to the centralized approach. Its learning curve (in orange) follows a similar trajectory but consistently trails the centralized model, achieving a final k-NN accuracy of 66.6% and linear accuracy 68.5%. This gap is expected, as each federated agent trains on only a fraction of the total data, and the periodic FedAvg introduces a communication overhead that can slow convergence. Nevertheless, this result proves that meaningful representations can be learned in a distributed setting when the data is shared uniformly.

Having established these benchmarks and validated our core components, we can proceed to investigate the primary challenge addressed by this thesis: the impact of non-IID data distributions

on multi-agent learning systems. The subsequent experiments will replace this idealized IID setup with more realistic scenarios involving label skew and domain shift.

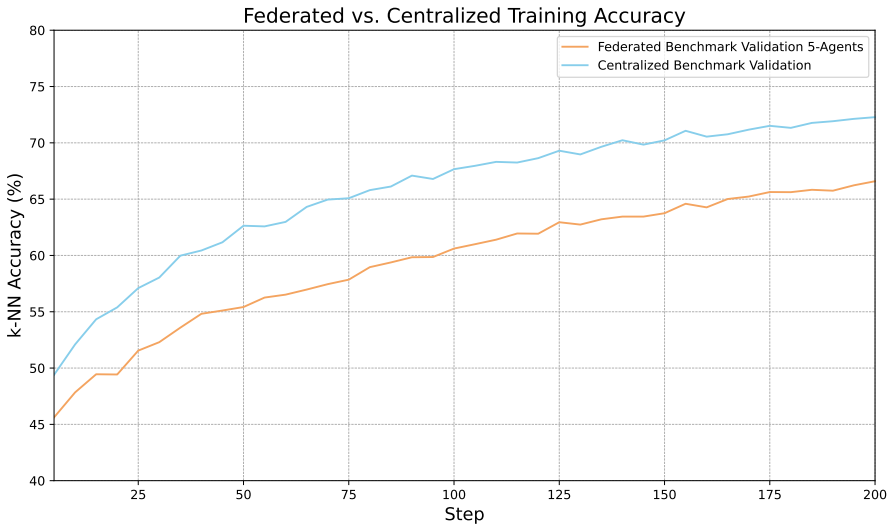


Figure 3.1: Validation of the Centralized and Federated Frameworks on the CIFAR-10 Dataset with an IID data split. The k-NN accuracy is evaluated periodically throughout training.

3.2.2 Impact of Label Skew on Federated Learning

While the IID benchmark confirms the viability of FedAvg, it does not reflect the typical conditions of decentralized systems where data is often statistically heterogeneous. To explicitly quantify this challenge, we analyze the performance of the standard FedAvg algorithm under increasing degrees of label skew. Using the CIFAR-10 dataset, we partitioned the data among five agents using a Dirichlet distribution with a concentration parameter α set to values of 100 (largely IID), 5 (moderate skew), and 0.5 (severe skew). A lower α indicates a higher degree of heterogeneity, with each agent observing a smaller, more specialized subset of the total classes.

Surprisingly, the results depicted in Figure 3.2 show that the global model's performance degrades only marginally as the label skew becomes more severe. The learning curve for the highly skewed case ($\alpha = 0.5$, blue) closely tracks the curves for the moderately skewed ($\alpha = 5$, orange) and nearly IID ($\alpha = 100$, green) scenarios. At the end of 200 communication rounds, the final k-NN accuracies are remarkably close: 66.6% for $\alpha = 100$, 66.2% for $\alpha = 5$, and 62.7% for $\alpha = 0.5$.

This unexpected resilience suggests that the VICReg self-supervised objective is exceptionally powerful. Even when individual agents learn from a narrow subset of classes, they produce highly distinct and semantically rich feature representations. The FedAvg protocol, by averaging

these specialist models, creates a global model that successfully aggregates this diverse knowledge, enabling it to distinguish between all 10 CIFAR classes effectively. This finding is further substantiated by linear evaluation tests, where the final accuracies were 68.5% ($\alpha = 100$), 68.9% ($\alpha = 5$), and 65.7% ($\alpha = 0.5$). The slight outperformance of the $\alpha = 5$ case over the IID baseline may be attributable to the non-IID partitioning acting as a form of implicit data augmentation or regularization, forcing the global model to learn more robust features than it would in a perfectly uniform setting.

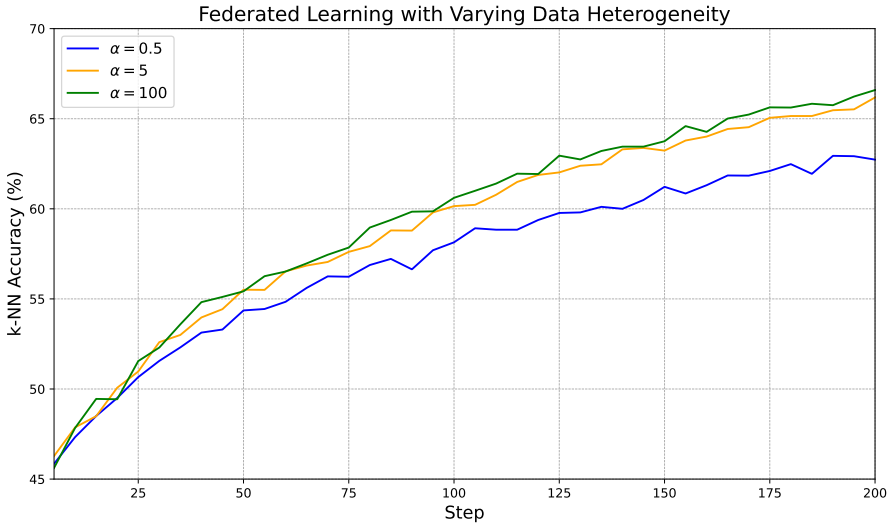


Figure 3.2: k-NN learning performance of the global model under varying degrees of label skew. Performance degradation is minimal, highlighting the robustness of the VICReg objective within a federated framework.

However, this federated setup relies on a central fusion server. The critical next step is to investigate whether this robustness to heterogeneity persists in a fully decentralized, P2P architecture where no central coordinator exists.

3.2.3 Impact of Heterogeneity on a Naive P2P Protocol

Having observed the robustness of VICReg in a federated setting, we now transition to a fully decentralized P2P framework. This section evaluates a baseline P2P protocol where agents, without a central coordinator, directly exchange and average their full model parameters using gossip averaging. This *naive* approach serves to highlight the intensified challenges of decentralized learning and motivate the development of our more sophisticated method. In this paradigm, since there is no global model, performance is measured by averaging the accuracy of each agent evaluated on the global test set.

First, we replicate the label skew experiment from the federated setup, using 5 agents in a fully connected graph with varying Dirichlet concentration parameters (α). As seen in Figure 3.3, the P2P framework is significantly more sensitive to heterogeneity than the federated one. While the nearly IID ($\alpha = 100$) and moderately skewed ($\alpha = 5$) settings perform similarly, achieving 63.4% and 62.8% global accuracy respectively, the highly skewed case ($\alpha = 0.5$) suffers a minor performance drop, at 60.5%. This demonstrates that the diffusion approach of the decentralized case can be quite effective compared to the brute force FedAvg. Although, FedAvg outperforms in absolute numbers, the % gap from moderately to highly skewed is smaller in the P2P case.

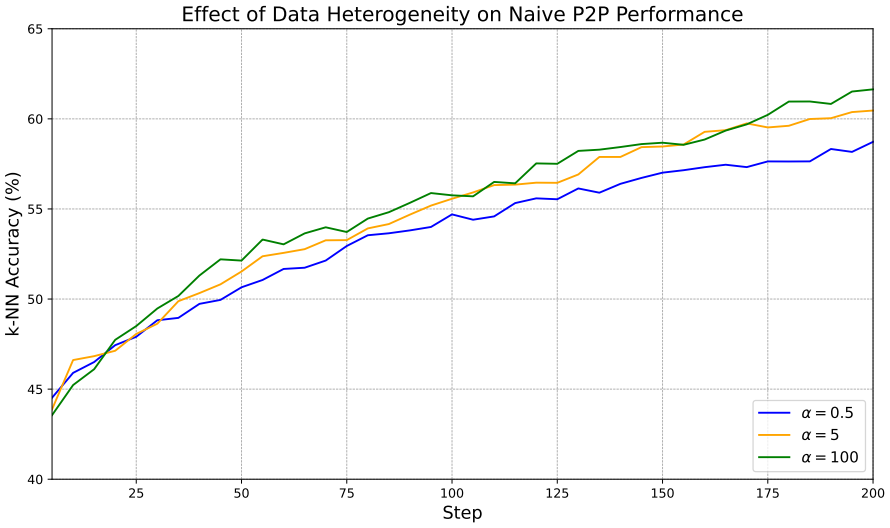


Figure 3.3: Global k-NN evaluation for a 5-agent, fully connected P2P network under varying label skew. Unlike the federated case, performance degrades as heterogeneity increases (lower α).

To further probe the limitations of this naive protocol, we conducted an extensive experiment under high heterogeneity ($\alpha = 0.5$), varying both the number of agents (5, 10 and 15) and the communication topology (fully connected, random, and disconnected). The results, presented in Figure 3.4, reveal a complex relationship between network scale, connectivity, and performance.

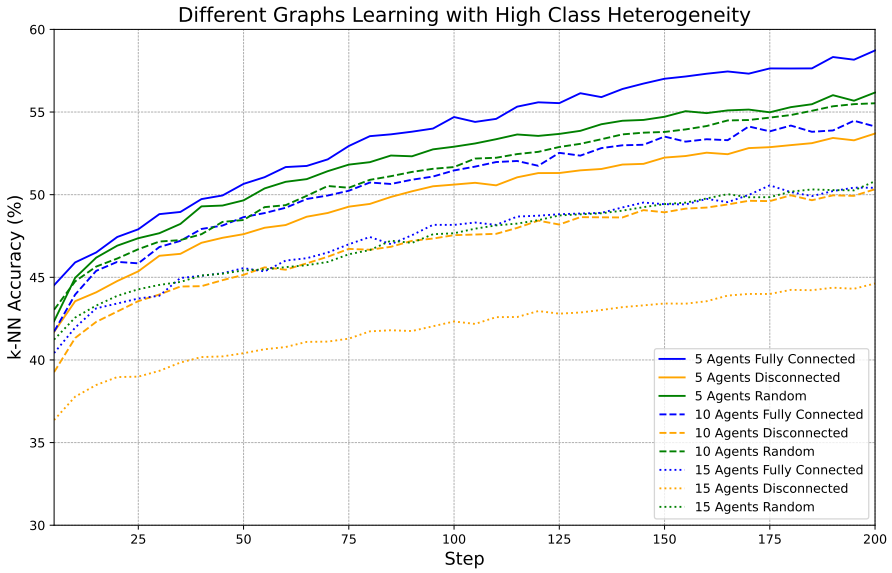


Figure 3.4: Impact of network scale and topology on personalized k-NN performance under high label skew ($\alpha = 0.5$). The results reveal that performance degrades as the number of specialist agents in the network increases.

The simulation results present an interesting trend: while communication is beneficial at any given scale, the overall performance degrades as the number of specialist agents in the network increases. The networks with 5 agents are the clear top performers, with the fully connected (solid blue) and random (solid green) topologies achieving the highest accuracies. Scaling up to 10 agents, communication still outperforms the 5 disconnected agents, but trails the 5 connected agents. A 15 agent network degrades further, though still edging a 10 agent disconnected test, with the 15 disconnected agents having the poorest performance overall.

This performance inversion reveals a fundamental scaling problem for naive P2P learning under high class heterogeneity. At any specific agent count (5, 10 or 15), communication helps; the fully connected usually outperforms the random, which outperforms the disconnected. Nevertheless, for 10 agents, a more sparsely connected network outperformed the fully connected one. In this case, due to the size of the network and the high heterogeneity with only 10 classes in training, it is possible that agents receive batches with only one or two classes. Thus, in a fully connected case, they may skew the learned representations with a stronger bias that hurts the overall performance.

This confirms that sharing parameters is better than training in isolation. However, the negative impact of increasing the number of specialists is a noteworthy effect. As the network grows, the semantic gap between any two agents widens, and their locally learned representations become increasingly misaligned. The naive gossip averaging protocol becomes progressively less effective

at reconciling this fragmented knowledge, leading to degraded performance.

These results reinforce our core hypothesis: communication is not enough. While it provides a benefit over training in isolation, its effectiveness is fundamentally constrained by the representational misalignment that arises from increasing decentralization and heterogeneity.

Table 3.1: Final Personalized Evaluation Accuracies (%) on CIFAR-10 Global Test Set with Label Skew ($\alpha = 0.5$).

Num. Agents	**Topology**	**Linear Evaluation (%)** *(Generalization)*	**k-NN Evaluation (%)** *(Specialization)*
5	Fully Connected	**60.47**	**58.72**
	Random	58.69	56.18
	Disconnected (Baseline)	56.51	53.70
10	Fully Connected	57.52	54.13
	Random	**57.59**	**55.53**
	Disconnected (Baseline)	53.22	50.33
15	Fully Connected	52.71	50.42
	Random	52.79	50.82
	Disconnected (Baseline)	47.93	44.63

Validating the Problem of Domain Shift

The previous experiments demonstrated that naive P2P model averaging struggles with label skew. However, a more challenging and realistic form of heterogeneity is domain shift, where agents observe data with fundamentally different underlying visual characteristics, even if the semantic classes are the same.

To simulate this paradigm, we first created an artificial domain shift scenario using the CIFAR-10 dataset. Each of the five agents was assigned a unique, fixed data augmentation (e.g. heavy Gaussian blur, random rotation) that was applied to its local data *before* the standard VICReg augmentations. This forces each agent to learn from a distinct visual domain. The agents then communicated using the same naive gossip averaging protocol.

The results, shown in Figure 3.5, confirm that this type of heterogeneity is a significant barrier. The learning curves for the communicating agents in a random graph (blue) and the non-

communicating, disconnected agents (orange) are nearly identical, with both achieving a final k-NN accuracy of approximately 49% and a final linear accuracy of 54.3% and 54.8% respectively. The fact that communication provides no benefit over training in isolation is a critical finding. It demonstrates that when faced with domain shift, the naive averaging of model parameters is entirely ineffective at producing a useful collaborative model. The agents' representations are so misaligned that averaging them offers no improvement.

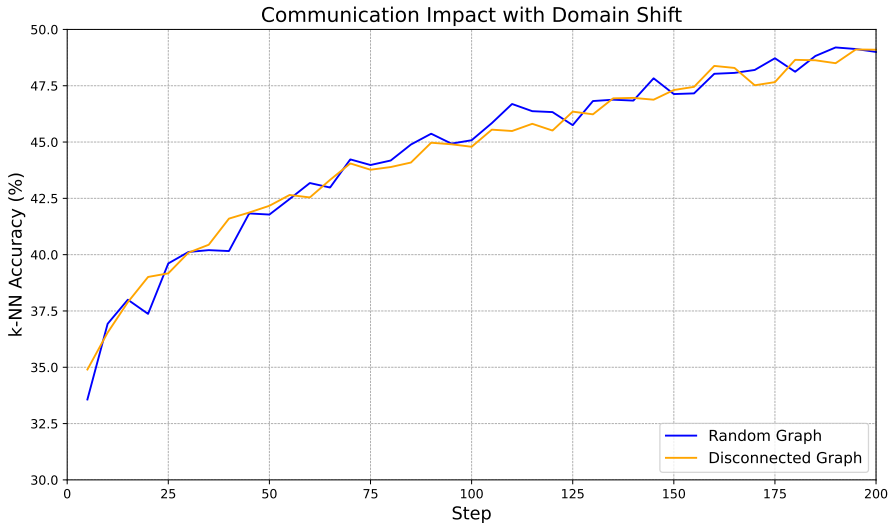


Figure 3.5: Impact of communication on personalized k-NN accuracy under an artificial domain shift on CIFAR-10. The performance of communicating agents (Random Graph) is identical to that of isolated agents (Disconnected Graph), indicating a failure of the naive communication protocol.

This result highlights the limitations of using a naive gossip averaging setup and motivates the need for a more robust framework to properly evaluate this challenge. Consequently, we adopted the OfficeHome dataset **officehome**, which contains images of common objects across four distinct, naturally occurring domains: Art, Clipart, Product, and Real-World. This dataset provides a more realistic and challenging testbed for developing a protocol capable of overcoming domain shift, which will be the focus of the remainder of this chapter.

3.2.4 Failure of Standard Benchmarks on OfficeHome

Having established that naive P2P communication fails to handle domain shift, we now evaluate the standard centralized and federated architectures on the more challenging OfficeHome dataset. This serves to demonstrate that the problem is not merely an artifact of P2P communication, but a fundamental challenge posed by domain-level heterogeneity. For these experiments, the training

architecture remained the same, with a minor adjustment of the learning rate from 0.01 to 0.001 to ensure stable training on the new dataset. In all baseline scenarios, performance is measured by evaluating the final model(s) on the global test set, which contains samples from all four domains, thereby testing for generalization.

The core issue stems from what [11] term the ”inconsistency of representation spaces.” In both centralized and federated settings, the optimization objective implicitly assumes that a single, unified representation space can be learned for all data. However, the distinct visual characteristics of the Art, Clipart, Product, and Real-World domains violate this assumption.

In the **centralized** scenario, the model is trained on a shuffled mixture of all four domains simultaneously. While VICReg is effective at learning semantic features, it also learns powerful domain-specific features. The model becomes confused as it tries to map, for example, a bottle in the Clipart domain and a bottle in the Real-World domain to a similar semantic location while also trying to separate them based on their vastly different stylistic features. This leads to a compromised representation space where domain features interfere with semantic learning. The t-SNE visualization of the centralized model's embeddings in Figure 3.6 provides clear evidence of this issue. While some loose class clusters are visible (e.g. ”bucket” in yellow), the overall structure is too spread out, with significant overlap between classes, indicating that the model failed to learn a well-separated feature space when exposed to all domains at once.

In the **federated** scenario, this problem is exacerbated. We configured the simulation with four agents, each exclusively trained on one of the four domains. While each agent becomes an expert in its local domain, the FedAvg protocol naively averages their model weights. This is fundamentally flawed, as it attempts to find a consensus between models that have learned incongruent feature maps. The representations for Clipart are inherently different from those for product pictures, and averaging the parameters of the models that produce them results in a global model that is proficient in neither domain when tested on the comprehensive global dataset.

As illustrated in Figure 3.7, this fundamental issue leads to a clear performance ceiling for all baseline methods. The centralized (light blue), federated (orange), and a P2P disconnected baseline (red)—all evaluated on the global test set—exhibit volatile and ultimately poor performance, converging to a final accuracy below 50%. Analysis of their exact poor results is found in Table 3.2. None of these standard approaches can overcome the representational inconsistency caused by the domain shift. This failure serves as the primary motivation for our novel protocol, which is designed to constructively align these inconsistent spaces rather than averaging them.

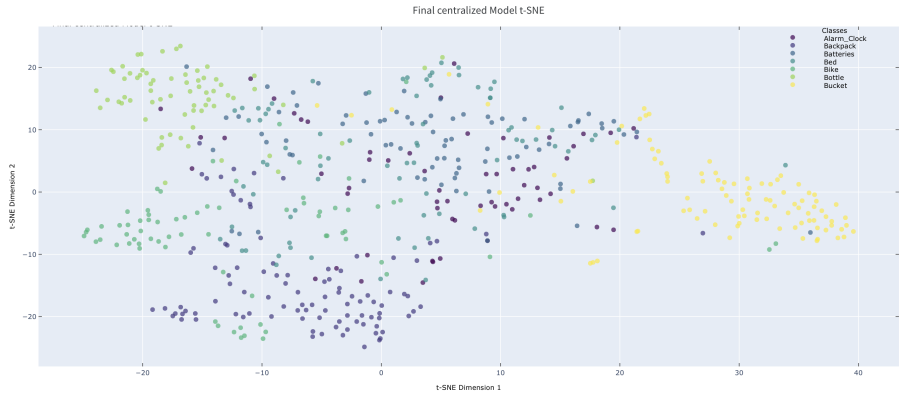


Figure 3.6: t-SNE visualization of the feature space from the centralized model trained on Office-Home. The lack of distinct, well-separated clusters illustrates the model's confusion when learning from multiple domains simultaneously.

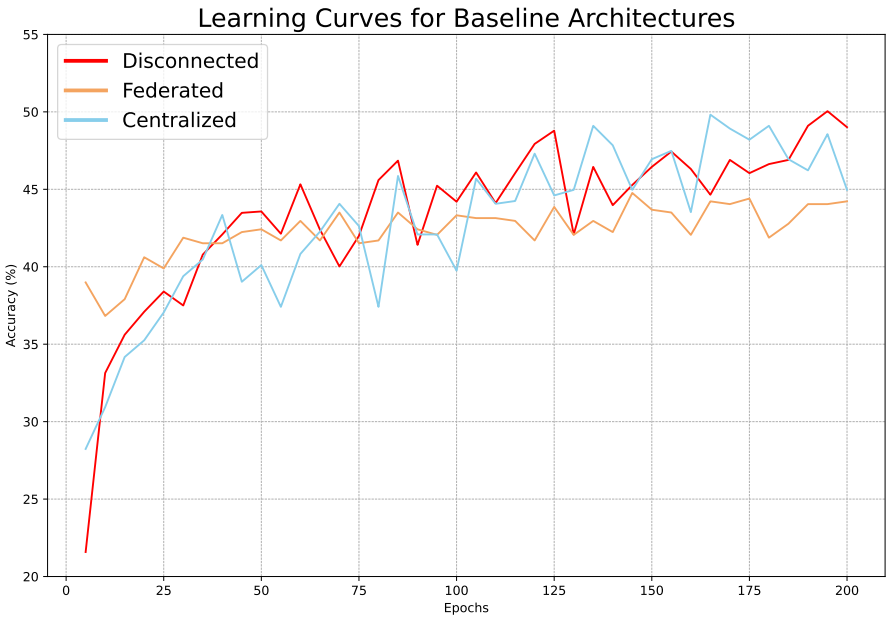


Figure 3.7: Learning curves for baseline architectures on the OfficeHome dataset. All three methods—Centralized, Federated, and Disconnected P2P—demonstrate poor and unstable performance, failing to surpass 50% accuracy due to the challenge of inconsistent representation spaces across different domains.

3.3 Evaluation of the Novel Collaborative Protocol

Having established that standard centralized, federated, and naive P2P methods fail to address the challenge of domain-level heterogeneity, this section presents the core results of this thesis. We evaluate the performance of our novel alignment-regularized collaborative method, conducting the primary experiments on the OfficeHome dataset to demonstrate its effectiveness in overcoming

domain shift. We will first present the direct quantitative improvements over the established baselines, followed by a series of ablation and diagnostic studies to analyze the specific contributions of each component of our method.

3.3.1 Quantitative Performance on OfficeHome

Our primary claim is that the proposed alignment-regularized protocol can significantly outperform standard benchmarks in a decentralized domain-shift scenario. The preceding sections established that all baseline methods fail on the OfficeHome dataset due to the problem of inconsistent representation spaces. Their communication protocols, based on naive parameter averaging, corrupt the specialized knowledge learned by each domain-expert agent. Our method, by contrast, is designed to explicitly align these disparate representation spaces, allowing for effective collaboration without destroying local learning.

The results of a 200-round simulation with four agents, each assigned a unique domain, are presented in Figure 3.8. The performance of our full method on a fully connected graph (labeled "Our Method") is compared against the baselines and a variant of our method using a random communication graph.

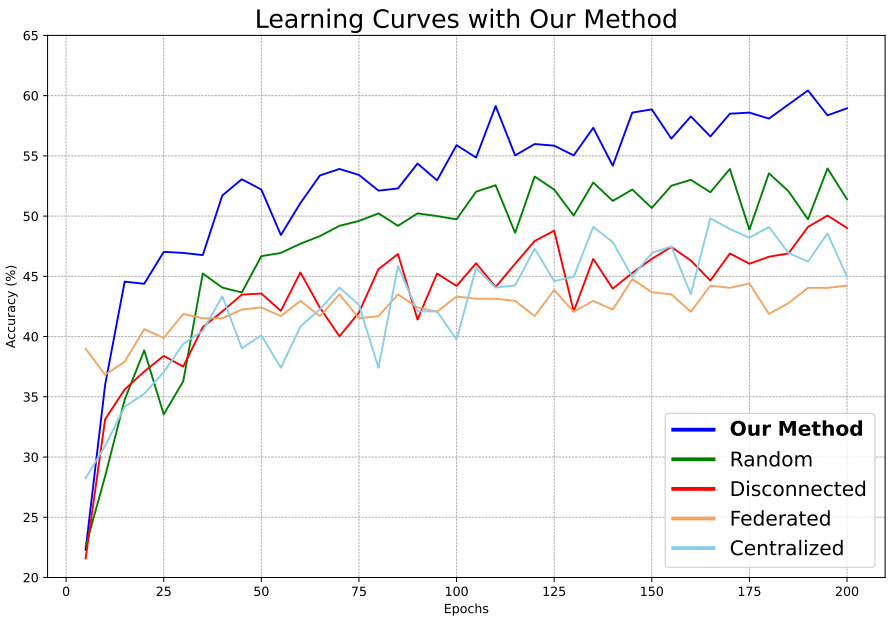


Figure 3.8: Learning curves comparing our novel protocol against baseline architectures on the OfficeHome dataset. The evaluation metric is the average linear classification accuracy across all agents on the global test set.

The results provide clear validation for our approach. As shown by the blue curve, our method

demonstrates superior performance and a more stable learning trajectory, consistently outperforming all baselines. It achieves a final average linear accuracy of **58.9%**, representing a significant improvement of over 10% compared to the centralized, federated, and disconnected agents, which all stagnate below 50%. Furthermore, our protocol with a sparser, random communication graph (green curve) also shows a marked improvement, achieving a final accuracy of **52.1%**, proving that the benefits of the method are not solely dependent on a fully connected topology. This substantial performance gain directly demonstrates that aligning latent spaces and facilitating classifier consensus is a far more effective strategy for collaboration under domain shift than naive parameter averaging.

3.3.2 Ablation Study: Deconstructing Our New Method

To verify that the performance gains are a direct result of our proposed protocol and to understand the individual contributions of its components, we conducted a comprehensive ablation study. We deconstructed the full protocol into its two main constituents: Representation Alignment over the public dataset and Classifier Consensus through diffusion. We compare the full protocol against two variants: one employing only the alignment loss ("Alignment Only") and another using only classifier averaging ("Classifier Sharing Only").

Table 3.2: Ablation study of the method's components on the OfficeHome dataset. The table compares the final average accuracies (Linear and k-NN) of the full method against variants with key components disabled, demonstrating their individual and combined contributions.

Method	**Linear Acc (%)**	**k-NN Acc (%)**
Proposed Method & Ablations		
Our Method (Fully Connected)	**58.9**	**51.9**
Our Method (Random)	52.1	51.7
Ablation: Alignment Only	50.1	50.1
Ablation: Classifier Sharing Only	44.9	51.4
Baseline Prototypes		
Disconnected (No Collaboration)	37.3	51.6
Federated (FedAvg)	47.3	44.2
Centralized	48.9	45.7

The final performance metrics are summarized in Table 3.2, and the learning dynamics are

visualized in Figure 3.9. Both the table and the figure include the full protocol ("Our Method") and the previously established baselines for a comprehensive comparison.

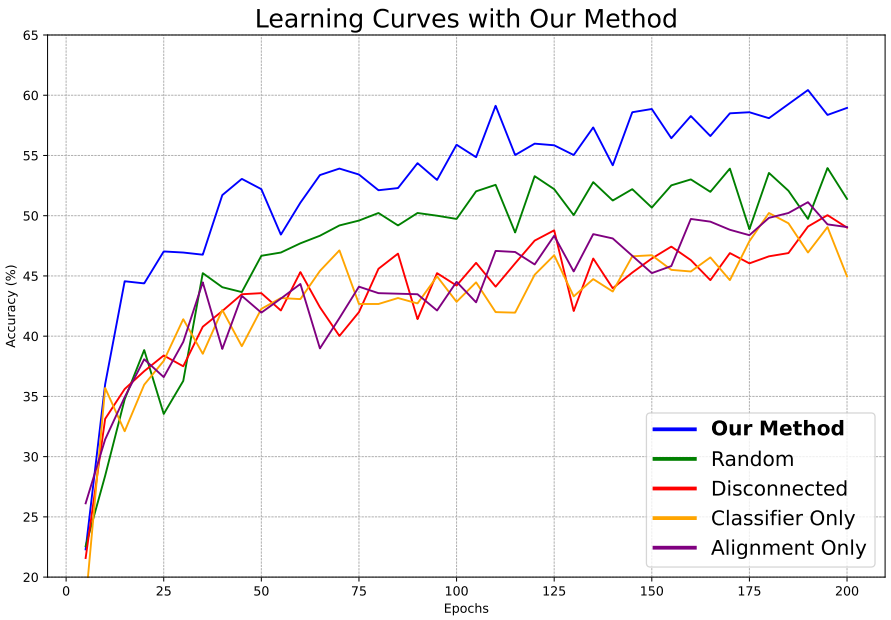


Figure 3.9: Learning curves for the ablation study. Our full method (blue) consistently outperforms all other versions, showing that both alignment (purple) and classifier sharing (orange) are necessary to achieve optimal performance.

The results clearly demonstrate the synergy between the two components. The "Alignment Only" protocol (purple curve, 50.1% final accuracy) substantially outperforms the "Disconnected" baseline (37.3%), proving that aligning the representation spaces is the most critical step for enabling cross-domain generalization. However, it still falls well short of the full method.

Conversely, the "Classifier Sharing Only" protocol (orange curve, 44.9% final accuracy) shows only a modest improvement over the disconnected baseline. Interestingly, it performs worse than the standard Federated and Centralized approaches. This result is crucial: it shows that averaging classifiers without first aligning the underlying feature spaces they are interpreting is an ineffective, and even detrimental, strategy. The shared classifier becomes confused by the inconsistent representations from different agents, as per our hypothesis.

As seen in Figure 3.9, our method achieves the fastest and most stable convergence to the highest final accuracy (58.9%). This confirms our central hypothesis: both representation alignment and classifier consensus are essential, and their combined effect is greater than the sum of their individual contributions. The alignment creates a common ground for representations, and the classifier sharing leverages this common ground to build a robust model that generalizes across

domains.

3.3.3 Qualitative and Diagnostic Analysis

While the quantitative results from the ablation study confirm the effectiveness of our protocol, a deeper analysis is required to understand *how* it achieves these gains. This section delves into the internal mechanics of the method, providing qualitative and diagnostic evidence to validate our core hypothesis: that successful collaboration hinges on the explicit alignment of agent representation spaces. Following the approach of [11], a key aspect of this analysis is tracking the angular alignment between agent representations over time, providing a direct measure of representational consistency.

Representation Alignment Over Time

A central hypothesis of this thesis is that the quantitative performance gains documented above are driven by the successful alignment of the agents' disparate domain latent spaces. To verify this, we provide a direct, diagnostic measurement of this alignment process. We measured the angles between the representations generated by different agents for the exact same batch of public data at every evaluation step throughout training.

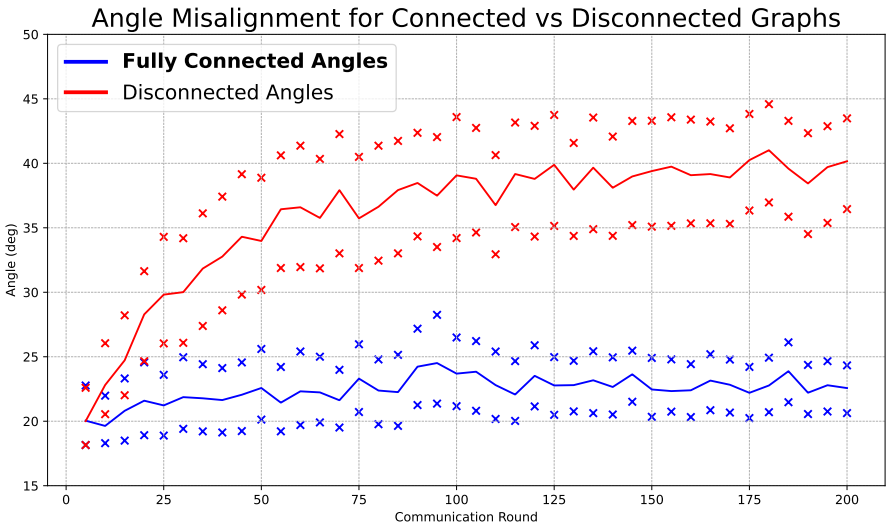


Figure 3.10: Distribution of angles between agent representations of public data over 200 communication rounds. The plot contrasts the behavior of our fully connected method against the disconnected case.

Figure 3.10 visualizes the evolution of the distribution of these angles for two key scenarios:

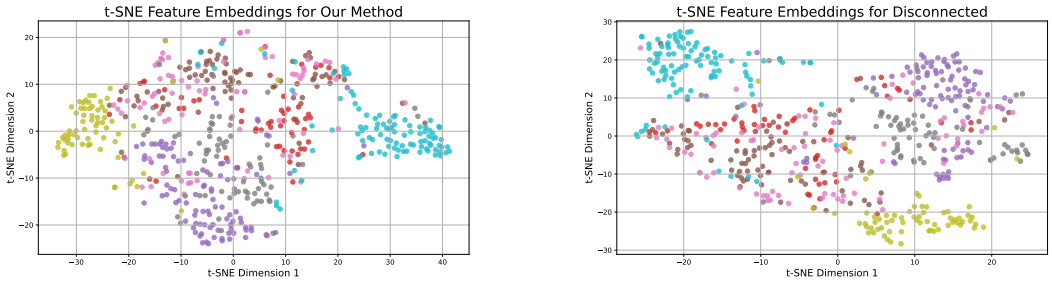
our full method with a fully connected graph (blue) and the disconnected baseline where agents do not communicate (red). The solid line represents the median angle, while the 'x' markers indicate the 25th and 75th percentiles, showing the distribution of angles across all agent pairs.

The results provide a clear and compelling validation of our alignment mechanism. For the disconnected agents, the median angle begins at approximately 20° and steadily increases, exceeding 40° by the end of training. The widening gap between the 25th and 75th percentiles shows that as the agents specialize on their local domains without communication, their representation spaces not only remain misaligned but actively diverge.

On the other hand, the agents communicating with our method exhibit completely different behavior. The median angle remains stable, at around 22° for the entire duration of the training. More importantly, the interquartile range (the distance between the 'x' markers) visibly narrows over time. This demonstrates that the alignment module is not only preventing divergence but is actively forcing the agents' representation spaces to converge towards a consensus. This sustained alignment is precisely the mechanism that allows a shared classifier to be effective, directly explaining the significant performance advantage of our method.

Visualizing the Final Representation Space

To complement the quantitative metrics, we visualize the structure of the learned feature space using t-SNE [30]. This provides a qualitative assessment of how well the different methods can group semantically similar items. Figure 3.11 presents a side-by-side comparison of the global representation space from two agents: one trained with our full collaborative protocol (left) and another trained in complete isolation (right, the "Disconnected" baseline).



(a) Our Method (Fully Connected) (b) Disconnected Baseline

Figure 3.11: Comparison of t-SNE visualizations for the global test set from a single agent's perspective. (a) Representations from an agent trained with our full collaborative protocol show tighter, better-separated class clusters. (b) Representations from an agent trained in isolation show more sparse and overlapping clusters, indicating a less structured feature space.

The visualizations offer compelling evidence for the efficacy of our alignment protocol. The plot for our method (Figure 3.11a) exhibits distinct, well-defined clusters. It is important to acknowledge that a 2D t-SNE plot is a low-dimensional projection of a 512-dimensional space, and some overlap is expected. However, the overall structure is strong; for example, the 'bottle' and 'bucket' classes form tight clusters on opposite ends of the plot. This indicates that the alignment process has produced a robust and structured representation space conducive to linear classification.

In contrast, the feature space of the disconnected agent (Figure 3.11b) is visibly less organized. While some local clustering is apparent—a testament to the power of the underlying VICReg objective—the clusters are more sparse and show significant overlap. This visualization helps to explain a key finding from our ablation study (Table 3.2): despite achieving a high k-NN accuracy (51.6%), which measures local clustering, the disconnected model fails at generalization on the global test set (37.3% linear accuracy).

The reason for this discrepancy is now visually clear. Each disconnected agent successfully clusters its own domain's data, but because their representation spaces are unaligned (as shown by the diverging angles in Figure 3.10), these well-formed local clusters are mapped to inconsistent locations in the global feature space. This makes it impossible for a single linear classifier to find a consistent decision boundary across all domains. Our method resolves this by ensuring that all agents map the same semantic classes to nearby regions in the latent space, which is the key to overcoming the domain shift challenge.

3.3.4 Sensitivity to Alignment Strength Hyperparameter

A critical component of our proposed protocol is the alignment loss term, scaled by the hyperparameter λ_{align}. This coefficient controls the trade-off between enforcing consensus on the shared representation space and allowing agents the flexibility to learn from their local data. A value that is too low may not be sufficient to overcome the domain shift, while a value that is too high could force an overfit, drowning the learning of domain-specific features.

To justify our choice of λ_{align} and to understand its impact on performance, we conducted a sensitivity analysis. We trained the model for 75 communication rounds across a range of λ_{align} values, from 1 to 150. Figure 3.12 plots the average linear accuracy on the global test set throughout training.

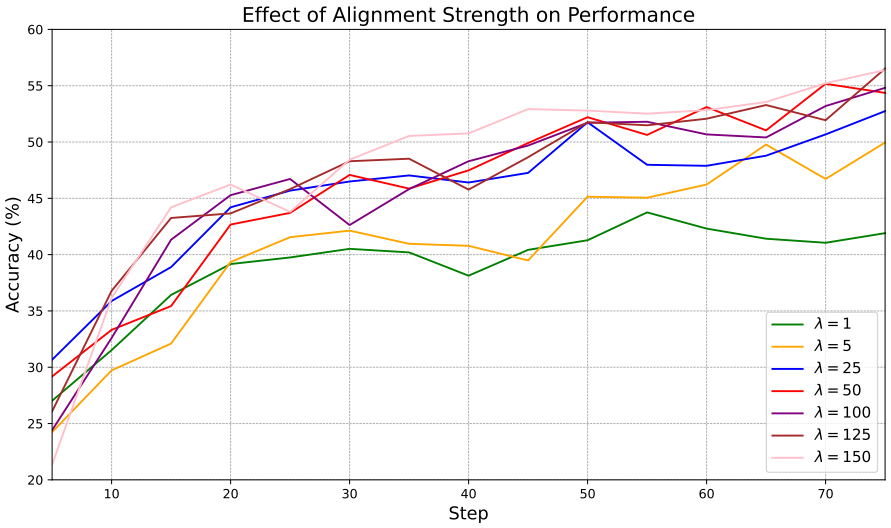


Figure 3.12: Average network linear accuracy over 75 communication rounds for varying alignment strengths (λ_{align}) on the OfficeHome dataset. The analysis shows a pattern of diminishing returns as the hyperparameter value increases.

The results provide a clear rationale for our methodological choice. As illustrated, a very low alignment strength ($\lambda_{align} = 1$, green line) results in the poorest performance, confirming that some alignment pressure is necessary. As the value increases ($\lambda_{align} = 5$, orange line), we see a significant performance boost.

Crucially, the learning curves for $\lambda_{align} \geq 25$ (blue, red, purple, etc.) begin to cluster together. While higher values show a slightly faster initial learning rate, their final performance after 75 rounds is very close to that of $\lambda_{align} = 25$, with values within 53-56% accuracy. This demonstrates a clear point of diminishing returns, as increasing the alignment strength beyond 25 yields little benefit.

Therefore, we selected $\lambda_{align} = 25$ for all experiments. This value is strong enough to enforce meaningful alignment but is also the lowest value to achieve strong performance. It is also small enough to mitigate the risk of overfitting to the alignment task while ensuring agents retain sufficient capacity for local representation learning.

3.3.5 Generalization to Label Skew on CIFAR-10

Having established the effectiveness of our method for the primary challenge of domain shift, we now evaluate how it generalizes to the initial heterogeneity challenge: a skewed label distribution. This series of experiments tests whether a method designed to align disparate domains can also

effectively synthesize knowledge from agents who are specialists on different sets of classes. For this purpose, we return to the CIFAR-10 dataset.

The experimental setup mirrors our initial explorations, with agents' data partitioned according to a Dirichlet distribution with concentration parameters $\alpha = [0.5, 5, 100]$ to represent high, moderate skew and IID conditions, respectively. Figure 3.13 plots the average linear accuracy of collaboration under these conditions, alongside a disconnected baseline where agents with highly skewed data ($\alpha = 0.5$) do not communicate.

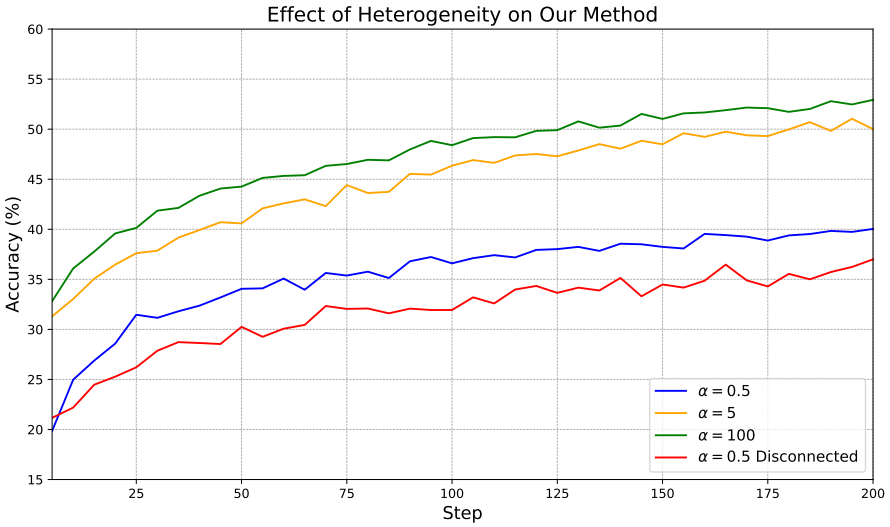


Figure 3.13: Performance of our method on CIFAR-10 under varying degrees of label skew. The plot shows the average linear accuracy on the global test set.

The results demonstrate two key findings. First, our collaborative method provides a substantial benefit over no communication. In the high-skew scenario ($\alpha = 0.5$), the communicating agents (blue line) achieve a final accuracy of approximately 40%, which outperforms the disconnected average accuracy of 37%. This confirms that the alignment and classifier sharing mechanisms are effective at mitigating the most adverse effects of label skew.

Second, the performance of our method is highly sensitive to the degree of skew, and it does not reach the performance achieved by either FedAvg or naive P2P for this specific task (as seen in Figures 3.2 and 3.3). The IID case ($\alpha = 100$, green line) performs best, reaching 52.9% accuracy. Performance degrades with moderate skew ($\alpha = 5$, orange line) to 50%, and further to 40% with high skew ($\alpha = 0.5$, blue line).

In the k-NN evaluation, similar to the results for the OfficeHome dataset, all simulations, even with no communication perform similarly. The uniform IID peaks in performance at 56.6%, while

the disconnected case trails at 53.5%, with the rest laying within this range.

As we showed in Section 3.3.3, we also notice the angle alignment distribution between the agent representations being effective. With no communication the representation angles diverge, while in the highly skewed case they decrease by almost 10°. This verifies that the alignment module helps coordinate agents' latent spaces so that the shared classifier can learning from their diverse class learning.

Overall, these results lead to a nuanced conclusion. While our method was specifically designed for the challenges of domain shift, it generalizes reasonably well to label skew, outperforming a naive disconnected baseline. However, it is not as robust to this type of heterogeneity as the federated approach, whose server-centric architecture is inherently more efficient at aggregating knowledge from class specialists. Furthermore, it does not come close to matching the naive P2P diffusion performance, for the same reason as FedAvg. This finding helps to define the specific application space where our decentralized alignment protocol is most impactful.

This result does not come as a surprise. The problem our method sought to solve was creating unique specialist representations within each agent, which can be translated as a form of multi-task learning where each agent solves a different problem. One learns to interpret art while another interprets clipart. On the other hand, in label heterogeneity, all agents are striving to solve the same problem. Which is simply to map all the classes in unique representations. Therefore, the naive collaboration seen in the federated and naive P2P scenario benefit the specific problem application.

3

4 Impact

Contents

4.1 Environmental and Social Impact

In accordance with engineering council guidelines, this section assesses the environmental and societal impact of the proposed decentralized representation learning solution. The analysis considers the entire life-cycle of the process, from the computational resources required for training to the potential real-world applications of the developed methods.

4.1.1 Environmental Impact Assessment

The environmental impact of this project presents a duality: the immediate, tangible cost of computation versus the long-term potential for creating more energy-efficient and sustainable ML methodologies.

Adverse Impacts

- **Energy Consumption of High-Performance Computing (HPC):** The most significant direct environmental impact is the consumption of electrical energy by the HPC cluster. Training deep learning models is computationally intensive, and the GPUs used consume a substantial amount of power and water needed for cooling. This contributes directly to the carbon footprint of AI [32], a growing concern that has been extensively documented in recent literature [33], [34]. Each simulation run, though necessary for research, adds to this cumulative energy demand.
- **Hardware Life-Cycle:** The research relies on specialized hardware like GPUs, which has its own considerable environmental footprint. This encompasses the entire life-cycle of the device, from raw material extraction and the energy-intensive manufacturing process to its eventual disposal as electronic waste [35].

Positive Impacts and Mitigation Strategies

The project's core methodologies—decentralization and self-supervision—are themselves powerful tools for mitigating the traditional environmental costs associated with large-scale AI.

- **Mitigating Data Transmission Costs through Decentralization:** In a conventional centralized training paradigm, vast quantities of raw data must be transmitted from edge devices to a central data center. This data transfer is not only a communication bottleneck but also a significant consumer of energy [36]. The peer-to-peer, decentralized approach explored in this thesis fundamentally mitigates this issue. By training models directly on the devices where data is collected, the need for large-scale, energy-intensive data transmission is drastically reduced, leading to a more sustainable training paradigm over the system's operational life-cycle.
- **Reducing Resource Waste via Self-Supervised Learning:** The conventional supervised learning pipeline is predicated on the availability of large, manually annotated datasets. This annotation process is not only labor-intensive but also has a substantial computational and resource footprint involving data storage, processing platforms, and human management systems. By developing robust SSL methods that learn directly from raw, unlabeled data, this project contributes to a more resource-efficient pre-training process, minimizing the energy and human capital costs associated with the data labeling phase.

- **Efficient Resource Utilization:** This research was conducted on a shared High Performance Computing (HPC) cluster, which represents an optimized, professionally managed resource. This is inherently more energy-efficient than utilizing disparate, less powerful individual machines. Furthermore, the practice of methodical experimentation—validating algorithms on smaller models (ResNet-18) and datasets (CIFAR-10) before scaling to more complex scenarios—was a conscious effort to minimize computational waste and use the allocated resources responsibly, in line with the principles of "Green AI" [34].

4.1.2 Societal Impact Assessment

The societal implications of the systems investigated in this thesis primarily revolve around data privacy, the democratization of model development, and the nuanced challenges of algorithmic fairness and security in a collaborative peer-to-peer framework.

Positive Impact

- **Enhancing Data Privacy and Security:** This research reinforces one of the most significant advantages of distributed optimization. By enabling collaborative model training without the need to centralize raw, sensitive data, the proposed methods offer a robust privacy-preserving solution. This has profound implications for fields such as medical diagnostics, where multiple hospitals, each with imaging equipment from different manufacturers (i.e. different domains), could collaboratively train a superior diagnostic model on their respective patient scans without ever sharing the underlying private health information. Similarly, in precision agriculture, a network of farms could use low-cost edge devices to build a shared model for crop disease identification from imagery captured under varied lighting and weather conditions, enhancing food security without compromising proprietary operational data.

- **Democratization of ML and Reduced Labor Costs:** The reliance on SSL fundamentally lowers the barrier to entry for developing sophisticated ML. The cost and time required for manual data annotation are often prohibitive for small enterprises with minimal resources. By developing methods that learn powerful representations directly from unlabeled data, this project contributes to the democratization of state-of-the-art ML. The collaborative aspect further empowers smaller entities, allowing them to pool their unlabeled, domain-specific

data to construct a powerful, generalized model that no single participant could have built in isolation.

- **Robustness to Real-World Conditions:** A core objective of this research was to develop models capable of overcoming domain shift. By demonstrating that agents can learn from heterogeneous data sources—simulating varied camera sensors, artistic styles, or lighting conditions—this project directly contributes to the creation of ML systems that are more robust and reliable when deployed in the unpredictable and variable conditions of the real world.

Potential Adverse Impact

The introduction of a representation alignment protocol and a shared classifier modifies the risk landscape compared to simpler FedAvg.

- **Algorithmic Bias via the Alignment Anchor:** While the proposed method of aligning personalized backbones avoids the classic "tyranny of the majority" [37] problem, it introduces a new, critical point of potential bias: the public anchor dataset. The alignment process assumes this small, shared dataset is a fair and representative anchor for the shared latent space. If this anchor dataset is biased—for instance, if it underrepresents certain demographics, environments, or device types—the alignment process could systematically pull all personalized models towards a biased representation. The resulting shared classifier, though trained on diverse private data, would learn to interpret features through this biased lens, potentially performing poorly for agents whose local data domains are dissimilar to the public anchor.

- **Privacy Threats:** A peer-to-peer network remains vulnerable to malicious actors. The novel protocol introduces specific vulnerabilities beyond general model poisoning. A malicious agent could disrupt the network by broadcasting manipulated embeddings during the representation alignment step, subtly skewing the shared latent space and degrading global performance. Furthermore, during the classifier consensus step, a malicious actor could repeatedly inject corrupted classifier weights into the gossip averaging protocol, preventing the shared model from converging or biasing its decision boundary.

Mitigation Strategies

- **For Bias and Fairness:** The primary mitigation strategy shifts from monitoring the global model to ensuring the integrity of the alignment anchor. The public dataset must be meticulously curated and audited for balance and representativeness across all expected participant domains and sensitive attributes. Furthermore, the performance of the final shared classifier must be evaluated not only globally but also on a per-agent or per-domain basis to detect any performance disparities that may indicate an alignment bias. The fact that agents retain their personalized backbones provides a degree of resilience, as the final specialized representation is a function of both the local data and the alignment signal, not just the latter.

- **For Security:** While robust aggregation methods can offer some protection against outlier manipulation, a more formally grounded approach is required to defend against both poisoning attacks and privacy inference threats. The principles of Differential Privacy, as pioneered by Dwork et al provide a powerful framework for such mitigation [38]. In this context, instead of sharing exact parameters, each agent would introduce a carefully calibrated amount of statistical noise (e.g. drawn from a Laplace or Gaussian distribution) to its updates before transmission. This perturbation would apply to both the embeddings shared during the alignment step and the classifier weights during the consensus step. The added noise provides a mathematical guarantee that the contribution of any single agent's data is statistically obscured. This has two key benefits: it makes it computationally infeasible for a malicious actor to infer sensitive information about an agent's private dataset from its public updates, and it degrades the efficacy of targeted poisoning attacks by masking the precise impact of any single malicious update. The central challenge, and a key avenue for future work, lies in tuning the noise level to balance the trade-off between the strength of the privacy guarantee and the convergence speed of the final shared model.

4

Conclusions and Future Directions

4.2 Conclusion

This project began with the objective of exploring how a robust self-supervised learning method, VICReg, would scale within federated and fully decentralized multi-agent frameworks. The primary anticipated challenge was the impact of non-IID data, specifically semantic heterogeneity or label skew, on collaborative model training. However, the research evolved significantly from this starting point, uncovering the fundamental challenge of multi-domain learning and culminating in the development of our novel method to address it.

Through a series of rigorous simulations, this thesis presents several key findings. First, it was established that standard distributed learning protocols are surprisingly resilient to label skew. In the federated learning framework, a global model trained with highly specialized agents (seeing only a fraction of the total classes) experienced a performance degradation of less than 4% compared to a centralized baseline. While the performance of a naive P2P framework degraded by less than 3%. Although it was more sensitive to the number of agents and their network topology, the results consistently demonstrated that communication provided a distinct advantage over agents training in complete isolation.

The pivotal insight, however, emerged from the investigation into domain shift. It was demonstrated that naive P2P communication, which relies on direct parameter averaging, is entirely ineffective when agents are trained on data from different visual domains. This protocol provided no performance benefit over a disconnected network, revealing that representational misalignment is a more significant barrier to effective collaboration than label skew. This finding directly motivated the central contribution of this thesis: a novel communication method incorporating two key modules for representation alignment and classifier consensus.

We prove the efficacy of this novel method, showing that it outperforms all established baselines—centralized, federated, and disconnected P2P—by over 10% on the challenging, multi-domain OfficeHome dataset. A systematic ablation study validated that both the alignment and classifier sharing modules were essential to this success. Further diagnostic analysis, measuring

the angular alignment between agent representations, provided direct evidence that our method successfully forces the agents' disparate latent spaces to converge, creating the shared "visual grammar" necessary for effective collaboration. Through t-SNE decomposition, we further validated that VICReg can cluster tightly each class in an unsupervised way, even when classes range from all four learned domains in the network.

Finally, the novel method was evaluated on the initial challenge of label skew. While it outperformed a disconnected baseline, its performance significantly trailed that of the naive P2P framework. This highlights a key conclusion: the proposed method is highly optimized for the multi-task scenario of domain adaptation, where agents are specialists in domain interpretation, however, it is not well-suited (at this current state of development) to the case of non-IID class distribution.

4.3 Challenges & Limitations

The development of the proposed methods was an iterative process that involved significant exploratory work and overcoming several key challenges. Here we reflect on the primary difficulties encountered and the resulting limitations of this study.

A foundational challenge was the initial selection of a suitable SSL algorithm. An extensive literature review was conducted, followed by prototyping with prominent methods including SimCLR [4], SwAV [17], and SimSiam [21]. The final selection of VICReg [13] was a strategic decision based on its architectural simplicity and robustness, particularly its suitability for multi-agent optimization by performing well with smaller batch sizes and avoiding the need for large memory banks. Subsequently, establishing a stable and high-performing centralized baseline for VICReg, as detailed in Section 2.3.1, proved to be a non-trivial task, requiring extensive debugging and tuning to ensure stable benchmark performance.

The most significant undertaking was the iterative design and implementation of the various multi-agent simulation frameworks. This process involved developing, testing, and prototyping numerous experimental setups before arriving at the final design. The initial tests focused on a naive P2P gossip averaging of the model representation backbones (Res-Net18 weights) to form a single consensus model under label skew, similar to a decentralized FedAvg. This proved to be a limitation, and the framework was re-engineered into personalized label skew, which instead evaluated each agent's model individually on a global test set, as described in Section 3.2.3. The

investigation into domain shift began with applying virtual augmentations to each agent's data, but this was deemed not robust enough. Consequently, the project transitioned to the OfficeHome dataset, where two modes were tested with the naive P2P protocol: sharing all domains randomly and assigning each agent only one domain, with the latter proving to be the more insightful approach. Furthermore, a complex hierarchical domain split was designed as a prototype, where the network is structured into distinct neighborhoods of agents. Within each neighborhood, agents specialize on a single domain, while the class distribution is skewed between neighborhoods. This setup, while designed to test both heterogeneity challenges simultaneously, was ultimately set aside as it would have derailed from the main research mission and was left as a direction for future work. It was this rigorous, iterative exploration—and the observed limitations of all naive protocols—that directly motivated the development of our novel method 2.3.4.

This research journey also imposed several limitations on the final work. The computational cost of running numerous multi-agent simulations necessitated the use of smaller-scale datasets (CIFAR-10 and a 10-class subset of OfficeHome). Consequently, while the findings are robust within this context, their direct generalization to large-scale datasets such as ImageNet remains an open question for future work. Furthermore, a key limitation of our novel protocol is its reliance on a small, labeled public dataset to facilitate classifier training. This requirement, while effective, represents a departure from a purely unsupervised paradigm and implies a dependency on some ground-truth data being available for the collaborative process to succeed.

4.4 Future Work

The novel method developed in this thesis successfully demonstrates that decentralized agents can overcome domain shift by aligning their latent representation spaces and co-training a shared classifier. This work, opens up several promising avenues for future research that can build directly upon its findings.

4.4.1 Transitioning to a Semi-Supervised Framework via Pseudo-Labeling

A primary direction for future work is to enhance the label efficiency of the proposed method, moving it closer to a truly unsupervised paradigm. While the current framework is label-efficient—by training the complex backbone via SSL—the classifier component still relies on labeled private datasets for supervised updates. The logical next step is to transition this into a semi-supervised

framework that operates with only a fraction of the labels. This could be achieved by implementing a *kNN-based pseudo-labeling* protocol, a technique that has shown promise in centralized settings [17]. The process within our multi-agent framework would be as follows:

- Each agent would be initialized with a very small, labeled seed set (e.g., 1% of its training data) which would be used to create an initial feature memory bank.
- For the vast majority of unlabeled private data, each agent would use k-NN classification against its memory bank to generate high-confidence "pseudo-labels" for its unlabeled samples.
- The local linear classifier would then be trained using a combined loss on both the ground-truth labels from the seed set and the machine-generated pseudo-labels.

This approach would more fully realize the goal of reducing the reliance on manual annotation. However, it introduces new challenges. A key area for investigation would be managing the risk of error propagation, where incorrect pseudo-labels are reinforced over successive training rounds, leading to a degradation in classifier performance. Techniques such as confidence-based thresholding for pseudo-label acceptance or incorporating consistency regularization would need to be explored. Furthermore, the computational overhead of performing k-NN inference for every batch during training would require efficient implementation to remain practical.

4.4.2 Exploring Complex, Hierarchical Heterogeneity

The current work established strong baselines for label skew and domain shift as independent challenges. A natural and more realistic extension would be to investigate performance in more complex, mixed-heterogeneity environments. As mentioned briefly in 4.3, the hierarchical data partitioning scheme—where agents within a "neighborhood" face a domain shift, while different neighborhoods are subject to label skew—was designed but not fully explored.

Validating and analyzing the performance of the novel alignment method in such a challenging scenario would test its robustness and limits. A key research question would be whether the representation alignment, which proved effective for domain shift, is sufficient to bridge the semantic gap between neighborhoods that are specialists on entirely different sets of classes. This would likely require a more sophisticated communication topology than simple gossip averaging.

4.4.3 Advanced Communication and Aggregation Strategies

The current framework relies on a simple and robust uniform diffusion protocol. Future research could explore more sophisticated communication and aggregation strategies to improve efficiency and performance. This could include:

- **Adaptive Topologies:** Investigating dynamic network topologies where communication links are formed or pruned based on metrics like representation similarity, allowing agents to preferentially learn from their most relevant peers [39].
- **Weighted Aggregation:** Implementing more advanced aggregation algorithms that move beyond a simple average. For instance, agents could weight contributions from their peers based on trust, confidence scores, or the amount of novel information a peer's update provides. This could lead to faster convergence and a more robust consensus, particularly in the presence of noisy or less reliable agents [40].
- **Communication Efficiency:** For edge-computing scenarios, reducing communication overhead is critical. Reducing the precision of the communicated weights through quantization or exchanging only the most significant parameter updates could be explored to balance performance with communication costs [41].

A

Appendix

Please find the codebase of the project on Github, along with a ReadMe going over the instructions on how to simulate the results at: https://github.com/spacebasie/multiagent-ssl.git

A

Bibliography

[1] Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013, ISSN: 1939-3539. DOI: 10.1109/TPAMI.2013.50. [Online]. Available: https://ieeexplore.ieee.org/document/6472238/metrics (visited on 07/15/2025).

[2] Y. LeCun and I. Misra, *Self-supervised learning: The dark matter of intelligence*. [Online]. Available: https://ai.meta.com/blog/self-supervised-learning-the-dark-matter-of-intelligence/.

[3] R. Balestriero, M. Ibrahim, V. Sobal, *et al.*, *A Cookbook of Self-Supervised Learning*, arXiv:2304.12210 [cs], Jun. 2023. DOI: 10.48550/arXiv.2304.12210. [Online]. Available: http://arxiv.org/abs/2304.12210 (visited on 06/23/2025).

[4] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, *A Simple Framework for Contrastive Learning of Visual Representations*, arXiv:2002.05709 [cs], Jul. 2020. DOI: 10.48550/arXiv.2002.05709. [Online]. Available: http://arxiv.org/abs/2002.05709 (visited on 06/03/2025).

[5] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, *Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty*, arXiv:1906.12340 [cs], Oct. 2019. DOI: 10.48550/arXiv.1906.12340. [Online]. Available: http://arxiv.org/abs/1906.12340 (visited on 07/15/2025).

[6] Y. Tian, D. Krishnan, and P. Isola, *Contrastive Multiview Coding*, arXiv:1906.05849 [cs], Dec. 2020. DOI: 10.48550/arXiv.1906.05849. [Online]. Available: http://arxiv.org/abs/1906.05849 (visited on 06/03/2025).

[7] S. Boyd, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," en, *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010, ISSN: 1935-8237, 1935-8245. DOI: 10.1561/2200000016. [Online]. Available: http://www.nowpublishers.com/article/Details/MAL-016 (visited on 09/01/2025).

[8] S. Vlaski, "Lecture Notes: Distributed Optimization & Learning," 2025.

[9] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, *Communication-Efficient Learning of Deep Networks from Decentralized Data*, arXiv:1602.05629 [cs], Jan. 2023. DOI: 10.48550/arXiv.1602.05629. [Online]. Available: http://arxiv.org/abs/1602.05629 (visited on 09/01/2025).

[10] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, *Communication-Efficient Learning of Deep Networks from Decentralized Data*, arXiv:1602.05629 [cs], Jan. 2023. DOI: 10.48550/arXiv.1602.05629. [Online]. Available: http://arxiv.org/abs/1602.05629 (visited on 08/17/2025).

[11] F. Zhang, K. Kuang, Z. You, *et al.*, *Federated Unsupervised Representation Learning*, arXiv:2010.08982 [cs], Oct. 2020. DOI: 10.48550/arXiv.2010.08982. [Online]. Available: http://arxiv.org/abs/2010.08982 (visited on 05/22/2025).

[12] R. Nassif, S. Vlaski, C. Richard, J. Chen, and A. H. Sayed, "Multitask learning over graphs: An Approach for Distributed, Streaming Machine Learning," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 14–25, May 2020, arXiv:2001.02112 [eess], ISSN: 1053-5888, 1558-0792. DOI: 10.1109/MSP.2020.2966273. [Online]. Available: http://arxiv.org/abs/2001.02112 (visited on 07/18/2025).

[13] A. Bardes, J. Ponce, and Y. LeCun, *VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning*, arXiv:2105.04906 [cs], Jan. 2022. DOI: 10.48550/arXiv.2105.04906. [Online]. Available: http://arxiv.org/abs/2105.04906 (visited on 06/23/2025).

[14] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, *Momentum Contrast for Unsupervised Visual Representation Learning*, arXiv:1911.05722 [cs], Mar. 2020. DOI: 10.48550/arXiv.1911.05722. [Online]. Available: http://arxiv.org/abs/1911.05722 (visited on 06/03/2025).

[15] A. v. d. Oord, Y. Li, and O. Vinyals, *Representation Learning with Contrastive Predictive Coding*, arXiv:1807.03748 [cs], Jan. 2019. DOI: 10.48550/arXiv.1807.03748. [Online]. Available: http://arxiv.org/abs/1807.03748 (visited on 06/03/2025).

[16] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, *Deep Clustering for Unsupervised Learning of Visual Features*, arXiv:1807.05520 [cs], Mar. 2019. DOI: 10.48550/arXiv.1807.05520. [Online]. Available: http://arxiv.org/abs/1807.05520 (visited on 08/16/2025).

[17] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, *Unsupervised Learning of Visual Features by Contrasting Cluster Assignments*, arXiv:2006.09882 [cs], Jan. 2021. DOI: 10.48550/arXiv.2006.09882. [Online]. Available: http://arxiv.org/abs/2006.09882 (visited on 06/11/2025).

[18] M. Cuturi, *Sinkhorn Distances: Lightspeed Computation of Optimal Transportation Distances*, arXiv:1306.0895 [stat], Jun. 2013. DOI: 10.48550/arXiv.1306.0895. [Online]. Available: http://arxiv.org/abs/1306.0895 (visited on 08/16/2025).

[19] G. Hinton, O. Vinyals, and J. Dean, *Distilling the Knowledge in a Neural Network*, arXiv:1503.02531 [stat], Mar. 2015. DOI: 10.48550/arXiv.1503.02531. [Online]. Available: http://arxiv.org/abs/1503.02531 (visited on 06/24/2025).

[20] J.-B. Grill, F. Strub, F. Altché, *et al.*, *Bootstrap your own latent: A new approach to self-supervised Learning*, arXiv:2006.07733 [cs], Sep. 2020. DOI: 10.48550/arXiv.2006.07733. [Online]. Available: http://arxiv.org/abs/2006.07733 (visited on 06/23/2025).

[21] X. Chen and K. He, *Exploring Simple Siamese Representation Learning*, en, arXiv:2011.10566 [cs], Nov. 2020. DOI: 10.48550/arXiv.2011.10566. [Online]. Available: http://arxiv.org/abs/2011.10566 (visited on 06/19/2025).

[22] S. Gidaris, A. Bursuc, G. Puy, N. Komodakis, M. Cord, and P. Pérez, *OBoW: Online Bag-of-Visual-Words Generation for Self-Supervised Learning*, arXiv:2012.11552 [cs], Oct. 2021. DOI: 10.48550/arXiv.2012.11552. [Online]. Available: http://arxiv.org/abs/2012.11552 (visited on 08/16/2025).

[23] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, *Barlow Twins: Self-Supervised Learning via Redundancy Reduction*, arXiv:2103.03230 [cs], Jun. 2021. DOI: 10.48550/arXiv.2103.03230. [Online]. Available: http://arxiv.org/abs/2103.03230 (visited on 06/23/2025).

[24] A. Ermolov, A. Siarohin, E. Sangineto, and N. Sebe, *Whitening for Self-Supervised Representation Learning*, arXiv:2007.06346 [cs], May 2021. DOI: 10.48550/arXiv.2007.06346. [Online]. Available: http://arxiv.org/abs/2007.06346 (visited on 08/16/2025).

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, ISSN: 1063-6919, Jun. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848. [Online]. Available: https://ieeexplore.ieee.org/document/5206848/ (visited on 09/10/2025).

[26] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," en,

[27] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, *Deep Hashing Network for Unsupervised Domain Adaptation*, arXiv:1706.07522 [cs], Jun. 2017. DOI: 10.48550/arXiv.1706.07522. [Online]. Available: http://arxiv.org/abs/1706.07522 (visited on 08/19/2025).

[28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[29] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, arXiv:1412.6980 [cs], Jan. 2017. DOI: 10.48550/arXiv.1412.6980. [Online]. Available: http://arxiv.org/abs/1412.6980 (visited on 08/19/2025).

[30] T. T. Cai and R. Ma, *Theoretical Foundations of t-SNE for Visualizing High-Dimensional Clustered Data*, arXiv:2105.07536 [stat], Nov. 2022. DOI: 10.48550/arXiv.2105.07536. [Online]. Available: http://arxiv.org/abs/2105.07536 (visited on 09/01/2025).

[31] Lightly AI, *Self-Supervised Learning — Lightly AI Documentation*, 2024. [Online]. Available: https://docs.lightly.ai/self-supervised-learning/index.html.

[32] O. Orikpete, A. Fawole, and D. Ewim, "Impact of Data Centers on Climate Change: A Review of Energy Efficient Strategies," *The Journal of Engineering and Exact Sciences*, vol. 9, pp. 1–15, Aug. 2023. DOI: 10.18540/jcecvl9iss6pp16397-01e.

[33] E. Strubell, A. Ganesh, and A. McCallum, "Energy and Policy Considerations for Deep Learning in NLP," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds., Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 3645–3650. DOI: 10.18653/v1/P19-1355. [Online]. Available: https://aclanthology.org/P19-1355/ (visited on 09/09/2025).

[34] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, *Green AI*, arXiv:1907.10597 [cs], Aug. 2019. DOI: 10.48550/arXiv.1907.10597. [Online]. Available: http://arxiv.org/abs/1907.10597 (visited on 09/09/2025).

[35] U. Gupta, Y. G. Kim, S. Lee, *et al.*, *Chasing Carbon: The Elusive Environmental Footprint of Computing*, arXiv:2011.02839 [cs], Oct. 2020. DOI: 10.48550/arXiv.2011.02839. [Online]. Available: http://arxiv.org/abs/2011.02839 (visited on 09/09/2025).

[36] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, *Federated Learning: Strategies for Improving Communication Efficiency*, arXiv:1610.05492 [cs], Oct. 2017. DOI: 10.48550/arXiv.1610.05492. [Online]. Available: http://arxiv.org/abs/1610.05492 (visited on 09/09/2025).

[37] M. Feffer, H. Heidari, and Z. C. Lipton, *Moral Machine or Tyranny of the Majority?* arXiv:2305.17319 [cs], May 2023. DOI: 10.48550/arXiv.2305.17319. [Online]. Available: http://arxiv.org/abs/2305.17319 (visited on 09/09/2025).

[38] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating Noise to Sensitivity in Private Data Analysis," en,

[39] S. Li, T. Zhou, X. Tian, and D. Tao, "Learning to Collaborate in Decentralized Learning of Personalized Models," en, in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA: IEEE, Jun. 2022, pp. 9756–9765, ISBN: 978-1-6654-6946-3. DOI: 10.1109/CVPR52688.2022.00954. [Online]. Available: https://ieeexplore.ieee.org/document/9880456/ (visited on 09/10/2025).

[40] Z. Li, T. Lin, X. Shang, and C. Wu, *Revisiting Weighted Aggregation in Federated Learning with Neural Networks*, arXiv:2302.10911 [cs], Jun. 2023. DOI: 10.48550/arXiv.2302.10911. [Online]. Available: http://arxiv.org/abs/2302.10911 (visited on 09/10/2025).

[41] L. Chen, W. Liu, Y. Chen, and W. Wang, *Communication-Efficient Design for Quantized Decentralized Federated Learning*, arXiv:2303.08423 [cs], Oct. 2023. DOI: 10.48550/arXiv.2303.08423. [Online]. Available: http://arxiv.org/abs/2303.08423 (visited on 09/10/2025).