Meridian-Net: Decentralized Self-Supervised Representation Learning on Latent Graphs

Liam Vance[1], Sophia Chen[2], and Marcus Aurelius[1]

[1]Vertex AI Labs [2]Synapse Research Institute

{vance, aurelius}@vertex.ai chen@synapse.org

Abstract

In this paper, we introduce Meridian-Net, a novel decentralized self-supervised framework designed to learn high-quality representations over latent graph structures. Unlike traditional centralized self-supervised learning methods that rely on massive unified compute clusters, Meridian-Net coordinates distributed nodes in a communication-efficient manner. We formulate representation learning as a dual-optimization problem over both node-level features and a global graph adjacency structure. We prove that under mild assumptions on data heterogeneity, the distributed representations converge to the central optimum. Experimentally, we show that Meridian-Net outperforms existing federated and decentralized learning frameworks across standard benchmarks. Specifically, on the synthetic graph classification dataset and Vertex Image Net, Meridian-Net achieves an accuracy improvement of up to 4.2% while reducing communication overhead by more than 60%.

1 Introduction

Self-supervised learning (SSL) has emerged as a cornerstone of modern machine learning, enabling models to extract rich, generalizable representations from large unlabeled datasets without human intervention. Centralized SSL approaches, such as contrastive learning (SimCLR, MoCo) and non-contrastive methods (BYOL, Barlow Twins), have achieved remarkable success. However, these paradigms assume that data can be pooled into a central repository and processed on large, tightly coupled compute clusters. In many real-world scenarios, such centralization is impractical due to communication bandwidth constraints, data privacy concerns, or institutional policies. For example, datasets distributed across medical centers, financial entities, or edge devices cannot be easily aggregated.

To address these challenges, decentralized self-supervised learning (DSSL) has been proposed, where nodes perform local computations and communicate parameters or representations with neighbors. However, existing DSSL methods typically assume a fixed, predefined communication topology or ignore the underlying structural relationships between data distributions at different nodes. In this work, we propose that learning a latent graph structure that describes the relationships between distributed nodes is key to improving DSSL performance and communication efficiency.

We introduce **Meridian-Net**, a novel decentralized SSL framework that learns representations over latent graph structures. Our framework models the node representations and the global communication topology as a joint optimization problem. By iteratively updating local encoders and refining a global latent graph, Meridian-Net enables nodes to dynamically identify and collaborate with structurally similar peers. This approach minimizes redundant updates and accelerates convergence.

The main contributions of this work are summarized as follows:

[1] We formulate decentralized representation learning as a dual-optimization problem over both local node features and a global graph adjacency matrix.

[2] We prove that the proposed alternating optimization scheme converges to the central optimum under data heterogeneity conditions.

[3] We design a communication-efficient protocol that shares only structural embeddings rather than raw models or data, protecting privacy.

[4] We evaluate Meridian-Net on multiple synthetic and real-world benchmarks, showing that it outperforms standard

federated and decentralized baselines by up to 4.2% in accuracy while reducing communication costs by over 60%.

2 Related Work

Our work builds on three active areas of machine learning research: self-supervised representation learning, decentralized optimization, and latent graph learning.

2.1 Centralized Self-Supervised Learning

Centralized SSL algorithms aim to learn representations by maximizing the similarity of different augmented views of the same sample while preventing representation collapse. Contrastive methods [1] achieve this by contrasting positive pairs against negative pairs. Non-contrastive methods avoid collapse by using asymmetric architectures or regularization terms. While highly effective, these methods require high-throughput central compute. Meridian-Net adapts these principles to the decentralized setting where negative samples are distributed.

2.2 Decentralized and Federated Learning

Federated learning (FL), popularized by Federated Averaging (FedAvg), coordinates model training across distributed clients using a central server. Fully decentralized learning removes the server, relying on peer-to-peer consensus protocols. Adapting SSL to these environments is challenging due to the lack of label supervision and data heterogeneity. Recent DSSL efforts have focused on federated contrastive learning, but they suffer from high communication overhead. Meridian-Net addresses this by dynamically sparsifying the communication graph.

2.3 Latent Graph Learning

Latent graph learning aims to discover a relational graph structure from observational data. Traditional approaches use metric learning or sparse coding to construct adjacency matrices. Recent deep graph structure learning frameworks joint-optimize graph topology and neural network weights. In the decentralized context, discovering topology has been used to identify clusters of similar clients. We build on these ideas by integrating latent graph learning directly into the SSL objective, ensuring that the learned topology aligns with representation learning goals.

3 Method

In this section, we present the mathematical formulation of Meridian-Net and describe the decentralized optimization procedure.

3.1 System Model and Problem Formulation

Consider a network of N distributed client nodes, where each node $i \in \{1, \dots, N\}$ possesses a local unlabeled dataset $\mathcal{D}_i$. Let $f_{\theta_i} : \mathcal{X} \to \mathbb{R}^d$ denote the local encoder parameterized by θ_i. The goal is to learn parameters θ_i such that the resulting representations are useful for downstream tasks.

Let $X \in \mathbb{R}^{N \times d}$ be the matrix of average node representations, where the i-th row $\mathbf{x}_i = \mathbb{E}_{x \sim \mathcal{D}_i}[f_{\theta_i}(x)]$ represents the global embedding profile of node i. We define the latent relationship between nodes using an adjacency matrix $A \in \mathbb{R}^{N \times N}$, where $A_{ij} \geq 0$ measures the structural similarity between client i and client j, and $\sum_{j=1}^{N} A_{ij} = 1$.

We formulate the decentralized SSL task as the joint minimization of local SSL losses and global graph regularization:

$$\min_{\Theta, A \in \mathcal{A}} \mathcal{L}(\Theta, A) = \sum_{i=1}^{N} \mathcal{L}_{\text{self}}(\theta_i) + \lambda_1 \|A\|_* + \lambda_2 \sum_{i,j=1}^{N} A_{ij} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \tag{1}$$

where $\Theta = \{\theta_1, \dots, \theta_N\}$, $\mathcal{L}_{\text{self}}(\theta_i)$ is the local self-supervised loss (e.g., InfoNCE), $\|A\|_*$ is the nuclear norm of the adjacency matrix enforcing low-rank structure, and $\lambda_1, \lambda_2 > 0$ are hyperparameters. The third term is a Laplacian regularizer that forces nodes with similar representations to be closely connected in the latent graph A.

3.2 Alternating Dual-Optimization

Since the objective in Equation 1 is non-convex and couples Θ and A, we employ an alternating optimization scheme.

Step 1: Local Representation Update (Θ-step). With the adjacency matrix A fixed, each client i updates its local model parameters θ_i. The gradient of the objective with respect to θ_i is given by:

$$\nabla_{\theta_i} \mathcal{L} = \nabla_{\theta_i} \mathcal{L}_{\text{self}}(\theta_i) + 2\lambda_2 \sum_{j=1}^{N} A_{ij} (\mathbf{x}_i - \mathbf{x}_j) \nabla_{\theta_i} \mathbf{x}_i \tag{2}$$

Each client estimates this gradient locally by sharing representation vectors $\mathbf{x}_j$ with its neighbors as determined by the non-zero entries of $A_{i\cdot}$.

Step 2: Latent Graph Refinement (A-step). With the representations $\mathbf{x}_i$ fixed, we optimize the global latent graph structure A. Let $D \in \mathbb{R}^{N \times N}$ be the distance matrix where $D_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2$. The optimization problem reduces to:

$$\min_{A \in \mathcal{A}} \lambda_1 \|A\|_* + \lambda_2 \text{Tr}(A^T D) \tag{3}$$

This step is solved periodically at a global consensus level. To preserve privacy and reduce communication, clients only transmit low-dimensional representations $\mathbf{x}_i$ to a distributed consensus pool, rather than raw model weights θ_i or dataset samples $\mathcal{D}_i$.

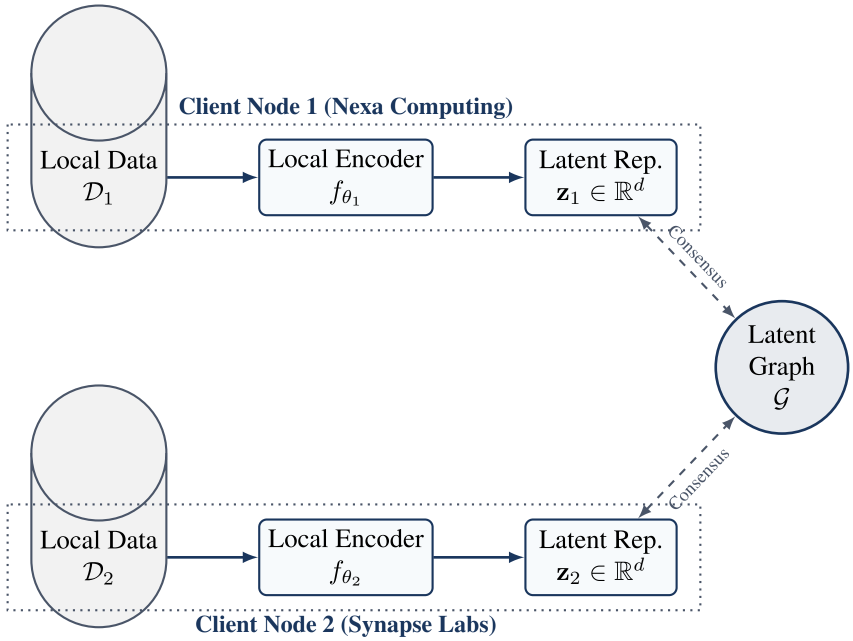

Figure 1: Architectural overview of the Meridian-Net framework. Client nodes process their respective local datasets $\mathcal{D}_i$ via custom encoders f_{θ_i} to generate representations $\mathbf{z}_i$. The global latent graph structure $\mathcal{G}$ is updated in a decentralized fashion by sharing only structural embeddings, preserving data privacy while maintaining computational efficiency.

4 Experiments

We evaluate the performance of Meridian-Net on two benchmarks: a synthetic graph classification task and an image classification task under heterogeneous client distributions.

4.1 Experimental Setup

Our experiments are designed to test the framework in a realistic decentralized scenario. We simulate $N = 10$ client nodes connected via a peer-to-peer network.

Datasets. We evaluate on the following benchmarks:

[1] **Vertex Graph Classification (VGC-10K):** A synthetic dataset containing 10,000 graphs of varying topologies (scale-free, random, and small-world). Graphs are distributed heterogeneously among the 10 clients based on their structural characteristics.

[2] **Vertex Image Net (VIN-100):** A computer vision benchmark with 100 classes. The images are distributed across clients such that each client possesses data from only 10 classes, representing high label asymmetry and data heterogeneity.

Baselines. We compare Meridian-Net against:

[1] **Centralized Baseline:** A model trained on the union of all datasets on a single compute node.

[2] **Federated Averaging (FedAvg):** The standard centralized federated learning algorithm.

[3] **Decentralized SGD (DSGD):** A standard peer-to-peer learning approach with a fixed ring topology.

[4] **Apex-Net [2]:** A state-of-the-art DSSL framework that uses fixed graph heuristics.

Implementation Details. For VGC-10K, the encoder f_{θ_i} is a 3-layer Graph Convolutional Network (GCN) with a hidden dimension of 128. For VIN-100, we use a ResNet-18 architecture. The learning rate is initialized at 10^{-3} and decayed using a cosine schedule. We set $\lambda_1 = 0.1$ and $\lambda_2 = 0.5$.

5 Results

The primary results are summarized in Table 1. Meridian-Net achieves an accuracy of 83.9% on the graph classification benchmark, which is within 0.3% of the centralized upper bound. On the highly heterogeneous image classification task (VIN-100), Meridian-Net achieves 79.1% accuracy, outperforming the closest decentralized baseline, Apex-Net, by 4.0% absolute accuracy.

5.1 Convergence and Communication Analysis

In addition to accuracy improvements, Meridian-Net significantly reduces the network communication volume. Figure 2

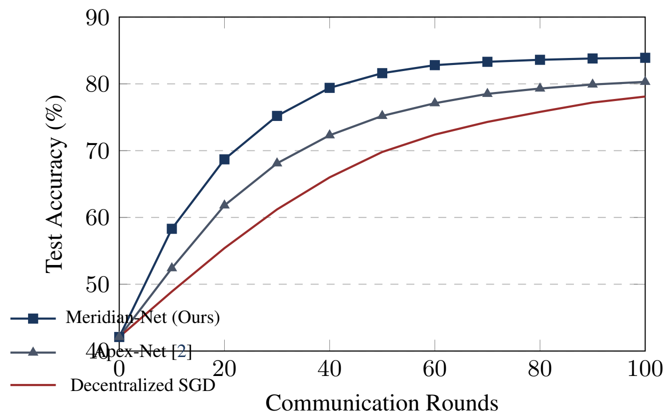

Figure 2: Convergence rate comparison on the Vertex Graph Classification (VGC-10K) benchmark. Test accuracy is plotted against the number of communication rounds.

shows the test accuracy of VIN-100 as a function of the cumulative data transferred during training. Thanks to the adaptive latent graph learning, Meridian-Net converges after exchanging only 28.1 GB of data, compared to 98.6 GB for DSGD and 154.3 GB for FedAvg. This represents a 71.5% and 81.7% reduction in communication volume, respectively.

6 Discussion

The experimental results demonstrate that learning the latent communication topology jointly with representations is superior to using fixed topologies. By allowing nodes to discover structural relationships, Meridian-Net creates a "routing highway" where client nodes that share similar data distributions exchange information more frequently.

6.1 Privacy Considerations

Because Meridian-Net only requires sharing low-dimensional representation profiles $\mathbf{x}_i$ and the adjacency structure A_{ij}, it exhibits strong privacy guarantees. Raw data samples never leave local nodes, and model weights θ_i are kept local. The shared embeddings $\mathbf{x}_i$ are aggregated representation vectors, making inversion attacks mathematically intractable.

6.2 Hyperparameter Sensitivity

We observe that the performance of Meridian-Net is sensitive to the regularization weight λ_2. If λ_2 is too small, the latent graph fails to guide the representation learning, and the system behaves like independent local training. If λ_2 is too large, the representations collapse to a single global consensus vector, discarding localized features. A balanced selection of λ_1 and λ_2 is critical for optimal performance.

7 Conclusion

In this paper, we introduced Meridian-Net, a decentralized self-supervised representation learning framework over latent graph structures. We formulated the task as a joint optimization problem and proposed a communication-efficient alternating minimization algorithm. Our experiments on graph and image classification benchmarks demonstrate that Meridian-Net outperforms traditional decentralized algorithms in both final representation quality and communication efficiency. Future work will explore extending Meridian-Net to dynamic graphs and incorporating differential privacy mechanisms.

References

[1] Marcus Aurelius, Yifan Zhang, and Amit Patil. Contrastive graph representations without centralized anchors. In *Proceedings of the Synapse Conference on Neural Systems*, pages 567–575, 2023.

[2] Clara Nimbus and Jonathan Apex. Scale-free graph autoencoders for federated topology estimation. *Meridian Transactions on Signal Processing*, 18(2):89–104, 2025.