1st International Workshop on Distributed and Parallel Programming for Extreme-scale AI
June 16-17th, 2025 | Paris, France

Spectral Embedding to Compress Neural Architectures Without Performance Loss

Quentin Petit
Mines Paris - PSL University France
Paris, France
quentin.petit@minesparis.psl.eu

Chong Li
Huawei Paris Research Center
Boulogne-Billancourt, France
ch.l@huawei.com

Nahid Emad
MDLS, Li-PARaD & UVSQ
Saclay, France
nahid.emad@uvsq.fr

Introduction

Challenge: High-dimensional inputs → large models, slow training, overfitting

Problem: How can we compress input dimensionality while preserving accuracy?

Goal: Construct meaningful low-dimensional representations that preserve variance and accelerate training.

We propose a spectral embedding approach that builds a compact input representation using the dominant eigenvectors of a matrix capturing data structure (e.g., co-occurrence, covariance). These embeddings retain the most significant variance in the dataset.

Our method uses MIRAMnsa scalable, parallel eigensolverto extract these dominant components efficiently, enabling direct projection of high-dimensional input data onto a compact subspace defined by the dominant eigenvectors.

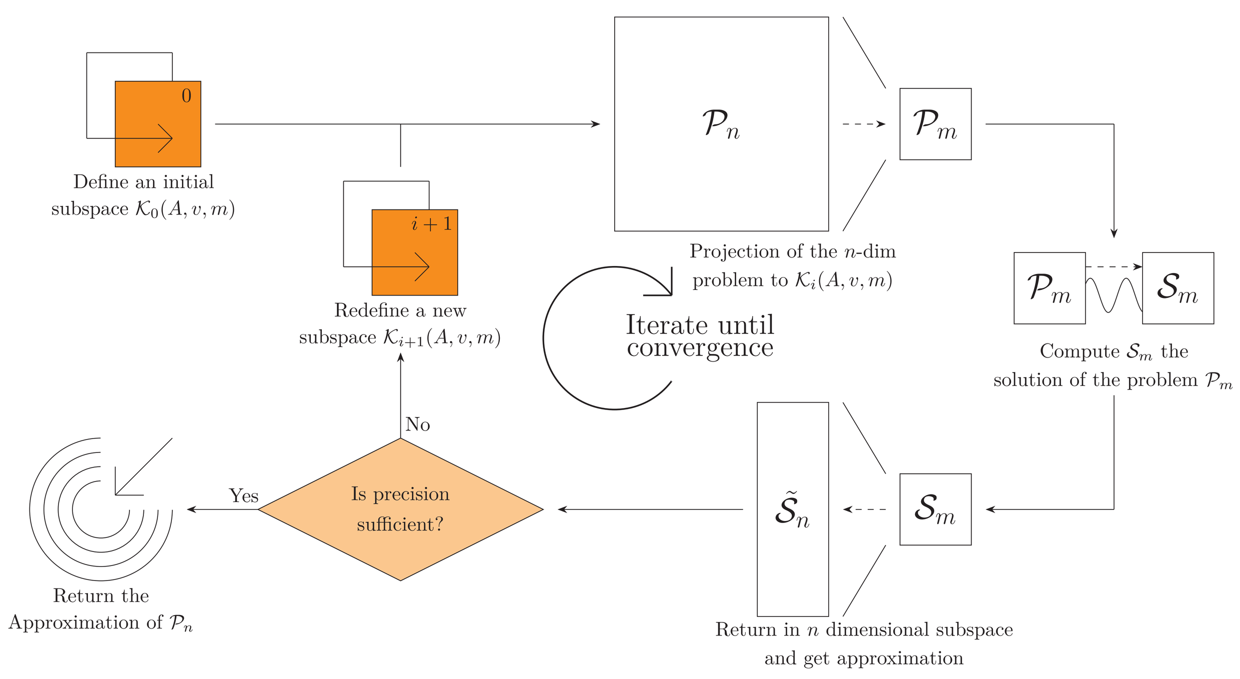
Why Use Restarted Projection Methods?

Motivation:

- For very large matrices, standard eigen-decomposition is too costly.
- We only need the k dominant eigenvectors, not the full spectrum.
- Restarted projection methods focus computation to detect the most relevant directions.

Why it fits here:

- Efficient for large, sparse or structured matrices.
- Well-suited for high parallel and distributed computing.
- Basis for Multiple IRAM with nested subspaces (MIRAMns), our spectral embedding engine.



How it works:

- Project the problem onto a set of smaller Krylov subspaces.
- Select the best of them.
- Solve reduced problem approximate Ritz eigenvectors.
- Evaluate residuals to assess approximation quality.
- Refine via restarts until convergence is reached.

MIRAMns as Eigensolver

Why use MIRAMns?

- Extracts k dominant eigenvectors from large matrices.

$$A \cdot u_i = \lambda_i \cdot u_i$$

- Handles *clustered eigenvalues* via nested Krylov subspaces:

$$\mathcal{K}_{m_i}(A, v) = \text{span}\{v, Av, A^2v, \ldots, A^{m_i-1}v\}$$
$$\text{with } \mathcal{K}_{m_1} \subset \mathcal{K}_{m_2} \subset \cdots \subset \mathcal{K}_{m_l}$$

- Selects the best subspace for accurate approximation.
- Implicitly restarted avoids full recomputation.
- Parallel-friendly: uses matrix-vector multiplications.

More stable than IRAM with faster convergence

Methodology Overview

Spectral Embedding Procedure:

1. **Compute** a square matrix (e.g., co-occurrence matrix) to represent dataset.
2. **Extract** k dominant eigenvectors via MIRAMns.
3. **Embed** original data into the k-dimensional spectral space.
4. **Train** neural networks using the compressed input space.

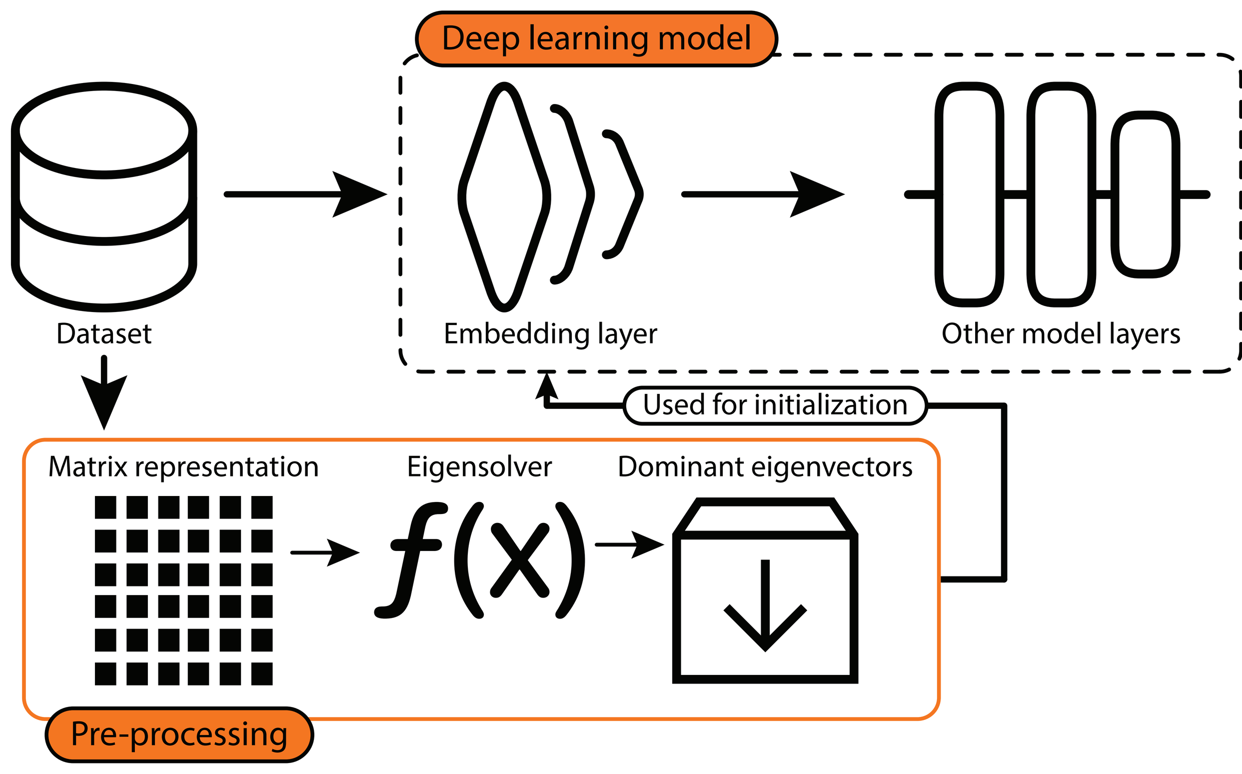
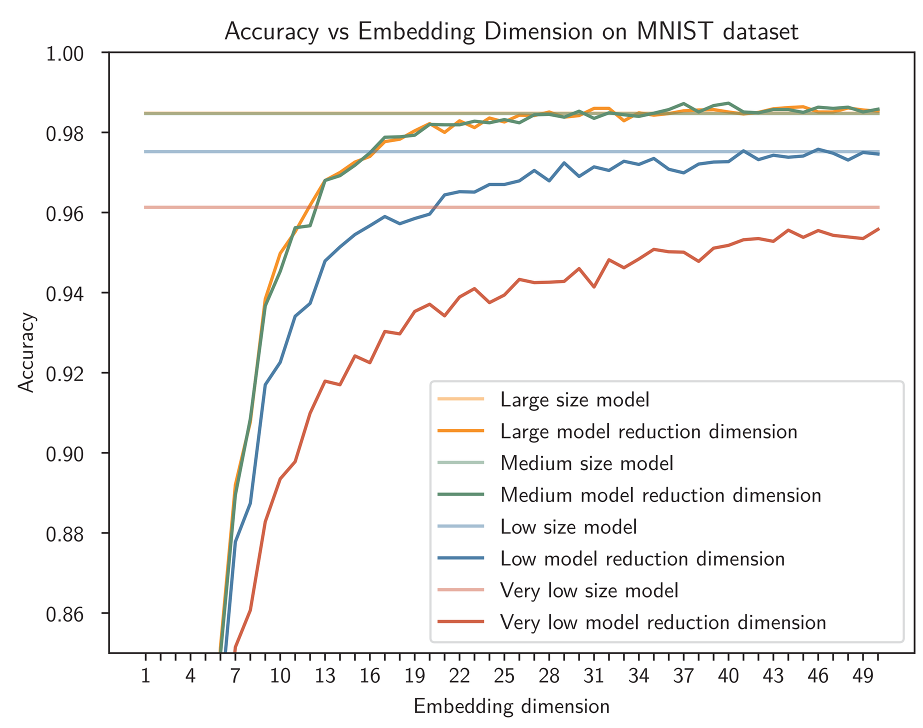


Figure: Spectral embedding pipeline.

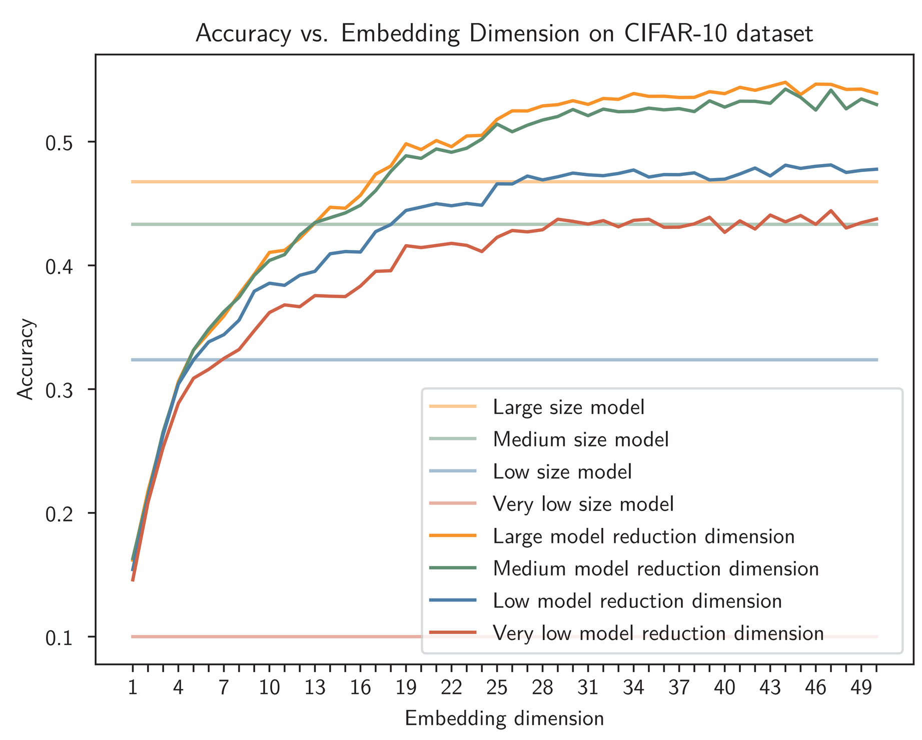
Key Results



Performance Highlights:

- Low-dimensional embeddings retain 95-98% accuracy of full-input performance.
- Some datasets show improved performance
  → suggesting reduced overfitting
- Significant model compression

Smaller Models, Same Accuracy



Radar Dataset Example:

	Baseline	Embedding
Input dim	175	10
# params	1,143,815	233,991
Train. Time (s)	1180	727
Test Acc. (%)	98.772	98.390

- Input dimension: 17× reduction
- Parameters: 4.8× smaller
- Training time: -35% with <0.5% accuracy loss

Conclusion

- **Effective dimensionality reduction**: Spectral embeddings based on dominant eigenvectors successfully reduce input dimension while maintaining model accuracy.
- **Performance preservation**: Data-specific embeddings enable the development of smaller and faster neural networks without compromising computational performance or predictive capabilities.
- **Robust algorithmic framework**: MIRAMns ensures reliable convergence, effectively handles clustered eigenvalues, and scales efficiently in distributed environments.

Future Work

- **Extend to complex data types**: Apply the approach to graphs and text data, where structure-aware embeddings can bring additional benefits.
- **Integrate with modern NNs**: Explore compatibility with convolutional NNs and transformer-based models to broaden the method's applicability.
- **Scale to large language models**: Extend the methodology to more complex deep learning models such as LLMs.

Explore our Research