Dimensionality Reduction

Seetha Abhinav Aarav Desai Piyush Kumar Abhinav Goyal

Mathematics and Computing
Indian Institute of Science

Abstract

Dimensional reduction is an essential technique in modern data analysis and machine learning, mitigating the challenges of high-dimensional data while preserving inherent structural characteristics. In this paper, we present a comprehensive study of dimensional reduction methods—ranging from linear random projections to nonlinear techniques such as t-SNE for visualization, and extending to deep, stochastic approaches exemplified by the contrastive Gaussian mixture variational autoencoder (C-GMVAE). These methods span the spectrum from deterministic, shallow algorithms to sophisticated deep learning-based architectures.

1 Introduction

High-dimensional data presents significant challenges in data analysis and machine learning, including increased computational complexity, risk of overfitting, and difficulties in visualization and interpretation—commonly referred to as the **"curse of dimensionality."** As the number of features grows, data points become sparse, patterns harder to discern, and models more prone to poor generalization. Dimensional reduction addresses these issues by transforming high-dimensional data into a lower-dimensional space while preserving its essential structure. This not only improves computational efficiency and model performance but also enables meaningful visualization and better insights into the underlying data distributions.
This paper first discusses random projections as a linear dimensional reduction method, comparing it with other methods. Next, it examines t-SNE for visualization, highlighting its advantages over other nonlinear methods. Finally, it explores C-GMVAE, a deep generative approach for handling complex, high-dimensional, multi-label data. Together, these methods illustrate the evolution from shallow to deep dimensional reduction techniques. Here is the repo file containing all the experiments : GitHub Repository: Dimensionality Reduction. Link for video demo: Video

2 Methodology

First we start with the idea of random projection. Given a dataset $X \in \mathbb{R}^{d \times N}$ with N observations in d-dimensional space, random projection maps it to a k-dimensional subspace ($k \ll d$) using a random matrix $R \in \mathbb{R}^{k \times d}$:

$$X^{RP}_{k \times N} = R_{k \times d} \cdot X_{d \times N}$$

The matrix R is designed to approximately preserve pairwise distances between data points, as guaranteed by the **Johnson-Lindenstrauss (JL) lemma**, if points in a vector space are projected onto a randomly selected subspace of suitably high dimension, then the distances between the points are approximately preserved. Random projection is computationally very simple: forming the random matrix R and projecting the $d \times N$ data matrix X into k dimensions is of order $O(dkN)$, and if the data matrix X is sparse with about c nonzero entries per column, the time complexity is of order

NeurIPS 2023 Generative AI and Biology Workshop.

$O(ckN)$. The results of random projections with other methods are compared in the Experiment section.

Next we describe how dimensional reduction plays a key role in visualization of the data. **Stochastic Neighbor Embedding (SNE)** is a nonlinear dimensionality reduction technique designed to preserve the local structure of high-dimensional data. It works by converting pairwise similarities between data points into probabilities that represent the likelihood of one point being a neighbor of another. But SNE suffers from what's called the **crowding problem**: in high-dimensional space, moderate distances can represent significant dissimilarities, but when reducing to a low-dimensional space, there simply isn't enough "room" to accurately preserve all those moderate pairwise distances. As a result, points that are not truly neighbors can get squished together, distorting the structure.
t-Distributed Stochastic Neighbor Embedding (t-SNE) improves upon SNE by modifying the similarity measure in the low-dimensional space. Specifically, it replaces the Gaussian distribution with a **Student t-distribution** (with one degree of freedom), which has heavier tails. This adjustment helps spread out the points and alleviates the crowding issue by allowing dissimilar points to be modeled as farther apart. One of the key parameters in t-SNE is **perplexity**, which can be thought as a smooth measure of the number of effective neighbors each point considers. The overall complexity of t-SNE is $O(N^2)$ as it requires computing pairwise distances between each points making it less feasible for large datasets. The **Barnes-Hut approximation** is an optimization technique used in t-SNE to reduce its computational complexity from $O(N^2)$ to $O(NlogN)$, making it feasible for large datasets.

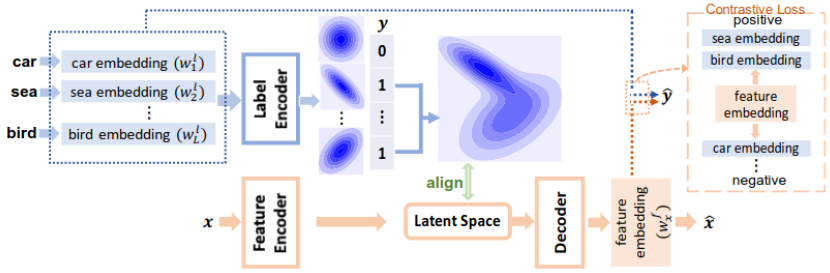


Figure 1: Pipeline of C-GMVAE

To further enhance our understanding and modeling of complex data, generative models like **Variational Autoencoders (VAEs)** have been introduced. VAEs not only reduce dimensionality but also learn probabilistic representations, providing insights into the data's latent structure. It uses a stochastic sampling process to generate new data.
A **Gaussian Mixture Model (GMM)** is a probabilistic model that assumes data points are generated from a mixture of several Gaussian distributions, each representing a different cluster within the data. GMMs are useful because they can model complex, multi-modal data distributions and provide a natural way to perform soft clustering, where each data point is associated with a probability of belonging to each cluster. **Gaussian Mixture Variational Autoencoders (GMVAEs)** build on the strengths of both VAEs and GMMs by replacing the standard unimodal Gaussian prior of a VAE with a mixture of Gaussians. This enhancement allows the latent space to naturally accommodate multiple clusters or subpopulations, making GMVAEs particularly effective for tasks where the data inherently exhibits multi-modality.
Contrastive learning can be incorporated in GMVAE to enhance the discriminative power of the latent space. It helps fine-tune the clustering by promoting consistency in how similar instances are embedded, potentially leading to more distinct and well-separated clusters within the Gaussian mixture model.
As shown in Figure.1, each label category is mapped to a learnable embedding vector w_i^l. The label encoder transforms each label embedding into a multivariate Gaussian distribution in the latent space. Simultaneously, the input feature is mapped into the latent space using a feature encoder, producing a posterior distribution. This posterior is aligned with the label-based prior using a **KL-divergence loss**. A decoder then samples from the latent distribution to produce a feature embedding w_x^f. A contrastive loss is applied to pull the feature embedding w_x^f closer to the positive label embeddings while pushing it away from the embeddings of negative (non-associated) labels. The final prediction $\hat{y}$ is obtained

by computing the inner product between the feature embedding w_x^f and all label embeddings w_i^l, followed by a sigmoid activation for each label to yield multi-label predictions.

3 Experiments

3.1 Dimensional Reduction using Gaussian and Achiloptas Random Projections vs. Other classical methods

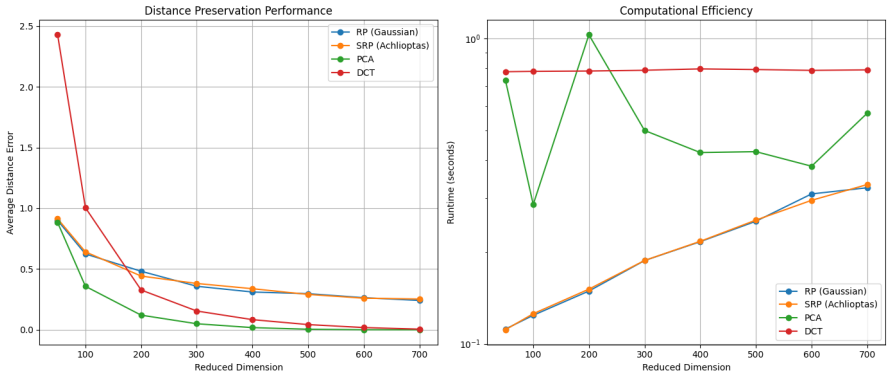


Figure 2: Dimension Reduction on **Extended MNIST dataset** using different methods, i) Error in distance Preserving, ii) Computational cost of each method

As seen in the graph, as the number of dimensions increases the error in distance preservation becomes less. Moreover, the dimensional reductions using random projections have similar accuracy as compared to PCA and DCT, while the computational cost using random methods is very less.

3.2 Visualizations using t-SNE and other methods

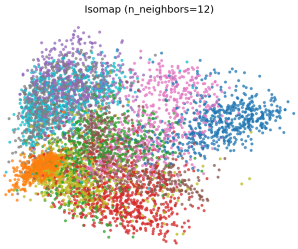


Figure 3: *
(a) Isomap with 12 neighbors

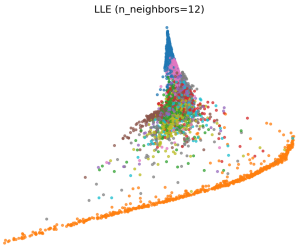


Figure 4: *
(b) LLE with 12 neighbors

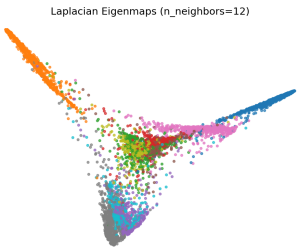


Figure 5: *
(c) Laplacian Eigenmap with 12 neighbors

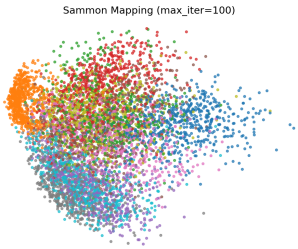


Figure 6: *
(d) Sammon mapping

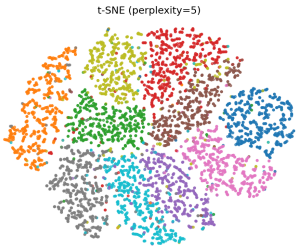


Figure 7: *
(e) tSNE with perplexity= 5

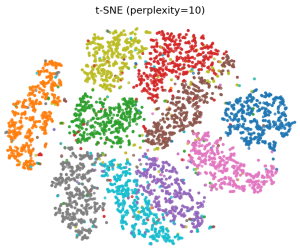


Figure 8: *
(f) tSNE with perplexity= 10

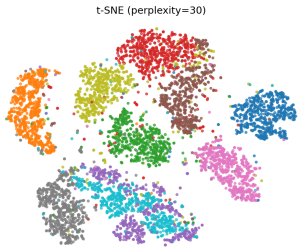


Figure 9: *
(g) tSNE with perplexity= 30

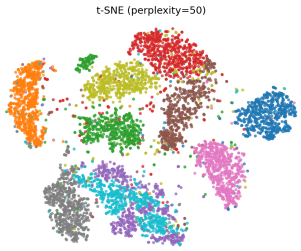


Figure 10: *
(h) tSNE with perplexity= 50

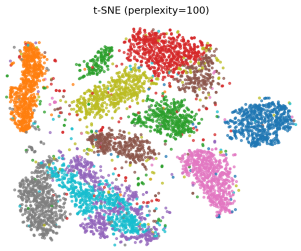


Figure 11: *
(i) tSNE with perplexity= 100

The above plots are constructed by randomly sampling 6000 datapoints from MNIST dataset, which has 60,000 images. As seen from the plots, tSNE is much superior as compared to other methods for visualization of the data. When perplexity is high, (in this case 50, 100) the algorithm tries to preserve larger-scale structure — considering more distant points when computing similarities. When the perplexity is low (in this case 5, 10), we get tight, well-separated clusters, but it may miss global structure as related clusters may be placed far apart.

3.3 C-GMVAE for Multi-label classification

Method	example-F1	micro-F1	macro-F1	Hamming Accuracy
BR	0.325	0.371	0.182	0.866
MLKNN	0.383	0.415	0.266	0.877
HARAM	0.432	0.447	0.284	0.684
SLEEC	0.416	0.413	0.364	0.870
C2AE	0.501	0.545	0.393	0.897
LaMP	0.492	0.535	0.387	0.897
MPVAE	0.514	0.552	0.422	0.898
ASL	0.477	0.525	0.410	0.893
RBCC	0.468	0.513	0.409	0.888
C-GMVAE	0.534	0.575	0.440	0.903

Table 1: Performance comparison on the mirflickr dataset across different metrics and methods

For the above table, note that the values for the methods other than C-GMVAE are taken from relevant research papers, as available in the reference, for the mirFlickr dataset, while the calculation for C-GMVAE was implemented.
It should be noted that C-GMVAE was able to achieve **state-of-art** performance across the datasets for multilabel classification.

4 Conclusion

Our comprehensive study of dimensional reduction techniques reveals important insights across the spectrum from linear to deep learning approaches. Random projections demonstrate competitive performance with remarkable computational efficiency, confirming their value in large-scale applications. Visualization experiments with t-SNE highlight its strength in preserving local structures while cautioning against over-interpretation of global patterns. Most significantly, C-GMVAE shows impressive performance in multi-label classification tasks, achieving comparable or superior results to traditional methods even with reduced training data.
These findings illustrate the evolution of dimensional reduction techniques and their inherent trade-offs. While simpler methods offer computational advantages, deep approaches like C-GMVAE capture more complex data relationships at higher computational cost. Future work should explore hybrid approaches balancing efficiency with representational power. Ultimately, our research emphasizes that dimensional reduction method selection should be guided by specific application requirements, computational constraints, and data characteristics rather than a universal approach.

References

[1] https://www.stat.cmu.edu/~larry/=sml/DimRed.pdf
This document from Carnegie Mellon University provides an overview of various dimensionality reduction techniques along with theoretical insights and practical examples.

[2]https://users.ics.aalto.fi/ella/publications/randproj_kdd.pdf
A paper by Bingham and Mannila that explores random projection techniques for dimensionality reduction, detailing both theoretical foundations and experimental results.

[3] https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf
The influential paper by van der Maaten and Hinton on t-SNE, presenting a novel approach for visualizing high-dimensional data through nonlinear dimensionality reduction.

[4] https://arxiv.org/abs/2112.00976
A recent arXiv preprint that introduces innovative advancements in dimensionality reduction, incorporating modern machine learning techniques.

[5] https://cs.gmu.edu/~jessica/publications/lsi_sdm_workshop03.pdf
A study on Latent Semantic Indexing (LSI) presented at an SDM workshop, focusing on dimensionality reduction methods applied to text and document analysis.

[6] https://lvdmaaten.github.io/publications/misc/Supplement_JMLR_2008.pdf
Supplementary material for the t-SNE paper by van der Maaten, which offers additional experimental details and technical insights.

[7] https://cseweb.ucsd.edu/~dasgupta/papers/jl.pdf
A paper discussing the Johnson-Lindenstrauss lemma, a fundamental result underlying random projection methods used in dimensionality reduction.

[8] https://github.com/JunwenBai/C-GMVAE
The GitHub repository hosting the implementation of the Contrastive Gaussian Mixture Variational Autoencoder (C-GMVAE) for multi-label classification.

[9] https://lvdmaaten.github.io/tsne/
The official t-SNE webpage by Laurens van der Maaten, featuring details, examples, and software related to the t-SNE dimensionality reduction technique.

Appendix

Theoretical Framework of Random Projection

1. **Dimensionality Reduction via Random Projection** In random projection, the original d-dimensional data is projected to a k-dimensional ($k \ll d$) subspace through the origin, using a random $k \times d$ matrix R whose columns have unit lengths. Using matrix notation where $X_{d\times N}$ is the original set of N d-dimensional observations,

$$X_{k\times N}^{RP} = R_{k\times d} X_{d\times N}$$

is the projection of the data onto a lower k-dimensional subspace. The key idea of random mapping arises from the Johnson-Lindenstrauss lemma

2. **Key Theoretical Components**

2.1 **Distance Preservation** For any two data vectors $x_1, x_2 \in \mathbb{R}^d$, the Euclidean distance in the projected space is scaled to approximate the original distance:

$$\|x_1 - x_2\| \approx \frac{d}{k} \cdot \|Rx_1 - Rx_2\|$$

The scaling factor $\frac{d}{k}$ compensates for the reduced dimensionality, ensuring:

$$E[\|Rx\|^2] = \frac{k}{d}\|x\|^2$$

for a unit vector x.

2.2 **Random Matrix Construction**

Gaussian Random Projection (RP): Elements of R are sampled as:

$$R_{ij} \sim N\Big(0, \frac{1}{d}\Big)$$

Sparse Random Projection (SRP): Proposed by Achlioptas, elements follow:

$$R_{ij} = \begin{cases} +\sqrt{3} & \text{with probability } \frac{1}{6}, \\ 0 & \text{with probability } \frac{2}{3}, \\ -\sqrt{3} & \text{with probability } \frac{1}{6}. \end{cases}$$

This sparse construction reduces computational complexity to $O(ckN)$ for c-sparse data.

2.3 **Orthogonality Approximation** In high-dimensional spaces, random vectors are almost orthogonal:

$$E[RR^T] = I_k \quad \text{and} \quad E\Big[\frac{1}{d}R^T R\Big] \approx I_d.$$

The mean squared error (MSE) per element between $R^T R$ and I_d is:

$$\text{MSE} = E\Big[\Big((R^T R - I_d)_{ij}\Big)^2\Big] = \frac{1}{k}.$$

2.4 **Similarity Measures**

Euclidean Distance: Preserved under the JL lemma.

Inner Product/Cosine Similarity: For unit vectors x_1, x_2:

$$E[\langle Rx_1, Rx_2\rangle] = \langle x_1, x_2\rangle.$$

Variance is bounded by $\frac{1+2\langle x_1,x_2\rangle^2}{k}$.

3. **Computational Considerations**

Complexity:

Dense R: $O(dkN)$.

Sparse R: $O(ckN)$.

Trade-offs:

Gaussian R: Optimal distance preservation.

Sparse R: Integer arithmetic efficiency with minimal accuracy loss.

The Johnson–Lindenstrauss Theorem

Let $V \subset \mathbb{R}^d$ be any set of n points, and let $\varepsilon \in (0,1)$. Define the target dimension by

$$k \geq \frac{4\ln n}{\varepsilon^2/2 - \varepsilon^3/3}.$$

Then the following holds.

Theorem .1 *For any $0 < \varepsilon < 1$ and any integer n, if*

$$k \geq \frac{4\ln n}{\varepsilon^2/2 - \varepsilon^3/3},$$

then for any set $V \subset \mathbb{R}^d$ of n points, there exists a linear mapping

$$f : \mathbb{R}^d \to \mathbb{R}^k$$

such that for all distinct $u, v \in V$,

$$(1-\varepsilon)\|u-v\|^2 \leq \|f(u)-f(v)\|^2 \leq (1+\varepsilon)\|u-v\|^2.$$

Moreover, such a map can be computed in randomized polynomial time.

Overview of the Proof

The idea is to use a random projection onto a k-dimensional subspace. For any unit vector x (which may represent the direction of $u-v$), its projection has squared norm concentrated around its expected value of $\frac{k}{d}$. The following lemma provides the necessary concentration bound.

Lemma .2 *Let $X_1, X_2, \ldots, X_d \sim N(0,1)$ be independent Gaussian random variables. Define*

$$L = \frac{X_1^2 + \cdots + X_k^2}{X_1^2 + \cdots + X_d^2}.$$

Then for $k \leq d$ the following holds:

(a) If $0 < \varepsilon < 1$, then

$$\Pr\left[\frac{d}{k}\,L \notin [1-\varepsilon,\, 1+\varepsilon]\right] \leq \exp\left(-\frac{k\varepsilon^2}{2}\left(1-\frac{\varepsilon}{3}\right)\right).$$

(b) If $\varepsilon \geq 1$, then

$$\Pr\left[\frac{d}{k}\,L \notin [1-\varepsilon,\, 1+\varepsilon]\right] \leq \exp\left(-\frac{k\varepsilon^2}{2}\right).$$

Proof of Theorem .1

If $d \leq k$, the embedding is trivial. Otherwise, suppose $d > k$. Choose a random k-dimensional subspace $S \subset \mathbb{R}^d$ and let P_S denote the orthogonal projection onto S. Define the mapping f by

$$f(v) = \sqrt{\frac{d}{k}}\, P_S(v), \quad \text{for } v \in V.$$

This scaling ensures that for any unit vector x,

$$\mathbb{E}\|P_S(x)\|^2 = \frac{k}{d} \quad \Longrightarrow \quad \mathbb{E}\Big\|\sqrt{\frac{d}{k}}\, P_S(x)\Big\|^2 = 1.$$

For any two points $u, v \in V$, let $x = \frac{u-v}{\|u-v\|}$ be the corresponding unit vector. Its projected squared norm is given by

$$L = \frac{\|P_S(u-v)\|^2}{\|u-v\|^2}.$$

By Lemma .2,

$$\Pr\left[\frac{d}{k}\, L \notin [1-\varepsilon,\, 1+\varepsilon]\right] \leq \frac{2}{n^2},$$

after invoking standard inequalities (e.g., using properties of the logarithm). A union bound over all $\binom{n}{2}$ pairs shows that the probability any pair's distance is distorted outside the desired $(1 \pm \varepsilon)$ factor is at most

$$\binom{n}{2} \cdot \frac{2}{n^2} \leq 1 - \frac{1}{n}.$$

Thus, with high probability, the map f satisfies

$$(1-\varepsilon)\|u-v\|^2 \leq \|f(u) - f(v)\|^2 \leq (1+\varepsilon)\|u-v\|^2$$

for all $u, v \in V$. By repeating the random projection $O(n)$ times and choosing an instance that works, one obtains the desired mapping in randomized polynomial time.

Proof of Lemma .2

We present a sketch for part (a); the proof for (b) is similar.

Let

$$S_k = X_1^2 + \cdots + X_k^2 \quad \text{and} \quad S_d = X_1^2 + \cdots + X_d^2.$$

Since S_k and S_d are chi-square distributed with k and d degrees of freedom, respectively, and because the moment generating function for X^2 (with $X \sim N(0,1)$) is

$$E\left[e^{sX^2}\right] = \frac{1}{\sqrt{1-2s}}, \quad s < \frac{1}{2},$$

one can apply Markov's inequality to the random variable

$$\exp\Big(t\Big(S_k - \frac{k}{d} S_d\Big)\Big)$$

for $t > 0$ (with t constrained by $t < \frac{1}{2k}$). By splitting the expectation into the product corresponding to S_k and $S_d - S_k$ (which are independent), and optimizing over the parameter t, one obtains

$$\Pr\left[\frac{d}{k}\, L \notin [1-\varepsilon,\, 1+\varepsilon]\right] \leq \exp\left(-\frac{k\varepsilon^2}{2}\left(1 - \frac{\varepsilon}{3}\right)\right).$$

The optimal choice turns out to be

$$t_0 = \frac{1}{2}\,\frac{d-k}{k\,d},$$

and substituting t_0 yields the desired bound.

The detailed algebra, which involves standard large deviation techniques for chi-square distributions, is omitted for brevity.

Conclusion

Combining Lemma .2 with a union bound over all point pairs, we deduce that a random projection (with appropriate scaling) preserves all pairwise distances within a multiplicative factor of $1 \pm \varepsilon$ with high probability. This completes the proof of Theorem .1.

Mathematical Description of Dimensionality Reduction Methods

Principal Component Analysis (PCA)

Objective: Project data onto a lower-dimensional subspace while minimizing mean-squared reconstruction error.

Mathematical Framework:

Let $X \in \mathbb{R}^{d \times N}$ represent a centered data matrix with d-dimensional observations and N samples.

The covariance matrix is computed as:

$$\mathbf{C} = \mathbb{E}[XX^T] = \frac{1}{N-1} XX^T$$

Eigenvalue decomposition of $\mathbf{C}$:

$$\mathbf{C} = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^T$$

where $\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_d]$ contains orthonormal eigenvectors, and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_d)$ holds eigenvalues (ordered as $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$).

Dimensionality Reduction:

Select the top k eigenvectors $\mathbf{E}_k = [\mathbf{e}_1, \ldots, \mathbf{e}_k]$.

Project data onto the subspace spanned by $\mathbf{E}_k$:

$$X_{\text{PCA}} = \mathbf{E}_k^T X \in \mathbb{R}^{k \times N}$$

Optimality: PCA minimizes the reconstruction error $\|X - \mathbf{E}_k \mathbf{E}_k^T X\|_F^2$, where $\|\cdot\|_F$ is the Frobenius norm.

Complexity: $\mathcal{O}(d^2 N + d^3)$.

Singular Value Decomposition (SVD)

Objective: Factorize the data matrix to extract latent structure.

Mathematical Framework:

For $X \in \mathbb{R}^{d \times N}$, the SVD is:

$$X = \mathbf{U}\mathbf{S}\mathbf{V}^T$$

where:

- $\mathbf{U} \in \mathbb{R}^{d \times d}$, $\mathbf{V} \in \mathbb{R}^{N \times N}$ are orthogonal matrices (left/right singular vectors)
- $\mathbf{S} \in \mathbb{R}^{d \times N}$ is diagonal with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$, $r = \text{rank}(X)$

Dimensionality Reduction:

Truncate to retain k largest singular values:

$$\mathbf{U}_k = [\mathbf{u}_1, \ldots, \mathbf{u}_k], \quad \mathbf{S}_k = \text{diag}(\sigma_1, \ldots, \sigma_k)$$

Project data as:

$$X_{\text{SVD}} = \mathbf{U}_k^T X \in \mathbb{R}^{k \times N}$$

Relation to PCA: For centered data, $\mathbf{U}$ corresponds to $\mathbf{E}$ in PCA, and $\sigma_i^2/(N-1) = \lambda_i$.

Complexity: $\mathcal{O}(dcN)$ for sparse X with c non-zeros per column.

Discrete Cosine Transform (DCT)

Objective: Compress data by discarding high-frequency components.

Mathematical Framework:

For a 1D signal $\mathbf{x} \in \mathbb{R}^d$, the DCT coefficients $\mathbf{y} \in \mathbb{R}^d$ are:

$$y_m = \alpha_m \sum_{n=0}^{d-1} x_n \cos\left(\frac{\pi(2n+1)m}{2d}\right), \quad \alpha_m = \begin{cases} \sqrt{\frac{1}{d}}, & m = 0 \\ \sqrt{\frac{2}{d}}, & m > 0 \end{cases}$$

For 2D data (e.g., images), apply DCT separably along rows and columns.

Dimensionality Reduction:

Retain the first k coefficients and discard the rest.

The inverse DCT (IDCT) reconstructs the approximate signal:

$$\hat{x}_n = \sum_{m=0}^{k-1} \alpha_m y_m \cos\left(\frac{\pi(2n+1)m}{2d}\right)$$

Advantages:

- Fixed basis (data-independent), unlike PCA/SVD
- Near-optimal energy compaction for natural images

Complexity: $\mathcal{O}(dN \log_2(dN))$ via fast Fourier transform (FFT)-based algorithms.

Supplementary Experiments 1

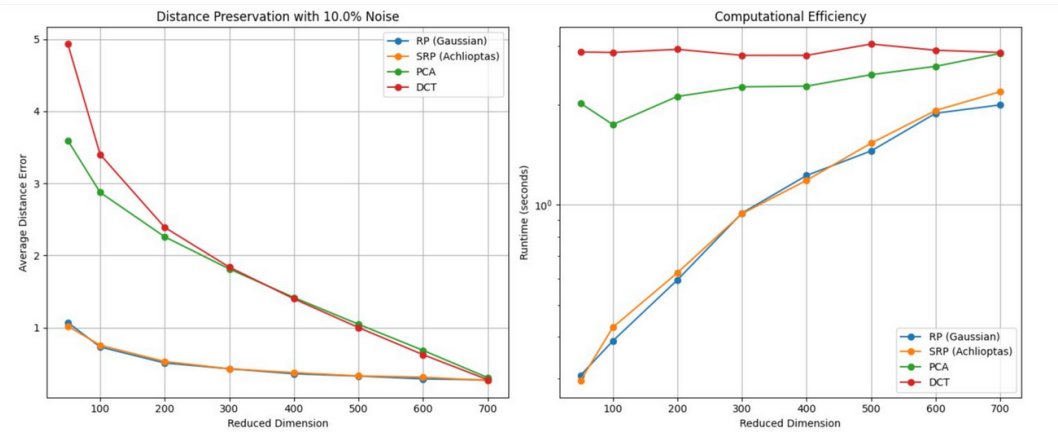


Figure 12: Dimension reduction on dataset with train data having 10 percent noise

The accuracy of random projections when the dataset contains noise is much larger as compared to the standard methods while having less computational complexity.
Figure 13 shows the dimension reduction on text data. While the accuracy produced by random projection is not very high as compared to image dataset, the accuracy is good enough as the cost of computation of random projections is much less.

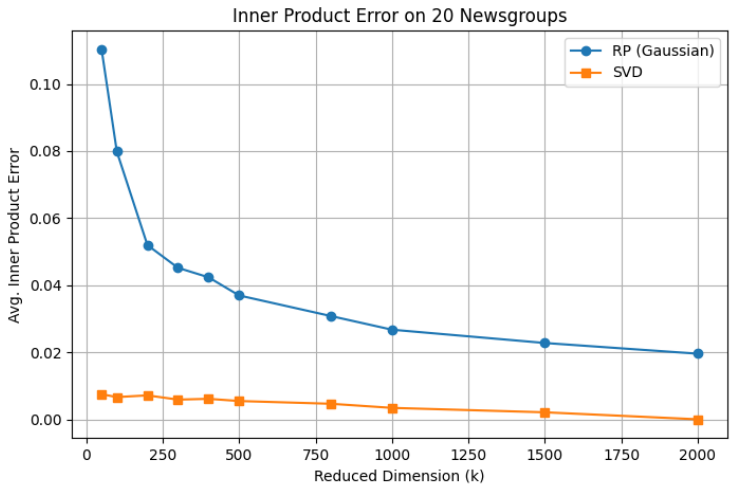


Figure 13: Dimension reduction on 20 Newsgroup dataset

Mathematical Description of Stochastic Neighbor Embedding (SNE)

Objective

Stochastic Neighbor Embedding (SNE) is a nonlinear dimensionality reduction technique that preserves local structures in high-dimensional data by modeling pairwise similarities as probability distributions. It minimizes the mismatch between similarity distributions in the original space and a low-dimensional embedding.

Similarity Modeling

High-Dimensional Space (Input)

Let

$$X = \{x_1, \ldots, x_N\} \in \mathbb{R}^d$$

be the high-dimensional data.

The similarity of x_j to x_i is modeled as a conditional probability under a Gaussian centered at x_i:

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)}, \quad p_{i|i} = 0$$

where σ_i controls the spread of the Gaussian at x_i.

Low-Dimensional Space (Embedding)

Let

$$Y = \{y_1, \ldots, y_N\} \in \mathbb{R}^k \quad (k \ll d)$$

be the low-dimensional embedding.

The similarity of y_j to y_i is modeled as:

$$q_{j|i} = \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_{k \neq i} \exp\left(-\|y_i - y_k\|^2\right)}, \quad q_{i|i} = 0$$

Here, the variance of the Gaussian is fixed to $\frac{1}{\sqrt{2}}$ for simplicity.

Cost Function

SNE minimizes the Kullback-Leibler (KL) divergence between the distributions P_i (high-dimensional) and Q_i (low-dimensional) for all datapoints:

$$C = \sum_{i=1}^{N} KL(P_i \parallel Q_i) = \sum_{i=1}^{N} \sum_{j \neq i} p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

KL divergence asymmetry: Heavy penalties occur if $q_{j|i} \ll p_{j|i}$ (widely separated embeddings for nearby points), ensuring local structure preservation.

Adaptive Variance (σ_i) via Perplexity

Perplexity: A user-defined parameter that smooths the effective number of neighbors. For each x_i, σ_i is chosen such that:

$$\text{Perp}(P_i) = 2^{H(P_i)}, \quad H(P_i) = -\sum_{j \neq i} p_{j|i} \log_2 p_{j|i}$$

where $H(P_i)$ is the Shannon entropy of P_i.

Binary search: For each x_i, σ_i is tuned to achieve the target perplexity (typically 5–50).

Gradient Descent Optimization

Gradient of the Cost Function

The gradient of C with respect to y_i is:

$$\frac{\partial C}{\partial y_i} = 2 \sum_{j \neq i} \Big(p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j} \Big)(y_i - y_j)$$

Physical interpretation: Each pair (y_i, y_j) is connected by a spring. The force exerted is proportional to the stiffness $\Big(p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j} \Big)$ and the distance $(y_i - y_j)$.

Update Rule with Momentum

The gradient is minimized using momentum-augmented gradient descent:

$$Y(t) = Y(t-1) + \eta \frac{\partial C}{\partial Y} + \alpha(t) \Big(Y(t-1) - Y(t-2) \Big)$$

where:

- η: Learning rate.
- $\alpha(t)$: Momentum coefficient at iteration t.

Simulated Annealing

Gaussian noise is added to Y in early iterations to escape poor local minima.

Noise variance decays gradually, analogous to annealing.

Mathematical Description of t-Distributed Stochastic Neighbor Embedding (t-SNE)

Objective

t-SNE is a nonlinear dimensionality reduction method designed to address the limitations of SNE (e.g., difficult optimization, crowding problem). It preserves local and global structures by:

Using a symmetrized cost function with simpler gradients.

Employing a heavy-tailed Student t-distribution in the low-dimensional space to alleviate crowding.

Symmetric SNE

joint Probability Distributions
High-dimensional space: Define pairwise similarities p_{ij} as symmetrized conditional probabilities:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}, \quad p_{ii} = 0$$

where $p_{j|i}$ follows the Gaussian-based SNE formulation:

$$p_{j|i} = \frac{\exp\Big(-\|x_i - x_j\|^2/(2\sigma_i^2)\Big)}{\sum_{k\neq i}\exp\Big(-\|x_i - x_k\|^2/(2\sigma_i^2)\Big)}$$

Low-dimensional space: Define q_{ij} using a Gaussian kernel:

$$q_{ij} = \frac{\exp\Big(-\|y_i - y_j\|^2\Big)}{\sum_{k\neq l}\exp\Big(-\|y_k - y_l\|^2\Big)}, \quad q_{ii} = 0$$

Symmetric Cost Function
Minimize the Kullback-Leibler (KL) divergence between joint distributions P and Q:

$$C = KL(P \parallel Q) = \sum_{i\neq j} p_{ij}\log\frac{p_{ij}}{q_{ij}}$$

Gradient

$$\frac{\partial C}{\partial y_i} = 4\sum_{j\neq i}(p_{ij} - q_{ij})(y_i - y_j)$$

The Crowding Problem

Issue: In high-dimensional space, the volume of a sphere scales as r^m (for radius r, dimension m). When embedding into lower dimensions (e.g., 2D), moderately distant points in high-D occupy insufficient area in low-D, causing "crowding" (points collapse into clusters).

Example: For a 10D manifold embedded in 2D, the ratio of areas for moderate vs. small distances is mismatched, leading to artificial repulsion/attraction forces.

t-SNE: Heavy-Tailed Distributions

Low-Dimensional Similarities
Replace the Gaussian kernel with a Student t-distribution (1 degree of freedom, Cauchy):

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k\neq l}(1 + \|y_k - y_l\|^2)^{-1}}$$

Key Property: The heavy tails alleviate crowding by allowing moderate high-D distances to map to larger low-D distances without excessive penalty.

Gradient of t-SNE Cost Function

$$\frac{\partial C}{\partial y_i} = 4\sum_{j\neq i}(p_{ij} - q_{ij})(y_i - y_j)(1 + \|y_i - y_j\|^2)^{-1}$$

Interpretation: Attraction forces scale inversely with distance, while repulsion forces dominate for mismatched large p_{ij} and small q_{ij}.

Optimization Strategies

Early Exaggeration
Multiply all p_{ij} by a factor (e.g., 4) for the first 50 iterations. This encourages tight clusters and creates space for global reorganization.

Effect: Focuses optimization on large p_{ij}, pulling similar points together early.

Momentum and Learning Rate
Update rule with momentum:

$$Y(t) = Y(t-1) + \eta \frac{\partial C}{\partial Y} + \alpha(t)\Big(Y(t-1) - Y(t-2)\Big)$$

Initial momentum $\alpha(t) = 0.5$ (first 250 iterations), then $\alpha(t) = 0.8$.
Learning rate $\eta = 100$, adapted using Jacobs' scheme.

Adaptive Learning Rate
Adjust η based on gradient stability to accelerate convergence.

Algorithm (Pseudocode)

Input: Data X, perplexity $Perp$, iterations T, learning rate η, momentum $\alpha(t)$.

Compute p_{ij} via binary search for σ_i to match $Perp$.

Initialize $Y(0) \sim N(0, 10^{-4}I)$.

For $t = 1$ to T:
Compute q_{ij} using Student t-distribution.
Compute gradient $\frac{\partial C}{\partial Y}$.
Update $Y(t)$ with momentum.

Output: Low-dimensional embedding $Y(T)$.

Derivation of the t-SNE gradient

t-SNE minimizes the Kullback-Leibler divergence between the joint probabilities p_{ij} in the high-dimensional space and the joint probabilities q_{ij} in the low-dimensional space. The values of p_{ij} are defined to be the symmetrized conditional probabilities, whereas the values of q_{ij} are obtained by means of a Student-t distribution with one degree of freedom:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n},$$
$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|y_k - y_l\|^2)^{-1}},$$

where $p_{j|i}$ and $p_{i|j}$ are either obtained by conditional probability (Gaussian that is centered on datapoint xi) or from the random walk procedure. The values of p_{ii} and q_{ii} are set to zero. The Kullback-Leibler divergence between the two joint probability distributions P and Q is given by

$$C = \mathrm{KL}(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} = \sum_i \sum_j \left(p_{ij} \log p_{ij} - p_{ij} \log q_{ij}\right). \tag{.1}$$

In order to make the derivation less cluttered, we define two auxiliary variables d_{ij} and Z as follows:

$$d_{ij} = \|y_i - y_j\|,$$
$$Z = \sum_{k \neq l}(1 + d_{kl}^2)^{-1}.$$

Note that if y_i changes, the only pairwise distances that change are d_{ij} and d_{ji} for $\forall j$. Hence, the gradient of the cost function C with respect to y_i is given by

$$\frac{\partial C}{\partial y_i} = \sum_j \left(\frac{\partial C}{\partial d_{ij}} + \frac{\partial C}{\partial d_{ji}}\right)(y_i - y_j) = 2\sum_j \frac{\partial C}{\partial d_{ij}}(y_i - y_j). \tag{.2}$$

The gradient $\frac{\partial C}{\partial d_{ij}}$ is computed from the definition of the Kullback-Leibler divergence in Equation 1:

$$\begin{aligned}\frac{\partial C}{\partial d_{ij}} &= -\sum_{k\neq l} p_{kl}\frac{\partial(\log q_{kl})}{\partial d_{ij}} \\ &= -\sum_{k\neq l} p_{kl}\frac{\partial}{\partial d_{ij}}\left(\log\frac{(1+d_{kl}^2)^{-1}}{Z}\right) \\ &= -\sum_{k\neq l} p_{kl}\left[\frac{1}{q_{kl}Z}\frac{\partial(1+d_{kl}^2)^{-1}}{\partial d_{ij}} - \frac{1}{Z}\frac{\partial Z}{\partial d_{ij}}\right].\end{aligned}$$

The gradient $\frac{\partial(1+d_{kl}^2)^{-1}}{\partial d_{ij}}$ is only non-zero when $k=i$ and $l=j$. Hence, the gradient $\frac{\partial C}{\partial d_{ij}}$ simplifies to:

$$\begin{aligned}\frac{\partial C}{\partial d_{ij}} &= 2p_{ij}(1+d_{ij}^2)^{-1} - 2q_{ij}(1+d_{ij}^2)^{-1} \\ &= 2(p_{ij}-q_{ij})(1+d_{ij}^2)^{-1}.\end{aligned}$$

Substituting this term into the gradient equation we obtain:

$$\frac{\partial C}{\partial y_i} = 4\sum_j (p_{ij}-q_{ij})(1+\|y_i-y_j\|^2)^{-1}(y_i-y_j). \tag{.3}$$

Scalable t-SNE for Large Datasets via Landmark Sampling and Random Walks

Challenge: Quadratic Complexity

Standard t-SNE has computational/memory complexity $\mathcal{O}(N^2)$, infeasible for $N > 10,000$. Direct subsampling loses structural information. Solution: Use **landmark points** $\mathcal{L} \subset \{1,...,N\}$ with random walk affinities.

Methodology

Neighborhood Graph Construction

- Build k-NN graph ($k=20$) with edge weights:

$$w_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2)$$

- Edge exists only between k-nearest neighbors

Random Walk Affinity Calculation

For each landmark $\mathbf{x}_i \in \mathcal{L}$:

$$P(\mathbf{x}_j|\mathbf{x}_i) = \frac{w_{ij}}{\sum_{k\in\mathcal{N}(i)} w_{ik}} \tag{.4}$$

Affinity $p_{j|i}$: Fraction of walks from $\mathbf{x}_i$ terminating at $\mathbf{x}_j$:

$$p_{j|i} = \frac{\text{\# walks from } \mathbf{x}_i \text{ to } \mathbf{x}_j}{\text{Total \# walks from } \mathbf{x}_i}$$

Analytical Alternative

Solve linear system for absorption probabilities:

$$(\mathbf{I} - \mathbf{P}_{\text{non-landmark}})\mathbf{F} = \mathbf{P}_{\text{landmark}} \tag{.5}$$

where $\mathbf{P}_{\text{non-landmark}}$ = transition matrix between non-landmarks, $\mathbf{F}$ contains $p_{j|i}$.

t-SNE on Landmarks

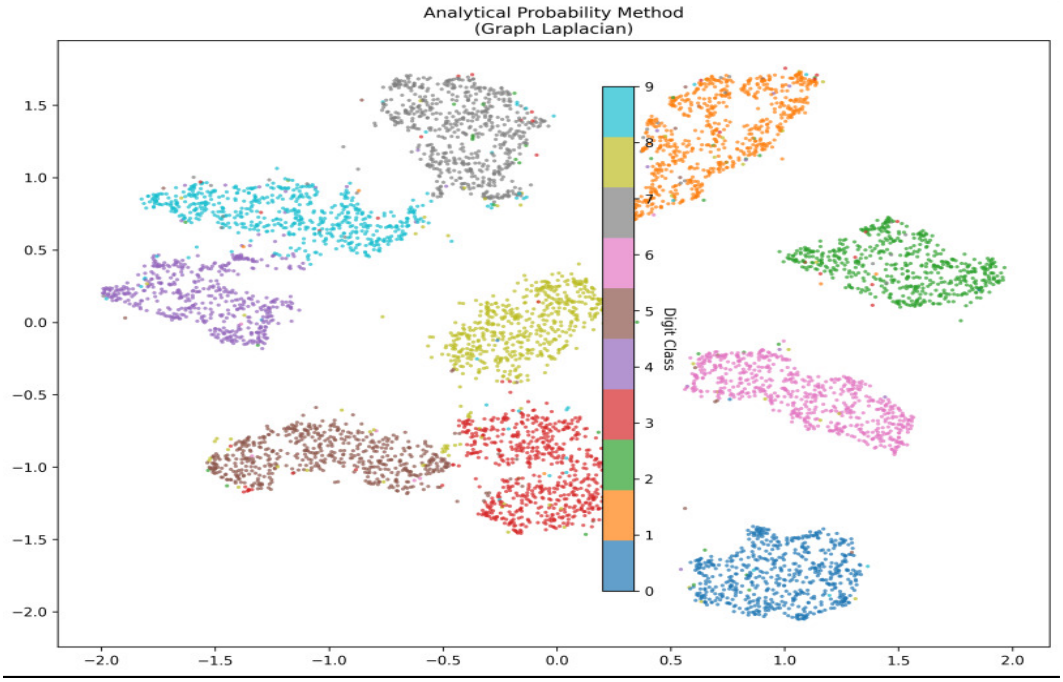
$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq l \in \mathcal{L}} (1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}} \tag{.6}$$

Cost function:

$$C = \sum_{i,j \in \mathcal{L}} p_{j|i} \log \frac{p_{j|i}}{q_{j|i}} \tag{.7}$$

Gradient:

$$\frac{\partial C}{\partial \mathbf{y}_i} = 4 \sum_{j \in \mathcal{L}} (p_{j|i} - q_{j|i})(\mathbf{y}_i - \mathbf{y}_j)(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1} \tag{.8}$$



t-SNE for Large Datasets (MNIST)

Experimental Validation (MNIST)

- Dataset: 6000 MNIST digits, 6000 landmarks
- Results: Clear class separation
- Computation: 5 hour CPU time

Advantages

- Complexity: $\mathcal{O}(|\mathcal{L}|^2 + N \log N)$
- Robustness: Integrates global structure via paths
- Preservation: Non-landmarks influence through walks

Limitations

- Preprocessing: $\mathcal{O}(N \log N)$ for k-NN
- Sparsity: Requires dense landmark sampling

Summary Table

Component	Implementation
Complexity Reduction	$\mathcal{O}(\lvert\mathcal{L}\rvert^2)$ pairwise calculations
Structure Preservation	Random walk path integration
Noise Robustness	Multiple path averaging
Landmark Initialization	Uniform random sampling
Gradient Updates	Momentum-based optimization

Analytical Solution to Random Walk Probabilities

It can be shown that computing the probability that a random walk initiated from a non-landmark point (on a graph specified by adjacency matrix W) first reaches a specific landmark point is equivalent to solving the combinatorial Dirichlet problem where boundary conditions are at landmark locations, with the considered landmark fixed to unity and others set to zero . In practice, the solution is obtained by minimizing the Dirichlet integral:

$$D[x] = \frac{1}{2} x^\top L x, \tag{.9}$$

where L represents the graph Laplacian given by $L = D - W$, with $D = \operatorname{diag}\left(\sum_j w_{1j}, \sum_j w_{2j}, \ldots, \sum_j w_{nj}\right)$. Reordering landmarks first, the Dirichlet integral decomposes as:

$$D[x_N] = \frac{1}{2} \begin{bmatrix} x_L^\top & x_N^\top \end{bmatrix} \begin{bmatrix} L_{LL} & B \\ B^\top & L_{NN} \end{bmatrix} \begin{bmatrix} x_L \\ x_N \end{bmatrix} = \frac{1}{2} \left(x_L^\top L_{LL} x_L + 2 x_N^\top B^\top x_L + x_N^\top L_{NN} x_N \right), \tag{.10}$$

where subscript $\cdot_L$ denotes landmarks and $\cdot_N$ non-landmarks. Differentiating $D[x_N]$ with respect to x_N yields the linear system:

$$L_{NN} x_N = -B^\top. \tag{.11}$$

Here $B^\top$ contains columns from L corresponding to landmarks (excluding landmark rows). After normalizing solutions X_N, its columns contain termination probabilities for random walks from non-landmarks. This system is nonsingular if the graph is connected or each component contains at least one landmark.

To compute landmark-to-landmark walk probabilities, we duplicate landmarks and solve:

$$L_{NN} = CC^\top \quad \text{(Cholesky factorization)}, \tag{.12}$$

$$Cy = -B^\top \quad \text{and} \quad C^\top x_N = y \quad \text{(backsubstitution)}. \tag{.13}$$

Comparative Analysis of Dimensionality Reduction Techniques

Classical Scaling (Multidimensional Scaling, MDS)

Objective: Preserve pairwise distances via linear projection.

Cost Function:

$$C_{\text{MDS}} = \sum_{i<j} \left(\|\mathbf{x}_i - \mathbf{x}_j\| - \|\mathbf{y}_i - \mathbf{y}_j\| \right)^2 \tag{.14}$$

Weaknesses:

- Focuses on large pairwise distances, neglecting local structure
- Linear method; fails to model curved manifolds

Sammon Mapping

Objective: Preserve distances with weighted emphasis on small errors.

Cost Function:

$$C_{\text{Sammon}} = \frac{1}{\sum_{i<j} \|\mathbf{x}_i - \mathbf{x}_j\|} \sum_{i<j} \frac{\left(\|\mathbf{x}_i - \mathbf{x}_j\| - \|\mathbf{y}_i - \mathbf{y}_j\|\right)^2}{\|\mathbf{x}_i - \mathbf{x}_j\|} \tag{.15}$$

Weaknesses:

- Overemphasizes small distances
- No adaptive local scaling

Isomap

Objective: Preserve geodesic distances on a neighborhood graph.

Steps:

1. Construct k-NN graph
2. Compute geodesic distances (shortest paths)
3. Apply MDS to geodesic distances

Weaknesses:

- Susceptible to short-circuiting
- Prioritizes large geodesic distances

Locally Linear Embedding (LLE)

Objective: Preserve local linear reconstructions.

Reconstruction:

$$\min_{\{w_{ij}\}} \left\| \mathbf{x}_i - \sum_{j \in \mathcal{N}(i)} w_{ij} \mathbf{x}_j \right\|^2 \quad \text{s.t.} \quad \sum_j w_{ij} = 1 \tag{.16}$$

Weaknesses:

- Covariance constraint causes collapse
- Fails on disconnected manifolds

Diffusion Maps

Objective: Preserve diffusion distances integrating all paths.

Diffusion Distance:

$$D^{(t)}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_k \frac{\left(p_{ik}^{(t)} - p_{jk}^{(t)}\right)^2}{\psi(\mathbf{x}_k)}} \tag{.17}$$

Weaknesses:

- Emphasizes large distances
- No principled way to select t

Curvilinear Component Analysis (CCA)

Objective: Preserve local distances with hard threshold.

Cost Function:

$$C_{\text{CCA}} = \sum_{i<j} \left(\|\mathbf{x}_i - \mathbf{x}_j\| - \|\mathbf{y}_i - \mathbf{y}_j\|\right)^2 \cdot \mathbb{I}(\|\mathbf{x}_i - \mathbf{x}_j\| < \lambda) \tag{.18}$$

Weaknesses:

- Hard threshold creates abrupt transitions
- Overpenalizes small errors

Maximum Variance Unfolding (MVU)

Objective: Maximize variance while preserving local distances.

Optimization:

$$\max_{\{\mathbf{y}_i\}} \text{Tr}(\mathbf{Y}^T\mathbf{Y}) \quad \text{s.t.} \quad \|\mathbf{y}_i - \mathbf{y}_j\|^2 = \|\mathbf{x}_i - \mathbf{x}_j\|^2 \quad \forall (i,j) \in \mathcal{E} \tag{.19}$$

Weaknesses:

- Sensitive to noisy constraints
- Ignores global structure

Laplacian Eigenmaps

Objective: Minimize graph Laplacian-based energy.

Cost Function:

$$\min_{\{\mathbf{y}_i\}} \sum_{i<j} \|\mathbf{y}_i - \mathbf{y}_j\|^2 w_{ij} \quad \text{s.t.} \quad \mathbf{Y}^T\mathbf{D}\mathbf{Y} = \mathbf{I} \tag{.20}$$

Weaknesses:

- Covariance constraint limitations
- Requires connected graphs

t-SNE: Key Advantages

- **Student t-Distribution:**

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}} \tag{.21}$$

- **Adaptive Perplexity:**

$$\text{Perp}(P_i) = 2^{H(P_i)}, \quad H(P_i) = -\sum_j p_{j|i} \log_2 p_{j|i} \tag{.22}$$

- **Gradient Efficiency:**

$$\frac{\partial C}{\partial \mathbf{y}_i} = 4 \sum_j (p_{ij} - q_{ij})(\mathbf{y}_i - \mathbf{y}_j)(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1} \tag{.23}$$

Summary Table

Method	Key Weakness	t-SNE Advantage
Classical Scaling	Prioritizes large distances	Balances local/global
Sammon	Overemphasizes tiny errors	Soft border via perplexity
Isomap	Short-circuiting	Path integration
LLE	Covariance collapse	Probabilistic embeddings
Diffusion Maps	Fixed t parameter	Adaptive structure
CCA/MVU	Hard thresholds	Probabilistic affinities
Laplacian	Connected graphs only	Disconnected handling

Supplementary experiments 2

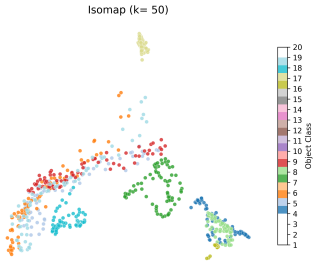


Figure 14: *
(a) Isomap with 50 neighbors

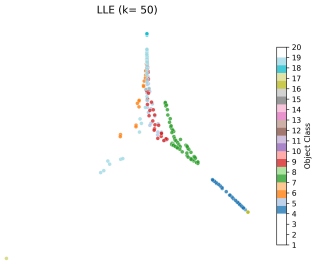


Figure 15: *
(b) LLE with 50 neighbors

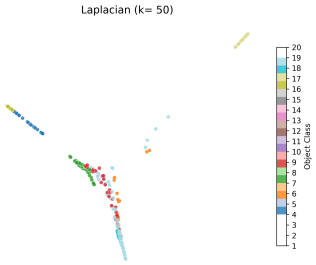


Figure 16: *
(c) Laplacian Eigenmap with 50 neighbors

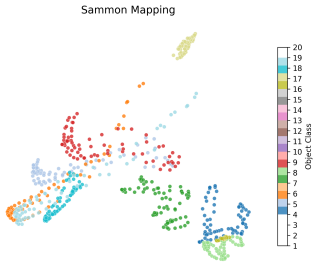


Figure 17: *
(d) Sammon mapping

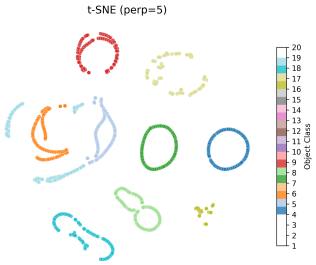


Figure 18: *
(e) tSNE with perplexity= 5

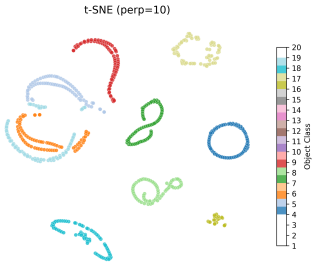


Figure 19: *
(f) tSNE with perplexity= 10

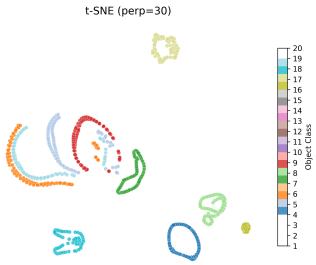


Figure 20: *
(g) tSNE with perplexity= 30

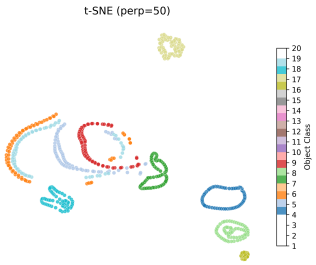


Figure 21: *
(h) tSNE with perplexity= 50

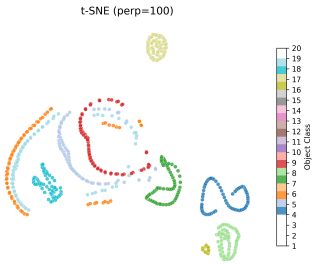


Figure 22: *
(i) tSNE with perplexity= 100

The above images are the visualizations of Coil-20 dataset, 1440 grayscale images of 20 objects with 72 views. As seen from the images, t-SNE is a much superior method for visualization of the data. The following are the visualizations generated for Olivetti faces dataset, which consists of 400 images of 40 individuals, each having 10 images. From below again tSNE is preferred over other methods for visualization.

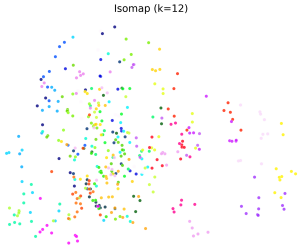


Figure 23: *
(a) Isomap with 12 neighbors

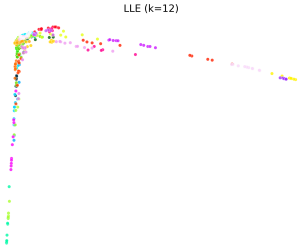


Figure 24: *
(b) LLE with 12 neighbors

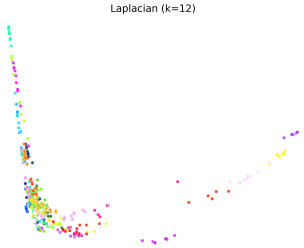


Figure 25: *
(c) Laplacian Eigenmap with 12 neighbors

Figure 26: *
(d) Sammon mapping

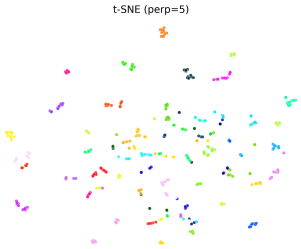


Figure 27: *
(e) tSNE with perplexity= 5

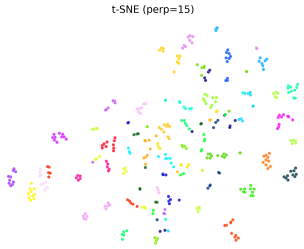


Figure 28: *
(f) tSNE with perplexity= 15

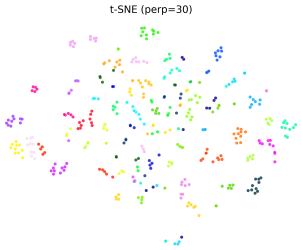


Figure 29: *
(g) tSNE with perplexity= 30

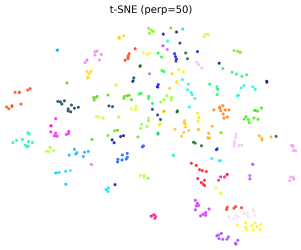


Figure 30: *
(h) tSNE with perplexity= 50

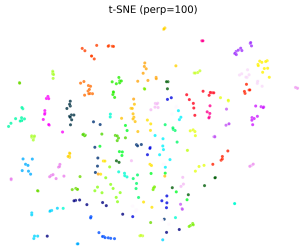


Figure 31: *
(i) tSNE with perplexity= 100

Gaussian Mixture Variational Autoencoders for Multi-Label Classification

Problem Setup

Multi-Label Classification (MLC):
Given a dataset

$$D = \{(x_n, y_n)\}_{n=1}^{N}$$

where $x_n \in \mathbb{R}^D$ are features and $y_n \in \{0,1\}^L$ are binary labels, learn a mapping

$$f : \mathbb{R}^D \to \{0,1\}^L.$$

Goal: Capture correlations between L labels to improve classification.

Gaussian Mixture VAE (GMVAE)

Key Idea: Use a Gaussian mixture prior $p(z)$ to model label correlations.

Gaussian Mixture Prior
For K label-associated components:

$$p(z) = \frac{1}{k} \sum_{i=1}^{k} \mathcal{N}(z \mid \mu_i, \sigma_i^2)$$

μ_i, σ_i^2: Learnable parameters for label i.
Given a random variable z, the probability density function (PDF) in the subspace is defined as

$$p_\psi(z \mid y) = \frac{1}{\sum_i y_i} \sum_{i=1}^{L} \mathbb{1}\{y_i = 1\} \mathcal{N}(z \mid \mu_i, \text{diag}(\sigma_i^2))$$

where $\mathbb{1}(\cdot)$ is the indicator function and the label encoder is parameterized by ψ (NN).

Modified ELBO

$$\mathcal{L}_{\text{GMVAE}} = \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x|z)\right] - \text{KL}\Big(q_\phi(z|x) \,\|\, p(z|y)\Big)$$

KL Divergence Challenge: $\text{KL}(q \,\|\, p)$ between Gaussian q and mixture p lacks closed-form.
Solution: Monte Carlo approximation:

$$\text{KL}(q \,\|\, p) \approx \log q(z|x) - \log \sum_{i \in S} \mathcal{N}\big(z \mid \mu_i,\, \sigma_i^2 I\big)$$

Contrastive Gaussian Mixture Variational Autoencoder (C-GMVAE)

Framework Overview

C-GMVAE extends the VAE framework with a learnable Gaussian mixture (GM) prior and contrastive learning to model multi-label correlations. Key components:

- Label-Driven GM Prior: Each label corresponds to a Gaussian subspace; active labels define a mixture prior.
- Contrastive Loss: Aligns feature and label embeddings to capture co-occurrence patterns.
- Cross-Entropy Loss: Directly optimizes label predictions.

Gaussian Mixture Latent Space

Label Embeddings as Gaussians
Each label i is mapped to a Gaussian

$$\mathcal{N}\big(\mu_i,\, \mathrm{diag}(\sigma_i^2)\big).$$

Label encoder:

$$\mu_i,\, \sigma_i^2 = f_\theta(w_{il}),$$

where $w_{il} \in \mathbb{R}^E$ is the label embedding.

Mixture Prior for Multi-Label Data
Given labels $y \in \{0,1\}^L$, the prior is a mixture of Gaussians for active labels:

$$p_\psi(z \mid y) = \frac{1}{\sum_i y_i} \sum_{i=1}^{L} \mathbb{1}\{y_i = 1\}\, \mathcal{N}(z \mid \mu_i, \mathrm{diag}(\sigma_i^2))$$

Posterior and KL Divergence Posterior:

$$q_\phi(z \mid x) = \mathcal{N}\big(z \mid \mu_\phi(x),\, \mathrm{diag}(\sigma_\phi^2(x))\big).$$

KL Approximation: Monte Carlo estimation for non-analytic KL divergence:

$$\mathcal{L}_{KL} \approx \log q_\phi(z_0 \mid x) - \log p_\psi(z_0 \mid y)$$

The reconstruction loss is a standard negative log-likelihood with decoder parameters

$$\mathcal{L}_{\text{recon}} = -\mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x \mid z)\right]$$

Contrastive Learning Module

Feature and Label Embeddings
Feature embedding:

$$w_x^f = f_d(z) \in \mathbb{R}^E.$$

Label embedding:

$$w_i^l \in \mathbb{R}^E \quad \text{(learned)}.$$

Contrastive Loss
For a batch B, contrastive loss enforces feature-label alignment:

$$L_{CL} = -\frac{1}{|B|} \sum_{(x,y)\in B} \frac{1}{|P(y)|} \sum_{p\in P(y)} \log \frac{\exp\Big(w_x^f \cdot w_p^l/\tau\Big)}{\sum_{t\in A} \exp\Big(w_x^f \cdot w_t^l/\tau\Big)},$$

where

$$P(y) = \{i : y_i = 1\} \quad \text{(Positive labels)},$$
$$A = \{1, \ldots, L\} \quad \text{(All labels)},$$

and τ is the temperature scaling parameter.

Cross-Entropy Loss
Direct label prediction via sigmoid probabilities:

$$L_{CE} = -\sum_{i=1}^{L} \Big[y_i \log s\Big(w_x^f \cdot w_i^l\Big) + (1-y_i) \log\Big(1 - s\Big(w_x^f \cdot w_i^l\Big)\Big)\Big],$$

where $s(\cdot)$ is the sigmoid function.

Total Objective Function

$$\mathcal{L} = \mathcal{L}_{KL} + \mathcal{L}_{\text{recon}} + \alpha\mathcal{L}_{CL} - \beta\mathcal{L}_{CE}$$

where α and β are trade-off weights.

Connection with Triplet Loss

Triplet loss is one of the popular ranking losses used in multi-label learning.

Given an anchor embedding $\mathbf{v}_x^f$, a positive embedding $\mathbf{v}_+$ and a negative embedding $\mathbf{v}_-$, they form a triplet $(\mathbf{v}_x^f, \mathbf{v}_+, \mathbf{v}_-)$. A triplet loss is defined as

$$L_{\text{trip}}(\mathbf{v}_x^f, \mathbf{v}_+, \mathbf{v}_-) = \max\{0,\ g + \text{dist}(\mathbf{v}_x^f, \mathbf{v}_+)\ -\ \text{dist}(\mathbf{v}_x^f, \mathbf{v}_-)\} \tag{1}$$

where g is a gap parameter measuring the distance between $(\mathbf{v}_x^f, \mathbf{v}_+)$ and $(\mathbf{v}_x^f, \mathbf{v}_-)$, and $\text{dist}(\cdot)$ is a distance function. This hinge loss L_{trip} encourages fewer violations to "positive $>$ negative" ranking order. Let $\alpha = \frac{1}{2}$. With the same triplet, we can write down a contrastive loss

$$\begin{aligned}
L_{CL}(\mathbf{v}_x^f, \mathbf{v}_+, \mathbf{v}_-) &= -\log\left[\frac{\exp\big(2\,\mathbf{v}_x^f \cdot \mathbf{v}_+\big)}{\sum_t \exp\big(2\,\mathbf{v}_x^f \cdot \mathbf{v}_t\big)}\right] \\
&= \log\Big(1 + \frac{\exp\big(2\,\mathbf{v}_x^f \cdot \mathbf{v}_-\big)}{\exp\big(2\,\mathbf{v}_x^f \cdot \mathbf{v}_+\big)}\Big) \\
&= 1 + \big(2\,\mathbf{v}_x^f \cdot \mathbf{v}_-\ -\ 2\,\mathbf{v}_x^f \cdot \mathbf{v}_+\big) \\
&= 1 + \Big(\|\mathbf{v}_x^f\|^2 + 2\,\mathbf{v}_x^f \cdot \mathbf{v}_- - \mathbf{v}_- \cdot \mathbf{v}_-\ +\ \|\mathbf{v}_x^f\|^2 - 2\,\mathbf{v}_x^f \cdot \mathbf{v}_+ + \mathbf{v}_+ \cdot \mathbf{v}_+\Big) \\
&= \|\mathbf{v}_x^f - \mathbf{v}_+\|^2 + \|\mathbf{v}_x^f - \mathbf{v}_-\|^2 + 1
\end{aligned} \tag{2}$$

Note that in the second to the last equation, $\mathbf{v}_+$ and $\mathbf{v}_-$ have the same norm due to the normalization in our contrastive learning module.

By setting $\text{dist}(\cdot)$ to the commonly used ℓ_2 distance and $g = 1$, Eq. (2) is a fair approximation of Eq. (1). Therefore, triplet loss can be viewed as a special case of the contrastive loss. In the contrastive loss, embeddings are normalized and more positives/negatives are available.contrastive loss generally outperforms triplet loss.

Supplementary Experiments 3

	method (data %)	HA	ex-F1	mi-F1	ma-F1
mirflickr	MPVAE (100%)	0.898	0.514	0.552	0.422
	C-GMVAE (50%)	0.899	0.512	0.553	0.412

Note that in the above, the value for MPVAE is taken from reference paper while the calculation for C-GMVAE was computed. The above shows that even when training only 50 percent of the data, C-GMVAE performs in similar level as compared to MPVAE.

Project Team: Individual Contributions

Abhinav Goyal

- Prepared the main part of the report (4 pages), excluding the appendix.
- Performed all the experiments, including:
  - Dimensionality reduction using random projection and other methods.
  - Visualizations using t-SNE and other methods.
  - C-GMVAE for multi-label classification.

Seetha Abhinav

- Prepared the majority of the appendix section of the report.
- Prepared the presentation slides (PPT).

Piyush Kumar

- Prepared a few parts of the appendix section of the report.
- Prepared the presentation slides (PPT).

Aarav Desai

- Created the GitHub repository for all the experiments.
- Collected the datasets used in all the experiments.
- Prepared a few parts of the appendix section of the report.