

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/387835819>

# t-SNE-PSO: Optimizing t-SNE using particle swarm optimization

Article in Expert Systems with Applications · April 2025

DOI: 10.1016/j.eswa.2025.126398

---

CITATIONS  
0

READS  
94

5 authors, including:



Mebarka Allaoui

Bishop's University

13 PUBLICATIONS 218 CITATIONS

[SEE PROFILE](#)



Samir Brahim Belhaouari

Hamad bin Khalifa University

298 PUBLICATIONS 2,969 CITATIONS

[SEE PROFILE](#)



Rachid Hedjam

Sultan Qaboos University

69 PUBLICATIONS 1,123 CITATIONS

[SEE PROFILE](#)



Khadra Bouanane

Kasdi Merbah University, Ouargla, Algeria

15 PUBLICATIONS 27 CITATIONS

[SEE PROFILE](#)

# t-SNE-PSO: Optimizing t-SNE using Particle Swarm Optimization

Mebarka Allaoui<sup>a,\*</sup>, Samir Brahim Belhaouari<sup>b</sup>, Rachid Hedjam<sup>a</sup>, Khadra Bouanane<sup>c,d</sup>,  
Mohammed Lamine Kherfi<sup>e,f</sup>

<sup>a</sup>*Department of Computer Science, Bishop's University, Lennoxville, Sherbrooke, Quebec, Canada*

<sup>b</sup>*Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa  
University, Education City, Doha, Qatar*

<sup>c</sup>*Kasdi Merbah University Ouargla, Ouargla, 30000, Ouargla, Algeria*

<sup>d</sup>*LINATI Laboratory, Kasdi Merbah University Ouargla, Ouargla, 30000, Ouargla, Algeria*

<sup>e</sup>*Department of Computer Science, Sultan Qaboos University, Oman*

<sup>f</sup>*LAMIA Laboratory, University of Quebec, 3351 Boulevard des Forges, C.P. 500, Trois-Rivières, Canada*

---

\*Corresponding author

Email addresses: [malliaoui@ubishops.ca](mailto:malliaoui@ubishops.ca) (Mebarka Allaoui), [sbelhaouari@hbku.edu.qa](mailto:sbelhaouari@hbku.edu.qa) (Samir Brahim Belhaouari), [rhedjam@ubishops.ca](mailto:rhedjam@ubishops.ca) (Rachid Hedjam), [bouanane.khadra@univ-ouargla.dz](mailto:bouanane.khadra@univ-ouargla.dz) (Khadra Bouanane), [Mohammedlamine.Kherfi@uqtr.ca](mailto:Mohammedlamine.Kherfi@uqtr.ca) (Mohammed Lamine Kherfi)

---

## Abstract

t-distributed Stochastic Neighbor Embedding (t-SNE) is a manifold embedding technique that utilizes the Stochastic Gradient Descent (GD) algorithm to optimize objective functions to preserve pairwise distances between high-dimensional inputs in the lower-dimensional representation. Gradient-based methods are known by their local searches and may not explore the search space widely, making them susceptible to becoming trapped in local minima. To address this limitation, an adapted Particle Swarm Optimization (PSO) technique for t-SNE is proposed. The proposed t-SNE-PSO algorithm aims to overcome the limitations of GD by introducing a dynamic update of cognitive and social coefficients in PSO for optimizing t-SNE, enhancing its ability to find global optima and strike a balance between exploration and exploitation. Furthermore, the updated PSO contributes to developing a more efficient and effective dimensionality reduction technique, demonstrating superior qualities in clustering and visualization. The evaluation results on various benchmarks demonstrate the effectiveness of the proposed t-SNE-PSO algorithm.

*Keywords:*

t-SNE, PSO, Optimization, Manifold embedding.

---

## 1. Introduction

The explosive growth of data in diverse fields has underscored the importance of efficient Dimensionality Reduction (DR) techniques to distill meaningful patterns from high-dimensional datasets. DR approaches are broadly categorized into several classes, each addressing distinct aspects of data representation. The main three classes mentioned in this study are namely (i) projection techniques like Principal Component Analysis (PCA) (Pearson, 1901), grounded in a linear transformation of the input data into a new feature space, where the variance of data is maximized along a select few principal components ; (ii) deep embedding techniques, exemplified by methods such as the Denoising Autoencoder (DAE) (Ranzato et al., 2007) and Convolutional Neural Networks (CNNs) (Radford et al., 2015), focusing on learning intricate hierarchical representations; and Lastly (iii) Manifold Embedding (ME) techniques with a focus on preserving the local structure of data points in the lower rank space, such as Locally Linear Embedding (LLE) (Roweis & Saul, 2000), Multidimensional Scaling (MDS), Isomap (Tenenbaum et al., 2000), t-distributed Stochastic Neighbor Embedding (t-SNE) (Van der Maaten & Hinton, 2008), and Uniform Manifold Approximation and Projection (UMAP) (McInnes et al., 2018).

t-SNE is an ME technique commonly used for visualizing high-dimensional data in lower-dimensional spaces, typically two or three dimensions. It works by modeling the relationships between data points in the high-dimensional space and then finding a low-dimensional representation that preserves these relationships as much as possible (Van der Maaten & Hinton, 2008). t-SNE achieves this by defining a probability distribution over pairs of data points in both the high-dimensional and low-dimensional spaces and minimizing the difference between these distributions using gradient descent to optimize t-SNE's Kullback-Leibler (KL) divergence loss function. The resulting low-dimensional representation often reveals the underlying structure of the data, making it useful for exploratory data analysis and visualization.

t-SNE has several challenges, particularly t-SNE struggles with high computational cost when dealing with large datasets. To address this, Several improvements and variations have been proposed to address t-SNE's challenges. For instance, the Barnes-Hut t-SNE (t-SNE-BH) variant introduced by van der Maaten incorporates the Barnes-Hut approximation to enhance scalability (Van Der Maaten, 2013). Another approach by Linderman et al. employs

an interpolation-based approximation to reduce time complexity for large datasets (Linderman et al., 2019). Another key drawback of t-SNE is its lack of out-of-sample extension, meaning new data points cannot be directly mapped into an existing embedding without re-running the entire optimization. This limitation stems from t-SNE’s non-parametric nature, which relies on dataset-specific pairwise similarities, making it computationally expensive to update embeddings for new data. While parametric approaches, such as training neural networks to approximate the embedding function, have been proposed (Gisbrecht et al., 2012; Crecchi et al., 2020).

Despite these advancements, the core optimization technique in these methods remains gradient descent, which is inherently prone to issues such as local minima and non-convex landscapes. It is well known that the optimization landscape of t-SNE is non-convex due to several key factors. First, t-SNE aims to preserve pairwise similarities between points in a high-dimensional space and maps them into a lower-dimensional space, but this optimization involves a non-convex cost function that may have multiple local minima (Van der Maaten & Hinton, 2008; Van Der Maaten, 2014). Second, the optimization problem is sensitive to the initialization of the embedding, and different initializations can lead to different local optima (Van Der Maaten, 2014). Additionally, the objective function of t-SNE relies on the Kullback-Leibler divergence, which is not convex and can result in non-smooth optimization landscapes (Hinton & Roweis, 2002). The perplexity parameter in t-SNE can also affect the cost function’s shape and create different optimization dynamics (Van Der Maaten et al., 2009). Finally, t-SNE’s reliance on stochastic gradient descent for optimization further contributes to its non-convex nature, as the process may converge to local minima depending on the learning rate and iteration settings. According to (Hinton & Roweis, 2002) “Steepest descent in which all of the points are adjusted in parallel is inefficient and can get stuck in poor local optima”. Therefore, Particle Swarm Optimization (PSO) is more suitable than gradient descent for non-convex problems because it does not rely on gradient information and avoids getting stuck in local minima (Kennedy & Eberhart, 1995). PSO uses a population of particles that explore the search space globally, and each particle is guided by its own experience and the experience of its neighbors, making it more robust to complex, non-convex landscapes.

By leveraging the power of PSO in optimizing the loss function of t-SNE, the proposed t-SNE-PSO algorithm seeks to address the challenges faced in traditional gradient-based optimization methods. Furthermore, we propose a novel rule for automatically updating the cognitive and social factors of PSO, diverging from the conventional preset values. This dynamic update of the cognitive and social coefficients allows our algorithm to systematically control the balance between exploration and exploitation, making it more robust for high-dimensional data. The detail is given in Sec. 4.

The rest of this paper is organized as follows: Section 2 reviews the related works. In Sec. 3, we provide the notations and the preliminaries of t-SNE and PSO techniques. Section 4 provides a detailed description of the proposed methodology, optimizing the loss function of t-SNE using PSO adaptation and the role of PSO parameters. We present experimental results demonstrating the efficacy of our approach across various datasets, showcasing improvements in both computational performance and the preservation of intrinsic data characteristics in Sec. 5. The advantages, limitations, and some applications of t-SNE-PSO are provided in Sec. 6. Finally, Sec. 7 concludes the paper.

## 2. Related works

### 2.1. Variants of t-SNE

t-SNE has garnered significant attention due to its ability to handle large datasets and unveil meaningful clustering structures. However, despite its usefulness, t-SNE has faced criticism

for its limited scalability when dealing with substantial data volumes and its tendency to lack in preserving the global structure. Specifically, t-SNE tends to focus on local clusters that are arbitrarily scattered in the low-dimensional space, and it lacks theoretically-founded implementations for mapping new data into existing embeddings (Ding et al., 2018; Becht et al., 2019). These identified shortcomings of t-SNE have spurred various studies aimed at addressing these limitations. Notably, van der Maaten proposed an alternative approach utilizing neural networks (Van Der Maaten, 2009). To tackle scalability and computation speed issues for large datasets, an optimized variant known as Barnes-Hut t-SNE was introduced by van der Maaten (Van Der Maaten, 2013). This variant incorporates the Barnes-Hut approximation technique, which enhances the efficiency of force-directed algorithms, contributes to improved scalability, and reduces the computational complexity from  $O(n^2)$  to  $O(n \log n)$ . While it generally maintains high quality, this approximation can introduce small inaccuracies in the embeddings. In addition, the use of additional data structures for the Barnes-Hut approximation may increase memory usage, which can be a concern for extremely large datasets or memory-limited environments. The Fast Interpolation-based t-SNE (FIt-SNE) (Linderman et al., 2019) significantly accelerates the t-SNE algorithm by using interpolation-based techniques. This effectively reduces the time complexity to be linearly dependent on the number of samples. While FIt-SNE provides significant speedups, it does so by approximating the calculations. These approximations introduce errors, especially in complex or noisy datasets. Although faster, FIt-SNE is resource-intensive with large datasets, particularly in terms of memory usage. This might limit its applicability on standard desktop computers. Building upon this work, Han developed c-TSNE, an explainable variant of t-SNE that incorporates biologically meaningful distance metrics, offering enhanced interpretability in the visualization process (Han et al., 2022). However, this method, based on individual genes' contribution to the t-SNE embedding, works only for biological studies and can not extend to other types of datasets. In a different vein, Kobak et al. presented an efficient implementation of t-SNE specifically designed for a t-distribution kernel with an arbitrary degree of freedom (Kobak et al., 2019). Their research demonstrated that setting the degree of freedom to less than one can effectively address the crowding problem and reveal finer cluster structures. Kobak et al. also proposed additional techniques to fine-tune hyperparameters and enhance global positioning, such as estimating similarities using a mixture of Gaussian kernels (Kobak & Berens, 2019). Although these two studies provide the t-SNE user with how to manipulate its hyper-parameters to obtain better results, they do not highlight any new involvement in the t-SNE components. For large-scale vision datasets, tSNE-CUDA (Chan et al., 2019), an optimized implementation of the t-SNE algorithm on the GPU, was introduced. This approach leverages natural parallelism and techniques designed for the n-body problem, enabling scalability to handle extensive datasets. However, GPU acceleration requires access to suitable hardware, which may not be universally available, costly, or accessible to all researchers. This hardware dependency can limit the broader adoption of GPU-accelerated t-SNE. A notable development in t-SNE optimization is the use of Nesterov's momentum for accelerating convergence without the need for early exaggeration. This approach improves the optimization dynamics and neighborhood preservation in embeddings, making it particularly effective for large and complex datasets (Lambert et al., 2023). Another variant, t-SimCNE, adapts contrastive learning techniques to generate 2D embeddings by replacing cosine similarity with a heavy-tailed Cauchy kernel, which reduces the crowding problem common in standard t-SNE (Böhm et al., 2022).

While these variation techniques of t-SNE have demonstrated advancements in addressing certain limitations of t-SNE, it's noteworthy that many of these techniques leverage GD optimization for dimensionality reduction. However, it is essential to highlight that using GD comes with challenges, such as potential convergence to local minima and difficulty handling noisy or non-convex landscapes.

## 2.2. PSO for Dimensionality Reduction

Several alternative optimization techniques have been explored to tackle the limitations of GD. Mainly, numerous variants of Swarm Intelligence (SI) techniques have been developed to address complex and large-scale optimization challenges (Gad, 2022). SI constitutes a significant domain within optimization methods, characterized by systems where agents interact locally with their surroundings, leading to the emergence of coherent global patterns. Drawing inspiration from natural phenomena such as fish schooling, honey bee swarming, bacterial growth, animal herding, bird flocking, and ant colony foraging, SI techniques model agents' simple behaviors and self-organizing interactions to tackle optimization tasks effectively. The exploration of Swarm Intelligence optimization techniques (Yab et al., 2022; Yang & Du, 2011; Duan et al., 2017), in dimensionality reduction, has garnered substantial attention due to their ability to navigate complex, high-dimensional spaces in search of optimal solutions. In our study, we use PSO to address the limitations of the GD algorithms. We chose to employ PSO to complement t-SNE for the following reasons:

- The main advantage of PSO is that it has fewer parameters to tune than other swarm intelligence techniques (Esmin et al., 2015).
- PSO has several characteristics that make it a good choice: its parallelization and ability to balance exploration and exploitation. The swarm's collective behavior allows particles to explore the solution space globally while converging toward promising regions.
- PSO's adaptability and simplicity suit many optimization problems. Due to its versatility, it has been successfully applied in various domains, including health care, smart city, and general aspects (Gad, 2022).

Prior research has demonstrated the efficacy of PSO in various optimization tasks, and its application to dimensionality reduction has been explored extensively (Shi & Eberhart, 1999). Early works by Kennedy and Eberhart (Kennedy & Eberhart, 1995) laid the foundation for PSO, and subsequent adaptations showcased its ability to navigate the solution space efficiently. In another study (Petrantonakis & Kompatsiaris, 2020), the Dimensionality Reduction based on PSO (PSO-DR) is proposed. PSO minimizes the square root function between the distances of high dimensional data points and those of low dimensional data. Recent work applied Wrapper-Based Binary PSO (WBPSO) for dimensionality reduction in analyzing gene expression data for Alzheimer's disease prediction. This study highlights the potential of PSO in effectively selecting informative features while handling high-dimensional spaces (Salem et al., 2023). In another study, PSO has been employed to optimize embedding spaces in complex datasets, particularly in bioinformatics and image analysis, demonstrating robust adaptability to non-convex optimization problems (Jeremiah et al., 2023). However, PSO shows poor quality results when it deals with high-dimensional data sets (Esmin et al., 2015), and to address this issue, several updated variants were proposed, thereby adapting the parameters of the PSO algorithm, Whether by introducing new velocity vectors (Thangaraj et al., 2012; Liu et al., 2020; Kassoul et al., 2022), new inertia weight (Liu et al., 2020), or by proposing adaptive social and cognitive coefficients (Kassoul et al., 2021; Ghasemi et al., 2019; Fielding & Zhang, 2020).

In our study, we introduce an enhanced version of PSO for optimizing the t-SNE algorithm (t-SNE-PSO). We dynamically adjust the values of the cognitive ( $c_1$ ) and social ( $c_2$ ) coefficients throughout the optimization iterations. The adaptive adjustment of  $c_1$  and  $c_2$  allows our updated PSO algorithm to effectively explore the solution space, exploit promising regions, and converge and achieve high-quality dimensionality reduction results. By leveraging PSO, we aim to overcome the limitations of GD, ensuring more reliable convergence to the global minimum and improving the quality of the low-dimensional representation. In the subsequent sections, we delve into the details of our proposed approach, highlighting the role of PSO as an optimization step for the t-SNE technique.

### 3. Notations and Preliminaries

First, let's define, in Tab. 1, the notations required for the presentation of the proposed model. Then, we provide the preliminaries of t-SNE and PSO.

Notation	Description
<b>t-SNE</b>	
$i$	Index of the data point in a space.
$j$	Index of the neighboring data point in a space.
$D \in \mathbb{N}$	Dimension (high) of original data space.
$d \in \mathbb{N}; d \ll D$	Dimension of reduced data space.
$x_i \in \mathbb{R}^D$	$D$ -dimensional data point.
$X = \{x_1, x_2, \dots, x_n\}$	A set of $n$ input high-dimensional data points.
$\mathcal{N}_i^x$	Data neighborhood of $x_i$ .
$x_j \in \mathcal{N}_i^x$	A neighbor of $x_i$ .
$p_{j i}$	Probability that $x_j \in \mathcal{N}_i^x$ .
$p_{ij}$	Similarity between $x_i$ and its neighbor $x_j$ .
$P = \{p_{ij}\}$	Similarity distribution of $p_{ij}, \forall i \neq j$ .
$\sigma$	Bandwidth parameter for computing data similarities in the original space.
$y_i \in \mathbb{R}^d$	$d$ -dimensional data point.
$Y = \{y_1, y_2, \dots, y_n\}$	A set of $n$ output lower-dimensional data points.
$\mathcal{N}_i^y$	Data neighborhood of $y_i$ .
$y_j \in \mathcal{N}_i^y$	A neighbor of $y_i$ .
$q_{ij}$	Similarity between $y_i$ and its neighbor $y_j$ .
$Q = \{q_{ij}\}$	Similarity distribution of $q_{ij}, \forall i \neq j$ .
$\tau$	Perplexity parameter, used to adjust the number of neighbors to consider when computing similarities.
<b>PSO</b>	
$h$	Positive integer.
$Y_k = \{y_1, \dots, y_n\}$	$k^{th}$ Particle, a set of $n$ lower-dimensional data.
$\mathcal{S} = \{Y_1, Y_2, \dots, Y_m\}$	Swarm of $m$ particles.
$v_k$	Velocity of particle $k$ .
$w$	Inertia weight.
$c_1$	Cognitive coefficient.
$c_2$	Social coefficient.
$r_1, r_2$	Random values associated with the cognitive and social coefficients.
$B_k$	The best position of $Y_k$ in the reduced space.
$G$	The global best position of the swarm (all $Y_k$ ).
$Loss_{B_k}$	The best loss value of the particle $k$ .
$Loss_G$	The global best loss value of the swarm (all $Y_k$ ).
$N_{iter}$	Number of iterations.
$t$	Ietration number.

Table 1: Table of notations.

#### 3.1. Preliminary of t-SNE

The algorithm t-SNE reduces the dimensions of a high-dimensional data  $X$ , to a lower dimensional space  $Y$ , while preserving its underlying structure. The algorithm begins by computing pairwise similarities between data points in the high-dimensional space using the probability distribution  $p_{ij}$  that represents the likelihood of a point  $x_i$  choosing another point  $x_j$  as its neighbor from  $\mathcal{N}_i^x$ . The probability distribution is computed using a Gaussian kernel as follows:

$$p_{i|j} = \frac{\exp(-\|x_i - x_j\|^2)/2\sigma_i^2}{\sum_{k \neq l} \exp(-\|x_k - x_l\|^2)/2\sigma_i^2}, \quad (1)$$

then it symmetries the similarities to obtain the joint probability distribution  $P$  defined as:

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n}. \quad (2)$$

The parameter  $\sigma_i^2$  in Eq. 1 is found using binary search until a pre-defined perplexity  $\tau$  value is reached, in such a way that  $\tau_{(i)} = 2\sum_j p_{j|i} \log_2 p_{j|i}$ . The Student t-distribution with one degree of freedom is used in the second step to compute the similarities in the low-dimensional space. The Student t-distribution has heavier tails than the Gaussian distribution, which allows for better modeling of far-apart distances. Each pair of low-dimensional points  $(y_i, y_j)$  is modeled as a probability distribution  $q_{ij}$  given as follows:

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}} \quad (3)$$

The goal of t-SNE is to find the best low dimensional representation  $Q$ , which includes all low dimensional pairwise probabilities  $q_{ij}$  that faithfully represents  $P$ , which includes all high dimensional pairwise probabilities  $p_{ij}$ . In other terms, the positions of the points  $y_i$  on the map are determined by minimizing the KL divergence of the distribution  $P$  from the distribution  $Q$ . This divergence is non-symmetric and is defined as:

$$KL(P\|Q) = \sum_i KL(P_i\|Q_i) = \sum_i \sum_{j \neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}}. \quad (4)$$

The minimization of the KL divergence with respect to the points  $y_i$  is performed using gradient descent. This optimization process yields a map that effectively captures the similarities between the high-dimensional inputs.

### 3.2. Preliminary of PSO

The PSO algorithm is a population-based optimization technique aimed at identifying optimal solutions by employing a group of particles (Kennedy & Eberhart, 1995; Shi & Eberhart, 1999). Within this method, each particle represents an individual entity, collectively forming a swarm of particles. Notably, PSO offers several appealing attributes, such as straightforward implementation and independence from gradient information. In PSO, the problem's solution space is conceptualized as a search space, with each position representing a potential solution. Through collaborative efforts, particles strive to uncover the optimal position or best solution within this solution space, which is recalculated at each iteration using the following equation:

$$Y_k^{t+1} = Y_k^t + v_k^{t+1}, \quad (5)$$

where the superscript  $t$  designates the iteration at run  $t$ . The movement of each particle is determined by its velocity  $v_k$ , which is defined as follows:

$$v_k^{t+1} = w \cdot v_k^t + c_1^{t+1} r_1 (B_k - Y_k^t) + c_2^{t+1} r_2 (G - Y_k^t). \quad (6)$$

where  $w$  is an inertia weight scaling the previous time step velocity,  $B_k$  does the particle itself find the best position so far,  $G$  is the best position found by the whole swarm so far,  $c_1$  and  $c_2$  are two acceleration coefficients that scale the influence of the best personal position of the particle ( $B_k$ ) and the best global position ( $G$ ). In the standard PSO algorithm, the coefficients  $c_1$  and  $c_2$  typically have fixed values,  $c_1 = c_2 = 2$ .  $r_1$  and  $r_2$  are random variables within the range of 0 and 1. The  $B_k$  and  $G$  values are updated at time  $t$  using the following equations, respectively:

$$B_k(t+1) = \begin{cases} B_k^t & \text{iff } (B_k^t) \leq f(Y_k^{t+1}) \\ Y_k^{t+1} & \text{iff } (B_k^t) > f(Y_k^{t+1}) \end{cases}$$

$$G^{t+1} = \min\{f(B_k), f(G^t)\}$$

where  $k \in \{1, 2, \dots, m\}$

#### 4. t-SNE-PSO: Proposed method

To tackle the localized search limitation of the positions of the data points  $Y_k$  conducted by GD in the original t-SNE, we propose a modified PSO algorithm as an alternative optimization step for t-SNE (t-SNE-PSO). This approach allows for efficient navigation of the solution space, enabling the discovery of low-dimensional representations using the KL divergence defined in Eq. 4 to minimize pairwise similarities in the high-dimensional and low-dimensional spaces. The coordinates of data points in particle  $k$  are updated using the update rule Eq. 5 based on the velocity Eq. 6. Figure 1 represents the overall structure of t-SNE-PSO (Kazemi et al., 2024). The process begins with parameter initialization (pairwise distance computation, particle positions, velocities, and coefficient settings), iterative optimization involving KL divergence evaluation and particle updates, convergence checking, and outputting the low-dimensional embedding according to the global best solution.

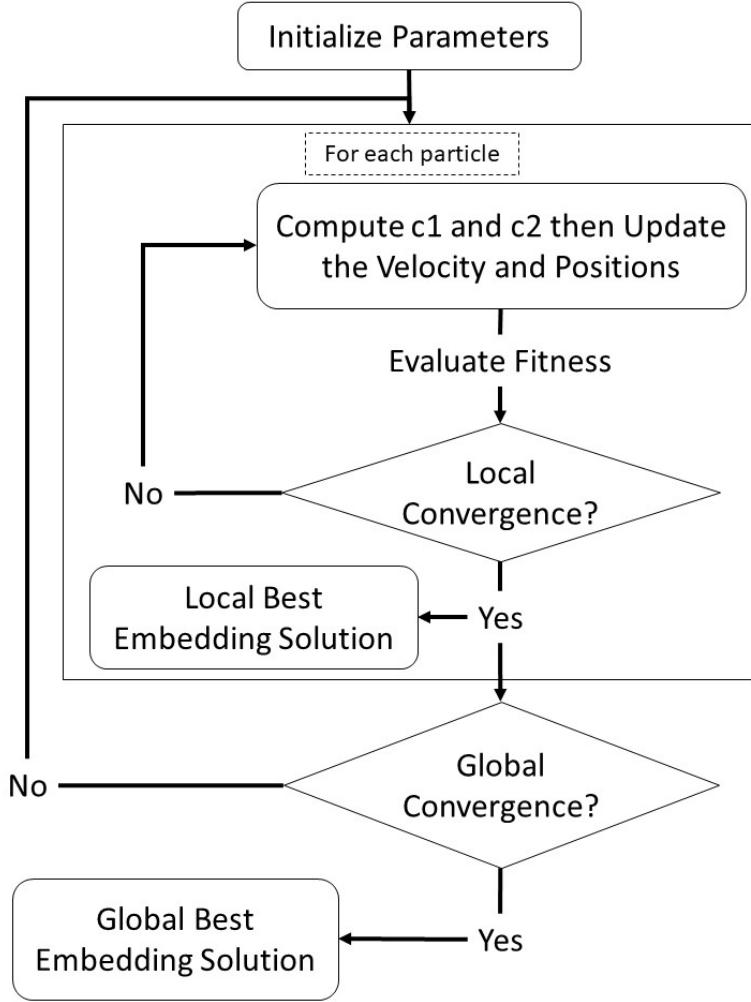


Figure 1: Flowchart illustrating the t-SNE-PSO workflow.

##### 4.1. Algorithmic Steps Details

Algorithm 1 presents the complete steps of the t-SNE-PSO technique. Initially, the algorithm computes pairwise similarities  $p_{ij}$  and sets initial values for particle positions  $Y_k$  and velocities  $v_k$ . For each particle, the loss value is computed using the Kullback-Leibler divergence to find the best loss value  $L_{B_k}$ , and based on this, the particle locates its best personal position  $B_k$ . During the iterations, these local best losses are continually compared to determine the global best loss  $L_G$ , which represents the best solution  $G$  found by the swarm. This iterative

comparison allows the algorithm to identify and converge on the most optimal low-dimensional representation of the data, leveraging all particles' collective experiences and search capabilities. In addition, the cognitive and social coefficients ( $c_1$  and  $c_2$ ) are dynamically adjusted in each iteration to balance exploration and exploitation. Based on these coefficients, particles update their positions and velocities, with  $c_1$  emphasizing individual particle success early on and  $c_2$  promoting collective swarm intelligence over time. This dynamic adjustment allows the particles to navigate the solution space efficiently, balancing the individual learning and shared information to discover optimal low-dimensional representations and avoid local minima.

---

**Algorithm 1:** t-SNE-PSO

---

**Input:**  $X$ ,  $n$ ,  $\tau$ ,  $d$ ,  $m$ , and  $N_{iter}$ .

**Output:**  $\hat{Y}$ .

1. Initialize  $Loss_G$ .
  2. Compute  $p_{ij}, \forall i, j \neq i$  using Eqs. (1) and (2), with  $\tau$ .
  3. For  $k = 1$  to  $m$ :
    - Initialize  $Y_k$  and  $v_k$  of particle  $k$ .
    - Compute the Loss ( $Loss$ ) by Eq. (4).
    - $Loss_{B_k} \leftarrow Loss$
    - $B_k \leftarrow Y_k$ .
    - If  $Loss < Loss_G$ :
      - $Loss_G \leftarrow Loss$ .
      - $G \leftarrow Y_k$ .
  4. For  $t = 1$  to  $N_{iter}$ :
 

Cmpute  $c_1$  and  $c_2$  by Eq. 7 and 8.

For  $k = 1$  to  $m$ :

    - Compute  $v_k^t$  by Eq. 6.
    - Update  $Y_k^t$  by Eq. 5.
    - Compute the Loss ( $Loss$ ) by Eq. 4.
    - If  $Loss < Loss_{B_k}$ :
      - $Loss_{B_k} \leftarrow Loss$ .
      - $B_k \leftarrow Y_k^t$ .
    - If  $Loss < Loss_G$ :
      - $Loss_G \leftarrow Loss$ .
      - $G \leftarrow Y_k^t$ .
- 

#### 4.2. The Dynamic Update of $c_1$ and $c_2$

The cognitive coefficient  $c_1$  and the social coefficient  $c_2$  in PSO play a crucial role in balancing the exploration and exploitation tasks during optimization. The cognitive coefficient,  $c_1$  represents the weight given to the particle's individual best position, emphasizing the historical success of each particle. On the other hand, the social coefficient,  $c_2$  represents the influence of the global best position, encouraging particles to move towards the overall best solution found by the entire swarm. A dynamic adjustment of  $c_1$  and  $c_2$  during optimization serves to fine-tune the balance between exploration and exploitation.

We aim to adjust both the cognitive and the social coefficients according to the complexity and size of the data, which will lead to better performance. First, the optimization process focusing on individual learning allows the particles to exploit promising regions of the search space, facilitating rapid convergence toward potential solutions. Simultaneously, the focus turns more toward the Social coefficient  $c_2$ . This gradual increase places greater importance on the shared information within the swarm, encouraging particles to explore the search space more

extensively. This shift toward social learning potentially discovers global optima and enhances the algorithm's ability to escape local minima. To do so, we propose the following formulas  $c_1$  and  $c_2$ :

$$c_1 = h - \frac{h}{1 + \frac{f}{t}}, \quad (7)$$

and,

$$c_2 = \frac{h}{1 + \frac{f}{t}} \quad (8)$$

The factor  $h$  represents the initial value of the Cognitive coefficient and provides a baseline for the behavior of  $c_1$  and  $c_2$ . It's set to 50 in our experiments. On the other hand,  $f$  and  $t$  control the rate at which  $c_1$  and  $c_2$  evolve over iterations  $t$ . The parameter  $f$  acts as a scaling factor, influencing the speed of the coefficient's decay. A higher value of  $f$  would result in a more gradual decrease in  $c_1$  and an increment in  $c_2$ , providing a longer duration for the algorithm to prioritize exploitation over exploration. The value of  $f$  introduces a parameter for which the social factor and cognitive factor become equal at iteration  $t = f$ , and after that, the social factor will dominate over the cognitive factor. Figure 2 shows how the  $c_1$  and  $c_2$  change over the time iteration for different values of  $f$ .

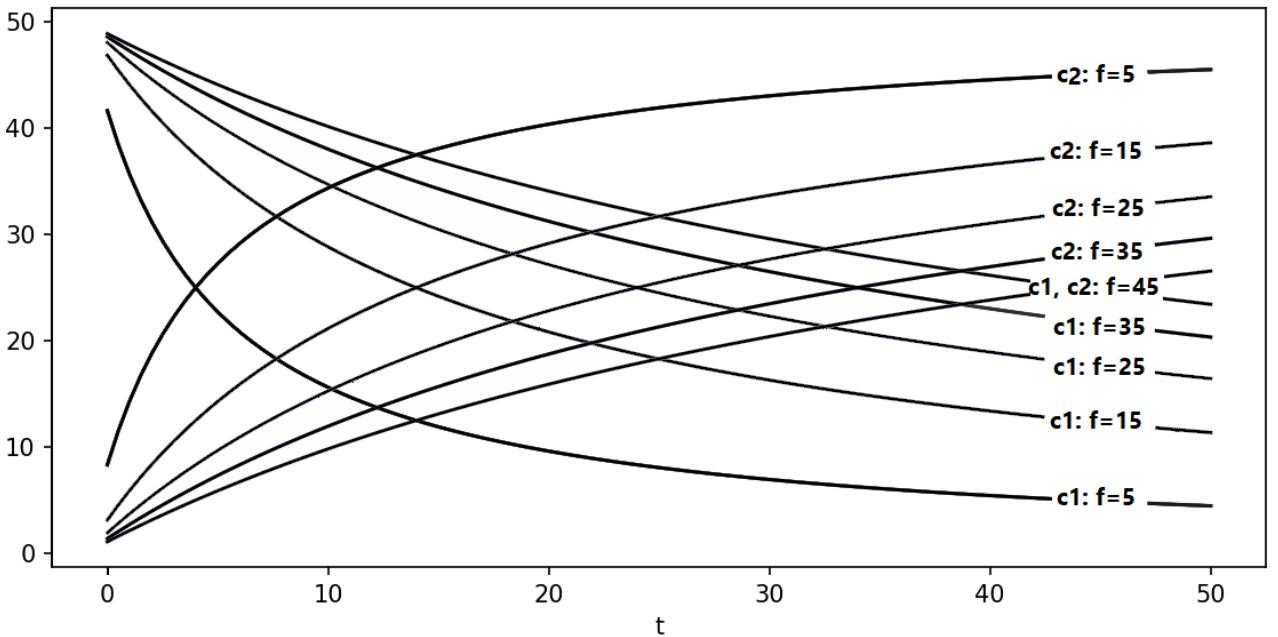


Figure 2: A graph representation of  $c_1$  and  $c_2$  change over the time iteration  $t$ .

#### 4.3. Time Complexity

The time complexity of t-SNE primarily depends on the number of data points  $n$  and the number of iterations  $t$   $O(n^2)$  due to the pairwise distance calculations required for the affinity matrix (Van Der Maaten, 2013). On the other hand, the time complexity of PSO is  $O(mt)$ , where  $m$  is the number of particles, and  $t$  is the number of iterations (Mapetu et al., 2019). In t-SNE-PSO, the dynamic adjustment of  $c_1$  and  $c_2$  are independent of the data points  $n$  and the dimensionality of the search space. The operations required for these adjustments (e.g., subtraction, division) are constant-time operations for each particle, which means the added computational cost is offset by the improved convergence behavior and solution quality of the proposed t-SNE-PSO, which makes it a neglected computational complexity compared

to the overall complexity of the PSO-based optimization process. The overall time complexity of the t-SNE-PSO method is dominated by the PSO iterations, resulting in an approximate complexity of  $O(tmn^2)$ .

## 5. Experiments: Evaluation and Discussion

This section shows the performance of the proposed t-SNE-PSO on a set of benchmark datasets, measured by several evaluation metrics. Specifically, The k-means algorithm is applied to the embedding spaces produced by the different manifold embedding techniques: UMAP, t-SNE, t-SNE-BH, t-SNE optimized using the Genetic Algorithm (t-SNE-GA), t-SNE optimized using the original PSO (t-SNE-OPSO), and our proposed techniques, t-SNE-PSO. We conduct the following experiments: i) In the first experiment, we study the effect of the different parameters of t-SNE-PSO; ii) in the second experiment, we study the changes in the loss function of the proposed algorithm over time iterations; iii) in the third experiment, we compare the performance of t-SNE-PSO against the baselines embedding techniques using the different evaluation metrics; iv) in the last experiment, we provide qualitative results by comparing the visualization of t-SNE-PSO against t-SNE and UMAP. All the experiments were performed on a desktop computer with AMD Ryzen 5 5600x processor and 16 GB RAM. These specifications are provided to offer context for the computational performance results presented. The source code of t-SNE-PSO is available on GitHub: <https://github.com/M-Allaoui/t-SNE-PSO>.

**Datasets.** We conducted experiments on four benchmark datasets: Pendigits, USPS, MNIST-Digit, and Fashion MNIST. Pendigits (Alimoglu et al., 1996) comprises 1797 images representing 10 classes of numbers from 0 to 9, each with a size of 8x8 pixels. The USPS dataset is a digit dataset automatically scanned from envelopes by the U.S. Postal Service containing a total of 9,298 16×16 pixel grayscale samples; the images are centered, normalized, and show a broad range of font styles (Hull, 1994). MNIST consists of 70,000 handwritten digits divided into 10 classes, with each image being a 28x28 grayscale image (LeCun et al., 1998). Fashion-MNIST (Xiao et al., 2017) consists of 70,000 samples distributed across 10 classes, with each sample being a 28x28 image depicting fashion items.

**Evaluation Metrics.** We assess the performance of the K-means algorithm across various embedding spaces obtained from dimension reduction techniques using both external and internal metrics. The external metrics include Accuracy (ACC) (Kuhn, 1955) and Normalized Mutual Information (NMI) (Xu et al., 2003), which yield scores ranging from 0 (indicating poor performance) to 1 (indicating perfect results). Additionally, we utilize two internal validation indices: Davis-Bouldin (DB) (Davies & Bouldin, 1979) and Silhouette coefficient (SC) (Rousseeuw, 1987). The DB index ranges from 0 to infinity, with lower values indicating superior clustering, while the S index ranges from -1 to 1, where values closer to 1 signify better results and -1 indicates poorer results.

### 5.1. The Impact of Parameter Settings.

To analyze the impact of the parameters  $w$ ,  $h$ , and  $f$ , we report the performance values in terms of Accuracy, NMI, SC, and DB while varying  $w$  and  $h$  in the interval [0, 80], and  $f$  in [0, 50] with a step size of 1. For illustration purposes and to get the first insight, we show the plot for the Pendigits dataset (see Fig. 3); and since we assume that all metrics exhibit some correlation, we repeat the experiment for all datasets (see Fig. 4), focusing on NMI, which is well-known in the clustering community.

Figure 4 represents the change effect of the  $w$ ,  $h$  and  $f$  parameters on all datasets in terms of NMI metric. These findings underscore the critical role of meticulous parameter tuning in attaining optimal results with our proposed method. The chosen parameter values

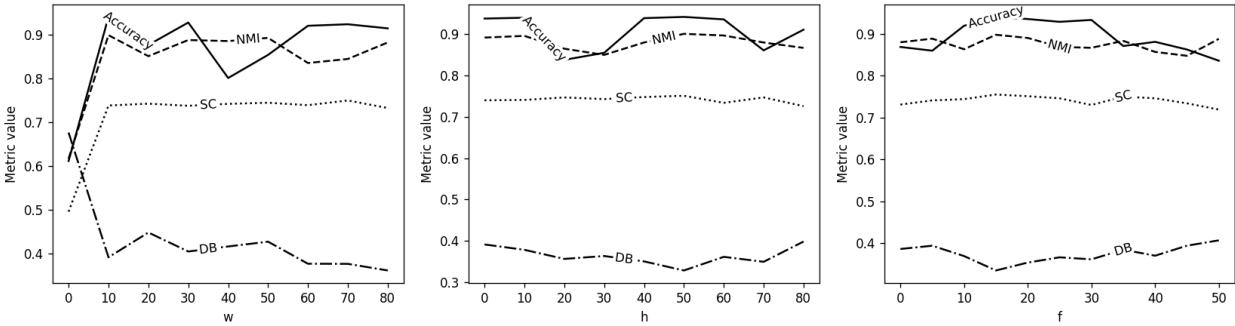


Figure 3: The change effect of  $w$ ,  $h$ , and  $f$  parameters on Pendigits datasets in terms of all evaluation metrics.

signify a carefully crafted configuration that amplifies the convergence and efficiency of the optimization process. As we observe from the three figures, the tuple parameter values ( $w$ ,  $h$ ,  $f$ ) for the datasets Pendigits, FMNIST, USPS, MNIST are: (15, 50, 15), (45, 45, 15), (55, 55, 20), (60, 60, 25) respectively. Since  $w$  for MNIST ( $w = 60$ ) is greater than for the other datasets, this may indicate that the algorithm favors including the historical learned velocity for MNIST and the inverse for Pendigits ( $w = 15$ ). One possible explanation is that the search space for MNIST is more complex, while Pendigits has the least complexity. As  $h$  and  $f$  are used to compute  $c_1$  (cognitive factor) and  $c_2$  (social factor), we suggest analyzing  $c_1$  and  $c_2$  directly gaining deeper insights into the influence of cognitive and social factors. Let us now analyze the scenarios in which  $c_1$  and  $c_2$  yield the best performance for each dataset. Theoretically, it is straightforward to demonstrate, from Eq. (7) and (8), that  $c_1$  and  $c_2$  become equal (i.e., the cognitive and social factors are balanced) when  $t = f$ . We aim to investigate whether  $c_1$  or  $c_2$  should be prioritized to achieve optimal performance for each dataset. Figure 5 illustrates the variation of  $c_1$  and  $c_2$  with respect to the number of iterations. The vertical dashed lines indicate the iteration where  $c_1$  and  $c_2$  become equal, signifying identical cognitive and social factors. From the graph, it can be observed that for the MNIST dataset, the algorithm prioritizes the cognitive factor over the social factor for a longer duration ( $\sim 24$  iterations), while for Pendigits and FMNIST, it shifts towards the social factor more quickly ( $\sim 14$  iterations). A possible explanation for this observation could lie in the complexity and characteristics of the search spaces for the datasets. For MNIST, the search space may be more intricate, requiring a longer exploration phase to identify promising regions. This justifies why the algorithm prioritizes the cognitive factor, which emphasizes individual particle exploration, for a longer duration. In contrast, for Pendigits and FMNIST, the search spaces might be less complex or better suited to benefit from collaboration among particles. Thus, the algorithm transitions to prioritizing the social factor, which emphasizes group convergence, more quickly. This can support the observation that the value of  $w$  for the MNIST dataset is larger than that of Pendigits and FMNIST. This adaptive behavior suggests the algorithm is tailoring its search dynamics to the structure of the optimization landscape presented by each dataset.

As guidelines to choose the values of three parameters for other datasets, we propose the following instructions:

- $w$  controls how much of the particle's previous velocity influences its current velocity, which means the larger the value of  $w$  the more importance is given to the historical learned velocity.
- $h$  parameter scales the higher value that  $c_1$  and  $c_2$  could have, which represents the importance provided to the exploitation and exploration.
- $f$  parameter scales the balance between  $c_1$  and  $c_2$ , where a higher value of  $f$  provides a longer duration for the algorithm to prioritize the cognitive over the social factor and vice versa.

Overall, understanding the importance of the parameter selection while considering the dataset's complexity and size leads to superior performance.

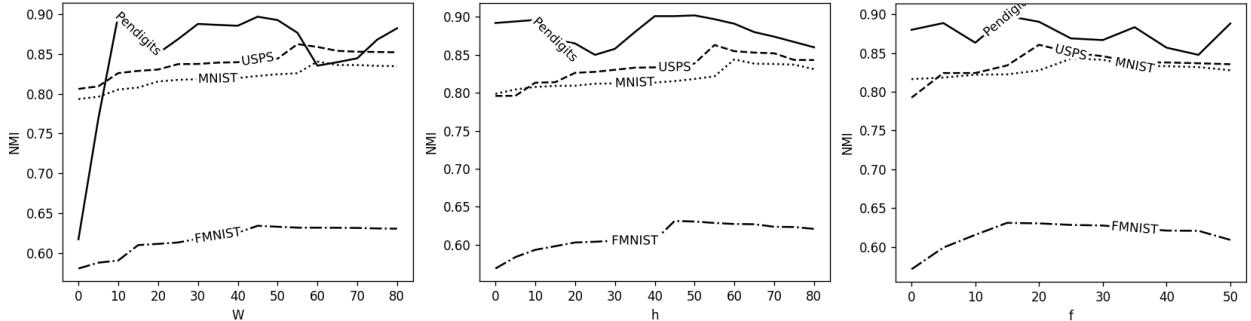


Figure 4: The change effect of  $w$ ,  $h$ , and  $f$  parameters on all datasets in term of NMI.

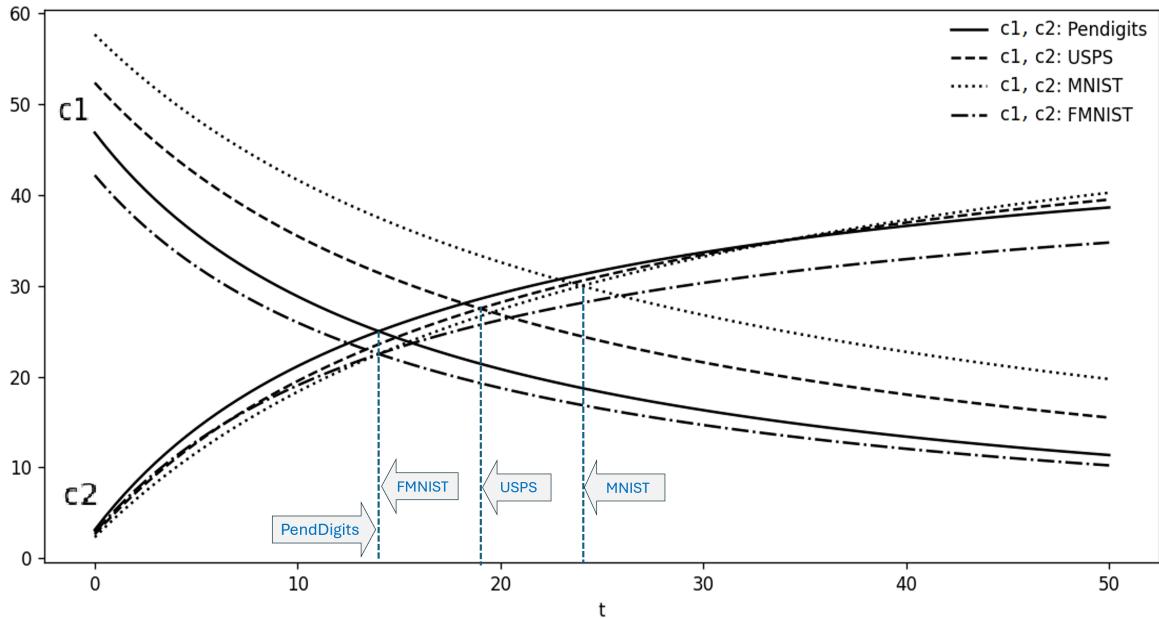


Figure 5: The change of  $c_1$  and  $c_2$  over time step  $t$  using the optimal tuples  $(w, h, f)$  identified above for each dataset.

## 5.2. Convergence Analysis

This experiment studies the changes in the KL divergence loss function of the proposed algorithm over time iterations against the changes in the loss function of the original t-SNE performed on the four datasets, Pendigits, USPS, MNIST, and FMNIST. Figure 6 illustrates the algorithm's ability to converge towards a solution that minimizes the KL divergence loss. The observed decrease in the loss function throughout iterations in the proposed algorithm signifies the effectiveness of our optimization approach. This decreasing trend indicates the optimization process successfully navigating the solution space, iteratively refining the embedding to capture the underlying structures of the data better. The consistent reduction in the loss function attests to the algorithm's capacity to optimize the coordinates of data points in the low-dimensional space, aligning them with the desired objectives. This experiment's results reinforce the robustness and efficacy of our proposed method, showcasing its capability to improve and refine the embedding iteratively, ultimately leading to a solution that optimally represents the complex relationships within the data.

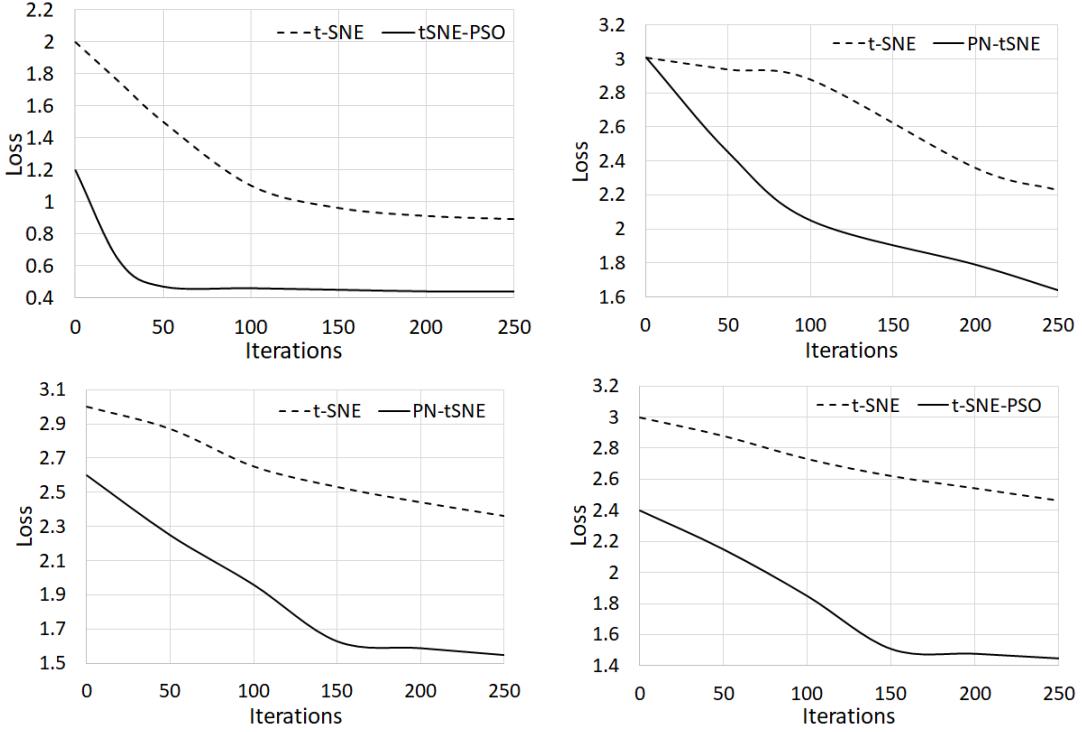


Figure 6: A graph representation of loss error over the time iteration  $t$ , t-SNE-PSO against t-SNE. The datasets from left to right and top to bottom are: Pendigits, USPS, MNIST, FMNIST.

To compare the convergence of t-SNE and t-SNE-PSO based on their loss values, running each model 50 times and computing the loss values for each run provides a dataset for statistical analysis. The goal is to determine if there is a significant difference in the performance of the two models, specifically whether one model converges to a better minimum than the other. Therefore, using these loss values, we generated Kernel Density Estimation (KDE) plots to visualize the distribution of loss values for both algorithms as depicted in Fig. 7. From the figure, t-SNE has wider (fat) KDE plots in most datasets compared to t-SNE-PSO, which could mean that it has difficulty finding a consistent solution or that the optimization is stuck in several local minima. It can also indicate that the loss surface is highly irregular, with many possible solutions that are equally valid. This could be due to the high dimensionality of the data or the initialization of t-SNE causing variability in the results. Additionally, it might suggest that t-SNE is sensitive to the choice of initial embedding, which can affect the outcome of the optimization. However, a less fat distribution of losses typically means the optimization process has a clear and predictable outcome, and t-SNE-PSO is likely to find a stable minimum for the KL Divergence compared to t-SNE during the optimization and may make the landscape of the loss function smooth and not prone to many local minima.

Subsequently, we computed the  $t$ -test and corresponding  $p$ -value (Witte & Witte, 2017) for the loss values of each model to assess their differences. The two statistics are used to test the null hypothesis ( $H_0$ ), stating that there is no effect or no difference between the two models. The  $T$ -Test quantifies the difference between sample means relative to variability, and the  $p$ -value measures the strength of evidence against the null hypothesis. Table 2 reports the results based on all datasets. These high  $T$ -Test values indicate a significant difference in the loss distributions between the two algorithms across all datasets. In addition, if we consider a significance level  $\alpha = 0.05$ , and since  $p\text{-value} \leq \alpha$  for all datasets, this rejects  $H_0$  and concludes that there is a statistically significant difference between the two models t-SNE and t-SNE-PSO. Therefore, the results reinforce the conclusion that t-SNE-PSO outperforms t-SNE in terms of optimization consistency and quality, as its loss values are not only lower but also

exhibit significantly less variability, as observed in the KDE plots.

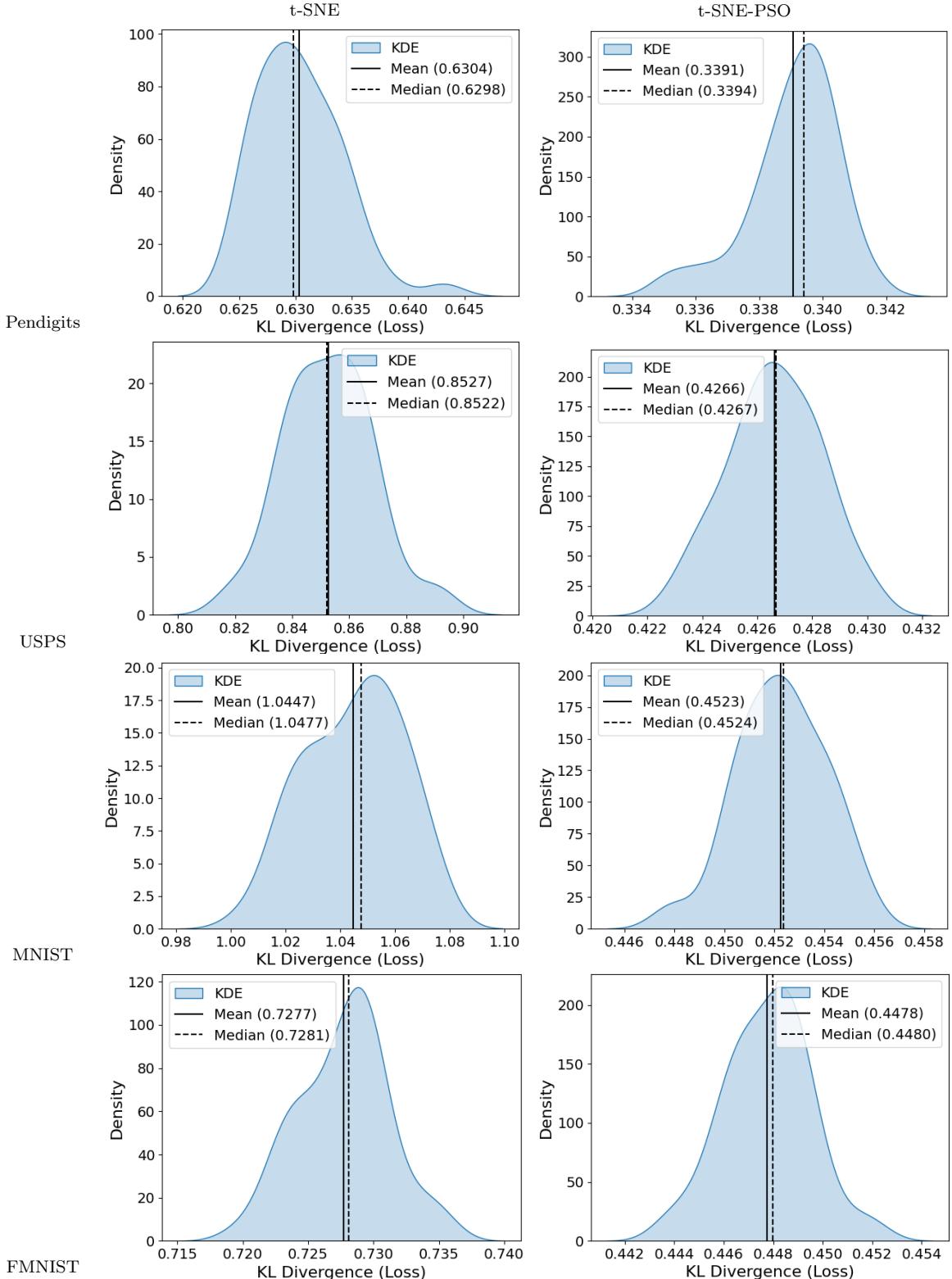


Figure 7: Kernel Density Estimation (KDE) plots illustrating the distribution of loss values for t-SNE and t-SNE-PSO, based on 50 independent runs each.

### 5.3. Computational Complexity Analysis

This section addresses two aspects: First, we compared the execution time of t-SNE and t-SNE-PSO by increasing the dataset size, using the PenDigit dataset. The second aspect assesses the execution time of t-SNE-PSO against all the baseline techniques on all datasets.

Table 2: t-SNE-PSO vs. t-SNE using the T-Test measures: checking the difference between the losses' distribution.

Dataset	Algorithm	Mean	STD	Median	T-Test	p-value
Pendigits	t-SNE	0.6205	0.0044	0.6202	425.7669	0.0000
	t-SNE-PSO	0.3391	0.0014	0.3394		
USPS	t-SNE	0.8527	0.0155	0.8522	189.4529	0.0000
	t-SNE-PSO	0.4266	0.0017	0.4267		
MNIST	t-SNE	1.0447	0.0176	1.0477	231.5963	0.0000
	t-SNE-PSO	0.4523	0.0018	0.4524		
FMNIST	t-SNE	0.7277	0.0034	0.7281	510.6772	0.0000
	t-SNE-PSO	0.4478	0.0017	0.4480		

The results, depicted in Fig. 8, indicate that while both methods experience an increase in computational time with larger datasets, t-SNE-PSO shows a more gradual growth in processing time compared to standard t-SNE. However, even though in theory t-SNE-PSO is more complex than t-SNE (that is,  $O(tmn^2)$  vs  $O(n^2)$ ), it takes less time to converge in practice because it requires fewer iterations. This reduction in iteration count helps to offset the higher per-iteration computational cost. Consequently, t-SNE-PSO demonstrates a more controlled and scalable computational demand. This characteristic makes t-SNE-PSO a viable alternative for visualizing larger datasets, balancing the complexity between the dimensionality reduction task and the optimization process.

In addition, to assess the execution time of the different dimensionality reduction methods, Tab. 3 is provided to summarize the results in seconds. As observed, t-SNE-PSO achieves the fastest execution time among the compared methods, requiring fewer iterations to converge due to its effective optimization process. The dynamic particle reduction strategy, where fewer particles are maintained in later iterations, significantly reduces computational overhead while preserving solution quality. While t-SNE-PSO has a computational complexity of  $O(tmn^2)$ , higher than t-SNE  $O(n^2)$ , t-SNE-BH  $O(n\log n)$ , and UMAP  $O(n^{1.14})$ , the reduced iteration count and particle adjustments ensure improved efficiency. t-SNE-PSO offers a balanced trade-off between execution time and embedding quality, producing more accurate and well-separated clusters.

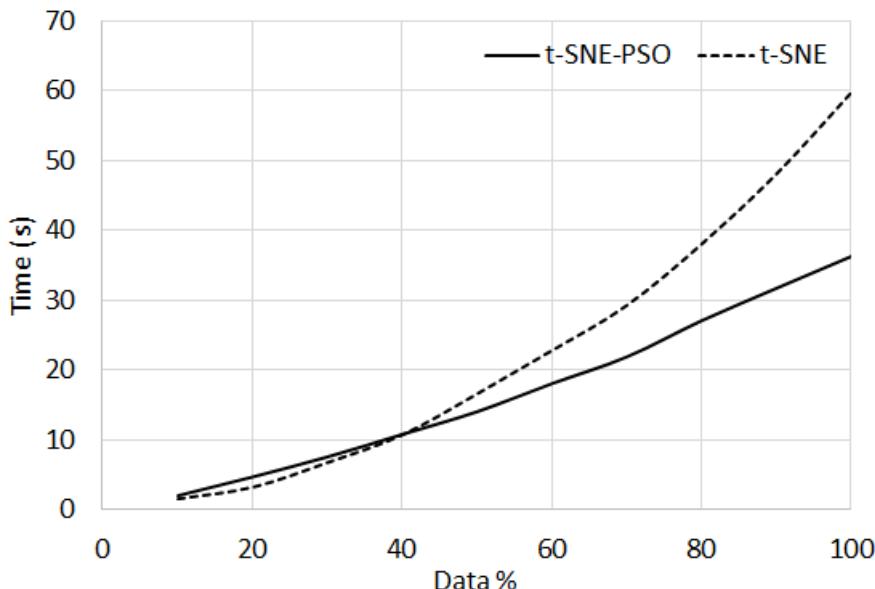


Figure 8: A graph representation of time execution wrt data size of both algorithm t-SNE-PSO and t-SNE on Pendigit dataset.

Table 3: t-SNE-PSO vs. other baselines using the execution time in seconds.

Algorithm	Dataset			
	Pendigits	USPS	MNIST	FMNIST
UMAP	228.62±3.73	1056.89±51.26	1073.22±58.79	1102.34±42.76
t-SNE	59.29±0.04	1418.34±45.61	2585.53±65.32	2769.83±83.42
t-SNE-BH	3.35±0.02	54.07±3.58	115.94±0.48	126.73±0.25
t-SNE-GA	74.56±1.57	2124.12±233.26	2815.76±5.58	2740.94±54.50
t-SNE-OPSO	39.70±1.47	2059.65±70.98	2721.83±127.95	2549±67.24
<b>t-SNE-PSO</b>	<b>3.24±0.12</b>	<b>48.42±4.25</b>	<b>102.45±5.63</b>	<b>115.87±2.75</b>

#### 5.4. Clustering Quality

This section presents a comparative analysis of our proposed technique, t-SNE-PSO, against prominent dimensionality reduction methods, UMAP, t-SNE, and Barnes-Hut (t-SNE-BH), as well as optimization techniques including t-SNE optimized using the Genetic Algorithm (t-SNE-GA) and t-SNE optimized using the original PSO (t-SNE-OPSO). We utilized K-means clustering on the embedding spaces generated by these techniques and evaluated their performance using the four metrics mentioned above: Accuracy (ACC), Normalized Mutual Information (NMI), Davis-Bouldin Index (DBI), and Silhouette Coefficient (SC). The goal was to assess the relative effectiveness of each method and showcase the superiority of t-SNE-PSO.

**External Evaluation Metrics.** The ACC metric measures the percentage of correctly clustered data points relative to the ground truth labels, with higher values indicating better performance. The NMI metric quantifies the mutual dependence between the clustering results and the ground truth labels, ranging from 0 (no mutual information) to 1 (perfect correlation). High ACC and NMI scores signify that the embedding space accurately reflects the underlying data structure. Using these external evaluation metrics, t-SNE-PSO demonstrated significantly higher ACC and NMI scores than UMAP, t-SNE, t-SNE-BH, t-SNE-GA, and t-SNE-OPSO (see Tab. 4). Our proposed method better maintained the underlying structures and relationships within the data, resulting in superior clustering. For instance, on the MNIST dataset, t-SNE-PSO achieved an ACC of 0.84 and an NMI of 0.84, compared to the next best performance by t-SNE-GA and UMAP, respectively, which had an ACC of 0.82 and an NMI of 0.83.

**Internal Evaluation Metrics.** The DBI metric assesses cluster compactness and separation, with lower values indicating more compact and well-separated clusters. The SC metric evaluates how similar an object is to its cluster compared to other clusters, ranging from -1 to 1, where higher values indicate better-defined clusters. Regarding these internal evaluation metrics, t-SNE-PSO achieved lower DBI scores, indicating more compact and well-separated clusters. This highlights the effectiveness of our method in addressing the crowding issue commonly encountered with t-SNE and t-SNE-BH and surpassing UMAP, t-SNE-GA, and t-SNE-OPSO in terms of clustering quality. Additionally, the SC results further underscored the superiority of t-SNE-PSO in achieving well-defined and distinct clusters. For example, on the Fashion-MNIST dataset, t-SNE-PSO achieved a DBI of 0.61 and an SC of 0.57, outperforming UMAP, that have the second best scores, which had a DBI of 0.62 and an SC of 0.56.

**Performance Comparison: t-SNE-PSO vs. t-SNE-OPSO and t-SNE-GA.** Comparing t-SNE-PSO against t-SNE-OPSO and t-SNE-GA, our proposed technique demonstrates superior performance across various datasets and evaluation metrics. Firstly, t-SNE-PSO consistently outperforms t-SNE-OPSO and t-SNE-GA in terms of accuracy (Acc) and normalized mutual information (NMI). For instance, on the USPS dataset, t-SNE-PSO achieves an accuracy of 0.83 and an NMI of 0.86, compared to 0.80 and 0.82 for t-SNE-OPSO, and 0.80 and 0.83 for t-SNE-GA, respectively. Secondly, the SC and DB scores for t-SNE-PSO are

also higher across most datasets, reflecting better-defined, compact, and more distinct clusters. On the Pendigits dataset, t-SNE-PSO achieves an SC of 0.76 and a DB score of 0.36, outperforming t-SNE-OPSO with scores of 0.56 and 0.62 and t-SNE-GA with scores of 0.57 and 0.60 for SC and DB. These scores signify that our method reduces intra-cluster variance while increasing inter-cluster distance, leading to superior clustering performance. The dynamic adjustment of the cognitive and social coefficients in PSO plays a crucial role in this enhanced performance. By initially prioritizing individual experiences (cognitive coefficient) and gradually shifting towards collective learning (social coefficient), t-SNE-PSO effectively balances exploration and exploitation. This dynamic strategy enables the particles to explore the search space thoroughly while converging to optimal solutions, thereby overcoming the limitations of t-SNE-OPSO and t-SNE-GA in escaping local minima. Thorough analysis across these evaluation metrics illustrates the superiority of t-SNE-PSO in manifold embedding tasks over UMAP, t-SNE, t-SNE-BH, t-SNE-GA, and t-SNE-OPSO. The strategic incorporation of Particle Swarm Optimization (PSO) with the dynamic adjustment into the t-SNE framework enhances convergence, addresses local minima issues, and offers a more effective optimization landscape. These advantages collectively contribute to the exceptional performance of t-SNE-PSO, making it a compelling choice for dimensionality reduction tasks where preserving both local and global structures is paramount.

Table 4: t-SNE-PSO vs. other baselines using the evaluation metrics: Accuracy, NMI, DB, SC scores.

Dataset	Algorithm	Acc	NMI	SC	DB
Pendigits	UMAP	0.91 ± 0.03	0.88 ± 0.02	0.74 ± 0.01	0.38 ± 0.02
	t-SNE	0.93 ± 0.01	0.88 ± 0.01	0.58 ± 0.01	0.60 ± 0.01
	t-SNE-BH	0.93 ± 0.00	0.89 ± 0.00	0.58 ± 0.01	0.59 ± 0.01
	t-SNE-GA	0.87 ± 0.03	0.86 ± 0.02	0.57 ± 0.00	0.60 ± 0.02
	t-SNE-OPSO	0.92 ± 0.01	0.87 ± 0.01	0.56 ± 0.01	0.62 ± 0.01
	t-SNE-PSO	<b>0.94 ± 0.01</b>	<b>0.90 ± 0.00</b>	<b>0.76 ± 0.01</b>	<b>0.36 ± 0.01</b>
USPS	UMAP	0.81 ± 0.00	0.85 ± 0.00	0.68 ± 0.00	0.46 ± 0.01
	t-SNE	0.80 ± 0.02	0.82 ± 0.00	0.45 ± 0.01	0.75 ± 0.02
	t-SNE-BH	0.82 ± 0.02	0.83 ± 0.01	0.46 ± 0.00	0.75 ± 0.01
	t-SNE-GA	0.80 ± 0.00	0.83 ± 0.00	0.46 ± 0.00	0.77 ± 0.01
	t-SNE-OPSO	0.80 ± 0.01	0.82 ± 0.00	0.46 ± 0.00	0.76 ± 0.01
	t-SNE-PSO	<b>0.83 ± 0.01</b>	<b>0.86 ± 0.00</b>	<b>0.69 ± 0.00</b>	<b>0.43 ± 0.01</b>
MNIST	UMAP	0.80 ± 0.00	0.83 ± 0.00	0.65 ± 0.00	0.49 ± 0.01
	t-SNE	0.81 ± 0.04	0.77 ± 0.02	0.47 ± 0.01	0.71 ± 0.03
	t-SNE-BH	0.81 ± 0.03	0.77 ± 0.01	0.47 ± 0.00	0.72 ± 0.01
	t-SNE-GA	0.82 ± 0.00	0.78 ± 0.01	0.47 ± 0.00	0.73 ± 0.00
	t-SNE-OPSO	0.82 ± 0.01	0.78 ± 0.01	0.48 ± 0.01	0.72 ± 0.01
	t-SNE-PSO	<b>0.84 ± 0.01</b>	<b>0.84 ± 0.01</b>	<b>0.67 ± 0.01</b>	<b>0.47 ± 0.01</b>
FMNIST	UMAP	0.57 ± 0.01	0.62 ± 0.03	0.56 ± 0.01	0.62 ± 0.01
	t-SNE	0.58 ± 0.01	0.59 ± 0.01	0.43 ± 0.01	0.77 ± 0.02
	t-SNE-BH	0.59 ± 0.03	0.59 ± 0.01	0.42 ± 0.00	0.78 ± 0.03
	t-SNE-GA	0.55 ± 0.02	0.57 ± 0.01	0.42 ± 0.00	0.78 ± 0.01
	t-SNE-OPSO	0.57 ± 0.01	0.58 ± 0.00	0.41 ± 0.00	0.78 ± 0.01
	t-SNE-PSO	<b>0.60 ± 0.02</b>	<b>0.63 ± 0.00</b>	<b>0.57 ± 0.01</b>	<b>0.61 ± 0.00</b>

### 5.5. Qualitative results

In this comprehensive experiment, we investigated the performance of our proposed algorithm against t-SNE and UMAP under varying perplexity and  $n\_neighbors$  values. The *perplexity* ( $\tau$ ) parameter for our method and t-SNE, along with the *n\_neighbors* parameter for UMAP, was systematically adjusted within the range of [5, 50]. Figure 9 represents the visualization of the three techniques applied to the Pendigit dataset while varying perplexity and *n\_neighbors* values. The visualization is colored according to the ground truth label, where each color represents one particular class.

We observe through Fig. 9 that the larger the value of  $\tau$ , the more this leads to overlapping of the clusters for t-SNE and UMAP, while for t-SNE-PSO it leads to a well-separation of the clusters. UMAP tends to produce more tightly packed clusters, which can sometimes lead to overlap and indistinct boundaries between different clusters. This crowding can obscure the true relationships and structures within the data. In contrast, the t-SNE visualization provides better separation between clusters compared to UMAP. However, t-SNE still encounters some crowding issues, particularly in dense regions, resulting in less clearly defined cluster boundaries. The t-SNE-PSO visualization demonstrates superior separation between clusters, with distinct and well-defined boundaries. The optimization process in t-SNE-PSO effectively addresses the crowding problem, allowing for a more accurate representation of the underlying data structure.

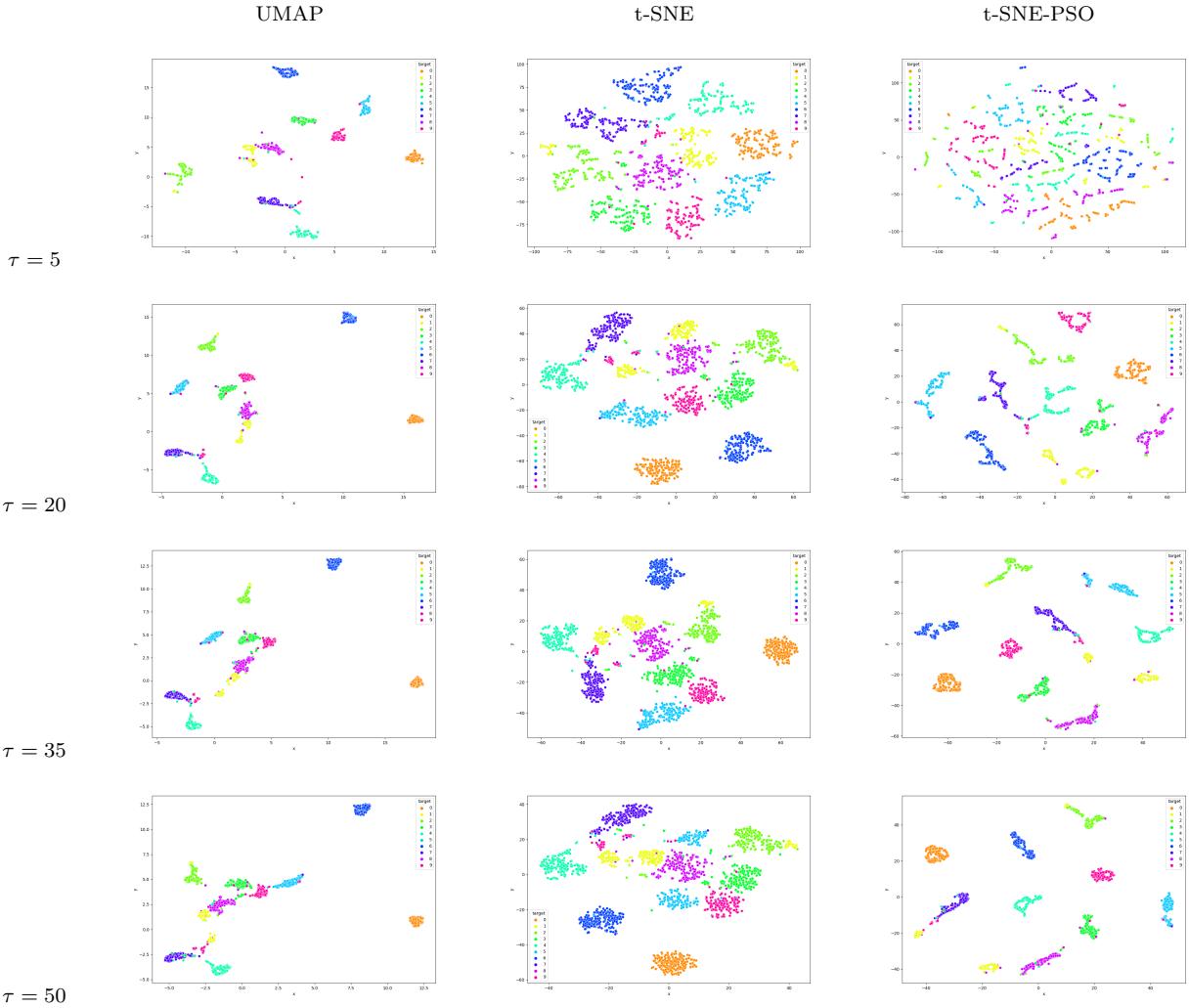


Figure 9: The visualization of the three techniques, UMAP, t-SNE, and t-SNE-PSO, of Pendigits varying the number of neighbors in UMAP and the perplexity in t-SNE and t-SNE-PSO.

Furthermore, Fig. 10, 11, 12, and 13 represent a 2D visualization of Pendigits, USPS, MNIST, and Fashion MNIST datasets respectively using all considered techniques. This experiment underscores our proposed algorithm’s versatility and effectiveness compared to existing state-of-the-art methods, solidifying its position as a powerful tool for manifold embedding and dimensionality reduction across diverse datasets and scenarios.

## 6. Advantages, Limitations, and Applications of t-SNE-PSO

In this section, we present a thorough evaluation of the advantages and disadvantages of our proposed t-SNE-PSO method. Understanding both the strengths and limitations of t-SNE-PSO

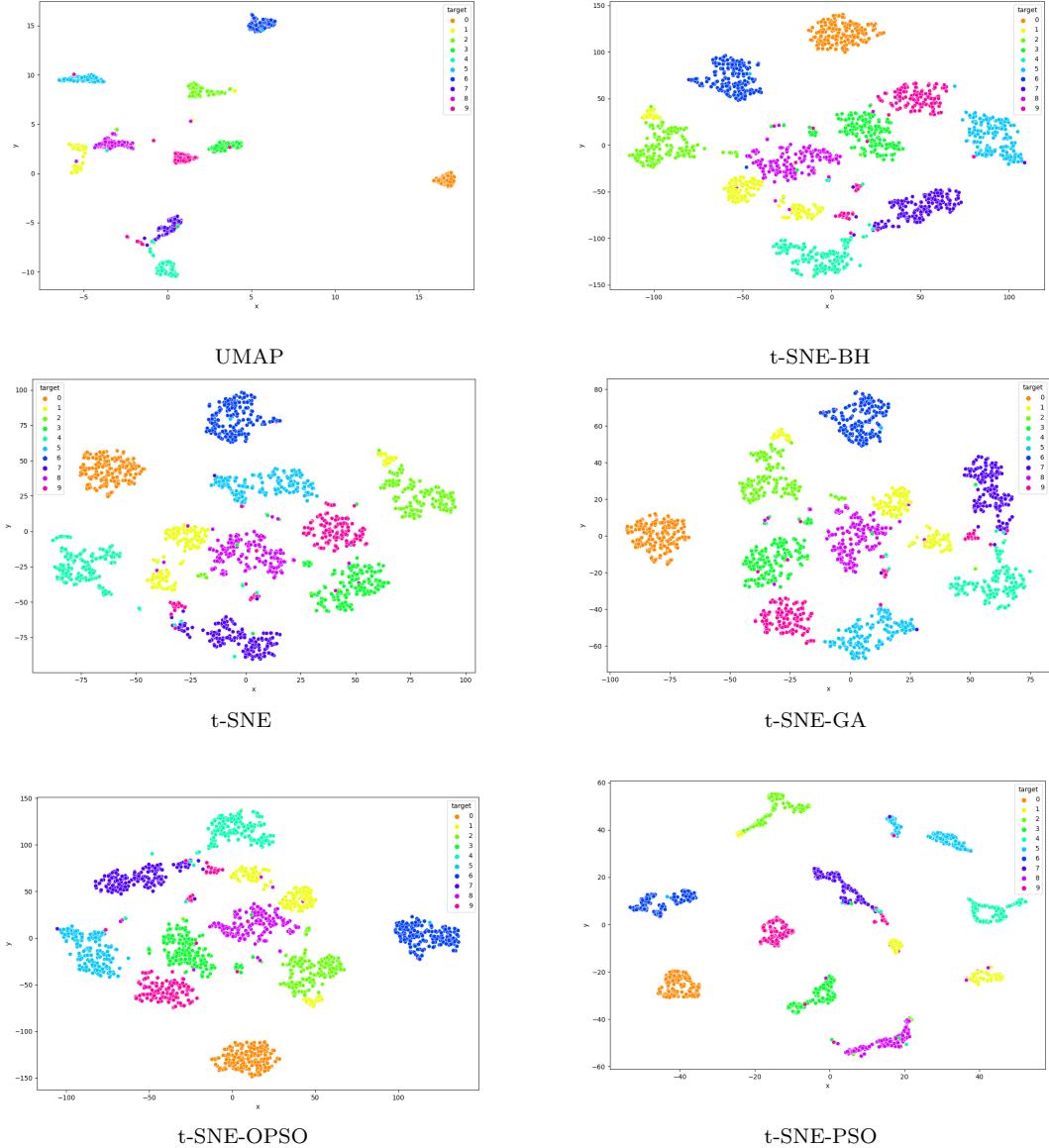


Figure 10: The visualization of all techniques of the Pendigits dataset.

is crucial for assessing its applicability to various dimensionality reduction and clustering tasks. We also highlight specific scenarios where t-SNE-PSO demonstrates superior performance, providing insights into its practical utility and potential challenges.

### 6.1. Advantages

- **Superior Performance on Evaluation Metrics:** t-SNE-PSO consistently outperformed UMAP and traditional t-SNE across all evaluation metrics, including ACC, NMI, DBI, and SC. This indicates its effectiveness in addressing the crowding issue commonly encountered with t-SNE and maintaining the dimensionality reduction, clustering, and visualization tasks.
- **Improved Optimization Landscape for Non-Convex Challenges:** The integration of PSO into the t-SNE framework enhances convergence and addresses local minima issues, providing a more effective optimization landscape. While gradient descent provides more intuitive and interpretable optimization processes, PSO aligns better with our goals due to its ability to excel in non-convex, multimodal optimization tasks, where gradient methods often risk converging to suboptimal solutions. Additionally, PSO's

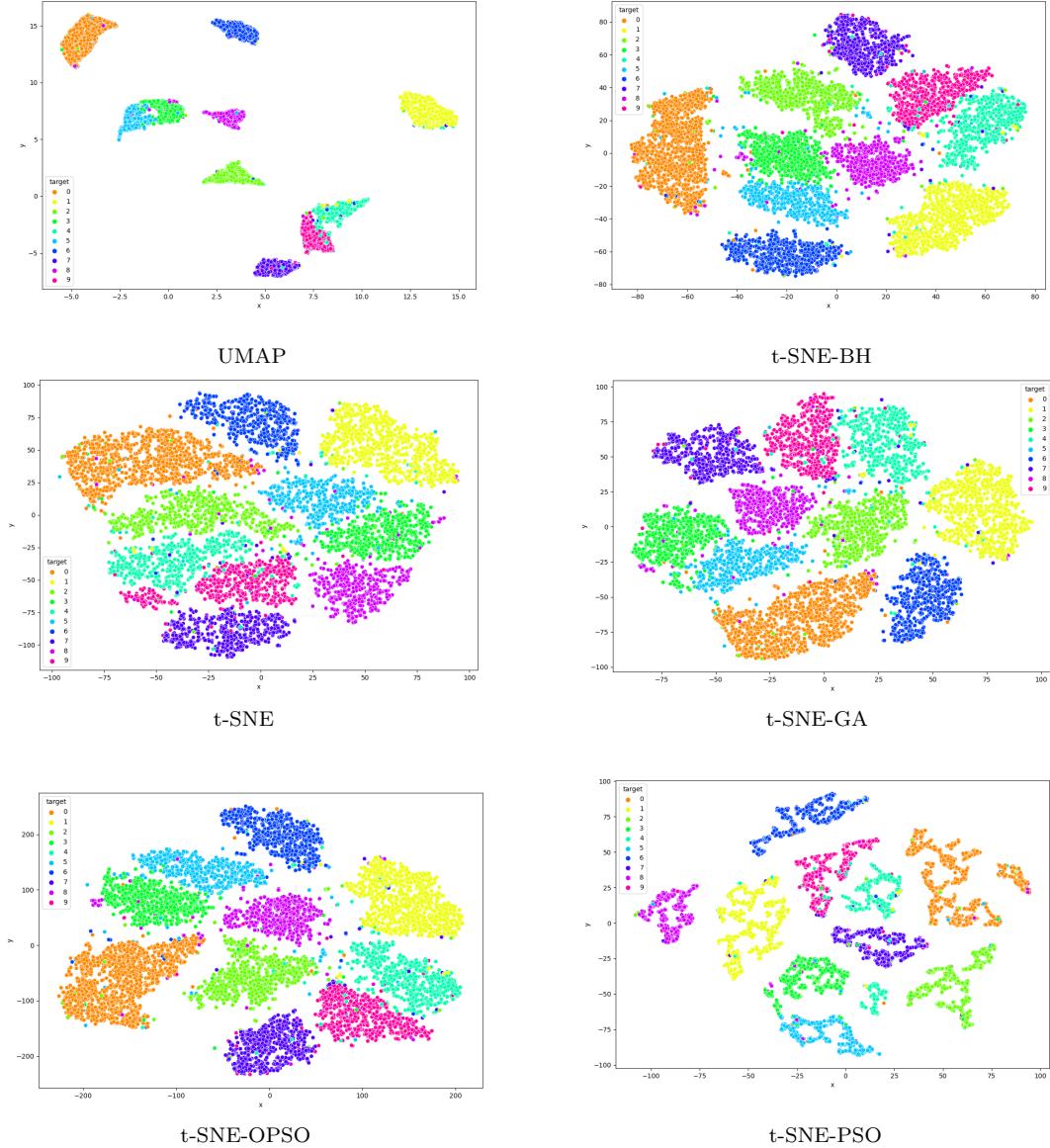


Figure 11: The visualization of all techniques of the USPS dataset.

swarm dynamics offer alternative insights into the search process and fitness landscape, complementing the interpretability of gradient-based methods. Its capacity to optimize non-differentiable objectives further broadens its applicability, making it a robust choice for addressing the complexities of dimensionality reduction.

## 6.2. Limitations

- **Computational Complexity:** The theoretical computational complexity shows that t-SNE-PSO has a complexity higher than t-SNE. In addition, the practical results show that the execution time of t-SNE-PSO is higher than t-SNE-BH and UMAP, which can be a drawback for larger datasets.
- **Parameter Sensitivity:** The effectiveness of t-SNE-PSO may depend on the proper tuning of t-SNE and PSO hyper-parameters, which could require additional experimentation and expertise.

## 6.3. Applications of t-SNE-PSO

- **Noisy and large-scale Datasets with Complex Structures:** t-SNE-PSO is particularly effective for noisy and high-dimensional data where preserving both local and global

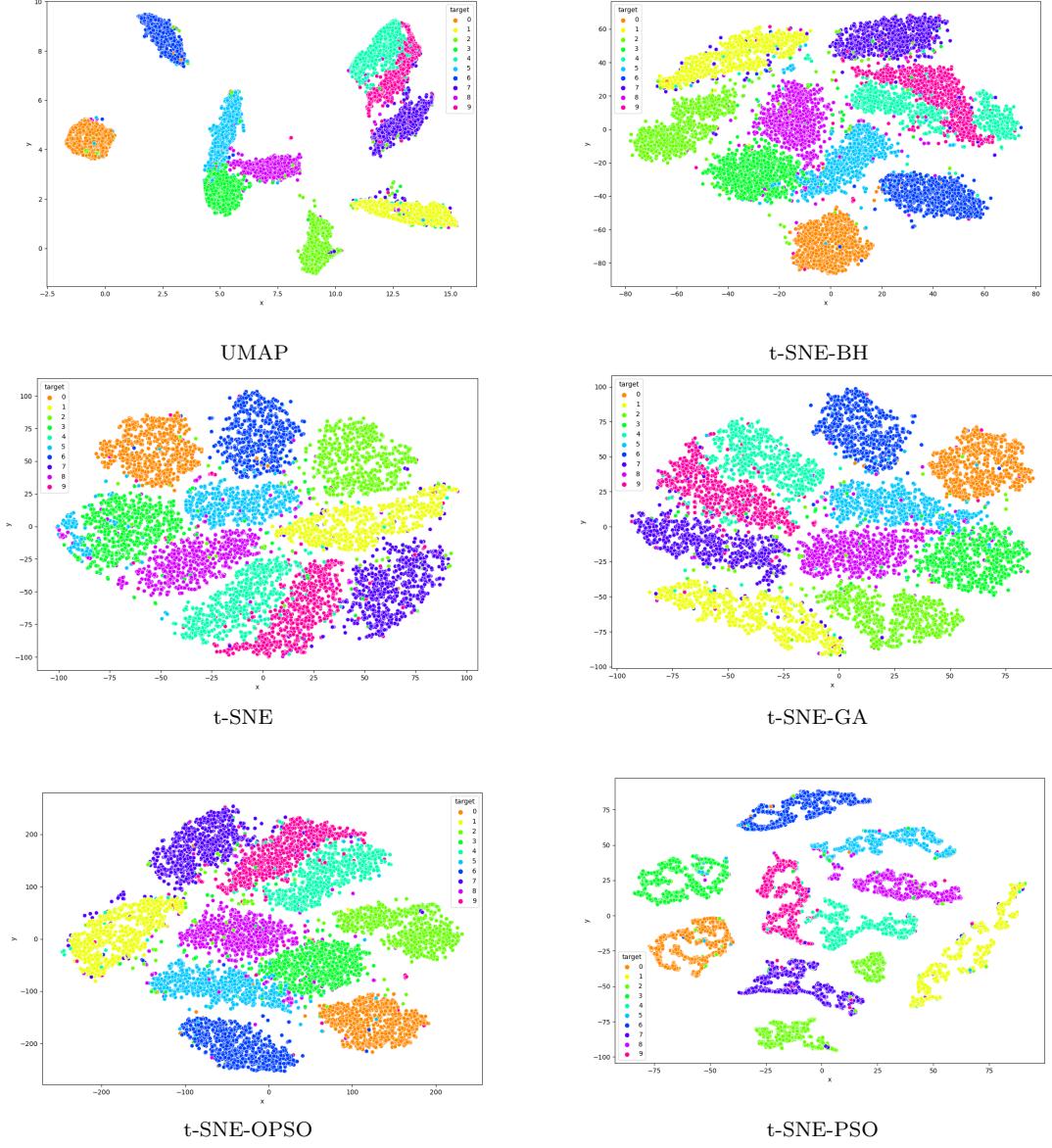


Figure 12: The visualization of all techniques of the MNIST dataset.

structures is crucial.

- **Clustering and Classification Applications:** The method excels in applications requiring well-defined and distinct clusters and classes, as demonstrated in Sec. 5.4 for clustering or in Appendix .1 for classification.
- **Out-of-sample extension:** While t-SNE has proven highly effective for visualizing high-dimensional data, its inherent lack of a direct out-of-sample extension remains a limitation (Gisbrecht et al., 2012; Crecchi et al., 2020). Traditional t-SNE must recompute the entire embedding to incorporate new data points, which is computationally intensive and impractical for dynamic or streaming datasets. To address this approaches such as neural network-based mappings have been explored, offering a way to approximate the t-SNE transformation for unseen data. Aligning with this concept, we evaluate and compare the performance of t-SNE and t-SNE-PSO, considering their handling of this critical aspect. For more detail about this, readers are referred to Appendix .2

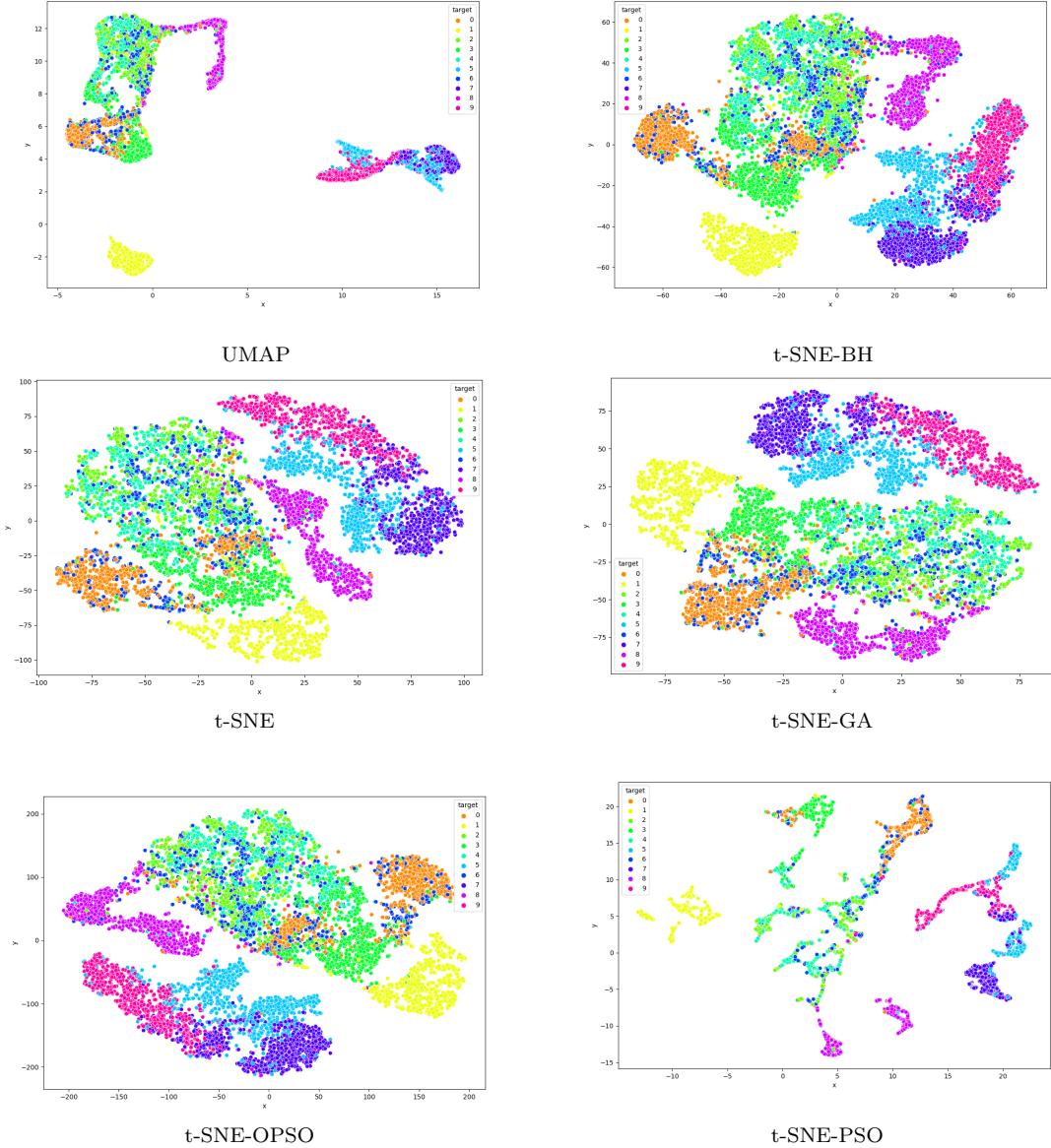


Figure 13: The visualization of all techniques of the Fashion MNIST dataset.

## 7. Conclusion

This paper introduced t-SNE-PSO, a novel approach for manifold embedding and dimensionality reduction. By integrating PSO into the t-SNE framework, our proposed method overcomes critical limitations associated with GD optimization techniques, providing improved convergence and effectiveness.

- Through extensive experimental analysis, we methodically investigated the influence of parameters such as the Inertia Weight, the cognitive, and the social coefficients on the performance optimization of t-SNE-PSO. This precise adjustment reflects a delicate equilibrium between exploration and exploitation throughout the optimization process, thereby enhancing the efficacy of t-SNE-PSO.
- The comparative experiment against baseline methods highlighted the superior performance and visualization capabilities of t-SNE-PSO across multiple evaluation metrics.
- Results underscored the efficacy of t-SNE-PSO in preserving both local and global structures within the data, surpassing existing techniques in manifold embedding tasks.

Future work may investigate the possibilities of reducing the computational complexity of t-

SNE-PSO. Furthermore, one of the main perspectives is the exploration of t-SNE-PSO's application in various domains, such as healthcare, finance, and bioinformatics, where high-dimensional data visualization and clustering are crucial. Additionally, further research can investigate the algorithm's adaptability to different datasets, including those with varying scales, noise levels, and dimensional complexities. Exploring hybrid optimization strategies, combining PSO with other optimization techniques, can also provide insights into enhancing the performance and robustness of the t-SNE-PSO algorithm. Overall, t-SNE-PSO is a promising solution in dimensionality reduction, offering new perspectives and avenues for exploration. This optimization technique enhances the versatility and flexibility of the dimensionality reduction process.

## CRediT authorship contribution statement

**Mebarka Allaoui:** Conceptualization, Software, Validation, Data Curation, Formal analysis, Writing - Original Draft. **Samir Brahim Belhaouari:** Conceptualization, Methodology, Formal analysis, Review and Editing. **Rachid Hedjam:** Conceptualization, Methodology, Validation, Formal analysis, Writing - Review and Editing. **Khadra Bouanane:** Formal analysis, Review and Editing. **Mohammed Lamine Kherfi:** Formal analysis, Review and Editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

The used datasets are publicly published.

### Appendix .1. Classification Quality

This section presents a comparative analysis of our proposed technique t-SNE-PSO against the other baseline techniques: UMAP, t-SNE-BH, t-SNE-GA, and t-SNE-OPSO. We utilized classification on the embedding spaces generated by these techniques and evaluated their performance using the Accuracy evaluation metrics. From the results reported in Tab. .5, t-SNE-PSO demonstrated significantly higher scores than the other embedding techniques in terms of all metrics. Our proposed method seems to better maintain the underlying structures and relationships within the data, resulting in a better classification.

Table .5: t-SNE-PSO vs. other baselines based on the Accuracy metric.

Algorithm	Pendigits	USPS	MNIST	FMNIST
UMAP	$0.95 \pm 1.11$	$0.94 \pm 0.00$	$0.93 \pm 0.00$	$0.71 \pm 0.00$
t-SNE	$0.93 \pm 0.00$	$0.92 \pm 0.00$	$0.93 \pm 0.00$	$0.71 \pm 0.00$
t-SNE-BH	$0.94 \pm 1.11$	$0.93 \pm 1.11$	$0.94 \pm 0.00$	$0.72 \pm 0.00$
t-SNE-GA	$0.93 \pm 0.00$	$0.92 \pm 0.00$	$0.93 \pm 0.00$	$0.71 \pm 0.00$
t-SNE-OPSO	$0.92 \pm 0.00$	$0.92 \pm 0.00$	$0.92 \pm 0.00$	$0.70 \pm 0.00$
<b>t-SNE-PSO</b>	<b><math>0.96 \pm 0.00</math></b>	<b><math>0.95 \pm 1.11</math></b>	<b><math>0.95 \pm 0.00</math></b>	<b><math>0.73 \pm 0.00</math></b>

### Appendix .2. Out-of-sample extension

To enable out-of-sample extension with t-SNE (t-SNE-PSO), we can use a technique like training a Multi-Layer Perceptron regressor (MLP). This neural network learns the mapping between the high-dimensional input and the low-dimensional embeddings produced by t-SNE (and t-SNE-PSO as well). Once trained, the MLP regressor can effectively apply the learned transformation to new, unseen data without the need to re-run t-SNE (t-SNE-PSO), which is

computationally expensive. By training the MLP regressor on original high-dimensional data and their corresponding t-SNE (t-SNE-PSO) embeddings, this method effectively bridges the gap in t-SNE’s (t-SNE-PSO) inherent limitation of handling out-of-sample data. To assess the performance of the two models, we compute the reconstruction accuracy, which is the mean square error (MSE) between the predicted and actual t-SNE (t-SNE-PSO) embeddings for the test set (unseen samples). Table .6 compares t-SNE and t-SNE-PSO in terms of MSE-based reconstruction error. The results show that the reconstruction error between the predicted and actual t-SNE-PSO embeddings is, on average, approximately 22% lower than that of t-SNE across all datasets. Individually, T-SNE-PSO consistently achieves better reconstruction errors for each dataset. This demonstrates that t-SNE-PSO outperforms t-SNE in handling out-of-sample extensions.

Table .6: The reconstruction accuracy in terms of MSE for the out-of-sample extension using MLP regressor on the embeddings of t-SNE and t-SNE-PSO.

<b>Algorithms</b>	MSE			
	Pendigits	USPS	MNIST	FMNIST
t-SNE	3419.26	3915.78	2949.04	2776.53
t-SNE-PSO	<b>882.33</b>	<b>638.19</b>	<b>629.94</b>	<b>658.28</b>
Gain Rate (%)	25.80	16.30	21.36	23.71
Average Gain Rate (%)			21.79	

## References

- Alimoglu, F., Doc, D., Alpaydin, E., & Denizhan, Y. (1996). Combining multiple classifiers for pen-based handwritten digit recognition.(1996). URL <https://citeseerx.ist.psu.edu/viewdoc/download, .>
- Becht, E., McInnes, L., Healy, J., Dutertre, C.-A., Kwok, I. W., Ng, L. G., Ginhoux, F., & Newell, E. W. (2019). Dimensionality reduction for visualizing single-cell data using umap. *Nature biotechnology*, 37, 38–44.
- Böhm, J. N., Berens, P., & Kobak, D. (2022). Unsupervised visualization of image datasets using contrastive learning. *arXiv preprint arXiv:2210.09879*, .
- Chan, D. M., Rao, R., Huang, F., & Canny, J. F. (2019). Gpu accelerated t-distributed stochastic neighbor embedding. *Journal of Parallel and Distributed Computing*, 131, 1–13.
- Crecchi, F., de Bodt, C., Verleysen, M., Lee, J. A., & Bacciu, D. (2020). Perplexity-free parametric t-sne. *arXiv preprint arXiv:2010.01359*, .
- Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (pp. 224–227).
- Ding, J., Condon, A., & Shah, S. P. (2018). Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. *Nature communications*, 9, 2002.
- Duan, Q., Shao, C., Li, X., & Shi, Y. (2017). Visualizing the search dynamics in a high-dimensional space for a particle swarm optimizer. In *Simulated Evolution and Learning: 11th International Conference, SEAL 2017, Shenzhen, China, November 10–13, 2017, Proceedings* 11 (pp. 994–1002). Springer.
- Esmin, A. A., Coelho, R. A., & Matwin, S. (2015). A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data. *Artificial Intelligence Review*, 44, 23–45.

- Fielding, B., & Zhang, L. (2020). Evolving deep denseblock architecture ensembles for image classification. *Electronics*, *9*, 1880.
- Gad, A. G. (2022). Particle swarm optimization algorithm and its applications: a systematic review. *Archives of computational methods in engineering*, *29*, 2531–2561.
- Ghasemi, M., Akbari, E., Rahimnejad, A., Razavi, S. E., Ghavidel, S., & Li, L. (2019). Phasor particle swarm optimization: a simple and efficient variant of pso. *Soft Computing*, *23*, 9701–9718.
- Gisbrecht, A., Lueks, W., Mokbel, B., Hammer, B. et al. (2012). Out-of-sample kernel extensions for nonparametric dimensionality reduction. In *ESANN* (pp. 531–536). volume 2012.
- Han, H., Zhang, T., Li, C., Benton, M. L., Wang, J., & Li, J. (2022). Explainable t-sne for single-cell rna-seq data analysis. *bioRxiv*, (pp. 2022–01).
- Hinton, G. E., & Roweis, S. (2002). Stochastic neighbor embedding. *Advances in neural information processing systems*, *15*.
- Hull, J. J. (1994). A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, *16*, 550–554.
- Jeremiah, I., Abdullahi, M., Yusuf, S. A., & Hassan, I. H. (2023). Towards an improved particle swarm optimization for feature selection: A survey. *Sule Lamido Univ. J. Sci. Technol*, *6*, 59–73.
- Kassoul, K., Belhaouari, S. B., & Cheikhrouhou, N. (2021). Dynamic cognitive-social particle swarm optimization. In *2021 7th International Conference on Automation, Robotics and Applications (ICARA)* (pp. 200–205). IEEE.
- Kassoul, K., Zufferey, N., Cheikhrouhou, N., & Belhaouari, S. B. (2022). Exponential particle swarm optimization for global optimization. *IEEE access*, *10*, 78320–78344.
- Kazemi, F., Asgarkhani, N., & Jankowski, R. (2024). Optimization-based stacked machine-learning method for seismic probability and risk assessment of reinforced concrete shear walls. *Expert Systems with Applications*, *255*, 124897.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks* (pp. 1942–1948). IEEE volume 4.
- Kobak, D., & Berens, P. (2019). The art of using t-sne for single-cell transcriptomics. *Nature communications*, *10*, 5416.
- Kobak, D., Linderman, G., Steinerberger, S., Kluger, Y., & Berens, P. (2019). Heavy-tailed kernels reveal a finer cluster structure in t-sne visualisations. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 124–139). Springer.
- Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval research logistics quarterly*, *2*, 83–97.
- Lambert, P., Lee, J. A., Couplet, E., & De Bodt, C. (2023). Nesterov momentum and gradient normalization to improve t-sne convergence and neighborhood preservation, without early exaggeration. *ESANN proceedings*, *1*, 1.

- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86, 2278–2324.
- Linderman, G. C., Rachh, M., Hoskins, J. G., Steinerberger, S., & Kluger, Y. (2019). Fast interpolation-based t-sne for improved visualization of single-cell rna-seq data. *Nature methods*, 16, 243–245.
- Liu, H., Zhang, X.-W., & Tu, L.-P. (2020). A modified particle swarm optimization using adaptive strategy. *Expert systems with applications*, 152, 113353.
- Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9.
- Mapetu, J. P. B., Chen, Z., & Kong, L. (2019). Low-time complexity and low-cost binary particle swarm optimization algorithm for task scheduling and load balancing in cloud computing. *Applied Intelligence*, 49, 3308–3330.
- McInnes, L., Healy, J., & Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, .
- Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2, 559–572.
- Petrantonakis, P. C., & Kompatsiaris, I. (2020). A dimensionality reduction method for data visualization using particle swarm optimization. In *IJCCI* (pp. 131–138).
- Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, .
- Ranzato, M., Huang, F. J., Boureau, Y.-L., & LeCun, Y. (2007). Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *2007 IEEE conference on computer vision and pattern recognition* (pp. 1–8). IEEE.
- Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20, 53–65.
- Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *science*, 290, 2323–2326.
- Salem, H. S., Mead, M. A., & El-Tawee, G. S. (2023). Wrapper-based modified binary particle swarm optimization for dimensionality reduction in big gene expression data analytics. *International Journal of Advanced Computer Science and Applications*, 14.
- Shi, Y., & Eberhart, R. C. (1999). Empirical study of particle swarm optimization. In *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)* (pp. 1945–1950). IEEE volume 3.
- Tenenbaum, J. B., De Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *science*, 290, 2319–2323.
- Thangaraj, R., Pant, M., Abraham, A., & Snasel, V. (2012). Modified particle swarm optimization with time varying velocity vector. *International Journal of Innovative Computing, Information and Control*, 8, 201–218.
- Van Der Maaten, L. (2009). Learning a parametric embedding by preserving local structure. In *Artificial intelligence and statistics* (pp. 384–391). PMLR.

- Van Der Maaten, L. (2013). Barnes-hut-sne. *arXiv preprint arXiv:1301.3342*, .
- Van Der Maaten, L. (2014). Accelerating t-sne using tree-based algorithms. *The journal of machine learning research*, 15, 3221–3245.
- Van Der Maaten, L., Postma, E., Van den Herik, J. et al. (2009). Dimensionality reduction: a comparative. *J Mach Learn Res*, 10.
- Witte, R. S., & Witte, J. S. (2017). *Statistics*. John Wiley & Sons.
- Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, .
- Xu, W., Liu, X., & Gong, Y. (2003). Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 267–273).
- Yab, L. Y., Wahid, N., & Hamid, R. A. (2022). A meta-analysis survey on the usage of meta-heuristic algorithms for feature selection on high-dimensional datasets. *IEEE Access*, 10, 122832–122856.
- Yang, H., & Du, Q. (2011). Particle swarm optimization-based dimensionality reduction for hyperspectral image classification. In *2011 IEEE International Geoscience and Remote Sensing Symposium* (pp. 2357–2360). IEEE.