AIMS Mathematics, 11(8): 27085–27115.
DOI: 10.3934/math.20261085
Received: 03 June 2026
Revised: 16 August 2026
Accepted: 19 August 2026
Published: 27 August 2026

https://www.aimspress.com/journal/Math

Research article

Adaptive graph regularized nonnegative tensor ring decomposition for data clustering

Xinyu Yao[1] **and Guimin Liu**[2,*]

[1] School of Information Engineering and Artificial Intelligence, Lanzhou University of Finance and Economics, Lanzhou 730000, Gansu, China

[2] School of Mathematics and Statistics, Lanzhou University, Lanzhou 730000, Gansu, China

* **Correspondence:** Email: 120220907870@lzu.edu.cn.

Abstract: Nonnegative tensor ring (NTR) decomposition has been successfully applied to clustering tasks due to its ability to preserve the multilinear structure of tensor data while effectively extracting physically interpretable nonnegative features. Although existing graph-regularized NTR methods capture local manifold structures, their reliance on fixed predefined similarity graphs often leads to inaccurate manifold approximations, thereby degrading clustering performance. To address this limitation, this paper proposes an adaptive graph regularized NTR (AGRNTR) model, which integrates graph regularization and similarity matrix learning within the NTR framework. This design allows the similarity graph to be adaptively refined during tensor decomposition. By jointly learning the similarity matrix and the tensor ring factors, AGRNTR captures the intrinsic geometry of tensor data more accurately than methods using fixed predefined graphs, achieving superior clustering performance. To solve the AGRNTR model, an efficient block coordinate descent algorithm combined with multiplicative updates is developed, and its convergence is theoretically guaranteed. Numerical experiments conducted on benchmark datasets demonstrate the effectiveness and superiority of the proposed model.

Keywords: nonnegative tensor ring decomposition; adaptive graph regularization; block coordinate descent; clustering
Mathematics Subject Classification: 15A69, 65F15, 68W25

1. Introduction

Clustering is a fundamental task in unsupervised learning and has been extensively applied across diverse fields, such as image and video analysis, bioinformatics, remote sensing, document organization, and multi-view learning [1–3]. Its primary goal is to automatically discover the intrinsic

grouping structure within unlabeled data by leveraging internal similarities, thereby ensuring that samples within the same group are as similar as possible, while samples across different groups are as distinct as possible. However, clustering performance is highly dependent on the quality of data representation, particularly with high-dimensional data, where the abundant redundancy within the data often masks its underlying structure and increases the difficulty of effective clustering. To address this, various dimensionality reduction techniques have been developed, such as principal component analysis (PCA) [4], singular value decomposition (SVD) [5], and nonnegative matrix factorization (NMF) [6]. Among them, NMF has received particular attention because of its ability to produce part-based and interpretable latent features, making it not only a dimensionality reduction tool but also a natural and effective clustering framework. However, standard NMF overlooks the potential geometric structural information among data. To more fully leverage the manifold structure of the data, numerous graph-regularized variants of NMF have been proposed [7–11], which incorporate neighborhood relationships to preserve local geometric properties during factorization.

While graph regularization methods have significantly enhanced the ability of NMF to capture the geometric structure of data, they are inherently limited to modeling data in matrix form. In many practical applications, data naturally possess higher dimensional tensor structures, such as multispectral images [12], video sequences [13], and multimodal data [14]. To more fully exploit the multidimensional intrinsic correlations within such data, many nonnegative tensor decomposition methods and their graph-regularized variants have been proposed. For example, as a natural extension of nonnegative matrix factorization, nonnegative Tucker decomposition (NTD) [15] has been widely applied to clustering and representation learning tasks based on tensor data. Furthermore, several graph regularized Tucker decomposition models [16–20] have been proposed by incorporating geometric structure into NTD for preserving local neighborhood relationships. Despite their effectiveness, the size of the core tensor grows exponentially with the order of the tensor, resulting in the curse of dimensionality and consequently imposing prohibitively high computational complexity. To alleviate this issue, tensor train (TT) decomposition [21] was introduced, which represents a high-order tensor as a sequential chain of third-order core tensors, thereby drastically reducing storage requirements and computational cost. Benefiting from its linear complexity with respect to tensor order, several TT decomposition-based clustering and representation learning methods have also been developed [22,23]. However, the first and last core tensors in TT decomposition are essentially matrices. This boundary rank not only limits the expressive flexibility of the model but also constrains the interactions between modes at both ends of the chain. As a consequence, these clustering and representation learning methods based on TT decomposition may fail to fully capture the intrinsic correlations among tensor modes in complex data.

To overcome the structural limitations of TT decomposition, tensor ring (TR) decomposition [24] was introduced by connecting the first and last core tensors to form a circular structure, where every core tensor remains third order, thereby achieving structural symmetry. This cyclic structure effectively enhances both flexibility and representational capacity within a compact framework. However, TR decomposition allows negative values in its factor core tensors, which makes it difficult to provide an intuitive physical interpretation in many practical applications that require nonnegative representations, such as image analysis, text mining, and recommender systems. Therefore, nonnegative tensor ring (NTR) decomposition was proposed [25] by imposing nonnegativity constraints on all core tensors. This form significantly enhances interpretability and facilitates the discovery of meaningful latent

factors, making NTR decomposition more suitable for clustering and unsupervised learning tasks. However, NTR decomposition remains insufficient in preserving the intrinsic geometric structure among data samples. In particular, it does not explicitly exploit neighborhood relationships or manifold information embedded in the data, which are known to play a crucial role in clustering performance. To address this limitation, a graph-regularized nonnegative tensor ring decomposition (GNTR) was subsequently introduced [25], which incorporates a graph Laplacian regularization term into the NTR framework. This mechanism can effectively preserve the local manifold structure in the data while learning discriminative tensor representations, thereby significantly improving the performance of clustering tasks. However, GNTR relies on a simple pairwise graph that captures only second-order relationships between samples. Such a graph is often insufficient to model complex and high-order correlations that naturally arise in multi-dimensional data. To better capture such higher-order similarities among samples, hypergraph-regularized nonnegative tensor ring decomposition (HGNTR) was proposed [26]. By replacing the conventional graph with a hypergraph, HGNTR allows each hyperedge to connect multiple samples simultaneously, thereby enhancing the capacity of the model to represent complex relationships and leading to more accurate clustering performance.

Despite the superior performance of GNTR and HGNTR models in clustering tasks, their performance heavily depends on a predefined similarity graph structure that remains fixed. Once the graph is poorly constructed, the performance of these models inevitably degrades. This limitation stems from their inability to dynamically refine the graph structure during learning, making them sensitive to noise and suboptimal initial similarity assumptions. Recent studies have demonstrated the effectiveness of adaptive graph construction in multi-view and matrix-factorization-based clustering for addressing the fixed-graph limitation, including self-supervised graph completion [27], intrinsic graph learning [28], and collaborative topological graph learning [29]. Inspired by these adaptive graph learning methods, we propose an adaptive graph-regularized nonnegative tensor ring decomposition (AGRNTR) model for tensor data clustering. This model not only preserves the inherent multilinear relationships in the tensor data but also dynamically captures the underlying local manifold structure, leading to more discriminative representations for clustering tasks. The main contributions of this paper can be summarized as follows:

- An adaptive graph-regularized nonnegative tensor ring decomposition (AGRNTR) model is proposed. Unlike existing static-graph-based NTR methods such as GNTR [25] and HGNTR [26], AGRNTR dynamically learns the similarity matrix directly from the tensor-ring factorization process, where the graph structure is progressively refined based on evolving latent representations. This adaptive mechanism enables more accurate characterization of the intrinsic local geometry, ultimately enhancing clustering performance.
- AGRNTR integrates tensor ring decomposition, adaptive graph learning, and spectral clustering embedding into a unified optimization framework, which allows representation learning, graph construction, and clustering to guide each other during iterations. Moreover, in contrast to existing adaptive graph-regularized methods, which either flatten tensor data into matrices (e.g., adaptive graph NMF [8]) or constrain the learned graph to approximate the similarity matrix (e.g., adaptive graph NTD [20]), AGRNTR fully preserves the multilinear structure of tensor data via tensor ring decomposition while allowing the similarity graph to evolve freely without any prior constraints. This avoids both structural information loss and biased graph construction, thus yielding more discriminative representations for clustering.

- An effective optimization algorithm for solving AGRNTR is developed via the block coordinate descent method combined with the multiplicative update rule, and its convergence and computational complexity are rigorously analyzed.
- Experimental results on several real-world tensor datasets demonstrate that the proposed AGRNTR method outperforms several state-of-the-art clustering methods, including both matrix-based and tensor-based approaches.

The remainder of this paper is organized as follows. Section 2 introduces essential notations and preliminaries and provides a brief review of TR, NTR, GNTR, and adaptive graph learning. Section 3 presents the AGRNTR model, along with the corresponding optimization algorithm, convergence analysis, and computational complexity discussion. Section 4 provides extensive experiments to validate the effectiveness of the proposed method. Finally, Section 5 concludes the paper.

2. Preliminaries

This section provides an overview of tensor notations and introduces the fundamental concepts of tensor ring decomposition that serve as the foundation for the subsequent discussion.

2.1. Notations and definitions

We use $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ to denote the set of all real-valued tensors of size $I_1 \times I_2 \times \cdots \times I_N$. The main symbols and notations used in this paper are summarized in Table 1.

Table 1. List of the notations.

Notations	Descriptions	Notations	Descriptions
x	A scalar	$\mathbf{I}$	Identity matrix
$\mathbf{x}$	A vector	$\mathbf{e}$	A vector with all elements 1
$\mathbf{X}$	A matrix	$\langle \cdot, \cdot \rangle$	Inner product of matrix
$\mathcal{X}$	A tensor	$\|\| \cdot \|\|_F$	Frobenius norm
$\mathrm{Tr}(\cdot)$	Trace of matrix	$\overline{\times}^1$	Multilinear product
$\otimes$	Kronecker product	$\times_n$	Mode-n product

Definition 2.1. (Mode-n Unfolding) [30] For an N-order tensor $\mathcal{U} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, its mode-$n$ unfolding rearranges the tensor $\mathcal{U}$ into a matrix $\mathbf{U}_{(n)} \in \mathbb{R}^{I_n \times I_1 \cdots I_{n-1} I_{n+1} \cdots I_N}$, whose entries are given by

$$\mathbf{U}_{(n)}\left(i_n, \overline{i_1 \cdots i_{n-1} i_{n+1} \cdots i_N}\right) = \mathcal{U}(i_1, i_2, \ldots, i_N),$$

in which $\overline{i_1 \cdots i_{n-1} i_{n+1} \cdots i_N} = 1 + \sum_{k \neq n} (i_k - 1) \prod_{\substack{j<k, \\ j \neq n}} I_j$.

Different from the classical mode-n unfolding, Zhao et al. [24] introduced that in a cyclic mode-n unfolding, which rearranges the tensor $\mathcal{U}$ into a matrix $\mathbf{U}_{[n]} \in \mathbb{R}^{I_n \times I_{n+1} \cdots I_N I_1 \cdots I_{n-1}}$, its entries are defined as

$$\mathbf{U}_{[n]}\left(i_n, \overline{i_{n+1} \cdots i_N i_1 \cdots i_{n-1}}\right) = \mathcal{U}(i_1, i_2, \ldots, i_N).$$

Notably, these two unfolding methods produce matrices of identical size, differing only by a column permutation. The cyclic mode-n unfolding is primarily employed in tensor ring decomposition.

Definition 2.2. (Inner Product) [30] Let $\mathcal{U}$ and $\mathcal{V} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. Then their inner product is defined as the sum of their elementwise products over all entries, i.e.,

$$\langle \mathcal{U}, \mathcal{V} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} u_{i_1 i_2 \cdots i_N} v_{i_1 i_2 \cdots i_N}. \tag{2.1}$$

Definition 2.3. (Mode-n Product) [30] For a tensor $\mathcal{U} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and a matrix $\mathbf{V} \in \mathbb{R}^{J \times I_n}$, their mode-$n$ product is denoted as $\mathcal{U} \times_n \mathbf{V}$ and is of size $I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N$, whose entries are given by

$$(\mathcal{U} \times_n \mathbf{V})_{i_1 \cdots i_{n-1} j i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} u_{i_1 i_2 \cdots i_N} v_{j i_n}. \tag{2.2}$$

Definition 2.4. (Multilinear Product) [31] Let $\mathcal{G}^{(n)} \in \mathbb{R}^{R_{n-1} \times I_n \times R_n}$ and $\mathcal{G}^{(n+1)} \in \mathbb{R}^{R_n \times I_{n+1} \times R_{n+1}}$. Then the multilinear product of $\mathcal{G}^{(n)}$ and $\mathcal{G}^{(n+1)}$ is defined by $\mathcal{G}^{(n,n+1)} = \mathcal{G}^{(n)} \overline{\times}^1 \mathcal{G}^{(n+1)}$, where $\mathcal{G}^{(n,n+1)} \in \mathbb{R}^{R_{n-1} \times (I_n I_{n+1}) \times R_{n+1}}$ and its entries are given by

$$\mathcal{G}^{(n,n+1)}_{r_{n-1},(i_n-1)I_{n+1}+i_{n+1},r_{n+1}} = \sum_{r_n=1}^{R_n} \mathcal{G}^{(n)}_{r_{n-1},i_n,r_n} \mathcal{G}^{(n+1)}_{r_n,i_{n+1},r_{n+1}} \tag{2.3}$$

in which $i_n = 1, \ldots, I_n$ and $i_{n+1} = 1, \ldots, I_{n+1}$.

2.2. Tensor ring and its variants

In this subsection, we briefly review tensor ring decomposition and its several variants, including NTR and GNTR.

As a generalized extension of the tensor train decomposition, TR decomposition effectively preserves multi-way correlations while maintaining controllable model complexity. Consequently, it has garnered increasing attention across fields such as signal processing, computer vision, and machine learning. A TR decomposition [24] represents a higher-order tensor as the circular contraction of a sequence of third-order core tensors. Specifically, given an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, its TR decomposition is defined as $\mathcal{X} = \mathrm{TR}(\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(N)})$, where $\mathcal{G}^{(n)} \in \mathbb{R}^{R_{n-1} \times I_n \times R_n}$, $n = 1, \ldots, N$, is called a core tensor and $R_0 = R_N$. The entries of $\mathcal{X}$ are given by

$$\mathcal{X}(i_1, \ldots, i_N) = \sum_{r_1=1}^{R_1} \cdots \sum_{r_N=1}^{R_N} \prod_{n=1}^{N} \mathcal{G}^{(n)}(r_{n-1}, i_n, r_n). \tag{2.4}$$

$\mathbf{R} = [R_1, R_2, \ldots, R_N]$ is called the TR rank of $\mathcal{X}$.

Although TR decomposition offers a flexible framework for high-order tensor representation, its application to inherently nonnegative data (such as in image processing) may lead to reduced interpretability of the learned factors due to the absence of nonnegativity constraints. To address this issue, Yu et al. [25] proposed the nonnegative tensor ring (NTR) model, which enforces element-wise nonnegativity on all core tensors and thus yields more interpretable additive representations. The NTR decomposition is specially given as follows:

$$\begin{aligned} \min_{\mathcal{G}^{(1)},\ldots,\mathcal{G}^{(N)}} \quad & \frac{1}{2} \left\| \mathcal{X} - \mathrm{NTR}\left(\mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \ldots, \mathcal{G}^{(N)}\right) \right\|_F^2 \\ \text{s.t.} \quad & \mathcal{G}^{(n)} \geq 0, \ n = 1, 2, \ldots, N, \end{aligned} \tag{2.5}$$

where $\mathcal{G}^{(n)} \in \mathbb{R}^{R_{n-1}\times I_n\times R_n}$ denotes the n-th nonnegative core tensor. As shown in [24], the NTR decomposition (2.5) can be equivalently reformulated into the following N least squares problems, and thus (2.5) is typically solved by the alternating least squares (ALS) algorithm,

$$\begin{aligned} &\min_{\mathbf{G}_{(2)}^{(n)}} \frac{1}{2}\left\|\mathbf{X}_{[n]} - \mathbf{G}_{(2)}^{(n)}\left(\mathbf{G}_{[2]}^{\neq n}\right)^{T}\right\|_F^2 \\ &\text{s.t.} \quad \mathbf{G}_{(2)}^{(n)} \geq 0, \quad n = 1, 2, \cdots, N, \end{aligned} \tag{2.6}$$

where $\mathbf{G}_{(2)}^{(n)} \in \mathbb{R}^{I_n\times R_n R_{n+1}}$ denotes the mode-2 unfolding matrix of the n-th core tensor $\mathcal{G}^{(n)}$. Meanwhile, $\mathbf{G}_{[2]}^{\neq n} \in \mathbb{R}^{(I_{n+1}\cdots I_N I_1\cdots I_{n-1})\times R_{n-1}R_n}$ represents the cyclic mode-2 unfolding of the subchain tensor $\mathcal{G}^{(\neq n)}$. The subchain tensor $\mathcal{G}^{(\neq n)}$ is obtained by contracting all core tensors except the n-th one, that is, it is expressed as follows:

$$\mathcal{G}^{(\neq n)} = \mathcal{G}^{(n+1)}\overline{\times}^1 \cdots \overline{\times}^1 \mathcal{G}^{(N)}\overline{\times}^1 \mathcal{G}^{(1)}\overline{\times}^1 \cdots \overline{\times}^1 \mathcal{G}^{(n-1)}. \tag{2.7}$$

Although NTR decomposition improves the interpretability of the obtained factors partially, it does not explicitly model the manifold structure or local similarity relationships among samples. To further preserve the intrinsic geometric consistency of the data and enhance the discriminative capability of the representation, Yu et al. proposed the GNTR model in [25]. In the GNTR model, a neighborhood graph is first constructed based on pairwise sample similarities, from which the corresponding graph Laplacian is derived. The resulting graph regularization term is then incorporated into the NTR objective function and imposed on the last core tensor of the tensor ring decomposition. In this way, the factorization process is guided by the underlying manifold geometry of the data while maintaining the nonnegativity constraints. The mathematical formulation of GNTR is given as follows:

$$\begin{aligned} &\min_{\mathbf{G}_{(2)}^{(n)}} \frac{1}{2}\left\|\mathbf{X}_{[n]} - \mathbf{G}_{(2)}^{(n)}\left(\mathbf{G}_{[2]}^{\neq n}\right)^{\mathrm{T}}\right\|_F^2 + \frac{\alpha}{2}\operatorname{Tr}\left(\left(\mathbf{G}_{(2)}^{(N)}\right)^{\mathrm{T}}\mathbf{L}_s\mathbf{G}_{(2)}^{(N)}\right) \\ &\text{s.t.} \quad \mathbf{G}_{(2)}^{(n)} \geq 0,\ \mathbf{G}_{[2]}^{\neq n} \geq 0,\ n = 1, 2, \ldots, N, \end{aligned} \tag{2.8}$$

where $\mathbf{L}_s = \mathbf{D} - \mathbf{S}$ denotes the graph Laplacian matrix, $\mathbf{S} \in \mathbb{R}^{n\times n}$ is the similarity matrix, $\mathbf{D} \in \mathbb{R}^{n\times n}$ is a diagonal degree matrix with entries $\mathbf{D}_{ii} = \sum_j s_{ij}$, and $\alpha \geq 0$ is a trade-off parameter to control the contribution of the graph regularization term.

2.3. Adaptive graph

In clustering tasks, the quality of the constructed graph critically determines the performance of clustering algorithms. Conventional graph construction methods based on fixed rules usually rely on predefined parameters (e.g., neighborhood size or kernel bandwidth). These parameters not only require laborious manual tuning for different datasets but also have limited adaptability to the intrinsic non-uniformity of data density and underlying structures. To overcome these limitations, a series of adaptive graph learning methods has been developed [20, 32, 33]. These approaches reframe the construction of a similarity graph as an optimizable learning problem, enabling the automatic inference of optimal pairwise affinities directly from the data. This adaptive mechanism not only reduces the dependency on manually defined parameters but also dynamically aligns the graph structure with the intrinsic geometric relationships in the data, leading to more flexible and accurate representations.

Among these, a widely adopted framework was introduced by Nie et al. [33], formulated as follows:

$$\begin{cases} \min\limits_{\mathbf{S}} \sum\limits_{i,j} \left(\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2 \right) \\ \text{s.t. } \forall i, \sum_j s_{ij} = 1;\ 0 \le s_{ij} \le 1, \end{cases} \tag{2.9}$$

where $\{x_1, x_2, \ldots, x_n\}$ represents the set of sample data points and $\mathbf{S} \in \mathbb{R}^{n \times n}$ is a similarity matrix. In (2.9), $\|x_i - x_j\|_2^2$ measures the pairwise Euclidean distance between data points, encouraging larger similarity values for closer neighbors, while the quadratic regularization term s_{ij}^2 prevents the similarity matrix from degenerating into a trivial identity assignment. The parameter $\gamma > 0$ controls the smoothness of the learned similarity distribution.

Although the formulation in (2.9) enables adaptive similarity learning, it does not explicitly guarantee a well-separated clustering structure. In practice, the resulting graph often forms a single connected component, which is undesirable for clustering tasks. To address this issue, Nie et al. further imposed a structural constraint on the graph Laplacian, and thus obtained the following model:

$$\begin{cases} \min\limits_{\mathbf{S}} \sum\limits_{i,j} \left(\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2 \right) \\ \text{s.t. } \forall i, \sum_j s_{ij} = 1,\ 0 \le s_{ij} \le 1,\ \text{rank}(\mathbf{L}_S) = n - c, \end{cases} \tag{2.10}$$

where $\mathbf{L}_S = \mathbf{D} - \mathbf{S}$ is the Laplacian matrix associated with $\mathbf{S}$, $\mathbf{D} \in \mathbb{R}^{n \times n}$ is a diagonal degree matrix with entries $\mathbf{D}_{ii} = \sum_j s_{ij}$, and $\text{rank}(\mathbf{L}_S) = n - c$ ensures that the learned graph contains exactly c connected components, corresponding to c clusters.

However, directly handling the rank constraint is computationally intractable. According to the Ky Fan theorem [34], the rank condition can be equivalently relaxed by minimizing the sum of the smallest c eigenvalues of $\mathbf{L}_S$, which leads to the following equivalent formulation:

$$\sum_{i=1}^{c} \sigma_i(\mathbf{L}_S) = \min_{\mathbf{F} \in \mathbb{R}^{n \times c},\ \mathbf{F}^\top \mathbf{F} = \mathbf{I}} \text{Tr}(\mathbf{F}^\top \mathbf{L}_S \mathbf{F}),$$

where $\mathbf{F}$ can be interpreted as a continuous cluster indicator matrix. Incorporating the above equation into (2.9) yields the following optimization model:

$$\begin{cases} \min\limits_{\mathbf{S},\mathbf{F}} \sum\limits_{i,j} \left(\|x_i - x_j\|_2^2 s_{ij} + \gamma s_{ij}^2 \right) + 2\lambda\, \text{Tr}(\mathbf{F}^\top \mathbf{L}_S \mathbf{F}) \\ \text{s.t. } \forall i, \sum_j s_{ij} = 1,\ 0 \le s_{ij} \le 1,\ \mathbf{F}^\top \mathbf{F} = \mathbf{I}, \end{cases} \tag{2.11}$$

where λ is a trade-off parameter controlling the strength of the rank constraint. Particularly, when λ is sufficiently large, the optimization enforces $\sum_{i=1}^{c} \sigma_i(\mathbf{L}_S) = 0$, thereby satisfying $\text{rank}(\mathbf{L}_S) = n - c$.

As shown in Nie et al. [33], jointly optimizing the similarity graph and clustering representation is crucial for uncovering the intrinsic data structure. This adaptive strategy allows the learned graph to align with the underlying data geometry, avoids the bias of fixed neighborhood assumptions, and yields clearer cluster structures with improved robustness and accuracy.

3. Adaptive graph-regularized nonnegative tensor ring decomposition

In this section, we first construct an adaptive graph-regularized nonnegative tensor ring model (AGRNTR). Then, we efficiently solve this model based on the block coordinate descent (BCD) method. Finally, convergence analysis and computational complexity are provided.

3.1. Objective function of AGRNTR

To enable the graph structure to adaptively evolve with the latent representations during factorization, we incorporate the adaptive graph learning model in Eq (2.11) into the NTR decomposition framework, yielding the AGRNTR model, which is formulated as follows:

$$\begin{aligned} &\min_{\mathcal{G}^{(1)},\ldots,\mathcal{G}^{(N)},\mathbf{S},\mathbf{F}} \frac{1}{2}\Big[\Big\|\mathcal{X}-\mathrm{TR}\left((\mathcal{G}^{(k)})_{k=1}^{N}\right)\Big\|_F^2+\alpha\,\mathrm{tr}\left(\left(\mathbf{G}_{(2)}^{(N)}\right)^T\mathbf{L_S}\mathbf{G}_{(2)}^{(N)}\right)\Big]+\gamma\|\mathbf{S}\|_F^2+2\lambda\mathrm{tr}\left(\mathbf{F}^T\mathbf{L_S}\mathbf{F}\right) \\ &\text{s.t. } \mathcal{G}^{(1)}\geq 0,\ldots,\mathcal{G}^{(N)}\geq 0,\mathbf{Se}=\mathbf{e},\mathbf{S}\geq 0,\mathbf{F}^T\mathbf{F}=\mathbf{I}, \end{aligned} \tag{3.1}$$

where $\mathcal{X}\in\mathbb{R}^{I_1\times\cdots\times I_N}$ is the high-order input tensor, $\mathbf{S}\in\mathbb{R}^{I_N\times I_N}$ is the learned similarity matrix, $\mathbf{L_S}$ is the corresponding graph Laplacian, and $\mathbf{F}\in\mathbb{R}^{I_N\times c}$ is the cluster indicator matrix with c clusters. Note that the last mode of the input tensor $\mathcal{X}$ corresponds to the sample dimension, i.e., I_N denotes the number of samples. Consequently, the unfolded core factor along this mode, denoted as $\mathbf{G}_{(2)}^{(N)}$, provides the low-dimensional representation of the samples in the NTR decomposition. Hence, the graph regularization term $\mathrm{tr}\left(\mathbf{G}_{(2)}^{(N)^T}\mathbf{L_S}\mathbf{G}_{(2)}^{(N)}\right)$ is incorporated to preserve the local geometric structure of the samples.

According to (2.6), the tensor-based formulation in (3.1) can be equivalently rewritten in matrix form as:

$$\begin{aligned} \mathcal{L}=&\min_{\mathbf{G}_{(2)}^{(k)},\mathbf{S},\mathbf{F}} \frac{1}{2}\Big[\Big\|\mathbf{X}_{[k]}-\mathbf{G}_{(2)}^{(k)}\left(\mathbf{G}_{[2]}^{\neq k}\right)^T\Big\|_F^2+\alpha\,\mathrm{tr}\left(\left(\mathbf{G}_{(2)}^{(N)}\right)^T\mathbf{L_S}\mathbf{G}_{(2)}^{(N)}\right)\Big]+\gamma\|\mathbf{S}\|_F^2+2\lambda\mathrm{tr}\left(\mathbf{F}^T\mathbf{L_S}\mathbf{F}\right) \\ &\text{s.t.}\quad \mathbf{G}_{(2)}^{(k)}\geq 0\ (k=1,2,\ldots,N),\mathbf{Se}=\mathbf{e},\ \mathbf{S}\geq 0,\mathbf{F}^T\mathbf{F}=\mathbf{I}. \end{aligned} \tag{3.2}$$

3.2. Optimization

To efficiently solve the AGRNTR model, a block coordinate descent (BCD) algorithm combined with multiplicative update (MU) rules is developed.

3.2.1. Update of $\mathbf{G}_{(2)}^{(k)}$

When $\mathbf{S}$ and $\mathbf{F}$ are fixed, updating $\mathbf{G}_{(2)}^{(k)}$ reduces problem (3.2) to the following subproblem:

$$\min_{\mathbf{G}_{(2)}^{(k)}\geq 0}\Psi(\mathbf{G}_{(2)}^{(k)})=\frac{1}{2}\|\mathbf{X}_{[k]}-\mathbf{G}_{(2)}^{(k)}\left(\mathbf{G}_{[2]}^{\neq k}\right)^T\|_F^2+\frac{\alpha}{2}\,\mathrm{tr}\left(\left(\mathbf{G}_{(2)}^{(N)}\right)^T\mathbf{L_S}\mathbf{G}_{(2)}^{(N)}\right). \tag{3.3}$$

The formulation of the Lagrangian function $\mathcal{L}^{(k)}$ is expressed as follows:

$$\begin{aligned} \mathcal{L}^{(k)}=&\frac{1}{2}\mathrm{tr}\Big(\mathbf{X}_{[k]}(\mathbf{X}_{[k]})^\top-2\mathbf{G}_{(2)}^{(k)}\left(\mathbf{G}_{[2]}^{\neq k}\right)^\top(\mathbf{X}_{[k]})^\top+\mathbf{G}_{(2)}^{(k)}\left(\mathbf{G}_{[2]}^{\neq k}\right)^\top\mathbf{G}_{(2)}^{(k)}\left(\mathbf{G}_{[2]}^{\neq k}\right)^\top\Big) \\ &+\frac{\alpha}{2}\mathrm{tr}\left(\left(\mathbf{G}_{(2)}^{(N)}\right)^\top\mathbf{L_S}\mathbf{G}_{(2)}^{(N)}\right)+\mathrm{tr}\left(\mathbf{Z}\left(\mathbf{G}_{(2)}^{(k)}\right)^\top\right), \end{aligned} \tag{3.4}$$

where $\mathbf{Z}^{(k)} = [z_{ij}^{(k)}] \in \mathbb{R}^{I_k \times R_k R_{k+1}}$ denotes the Lagrange multiplier matrix associated with the nonnegative constraint $\mathbf{G}_{(2)}^{(k)} \geq 0$.

In the case of $k = N$, the derivatives of $\mathcal{L}^{(k)}$ with respect to $\mathbf{G}_{(2)}^{(N)}$ are as follows:

$$\frac{\partial \mathcal{L}^{(k)}}{\partial \mathbf{G}_{(2)}^{(N)}} = -\mathbf{X}_{[N]}\mathbf{G}_{[2]}^{\neq N} + \mathbf{G}_{(2)}^{(N)}\left(\mathbf{G}_{[2]}^{\neq N}\right)^{\top}\mathbf{G}_{[2]}^{\neq N} + \alpha \mathbf{L}_{\mathbf{S}}\mathbf{G}_{(2)}^{(N)} + \mathbf{Z}^{(N)}. \tag{3.5}$$

According to the Karush-Kuhn-Tucker (KKT) conditions, we have $\frac{\partial \mathcal{L}^{(k)}}{\partial \mathbf{G}_{(2)}^{(N)}} = 0$ and the complementarity condition $z_{ij}^{(N)}\left(\mathbf{G}_{(2)}^{(N)}\right)_{ij} = 0$. Combining these conditions with Eq. (3.5) yields the following element-wise update rule for $\mathbf{G}_{(2)}^{(N)}$:

$$\left(\mathbf{G}_{(2)}^{(N)}\right)_{iN} \leftarrow \left(\mathbf{G}_{(2)}^{(N)}\right)_{iN} \frac{\left(\mathbf{X}_{[N]}\mathbf{G}_{[2]}^{\neq N} + \alpha \mathbf{S}\mathbf{G}_{(2)}^{(N)}\right)_{iN}}{\left(\mathbf{G}_{(2)}^{(N)}\left(\mathbf{G}_{[2]}^{\neq N}\right)^{\top}\mathbf{G}_{[2]}^{\neq N} + \alpha \mathbf{D}\mathbf{G}_{(2)}^{(N)}\right)_{iN}}. \tag{3.6}$$

When $k = 1, 2, \ldots, N-1$, the graph Laplacian regularization term is absent from the objective $\mathcal{L}^{(k)}$. Consequently, the derivative of $\mathcal{L}^{(k)}$ with respect to $\mathbf{G}_{(2)}^{(k)}$ takes the form

$$\frac{\partial \mathcal{L}^{(k)}}{\partial \mathbf{G}_{(2)}^{(k)}} = -\mathbf{X}_{[k]}\mathbf{G}_{[2]}^{\neq k} + \mathbf{G}_{(2)}^{(k)}\left(\mathbf{G}_{[2]}^{\neq k}\right)^{\top}\mathbf{G}_{[2]}^{\neq k} + \mathbf{Z}^{(k)}. \tag{3.7}$$

Applying the KKT condition, we obtain the update rule of $\mathbf{G}_{(2)}^{(k)}$ for $k = 1, 2, \ldots, N-1$ as follows:

$$\left(\mathbf{G}_{(2)}^{(k)}\right)_{ik} \leftarrow \left(\mathbf{G}_{(2)}^{(k)}\right)_{ik} \frac{\left(\mathbf{X}_{[k]}\mathbf{G}_{[2]}^{\neq k}\right)_{ik}}{\left(\mathbf{G}_{(2)}^{(k)}\left(\mathbf{G}_{[2]}^{\neq k}\right)^{\top}\mathbf{G}_{[2]}^{\neq k}\right)_{ik}}. \tag{3.8}$$

Remark 3.1. The multiplicative updates involve denominators that are theoretically nonnegative but may become vanishingly small in practice, leading to numerical instability. To address this, we adopt strictly positive random initialization for all core tensors in our experiments. The entries of these tensors are drawn from the uniform distribution on the interval $(0, 1)$. Meanwhile, a small constant $\epsilon = 10^{-16}$ is added to each denominator term.

3.2.2. Update of $\mathbf{S}$

By fixing the core tensors $\mathbf{G}_{(2)}^{(k)}$ $(k = 1, 2, \ldots, N)$ and the matrix F, it follows from (3.2) that the similarity matrix $\mathbf{S}$ is updated by solving the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{S}} \frac{\alpha}{2}\operatorname{tr}\left(\left(\mathbf{G}_{(2)}^{(N)}\right)^{T}\mathbf{L}_{\mathbf{S}}\mathbf{G}_{(2)}^{(N)}\right) + \gamma\|\mathbf{S}\|_F^2 + 2\lambda\operatorname{tr}\left(\mathbf{F}^{T}\mathbf{L}_{\mathbf{S}}\mathbf{F}\right) \\ &\text{s.t. } \mathbf{S}\mathbf{e} = \mathbf{e}, \mathbf{S} \geq 0. \end{aligned} \tag{3.9}$$

Note that

$$\operatorname{tr}\left(\mathbf{G}_{(2)}^{(N)T}\mathbf{L}_{\mathbf{S}}\mathbf{G}_{(2)}^{(N)}\right) = \frac{1}{2}\sum_{i=1}^{I_N}\sum_{j=1}^{I_N}\left\|\mathbf{G}_{(2)}^{(N)}(i,:)^{T} - \mathbf{G}_{(2)}^{(N)}(j,:)^{T}\right\|_2^2 s_{ij}.$$

Let $d_{ij}^{\mathbf{G}} = \left\|\mathbf{G}_{(2)}^{(N)}(i,:)^T - \mathbf{G}_{(2)}^{(N)}(j,:)^T\right\|_2^2$. Then the above equation can be rewritten as:

$$\mathrm{tr}\left(\mathbf{G}_{(2)}^{(N)T}\mathbf{L}_{\mathbf{S}}\mathbf{G}_{(2)}^{(N)}\right) = \frac{1}{2}\sum_{i=1}^{I_N}\sum_{j=1}^{I_N} d_{ij}^{\mathbf{G}}\, s_{ij}.$$

Similarly, by defining $d_{ij}^{\mathbf{F}} = \|f_i - f_j\|_2^2$, we have

$$2\,\mathrm{tr}\left(\mathbf{F}^T\mathbf{L}_{\mathbf{S}}\mathbf{F}\right) = \sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}^{\mathbf{F}}\, s_{ij}.$$

Let $\mathbf{d}_i = [d_{i1}, d_{i2}, \ldots, d_{iI_N}]^T \in \mathbb{R}^{I_N\times 1}$ be a vector whose j-th element is defined as $d_{ij} = \frac{\alpha}{4}d_{ij}^{\mathbf{G}} + \lambda d_{ij}^{\mathbf{F}}$. Then, optimization problem (3.9) can be equivalently rewritten as follows:

$$\begin{cases}\min_{\mathbf{S}_i} \mathbf{d}_i^T\mathbf{S}_i + \gamma\mathbf{S}_i^2 \\ \text{s.t.} \quad \mathbf{S}_i^T\mathbf{e} = \mathbf{e},\ \mathbf{S}_i \geq 0,\end{cases} \tag{3.10}$$

where $\mathbf{S}_i = [s_{i1}, s_{i2}, \ldots, s_{iI_N}]^T$. After straightforward derivation, optimization problem (3.10) corresponding to each row vector $\mathbf{S}_i$ can be equivalently reformulated as follows:

$$\min_{\substack{S_i\geq 0\\ S_i^\top e=e}} \left\|S_i + \frac{1}{2\gamma}d_i\right\|_2^2. \tag{3.11}$$

For each i, the following Lagrangian function is constructed to solve the constrained optimization problem (3.11).

$$\mathcal{L}(S_i, \eta, \theta_i) = \frac{1}{2}\left\|S_i + \frac{1}{2\gamma}d_i\right\|_2^2 - \eta\left(S_i^\top\mathbf{1} - 1\right) - \theta_i^\top S_i, \tag{3.12}$$

where η and θ_i denote the Lagrange multipliers corresponding to the imposed constraints.

To derive the optimal solution for each element s_{ij} of the vector $\mathbf{S}_i$, the KKT conditions are applied to (3.12), yielding the following results:

$$s_{ij} = \left(-\frac{d_{ij}}{2\gamma} + \eta\right)_+, \tag{3.13}$$

where $(\cdot)_+ = \max(\cdot, 0)$.

3.2.3. Update of F

By fixing all variables except $\boldsymbol{F}$, the optimization problem associated with (3.2) reduces to the following subproblem:

$$\begin{aligned}&\min_{\boldsymbol{F}}\ \mathrm{Tr}\left(\boldsymbol{F}^T\mathbf{L}_{\mathbf{S}}\boldsymbol{F}\right)\\ &\text{s.t. } \boldsymbol{F}^T\boldsymbol{F} = \boldsymbol{I},\end{aligned} \tag{3.14}$$

where $\mathbf{L}_{\mathbf{S}}$ denotes the graph Laplacian induced by the learned similarity matrix $\mathbf{S}$. We thus obtain the update of matrix $\boldsymbol{F}$ by (3.14), which is a standard trace minimization problem under orthogonality constraints. As established in [35, 36], its optimal solution is obtained by selecting the eigenvectors corresponding to the smallest c eigenvalues of the Laplacian matrix $\mathbf{L}_{\mathbf{S}}$.

After updating $\boldsymbol{F}$ through the above steps, a complete iteration of all variables in model (3.1) is completed. The detailed update steps are outlined in Algorithm 1.

Algorithm 1 Adaptive Graph-Regularized Nonnegative Tensor Ring Model

Require: Tensor dataset $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, target ranks $(R_1, R_2, \ldots, R_N)$, number of clusters c, parameters α, γ, λ, and maximum iterations for tensor core update $t_{\max}$.

Ensure: Nonnegative tensor cores $\mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \ldots, \mathcal{G}^{(N)}$, the similarity matrix $\boldsymbol{S}$.

Initialize the similarity matrix $\mathbf{S}$ from the dataset and nonnegative tensor cores $\mathcal{G}^{(k)} \geq 0$ $(k = 1, 2, \ldots, N)$.
repeat
  for $k = 1$ **to** N **do**
    for $t = 1$ **to** $t_{\max}$ **do**
      if $k = N$ **then**
        Update $\mathbf{G}_{(2)}^{(k)}$ by Eq. (3.6).
      else
        Update $\mathbf{G}_{(2)}^{(k)}$ by Eq. (3.8).
      end if
    end for
    $\mathcal{G}^{(k)} \leftarrow \text{folding}\left(\mathbf{G}_{(2)}^{(k)\, t_{\max}}\right)$, where folding$(\cdot)$ denotes the inverse of the mode-2 unfolding, which reshapes a matrix back into a tensor of the original dimensions.
  end for
  Update $\mathbf{F}$ by the eigenvectors of $\mathbf{L_S}$ associated with the c smallest eigenvalues.
  For each i, update the i-th row of matrix $\boldsymbol{S}$ by (3.13).
until convergence.

3.3. Convergence analysis

In this section, the convergence of Algorithm 1 for solving the proposed AGRNTR model is established. To this end, we first introduce the definition of the following auxiliary function.

Definition 3.1. [37] Given an objective function $\Phi(x)$, a bivariate function $P(x, x^t)$ is called its auxiliary function if the following two conditions hold:

$$P(x, x^t) \geq \Phi(x), \quad P(x^t, x^t) = \Phi(x^t), \tag{3.15}$$

where x^t represents the updated variable obtained at the t-th iterative step.

Lemma 3.1. [37] Let $P(x, x^t)$ be the auxiliary function of $\Phi(x)$. Suppose that the iterative update rule is formulated as

$$x^{t+1} = \arg\min_x P(x, x^t). \tag{3.16}$$

Then the objective function $\Phi(x)$ satisfies the monotonic non-increasing property, i.e., $\Phi(x^{t+1}) \leq \Phi(x^t)$.

Lemma 3.2. [38] Let $\boldsymbol{L}_S$ be a symmetric positive semidefinite matrix. For the optimization problem $\min_{\boldsymbol{F}^{\mathrm{T}}\boldsymbol{F}=\boldsymbol{I}} \operatorname{tr}\left(\boldsymbol{F}^{\mathrm{T}}\boldsymbol{L}_S\boldsymbol{F}\right)$, the optimal solution $\boldsymbol{F}$ is given by choosing the eigenvectors corresponding to the d smallest eigenvalues of $\boldsymbol{L}_S$.

To prove the monotonic non-increasing property of the objective function in (3.2) with respect to the core tensor under the updating rules (3.6) and (3.8), we need to construct an auxiliary function. For any $k = 1, 2, \ldots, N$, let $\Phi_{ij}^{(k)}(x)$ denote the part of objective function (3.2) that only involves the element

$(\mathbf{G}_{(2)}^{(k)})_{ij}$. When $k = N$, the first- and second-order partial derivatives of $\Phi_{ij}^{(N)}(x)$ with respect to $(\mathbf{G}_{(2)}^{(N)})_{ij}$ are given by

$$(\Phi^{(N)})'_{ij} = -(\mathbf{X}_{[N]}\mathbf{G}_{[2]}^{\neq N})_{ij} + \left(\mathbf{G}_{(2)}^{(N)}(\mathbf{G}_{[2]}^{\neq N})^\top \mathbf{G}_{[2]}^{\neq N}\right)_{ij} + \alpha(\mathbf{L}_\mathbf{S}\mathbf{G}_{(2)}^{(N)})_{ij} \tag{3.17}$$

and

$$(\Phi^{(N)})''_{ij} = \left((\mathbf{G}_{[2]}^{\neq N})^\top \mathbf{G}_{[2]}^{\neq N}\right)_{jj} + \alpha(\mathbf{L}_\mathbf{S})_{ii}. \tag{3.18}$$

To facilitate the convergence analysis, we construct an auxiliary function for $\Phi_{ij}^{(N)}(x)$. Unless otherwise specified, all variables appearing below are evaluated at the current iteration t.

Lemma 3.3. For the N-th mode, the function

$$\begin{aligned} P\left(x, (\mathbf{G}_{(2)}^{(N)})_{ij}^t\right) = \Phi_{ij}^{(N)}\left((\mathbf{G}_{(2)}^{(N)})_{ij}^t\right) + (\Phi^{(N)})'_{ij}\left((\mathbf{G}_{(2)}^{(N)})_{ij}^t\right)\left(x - (\mathbf{G}_{(2)}^{(N)})_{ij}^t\right) \\ + \frac{\left((\mathbf{G}_{(2)}^{(N)})^t(\mathbf{G}_{[2]}^{\neq N})^\top \mathbf{G}_{[2]}^{\neq N} + \alpha\mathbf{D}_\mathbf{S}(\mathbf{G}_{(2)}^{(N)})^t\right)_{ij}}{2(\mathbf{G}_{(2)}^{(N)})_{ij}^t}\left(x - (\mathbf{G}_{(2)}^{(N)})_{ij}^t\right)^2 \end{aligned} \tag{3.19}$$

is an auxiliary function of $\Phi_{ij}^{(N)}(x)$.

Proof. It is straightforward to verify that $P\left((\mathbf{G}_{(2)}^{(N)})_{ij}^t, (\mathbf{G}_{(2)}^{(N)})_{ij}^t\right) = \Phi_{ij}^{(N)}\left((\mathbf{G}_{(2)}^{(N)})_{ij}^t\right)$, hence, it only proves that $P\left(x, (\mathbf{G}_{(2)}^{(N)})_{ij}^t\right) \geq \Phi_{ij}^{(N)}(x)$. Using the second-order Taylor expansion of $\Phi_{ij}^{(N)}(x)$ around $(\mathbf{G}_{(2)}^{(N)})_{ij}^t$, we have

$$\begin{aligned} \Phi_{ij}^{(N)} = \Phi_{ij}^{(N)}\left((\mathbf{G}_{(2)}^{(N)})_{ij}^t\right) + (\Phi^{(N)})'_{ij}\left((\mathbf{G}_{(2)}^{(N)})_{ij}^t\right)\left(x - (\mathbf{G}_{(2)}^{(N)})_{ij}^t\right) \\ + \frac{1}{2}\left[\left((\mathbf{G}_{[2]}^{\neq N})^\top \mathbf{G}_{[2]}^{\neq N}\right)_{jj} + \alpha(\mathbf{L}_\mathbf{S})_{ii}\right]\left(x - (\mathbf{G}_{(2)}^{(N)})_{ij}^t\right)^2. \end{aligned} \tag{3.20}$$

Comparing (3.19) and (3.20), we see that to prove $P\left(x, (\mathbf{G}_{(2)}^{(N)})_{ij}^t\right) \geq \Phi_{ij}^{(N)}(x)$, it suffices to verify the following inequality:

$$\left((\mathbf{G}_{(2)}^{(N)})^t(\mathbf{G}_{[2]}^{\neq N})^\top \mathbf{G}_{[2]}^{\neq N} + \alpha\mathbf{D}_\mathbf{S}(\mathbf{G}_{(2)}^{(N)})^t\right)_{ij} \geq (\mathbf{G}_{(2)}^{(N)})_{ij}^t\left[\left((\mathbf{G}_{[2]}^{\neq N})^\top \mathbf{G}_{[2]}^{\neq N}\right)_{jj} + \alpha(\mathbf{L}_\mathbf{S})_{ii}\right]. \tag{3.21}$$

For the first term, since the entries of the tensor ring factors are nonnegative, we have

$$\begin{aligned} \left((\mathbf{G}_{(2)}^{(N)})^t(\mathbf{G}_{[2]}^{\neq N})^\top \mathbf{G}_{[2]}^{\neq N}\right)_{ij} &= \sum_{a=1, a\neq j}^{R_N R_1} (\mathbf{G}_{(2)}^{(N)})_{ia}^t\left((\mathbf{G}_{[2]}^{\neq N})^\top \mathbf{G}_{[2]}^{\neq N}\right)_{aj} + (\mathbf{G}_{(2)}^{(N)})_{ij}^t\left((\mathbf{G}_{[2]}^{\neq N})^\top \mathbf{G}_{[2]}^{\neq N}\right)_{jj} \\ &\geq (\mathbf{G}_{(2)}^{(N)})_{ij}^t\left((\mathbf{G}_{[2]}^{\neq N})^\top \mathbf{G}_{[2]}^{\neq N}\right)_{jj}. \end{aligned} \tag{3.22}$$

Moreover, together with the nonnegativity of $\mathbf{S}$ and $\mathbf{G}_{(2)}^{(N)}$, and the fact that $(\mathbf{D}_\mathbf{S})_{ii} = (\mathbf{L}_\mathbf{S})_{ii} + \mathbf{S}_{ii} \geq (\mathbf{L}_\mathbf{S})_{ii}$, we obtain

$$(\mathbf{D}_\mathbf{S}(\mathbf{G}_{(2)}^{(N)})^t)_{ij} = \sum_{a=1}^{I_N}(\mathbf{D}_\mathbf{S})_{ia}(\mathbf{G}_{(2)}^{(N)})_{aj}^t \geq (\mathbf{D}_\mathbf{S})_{ii}(\mathbf{G}_{(2)}^{(N)})_{ij}^t \geq (\mathbf{L}_\mathbf{S})_{ii}(\mathbf{G}_{(2)}^{(N)})_{ij}^t. \tag{3.23}$$

Hence, it follows from (3.22) and (3.23) that (3.21) holds, which implies that $P\left(x, (\mathbf{G}_{(2)}^{(N)})_{ij}^t\right) \geq \Phi_{ij}^{(N)}(x)$. Therefore, $P\left(x, (\mathbf{G}_{(2)}^{(N)})_{ij}^t\right)$ is an auxiliary function of $\Phi_{ij}^{(N)}(x)$. This completes the proof. □

For all remaining modes $k = 1, 2, \ldots, N-1$, by adopting similar construction techniques, we can obtain the auxiliary function

$$P\left(x, (\mathbf{G}_{(2)}^{(k)})_{ij}^{t}\right) = \Phi_{ij}^{(k)}\left((\mathbf{G}_{(2)}^{(k)})_{ij}^{t}\right) + (\Phi^{(k)})'_{ij}\left((\mathbf{G}_{(2)}^{(k)})_{ij}^{t}\right)\left(x - (\mathbf{G}_{(2)}^{(k)})_{ij}^{t}\right) + \frac{\left((\mathbf{G}_{(2)}^{(k)})^{t}(\mathbf{G}_{[2]}^{\neq k})^{\top}\mathbf{G}_{[2]}^{\neq k}\right)_{ij}}{(\mathbf{G}_{(2)}^{(k)})_{ij}^{t}}\left(x - (\mathbf{G}_{(2)}^{(k)})_{ij}^{t}\right)^{2}. \tag{3.24}$$

Lemma 3.4. If the similarity matrix $\mathbf{S}$ is iteratively generated according to the update rule (3.13), then the corresponding subproblem objective function (3.9) remains monotonically non-increasing during the iteration.

Proof. According to the derivation of the $\mathbf{S}$-subproblem, optimization problem (3.9) can be decomposed into a set of independent row-wise optimization problems as shown in (3.10).

For each row vector $\mathbf{S}_i$, the objective function in (3.10) is a quadratic function with respect to $\mathbf{S}_i$. Since $\gamma > 0$, the quadratic regularization term $\gamma\|\mathbf{S}_i\|_2^2$ is positive definite. Therefore, each row-wise subproblem admits a unique optimal solution. This shows that at each iteration, the updated solution $\mathbf{S}^{t+1}$ satisfies

$$\phi(\mathbf{S}^{t+1}) \leq \phi(\mathbf{S}^{t}), \tag{3.25}$$

where $\phi(\mathbf{S})$ denotes the objective function of the $\mathbf{S}$-subproblem (3.9). This implies that the objective value associated with the $\mathbf{S}$-subproblem is monotonically non-increasing during the iterative process. This completes the proof. □

Lemma 3.5. If the matrix $\mathbf{F}$ is updated by selecting the eigenvectors corresponding to the smallest eigenvalues of $\mathbf{L}_S$, then the corresponding $\mathbf{F}$-subproblem objective function (3.14) remains monotonically non-increasing during the iteration.

Proof. When all variables except $\mathbf{F}$ are fixed, the optimization problem in (3.2) reduces to the $\mathbf{F}$-subproblem (3.14).

Since $\mathbf{L}_S$ is a graph Laplacian matrix, it is symmetric and positive semidefinite. According to Lemma 3.2, the global optimum of (3.14) is achieved when the columns of $\mathbf{F}$ are chosen as the eigenvectors corresponding to the c smallest eigenvalues of $\mathbf{L}_S$, namely, $\mathbf{F} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_c]$, where $\mathbf{L}_S\mathbf{u}_i = \lambda_i\mathbf{u}_i, i = 1, 2, \ldots, c$. Hence, the updated matrix $\mathbf{F}^{t+1}$ is the global minimizer of the $\mathbf{F}$-subproblem, which yields

$$\mathrm{tr}\left((\mathbf{F}^{t+1})^{T}\mathbf{L}_S\mathbf{F}^{t+1}\right) \leq \mathrm{tr}\left((\mathbf{F}^{t})^{T}\mathbf{L}_S\mathbf{F}^{t}\right). \tag{3.26}$$

This shows that the objective value associated with the $\mathbf{F}$-subproblem is monotonically non-increasing during the iterative process. This completes the proof. □

By combining Lemmas 3.3, 3.4, and 3.5, the convergence of AGRNTR is established as follows.

Theorem 3.1. Under the update rules (3.13), (3.8), (3.6), and (3.14), the objective function of the AGRNTR model in (3.2) is monotonically non-increasing throughout the iterative optimization process.

Proof. According to Lemma 3.3, when $k = N$, $P(x, (\mathbf{G}_{(2)}^{(N)})_{ij}^t)$ is an auxiliary function of $\Phi_{ij}^{(N)}(x)$. Minimizing the auxiliary function with respect to x and applying the Karush–Kuhn–Tucker (KKT) conditions yields

$$(\mathbf{G}_{(2)}^{(N)})_{ij}^{t+1} = (\mathbf{G}_{(2)}^{(N)})_{ij}^{t} \frac{(\mathbf{X}_{[N]}\mathbf{G}_{[2]}^{\neq N} + \alpha \mathbf{S}\mathbf{G}_{(2)}^{(N)})_{ij}}{\left(\mathbf{G}_{(2)}^{(N)}(\mathbf{G}_{[2]}^{\neq N})^\top \mathbf{G}_{[2]}^{\neq N} + \alpha \mathbf{D}_S \mathbf{G}_{(2)}^{(N)}\right)_{ij}}, \tag{3.27}$$

which is exactly the update rule given in (3.6).

Similarly, for each mode $k = 1, 2, \ldots, N-1$, minimizing the corresponding auxiliary function (3.24) gives

$$(\mathbf{G}_{(2)}^{(k)})_{ij}^{t+1} = (\mathbf{G}_{(2)}^{(k)})_{ij}^{t} \frac{(\mathbf{X}_{[k]}\mathbf{G}_{[2]}^{\neq k})_{ij}}{\left(\mathbf{G}_{(2)}^{(k)}(\mathbf{G}_{[2]}^{\neq k})^\top \mathbf{G}_{[2]}^{\neq k}\right)_{ij}}, \tag{3.28}$$

which coincides with the update rule in (3.8).

Therefore, according to Lemma 3.1, the objective value associated with each updated tensor factor is non-increasing. Applying this property successively to $\mathbf{G}_{(2)}^{(1)}, \ldots, \mathbf{G}_{(2)}^{(N)}$, we obtain

$$\Delta_{\text{AGRNTR}}\left((\mathbf{G}_{(2)}^{(1)})^{t+1}, \ldots, (\mathbf{G}_{(2)}^{(N)})^{t+1}, \mathbf{S}^t, \mathbf{F}^t\right) \leq \Delta_{\text{AGRNTR}}\left((\mathbf{G}_{(2)}^{(1)})^{t}, \ldots, (\mathbf{G}_{(2)}^{(N)})^{t}, \mathbf{S}^t, \mathbf{F}^t\right), \tag{3.29}$$

where Δ_{AGRNTR} denotes the objective function defined in (3.2).

On the other hand, according to Lemma 3.4, the objective function of the $\mathbf{S}$-subproblem is monotonically non-increasing under the update rule (3.13) for fixing $\mathbf{F}$ and $\mathbf{G}_{(2)}^{(k)}$, $k = 1, \ldots, N$. Therefore,

$$\Delta_{\text{AGRNTR}}(\mathbf{G}_{(2)}^{(1)})^{t+1}, \ldots, \mathbf{G}_{(2)}^{(N)})^{t+1}, \mathbf{S}^{t+1}, \mathbf{F}^t) \leq \Delta_{\text{AGRNTR}}(\mathbf{G}_{(2)}^{(1)})^{t+1}, \ldots, \mathbf{G}_{(2)}^{(N)})^{t+1}, \mathbf{S}^t, \mathbf{F}^t). \tag{3.30}$$

In addition, by Lemma 3.5, the $\mathbf{F}$-subproblem objective function (3.14) remains monotonically non-increasing with the update of $\mathbf{F}$ obtained from (3.14) by fixing $\mathbf{S}$ and $\mathbf{G}_{(2)}^{(k)}$, $k = 1, \ldots, N$. Consequently, we obtain

$$\Delta_{\text{AGRNTR}}(\mathbf{G}_{(2)}^{(1)})^{t+1}, \ldots, \mathbf{G}_{(2)}^{(N)})^{t+1}, \mathbf{S}^{t+1}, \mathbf{F}^{t+1}) \leq \Delta_{\text{AGRNTR}}(\mathbf{G}_{(2)}^{(1)})^{t+1}, \ldots, \mathbf{G}_{(2)}^{(N)})^{t+1}, \mathbf{S}^{t+1}, \mathbf{F}^t). \tag{3.31}$$

It follows from (3.29), (3.30), and (3.31) that

$$\Delta_{\text{AGRNTR}}(\mathbf{G}_{(2)}^{(1)})^{t+1}, \ldots, \mathbf{G}_{(2)}^{(N)})^{t+1}, \mathbf{S}^{t+1}, \mathbf{F}^{t+1}) \leq \Delta_{\text{AGRNTR}}(\mathbf{G}_{(2)}^{(1)})^{t}, \ldots, \mathbf{G}_{(2)}^{(N)})^{t}, \mathbf{S}^t, \mathbf{F}^t),$$

which demonstrates that the objective function of AGRNTR is monotonically non-increasing throughout the iterative optimization process. This completes the proof. □

3.4. Complexity analysis

In this subsection, we analyze the computational complexity of solving the proposed AGRNTR model using Algorithm 1. For clarity, we assume that the mode dimensions of the tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ satisfy that $I_1 = \cdots = I_N = I$, and its TR rank $\mathbf{R} = [R_1, R_2, \ldots, R_N]$ satisfies $R_k R_{k+1} = R$ for all $k = 1, \ldots, N$. These assumptions are widely adopted in the complexity analysis of tensor ring decomposition, e.g., [25, 26], which serve solely to simplify the complexity derivation and do not

affect the algorithmic implementation in practice. Under these assumptions, the overall per-iteration computational complexity of AGRNTR is given by

$$O\Big((I^2 + \cdots + I^{N-1})R^{3/2} + (N-1)IR^2 + (N-2)R^3 + I^N R + IR + IR^2 + I^2(R+c) + I^2 c\Big).$$

Here, the terms $O((I^2 + \cdots + I^{N-1})R^{3/2})$ and $O((N-1)IR^2 + (N-2)R^3)$ correspond respectively to the computational cost of computing the subchain tensor $\mathcal{G}^{(\neq k)}$ and the product $\left(\mathbf{G}_{[2]}^{\neq k}\right)^\top \mathbf{G}_{[2]}^{\neq k}$ during updating $\mathbf{G}_{(2)}^{(k)}$ in Algorithm 1. The term $O(I^N R + IR^2 + IR)$ accounts for the computational cost of computing $\mathbf{X}_{[k]}\mathbf{G}_{[2]}^{\neq k}$, multiplying $\mathbf{G}_{(2)}^{(k)}$ by the Gram matrix $\left(\mathbf{G}_{[2]}^{\neq k}\right)^\top \mathbf{G}_{[2]}^{\neq k}$, and performing the element-wise operations in the multiplicative update. The term $O(I^2(R+c))$ is the cost of computing the similarity matrix $\mathbf{S}$ via (3.13). The term $O(I^2 c)$ corresponds to the cost of updating the embedding matrix $\mathbf{F}$ via partial eigendecomposition of $\mathbf{L}_S$.

4. Numerical experiments

In this section, we compare the proposed method with several representative approaches, including K-means, NMF [6], GNMF [10], AGNMF [8], NTR-MU [25], GNTR-MU [25], HGNTR-MU [26], GNTT-MU [22], GNTD-MU [39], and AGRNTD [20], to evaluate the effectiveness of the proposed approach. All experiments were conducted in the MATLAB R2023b environment. The workstation was equipped with an Intel Core i7-13700 processor (2.10 GHz) and 64 GB RAM. The source code is available at the following link: `https://github.com/star789123/AGRNTR`.

4.1. Datasets

In the experiments, the performance of the proposed algorithm is validated using six benchmark datasets. A brief description of each dataset is provided below, with corresponding statistical properties summarized in Table 2 and illustrative data samples depicted in Figure 1.

Table 2. Details of the six datasets.

Name	Size of objects	Number of samples	Number of categories
Yale	1024	165	15
Umist	10,304	574	20
WarpAR10P	2400	130	10
GT	6912	750	50
COIL100	3072	1440	20
Faces94	6750	1440	72

(a) Sample images from the Umist dataset

(b) Sample images from the WarpAR10P dataset

(c) Sample images from the COIL100 dataset

(d) Sample images from the Yale dataset

(e) Sample images from the GT dataset

(f) Sample images from the Faces94 dataset

Figure 1. Sample images from all image datasets.

(1) *Yale Dataset:* The Yale University dataset comprises 165 grayscale facial images collected from 15 distinct individuals, with each subject providing 11 samples under varying conditions. In our experiments, we constructed the dataset as $\mathcal{T} \in \mathbb{R}^{32\times32\times165}$.

(2) *Umist Dataset:* The Umist face dataset contains 574 grayscale face images belonging to 20 subjects, with each subject contributing between 19 and 48 samples. The images exhibit substantial pose variations, ranging from profile to frontal views. All images are resized to 112×92, forming a three-way tensor $\mathcal{T} \in \mathbb{R}^{112\times92\times574}$. This tensor representation is used for subsequent clustering experiments.

(3) *WarpAR10P Dataset:* The AR face dataset contains approximately 4000 color and grayscale face images from 100 subjects, including various lighting conditions, expression changes, and occlusion scenarios. Based on this database, we selected 10 classes, with 13 grayscale images per class, that have been aligned and normalized following the standard Warped AR preprocessing protocol. All images are uniformly resized to a resolution of 60×40. Thus, the WarpAR10P dataset used in this study can be represented as a three-way tensor $\mathcal{T} \in \mathbb{R}^{60\times40\times130}$.

(4) *GT Dataset:* The GT face dataset consists of 750 color facial images from 50 subjects, with each subject providing 15 images at a resolution of 640×480. The images in this database feature complex backgrounds and exhibit significant variations in pose, expression, lighting, and scale. To construct the data structure required for our experiments, we uniformly resized all images to dimensions $48 \times$

48×3. Therefore, the GT dataset used in this study can be represented as a fourth-order tensor $\mathcal{T} \in \mathbb{R}^{48\times48\times3\times750}$.

(5) *COIL-100 Dataset:* The COIL-100 dataset comprises 7200 color images of 100 distinct objects, each represented by 72 views captured from varying angles. Each image has a resolution of 128×128×3 pixels. In this study, the first 20 categories were selected, and all images were resized to $32 \times 32 \times 3$ pixels. The resulting subset, referred to as the COIL-100 dataset, forms a fourth-order tensor $\mathcal{T} \in \mathbb{R}^{32\times32\times3\times1440}$.

(6) *Faces94 Dataset:* The Faces94 database consists of facial expression images captured under 153 fixed camera setups, totaling 3060 color images representing 20 distinct expressions. In this study, we selected the images of the first 72 individuals and resized them to a resolution of $50 \times 45 \times 3$ pixels. Consequently, the processed dataset is organized as a four-dimensional tensor $\boldsymbol{T} \in \mathbb{R}^{50\times45\times3\times1440}$ for subsequent analysis.

4.2. Comparison methods

We employed the following state-of-the-art clustering methods for comparison with our proposed approach:

(1) **K-means**: A classic unsupervised clustering algorithm that alternates between assigning samples to the nearest cluster centers and updating these centers, aiming to minimize intra-cluster variance and maximize inter-cluster separation.

(2) **NMF** [6]: Nonnegative matrix factorization decomposes a nonnegative data matrix into two lower-rank nonnegative matrices, enabling effective feature extraction and parts-based representation learning.

(3) **GNMF** [10]: By incorporating graph regularization into NMF, GNMF preserves local geometric structures of data while performing nonnegative factorization, achieving structure-aware feature learning.

(4) **AGNMF** [8]: AGNMF integrates adaptive graph learning with nonnegative matrix factorization, enabling joint optimization of global data patterns and adaptive local manifold preservation for effective data representation.

(5) **NTR-MU** [25]: Nonnegative tensor ring decomposition via multiplicative updates iteratively optimizes core tensors within the tensor ring structure, ensuring nonnegativity and reducing reconstruction error.

(6) **GNTR-MU** [25]: This extends NTR-MU by integrating graph regularization to preserve sample manifold relationships, leading to more discriminative multidimensional representations.

(7) **HGNTR-MU** [26]: This introduces hypergraph regularization into NTR-MU to capture higher-order relationships among samples, enhancing the representation of complex structural dependencies.

(8) **GNTT-MU** [22]: A graph-regularized nonnegative tensor train model that iteratively optimizes TT cores through multiplicative updates while incorporating manifold structure via graph constraints.

(9) **GNTD-MU** [39]: A graph-regularized nonnegative Tucker decomposition that updates factor matrices and the core tensor through multiplicative updates under graph regularization to preserve local similarity structures.

(10) **AGRNTD** [20]: Adaptive graph regularized nonnegative Tucker decomposition jointly optimizes graph, core, and factors via multiplicative updates for manifold preservation.

4.3. Evaluation metrics

To quantitatively evaluate clustering performance, this experiment employs accuracy (ACC) and normalized mutual information (NMI) as evaluation metrics.

Accuracy (ACC) measures the proportion of correctly classified samples after optimal label matching using the Hungarian algorithm. It is calculated as:

$$\mathrm{ACC} = \frac{1}{N}\sum_{i=1}^{N}\delta\left(\mathrm{map}(r_i), l_i\right), \tag{4.1}$$

where N is the total number of samples, r_i is the predicted cluster label, l_i is the ground-truth label, and $\delta(\cdot,\cdot)$ is the Kronecker delta function. The function $\mathrm{map}(\cdot)$ performs optimal label matching between predicted clusters and ground-truth classes using the Hungarian algorithm to maximize the number of correct assignments, implemented by maximizing the sum of the confusion matrix entries through optimal assignment.

Normalized mutual information (NMI) is used to quantify the agreement between the predicted clustering assignments and the ground-truth labels. In this work, we report the geometric-mean normalized variant.

The mutual information between the predicted clusters (R) and the true labels (L) is computed as

$$I(R,L) = \sum_{r\in R}\sum_{l\in L} P(r,l)\log\left(\frac{P(r,l)}{P(r)P(l)}\right), \tag{4.2}$$

where $P(r,l)$ represents the joint probability of cluster r and label l, and $P(r)$ and $P(l)$ are their marginal distributions.

The geometric-mean normalized NMI is then defined as

$$\mathrm{NMI} = \frac{I(R,L)}{\sqrt{H(R)\cdot H(L)}}, \tag{4.3}$$

where $H(R)$ and $H(L)$ denote the entropies of the predicted clusters and the ground-truth labels, respectively. The entropy is calculated as

$$H(X) = -\sum_{x\in X} P(x)\log P(x). \tag{4.4}$$

4.4. Clustering results

This subsection conducts comprehensive clustering experiments across six real-world datasets to evaluate the effectiveness and superiority of the proposed AGRNTR method. The parameter selection for the methods involved in the experiment is as follows: all comparison methods extract k-dimensional feature representations from the data, where k denotes the number of categories in each dataset. To ensure fair comparison, each experiment utilizes the maximum available sample size for the corresponding dataset. For traditional matrix-based methods (including NMF, GNMF, and K-means clustering), the original tensor data is unfolded into matrix form via reshaping operations in MATLAB. In the GNTD method, the Tucker rank is set to $R_1 = R_2 = 10$. The core tensor dimensions are configured as follows: For 3rd-order tensors, the third-dimension rank $R_3 = k$ is retained, resulting in

a core tensor dimension of $10 \times 10 \times k$. For 4th-order tensors, the ranks of each dimension are set to $R_1 = R_2 = R_3 = 10$ and $R_4 = k$, yielding a core tensor dimension of $10 \times 10 \times 10 \times k$. Regarding the GNTT method, the rank parameters are empirically configured as follows: For 3rd-order tensors, we set $R_1 = 1, R_2 = 10, R_3 = k$, corresponding to a core tensor dimension of $1 \times 10 \times k$. For 4th-order tensors, the rank configuration is $R_1 = 1$, $R_2 = 10$, $R_3 = 10$, $R_4 = k$, resulting in a core tensor dimension of $1 \times 10 \times 10 \times k$. For methods based on tensor ring decomposition, the rank of the terminal core tensor is constrained to be an integer multiple of k. Specifically, we are given a tensor ring rank vector $R = [R_1, R_2, \ldots, R_N]$ satisfying $R_1 R_N = k$. This setting originates from the input tensor construction, all samples are stacked along specific patterns in their respective sample dimensions, necessitating that the boundary core ranks R_1 and R_N align with the number of classes.

Beyond the rank setting for the proposed AGRNTR method, three key hyperparameters (i.e, the similarity matrix regularization coefficient γ, graph regularization parameter α, and Laplace regularization parameter λ) are empirically tuned within the ranges specified in Section 4.5 (i.e., the parameter selection section). The optimal value of each parameter is determined by Figure 3 and Figure 4. In addition, all other parameters for each comparison method adopt the optimal values reported in their respective original papers.

To reduce randomness and enhance statistical reliability, experiments on each dataset were repeated ten times. Tables 3 and 4 list the ACC, NMI, and their corresponding standard deviations, with optimal results highlighted in bold. Based on these experimental results, the following conclusions can be drawn:

- As shown in Tables 3 and 4, the proposed AGRNTR method outperforms all benchmark methods and achieves the best performance across all evaluation datasets. For instance, on the WarpAR10P dataset, AGRNTR achieves an ACC of 33.85%, representing an improvement of approximately 3.62% over the second-best method GNTR-MU (30.23%). While on the COIL100 dataset, its performance advantage is even more pronounced, with an ACC of 73.90%, nearly 3.19% higher than the best-performing method, HGNTR-MU (70.71%). These improvements directly and powerfully demonstrate the superiority of AGRNTR in clustering tasks.
- The performance advantage of models incorporating graph information over graph-free approaches underscores the critical role of graph structural data. This is evidenced by the consistent superiority of GNTR-MU over NTR-MU, indicating that leveraging graph structures to capture intrinsic data relationships enhances the discriminative power of low-dimensional representations, thereby improving clustering outcomes. Furthermore, the proposed AGRNTR method outperforms all baseline methods that rely on fixed similarity graphs, such as GNTR-MU, across all datasets. The primary reason for this improvement lies in the adaptive similarity learning mechanism introduced by AGRNTR, which dynamically optimizes the similarity matrix $\mathbf{S}$. By adjusting the similarity graph adaptively, this mechanism more accurately uncovers latent relationships among samples, thereby learning more discriminative feature representations and ultimately achieving superior clustering performance.
- It follows from Tables 3 and 4 that the proposed AGRNTR method achieves superior performance over AGRNTD and AGNMF on both ACC and NMI on all test datasets. This result fully demonstrates that the adaptive graph-regularization strategy adopted by AGRNTR can more effectively uncover the underlying structure of the data, thereby achieving higher-quality clustering results.

- For Tables 3 and 4, tensor-based methods outperform matrix-based methods in most cases. This advantage stems from the ability of tensor methods to directly model the raw multidimensional structure of the data, whereas matrix-based methods necessitate flattening high-dimensional data into a two-dimensional form. This flattening inevitably destroys multidimensional relationships. By preserving and leveraging these inherent structural dependencies, tensor methods capture richer intrinsic information, thereby achieving superior clustering performance.
- Experimental results demonstrate that AGRNTR and GNTR generally outperform GNTT and GNTD in overall performance, indicating that TR decomposition holds advantages over tensor train (TT) decomposition and Tucker decomposition in characterizing and uncovering the intrinsic structure of high-order data. This phenomenon indicates that TR decomposition can more flexibly model complex intermodal associations, thereby preserving richer structural information and generating more discriminative feature representations. This further validates its advantages and effectiveness in modeling high-order data.

Table 3. ACC (Mean ± Standard Deviation) of eleven clustering algorithms across six datasets.

Method	Yale	Umist	WarpAR10P	GT	COIL100	Faces94
K-means	38.85 ± 2.29	41.05 ± 1.65	21.62 ± 2.36	66.84 ± 2.65	70.39 ± 5.10	76.45 ± 2.03
NMF	39.64 ± 3.16	39.76 ± 2.06	28.23 ± 3.72	65.55 ± 1.38	66.32 ± 3.63	73.90 ± 2.11
GNMF	39.21 ± 3.28	40.96 ± 0.88	23.54 ± 1.55	66.23 ± 3.66	70.24 ± 3.27	73.33 ± 1.68
AGNMF	38.97 ± 2.41	41.10 ± 2.11	26.77 ± 2.87	67.00 ± 2.04	68.51 ± 3.10	77.88 ± 2.60
NTR-MU	39.09 ± 1.32	44.11 ± 3.09	30.23 ± 4.73	65.51 ± 1.35	67.28 ± 4.91	79.06 ± 3.39
GNTR-MU	43.39 ± 2.31	44.55 ± 1.98	30.23 ± 4.25	66.47 ± 2.30	70.61 ± 2.06	78.95 ± 2.40
HGNTR-MU	42.00 ± 3.73	44.49 ± 2.23	32.31 ± 5.10	66.61 ± 2.39	70.71 ± 3.26	76.97 ± 2.71
GNTT-MU	41.33 ± 2.45	39.27 ± 1.99	28.38 ± 5.22	66.17 ± 1.60	69.30 ± 3.87	71.15 ± 2.49
GNTD-MU	43.15 ± 2.06	40.56 ± 1.99	29.38 ± 3.99	64.97 ± 1.63	65.97 ± 4.37	72.31 ± 2.91
AGRNTD	42.67 ± 2.06	39.56 ± 1.27	28.77 ± 1.16	63.40 ± 2.70	63.08 ± 1.87	71.60 ± 4.43
AGRNTR	**43.88 ± 3.79**	**47.60 ± 2.75**	**33.85 ± 4.05**	**67.60 ± 2.28**	**73.90 ± 3.81**	**80.42 ± 3.07**

Table 4. NMI (Mean ± Standard Deviation) of eleven clustering algorithms across six datasets.

Method	Yale	Umist	WarpAR10P	GT	COIL100	Faces94
K-means	44.65 ± 2.59	60.92 ± 1.54	18.81 ± 3.39	87.10 ± 0.63	81.57 ± 2.43	92.97 ± 0.69
NMF	44.10 ± 1.48	58.99 ± 1.63	28.86 ± 5.35	86.58 ± 0.43	78.83 ± 1.64	90.89 ± 0.85
GNMF	44.23 ± 2.25	59.32 ± 1.15	23.30 ± 2.26	86.94 ± 0.76	81.15 ± 1.46	91.06 ± 0.90
AGNMF	42.42 ± 1.46	56.27 ± 1.64	20.84 ± 3.17	84.23 ± 0.76	74.29 ± 2.39	91.15 ± 1.03
NTR-MU	44.33 ± 1.76	63.04 ± 2.72	30.93 ± 6.64	86.92 ± 0.56	78.95 ± 2.04	93.17 ± 1.32
GNTR-MU	48.01 ± 2.79	63.28 ± 1.93	31.80 ± 5.92	87.31 ± 0.69	80.86 ± 1.37	93.69 ± 0.92
HGNTR-MU	46.83 ± 3.30	63.15 ± 1.92	32.89 ± 5.29	87.35 ± 0.46	80.95 ± 1.96	92.96 ± 0.85
GNTT-MU	46.02 ± 1.54	57.65 ± 2.29	28.67 ± 8.49	86.93 ± 0.48	80.43 ± 2.17	88.54 ± 1.39
GNTD-MU	47.68 ± 1.33	58.78 ± 2.12	28.86 ± 5.06	86.26 ± 0.62	78.75 ± 2.03	90.21 ± 1.84
AGRNTD	45.79 ± 1.70	58.71 ± 1.05	28.47 ± 1.55	82.28 ± 1.29	75.70 ± 1.34	89.07 ± 2.05
AGRNTR	**48.56 ± 3.00**	**65.19 ± 2.38**	**36.45 ± 5.86**	**87.52 ± 0.40**	**82.25 ± 2.17**	**93.82 ± 0.93**

4.5. Parameter selection

As described above, the AGRNTR model involves four hyperparameters: TR rank R, similarity-regularization coefficient γ, graph-regularization parameter α, and Laplace regularization parameter λ. For each dataset, we fix a pair (R_1, R_N) satisfying the experimental setting specified in Section 4.4. Rank sensitivity is then evaluated in two settings: first, the intermediate ranks are set as $R_2 = \cdots = R_{N-1} = r$, where r varies from 2 to 5; second, the values of R_1 and R_N are exchanged. The remaining three parameters are analyzed through grid searches on each dataset within the following ranges: $\lambda \in \{10^{-3}, 10^{-2}, 10^{-1}, 10^{1}, 10^{2}, 10^{3}\}$, $\gamma \in \{10^{-2}, 10^{-1}, 10^{1}, 10^{2}, 10^{3}\}$, and $\alpha \in \{0.001, 0.002, 0.003, 0.004, 0.005, 0.006\}$. Figure 2 presents the results of the TR rank sensitivity analysis, while Figures 3 and 4 display the average ACC and NMI achieved under different combinations of λ, γ, and α. Based on these results, the following observations can be concluded.

- The results in Figure 2 indicate that AGRNTR exhibits relatively stable clustering performance across the tested TR-rank settings. With R_1 and R_N fixed, varying r from 2 to 5 does not consistently enhance clustering performance, while smaller values of r generally achieve competitive results. Adjusting R_1 and R_N under the constraint $R_1 R_N = k$ leads to moderate, dataset-dependent performance variations. Overall, AGRNTR is relatively insensitive to the TR-rank settings considered in this experiment, and smaller values of r generally provide a favorable trade-off between clustering performance and model complexity.
- As shown in Figures 3 and 4, the clustering performance of AGRNTR varies only mildly across the entire parameter grid, with the gap between the best and worst accuracy generally remaining within a limited range. This observation suggests that AGRNTR exhibits stable behavior and low sensitivity to parameter selection, reflecting both its effectiveness and robustness.
- The parameter λ governs the strength of the Laplacian regularization term, which enforces consistency between the learned feature matrix $\mathbf{F}$ and the similarity matrix $\mathbf{S}$. Typically, when λ is within the range $\{10^{-2}, 10^{-1}, 10^{1}, 10^{2}\}$, relatively good clustering performance can be achieved. When λ is set too small, the Laplacian constraint becomes insufficient, hindering $\mathbf{F}$ from adequately capturing the intrinsic relationships among samples. In contrast, overly large values of λ impose excessive smoothness, restricting the flexibility of $\mathbf{F}$ and diminishing its ability to discriminate between clusters.
- The parameter γ controls the Frobenius norm regularization on the similarity matrix $\mathbf{S}$, thereby influencing its size and sparsity. Empirical results indicate that higher accuracy is typically obtained when γ falls within the range $\{10^{-1}, 10^{1}, 10^{2}\}$. Extremely small values of γ result in weak regularization, which may yield unstable or noisy similarity estimates, whereas excessively large values overly penalize $\mathbf{S}$, suppressing meaningful similarity patterns and degrading clustering performance.
- The graph regularization coefficient α encourages the learning of locally smooth feature representations. Experimental results show that α values between 0.002 and 0.004 consistently produce stable and competitive performance across different datasets, with $\alpha = 0.003$ yielding the best results on datasets such as Yale and COIL100. When α is too small, the smoothing effect is insufficient and the learned features become more sensitive to noise; when α is too large, excessive smoothing reduces inter-sample discrimination and weakens clustering quality.

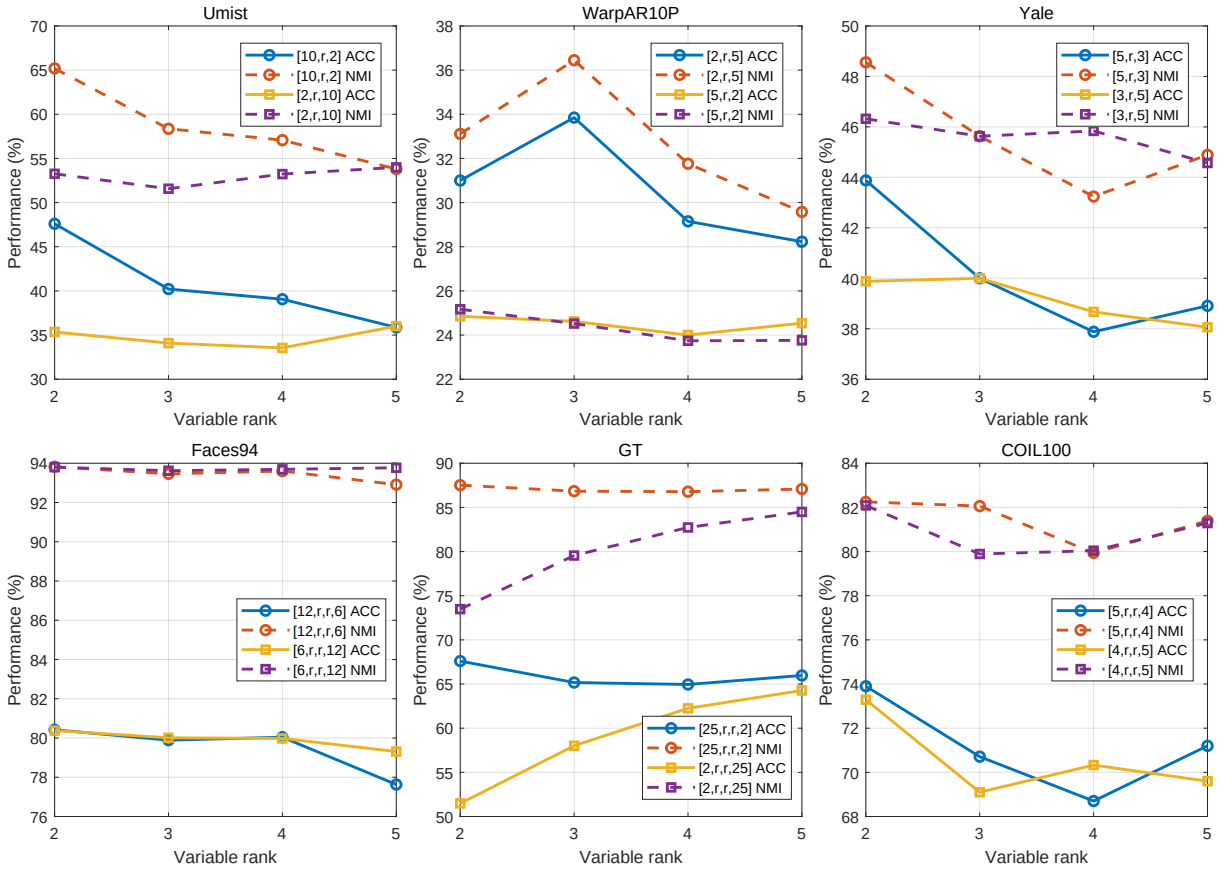
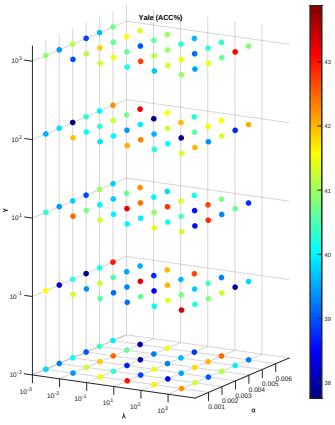
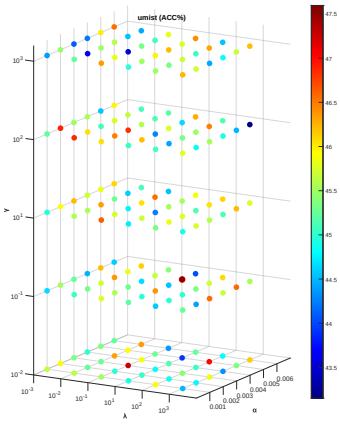


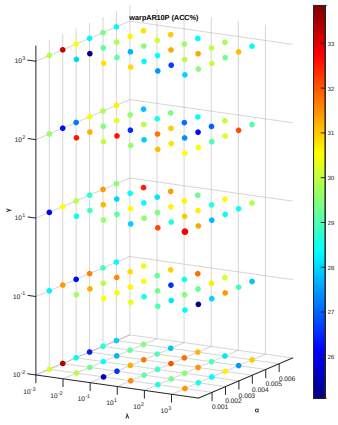
Figure 2. Clustering performance of AGRNTR under different TR rank configurations.



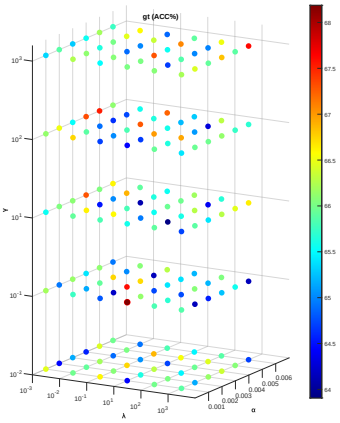
(a) Yale



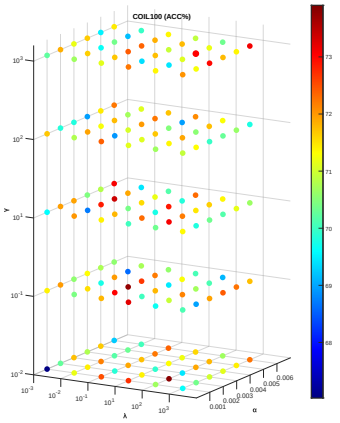
(b) Umist



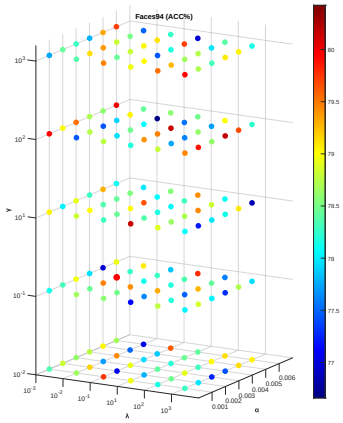
(c) WarpAR10P



(d) GT

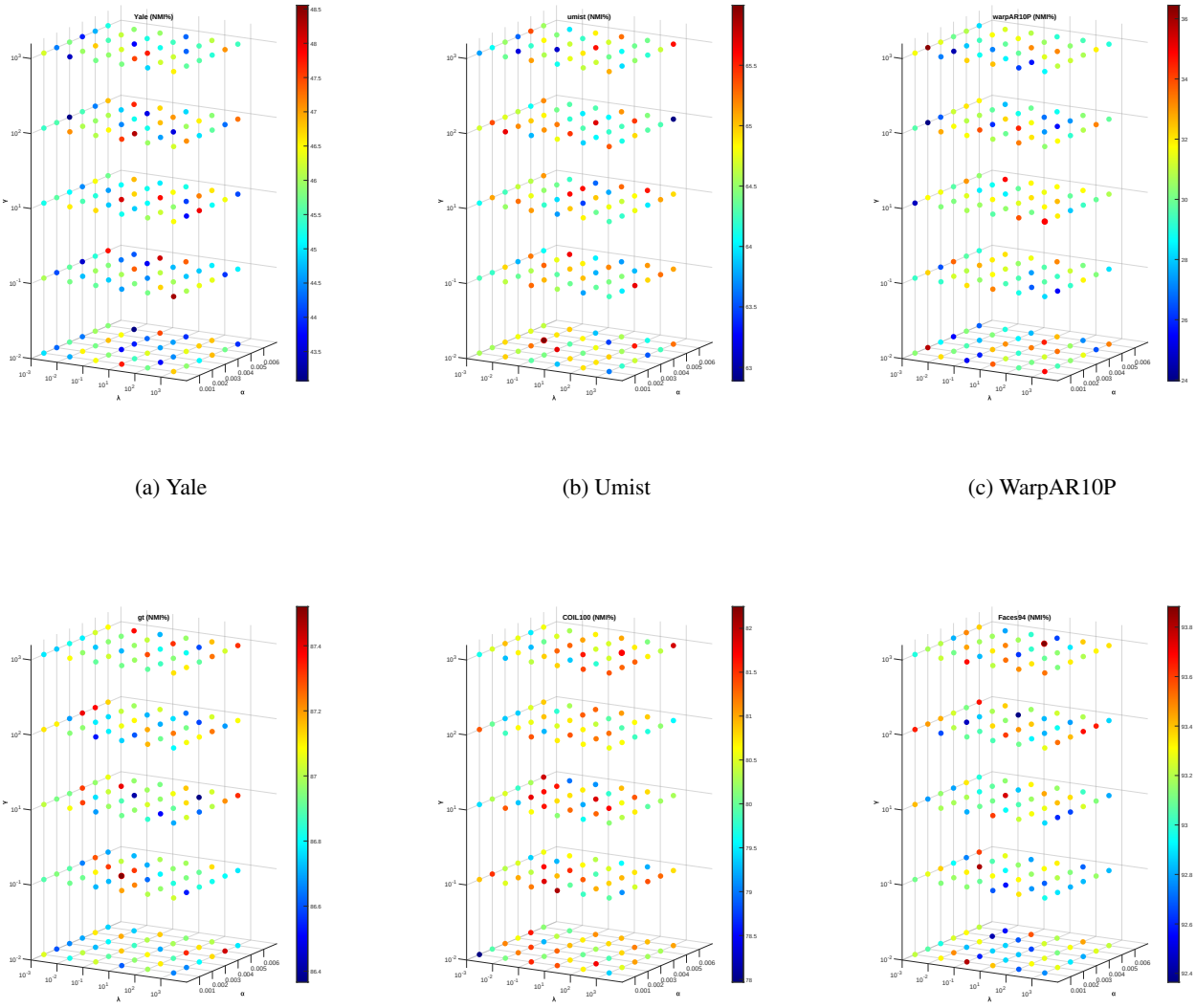


(e) COIL100



(f) Faces94

Figure 3. ACC of the proposed AGRNTR method across all six datasets relative to parameters λ, γ, and α.



(a) Yale (b) Umist (c) WarpAR10P

(d) GT (e) COIL100 (f) Faces94

Figure 4. NMI of the proposed AGRNTR method across all six datasets relative to parameters λ, γ, and α.

4.6. Comparison between the fixed graph and adaptive graph

To evaluate the effectiveness of adaptive graph learning, we compare the proposed AGRNTR with its fixed-graph variant, in which the learned similarity matrix $\mathbf{S}$ is replaced with a predefined KNN graph constructed from the raw data, while all other optimization procedures and model parameters remain unchanged to ensure a fair comparison. The results are reported in Table 5, with the best results highlighted in bold.

AGRNTR consistently outperforms its fixed KNN variant in terms of both ACC and NMI across all six datasets. Notably, ACC improves by 4.73% and 5.56% on the Yale and COIL100 datasets,

respectively. These improvements demonstrate that jointly learning the similarity matrix with the tensor ring representation enables AGRNTR to capture the underlying data structure more effectively. Unlike the predefined KNN graph, which is constructed solely from the original feature space, the adaptive graph is updated based on the learned representations, thereby providing more appropriate structural constraints for representation learning and clustering.

Table 5. Comparison of clustering performance using adaptive and fixed KNN graphs.

Dataset	AGRNTR		Fixed KNN	
	ACC (%)	NMI (%)	ACC (%)	NMI (%)
Yale	**43.88 ± 3.79**	**48.56 ± 3.00**	39.15 ± 4.77	44.49 ± 0.81
Umist	**47.60 ± 2.75**	**65.19 ± 2.38**	45.59 ± 2.64	64.17 ± 1.59
WarpAR10P	**33.85 ± 4.05**	**36.45 ± 5.86**	30.15 ± 5.34	30.32 ± 7.27
GT	**67.60 ± 2.28**	**87.52 ± 0.40**	63.99 ± 1.47	84.44 ± 0.71
Faces94	**80.42 ± 3.07**	**93.82 ± 0.93**	77.92 ± 2.22	93.17 ± 0.71
COIL100	**73.90 ± 3.81**	**82.25 ± 2.17**	68.34 ± 5.16	78.73 ± 3.30

4.7. Visualization

To qualitatively examine the learned representations, we visualize the original features and the representations obtained using the fixed KNN graph and AGRNTR on selected classes of the COIL100 dataset through t-distributed stochastic neighbor embedding (t-SNE), as shown in Figure 5. Compared with the original features, both graph-based representations exhibit clearer clustering structures. Relative to the fixed KNN graph, AGRNTR provides relatively compact within-class distributions and clear separation among most classes. These results provide qualitative evidence that adaptive graph learning can improve the discriminative structure of the learned representation.

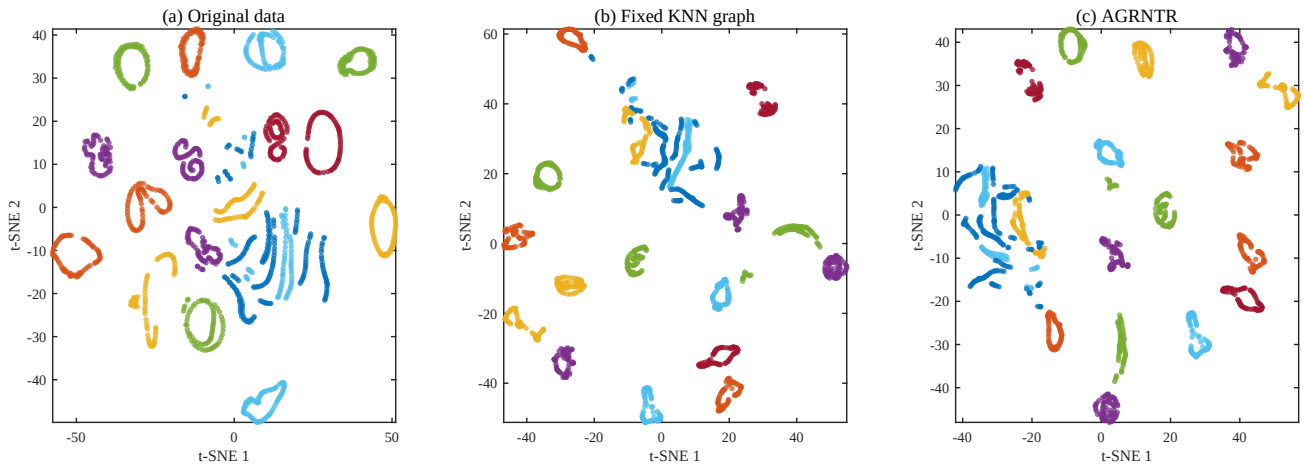


Figure 5. t-SNE visualization of the original data and the representations obtained using the fixed KNN graph and AGRNTR on the COIL100 subset.

4.8. Ablation study

To further examine the contributions of the key components of AGRNTR, we conduct an ablation study. Note that removing the second term would decouple the adaptive similarity matrix from the tensor-ring representation. Consequently, the learning of the tensor-ring factors would reduce to standard NTR, while the remaining graph-related terms would no longer affect the learned representation. Therefore, rather than removing this coupling term alone, we consider two meaningful ablation variants. In the first ablation setting, the Frobenius norm regularization term $\gamma\|\mathbf{S}\|_F^2$ is discarded, while the Laplacian regularization term $2\lambda\,\mathrm{tr}(\mathbf{F}^\top\mathbf{L_S}\mathbf{F})$ is preserved. This variant, referred to as AGRNTR-1, is designed to assess the influence of scale control on the similarity matrix $\mathbf{S}$ and to examine whether the absence of magnitude regularization degrades the quality of the learned kernel tensor. In the second ablation setting, the Laplacian regularization is removed, and only the Frobenius norm constraint on $\mathbf{S}$ is maintained. This variant, denoted as AGRNTR-2, aims to evaluate the contribution of graph-structural regularization in capturing the intrinsic manifold structure during kernel tensor learning. By comparing the kernel tensor obtained in this case with that of the complete model, we evaluate the extent to which the geometric structure encoded in $\mathbf{S}$ facilitates capturing intrinsic data correlations, and conduct evaluations across all benchmark datasets.

The corresponding results are reported in Tables 6 and 7. It can be observed that the proposed AGRNTR consistently outperforms its ablation variants, including AGRNTR-1 and AGRNTR-2, in terms of both ACC and NMI. For example, on the Umist dataset, AGRNTR attains an ACC of 47.60, which is markedly higher than the values achieved by AGRNTR-1 (45.30) and AGRNTR-2 (45.17). This comparison indicates that the joint optimization of the kernel tensor and the adaptive similarity matrix is critical for improving clustering performance.

Moreover, on most datasets, such as Yale and WarpAR10P, AGRNTR demonstrates stable superiority over ablation models that incorporate only a single component. This further confirms that the complementary regularization imposed by the parameters γ and λ enables more accurate modeling of the underlying geometric structure of the data. Overall, the ablation study clearly verifies the necessity of the complete AGRNTR framework. By jointly optimizing the kernel tensor and the adaptively learned similarity matrix under the coordinated constraints of γ and λ, the proposed model effectively captures latent structural information and enhances the discriminative capability of the learned representations.

Table 6. Results of ablation experiments' ACC.

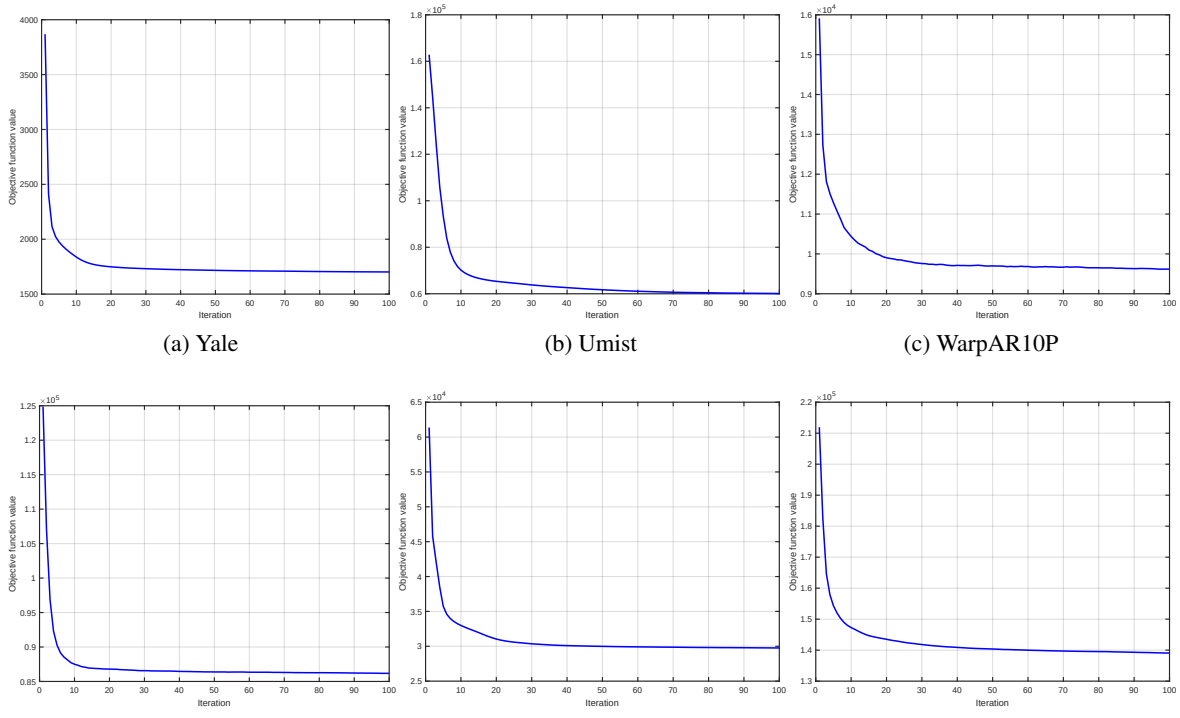
ACC	Yale	Umist	WarpAR10P	COIL100	GT	Faces94
AGRNTR-1	42.87	45.30	33.46	71.58	67.49	79.86
AGRNTR-2	42.61	45.17	33.77	72.11	67.52	79.85
AGRNTR	43.88	47.60	33.85	73.90	67.60	80.42

Table 7. Results of ablation experiments' NMI.

NMI	Yale	Umist	WarpAR10P	COIL100	GT	Faces94
AGRNTR-1	47.59	64.38	35.58	81.34	87.26	93.35
AGRNTR-2	48.48	64.33	36.03	81.39	87.37	93.14
AGRNTR	48.56	65.19	36.45	82.25	87.52	93.81

4.9. Convergence

As detailed in Section 3.3, the convergence properties of AGRNTR were theoretically established, showing that the objective function decreases monotonically. To empirically validate this convergence property, we evaluated its behavior on six image datasets. The corresponding convergence curves are presented in Figure 6. As shown, the objective function consistently exhibits a strictly non-increasing trend across all datasets and rapidly stabilizes within roughly 100 iterations, which is consistent with the theoretical results.



(a) Yale (b) Umist (c) WarpAR10P

(d) GT (e) COIL100 (f) Faces94

Figure 6. Convergence curves of AGRNTR on six image datasets.

5. Conclusions

This paper proposes an adaptive graph-regularized nonnegative tensor ring decomposition (AGRNTR) model for tensor data clustering. Unlike conventional graph-regularized NTR methods that rely on fixed predefined similarity graphs, AGRNTR integrates adaptive graph learning into the tensor ring decomposition framework to jointly optimize tensor representations and the similarity matrix.

This joint learning mechanism retains both the multilinear structure and intrinsic geometric manifold of tensor data, which facilitates more precise manifold approximation and remarkable clustering improvement. To solve this model, we develop an efficient optimization algorithm by combining block coordinate descent (BCD) with multiplicative update rules, followed by a rigorous convergence proof and computational complexity analysis. Extensive experimental results conducted on real-world tensor datasets demonstrate that AGRNTR consistently achieves superior clustering performance in terms of both clustering accuracy and normalized mutual information compared with representative matrix-based and tensor-based decomposition methods. These results further validate the effectiveness and advantages of jointly learning an adaptive graph structure within the tensor ring decomposition framework.

Author contributions

Xinyu Yao: Writing—original draft, Validation, Resources, Software, Formal analysis, Data curation, Visualization. Guimin Liu: Writing—review and editing, Methodology, Data curation, Software, Supervision.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgements

The work is supported by the National Natural Science Foundation of China (Grant No. 12361081).

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. A. K. Jain, Data clustering: 50 years beyond k-means, *Pattern Recogn. Lett.,* **31** (2010), 651–666. https://doi.org/10.1016/j.patrec.2009.09.011
2. J. Duan, Y. Zhao, J. Wang, L. Dang, Tensor-ring based multi-view contrastive graph clustering with high-quality pseudo-labels, *Knowledge-Based Syst.,* **332** (2026), 114847. https://doi.org/10.1016/j.knosys.2025.114847
3. M. Li, J. Cao, Q. Sun, Fuzzy Clustering Means Integrated Multi-Level Learning for Contrastive Multi-View Clustering, *IEEE Access,* **14** (2026), 48831–48841. https://doi.org/10.1109/ACCESS.2026.3678396
4. N. E. D. Salem, S. Hussein, Data dimensional reduction and principal components analysis, *Procedia Comput. Sci.,* **163** (2019), 292–299. https://doi.org/10.1016/j.procs.2019.12.111
5. K. Meng, S. Li, Fast one-step multi-view clustering via low-rank singular value decomposition, *Neurocomputing,* **678** (2026), 133154. https://doi.org/10.1016/j.neucom.2026.133154

6. D. D. Lee, H. S. Seung, Learning the parts of objects by non-negative matrix factorization, *Nature,* **401** (1999), 788–791. https://doi.org/10.1038/44565
7. M. Wan, Y. Zhao, J. Yin, C. Sun, G. Yang, Robust locality regularized non-negative matrix factorization with structure preservation for image classification, *Pattern Recog.,* **171** (2026), 112241. https://doi.org/10.1016/j.patcog.2025.112241
8. L. Zhang, Z. Liu, J. Pu, B. Song, Adaptive graph regularized nonnegative matrix factorization for data representation, *Appl. Intell.,* **50** (2020), 438–447. https://doi.org/10.1007/s10489-019-01539-9
9. K. Liang, W. Wang, L. Xiao, W. Liang, Q. Liu, Heterogeneous information disentangling via low-rank splitting non-negative matrix factorization with adaptive graph learning, *Inform. Sciences,* **733** (2026), 122943. https://doi.org/10.1016/j.ins.2025.122943
10. D. Cai, X. He, J. Han, T. S. Huang, Graph regularized nonnegative matrix factorization for data representation, *IEEE T. Pattern Anal.*, **33** (2011), 1548–1560. https://doi.org/10.1109/TPAMI.2010.231
11. Z. Xing, Y. Ma, X. Yang, F. Nie, Graph regularized nonnegative matrix factorization with label discrimination for data clustering, *Neurocomputing,* **440** (2021), 297–309. https://doi.org/10.1016/j.neucom.2021.01.064
12. Y. Yang, J. Li, Y. Li, Z. Wang, Hyperspectral and multispectral image fusion guided by latent diffusion and spectral expert models, *Knowledge-Based Syst.,* (2026), 115319. https://doi.org/10.1016/j.knosys.2026.115319
13. P. Gu, H. Hu, G. Xu, Modeling multi-behavior sequence via HyperGRU contrastive network for micro-video recommendation, *Knowledge-Based Syst.,* **295** (2024), 111841. https://doi.org/10.1016/j.knosys.2024.111841
14. K. Yang, R. Yu, B. Guo, S. Zhen, Is multi-level data enhancement helpful for knowledge graph? A new perspective on multimodal fusion, *Knowledge-Based Syst,,* **301** (2024), 112285. https://doi.org/10.1016/j.knosys.2024.112285
15. Y. D. Kim, S. Choi, Nonnegative Tucker decomposition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2007, 1–8. https://doi.org/10.1109/CVPR.2007.383405
16. W. Jing, L. Lu, Q. Liu, Z. Chen, Multi-graph regularized non-negative Tucker decomposition and its semi-supervised extension for image clustering, *Expert Syst. Appl.,* **299** (2026), 130005. https://doi.org/10.1016/j.eswa.2025.130005
17. G. Liu, R. Zhao, B. Zheng. F. Yang, Auto-weighted multiple graph regularized non-negative tensor Tucker decomposition for clustering, *J. Sci. Comput.,* **102** (2025), 86. https://doi.org/10.1007/s10915-025-02817-0
18. Q. Liu, L. Lu. Z. Chen, Non-negative Tucker decomposition with graph regularization and smooth constraint for clustering, *Pattern Recogn.,* **148** (2024), 110207. https://doi.org/10.1016/j.patcog.2023.110207

19. W. Jing, L. Lu, Q. Liu, Graph regularized discriminative nonnegative Tucker decomposition for tensor data representation, *Appl. Intell.,* **53** (2023), 23864–23882. https://doi.org/10.1007/s10489-023-04738-7

20. D. Chen, G. Zhou, Y. Qiu, Y. Yu, Adaptive graph regularized non-negative Tucker decomposition for multiway dimensionality reduction, *Multimed. Tools Appl.,* **83** (2023), 9647–9668. https://doi.org/10.1007/s11042-023-15622-4

21. I. V. Oseledets, Tensor-train decomposition, *SIAM J. Sci. Comput.,* **33** (2011), 2295–2317. https://doi.org/10.1137/090752286

22. S. E. Sofuoglu, S. Aviyente, Graph-regularized tensor-train decomposition, in *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* IEEE, 2020, 3912–3916. https://doi.org/10.1109/ICASSP40776.2020.9054032

23. H. Dai, Y. Huang, G. Zhou, Clustering analysis based on Hyper-graph regularized non-negative tensor train decomposition, *Comput. Eng.*, **49** (2023), 81–89. https://doi.org/10.19678/j.issn.1000-3428.0064740

24. Q. Zhao, G. Zhou, S. Xie, L. Zhang, A. Cichocki, Tensor Ring Decomposition, arXiv preprint arXiv:1606.05535, 2016. https://doi.org/10.48550/arXiv.1606.05535

25. Y. Yu, G. Zhou, N. Zheng, Y. Qiu, S. Xie, Q. Zhao, Graph-regularized non-negative tensor-ring decomposition for multiway representation learning, *IEEE T. Cybernetics,* **53** (2023), 3114–3127. https://doi.org/10.1109/TCYB.2022.3157133

26. X. Zhao, Y. Yu, G. Zhou, Q. Zhao, W. Sun, Fast hypergraph regularized nonnegative tensor ring decomposition based on low-rank approximation, *Appl. Intell.,* **52** (2022), 17684–17707. https://doi.org/10.1007/s10489-022-03346-1

27. C. Liu, S. Wu, R. Li, D. Jiang. H. S. Wong, Self-Supervised graph completion for incomplete multi-view clustering, *IEEE T. Knowl. Data En.,* **35** (2023), 9394–9406. https://doi.org/10.1109/TKDE.2023.3238416

28. C. Liu, R. Li, H. Che, M. F. Leung, S. Wu, Z. Yu, et al., Latent Structure-Aware View Recovery for Incomplete Multi-View Clustering, *IEEE T. Knowl. Data En.,* **36** (2024), 8655–8669. https://doi.org/10.1109/TKDE.2024.3445992

29. C. Liu, R. Li, H. Che, M. F. Leung, S. Wu, Z. Yu, et al.,, Beyond Euclidean Structures: Collaborative Topological Graph Learning for Multiview Clustering, *IEEE T. Neur. Net. Lear.,* **36** (2025), 10606–10618. https://doi.org/10.1109/TNNLS.2024.3489585

30. T. G. Kolda, B. W. Bader, Tensor Decompositions and Applications, *SIAM Rev.,* **51** (2009), 455–500. https://doi.org/10.1137/07070111X

31. Y. Chen, W. He, N. Yokoya, T. Z. Huang, X. L. Zhao, Nonlocal tensor-ring decomposition for hyperspectral Image denoising, *IEEE T. Geosci. Remote,* **58** (2019), 1348–1362. https://doi.org/10.1109/TGRS.2019.2946050

32. Y. Li, C. Y. Hsieh, R. Lu, X. Gong, X. Wang, P. Li, et al., An adaptive graph learning method for automated molecular interactions and properties predictions, *Nat. Mach. Intell.,* **4** (2022), 645–651. https://doi.org/10.1038/s42256-022-00501-8

33. F. Nie, X. Wang, H. Huang, Clustering and Projected Clustering with Adaptive Neighbors, in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD),* ACM, 2014, 977–986. https://doi.org/10.1145/2623330.2623726

34. K. Fan, On a theorem of Weyl concerning eigenvalues of linear transformations. I, *P. Natl. A. Sci. USA.,* **35** (1949), 652–655. https://doi.org/10.1073/pnas.35.11.652

35. J. Shi, J. Malik, Normalized cuts and Image segmentation, *IEEE T. Pattern Anal.,* **22** (2000), 888–905. https://doi.org/10.1109/34.868688

36. M. Belkin. P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, *Neural Comput.,* **15** (2003), 1373–1396. https://doi.org/10.1162/089976603321780317

37. D. D. Lee, H. S. Seung, Algorithms for non-negative matrix factorization, in *Advances in Neural Inf. Process. Syst. (NeurIPS),* Cambridge, MA, USA: MIT Press, 2000, 535–541. https://dl.acm.org/doi/10.5555/3008751.3008829

38. G. H. Golub, C. F. Van Loan, *Matrix computations*, 4 Eds., Baltimore, MD: Johns Hopkins University Press, 2013. https://epubs.siam.org/doi/book/10.1137/1.9781421407944

39. Y. Qiu, G. Zhou, Y. Wang, Y. Zhang, S. Xie, A generalized graph regularized non-negative Tucker decomposition framework for tensor data representation, *IEEE T. Cybernetics,* **52** (2022), 594–607. https://doi.org/10.1109/TCYB.2020.2979344

AIMS Press