

# Uniform Tensor Clustering by Jointly Exploring Sample Affinities of Various Orders

Hongmin Cai<sup>1</sup>, Senior Member, IEEE, Fei Qi<sup>2</sup>, Junyu Li<sup>3</sup>, Yu Hu<sup>4</sup>, Bin Hu<sup>5</sup>, Fellow, IEEE, Yue Zhang<sup>6</sup>, and Yiu-Ming Cheung<sup>7</sup>, Fellow, IEEE

**Abstract**—Traditional clustering methods rely on pairwise affinity to divide samples into different subgroups. However, high-dimensional small-sample (HDLSS) data are affected by the concentration effects, rendering traditional pairwise metrics unable to accurately describe relationships between samples, leading to suboptimal clustering results. This article advances the proposition of employing high-order affinities to characterize multiple sample relationships as a strategic means to circumnavigate the concentration effects. We establish a nexus between different order affinities by constructing specialized decomposable high-order affinities, thereby formulating a uniform mathematical framework. Building upon this insight, a novel clustering method named uniform tensor clustering (UTC) is proposed, which learns a consensus low-dimensional embedding for clustering by the synergistic exploitation of multiple-order affinities. Extensive experiments on synthetic and real-world datasets demonstrate two findings: 1) high-order affinities are better suited for characterizing sample relationships in complex data and 2) reasonable use of different order affinities can enhance clustering effectiveness, especially in handling high-dimensional data.

**Index Terms**—Clustering, fusing affinity, high-order affinity, spectral graph, tensor.

Manuscript received 10 April 2023; revised 5 November 2023, 1 April 2024, and 21 June 2024; accepted 29 July 2024. Date of publication 14 August 2024; date of current version 5 May 2025. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFE0112200; in part by the National Natural Science Foundation of China (NSFC) under Grant U21A20520, Grant 62325204, and Grant 62172112; in part by the Key-Area Research and Development Program of Guangzhou City under Grant 202206030009; in part by the NSFC/Research Grants Council (RGC) Joint Research Scheme under Grant N\_HKBU214/21; in part by the General Research Fund of RGC under Grant 12201321, Grant 12202622, and Grant 12201323; and in part by the RGC Senior Research Fellow Scheme under Grant SRFS2324-2S02. (Corresponding authors: Hongmin Cai; Yiu-Ming Cheung.)

Hongmin Cai, Junyu Li, and Yu Hu are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: hmcai@scut.edu.cn; chunyulee018@foxmail.com; jasonscut@outlook.com).

Fei Qi is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China, and also with the School of Data Science and Information Engineering, Guizhou Minzu University, Guiyang 550029, China (e-mail: fqiscut@foxmail.com).

Bin Hu is with the School of Medical Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: bh@bit.edu.cn).

Yue Zhang is with the School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou 510665, China (e-mail: zhangyue@gpnu.edu.cn).

Yiu-Ming Cheung is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, SAR, China (e-mail: ymc@comp.hkbu.edu.hk).

Digital Object Identifier 10.1109/TNNLS.2024.3439545

## NOMENCLATURE

$n \in \mathbb{Z}^+$	Number of samples.
$k \in \mathbb{Z}^+$	Number of clusters.
$l \in \mathbb{Z}^+$	Order of affinity.
$S_2 \in \mathbb{R}^{n \times n}$	Dyadic affinity matrix.
$L_2 \in \mathbb{R}^{n \times n}$	Normalized dyadic affinity matrix.
$\hat{L}_2 \in \mathbb{R}^{n \times n}$	Laplacian matrix.
$H \in \mathbb{R}^{n \times k}$	Indicator matrix.
$V \in \mathbb{R}^{n \times k}$	Embedding matrix for clustering.
$S_3 \in \mathbb{R}^{n^2 \times n}$	Unfolded triadic affinity.
$S_4 \in \mathbb{R}^{n^2 \times n^2}$	Unfolded tetradic affinity.
$L_3 \in \mathbb{R}^{n^2 \times n}$	Normalized unfolding triadic affinity.
$L_4 \in \mathbb{R}^{n^2 \times n^2}$	Normalized unfolded tetradic affinity.
$S_3 \in \mathbb{R}^{n \times n \times n}$	Triadic tensor affinity.
$S_4 \in \mathbb{R}^{n \times n \times n \times n}$	Tetradic tensor affinity.
$\mathcal{L}_3 \in \mathbb{R}^{n \times n \times n}$	Normalized triadic affinity tensor.
$\mathcal{L}_4 \in \mathbb{R}^{n \times n \times n \times n}$	Normalized tetradic affinity tensor.
$\text{Tr}(\cdot)$	Trace operation.
$\odot$	Hadamard product.
$*$	Khatri–Rao product.
$\otimes$	Kronecker product.
$\otimes_k$	$K$ -mode product.

## I. INTRODUCTION

CLUSTERING, as one of the key techniques in the field of artificial intelligence, aims to partition unlabeled data into different clusters based on sample relationships [1], [2]. Traditional clustering methods, such as  $K$ -means [3] and spectral clustering (SC) [4], [5], [6], have been widely studied, and diverse variants have been designed to cater to various needs. Despite recent advancements in clustering techniques, existing methods still face challenges when dealing with high-dimensional small-sample (HDLSS) data. For instance, in the field of bioinformatics, many cells are damaged during the tissue isolation process for sequencing due to technical limitations, resulting in high-dimensionality single-cell sequencing data with a small sample size [7], [8]. Similar situations arise in computer vision [9], [10], natural language processing [11], and other domains.

The critical challenge in accurately clustering HDLSS data is the concentration effect [12], [13], [14]. When the data dimensionality is high, distance metrics between samples become ineffective. This failure in distance measurement

prevents an accurate description of sample relationships, making it difficult to obtain sufficient prior information for guiding data partitioning [15], [16], [17]. Earlier efforts [18] focused on learning an adaptive affinity matrix for clustering through a data-driven approach. For example, Nie et al. [19] proposed a model to dynamically learn affinity matrix by assigning adaptive neighborhoods to each sample. Fu et al. [20] learned optimal projected representations to capture an effective affinity graph within multiple subspaces. However, a poor-quality initial affinity matrix can have an adverse effect on the subsequent similarity learning step. To acquire high-quality affinity matrices, some researchers proposed measuring sample similarities from different kernel spaces to obtain multiple affinity matrices [21], [22]. For instance, Liu [23] introduced a minimax strategy to solve the kernel coefficient and the consensus affinity matrix simultaneously. It is essential to highlight that affinity matrices built from various kernel spaces still rely on pairwise relationships between samples. As a result, these methods are still affected by the concentration effect and do not fundamentally solve this problem.

Recent studies [24], [25] have shown that utilizing sophisticated sample affinity proves advantageous when dealing with complex data characterized by high dimensionality and small sample sizes. Many studies propose constructing high-order affinities by measuring multiple sample relationships to mitigate the concentration effect. For example, Zhou et al. [26] extended graph-based spectral methods to high-order affinity by hypergraphs and used hypergraph embeddings for clustering. Agarwal et al. [27] proposed maintaining a weighted graph to approximate hypergraph and achieving graph partitioning by spectral analysis. Inspired by adaptive affinity learning, Zhang et al. [28] first proposed a method to learn dynamic hypergraphs for clustering. However, in these methods, high-order affinity ultimately relies on an approximate representation of pairwise affinity matrices, limiting the potential of high-order information.

Tensor representation has recently provided a new perspective for high-order affinity-based clustering. Li et al. [29] introduced a tensor representation of hypergraphs, using tensors to preserve multiple samples' natural relationships. Ghoshdastidar and Dukkupati [30] utilized a symmetric tensor to represent  $K$ -order affinity and realized clustering by solving the multilinear singular value decomposition problem. However, these methods are limited for utilizing specific order affinity, restricting their ability to reveal underlying data structures. Peng et al. [31] proposed integrating tensor similarity and pairwise similarity (IPS2), which integrates tetradic and pairwise affinity for clustering, offering a novel solution to this problem. Following this investigation, this article establishes a mathematical connection between pairwise relationships and high-order affinity, leading to the proposal of the unified tensor clustering model. The uniform tensor clustering (UTC) model integrates different order affinities to comprehensively describe data relationships from various perspectives, ultimately generating consensus embeddings for clustering. Extensive experiments demonstrate the outstanding performance of this approach in handling HDLSS data.

The main advantages of the method proposed in this article are summarized as follows.

- 1) We establish a connection between dyadic affinity and decomposable triadic/tetradic tensor affinities defined via Khatri–Rao/Kronecker products. The corresponding indecomposable affinities are designed to provide the proximity of multiple samples, thus supplementing the pairwise affinity.
- 2) An UTC method is developed by integrating different order affinities into a uniform framework. The UTC jointly learns a low-dimensional embedding based on various order affinities. The UTC is elegantly formulated and works on affinities of arbitrary order.
- 3) Extensive experiments on HDLSS data demonstrate that the UTC achieved superior performance compared with baseline methods. In addition, the experiments show that using high-order affinities to characterize the sample spatial distribution can improve clustering performance, especially in small sample sizes.

## II. NOTATIONS AND PRELIMINARIES

### A. Notations

This section will provide explanations and definitions for the mathematical symbols and relevant concepts used in this article. Lowercase letters denote scalars, while bold lowercase letters represent vectors, such as  $\mathbf{v} \in \mathbb{R}^n$ . Bold uppercase letters indicate matrices, for example,  $\mathbf{X} \in \mathbb{R}^{n \times m}$ , and bold calligraphy letters signify tensors, like  $\mathcal{S} \in \mathbb{R}^{n \times m \times l}$ . The expressions  $\mathcal{S}(:, :, i)$ ,  $\mathcal{S}(:, i, :)$ , and  $\mathcal{S}(i, :, :)$  correspond to the  $i$ th frontal, lateral, and horizontal slices of a third-order tensor  $\mathcal{S}$ , respectively. The notation  $\mathcal{S}(:, :, i)$  can be succinctly as  $\mathcal{S}^{(i)}$ . In order to facilitate the analysis of high-order tensors, corresponding unfolding operations are commonly employed to restructure them into matrices. For example, the third-order and fourth-order tensor unfolding operations are as follows.

*Definition 1 (Unfolding Third-Order Tensor):* Let  $\mathcal{S}_3 \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  be a third-order tensor. It can unfold to an  $n_1 n_3 \times n_2$  matrix  $\mathcal{S}_3$  as follows:

$$\mathcal{S}_3 = \text{unfold}(\mathcal{S}_3) = \begin{bmatrix} \mathcal{S}_3^{(1)} \\ \mathcal{S}_3^{(2)} \\ \vdots \\ \mathcal{S}_3^{(n_3)} \end{bmatrix}. \quad (1)$$

*Definition 2 (Unfolding Fourth-Order Tensor):* Let  $\mathcal{S}_4 \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times n_4}$  be a fourth-order tensor. This tensor can unfold to an  $n_1 n_2 \times n_3 n_4$  matrix  $\mathcal{S}_4$ , with its  $(r, s)$ th entry given by

$$\mathcal{S}_{4_{rs}} = \mathcal{S}_{4_{ijkl}} \quad (2)$$

with  $r = n_1(j - 1) + i$  and  $s = n_3(l - 1) + k$ .

Two matrix multiplications are considered in this article, i.e., Kronecker product and Khatri–Rao product [32].

*Definition 3 (Kronecker Product):* The Kronecker product of two matrices  $\mathcal{S}_a \in \mathbb{R}^{m_1 \times n_1}$  and  $\mathcal{S}_b \in \mathbb{R}^{m_2 \times n_2}$  is defined

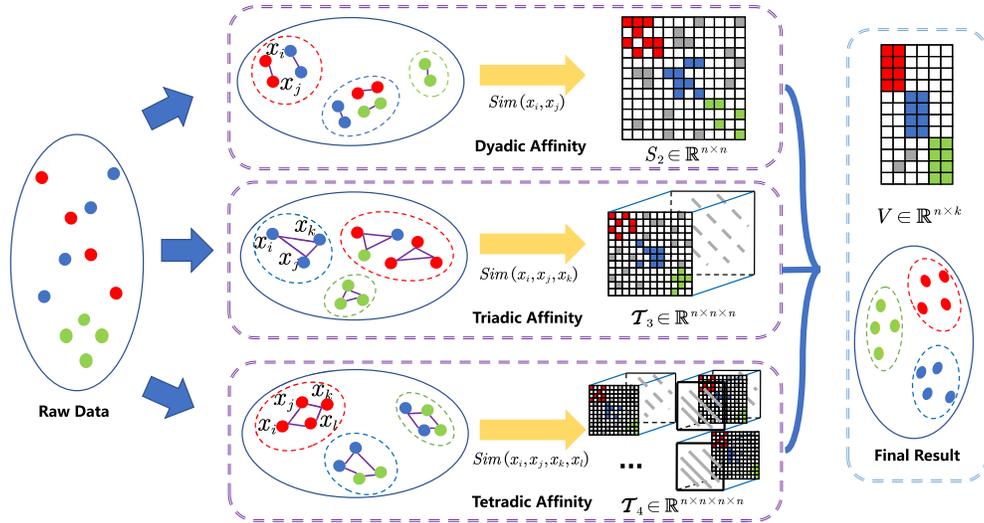


Fig. 1. Framework of the UTC involves jointly optimizing dyadic, triadic, and tetradic affinities among multiple samples to learn a consensus embedding. This approach allows for a comprehensive analysis of the data relationships, going beyond pairwise affinity alone.

as follows:

$$\mathbf{S}_a \otimes \mathbf{S}_b = \begin{bmatrix} \mathbf{S}_{a_{11}} \mathbf{S}_b & \cdots & \mathbf{S}_{a_{1n_1}} \mathbf{S}_b \\ \vdots & \ddots & \vdots \\ \mathbf{S}_{a_{m_1 1}} \mathbf{S}_b & \cdots & \mathbf{S}_{a_{m_1 n_1}} \mathbf{S}_b \end{bmatrix} \in \mathbb{R}^{m_1 m_2 \times n_1 n_2}. \quad (3)$$

**Definition 4 (Khatri–Rao Product):** The Khatri–Rao product of two matrices  $\mathbf{S}_a \in \mathbb{R}^{m_1 \times n}$  and  $\mathbf{S}_b \in \mathbb{R}^{m_2 \times n}$  is defined as the matrix

$$\mathbf{S}_a * \mathbf{S}_b = [\mathbf{S}_{a_{\cdot 1}} \otimes \mathbf{S}_{b_{\cdot 1}} \quad \cdots \quad \mathbf{S}_{a_{\cdot n}} \otimes \mathbf{S}_{b_{\cdot n}}] \in \mathbb{R}^{m_1 m_2 \times n}. \quad (4)$$

*K-mode product*, as an operation between a tensor and a matrix, is defined as follows.

**Definition 5 (K-Mode Product):** The *K*-mode product between a *m*th order tensor  $\mathcal{L} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_m}$  and a matrix  $\mathbf{V} \in \mathbb{R}^{n_k \times p}$ , denoted by  $\mathcal{L} \otimes_k \mathbf{V} \in \mathbb{R}^{n_1 \times \cdots \times n_{k-1} \times p \times n_{k+1} \times \cdots \times n_m}$ , with

$$(\mathcal{L} \otimes_k \mathbf{V})_{i_1, \dots, i_{k-1}, j, i_{k+1}, \dots, i_m} = \sum_{i_k=1}^{n_k} \mathcal{L}_{i_1, \dots, i_{k-1}, i_k, i_{k+1}, \dots, i_m} \mathbf{V}_{j i_k}. \quad (5)$$

For convenience, we summarize the frequently used notations and definitions in the Nomenclature.

### B. SC With Pairwise Affinity

Given a dataset with  $n$  samples, each having  $m$  features, denoted by  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n] \in \mathbb{R}^{m \times n}$ . Classical SC method [33] seeks a low-dimensional embedding by minimizing the graph cut cost. This method constructs a similarity matrix  $\mathbf{S}_2 \in \mathbb{R}^{n \times n}$  based on samples pairwise relationships, where each entry  $S_{2_{ij}}$  measures the similarity between  $\mathbf{x}_i$  and  $\mathbf{x}_j$ . Next, the Laplacian matrix for spectral analysis can be defined as  $\hat{\mathbf{L}} = \mathbf{I} - \mathbf{L}_2$ , where  $\mathbf{L}_2$  is the normalized affinity matrix given by  $\mathbf{L}_2 = \mathbf{D}_2^{-1/2} \mathbf{S}_2 \mathbf{D}_2^{-1/2}$ , and  $\mathbf{D}_{2_{ii}} = \sum_j S_{2_{ij}}$  represents the degree matrix of  $\mathbf{S}_2$ . Ultimately, SC aims to solve the following optimization problem:

$$\begin{aligned} \min_{\mathbf{V}} \text{tr}(\mathbf{V}^T \hat{\mathbf{L}} \mathbf{V}) \\ \text{s.t. } \mathbf{V}^T \mathbf{V} = \mathbf{I}. \end{aligned} \quad (6)$$

Typically, the optimization process of this objective function can be viewed as searching for the graph-cut strategy with minimal cost. It can also be expressed as a maximization problem via normalized pairwise affinity  $\mathbf{L}_2$  [34]. By using the tensor *K*-mode product, the SC in (6) can be rewritten as follows:

$$\begin{aligned} \max_{\mathbf{V}} \sum_{j=1}^k \mathbf{L}_2 \otimes_2 \mathbf{v}_j \otimes_1 \mathbf{v}_j \\ \text{s.t. } \mathbf{V}^T \mathbf{V} = \mathbf{I} \end{aligned} \quad (7)$$

where  $\mathbf{v}_j$  is the  $j$ th column of the embedding  $\mathbf{V}$ .

### C. Tensor SC With Tetradic Affinity

Tensor SC builds upon traditional spectral methods by incorporating high-order affinity to more accurately characterize sample relationships. IPS2 [31], the representative of these methods, introduces the use of tetradic affinities to gauge relationships between sample pairs, aiming for more precise analysis of complex data and improved clustering performance. This method utilizes dyadic affinities to construct decomposable tetradic affinities, showcasing the relationship between the two affinities represented by matrices and tensors, respectively. Furthermore, it builds a fourth-order tensor SC model based on the tetradic affinity as follows:

$$\begin{aligned} \max_{\mathbf{V}} \sum_{j=1}^k \mathcal{L}_4 \otimes_4 \mathbf{v}_j \otimes_3 \mathbf{v}_j \otimes_2 \mathbf{v}_j \otimes_1 \mathbf{v}_j \\ \text{s.t. } \mathbf{V}^T \mathbf{V} = \mathbf{I} \end{aligned} \quad (8)$$

where  $\mathcal{L}_4$  denotes the normalized tetradic affinity. This method demonstrates that tetradic affinities capture complex data structures that traditional similarities may overlook, and combining these two affinities can further enhance clustering performance. While IPS2 demonstrates the association between tetradic and dyadic affinities, and how tetradic affinity can describe additional data relationships, it lacks generalizability to other high-order affinity.

### III. PROPOSED METHOD

This section introduces the UTC. Section III-A provides a detailed introduction to high-order affinities. Section III-B presents a unified representation formulation for handling multi-order affinities. The UTC model is presented in Section III-C, while Section III-D discusses the optimization techniques utilized in UTC. To provide a visual summary of the UTC approach, we include an illustrative overview in Fig. 1.

#### A. Characterizing Sample's Affinity via High-Order Tensor Affinity

Emerging studies [27], [31] indicate that high-order affinity can reveal complex data relationships that traditional pairwise affinity fails to represent. However, there is limited research exploring the connections between different order affinities. This article introduces a specialized high-order affinity constructing strategy to explore the mathematical connections between different order affinities. Third-order and fourth-order affinities are discussed as representatives of odd and even orders.

1) *Decomposable Triadic and Tetradic Affinities*: We construct decomposable high-order affinity based on pairwise relationships to explore the connections between different order affinities. IPS2 [31] introduced decomposable tetradic affinity  $\mathcal{S}_4$  derived from traditional pairwise affinity, where each element of  $\mathcal{S}_4$  can be defined as follows:

$$\mathcal{S}_{4_{ijkl}} = \mathcal{S}_{2_{ik}} \mathcal{S}_{2_{jl}}, \quad i, j, k, l \in n. \quad (9)$$

It can be easily demonstrated that the relationship between the tetradic affinity tensor  $\mathcal{S}_4$  and the pairwise affinity matrix  $\mathcal{S}_2$  satisfies:  $\mathcal{S}_4 = \text{fold}(\mathcal{S}_2 \otimes \mathcal{S}_2)$ . The normalized tetradic affinity can be denoted by  $\mathcal{L}_4 = \text{fold}(\mathcal{D}_4^{-1/2} \mathcal{S}_4 \mathcal{D}_4^{-1/2})$ , where  $\mathcal{D}_4$  is the degree matrix of  $\mathcal{S}_4$  with the diagonal elements being  $\mathcal{D}_{4_{ii}} = \sum_j \mathcal{S}_{4_{ij}}$ . As shown in Theorem 1, the normalization process does not affect the relationship between normalized tetradic affinity  $\mathcal{L}_4$  and dyadic affinity  $\mathcal{L}_2$ .

*Theorem 1 (The Decomposable Fourth-Order Tensor)*: Let  $\mathcal{L}_2$  and  $\mathcal{L}_4 = \text{fold}(\mathcal{L}_4)$  be the normalized affinity matrix and fourth-order affinity tensor with the unfolding form  $\mathcal{L}_4$ , respectively. Then, we have the following equality:

$$\mathcal{L}_4 = \mathcal{L}_2 \otimes \mathcal{L}_2. \quad (10)$$

The relationship between eigenvectors  $\mathbf{v}$  of  $\mathcal{L}_2$  and eigenvectors  $\hat{\mathbf{v}}$  of unfolded  $\mathcal{L}_4$  satisfies the Kronecker product

$$\hat{\mathbf{v}} = \mathbf{v} \otimes \mathbf{v}. \quad (11)$$

The decomposable triadic affinity  $\mathcal{S}_3 \in \mathbb{R}^{n \times n \times n}$  can also be constructed as the product of two pairwise affinities

$$\mathcal{S}_{3_{ijk}} = \mathcal{S}_{2_{ij}} \mathcal{S}_{2_{kj}}, \quad i, j, k \in n. \quad (12)$$

Under this definition, the relationship between these two types of affinities satisfies Theorem 2, as follows.

*Theorem 2*: Given a decomposable third-order tensor affinity  $\mathcal{S}_3$  defined in (12),  $\mathcal{S}_3 = \mathcal{S}_2 * \mathcal{S}_2$ , where  $\mathcal{S}_3$  is the matrix unfolded by the tensor  $\mathcal{S}_3$ .

*Proof*: By Definition 1, the unfolded matrix  $\mathcal{S}_3 \in \mathbb{R}^{n^2 \times n}$  can be written

$$\begin{aligned} \mathcal{S}_3 &= \begin{bmatrix} \mathcal{S}_{3_{111}} & \cdots & \mathcal{S}_{3_{1n1}} \\ \vdots & \ddots & \vdots \\ \mathcal{S}_{3_{n1n}} & \cdots & \mathcal{S}_{3_{nnn}} \end{bmatrix} \\ &= \begin{bmatrix} \mathcal{S}_{2_{11}} \mathcal{S}_{2_{11}} & \cdots & \mathcal{S}_{2_{1n}} \mathcal{S}_{2_{1n}} \\ \vdots & \ddots & \vdots \\ \mathcal{S}_{2_{11}} \mathcal{S}_{2_{n1}} & \cdots & \mathcal{S}_{2_{1n}} \mathcal{S}_{2_{nn}} \\ \vdots & \ddots & \vdots \\ \mathcal{S}_{2_{n1}} \mathcal{S}_{2_{11}} & \cdots & \mathcal{S}_{2_{nn}} \mathcal{S}_{2_{1n}} \\ \vdots & \ddots & \vdots \\ \mathcal{S}_{2_{n1}} \mathcal{S}_{2_{n1}} & \cdots & \mathcal{S}_{2_{nn}} \mathcal{S}_{2_{nn}} \end{bmatrix} \\ &= \begin{bmatrix} \mathcal{S}_{2_{11}} \mathcal{S}_{2_{\cdot 1}} & \cdots & \mathcal{S}_{2_{1n}} \mathcal{S}_{2_{\cdot n}} \\ \vdots & \ddots & \vdots \\ \mathcal{S}_{2_{n1}} \mathcal{S}_{2_{\cdot 1}} & \cdots & \mathcal{S}_{2_{nn}} \mathcal{S}_{2_{\cdot n}} \end{bmatrix} \\ &= [\mathcal{S}_{2_{\cdot 1}} \otimes \mathcal{S}_{2_{\cdot 1}} \quad \cdots \quad \mathcal{S}_{2_{\cdot n}} \otimes \mathcal{S}_{2_{\cdot n}}] \\ &= \mathcal{S}_2 * \mathcal{S}_2. \end{aligned} \quad (13)$$

□

The normalized form of triadic affinity tensor  $\mathcal{S}_3$  can be defined as  $\mathcal{L}_3 = \text{fold}(\mathcal{D}_{3_1}^{-1/2} \mathcal{S}_3 \mathcal{D}_{3_2}^{-1/2})$ . Diagonal matrices  $\mathcal{D}_{3_1}$  and  $\mathcal{D}_{3_2}$  can be calculated as follows:

$$\begin{cases} \mathcal{D}_{3_1} = \text{diag}(\sqrt{\mathbf{d}}) \otimes \text{diag}(\sqrt{\mathbf{d}}) = \mathcal{D}_2 \otimes \mathcal{D}_2 \\ \mathcal{D}_{3_2} = \text{diag}(\sqrt{\mathbf{d}}) \odot \text{diag}(\sqrt{\mathbf{d}}) = \mathcal{D}_2 \odot \mathcal{D}_2 \end{cases} \quad (14)$$

where  $\mathbf{d}_j = \sum_{i=1}^n \mathcal{S}_{3_{ij}} = \mathcal{D}_{2_{jj}} \mathcal{D}_{2_{jj}}$  and  $\mathbf{d} \in \mathbb{R}^{n \times 1}$ ,  $\odot$  denotes the Hadamard product.

The relationship between the normalized decomposable triadic affinity tensor and pairwise affinity matrix is still preserved, as demonstrated in Theorem 3.

*Theorem 3 (The Decomposable Third-Order Tensor)*: Let a matrix  $\mathcal{S}_2$  be a pairwise affinity matrix with  $\mathcal{L}_2$  being its normalized affinity matrix. The decomposable triadic affinity  $\mathcal{S}_3$ , and its normalized tensor  $\mathcal{L}_3$  are obtained as above. We have

$$\mathcal{L}_3 = \mathcal{L}_2 * \mathcal{L}_2 \quad (15)$$

where  $\mathcal{L}_3$  is the unfolded form of the tensor  $\mathcal{L}_3$ .

*Proof*: By the definitions of the normalized affinity matrix  $\mathcal{L}_2$  and  $\mathcal{L}_3$ , we have

$$\begin{aligned} \mathcal{L}_3 &= \mathcal{D}_{3_1}^{-1/2} \mathcal{S}_3 \mathcal{D}_{3_2}^{-1/2} \\ &= (\mathcal{D}_2 \otimes \mathcal{D}_2)^{-1/2} (\mathcal{S}_2 * \mathcal{S}_2) (\mathcal{D}_2 \odot \mathcal{D}_2)^{-1/2} \\ &= \left( \mathcal{D}_2^{-1/2} \mathcal{S}_2 \mathcal{D}_2^{-1/2} \right) * \left( \mathcal{D}_2^{-1/2} \mathcal{S}_2 \mathcal{D}_2^{-1/2} \right) \\ &= \mathcal{L}_2 * \mathcal{L}_2. \end{aligned} \quad (16)$$

□

2) *Nondecomposable High-Order Affinities*: Now, we have presented a unification of mathematical formalism on decomposable dyadic, triadic, and tetradic affinities by three algebraic products (arithmetic, Khatri-Rao, and Kronecker products). However, a closer examination of the decomposable triadic

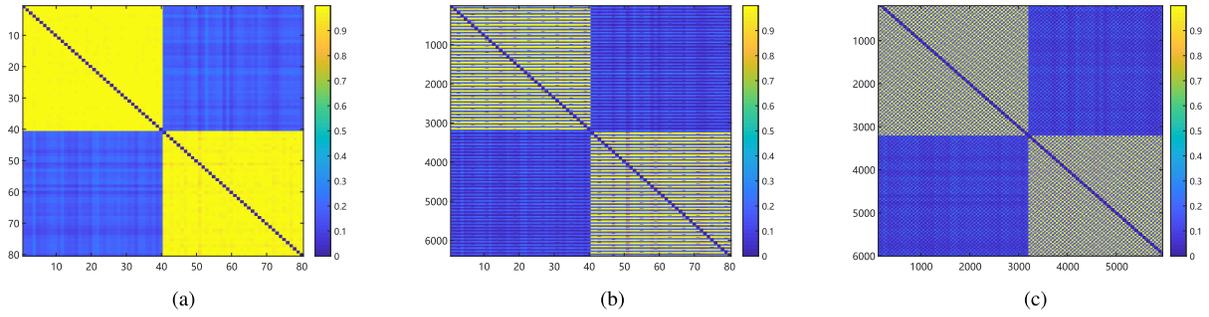


Fig. 2. Heatmap of (a) dyadic affinity, (b) decomposable triadic, and (c) tetradic affinity matrices on a synthetic dataset drawn from a normal distribution. There are obvious block structures in the heatmap on different orders of unfolded affinity, indicating that the decomposable affinities of different orders are consistent in terms of performance.

affinity  $\mathcal{S}_3$  reveals an interesting observation: the sum of  $\mathcal{S}_3$ 's lateral slices equals the traditional definition of high-order similarity  $S_2^2$  [35]. This suggests that the traditional high-order affinity derived from pairwise relationships does not introduce any additional valuable information. Furthermore, as shown in Fig. 2, the heatmaps of dyadic, decomposable triadic, and tetradic affinities also demonstrate the similarity between decomposable high-order affinity and pairwise relationships. Fig. 2 further highlights that decomposable high-order affinities do not provide supplementary information for pairwise relationships. This article proposes introducing nondecomposable high-order affinity to address this limitation.

The nondecomposable triadic affinity  $\mathcal{S}_3 \in \mathbb{R}^{n \times n \times n}$  is defined as follows:

$$\mathcal{S}_{3_{ijk}} = 1 - \frac{\langle (\mathbf{x}_i - \mathbf{x}_j), (\mathbf{x}_k - \mathbf{x}_j) \rangle}{\|\mathbf{x}_i - \mathbf{x}_j\|_2 \|\mathbf{x}_k - \mathbf{x}_j\|_2} \quad (17)$$

where  $i, j, k \in n$ . The entry  $\mathcal{S}_{3_{ijk}}$  denotes the affinity between the sample  $\mathbf{x}_i$  and the sample  $\mathbf{x}_k$  by the anchor sample  $\mathbf{x}_j$ .

The Fisher-ratio-like tensor affinity defined in IPS2 [31] is also adopted as the nondecomposable tetradic affinity

$$\mathcal{S}_{4_{ijkl}} = \exp\left(-\sigma \frac{d_{ij} + d_{kl}}{d_{ik} + d_{jl} + \varepsilon}\right) \quad (18)$$

for  $i, j, k, l \in n$ , where  $d_{ij}$  denotes the distance between samples  $\mathbf{x}_i$  and  $\mathbf{x}_j$ , the parameter  $\sigma$  is a scaling constant, and the parameter  $\varepsilon$  is a given small constant less than 0.001 to overcome the instability caused by a zero denominator.

### B. Unified Form of Multiorder Affinities

Traditional SC models aim to maximize the intracluster similarity of each subgraph after graph partitioning. The intracluster similarity is typically computed by evaluating the pairwise affinities within the same subgraph/cluster. However, the intracluster similarity is not solely reflected by their pairwise relationships, it should also extend to any set of  $k$  samples. To capture the uniform expressive patterns of clustering structures at different order affinities, we introduce a metric called the total normalized affinity entropy. This entropy is used to quantify the sum of intracluster affinities based on  $k$  samples within the same cluster:

*Definition 6 (The Total N-Affinity of a Cluster):* Let  $\mathcal{C}_i \subseteq \mathcal{X}$  be a group of samples and  $\mathcal{S} \in \mathbb{R}^{n \times n \times \dots \times n}$  be the  $l$ th order

affinity tensor. The total normalized affinity of  $\mathcal{C}_i$  is defined as follows:

$$\text{Sim}(\mathcal{C}_i) = \sum_{x_{i_1}, x_{i_2}, \dots, x_{i_l} \in \mathcal{C}_i} \mathcal{L}_{x_{i_1}, x_{i_2}, \dots, x_{i_l}} \quad (19)$$

where  $\mathcal{L}$  is the normalized affinity tensor calculated by  $\mathcal{S}$ . In addition, for a partition  $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k$  of  $\mathcal{X}$ , we define the normalized associativity of the clustering as follows:

$$\text{N-Assoc}(\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k) = \sum_{i=1}^k \frac{\text{Sim}(\mathcal{C}_i)}{|\mathcal{C}_i|^l} \quad (20)$$

where  $|\mathcal{C}_i|$  denotes the sample number of cluster  $\mathcal{C}_i$ .

In this way, the affinity among any  $k$  samples can provide valuable information about the sample space distribution. By maximizing the normalized affinity defined in (20), we seek the optimal sample assignment  $\mathcal{C}_1, \dots, \mathcal{C}_k$ . Let an indicator matrix  $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_k] \in \mathbb{R}^{n \times k}$  be the sample assignment, where  $\mathbf{H}_{ij} = |\mathcal{C}_j|^{-1}$  if the sample  $\mathbf{x}_i \in \mathcal{C}_j$ ; otherwise,  $\mathbf{H}_{ij} = 0$ . Therefore, the clustering problem based on  $K$ -order affinity is as follows:

$$\max_{\mathbf{H}} \sum_{j=1}^k \mathcal{L} \otimes_l \mathbf{h}_j \otimes_{l-1} \mathbf{h}_j \cdots \otimes_1 \mathbf{h}_j \quad (21)$$

where  $\mathbf{h}_j$  denotes the  $j$ th column of  $\mathbf{H}$ .

However, solving this maximum normalized associativity is NP-hard. Alternatively, one can relax the binary assignment matrix  $\mathbf{H}$  to the orthonormal matrix  $\mathbf{V} \in \mathbb{R}^{n \times k}$ :  $\mathbf{V}^T \mathbf{V} = \mathbf{I}$  to simplify the solution. Therefore, the optimization problem of (21) is equivalent to solving the following problem:

$$\max_{\mathbf{V}} \sum_{j=1}^k \mathcal{L} \otimes_l \mathbf{v}_j \otimes_{l-1} \mathbf{v}_j \cdots \otimes_1 \mathbf{v}_j \quad (22)$$

$$\mathbf{V}^T \mathbf{V} = \mathbf{I}$$

where  $\mathbf{v}_j$  represents the  $j$ th column of  $\mathbf{V}$ . It is worth noting that the traditional SC problem corresponds to this model when  $k = 2$ .

### C. Uniform Tensor Clustering

Building upon the unified framework provided by (22) for different order affinities, we propose the UTC model. This model fuses those above nondecomposable triadic and tetradic

affinities with dyadic affinity into a scalable framework that seeks a uniform low-dimensional embedding. The proposed model is formulated as follows:

$$\begin{aligned} \min_{\mathbf{V}} \quad & \sum_{j=1}^k -\mathcal{L}_2 \otimes_2 \mathbf{v}_j \otimes_1 \mathbf{v}_j \\ & - \lambda_1 \mathcal{L}_3 \otimes_3 \mathbf{v}_j \otimes_2 \mathbf{v}_j \otimes_1 \mathbf{v}_j \\ & - \lambda_2 \mathcal{L}_4 \otimes_4 \mathbf{v}_j \otimes_3 \mathbf{v}_j \otimes_2 \mathbf{v}_j \otimes_1 \mathbf{v}_j \\ \text{s.t.} \quad & \mathbf{V}^T \mathbf{V} = \mathbf{I} \end{aligned} \quad (23)$$

where  $\lambda_1$  and  $\lambda_2$  are the two introduced hyperparameters used to balance the weights between different order affinities. The corresponding parameter sensitivity analysis can be found in Appendix B.

Intuitively, the samples belonging to the same cluster should ideally exhibit spatial proximity, regardless of which order affinity is used for measurement. Therefore, the goal of this model is to maximize different order intracluster affinities. The resulting consensus embedding not only achieves consistency in data representation based on different order affinities but also captures complementary information between them. In other words, this model can leverage various order affinities to characterize the sample proximity from different perspectives. Once we obtain the consensus embedding  $\mathbf{V}$ , it can be used to infer the sample assignment. The popular strategy is to employ heuristic methods, such as  $K$ -means clustering, on the embedding matrix  $\mathbf{V}$  to obtain the final assignment.

#### D. Numerical Scheme to Solve UTC

This section provides the strategy to solve the proposed model (23). To facilitate solving, we introduce a slack variable  $\mathbf{V}_2$  to approximate the term  $\mathbf{V}_1 * \mathbf{V}_1$ . Through tensor unfolding operations, this model is equivalent to the following form:

$$\begin{aligned} \min_{\mathbf{V}_1, \mathbf{V}_2} \quad & -\text{tr}(\mathbf{V}_1^T \mathcal{L}_2 \mathbf{V}_1 + \lambda_1 \mathbf{V}_2^T \mathcal{L}_3 \mathbf{V}_1 + \lambda_2 \mathbf{V}_2^T \mathcal{L}_4 \mathbf{V}_2) \\ \text{s.t.} \quad & \mathbf{V}_1^T \mathbf{V}_1 = \mathbf{I}, \quad \mathbf{V}_2 = \mathbf{V}_1 * \mathbf{V}_1. \end{aligned} \quad (24)$$

We adopt an alternating optimization strategy to solve this problem efficiently. Its augmented Lagrangian formulation is presented as follows:

$$\begin{aligned} \mathcal{J}(\mathbf{V}_1, \mathbf{V}_2, \mathbf{Y}_1, \mathbf{Y}_2) & \\ = & -\text{tr}(\mathbf{V}_1^T \mathcal{L}_2 \mathbf{V}_1) - \lambda_1 \text{tr}(\mathbf{V}_2^T \mathcal{L}_3 \mathbf{V}_1) - \lambda_2 \text{tr}(\mathbf{V}_2^T \mathcal{L}_4 \mathbf{V}_2) \\ & + \langle \mathbf{Y}_1, \mathbf{V}_1 * \mathbf{V}_1 - \mathbf{V}_2 \rangle + \langle \mathbf{Y}_2, \mathbf{V}_1^T \mathbf{V}_1 - \mathbf{I} \rangle \\ & + \frac{\mu}{2} (\|\mathbf{V}_1 * \mathbf{V}_1 - \mathbf{V}_2\|_F^2 + \|\mathbf{V}_1^T \mathbf{V}_1 - \mathbf{I}\|_F^2) \end{aligned} \quad (25)$$

where  $\mathbf{Y}_1$  and  $\mathbf{Y}_2$  are the Lagrange multipliers. The constant  $\mu > 0$  is a penalty parameter. We then decompose the problem into two subproblems concerning the variables  $\mathbf{V}_1$  and  $\mathbf{V}_2$ . Each variable is solved while fixing other variables. The process is updated iteratively until convergence.

*Step 1 (Solving the Subproblem  $\mathbf{V}_1$ ):* By fixing the variable  $\mathbf{V}_2$ , the problem can be simplified as follows:

$$\begin{aligned} \min_{\mathbf{V}_1} \quad & -\text{tr}(\mathbf{V}_1^T \mathcal{L}_2 \mathbf{V}_1) - \lambda_1 \text{tr}(\mathbf{V}_2^T \mathcal{L}_3 \mathbf{V}_1) \\ & + \langle \mathbf{Y}_1, \mathbf{V}_1 * \mathbf{V}_1 - \mathbf{V}_2 \rangle + \langle \mathbf{Y}_2, \mathbf{V}_1^T \mathbf{V}_1 - \mathbf{I} \rangle \end{aligned}$$

#### Algorithm 1 Algorithm to Solve the Subproblem $\mathbf{V}_1$

**Input:** The embedding matrix  $\mathbf{V}_1 \in \mathbb{R}^{n \times k}$ .

**Output:** The updated embedding matrix  $\mathbf{V}_1$ .

- 1: Set  $\mathbf{V}_1^0 = \mathbf{V}_1$ ,  $\mathbf{M} = \mathbf{0}$ ,  $\alpha = 10^{-3}$ ,  $\gamma = 0.9$ ,  $t = 0$ ,  $\epsilon = 10^{-3}$ .
- 2: **while** Not Converged **do**
- 3:      $t = t + 1$ .
- 4:     Compute the partial derivatives  $\nabla \mathcal{J}(\mathbf{V}_1^{t-1} + \gamma \mathbf{M})$  by Eq. (27).
- 5:     Update  $\mathbf{M}$  by  $\mathbf{M} = \gamma \mathbf{M} - \alpha \nabla \mathcal{J}(\mathbf{V}_1^{t-1} + \gamma \mathbf{M})$ .
- 6:     Compute the  $\mathbf{V}_1^t$  by  $\mathbf{V}_1^t = \mathbf{V}_1^{t-1} + \mathbf{M}$ .
- 7:     Update  $\mathbf{V}_1$  by  $\mathbf{V}_1 = \mathbf{V}_1^t$ .
- 8:     Check the convergence conditions:  $\|\mathbf{V}_1^t - \mathbf{V}_1^{t-1}\|_\infty \leq \epsilon$ .
- 9: **return**  $\mathbf{V}_1$ .

$$+ \frac{\mu}{2} (\|\mathbf{V}_1 * \mathbf{V}_1 - \mathbf{V}_2\|_F^2 + \|\mathbf{V}_1^T \mathbf{V}_1 - \mathbf{I}\|_F^2). \quad (26)$$

Model (26) is an unconstrained optimization problem, and therefore, its solution can be obtained by computing partial derivatives as follows:

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial \mathbf{V}_1} &= -2\mathcal{L}_2 \mathbf{V}_1 - \lambda_1 \mathcal{L}_3^T \mathbf{V}_2 + \mu \sum_{j=1}^k (\mathbf{V}_{1:j} \otimes \mathbf{I} \\ &+ \mathbf{I} \otimes \mathbf{V}_{1:j})^T (\mathbf{V}_{1:j} * \mathbf{V}_{1:j} - \mathbf{V}_{2:j} + \mathbf{Y}_{1:j}/\mu) \otimes \mathbf{e}_j^T \\ &+ 2\mu \mathbf{V}_1 (\mathbf{V}_1^T \mathbf{V}_1 - \mathbf{I} + \mathbf{Y}_2/\mu). \end{aligned} \quad (27)$$

Subsequently, we use the Nesterov accelerated gradient method to obtain the locally optimal solution of  $\mathbf{V}_1$ . The specific procedure can be found in Algorithm 1.

*Step 2 (Solving the Subproblem  $\mathbf{V}_2$ ):* By fixing the variable  $\mathbf{V}_1$ , the augmented Lagrange function can be simplified as follows:

$$\begin{aligned} \min_{\mathbf{V}_2} \quad & -\lambda_1 \text{tr}(\mathbf{V}_2^T \mathcal{L}_3 \mathbf{V}_1) - \lambda_2 \text{tr}(\mathbf{V}_2^T \mathcal{L}_4 \mathbf{V}_2) \\ & + \mathbf{Y}_1 (\mathbf{V}_1 * \mathbf{V}_1 - \mathbf{V}_2) + \frac{\mu}{2} \|\mathbf{V}_1 * \mathbf{V}_1 - \mathbf{V}_2\|_F^2. \end{aligned} \quad (28)$$

The gradient of the objective function is

$$\frac{\partial \mathcal{J}}{\partial \mathbf{V}_2} = -\lambda_1 \mathcal{L}_3 \mathbf{V}_1 - 2\lambda_2 \mathcal{L}_4 \mathbf{V}_2 + \mu \left( \mathbf{V}_2 - \mathbf{V}_1 * \mathbf{V}_1 - \frac{\mathbf{Y}_1}{\mu} \right). \quad (29)$$

By setting the gradient to 0, one can obtain the implicit solution as follows:

$$\mathbf{V}_2^* = (\mu \mathbf{I} - 2\lambda_2 \mathcal{L}_4)^{-1} (\mu \mathbf{V}_1 * \mathbf{V}_1 + \lambda_1 \mathcal{L}_3 \mathbf{V}_1 + \mathbf{Y}_1). \quad (30)$$

*Step 3 (Updating the Multipliers  $\mathbf{Y}_1$  and  $\mathbf{Y}_2$ ):*

$$\mathbf{Y}_1^t = \mathbf{Y}_1^{t-1} + \mu (\mathbf{V}_1^{t-1} * \mathbf{V}_1^{t-1} - \mathbf{V}_2^{t-1}) \quad (31)$$

$$\mathbf{Y}_2^t = \mathbf{Y}_2^{t-1} + \mu (\mathbf{V}_1^{t-1T} \mathbf{V}_1^{t-1} - \mathbf{I}). \quad (32)$$

The three steps are iteratively updated until convergence or until a stopping criterion is met:  $\max(\|\mathbf{V}_1^t - \mathbf{V}_1^{t-1}\|_F^2, \|\mathbf{V}_2^t - \mathbf{V}_2^{t-1}\|_F^2)$ , and  $\|\mathbf{V}_1^t * \mathbf{V}_1^t - \mathbf{V}_2^t\|_F^2 < \epsilon$ . The alternate strategy to solve the problem is summarized in Algorithm 2.

**Algorithm 2** High-Order Affinity Clustering**Input:** Data  $X \in \mathbb{R}^{n \times d}$ , Cluster number  $k$ .**Output:** The fused low-dimensional embedding  $V_1$  and the clustering results  $Y_{pred}$ .

- 1: Set  $V_2^0 = Y_1^0 = \mathbf{0}$ ,  $V_1^0 = Y_2^0 = \mathbf{0}$ ,  $\mu^0 = 10^{-3}$ ,  $\mu^{max} = 10^2$ ,  $\varepsilon = 10^{-3}$ ,  $t = 0$ ,  $\rho = 1.1$ , and  $\sigma = 10^{-6}$ .
- 2: Construct dyadic affinity matrix  $L_2$  by Gaussian kernel function.
- 3: Construct triadic affinity tensor  $\mathcal{L}_3$  by Eq. (17).
- 4: Construct tetradic affinity tensor  $\mathcal{L}_4$  by Eq. (18).
- 5: Unfold triadic affinity tensor  $\mathcal{L}_3$  to matrix  $L_3$  by Eq. (1).
- 6: Unfold tetradic affinity tensor  $\mathcal{L}_4$  to matrix  $L_4$  by Eq. (2).
- 7: **while** Not Converged **do**
- 8:      $t = t + 1$ .
- 9:     Update  $V_1^t$  by Algorithm 1.
- 10:     Update  $V_2^t$  by Eq. (30).
- 11:     Update  $Y_1^t$  by Eq. (31).
- 12:     Update  $Y_2^t$  by Eq. (32).
- 13:      $\mu^t = \min(\rho\mu^{t-1}, \mu^{max})$ .
- 14:     Check the convergence conditions:  

$$\max(\|V_1^t - V_1^{t-1}\|_F^2, \|V_2^t - V_2^{t-1}\|_F^2, \|V_1^t * V_1^{t-1} - V_2^t\|_F^2) < \varepsilon$$
- 15:     Update  $V_1$  by  $V_1 = V_1^t$ .
- 16:     Perform SC on  $V_1 V_1^T$  to obtain  $Y_{pred}$ .
- 17: **return**  $V_1$  and  $Y_{pred}$ .

## IV. EXPERIMENTS

In this section, we demonstrate the superiority of the proposed UTC method on both synthetic and real datasets. Two synthetic datasets with special structures (referred to as Syndata1-1 and Syndata1-2) and one high-dimensional synthetic dataset (referred to as Syndata2) are constructed for experiments. The experiments on Syndata1-1 and Syndata1-2 demonstrate the limitations of pairwise affinities when dealing with data of specific structures, while high-order affinity can overcome this challenge. The comparison and visualization results on Syndata2 with increased dimensionality validate the robustness of our method. Finally, we conduct experiments on six real datasets to verify the proposed UTC's effectiveness compared with baseline methods.

## A. Experimental Settings

We employ nine popular clustering methods to benchmark the proposed UTC. A brief introduction to these methods is given as follows.

- 1) *Spectral Clustering (SC)* [33]: The classic SC gives a baseline on behalf of pairwise similarity.
- 2) *Clustering With Adaptive Neighbors (CANs)* [19]: Dynamically learning the affinity matrix by assigning the adaptive neighbors for each data point.
- 3) *Simple Multiple Kernel K-Means (SMKMKM)* [23]: Solving the multikernel clustering problem by a minimization–maximization strategy to obtain the kernel coefficient and clustering partition matrix simultaneously.

- 4) *Clique Averaging + ncut (CAVE)* [27]: Approximating hypergraphs which represent high-order sample relationships by the clique averaging to create a weighted graph for clustering.
- 5) *Self-Paced Active Clustering Ensemble (SPACE)* [36]: Automatically selecting and annotating significant data, clustering ensemble is achieved through a few annotations by propagating both must-link and cannot-link constraints.
- 6) *Adaptive Consensus Multiple K-Means (ACMK)* [37]: Integrating a set of base results into consensus learning to obtain the final clustering result.
- 7) *Natural Density Peaks K-Means (NDPK)* [1]: Based on the natural density peaks (NDPs), a new  $K$ -means algorithm is proposed to identify clusters of arbitrary shapes.
- 8) *Pair-to-Pair Clustering (PPC)* [31]: Tetradic undecomposable affinities are used to obtain the low-dimension embedding as the final similarity matrix of the spectral method.
- 9) *Integrating Tensor Similarity and Pairwise Similarity (IPS2)* [31]: Applying the SC method on the fused similarity that combines the pair-to-pair affinities and pairwise similarity.
- 10) *UTC*: The proposed method.

Regarding the type of affinity, the nine methods can be divided into three categories: pairwise affinity, high-order affinity, and fused-order affinity.

The performance of each method is evaluated with respect to five popular metrics: accuracy (ACC), adjusted rand index (ARI),  $F$ -score, normalized mutual information (NMI), and purity. The larger the value is, the better the performance is. To standardize the experimental criteria, both the comparison and UTC methods employed a Gaussian kernel function to construct the dyadic affinity matrix. We executed all the methods under the same settings 50 times and calculated the mean of all evaluation metrics.

## B. Experiment on Syndata1 to Validate the Potential of High-Order Affinity

To verify that high-order affinity can capture special data structures that pairwise affinity fails to recognize, we applied the pairwise relationship-based SC method and the UTC method, which incorporates third-order affinity, testing on Syndata1-1 and Syndata1-2 datasets. Syndata1-1 comprises three intersecting lines, while Syndata1-2 consists of a straight line intersecting a plane. The experimental results in Fig. 3 demonstrate that traditional pairwise affinities fail to accurately reflect the data relationships in these datasets. On the other hand, UTC with triadic affinities achieves 100% ACC, perfectly assigning samples. These observations show that methods based on pairwise relationships can only detect local neighborhood relationships of samples and perform poorly when handling data with special structures. Conversely, high-order affinities can extend the scope of measuring data relationships and accurately capturing complex data structures. Therefore, by integrating high-order affinities, clustering performance is enhanced.

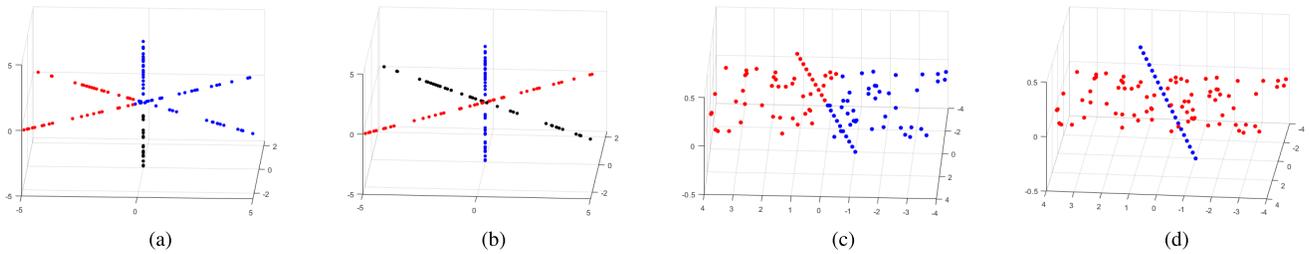


Fig. 3. Clustering comparisons between SC and UTC. (a) Result on Syndata1-1 by SC. (b) Result on Syndata1-1 by UTC. (c) Result on Syndata1-2 by SC. (d) Result on Syndata1-2 by UTC.

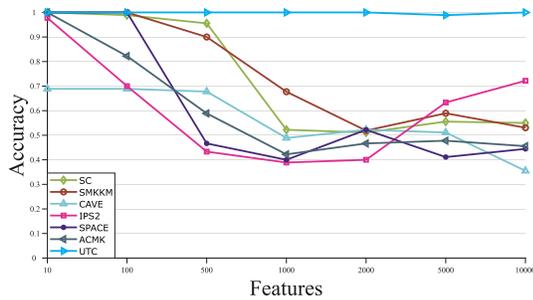


Fig. 4. Experiment to demonstrate the stability of UTC when dealing with high-dimensional feature corruption.

### C. Experiment on Syndata2 to Validate the Stability of UTC

*Syndata2* consists of three subgroups, each derived from independent and identically distributed normal distributions with an equal standard deviation of 0.5 and a mean of 2. The data from different groups are orthogonal and each group contains 30–40 samples.

We applied the proposed UTC on *Syndata2* to validate its stability in handling high-dimensional data by comparison with six benchmark methods. As shown in Fig. 4, when the feature dimension varies from 10 to 10 000, pairwise affinities tend to be ambiguous with expanding feature dimensionality, limiting the clustering performance. However, UTC, which incorporates high-order affinities, maintains robustness and achieves accurate clustering.

Moreover, when the data dimension reaches 10 000, Table I shows that our proposed UTC methods, including fusing pairwise with triadic affinity, tetradic affinity, and both, substantially outperform all nine methods. The clustering results of fusing triadic affinity and fusing tetradic affinity are higher than those of the second-highest baseline method by 14% and 5% in terms of ACC, demonstrating the effectiveness of fusing high-order affinity with pairwise relation under HDLSS data. In addition, the ACC achieves 100% when utilizing both high-order affinities, proving that fusing different order affinities is more helpful in fully explaining the data structure than merging only a single high-order affinity.

We evaluated the quality of embeddings learned by the UTC method via plotting t-distributed stochastic neighbor embedding (t-SNE) scatter plots and heatmaps of the affinity matrix on *Syndata2*, as shown in Fig. 5. When visualizing the raw data by t-SNE, most samples are mixed together. On the other hand, the heatmap of the pairwise affinity matrix

has no clear-cut structure. In contrast, when fusing triadic or tetradic affinities in UTC, both visualization results reveal that these high-order affinities can accurately identify one of the clusters but struggle to differentiate samples in the remaining two clusters effectively. These two high-order affinities each identify completely different clusters, indicating that different high-order affinities can provide distinct information for clustering. Finally, from the visualization results of UTC incorporating triadic and tetradic affinities, we observe that the learned embeddings exhibit perfect separation in the t-SNE scatter plot, and the heatmap of the affinity matrix also reveals clear block structures. In summary, the above experimental results validate that the high-order tensor affinities make up for the insufficiency of the pairwise affinity matrix when corrupted by high-dimensional features.

### D. Experiments on Real Datasets

We validated the power of UTC by applying it to six public benchmark datasets. The six datasets are chosen to be representative of various sources, including facial images, object images, and bioinformatics data. The statistics for the six datasets are summarized in Table II. In addition, the detailed descriptions of these datasets are as follows.

- 1) *Yale* dataset has 165 grayscale images covering 15 different individuals. Each individual has 11 different facial and configuration images. We extracted 4096 dimensional raw pixel values of every image in five classes for our experiment.
- 2) *Coil20* dataset contains 1440 images of 20 categories of objects. Each category has 72 images from different views. We sample 20 samples from five classes, and all images have a size of  $128 \times 128$ .
- 3) *Lymphoma* dataset, one of the most common subtypes of non-Hodgkin's Lymphoma has two molecularly different forms of diffuse large B-cell Lymphoma (DLBCL), which have gene expression patterns that indicate different stages of B-cell differentiation. The dataset contains a total of 62 samples. Each sample has 4702 expression fragments.
- 4) *DriveFace* dataset contains image sequences of subjects while driving in real scenarios. It is composed of 606 samples of 6400 features each, acquired over different days from three drivers with various facial features, such as glasses and beards. For each type, we pick 26 samples for the experiment.

TABLE I  
CLUSTERING RESULTS ON SYNDATA-2 WHEN THE DIMENSIONALITY REACHES 10 000

Datasets	Methods	ACC	ARI	F-SCORE	NMI	PURITY
Syndata2	SC	0.5500	0.4427	0.6633	0.6544	0.6800
	CAN	0.6449	0.2306	0.5342	0.3323	0.6449
	SMKMM	0.5306	0.1465	0.4239	0.1636	0.4282
	SPACE	0.4444	0.0563	0.4902	0.2273	0.4444
	ACMK	0.4556	0.0994	0.4409	0.1339	0.5000
	NPDK	0.3667	-0.0038	0.4338	0.0126	0.3778
	CAVE	0.3551	-0.013	0.3734	0.0018	0.3551
	PPS	0.5636	0.3166	0.4472	0.4453	0.5636
	IPS2	0.7218	0.4847	0.5848	0.5862	0.7818
	UTC (Fusing dyadic and triadic affinity)	0.8600	0.6719	0.7865	0.7590	0.8600
	UTC (Fusing dyadic and tetradic affinity)	0.7700	0.5204	0.6799	0.5776	0.7700
UTC	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	

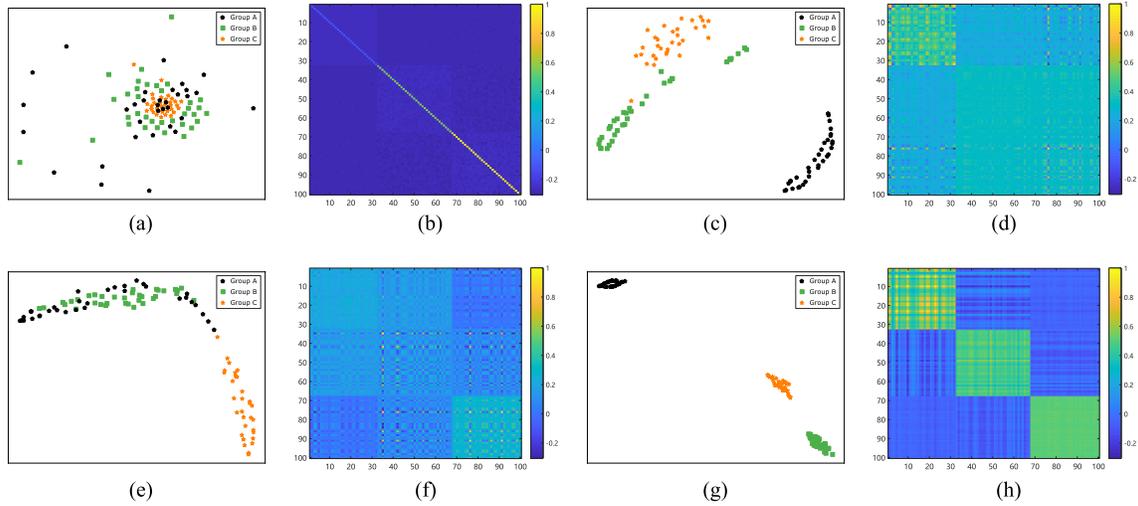


Fig. 5. Visualization of the uniform embeddings from Syndata2 with a feature dimensionality of 10 000 by t-SNE. Most of the samples are mixed together when applying t-SNE to (a) raw data and (b) corresponding affinity matrix has no clear-cut structure. In comparison, when fusing the triadic affinity with dyadic affinity, (c) samples in group A are well-segmented, while the other groups remain undistinguished. For example, (d) corresponding affinity matrix has blocky structures in the upper left. When fusing the tetradic affinity with pairwise affinity, (e) group C can be identified accurately while the remaining two types are merged, resulting in (f) less distinguished affinity. Finally, by fusing the triadic, tetradic, and dyadic affinities jointly, (g) each group can be perfectly identified, resulting in (h) three distinct blocks on the diagonal of the corresponding affinity matrix.

TABLE II  
STATISTICS ON SIX TESTED REAL-WORLD DATASETS

Dataset	Type	Samples	Features	Clusters
Yale	Facial Image	55	4,096	5
Coil20	Object Image	100	16,384	5
Lymphoma	Bioinformatics	62	4,702	3
DriveFace	Facial Image	78	6,400	3
Lung	Bioinformatics	100	12,600	5
GLI-85	Bioinformatics	85	22,283	2

- 5) *Lung* dataset contains a total of 203 samples that can be divided into five classes, with 149, 21, 20, 6, and 17 samples, respectively. Each sample has 12 600 genes. We selected 40 samples of the first category and all samples of the remaining four categories for experiments.
- 6) *GLI-85* dataset consists of the transcriptional data of 85 diffuse infiltrating gliomas from 74 patients. Those gliomas can be divided into two kinds of tumor subclasses. Each instance has 22 283 features.

The proposed UTC and nine comparative methods are applied to these six datasets, and the results are summarized in Table III. The bold value represents the best result. It can

be observed from Table III that our proposed method significantly outperforms the other nine methods across all types of datasets. In terms of NMI, our method outperforms SC by over 25% on the Yale datasets, surpassing the second-highest SMKMM by more than 12%. On the GLI-85 dataset, our results are more than 20% and 10% higher than those of the baseline and second-highest IPS2 methods. There are also 5%, 12%, and 3% improvements compared with the second-best performance on the Coil, DriveFace, and Lung datasets. Furthermore, our method correctly assigns all samples to their clusters in the Coil and Lymphoma datasets.

We also visualized the sample spatial distribution after t-SNE and the heatmap of the affinity matrix on the Yale, Coil, and DriveFace datasets in Fig. 6 and the Lymphoma, Lung, and GLI-85 datasets in Fig. 7, respectively. As shown in these figures, embeddings based on pairwise affinity mostly mix together, making it difficult to distinguish between different subgroups. The affinity matrix constructed from such embeddings also fails to capture relationships between samples. In contrast, the t-SNE obtained by the UTC method reveals samples from the same cluster clustering together, and the heatmap also displays a clear block structure. These

TABLE III  
CLUSTERING PERFORMANCE ON SIX REAL DATASETS

Datasets	Methods	ACC	NMI	F-SCORE	ARI	PURITY	Datasets	Methods	ACC	NMI	F-SCORE	ARI	PURITY
YALE	SC	0.5556	0.4399	0.4520	0.3186	0.4276	Lymphoma	SC	0.8065	0.6606	0.7167	0.5059	0.8548
	CAN	0.5000	0.4611	0.4562	0.2902	0.3612		CAN	0.9839	0.9254	0.9733	0.9471	0.9902
	SMKKM	0.6852	0.5689	0.5647	0.4515	0.5458		SMKKM	0.6129	0.6546	0.6846	0.4681	0.5746
	SPACE	0.6667	0.5391	0.5618	0.4544	<b>0.6668</b>		SPACE	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
	ACMK	0.6482	0.5215	0.5356	0.4167	0.5185		ACMK	0.9678	0.8648	0.9637	0.9281	0.9677
	NPDK	0.3889	0.3034	0.3514	0.0950	0.3778		NPDK	0.9516	0.7902	0.9350	0.8731	0.9516
	CAVE	0.5741	0.4568	0.4441	0.2994	0.3982		CAVE	0.6129	0.3726	0.4800	0.1380	0.3960
	PPS	0.6111	0.4380	0.4503	0.3245	0.4478		PPS	0.9194	0.7823	0.8696	0.7570	0.9194
	IPS2	0.6482	0.5410	0.5403	0.4290	0.5136		IPS2	0.9839	0.9255	0.9733	0.9471	0.9839
UTC	<b>0.8148</b>	<b>0.6961</b>	<b>0.6151</b>	<b>0.5156</b>	0.5546	UTC	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>		
Coil	SC	0.8300	0.7513	0.7375	0.6746	0.7421	Lung	SC	0.8000	0.6848	0.6750	0.5855	0.8100
	CAN	0.7500	0.7675	0.7207	0.6424	0.8190		CAN	0.7700	0.6995	0.6971	0.5973	0.6714
	SMKKM	0.8100	0.7432	0.7281	0.6632	0.7316		SMKKM	0.7600	0.6361	0.6146	0.4987	0.5961
	SPACE	0.9700	0.9460	0.9419	0.9280	0.9700		SPACE	0.8700	0.7723	0.7768	0.7043	<b>0.8700</b>
	ACMK	0.7900	0.7311	0.7191	0.6434	0.8600		ACMK	0.8700	0.7813	0.7082	0.7797	0.8400
	NPDK	0.9700	0.9460	0.9419	0.9280	0.9700		NPDK	0.5600	0.5569	0.5259	0.2694	0.5600
	CAVE	0.8400	0.7575	0.7462	0.6852	0.8400		CAVE	0.5200	0.3928	0.4295	0.2603	0.4480
	PPS	0.9300	0.8690	0.8694	0.8383	0.9300		PPS	0.7300	0.6311	0.6472	0.5410	0.8000
	IPS2	0.9700	0.9342	0.9407	0.9266	0.9373		IPS2	0.7600	0.6783	0.6983	0.5985	0.7290
UTC	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	UTC	<b>0.9100</b>	<b>0.8117</b>	<b>0.8341</b>	<b>0.7837</b>	0.8341		
DriveFace	SC	0.7436	0.5723	0.6549	0.4711	0.7046	GLI-85	SC	0.6941	0.1781	0.5976	0.1612	0.6413
	CAN	0.5182	0.6478	0.6566	0.4355	0.5442		CAN	0.6588	0.0343	0.6970	-0.0353	0.5620
	SMKKM	0.8974	0.7951	0.8252	0.7312	0.8523		SMKKM	0.7412	0.2524	0.6355	0.2247	0.6840
	SPACE	0.9231	0.8266	0.8610	0.7924	0.9231		SPACE	0.6353	0.1872	0.5769	0.0521	0.6471
	ACMK	0.8333	0.5931	0.7115	0.5669	0.8333		ACMK	0.7294	0.1450	0.6302	0.1989	0.7294
	NPDK	0.7436	0.5273	0.3752	0.6064	0.7692		NPDK	0.7529	0.2392	0.6465	0.2482	0.7529
	CAVE	0.3974	0.1482	0.4794	0.0234	0.3333		CAVE	0.7412	0.1710	0.7376	0.1225	0.7412
	PPS	0.7436	0.5259	0.6064	0.3752	0.5322		PPS	0.5294	0.1103	0.5777	-0.0447	0.6941
	IPS2	0.7821	0.5970	0.6882	0.5185	0.7528		IPS2	0.7294	0.2754	0.6267	0.2012	0.7294
UTC	<b>0.9872</b>	<b>0.9503</b>	<b>0.9739</b>	<b>0.9613</b>	<b>0.9734</b>	UTC	<b>0.8353</b>	<b>0.3827</b>	<b>0.7413</b>	<b>0.4422</b>	<b>0.8353</b>		

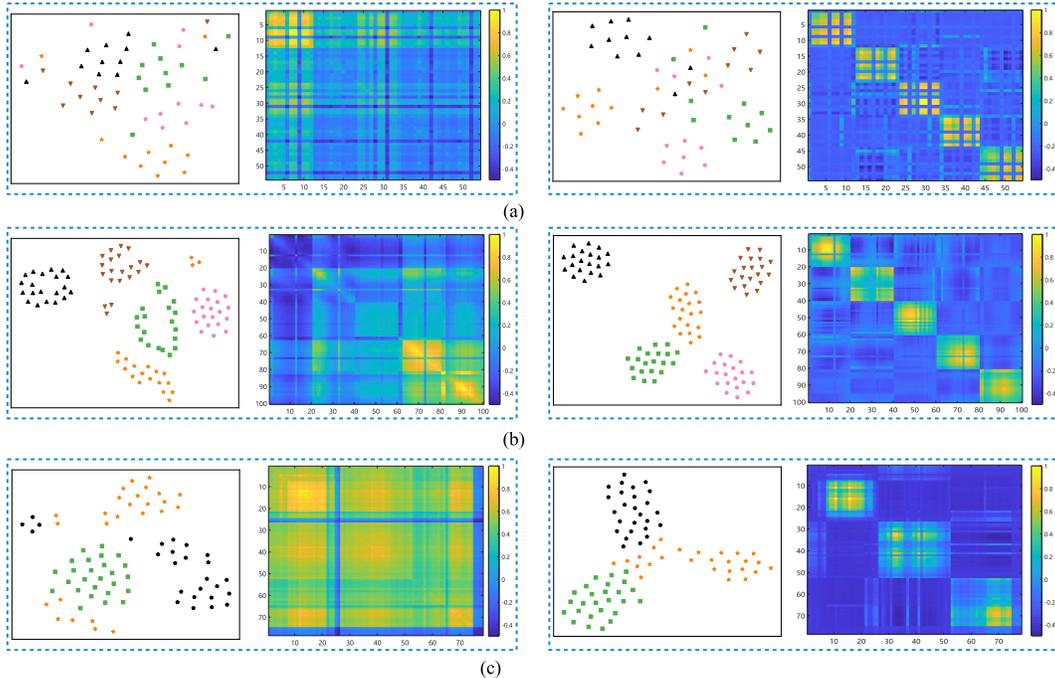


Fig. 6. Visualization of the latent representations with t-SNE and corresponding affinity heatmaps of (a) Yale, (b) Coil, and (c) DriveFace datasets. The first and second columns are the visualizations of the embedding by t-SNE on the raw data and the samples' affinity heatmap. The third and fourth columns are the visualization results of the obtained uniform embedding by UTC and the samples' affinity heatmap, respectively.

observations affirm that UTC can better reflect the true structure of the data compared with single-order methods.

In summary, the reason for the superior performance of UTC is that it utilizes the complementarity of high- and low-order affinities to comprehensively capture the spatial structure of data, thereby leading to a decisive effect on the clustering performance.

### E. Computational Complexity Study

1) *Computational Complexity Analysis*: The UTC consists of two major computational components: affinity construction and the optimization of UTC. In the affinity construction step, we adopt the  $K$ -nearest neighbors (KNN) strategy to calculate the triadic and tetradic affinities. KNN strategy can reduce the time complexity from  $\mathcal{O}(n^4)$  to  $\mathcal{O}(nk^4)$ . Thus, the

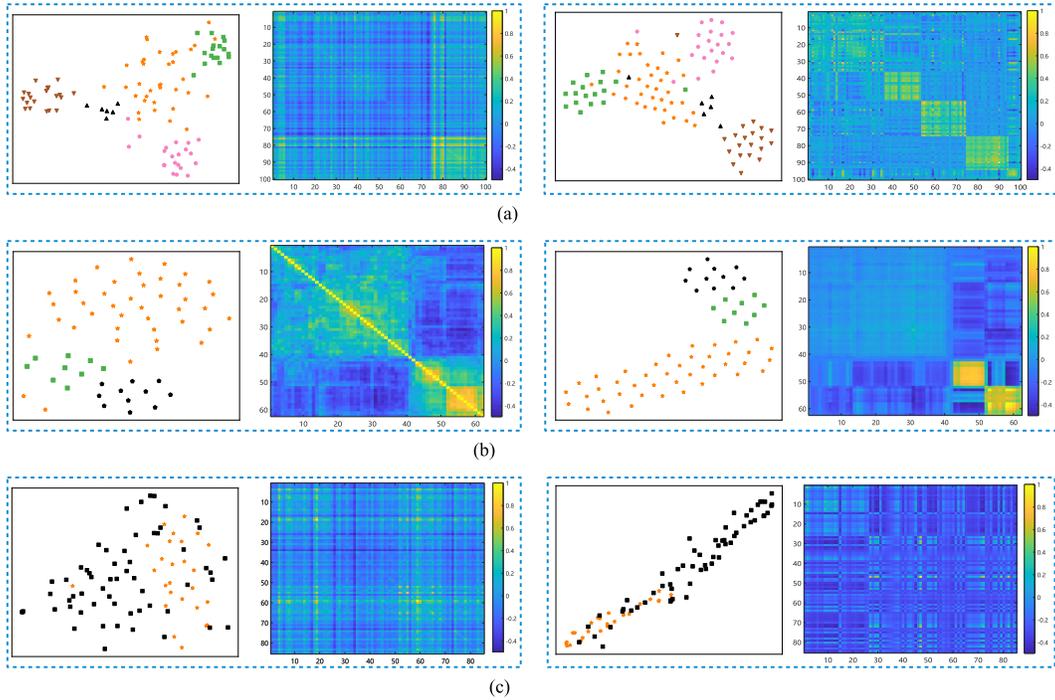


Fig. 7. Visualization of the latent representations with t-SNE and corresponding affinity heatmaps of (a) Lung, (b) Lymphoma, and (c) GLI-85 datasets. The first and second columns are the visualization by t-SNE of the raw data and the samples' affinity heatmap. The third and fourth columns are the visualization results of the obtained uniform embedding by UTC and the samples' affinity heatmap, respectively.

two affinity tensors are sparse, with the number of nonzero elements being  $s$ . For the latter, we solve the model using an iterative strategy. The subproblem of  $V_1$  based on the gradient descent strategy has a computational complexity of  $\mathcal{O}(mcsn^2)$ , where  $c$  denotes the number of clusters, and the maximum iteration number is  $m$ . Solving the other subproblem  $V_2$  takes  $\mathcal{O}(csn^2)$  by the Germs algorithm. Thus, each iteration has a total computational cost of  $\mathcal{O}(mcsn^2) + \mathcal{O}(csn^2) = \mathcal{O}((m+1)csn^2)$ . In total, the complexity of learning the uniform embedding in Algorithm 2 is  $\mathcal{O}[nk^4 + K((m+1)csn^2)]$ , with  $K$  being the number of iterations.

2) *Running Time Comparison*: To further validate the efficiency of UTC, we tested the running time together with the clustering performance of different methods on various datasets, including Syndata2, Lymphoma, GLI-85, and Drive-Face datasets. Fig. 8 illustrates a scatter plot depicting the relationship between running time ( $x$ -axis) and NMI ( $y$ -axis). It can be observed that pairwise-affinity-based methods such as SC, CAN, and SMKKM are less time-consuming but result in suboptimal clustering performance. Conversely, methods based on high-order affinity generally require more time but achieve superior performance. This result suggests a significant improvement in clustering performance relative to the increase in computation time. In practical applications, the increased computational cost is entirely acceptable when dealing with HDLSS data.

In addition, we examined the running time and performance of UTC when incorporating different orders, including triadic affinity with dyadic affinity, tetradic affinity with dyadic affinity, and dyadic with both two high-order affinities. As shown in Fig. 8(a), (b), and (d), the method only fusing triadic or tetradic affinity with dyadic affinity is inferior to the

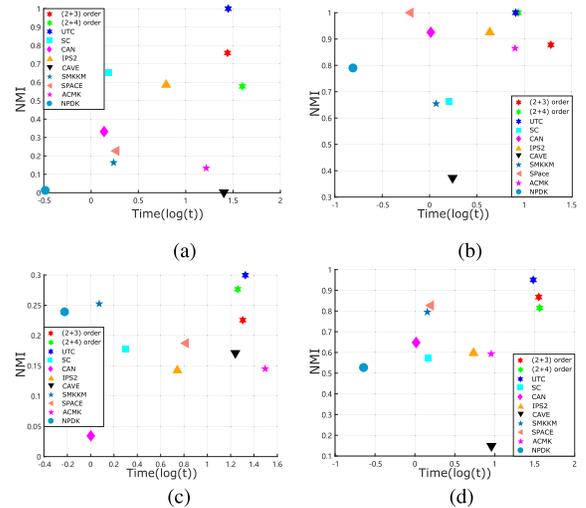


Fig. 8. Scatter plot between running time ( $x$ -axis) and NMI ( $y$ -axis) on four different datasets. (a) Syndata2, (b) Lymphoma, (c) GLI-85, and (d) DriveFace.

UTC that combines all three types of affinities, while also incurring additional time consumption. This demonstrates that introducing other high-order affinities provides supplementary information and accelerates UTC's learning efficiency, leading to faster convergence.

In summary, the integration of different high-order affinities can significantly enhance the clustering performance and boost convergence in most cases.

#### F. Robustness of UTC on Noisy Datasets

To evaluate the robustness of the proposed UTC, we tested the clustering performance of UTC under different types of

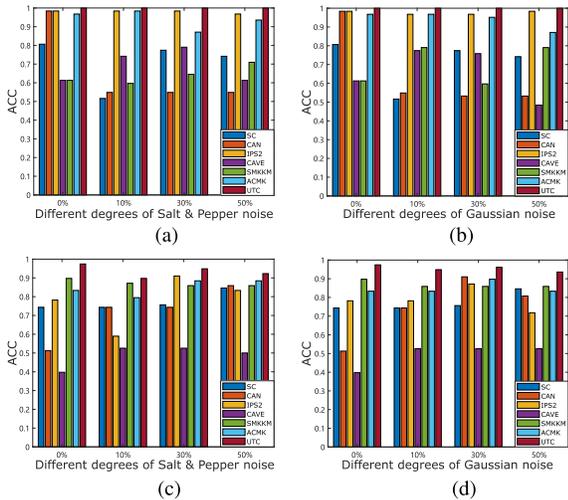


Fig. 9. Clustering results ACC on (a) Lymphoma corrupted by salt and pepper noise, (b) Lymphoma corrupted by Gaussian noise, (c) DriveFace corrupted by salt and pepper noise, and (d) DriveFace corrupted by Gaussian noise.

noise interference. Specifically, six noisy datasets were generated by adding 10%, 30%, and 50% Gaussian noise and salt and pepper noise to the Lymphoma and DriveFace datasets. The bar graphs of these noise datasets are shown in Fig. 9. It is evident that high-order affinity-based methods, such as IPS2 and UTC, exhibited superior robustness compared with other pairwise-based methods under both types of noise influence. Furthermore, UTC demonstrated excellent effectiveness on all noisy datasets, indicating that the embeddings learned by jointly incorporating multiorder affinity are more resilient to noise.

Overall, our experiments highlight the ability of UTC to handle noise interference and provide robust clustering performance. The incorporation of multiple order affinities in UTC proves to be advantageous, making it a promising approach for real-world scenarios where noise is present in high-dimensional datasets.

### G. Convergence Analysis

As shown in Fig. 10, we plotted the corresponding convergence curves on datasets Syndata2, Lymphoma, DriveFace, and Coil20 to demonstrate the convergence of the proposed method. The convergence curve is obtained by calculating the total error after each iteration. We computed the constraint error  $\|V_2^t - V_1^t * V_1^t\|_F^2$  and the variable changes  $\|V_1^t - V_1^{t-1}\|_F^2$  and  $\|V_2^t - V_2^{t-1}\|_F^2$  at each iteration, using the maximum value of these quantities as the total error:  $\text{Error} = \max(\|V_2^t - V_1^t * V_1^t\|_F^2, \|V_1^t - V_1^{t-1}\|_F^2, \|V_2^t - V_2^{t-1}\|_F^2)$ . The experimental results show that the total error on all four datasets rapidly decreases and reaches a stable state within 20 iterations. Taking advantage of the efficient solving of each subproblem, our method achieves high efficiency. Detailed convergence analysis of the UTC can be found in Appendix A.

## V. CONCLUSION

Popular unsupervised learning heavily pins on the accurate sample-to-sample affinity, which, unfortunately, is easily

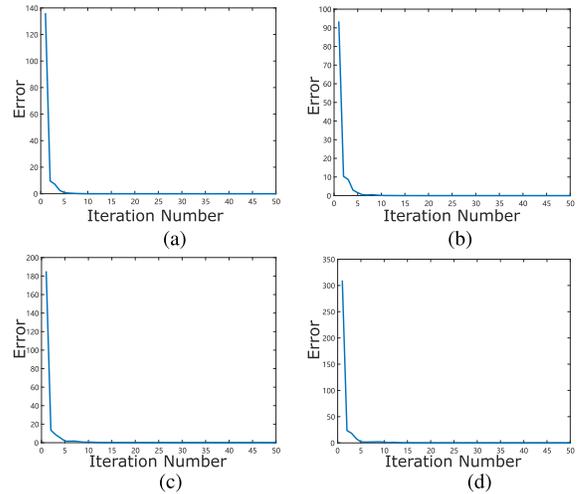


Fig. 10. Convergence curve on (a) Syndata2, (b) Lymphoma, (c) Driveface, and (d) Coil20 datasets.

broken down by noises or large feature dimensions. Such a notoriously “*curse of the dimension*” limits the learning capability of popular schemes. The current study demonstrates that high-order affinities, which describe relationships among multiple samples, exhibit significant advantages in handling data with complex structures. One can use multiple order affinities to characterize the samples’ proximity, other than merely using dyadic affinity. In elaborating the unified tensor SC model, tensor–vector products, including Arithmetical, Khatri–Rao, and Kronecker products, are used to formulate sample affinities uniformly for all orders, which opens a new way to study multiple samples. These operations organize the high-order decomposable affinity with popular affinity in a uniform form. On this basis, we have designed corresponding nondecomposable triadic and tetradic affinities to supplement dyadic affinity and proposed a unified learning model to effectively fuse multiple samples’ affinities of different orders to obtain uniform low-dimensional embedding. Experiments on synthetic data have demonstrated the power of fusing different order affinities. Furthermore, experiments on several real datasets with small sample sizes yet large feature dimensionality have shown the effectiveness and superiority of the method over other popular approaches.

## APPENDIX A

### CONVERGENCE ANALYSIS OF UTC

We utilize the alternating iterative algorithm shown in Algorithm 2 to solve the minimizing optimization solutions of UTC as indicated in (24). When solving for a variable, we fix the remaining variables as constants. By iteratively solving the subproblems alternately, the objective function of UTC eventually converges to a local solution. We first analyze each term of the primal objective function (23) is bounded. Specifically, the proof that each term is bounded is as follows.

- 1) The first term  $-\text{tr}(V_1^T L_3 V_1)$  of the objective function is bounded, as proven in the following equation:

$$\begin{aligned} & -\text{tr}(V_1^T L_2 V_1) \\ & = -\text{tr}(V_1 V_1^T L_2) \geq -\frac{1}{2}(\|V_1 V_1^T\|_F^2 + \|L_2\|_F^2) \end{aligned}$$

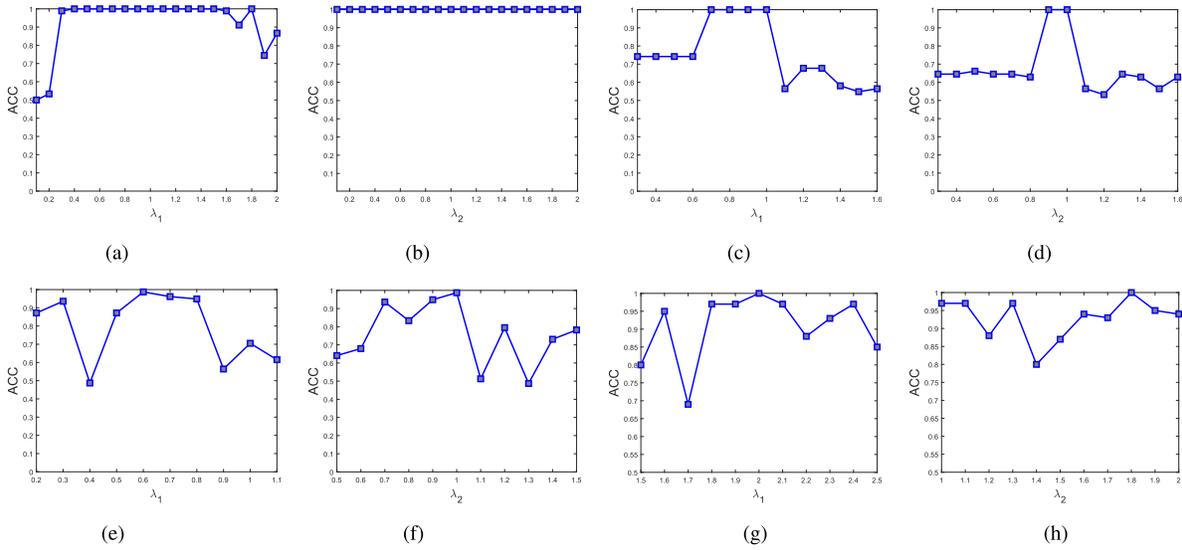


Fig. 11. Sensitivity of our proposed MKCTM method to fixed parameters  $\lambda_1$  and  $\lambda_2$  on Syndata2, Lymphoma, DriveFace, and Coil20 datasets. (a) Syndata2 with fixed  $\lambda_2$ . (b) Syndata2 with fixed  $\lambda_1$ . (c) Lymphoma with fixed  $\lambda_2$ . (d) Lymphoma with fixed  $\lambda_1$ . (e) DriveFace with fixed  $\lambda_2$ . (f) DriveFace with fixed  $\lambda_1$ . (g) Coil20 with fixed  $\lambda_2$ . (h) Coil20 with fixed  $\lambda_1$ .

$$\begin{aligned}
 &= -\frac{1}{2}(\text{tr}(\mathbf{V}_1 \mathbf{V}_1^T \mathbf{V}_1 \mathbf{V}_1^T) + \|\mathbf{L}_2\|_F^2) \\
 &= -\frac{1}{2}(\text{tr}(\mathbf{I}_k) + \|\mathbf{L}_2\|_F^2) = -\frac{1}{2}(k + c_1) \quad (33)
 \end{aligned}$$

where  $\mathbf{I}_k \in \mathbb{R}^{k \times k}$  is the identity matrix and  $c_1 = \|\mathbf{L}_2\|_F^2$  is a constant.

2) To prove the boundedness of the second term of UTC, we first provide the following property.

*Property 1:* Let  $\mathbf{V}_2 = \mathbf{V}_1 * \mathbf{V}_1$  and  $\mathbf{V}_1^T \mathbf{V}_1 = \mathbf{I}_k$ , we have  $\mathbf{V}_2^T \mathbf{V}_2 = \mathbf{I}_k$ .

*Proof:* Since the Khatri-Rao product is the column-wise Kronecker product, we analyze  $\mathbf{V}_2^T \mathbf{V}_2$  column by column. For the  $i$ th column of  $\mathbf{V}_2^T \mathbf{V}_2$ , we have

$$\begin{aligned}
 \mathbf{V}_{2_i}^T \mathbf{V}_{2_j} &= (\mathbf{V}_{1_i} \otimes \mathbf{V}_{1_i})^T (\mathbf{V}_{1_j} \otimes \mathbf{V}_{1_j}) \\
 &= (\mathbf{V}_{1_i}^T \mathbf{V}_{1_j}) \otimes (\mathbf{V}_{1_i}^T \mathbf{V}_{1_j}) \\
 &= \begin{cases} 1, & \text{if } i = j \\ 0, & \text{otherwise.} \end{cases} \quad (34)
 \end{aligned}$$

□

Based on Property 1, the second term  $-\text{tr}(\mathbf{V}_2^T \mathbf{L}_3 \mathbf{V}_1)$  can be deduced as follows:

$$\begin{aligned}
 &-\text{tr}(\mathbf{V}_2^T \mathbf{L}_3 \mathbf{V}_1) \\
 &\geq -\frac{1}{2}(\|\mathbf{V}_2\|_F^2 + \|\mathbf{L}_3 \mathbf{V}_1\|_F^2) \\
 &= -\frac{1}{2}(\text{tr}(\mathbf{V}_2^T \mathbf{V}_2) + \text{tr}(\mathbf{V}_1^T \mathbf{L}_3^T \mathbf{L}_3 \mathbf{V}_1)) \\
 &\geq -\frac{1}{2}\left(\mathbf{I}_k + \frac{1}{2}\|\mathbf{V}_1 \mathbf{V}_1^T\|_F^2 + \frac{1}{2}\|\mathbf{L}_3^T \mathbf{L}_3\|_F^2\right) \\
 &= -\frac{1}{2}\left(k + \frac{1}{2}\text{tr}(\mathbf{V}_1 \mathbf{V}_1^T \mathbf{V}_1 \mathbf{V}_1^T) + \frac{1}{2}\|\mathbf{L}_3^T \mathbf{L}_3\|_F^2\right) \\
 &= -\frac{1}{2}\left(k + \frac{1}{2}k + \frac{1}{2}c_2\right) \quad (35)
 \end{aligned}$$

where  $c_2 = \|\mathbf{L}_3^T \mathbf{L}_3\|_F^2$  is a constant.

3) Similarly, based on Property 1, from the third term  $-\text{tr}(\mathbf{V}_2^T \mathbf{L}_4 \mathbf{V}_2)$ , we can obtain

$$\begin{aligned}
 -\text{tr}(\mathbf{V}_2^T \mathbf{L}_4 \mathbf{V}_2) &= -\text{tr}(\mathbf{V}_2 \mathbf{V}_2^T \mathbf{L}_4) \\
 &\geq -\frac{1}{2}(\|\mathbf{V}_2 \mathbf{V}_2^T\|_F^2 + \|\mathbf{L}_4\|_F^2) \\
 &= -\frac{1}{2}(\text{tr}(\mathbf{V}_2 \mathbf{V}_2^T \mathbf{V}_2 \mathbf{V}_2^T) + \|\mathbf{L}_4\|_F^2) \\
 &= -\frac{1}{2}(\text{tr}(\mathbf{I}_k) + \|\mathbf{L}_4\|_F^2) \\
 &= -\frac{1}{2}(k + c_3) \quad (36)
 \end{aligned}$$

where  $c_3 = \|\mathbf{L}_4\|_F^2$  is a constant. Therefore, the objective function is bounded with a lower bound  $c = -(1/4)(2 + \lambda_1 + 2\lambda_2)k - (1/2)c_1 - (1/4)\lambda_1 c_2 - (1/2)\lambda_2 c_3$ . During the iterative optimization process,  $\mathcal{J}(\mathbf{V}_1^k, \mathbf{V}_2^k, \mathbf{Y}_1^k, \mathbf{Y}_2^k)$  represents the augmented Lagrangian multiplier form of (7) at the  $k$ th iteration. As each subproblem can find the current local solution during the iteration, the update for the  $(k+1)$ th iteration is

$$\begin{aligned}
 &\mathcal{J}(\mathbf{V}_1^k, \mathbf{V}_2^k, \mathbf{Y}_1^k, \mathbf{Y}_2^k) \\
 &\geq \mathcal{J}(\mathbf{V}_1^{k+1}, \mathbf{V}_2^k, \mathbf{Y}_1^k, \mathbf{Y}_2^k) \\
 &\geq \mathcal{J}(\mathbf{V}_1^{k+1}, \mathbf{V}_2^{k+1}, \mathbf{Y}_1^k, \mathbf{Y}_2^k) \\
 &\geq \mathcal{J}(\mathbf{V}_1^{k+1}, \mathbf{V}_2^{k+1}, \mathbf{Y}_1^{k+1}, \mathbf{Y}_2^k) \\
 &\geq \mathcal{J}(\mathbf{V}_1^{k+1}, \mathbf{V}_2^{k+1}, \mathbf{Y}_1^{k+1}, \mathbf{Y}_2^{k+1}) \\
 &\geq c. \quad (37)
 \end{aligned}$$

Therefore, the iterative solving process of Algorithm 2 can monotonically reduce the value of  $\mathcal{J}(\mathbf{V}_1^k, \mathbf{V}_2^k, \mathbf{Y}_1^k, \mathbf{Y}_2^k)$ . Combining this with the fact that  $\mathcal{J}(\mathbf{V}_1^k, \mathbf{V}_2^k, \mathbf{Y}_1^k, \mathbf{Y}_2^k)$  is bounded below, Algorithm 2 can obtain a convergent solution for UTC based on the monotone convergence theorem.

## APPENDIX B PARAMETER SENSITIVITY ANALYSIS

The UTC method includes two hyperparameters:  $\lambda_1$  and  $\lambda_2$ . These parameters control the weights of different order affinities. By adjusting one of these parameters, we can investigate the impact on model performance under different parameter settings. Fig. 11 shows the effects of variations in  $\lambda_1$  and  $\lambda_2$  on ACC through line graphs on Syndata2, Lymphoma, DriveFace, and Coil20 datasets.

From Fig. 11(a) and (b), it can be observed that complex metrics, i.e., high-order affinities, can better describe complex data structures. The sensitivity to parameters was not evident on the synthetic dataset, as good performance was achieved under different parameter settings. Analyzing the experimental results on real datasets as shown in Fig. 11(c)–(h), the following observations were made: when  $\lambda_1$  or  $\lambda_2$  is too small or too large, the clustering performance shows a certain degree of decline and instability. However, the model can achieve the best performance under appropriate parameter settings. These observations support two conclusions: 1) different order affinities play important roles in guiding embedding learning and 2) different order similarities complement each other, and only by combining multiple order similarities can the model learn the best embeddings to enhance clustering performance.

## REFERENCES

- [1] D. Cheng, J. Huang, S. Zhang, S. Xia, G. Wang, and J. Xie, “K-means clustering with natural density peaks for discovering arbitrary-shaped clusters,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 8, pp. 11077–11090, Aug. 2023.
- [2] Z. Li, F. Nie, X. Chang, Y. Yang, C. Zhang, and N. Sebe, “Dynamic affinity graph construction for spectral clustering using multiple features,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6323–6332, Dec. 2018.
- [3] H. Liu, J. Chen, J. Dy, and Y. Fu, “Transforming complex problems into K-means solutions,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 7, pp. 9149–9168, Jul. 2023.
- [4] S. Pei, H. Chen, F. Nie, R. Wang, and X. Li, “Centerless clustering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 167–181, Jan. 2023.
- [5] M. Chen and X. Li, “Robust matrix factorization with spectral embedding,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5698–5707, Dec. 2021.
- [6] F. Nie, J. Xue, R. Wang, L. Zhang, and X. Li, “Fast clustering by directly solving bipartite graph clustering problem,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 7, pp. 9174–9185, Jul. 2022.
- [7] H. Li et al., “Modern deep learning in bioinformatics,” *J. Mol. Cell Biol.*, vol. 12, no. 11, pp. 823–827, Jun. 2020.
- [8] T. Tian, J. Wan, Q. Song, and Z. Wei, “Clustering single-cell RNA-seq data with a model-based deep learning approach,” *Nature Mach. Intell.*, vol. 1, no. 4, pp. 191–198, Apr. 2019.
- [9] X. Peng, J. Feng, J. T. Zhou, Y. Lei, and S. Yan, “Deep subspace clustering,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 12, pp. 5509–5521, Dec. 2020.
- [10] Y. Lu, Y.-M. Cheung, and Y. Y. Tang, “Self-adaptive multiprototype-based competitive learning approach: A k-means-type algorithm for imbalanced data clustering,” *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1598–1612, Mar. 2021.
- [11] H. Li, “Deep learning for natural language processing: Advantages and challenges,” *Nat. Sci. Rev.*, vol. 5, no. 1, pp. 24–26, Jan. 2018.
- [12] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, “On the surprising behavior of distance metrics in high dimensional space,” in *Proc. Int. Conf. Database Theory*. Berlin, Germany: Springer-Verlag, 2001, pp. 420–434.
- [13] Y. Chen et al., “KNN-BLOCK DBSCAN: Fast clustering for large-scale data,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 6, pp. 3939–3953, Jun. 2021.
- [14] X. Xu, J. Li, M. Zhou, J. Xu, and J. Cao, “Accelerated two-stage particle swarm optimization for clustering not-well-separated data,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 11, pp. 4212–4223, Nov. 2020.
- [15] S. Sarkar and A. K. Ghosh, “On perfect clustering of high dimension, low sample size data,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 9, pp. 2257–2272, Sep. 2020.
- [16] P. Borysov, J. Hannig, and J. S. Marron, “Asymptotics of hierarchical clustering for growing dimension,” *J. Multivariate Anal.*, vol. 124, pp. 465–479, Feb. 2014.
- [17] P. Hall, J. S. Marron, and A. Neeman, “Geometric representation of high dimension, low sample size data,” *J. Roy. Stat. Soc. B, Stat. Methodol.*, vol. 67, no. 3, pp. 427–444, Jun. 2005.
- [18] Q. Wang, Z. Qin, F. Nie, and X. Li, “Spectral embedded adaptive neighbors clustering,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1265–1271, Apr. 2019.
- [19] F. Nie, X. Wang, and H. Huang, “Clustering and projected clustering with adaptive neighbors,” in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 977–986.
- [20] Z. Fu, Y. Zhao, D. Chang, X. Zhang, and Y. Wang, “Double low-rank representation with projection distance penalty for clustering,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5320–5329.
- [21] L. Li et al., “Local sample-weighted multiple kernel clustering with consensus discriminative graph,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 2, pp. 1721–1734, Feb. 2022.
- [22] S. Zhou et al., “Multiple kernel clustering with neighbor-kernel subspace segmentation,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 4, pp. 1351–1362, Apr. 2020.
- [23] X. Liu, “SimpleMKKM: Simple multiple kernel K-means,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 5174–5186, Apr. 2023.
- [24] H. Wang, G. Xiao, Y. Yan, and D. Suter, “Searching for representative modes on hypergraphs for robust geometric model fitting,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 3, pp. 697–711, Mar. 2019.
- [25] C. H. Nguyen and H. Mamitsuka, “Learning on hypergraphs with sparsity,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2710–2722, Aug. 2021.
- [26] D. Zhou, J. Huang, and B. Schölkopf, “Learning with hypergraphs: Clustering, classification, and embedding,” in *Proc. 19th Int. Conf. Neural Inf. Process. Syst.*, vol. 19, 2006, pp. 1601–1608.
- [27] S. Agarwal, J. Lim, L. Zelnik-Manor, P. Perona, D. J. Kriegman, and S. J. Belongie, “Beyond pairwise clustering,” in *Proc. Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2005, pp. 838–845.
- [28] Z. Zhang, H. Lin, and Y. Gao, “Dynamic hypergraph structure learning,” in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3162–3169.
- [29] G. Li, L. Qi, and G. Yu, “The Z-eigenvalues of a symmetric tensor and its application to spectral hypergraph theory,” *Numer. Linear Algebra Appl.*, vol. 20, no. 6, pp. 1001–1029, Dec. 2013.
- [30] D. Ghoshdastidar and A. Dukkipati, “Spectral clustering using multi-linear SVD: Analysis, approximations and applications,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 29, no. 1, 2015, pp. 2610–2616.
- [31] H. Peng, Y. Hu, J. Chen, H. Wang, Y. Li, and H. Cai, “Integrating tensor similarity to enhance clustering performance,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2582–2593, May 2022.
- [32] S. Rabanser, O. Shchur, and S. Günnemann, “Introduction to tensor decompositions and their applications in machine learning,” 2017, *arXiv:1711.10781*.
- [33] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
- [34] A. Kumar, P. Rai, and H. Daumé III, “Co-regularized multi-view spectral clustering,” in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, Dec. 2011, pp. 1413–1421.
- [35] W. Liang et al., “Multi-view spectral clustering with high-order optimal neighborhood Laplacian matrix,” *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 7, pp. 3418–3430, Jul. 2022.
- [36] P. Zhou, B. Sun, X. Liu, L. Du, and X. Li, “Active clustering ensemble with self-paced learning,” *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 15, 2023, doi: 10.1109/TNNLS.2023.3252586.
- [37] P. Zhou, L. Du, and X. Li, “Adaptive consensus clustering for multiple K-means via base results refining,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 10, pp. 10251–10264, Oct. 2023.



**Hongmin Cai** (Senior Member, IEEE) received the B.S. and M.S. degrees in mathematics from Harbin Institute of Technology, Harbin, China, in 2001 and 2003, respectively, and the Ph.D. degree in applied mathematics from The Hong Kong University, Hong Kong, in 2007.

He was a Guest Professor at Kyoto University, Kyoto, Japan. He is currently a Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. His areas of research interests include bioinformatics and artificial intelligence (AI) for medical science.



**Bin Hu** (Fellow, IEEE) received the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Science, Beijing, China, in 1998.

He is currently a Professor and the Dean of the School of Medical Technology, Beijing Institute of Technology, Beijing. He has co-authored more than 400 publications. His research interests include using computational approaches to decode emotion, mind, and behavior.

Dr. Hu is a fellow of Asia-Pacific Artificial Intelligence Association (AAIA) and The Institution of Engineering and Technology (IET). He serves as the Editor-in-Chief for IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS.



**Fei Qi** received the B.S. and M.S. degrees from Xiamen University, Xiamen, China, in 2013 and 2016, respectively. He is currently pursuing the Ph.D. degree in computer science and engineering with the South China University of Technology, Guangzhou, China.

He is currently a Lecturer with the School of Data Science and Information Engineering, Guizhou Minzu University, Guiyang, China. His research interests include machine learning and image processing.



**Yue Zhang** received the Ph.D. degree in computer science from Hong Kong Baptist University, Hong Kong, SAR, China, in 2017.

She is currently an Associate Professor with the School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou, China. Her research interests include bioinformatics and big data mining.



**Junyu Li** received the B.S. and M.S. degrees from Guangdong University of Technology, Guangzhou, China, in 2017 and 2020, respectively. He is currently pursuing the Ph.D. degree in computer science and engineering with the South China University of Technology, Guangzhou.

His research interests include machine learning and image processing.



**Yiu-Ming Cheung** (Fellow, IEEE) received the Ph.D. degree from the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong, in 2000.

He is currently a Chair Professor with the Department of Computer Science, Hong Kong Baptist University, Hong Kong. His current research interests include machine learning and visual computing, as well as their applications in data science, pattern recognition, multiobjective optimization, and information security.

Dr. Cheung is a fellow of American Association for the Advancement of Science (AAAS), The Institution of Engineering and Technology (IET), and British Computer Society (BCS). He is the Founding Chairperson of the Computational Intelligence Chapter of the IEEE Hong Kong Section. He is the Editor-in-Chief of IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE. Also, he has served as an Associate Editor for IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, *Pattern Recognition*, *Pattern Recognition Letters*, *Knowledge and Information Systems*, and *Neurocomputing*, just to name a few.



**Yu Hu** received the Bachelor of Science degree in electrical engineering and automation from the School of Information and Electric Engineering, China University of Mining and Technology, Xuzhou, China, in 2017, and the Ph.D. degree in computer science from the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China, in 2023.

He is currently a Research Associate with Guangdong Artificial Intelligence and Digital Economy Laboratory, Pazhou Lab, Guangzhou. His current research endeavors revolve around the domains of reinforcement learning, test-time domain adaptation, and life-long learning.