Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Fast multi-view clustering via tensor hyperbolic tangent-p norm minimization

Yongbo Yu [a], Zhoumin Lu [b], Jingjing Xue [c], Rong Wang [a], Zongcheng Miao [a], Feiping Nie [a,*]

[a] School of Artifcial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an, 710072, Shaanxi, PR China
[b] School of Computer Science, Northwestern Polytechnical University, Xi'an, 710072, Shaanxi, PR China
[c] School of Telecommunications Engineering, Xidian University, Xi'an, 710071, Shaanxi, PR China

ARTICLE INFO

Keywords:
Fast multi-view clustering
Anchor representation learning
Low-rank tensor representation
Tensor hyperbolic tangent-p norm

ABSTRACT

Tensor-based multi-view clustering methods have gained significant attention due to their ability to directly capture high-order information, often outperforming matrix-based approaches. However, these methods face challenges in efficiently processing large-scale datasets due to their high computational complexity. Moreover, most existing tensor-based approaches rely on the tensor nuclear norm (TNN) to approximate the tensor rank function. However, TNN penalizes larger singular values, which are essential for preserving critical structural information, thus constraining the extraction of multi-view information. To address these challenges, we propose a novel fast multi-view clustering method via tensor hyperbolic tangent-p norm minimization. First, we incorporate an efficient anchor selection strategy and construct tensors from anchor-based representations, significantly reducing the computational burden of tensor-based approaches for large-scale datasets. Second, we introduce the tensor hyperbolic tangent-p norm (THT$_p$N), a more robust and accurate approximation of the tensor rank function, enabling improved extraction of multi-view consistency and complementarity. Extensive experiments on eight real-world datasets show that our proposed model not only surpasses tensor-based methods in clustering performance but also outperforms matrix-based methods in computational efficiency, establishing a new benchmark for fast multi-view clustering. Code is available at https://github.com/usualheart/FTHMC.

1. Introduction

With the continuous development of modern sensor technologies, multi-view data has become increasingly prevalent. Multi-view data exhibits significant diversity [1], encompassing different modalities of the same object (e.g., audio, images, and textual descriptions) [2], various perspectives of the same object (e.g., facial photos captured from different angles) [3], or distinct types of features extracted from the same image [4]. Data from multiple views contain both distinctive information and consistent characteristics. A key challenge in multi-view clustering methods is how to explore both the diversity and consistency of multi-view data to improve clustering performance [1,5].

Based on different computational approaches, multi-view clustering can be divided into matrix-based methods and tensor-based methods. Matrix-based methods typically aim to merge the affinity matrices or self-representation matrices from multiple views into one [6,7], or bypass the merging process and directly learn a unified class indicator matrix as the clustering result [8,9]. Tensor-based multi-view clustering methods concatenate the matrices of multiple views into a three-dimensional tensor [10,11], capturing multi-view consistency information directly at the tensor level and producing the clustering results. Beyond this fundamental approach, advanced tensor-based techniques such as ATTN [3], which adaptively learns tensor decomposition structures, and MERA [2], which utilizes multi-scale entanglement renormalization, provide more nuanced mechanisms for capturing complex intra- and inter-view information. RIMC [12] addresses the challenge of incomplete multi-view clustering by constructing tensor graphs and learning cross-view low-rank structures. The superior performance of these methods stems from their inherent ability to model higher-order correlations across views through tensor operations. However, the tensor-based approaches typically involve high computational complexity because they require transforming data to the frequency domain and performing multiple SVD decompositions [13], making them feasible only for small- to medium-scale datasets.

[*] Corresponding author.
E-mail addresses: yongboyu@mail.nwpu.edu.cn (Y. Yu), walker.zhoumin.lu@gmail.com (Z. Lu), jingjing_xue@mail.nwpu.edu.cn (J. Xue), wangrong07@tsinghua.org.cn (R. Wang), miaozongcheng@nwpu.edu.cn (Z. Miao), feipingnie@gmail.com (F. Nie).

https://doi.org/10.1016/j.patcog.2025.112195
Received 22 December 2024; Received in revised form 8 July 2025; Accepted 21 July 2025
Available online 24 July 2025

With the increasing number of multi-view data samples in recent years, it has become necessary to develop multi-view clustering algorithms that can run on large-scale datasets. Existing large-scale multi-view clustering methods often employ anchor strategies to reduce computational complexity and enhance algorithm efficiency. Anchor-based clustering methods typically involve two steps: acquiring anchors and constructing bipartite graph $B \in \mathbb{R}^{m\times n}$. Based on the method of acquiring anchors, these methods can be divided into two categories: methods that directly select anchors (e.g., LMVSC [14], SFMC [15], TBGL [16], OrthNTF [17]) and methods that dynamically learn anchors (e.g., FPMVS_CAG [18], EMVGCLG [19], FMVACC [20], and FDAGF [21]). Recent advancements also explore global feature insights via pick-and-place transforms [5], and entropy-induced anchor learning to enhance bipartite graph reliability [22]. These methods have made some progress in reducing algorithmic complexity but still face several challenges. For example, the speed and robustness of direct selection-based methods need further improvement, while dynamic learning-based methods introduce additional computation during the anchor update process. This leads to a trade-off between efficiency and performance, with most approaches exhibiting relatively average performance. One major reason is that most of these methods are matrix-based, and during the mapping process from multi-view matrices to a consensus matrix, information loss is inevitable, which limits clustering performance. For tensor-based methods that incorporate anchor mechanisms, such as TBGL and OrthNTF, there remains a considerable gap in computational efficiency compared to matrix-based methods.

Furthermore, existing tensor methods typically rely on the tensor nuclear norm (TNN) [13,23] to approximate the tensor rank function. Nevertheless, TNN differs notably from the tensor rank function, leading to excessive penalization of larger singular values [13]. Such values often represent important structural information, and excessive penalization of these values limits the ability to learn low-rank tensor representations. To address this issue, recent studies have proposed alternative tensor rank approximations, such as the tensor schatten-p norm [17] and the tensor logarithmic schatten-p norm (TLS$_p$N) [13]. However, these methods still have some limitations, including the unbounded nature of the functions and their inability to provide a compact approximation of the tensor rank function. Moreover, some methods only perform well in specific cases, making them highly sensitive to the choice of hyperparameters. Additionally, while data uncertainty (e.g., fuzziness and inconsistency) [24] is a crucial factor affecting clustering performance in real-world scenarios, explicitly modeling it within tensor-based frameworks remains an open challenge.

To address the challenges mentioned above, we propose the tensor hyperbolic tangent-p norm (THT$_p$N), which serves as a more accurate and robust approximation of the tensor rank function. This approach facilitates the learning of superior low-rank tensor representations, enabling a more effective extraction of multi-view consistency and complementary information. Furthermore, by integrating THT$_p$N with an anchor-based representation mechanism, we propose a novel fast multi-view clustering method, named fast tensor hyperbolic tangent multi-view clustering (FTHMC). The key contributions of our work are as follows:

- We propose a novel tensor-based fast multi-view clustering method, FTHMC. By leveraging fast anchor selection and anchor representation tensor construction, our approach significantly reduces computational complexity, making FTHMC highly efficient for various large-scale datasets.
- We introduce THT$_p$N, a more accurate and robust approximation of the tensor rank function. Compared to existing methods, THT$_p$N demonstrates superior robustness and aids in learning better low-rank tensor representations, enabling FTHMC to effectively capture consistency and complementarity across views.
- We develop an efficient optimization algorithm for to the proposed method. Extensive experiments on eight real-world datasets show that FTHMC not only delivers superior clustering performance compared to tensor-based methods but also achieves higher computational efficiency than fast multi-view clustering approaches.

2. Related work

2.1. Anchor-based fast multi-view clustering

With the increasing volume of multi-view data samples, there is a growing need for algorithms capable of handling large-scale multi-view clustering [14]. To reduce the algorithm's complexity to meet this demand, anchor strategies are widely used [14–21]. Acquiring anchor points is a crucial step in anchor-based clustering methods, as the quality of the anchors directly determines the model's performance. The process of acquiring anchors can be further divided into two approaches: direct selection of anchors and dynamic learning of anchors. Methods such as LMVSC [14], SFMC [15], TBGL [16], and OrthNTF [17] follow the direct anchor selection approach. Among them, LMVSC selects representative anchors from multiple views using k-means, and uses these anchors as a dictionary to reconstruct the original samples, learning the bipartite graph. SFMC, TBGL, and OrthNTF first concatenate multi-view features and then select anchors via k-means or DAS [15], followed by constructing the bipartite graph. Methods like FPMVS_CAG [18], EMVGCLG [19], FMVACC [20], and FDAGF [21] fall into the category of dynamic anchor learning, where orthogonality constraints are imposed on the anchor matrix to make the learned anchors more discriminative.

2.2. Tensor-based multi-view clustering

Tensor methods, with their inherent advantage in capturing higher-order correlations, have been widely applied to multi-view clustering in recent years [17,25,26]. These methods can be formally expressed as follows:

$$\begin{aligned}&\min_{\mathbf{Z}^{(v)},\mathbf{E}^{(v)}} \mathcal{T}(\mathcal{Z}) + \lambda \sum_{v=1}^{V} \mathcal{L}\big(\mathbf{E}^{(v)}\big)\\ &\mathcal{Z} = \Phi\big(\mathbf{Z}^{(1)},\ldots,\mathbf{Z}^{(V)}\big), \quad v = 1,2,\ldots,V,\end{aligned} \tag{1}$$

where $\mathbf{Z}^{(v)}$ can be the similarity matrix, self-representation matrix, or feature matrix for view v, and $\Phi(\cdot)$ represents the operation of concatenating the $\mathbf{Z}^{(v)}$ matrices from multiple views into a three-dimensional tensor. $\mathcal{T}(\cdot)$ is the approximation of the tensor rank function. $\mathbf{E}^{(v)}$ represents the error matrix, and λ is the trade-off parameter. The key to the success of tensor methods is learning a low-rank representation of the tensor $\mathcal{Z}$ through $\mathcal{T}(\cdot)$, thereby uncovering higher-order multi-view consistency information. Since tensor methods process multi-view data directly in tensor form, avoiding the information loss that occurs when multiple view matrices are mapped to a single matrix, they generally have an advantage over matrix-based methods [13]. However, the majority of tensor methods are limited by high computational complexity, making them feasible only for small- to medium-scale datasets, which restricts the applicability of tensor methods [27].

Furthermore, existing tensor methods typically use TNN as an approximation of the tensor rank function [23,28]. However, TNN differs notably from the tensor rank function and over-penalizes larger singular values, thereby limiting the ability to learn low-rank tensor representations. To better approximate the tensor rank function, WTNN-MSC [29] proposes the weighted tensor nuclear norm [30], which assigns different weights to different singular values; TLS$_p$NM-MSC [13] introduces the tensor logarithmic schatten-p norm (TLS$_p$N), which reduces the penalty on large singular values using a logarithmic function; and OrthNTF [17] introduces the tensor schatten-p norm as a better approximation than TNN. However, these methods still do not fully resolve the issue. For example, the functions in these methods are unbounded

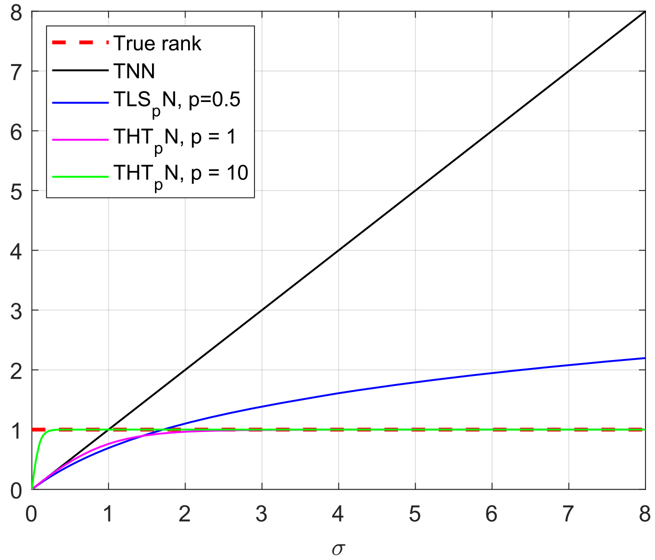


Fig. 1. Comparison of various tensor rank func approximation methods. $\mathrm{THT}_p\mathrm{N}$ demonstrates superior performance in approximating the rank compared to TNN and TLSpN. When singular values are small, $\mathrm{THT}_p\mathrm{N}$ increases rapidly and monotonically, while it saturates and asymptotically approaches 1 as $\sigma \to \infty$. This characteristic prevents excessive penalization of large singular values and aligns more closely with the true rank function. In contrast, TNN exhibits a larger deviation from the rank function, and TLSpN cannot ensure boundedness, continuing to deviate from the rank function for large singular values.

and do not provide compact approximations of the tensor rank function. Moreover, some of these methods only perform well in specific cases, making them highly sensitive to hyperparameter choices.

2.3. Notations and preliminaries

In this paper, we use bold lowercase letters to represent vectors, such as $\mathbf{x}$; bold uppercase letters to represent matrices, such as $\mathbf{X}$; and bold calligraphic fonts to represent three-dimensional tensors, such as $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$. For the elements of a tensor $\mathcal{X}$, we use lowercase letters a_{ijk} to denote them. Additionally, $\mathcal{X}^{(i)}$ refers to the ith slice of the tensor $\mathcal{X}$, while $\bar{\mathcal{X}}$ denotes its discrete fast fourier transform (FFT) along the third dimension, expressed as $\bar{\mathcal{X}} = \mathrm{fft}(\mathcal{X}, [], 3)$. Thus, the original tensor $\mathcal{X}$ can be recovered by performing an inverse FFT on $\bar{\mathcal{X}}$, i.e., $\mathcal{X} = \mathrm{ifft}(\bar{\mathcal{X}}, [], 3)$.

Next, we introduce some commonly used tensor operations [31] based on the tensors $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$ and $\mathcal{Y} \in \mathbb{R}^{n_2\times n_4\times n_3}$:

- **bdiag**$(\overline{\mathcal{X}})$: Represents the block diagonal matrix of tensor $\mathcal{X}$, i.e.,

$$\mathbf{bdiag}(\overline{\mathcal{X}}) = \mathrm{diag}\left(\overline{\mathbf{X}}^{(1)}; \overline{\mathbf{X}}^{(2)}; \dots; \overline{\mathbf{X}}^{(n_3)}\right). \quad (2)$$

- **bcric**$(\mathcal{X})$: For tensor $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$, its block circulant matrix is given by:

$$\mathbf{bcric}(\mathcal{X}) := \begin{bmatrix} \mathcal{X}^{(1)} & \mathcal{X}^{(n_3)} & \dots & \mathcal{X}^{(2)} \\ \mathcal{X}^{(2)} & \mathcal{X}^{(1)} & \dots & \mathcal{X}^{(3)} \\ \vdots & \ddots & \ddots & \vdots \\ \mathcal{X}^{(n_3)} & \mathcal{X}^{(n_3-1)} & \dots & \mathcal{X}^{(1)} \end{bmatrix}. \quad (3)$$

Definition 1 ((t-SVD) [31]). Let $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$ be a tensor, then its t-SVD (tensor singular value decomposition) can be expressed as:

$$\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top, \quad (4)$$

where $\mathcal{U} \in \mathbb{R}^{n_1\times n_1\times n_3}$ and $\mathcal{V} \in \mathbb{R}^{n_2\times n_2\times n_3}$ are orthogonal tensors, and $\mathcal{S} \in \mathbb{R}^{n_1\times n_2\times n_3}$ is an f-diagonal tensor.

3. Proposed method

3.1. Anchor selection

Classic subspace clustering methods learn self-representations based on all samples, obtaining an $n\times n$ self-representation matrix, which is then used for spectral clustering. However, for large-scale datasets, the $n\times n$ matrix presents significant computational and storage challenges, making it difficult for many self-representation based methods to work. For tensor-based methods that require processing multiple self-representation matrices, this results in even higher computational complexity. To solve these problems, a better approach is to perform representation learning based on m representative samples. This leads to an $m\times n$ anchor representation matrix, where $m \ll n$, significantly reducing the computational complexity of subsequent processing. The key to this approach lies in how to select representative samples, i.e., the anchor selection problem (Fig. 2).

Random sampling [32], and k-means [33] are common methods for selecting anchors. Random sampling is the fastest method, but the quality of the selected samples tends to be poor and unstable. Since random sampling completely ignores sample similarity and class information, it easily leads to non-representative samples. K-means, on the other hand, selects representative samples by clustering the data. However, the time complexity of k-means is $O(ndmt)$, where d, m, and t are the number of dimensions, anchors, and iterations, respectively. This makes it less ideal for large-scale datasets. In contrast to the aforementioned methods, we use a more efficient anchor selection method based on balanced and hierarchical k-means (BKHK) [34]. BKHK applies a divide-and-conquer strategy and iteratively divides the samples to form a balanced binary tree structure, with a time complexity of only $O(nd\log(m)t)$. In practice, BKHK is faster than k-means while still ensuring the representativeness of the selected anchors. Here, we use BKHK to divide the multi-view data $\{\mathbf{X}^{(v)}\}_{v=1}^{V}$ into m clusters and use the cluster centers as anchors, obtaining the multi-view anchor set $\{\mathbf{A}^{(v)}\}_{v=1}^{V}$.

3.2. Tensor hyperbolic tangent-*p* norm

Inspired by activation functions in deep learning [35], we propose a novel and simple tensor average rank function approximation, which is formally defined as follows:

Definition 2. For a tensor $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$, the Tensor Hyperbolic Tangent-p Norm ($\mathrm{THT}_p\mathrm{N}$) is defined as:

$$\|\mathcal{X}\|_{THT_pN} = \frac{1}{n_3}\sum_{k=1}^{n_3}\|\bar{\mathcal{X}}_f^{(k)}\|_{tanh\_p} \quad = \frac{1}{n_3}\sum_{k=1}^{n_3}\sum_{i=1}^{h}\frac{e^{p\mathcal{S}_f^{(k)}(i,i)} - e^{-p\mathcal{S}_f^{(k)}(i,i)}}{e^{p\mathcal{S}_f^{(k)}(i,i)} + e^{-p\mathcal{S}_f^{(k)}(i,i)}} \quad (5)$$

where $p > 0$, $h = \min(n_1, n_2)$, and $\mathcal{S}_f$ is an f-diagonal tensor obtained through the t-SVD of $\bar{\mathcal{X}} = \mathcal{U}_f\mathcal{S}_f\mathcal{V}_f{}^T$ in the Fourier domain, with $\mathcal{U}_f$ and $\mathcal{V}_f$ being orthogonal tensors, as defined in Definition 1.

The approximate function used by $\mathrm{THT}_p\mathrm{N}$ is $f_{tanh\_p}(x) = \tanh(px) = \frac{e^{px}-e^{-px}}{e^{px}+e^{-px}}$, which is inspired by the hyperbolic tangent function. Essentially, the condition $f_{tanh\_p}(0) = 0$ is satisfied, which is consistent with the true rank function. We compare $\mathrm{THT}_p\mathrm{N}$ with other existing methods, such as TNN [23] and $\mathrm{TLS}_p\mathrm{N}$ [13], as shown in Fig. 1. $\mathrm{THT}_p\mathrm{N}$ demonstrates the following advantages:

- **Unbiased and Compact Tensor Rank Function Approximation**: For singular values in the range $[0, +\infty]$, the range of $\mathrm{THT}_p\mathrm{N}$ is $[0, 1]$. Specifically, for smaller singular values, $\mathrm{THT}_p\mathrm{N}$ increases monotonically and rapidly, and it gradually saturates, approaching 1 as $\sigma \to \infty$. This behavior closely follows the curve of the rank function and effectively avoids excessive penalization for large singular values. In contrast, TNN exhibits a large discrepancy from the rank function, especially in the region of large singular values, while $\mathrm{TLS}_p\mathrm{N}$ fails to ensure boundedness.

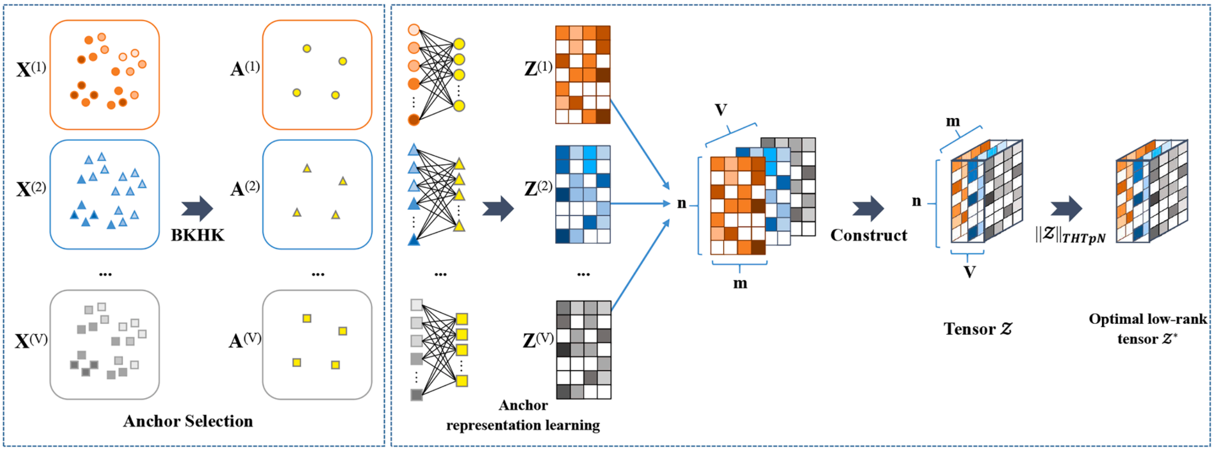

X(1)
X(2)
X(V)
A(1)
A(2)
A(V)
BKHK
Anchor Selection
Z(1)
Z(2)
Z(V)
Anchor representation learning
V
n
m
Construct
Tensor $\mathcal{Z}$
$\|\mathcal{Z}\|_{THTpN}$
Optimal low-rank tensor $\mathcal{Z}^*$

Fig. 2. The flowchart of the proposed FTHMC method. Given multi-view data $\{\mathbf{X}^{(v)}\}_{v=1}^{V}$, anchors are first selected using the BKHK method, resulting in the anchor dataset $\{\mathbf{A}^{(v)}\}_{v=1}^{V}$ composed of m samples. Then, anchor representation learning is conducted to obtain the multi-view anchor representation matrices $\{\mathbf{Z}^{(v)}\}_{v=1}^{V}$. These anchor representation matrices are concatenated into a tensor $\mathcal{Z}$. By minimizing $\|\mathcal{Z}\|_{THT_pN}$, an optimal low-rank tensor representation is learned to capture multi-view consistency and complementary information.

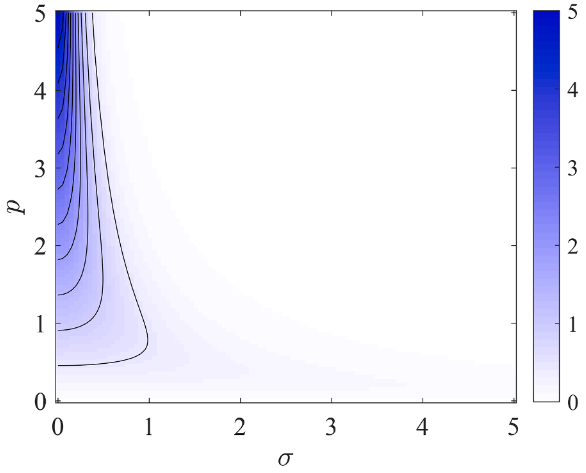


Fig. 3. Heatmap illustrating the THT$_p$N penalty weight distribution across varying p-values and singular values σ. The color gradient represents the corresponding penalty weight $w(p, \sigma) = p(1 - \tanh^2(p\sigma))$. Increasing the p-value results in a larger penalty weight for smaller singular values, while a smaller p-value ensures a more balanced distribution of penalty weights across different singular values.

- **More Stable Tensor Rank Function Approximation**: Regardless of the chosen p-value, THT$_p$N is a monotonically increasing function with a range of $[0, 1]$ that approaches 1, making it more stable and robust compared to other methods. Subsequent experimental results confirm this, as THT$_p$N is not sensitive to the selection of p-values, contributing to the robustness of multi-view clustering methods.
- **Flexible Singular Value Penalty Adjustment**: As illustrated in Fig. 3, a heatmap is presented to depict the distribution of THT$_p$N penalty weights across varying p-values and singular values σ, specifically the gradient function of $\tanh(p\sigma)$, defined as $w(p, \sigma) = p(1 - \tanh^2(p\sigma))$. It is observed that by modifying the p-value, the distribution of penalty weights can be adaptively adjusted across singular values of different magnitudes. This capability allows THT$_p$N to tailor the penalty strategy effectively, thereby enhancing its adaptability to diverse scenarios.

3.3. Objective function formulation

By combining anchor representation learning and THT$_p$N, we propose the fast multi-view clustering with tensor hyperbolic tangent-p norm minimization method (FTHMC), formalized as follows:

$$\begin{aligned} &\min_{\mathbf{Z}^{(v)}} \|\mathcal{Z}\|_{THT_pN} + \lambda \sum_{v=1}^{V} \|\mathbf{E}^{(v)}\|_{2,1} \\ &s.t. \mathbf{E}^{(v)} = \mathbf{X}^{(v)} - \mathbf{A}^{(v)}\mathbf{Z}^{(v)},\ v = 1, \dots, V, \\ &\quad \mathbf{X}^{(v)} \in \mathbb{R}^{d_v \times n},\ \mathbf{A}^{(v)} \in \mathbb{R}^{d_v \times m},\ \mathbf{Z}^{(v)} \in \mathbb{R}^{m \times n} \end{aligned} \tag{6}$$

where $\mathbf{X}^{(v)}$, $\mathbf{Z}^{(v)}$, and $\mathbf{E}^{(v)}$ represent the feature matrix, the anchor representation matrix, and the reconstruction error matrix for the v^{th} view, respectively. $\mathcal{Z} \in \mathbb{R}^{m \times V \times n}$ is a three-way tensor constructed by concatenating multiple anchor representation matrices $\{\mathbf{Z}^{(v)}\}_{v=1}^{V}$. λ is a balancing parameter that controls the influence of the anchor representation learning loss.

Unlike subspace learning, here we perform anchor representation learning based on the representative set of m samples $\{\mathbf{A}^{(v)}\}_{v=1}^{V}$, obtaining the multi-view anchor representation matrices $\{\mathbf{Z}^{(v)}\}_{v=1}^{V}$. Compared to the $n \times n$ self-representation matrix, the $m \times n$ anchor representation matrix $\{\mathbf{Z}^{(v)}\}_{v=1}^{V}$ reduces the overall computational complexity, making the model efficient even for large-scale datasets. By minimizing the THT$_p$N regularization on the tensor $\mathcal{Z}$, we can learn a low-rank representation of $\mathcal{Z}$, which helps fully exploit the complementary and consistent information between the multi-view anchor representation matrices.

After obtaining the optimal multi-view anchor representation matrices $\{\mathbf{Z}^{(v)}\}_{v=1}^{V}$, the standard approach constructs the consensus affinity matrix $\mathbf{S} = \frac{1}{V}\sum_{v=1}^{V} \mathbf{Z}^{(v)\top}\mathbf{Z}^{(v)}$ for spectral clustering. However, explicitly computing $\mathbf{S}$ requires $\mathcal{O}(n^2 V)$ operations and $\mathcal{O}(n^2)$ storage, which becomes prohibitive for large n. To overcome this limitation, we define the concatenated anchor representation matrix $\bar{\mathbf{Z}} = \frac{1}{\sqrt{V}}\left[\mathbf{Z}^{(1)\top}, \mathbf{Z}^{(2)\top}, \dots, \mathbf{Z}^{(V)\top}\right] \in \mathbb{R}^{n \times mV}$. Theorem 1 establishes that the spectral embedding $\mathbf{Q}$ (top c eigenvectors of $\mathbf{S}$) can be obtained directly from $\bar{\mathbf{Z}}$ without constructing $\mathbf{S}$ explicitly.

Theorem 1. *Given the concatenated anchor representation matrix* $\bar{\mathbf{Z}} = \frac{1}{\sqrt{V}}\left[\mathbf{Z}^{(1)\top}, \mathbf{Z}^{(2)\top}, \dots, \mathbf{Z}^{(V)\top}\right]$, *the spectral embedding* $\mathbf{Q} \in \mathbb{R}^{n \times c}$ *(top* c *eigenvectors of* $\mathbf{S}$*) is equivalent to the top* c *left singular vectors of* $\bar{\mathbf{Z}}$.

Proof. The equivalence follows directly from the construction of $\bar{\mathbf{Z}}$ and its spectral decomposition. First, observe that the consensus affinity matrix $\mathbf{S} = \frac{1}{V}\sum_{v=1}^{V} \mathbf{Z}^{(v)\top}\mathbf{Z}^{(v)}$ can be rewritten using $\bar{\mathbf{Z}}$:

$$\mathbf{S} = \frac{1}{V}\sum_{v=1}^{V} \mathbf{Z}^{(v)\top}\mathbf{Z}^{(v)} = \frac{1}{V}\left[\mathbf{Z}^{(1)\top} \quad \dots \quad \mathbf{Z}^{(V)\top}\right]\begin{bmatrix} \mathbf{Z}^{(1)} \\ \vdots \\ \mathbf{Z}^{(V)} \end{bmatrix} = \bar{\mathbf{Z}}\bar{\mathbf{Z}}^{\top}. \tag{7}$$

Consider the singular value decomposition $\bar{\mathbf{Z}} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$. Substituting this into the expression for $\mathbf{S}$ yields: $\mathbf{S} = \bar{\mathbf{Z}}\bar{\mathbf{Z}}^\top = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top)^\top = \mathbf{U}\boldsymbol{\Sigma}^2\mathbf{U}^\top$. This constitutes an eigendecomposition of $\mathbf{S}$ where $\mathbf{U}$ contains the eigenvectors and $\boldsymbol{\Sigma}^2$ the eigenvalues. The top c eigenvectors of $\mathbf{S}$ (columns of $\mathbf{Q}$) therefore correspond to the first c columns of $\mathbf{U}$, which are precisely the top c left singular vectors of $\bar{\mathbf{Z}}$. □

This equivalence enables efficient computation of $\mathbf{Q}$ via partial SVD on $\bar{\mathbf{Z}}$, reducing complexity from $\mathcal{O}(n^2c)$ to $\mathcal{O}(nmVc)$. Given $mV \ll n$ in anchor-based frameworks, this achieves orders-of-magnitude acceleration for large n.

3.4. Optimization process

To convert the objective function into a separable form, we introduce auxiliary variables $\mathcal{G}$ and rewrite model (6) as an unconstrained optimization problem in the following Lagrangian form:

$$\begin{aligned}
&\mathcal{L}\left(\{\mathbf{Z}^{(v)}\}, \{\mathbf{E}^{(v)}\}, \mathcal{G}, \{\mathbf{Y}^{(v)}\}, \mathcal{W}, \mu\right) \\
&= \|\mathcal{G}\|_{THT_pN} + \lambda \sum_{v=1}^{V} \|\mathbf{E}^{(v)}\|_{2,1} \\
&+ \sum_{v=1}^{V} \langle \mathbf{Y}^{(v)}, \mathbf{X}^{(v)} - \mathbf{E}^{(v)} - \mathbf{A}^{(v)}\mathbf{Z}^{(v)} \rangle + \langle \mathcal{W}, \mathcal{Z} - \mathcal{G} \rangle \\
&+ \sum_{v=1}^{V} \frac{\mu}{2} \|\mathbf{X}^{(v)} - \mathbf{E}^{(v)} - \mathbf{A}^{(v)}\mathbf{Z}^{(v)}\|_F^2 + \frac{\mu}{2}\|\mathcal{Z} - \mathcal{G}\|_F^2,
\end{aligned} \tag{8}$$

where $\mathcal{W}$ and $\{\mathbf{Y}^{(v)}\}_{v=1}^{V}$ denote the Lagrange multipliers, and μ represents a positive penalty parameter that helps regulate the convergence of the algorithm. We solve the Eq. (8) for each variable via the following subproblems.

3.4.1. $\{\mathbf{Z}^{(v)}\}_{v=1}^{V}$-subproblem

After fixing the other variables, the variable $\mathbf{Z}^{(v)}$ can be optimized by the following subproblem:

$$\begin{aligned}
&\min_{\mathbf{Z}^{(v)}} \langle \mathbf{Y}^{(v)}, \mathbf{X}^{(v)} - \mathbf{E}^{(v)} - \mathbf{A}^{(v)}\mathbf{Z}^{(v)} \rangle + \langle \mathcal{W}^{(v)}, \mathbf{Z}^{(v)} - \mathbf{G}^{(v)} \rangle \\
&+ \frac{\mu}{2}\|\mathbf{X}^{(v)} - \mathbf{E}^{(v)} - \mathbf{A}^{(v)}\mathbf{Z}^{(v)}\|_F^2 + \frac{\mu}{2}\|\mathbf{Z}^{(v)} - \mathbf{G}^{(v)}\|_F^2.
\end{aligned} \tag{9}$$

where $\mathcal{W}^{(v)}$ and $\mathbf{G}^{(v)}$ are the vth frontal slices of $\mathcal{W}$ and $\mathcal{G}$, respectively. The problem in (9) is convex, and setting the derivative of $\mathbf{Z}^{(v)}$ to zero yields::

$$\begin{aligned}
&-\mathbf{A}^{(v)\top}\mathbf{Y}^{(v)} + \mathcal{W}^{(v)} - \mu\mathbf{A}^{(v)\top}\left(\mathbf{X}^{(v)} - \mathbf{A}^{(v)}\mathbf{Z}^{(v)} - \mathbf{E}^{(v)}\right) \\
&+ \mu\left(\mathbf{Z}^{(v)} - \mathbf{G}^{(v)}\right) = 0.
\end{aligned} \tag{10}$$

Thus, the optimal solution is:

$$\begin{aligned}
&\mathbf{Z}^{(v)} = \frac{1}{\mu}\left(\mathbf{A}^{(v)\top}\mathbf{A}^{(v)} + \mathbf{I}\right)^{-1} \\
&\left(\mu\mathbf{G}^{(v)} + \mathbf{A}^{(v)\top}\left(\mathbf{Y}^{(v)} - \mu\mathbf{E}^{(v)} + \mu\mathbf{X}^{(v)}\right) - \mathcal{W}^{(v)}\right).
\end{aligned} \tag{11}$$

3.4.2. $\{\mathbf{E}^{(v)}\}_{v=1}^{V}$-subproblem

After fixing the other variables, the subproblem involving $\mathbf{E}^{(v)}$ can be expressed as:

$$\begin{aligned}
&\min_{\mathbf{E}^{(v)}} \lambda\|\mathbf{E}^{(v)}\|_{2,1} + \langle \mathbf{Y}^{(v)}, \mathbf{X}^{(v)} - \mathbf{E}^{(v)} - \mathbf{A}^{(v)}\mathbf{Z}^{(v)} \rangle \\
&+ \frac{\mu}{2}\|\mathbf{X}^{(v)} - \mathbf{E}^{(v)} - \mathbf{A}^{(v)}\mathbf{Z}^{(v)}\|_F^2.
\end{aligned} \tag{12}$$

This can be rewritten as:

$$\min_{\mathbf{E}^{(v)}} \frac{\lambda}{\mu}\|\mathbf{E}^{(v)}\|_{2,1} + \frac{1}{2}\left\|\mathbf{E}^{(v)} - \widetilde{\mathbf{E}}^{(v)}\right\|_F^2. \tag{13}$$

where $\widetilde{\mathbf{E}}^{(v)} = \mathbf{X}^{(v)} - \mathbf{A}^{(v)}\mathbf{Z}^{(v)} + \frac{1}{\mu}\mathbf{Y}^{(v)}$, and its solution can be obtained through the $\ell_{2,1}$ minimization threshold operator [36]:

$$[\mathbf{E}^{(v)}]_{:,i} = \begin{cases} \left(1 - \dfrac{\lambda}{\mu\|\left[\widetilde{\mathbf{E}}^{(v)}\right]_{:,i}\|_2}\right)\left[\widetilde{\mathbf{E}}^{(v)}\right]_{:,i}, & \text{if } \|\left[\widetilde{\mathbf{E}}^{(v)}\right]_{:,i}\|_2 > \frac{\lambda}{\mu}; \\ 0, & \text{otherwise,} \end{cases} \tag{14}$$

where $\widetilde{\mathbf{E}}^{(v)} = \mathbf{X}^{(v)} + \frac{1}{\mu}\mathbf{Y}^{(v)} - \mathbf{A}^{(v)}\mathbf{Z}^{(v)}$, and $[\widetilde{\mathbf{E}}^{(v)}]_{:,i}$ denotes the ith column of $\widetilde{\mathbf{E}}^{(v)}$.

3.4.3. $\mathcal{G}$-subproblem

After fixing the other variables, the subproblem involving $\mathcal{G}$ can be formulated as:

$$\min_{\mathcal{G}} \|\mathcal{G}\|_{THT_pN} + \langle \mathcal{W}, \mathcal{Z} - \mathcal{G} \rangle + \frac{\mu}{2}\|\mathcal{Z} - \mathcal{G}\|_F^2. \tag{15}$$

This can be rewritten as:

$$\min_{\mathcal{G}} \frac{1}{\mu}\|\mathcal{G}\|_{THT_pN} + \frac{1}{2}\|\mathcal{G} - \widetilde{\mathcal{G}}\|_F^2. \tag{16}$$

where $\widetilde{\mathcal{G}} = \mathcal{Z} + \mathcal{W}/\mu$. This is referred to as the tensor hyperbolic tangent-p norm minimization (THT$_p$NM) problem, which is difficult to solve directly with existing tools. According to Theorem 1, we can obtain the optimal solution of the optimization problem (16). Since Eq. (20) is a combination of a convex and a concave function, we can solve its closed-form solution using the Difference of Convex (DC) programming method [37]:

$$\tau^{iter+1} = \left(S_f^{(k)}(i,i) - \frac{\partial f(\tau^{iter})}{\mu}\right)_+, \tag{17}$$

where $\tau = Prox_{f,\beta}(S_f^{(k)}(i,i))$, and $iter$ denotes the iteration count.

Theorem 2. *Assume $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ with its t-SVD decomposition $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$, and $\beta > 0$. The tensor hyperbolic tangent-p norm minimization (THT$_p$NM) problem can be described as:*

$$\arg\min_{\mathcal{G}} \quad \beta\|\mathcal{G}\|_{THT_pN} + \frac{1}{2}\|\mathcal{G} - \mathcal{A}\|_F^2. \tag{18}$$

Then, the optimal solution $\mathcal{G}^*$ can be expressed as:

$$\mathcal{G}^* = \mathcal{U} * ifft(Prox_{f,\beta}(S_f), [], 3) * \mathcal{V}^T \tag{19}$$

where $ifft(Prox_{f,\beta}(S_f), [], 3) \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an f-diagonal tensor, and $Prox_{f,\beta}(S_f)$ satisfies:

$$Prox_{f,\beta}(S_f^{(k)}(i,i)) = \arg\min_{x \geq 0} \frac{1}{2}\left(x - S_f^{(k)}(i,i)\right)^2 + \beta\tanh(px) \tag{20}$$

where $\tanh(px) = \frac{e^{px} - e^{-px}}{e^{px} + e^{-px}}$.

Proof. To prove Theorem 2, we first introduce the following lemma. □

Lemma 1 ([38]). *Let $\mathbf{G}, \mathbf{A} \in \mathbb{R}^{m \times n}$, where $\mathbf{A}$ has the singular value decomposition (SVD) $\mathbf{A} = \mathbf{U}\mathbf{S}_A\mathbf{V}^T$. Given $\beta > 0$, the optimal solution to the problem:*

$$\min_{\mathbf{G}} \beta\|\mathbf{G}\|_{THT_pN} + \frac{1}{2}\|\mathbf{G} - \mathbf{A}\|_F^2 \tag{21}$$

is $\mathbf{G}^* = \mathbf{U}\mathbf{S}_G^*\mathbf{V}^T$, where $\mathbf{S}_G^* = \mathrm{diag}(\sigma^*)$ and $\sigma^* = \mathrm{prox}_{f,\beta}(\sigma_A)$. The operator $\mathrm{prox}_{f,\beta}$, known as the Moreau-Yosida operator, is defined as:

$$\mathrm{prox}_{f,\beta}(\sigma_A) := \arg\min_{\sigma \geq 0} \beta f(\sigma) + \frac{1}{2}\left\|\sigma - \sigma_A\right\|_2^2 \tag{22}$$

with $f(x) = \tanh(px)$.

In the Fourier domain, we have $\|\mathcal{X}\|_F^2 = \frac{1}{n_3}\|\bar{\mathcal{X}}\|_F^2$. Thus, the objective function in (21) becomes:

$$\begin{aligned}
&\frac{1}{2}\|\mathcal{G} - \mathcal{A}\|_F^2 + \beta\|\mathcal{G}\|_{THT_pN} \\
&= \frac{1}{n_3}\sum_{k=1}^{n_3}\left(\frac{1}{2}\left\|\mathcal{G}_f^{(k)} - \mathcal{A}_f^{(k)}\right\|_F^2 + \beta\left\|\mathcal{G}_f^{(k)}\right\|_{THT_pN}\right)
\end{aligned} \tag{23}$$

This transforms the original tensor optimization into n_3 independent matrix problems:

$$\arg\min_{\mathcal{G}_f^{(k)}} \frac{1}{2}\left\|\mathcal{G}_f^{(k)} - \mathcal{A}_f^{(k)}\right\|_F^2 + \beta\left\|\mathcal{G}_f^{(k)}\right\|_{THT_pN}, \tag{24}$$

for $k = 1, 2, \ldots, n_3$.

Let $\mathcal{A}_f^{(k)} = \mathcal{U}_f^{(k)} \mathcal{S}_f^{(k)} (\mathcal{V}_f^{(k)})^H$ denote the SVD of $\mathcal{A}_f^{(k)}$. By Lemma 1, the optimal solution to this problem is:

$$\mathcal{G}_f^{*(k)} = \mathcal{U}_f^{(k)} \operatorname{Prox}_{f,\beta}\left(\mathcal{S}_f^{(k)}\right)\left(\mathcal{V}_f^{(k)}\right)^H, \tag{25}$$

where $\operatorname{Prox}_{f,\beta}(\mathcal{S}_f^{(k)}(i,i))$ solves:

$$\operatorname{Prox}_{f,\beta}\left(\mathcal{S}_f^{(k)}(i,i)\right) = \arg\min_{x \geq 0} \frac{1}{2}\left(x - \mathcal{S}_f^{(k)}(i,i)\right)^2 + \beta f(x). \tag{26}$$

Finally, assembling all $\mathcal{G}_f^{*k}$ and applying the inverse Fourier transform yields the optimal solution to the original problem:

$$\mathcal{G}^* = \mathcal{U} * \operatorname{ifft}\left(\operatorname{Prox}_{f,\beta}(\mathcal{S}_f), [\,], 3\right) * \mathcal{V}^T, \tag{27}$$

where $\operatorname{Prox}_{f,\beta}(\mathcal{S}_f)$ applies the proximal operator to the diagonal elements of $\mathcal{S}_f$.

Algorithm 1: FTHMC optimization algorithm.

```
Require: Multi-view data: $\{\mathbf{X}^{(v)}\}_{v=1}^V$ , hyperparameters $\lambda$ and $p$.
Set initial values: $\mathbf{Z}^{(v)} = 0$, $\mathbf{E}^{(v)} = 0$ for each $v = 1, \dots, V$.
$\mathcal{W} = \mathcal{G} = 0$; $\epsilon = 10^{-7}$;$\eta = 2$ and $\mu = 10^{-5}$.
while convergence is not reached do
    Update $\mathbf{Z}^{(v)}$ using the formula from Eq. (11);
    Update $\mathbf{E}^{(v)}$ using the formula from Eq. (14);
    Update $\mathcal{G}$ according to Eq. (19) based on Theorem 1;
    Update $\mathbf{Y}^{(v)}$, $\mathcal{W}$, and $\mu$ using Eq. (28);
    Check stopping conditions:
    $\|\mathbf{X}^{(v)} - \mathbf{A}^{(v)}\mathbf{Z}^{(v)} - \mathbf{E}^{(v)}\|_\infty < \epsilon$, $\|\mathcal{Z} - \mathcal{G}\|_\infty < \epsilon$.
end while
Compute $\mathbf{Q}$ based on Theorem 1;
Output: Run k-means on $\mathbf{Q}$ to obtain clustering results.
```

3.4.4. Updating lagrange multipliers and penalty parameters

Finally, the procedures for updating the Lagrange multipliers and penalty parameters can be expressed as:

$$\begin{cases} \mathbf{Y}^{(v)} = \mathbf{Y}^{(v)} + \mu(\mathbf{X}^{(v)} - \mathbf{A}^{(v)}\mathbf{Z}^{(v)} - \mathbf{E}^{(v)}), \\ \mathcal{W} = \mathcal{W} + \mu(\mathcal{Z} - \mathcal{G}), \\ \mu = \eta\mu, \quad \mu = \min(\mu, \mu_{\max}), \end{cases} \tag{28}$$

where $\eta > 1$ serves to accelerate the convergence. The algorithm begins by initializing all variables, followed by alternately updating them iteratively until the convergence criteria are fulfilled. Algorithm 1 summarizes the entire optimization process.

3.5. Convergence and complexity analysis

Convergence analysis. For our FTHMC model, we theoretically prove through Theorem 3 that the sequence generated by Algorithm 1 converges to a Karush-Kuhn-Tucker (KKT) critical point. The proof is provided in the supplementary material.

Theorem 3. *Let $\{\mathcal{P}_k = (\mathbf{Z}_k^{(v)}, \mathbf{E}_k^{(v)}, \mathcal{G}_k, \mathbf{Y}_k^{(v)}, \mathcal{W}_k)\}_{k=1}^\infty$ be the sequence generated by Algorithm 1. This sequence satisfies:*

1. *Boundedness: $\{\mathcal{P}_k\}_{k=1}^\infty$ is bounded*
2. *Convergence: Has at least one accumulation point $\mathcal{P}_*$ that is a KKT point*

Complexity analysis. The computational complexity of the optimization algorithm primarily stems from updating the variables $\mathbf{Z}^{(v)}$, $\mathbf{E}^{(v)}$, and $\mathcal{G}$. The update for $\mathbf{Z}^{(v)}$ has a complexity of $\mathcal{O}(m^3 + m^2 d^{(v)})$. The update for $\mathbf{E}^{(v)} \in \mathbb{R}^{d^{(v)} \times n}$ has a complexity of $\mathcal{O}(d^{(v)} n)$. Since there are V views, the complexity of updating all $\mathbf{Z}^{(v)}$ and $\mathbf{E}^{(v)}$ is $\mathcal{O}(Vm^3 + Vm^2 d + Vdn)$, where $d = \max\{d^{(v)} | v = 1, \dots, V\}$. The update of tensor $\mathcal{G} \in \mathbb{R}^{m \times V \times n}$ involves FFT and inverse FFT, as well as tensor t-SVD operations. The update complexity of tensor $\mathcal{G}$ is $\mathcal{O}(Vmn\log(n) + V^2 mn)$. Considering that $V \ll n$, the total computational complexity is $\mathcal{O}(mn\log(n) + dm^2 + dn + m^3)$.

Table 1
Summary of the dataset statistics.

Dataset	Type	Sample	Cluster	View
MSRCv1	Image	210	7	4
Hfeat	Image	2000	10	6
scene15	Image	4485	15	3
CCV	Video	6774	20	3
Caltech256fea	Image	30,607	257	3
cifar10	Image	60,000	10	3
YouTubeFace10	Video	38,654	10	4
YouTubeFace20	Video	63,896	20	4

4. Experiments

This section compares FTHMC with other relevant clustering methods on eight widely used benchmark datasets, while also evaluating its running efficiency and performance.

4.1. Experimental setup

Datasets. In our experiments, we use eight real-world datasets for evaluation. These include four small- to medium-scale scale datasets: MSRCv1 [39], Hfeat [40], scene15 [41], and CCV [42]; and four large-scale datasets: Caltech256-fea [4], cifar10 [43], YouTubeFace10 [44], and YouTubeFace20 [44]. A summary of the dataset statistics is provided in Table 1.

Baselines. We adopted thirteen multi-view clustering methods as baselines for comparison. These include six anchor-based fast multi-view clustering algorithms: LMVSC [14], SFMC [15], FPMVS_CAG [18], FMVACC [20], FDAGF [21] and EMVGCLG [19]; as well as seven tensor-based multi-view clustering algorithms: LTMSC [28], t-SVD-MSC [23], WTNNM [29], TBGL [16], TLS$_p$NM [13], OrthNTF [17] and IWTSN [45]. It is noteworthy that the tensor-based methods TBGL and OrthNTF also use anchor-based approaches. For all baseline methods, we adjusted hyperparameters according to the experimental settings in the corresponding papers to obtain the best results. Each method was executed 10 times, and the average results were recorded to eliminate the effect of randomness. All experiments were carried out on a PC equipped with an Intel i7-14700 CPU, 96GB of RAM, and a 64-bit Windows 10 operating system, running MATLAB R2023b.

Evaluation metrics. To comprehensively evaluate clustering performance, five widely used metrics were adopted: accuracy (ACC), normalized mutual information (NMI), purity (PUR), adjusted rand index (ARI) and micro F1-score (F1). The higher the values of these metrics, the better the corresponding method's performance.

Parameter settings. Our FTHMC model has two hyperparameters: λ and p, where λ controls the impact of the anchor representation learning loss, and p controls the penalty impact on singular values of different sizes in THT$_p$N. We performed grid search to adjust the hyperparameters, with both λ and p chosen from the range $\{0.001, 0.01, 0.1, 1, 10, 100\}$. For each dataset, the number of anchors m was adjusted within the range $\{2^{\lfloor \log_2 c \rfloor}, 2^{\lfloor \log_2 2c \rfloor}\}$, where c is the number of classes. This choice ensures that the anchor samples cover all classes while satisfying BKHK's requirement.

4.2. Experimental results

Tables 2 and 3 presents the clustering performance of our proposed FTHMC method compared to other methods on multiple datasets. It can be seen that FTHMC achieved the best performance across most datasets in terms of all metrics, with significant performance improvements. Several key observations are summarized as follows:

Table 2
Clustering results of the proposed FTHMC method and baselines on small- to medium-scale datasets. The top-performing results are shown in bold, while the second-best are underlined.

Dataset	metric	LMVSC	SFMC	FPMVS	FMVACC	FDAGF	EMVGCLG	LTMSC	tSVD-MSC	WTNNM	TBGL	TLS$_p$NM	OrthNTF	IWTSN	Ours
MSRCv1	ACC	0.3429	0.7571	0.7619	0.7667	0.7476	0.8857	0.7619	0.3476	0.9762	0.5238	0.9905	0.8476	0.9381	**1.0000**
	NMI	0.2460	0.7222	0.6760	0.6693	0.6517	0.7918	0.6630	0.1696	0.9498	0.5338	0.9817	0.8247	0.8825	**1.0000**
	PUR	0.4000	0.7905	0.7810	0.7667	0.7762	0.8857	0.7762	0.3571	0.9762	0.5571	0.9905	0.8476	0.9381	**1.0000**
	F1	0.2474	0.6897	0.6634	0.6465	0.6434	0.7809	0.6500	0.2138	0.9537	0.4649	0.9810	0.7740	0.8804	**1.0000**
	ARI	0.1250	0.6330	0.6075	0.5885	0.5834	0.7449	0.5909	0.0854	0.9463	0.3316	0.9779	0.7376	0.8611	**1.0000**
Hfeat	ACC	0.6560	0.8835	0.8230	0.9350	0.9350	0.9425	0.9275	0.4380	0.9990	0.8565	0.8825	0.7890	0.9360	**0.9995**
	NMI	0.6380	0.9006	0.7814	0.8719	0.8708	0.8817	0.8599	0.2646	0.9973	0.8837	0.7980	0.7408	0.8777	**0.9986**
	PUR	0.6930	0.8835	0.8230	0.9350	0.9350	0.9425	0.9275	0.4385	0.9990	0.8780	0.8825	0.8115	0.9360	**0.9995**
	F1	0.5790	0.8739	0.7556	0.8730	0.8745	0.8867	0.8631	0.2785	0.9980	0.8477	0.7881	0.7084	0.8802	**0.9990**
	ARI	0.5289	0.8590	0.7269	0.8589	0.8606	0.8741	0.8479	0.1984	0.9978	0.8293	0.7647	0.6752	0.8669	**0.9989**
scene15	ACC	0.5211	0.3124	0.5289	0.5438	0.5614	0.6054	0.5271	0.1955	0.8812	0.2082	0.9269	0.6647	0.8540	**0.9353**
	NMI	0.5131	0.3588	0.5331	0.5478	0.5561	0.6056	0.5566	0.1449	0.9178	0.1403	0.8831	0.7531	0.8468	**0.9207**
	PUR	0.5770	0.3139	0.5311	0.5946	0.6125	0.6696	0.5969	0.2149	0.9240	0.2149	0.9269	0.7168	0.8758	**0.9353**
	F1	0.4008	0.2655	0.4256	0.4221	0.4385	0.4917	0.4307	0.1351	0.8776	0.1510	0.8774	0.5992	0.8145	**0.8882**
	ARI	0.3560	0.1678	0.3704	0.3776	0.3954	0.4537	0.3869	0.0718	0.8685	0.0302	0.8683	0.5679	0.8006	**0.8798**
CCV	ACC	0.1472	0.1090	0.2401	0.1965	0.2272	0.2410	0.2212	0.2361	0.3337	0.1171	0.4983	0.4760	0.5281	**0.7477**
	NMI	0.0999	0.0146	0.1711	0.1574	0.1873	0.1806	0.1742	0.1651	0.2747	0.0379	0.4237	0.5849	0.4090	**0.7558**
	PUR	0.1835	0.1143	0.2628	0.2317	0.2585	0.2621	0.2526	0.2702	0.3570	0.1327	0.5151	0.5681	0.5422	**0.7884**
	F1	0.0957	0.1081	0.1393	0.1183	0.1298	0.1327	0.1273	0.1302	0.1999	0.1063	0.3376	0.4081	0.3612	**0.6443**
	ARI	0.0283	0.0001	0.0814	0.0675	0.0728	0.0815	0.0754	0.0803	0.1530	0.0002	0.2999	0.3726	0.3249	**0.6232**

Table 3
Clustering results of the proposed FTHMC method and baselines on four large-scale datasets. The top-performing results are shown in bold, while the second-best are underlined. '–' indicates a memory exhaustion error.

dataset	metric	LMVSC	SFMC	FPMVS	FMVACC	FDAGF	EMVGCLG	LTMSC	tSVD-MSC	WTNNM	TBGL	TLS$_p$NM	OrthNTF	IWTSN	Ours
Caltech256-fea	ACC	0.0281	0.0313	0.4872	0.4098	0.6314	0.6931	–	–	–	–	–	0.2294	0.2700	**0.7271**
	NMI	0.1683	0.0089	0.6522	0.5528	0.7612	0.7978	–	–	–	–	–	0.3893	0.4012	**0.8405**
	PUR	0.0406	0.0352	0.4917	0.4261	0.7305	0.7296	–	–	–	–	–	0.2525	0.2829	**0.7804**
	F1	0.0087	0.0117	0.4197	0.2704	0.5363	0.4778	–	–	–	–	–	0.1264	0.1556	**0.6078**
	ARI	0.0001	0.0000	0.4151	0.2668	0.5336	0.4745	–	–	–	–	–	0.1222	0.1514	**0.6055**
cifar10	ACC	0.8898	0.9887	0.9899	0.9921	0.9872	0.9910	–	–	–	–	–	0.2658	0.9769	**0.9999**
	NMI	0.7891	0.9693	0.9730	0.9793	0.9658	0.9754	–	–	–	–	–	0.1465	0.9410	**0.9998**
	PUR	0.8898	0.9887	0.9899	0.9921	0.9872	0.9910	–	–	–	–	–	0.2807	0.9769	**0.9999**
	F1	0.7927	0.9777	0.9801	0.9845	0.9748	0.9822	–	–	–	–	–	0.1771	0.9549	**0.9999**
	ARI	0.7693	0.9752	0.9778	0.9827	0.9720	0.9802	–	–	–	–	–	0.0850	0.9499	**0.9999**
YouTubeFace10	ACC	0.7127	0.6342	0.7496	0.7569	0.7087	0.6944	–	–	–	–	–	0.6725	0.7679	**0.8486**
	NMI	0.7689	0.6427	0.7869	0.7773	0.7738	0.7964	–	–	–	–	–	0.6609	0.7634	**0.8722**
	PUR	0.7601	0.6583	0.8003	0.7875	0.7859	0.7791	–	–	–	–	–	0.7128	0.7773	**0.8917**
	F1	0.6799	0.4618	0.7077	0.5574	0.5996	0.6459	–	–	–	–	–	0.5786	0.7091	**0.8221**
	ARI	0.6773	0.3633	0.6706	0.5254	0.5721	0.6248	–	–	–	–	–	0.5233	0.6739	**0.8002**
YouTubeFace20	ACC	0.7083	0.5799	–	0.7516	0.6787	0.6909	–	–	–	–	–	0.4886	0.7754	**0.7818**
	NMI	0.7741	0.6252	–	**0.8346**	0.8056	0.8322	–	–	–	–	–	0.5663	0.7988	0.8344
	PUR	0.7638	0.6443	–	0.7862	0.7731	0.7644	–	–	–	–	–	0.5315	0.7870	**0.7937**
	F1	0.6242	0.3016	–	0.6294	0.5768	0.6262	–	–	–	–	–	0.3771	0.6832	**0.6934**
	ARI	0.6007	0.2337	–	0.6203	0.5657	0.6173	–	–	–	–	–	0.3396	0.6633	**0.6738**

(1) Overall, FTHMC achieved the best performance across all four metrics (ACC, PUR, F1, ARI) on all eight datasets. For the NMI metric, FTHMC achieved the best performance on seven datasets, and second-best on the YouTubeFace20 dataset. Notably, on the MSRCv1 dataset, FTHMC achieved perfect clustering performance, with all metrics reaching 1.0. Furthermore, on some datasets, the performance improvement of FTHMC is quite substantial. For example, on the CCV dataset, FTHMC outperformed the second-best method by 41.58 %, 29.21 %, 38.78 %, 57.88 %, and 67.26 % across the five metrics; on the YouTubeFace10 dataset, FTHMC improved by 10.50 %, 9.52 %, 11.42 %, 15.94 %, and 18.15 % on the respective metrics. This all-around performance advantage demonstrates the superiority of the FTHMC method.

(2) On small- to medium-scale datasets, tensor-based multi-view clustering methods exhibit more pronounced performance advantages compared to matrix-based methods. This is because tensor representation of multi-view data can directly capture high-order multi-view information and improve performance. However, even when compared to several tensor-based clustering methods, our proposed FTHMC still outperforms them. We believe that the tensor hyperbolic tangent-p norm (THT$_p$N) is a key factor contributing to this performance. Compared to TNN and some other tensor rank approximations, THT$_p$N is an unbiased and more compact tensor rank surrogate, and it can adjust the penalty weight distribution for singular values through the p parameter. These properties enable FTHMC to learn better low-rank tensor representations, thus capturing multi-view complementary and consistency information more effectively.

(3) On large-scale datasets, most tensor-based methods (such as LTMSC, tSVD-MSC, and WTNNM) fail to run successfully. The primary reason lies in their $\mathcal{O}(n^3)$ cubic computational complexity. When the sample size expands to $n > 10,000$, the computational time and hardware resource requirements exceed practical feasibility, rendering these algorithms incapable of completing effective computations. Given that the sample sizes of the large-scale datasets used here range from 30,000 to 70,000, the algorithms face stringent complexity requirements, making it impossible for these high-complexity

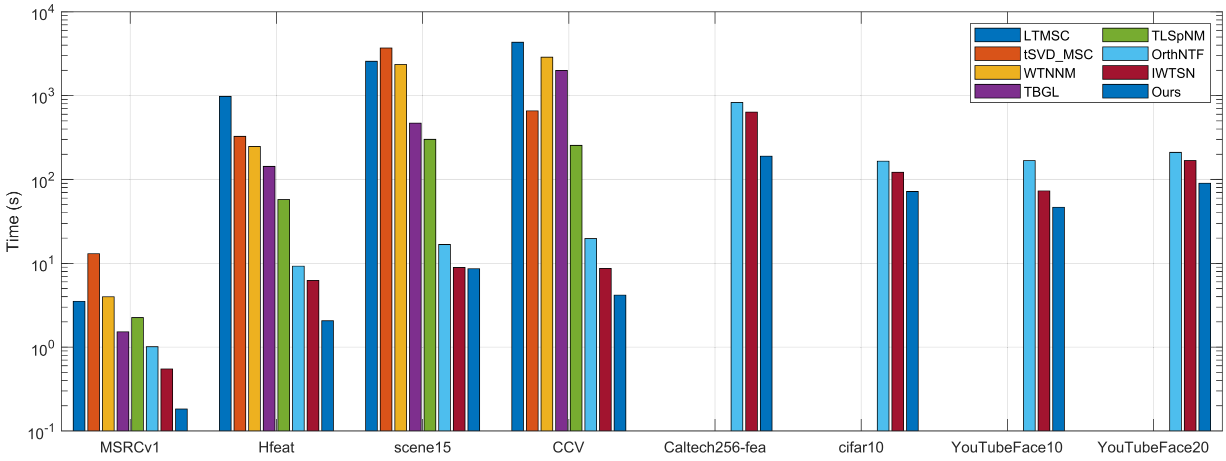


Fig. 4. The running time of FTHMC compared with tensor-based multi-view clustering methods on benchmark datasets. A blank bar signifies that the corresponding method exhausted available memory while processing the dataset.

traditional tensor methods to operate successfully. To address this, our proposed FTHMC reduces the overall computational complexity by leveraging anchor representation learning, enabling efficient performance on large-scale datasets. Compared to other anchor-based tensor methods like OrthNTF and IWTSN, FTHMC demonstrates significant improvements across all metrics on four large-scale datasets. This highlights FTHMC's ability to fully exploit the advantages of THT_pN for learning superior low-rank tensor representations on large-scale datasets, while also validating the effectiveness of anchor representation learning.

(4) FTHMC outperformed all anchor-based fast multi-view clustering methods across datasets of all scales, from small to medium to large. For example, on the MSRCv1 dataset, FTHMC improved the ACC metric by 12.90 % compared to the best anchor-based method EMVGCLG; on the YouTubeFace10 dataset, FTHMC improved the ACC metric by 10.50 % compared to the best anchor-based method IWTSN. The only performance disadvantage appears in the NMI metric on the YouTubeFace20 dataset, where FTHMC is slightly lower than FMVACC by −0.03 %, but in terms of ACC, FTHMC outperforms FMVACC by 4.02 %. Overall, compared to anchor-based fast multi-view clustering methods, FTHMC demonstrates a significant performance advantage.

4.3. Running time analysis

We first compared the running time of FTHMC with tensor-based multi-view clustering methods on eight datasets. To ensure fairness, all methods using the anchor mechanism, namely TBGL, OrthNTF, IWTSN and FTHMC, employed the same size of anchors, and all experiments were repeated five times to record the average running time. From Fig. 4, we can observe that on the four small- to medium-scale datasets, FTHMC's running time is only 1/10 to 1/1000 of other tensor-based methods, showing a significant advantage in speed. On the four large-scale datasets, only the OrthNTF and IWTSN methods, which adopt the anchor strategy, can run successfully. Compared to OrthNTF, FTHMC's runtime is only 21 % to 43 % of OrthNTF's, while compared to IWTSN, FTHMC's runtime is approximately 60 % on average. Overall, FTHMC exhibits the shortest runtime among tensor clustering methods, further underscoring its significant efficiency advantage.

To further assess the running efficiency of FTHMC, we compared its running time with anchor-based fast multi-view clustering methods on six datasets, using the same anchor sizes for all methods. The results in Fig. 5 show that FTHMC achieved the shortest running time on five out of six datasets, and the second-best on the YouTubeFace20 dataset. On all datasets, FTHMC outperformed the classic fast clustering method LMVSC in terms of running speed. In summary, FTHMC demonstrates leading computational efficiency both in the field of tensor multi-view methods and in fast multi-view clustering methods.

4.4. Ablation study

4.4.1. Comparison of different tensor norms

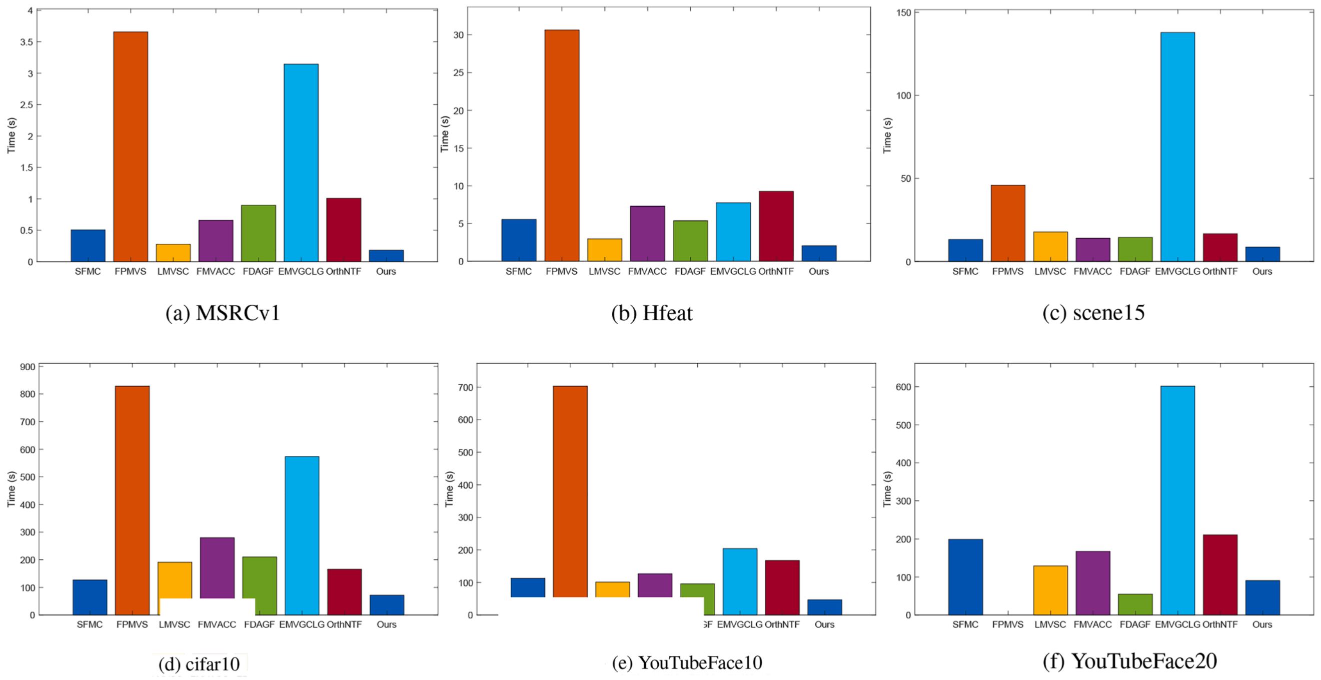
We replace THT_pN in FTHMC with the tensor nuclear norm (TNN) and tensor Schatten-p norm (SP), respectively, to compare the performance distribution of these three tensor low-rank approximation norms under different λ values. The experimental results are shown in Fig. 6. It can be observed that THT_pN not only achieves the best performance in terms of the ACC metric but also demonstrates stronger robustness and stability, significantly outperforming the tensor nuclear norm (TNN) and tensor Schatten-p norm (SP). We believe this is closely related to the functional characteristics of THT_pN. The value range of THT_pN is $[0, 1]$, and it is a monotonically increasing function approaching 1, enabling it to approximate the true rank function more stably. This makes it more stable compared to the other two tensor low-rank approximations, exhibiting better robustness. Moreover, by adjusting the value of p, THT_pN can flexibly control the distribution of penalty weights on singular values, resulting in greater adaptability. This is reflected in THT_pN achieving optimal clustering performance across different datasets and λ values.

4.4.2. Impact of anchor selection strategies

We compared the effects of different anchor selection strategies, including the commonly used random strategy, the k-means method, and the BKHK method proposed in this paper (FTHMC). The methods differ only in their anchor selection approaches, while other parameters (e.g., the number of anchors) remain identical to ensure a fair comparison. All experiments were repeated 10 times to mitigate randomness, and Table 4 reports the average results for ACC, NMI, and running time. The results indicate that the random strategy achieves the shortest running time but performs poorly in clustering. The k-means method improves clustering performance over the random strategy but is outperformed by BKHK, with the drawback of having the longest running time. In contrast, BKHK strikes a balance between performance and efficiency, delivering the best clustering accuracy while maintaining a competitive running time.

4.5. Parameter sensitivity analysis

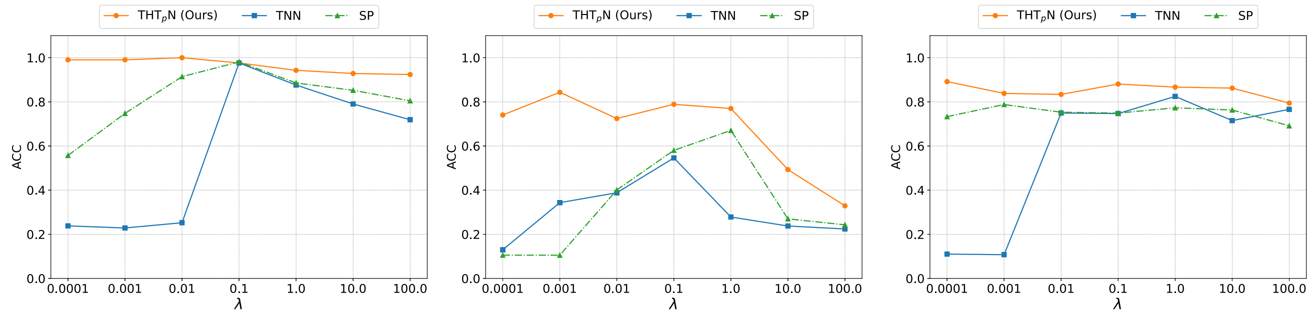
Our FTHMC algorithm has two hyperparameters, λ and p. In this section, we vary the values of λ and p and record the change in the ACC metric to discuss the effect of these two parameters on model performance.



(a) MSRCv1 (b) Hfeat (c) scene15

(d) cifar10 (e) YouTubeFace10 (f) YouTubeFace20

Fig. 5. The running time of FTHMC compared with anchor-based fast multi-view clustering methods on benchmark datasets. A blank bar signifies that the corresponding method exhausted available memory while processing the dataset.



(a) MSRCv1 (b) CCV (c) YouTubeFace10

Fig. 6. Ablation study: Replacing THT$_p$N in FTHMC with the tensor nuclear norm (TNN) and tensor Schatten-p norm (SP) to compare the performance impact of the three tensor norms.

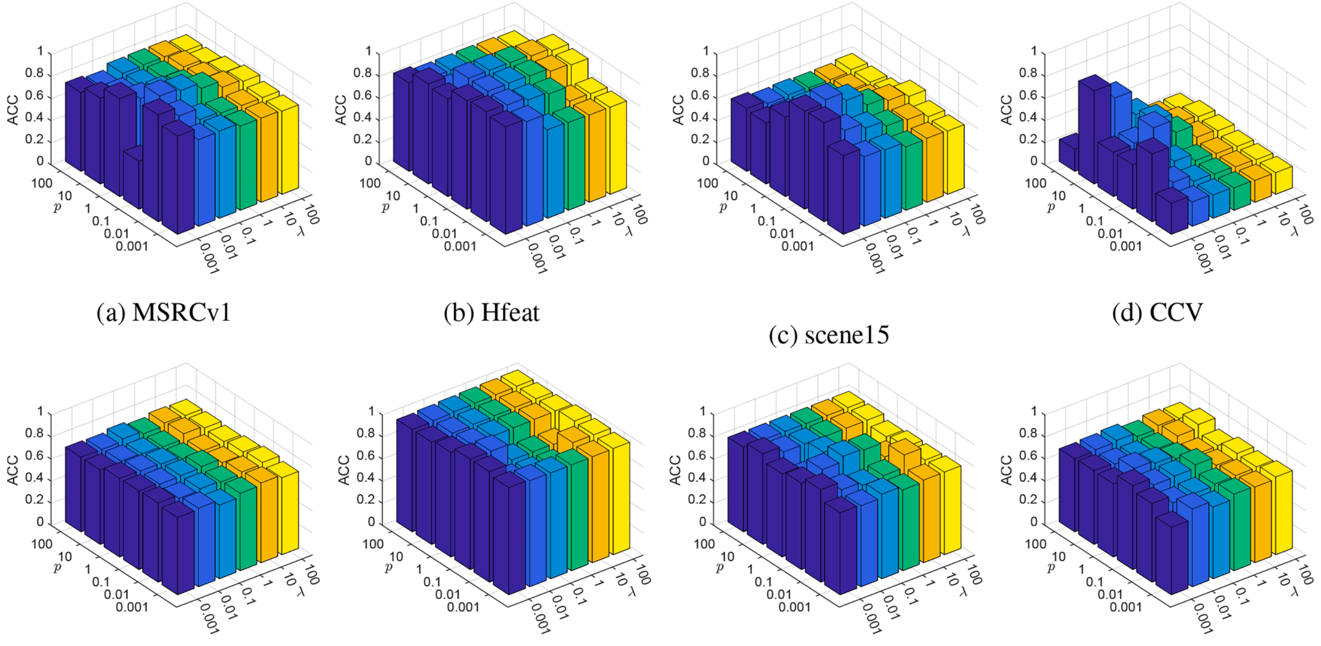
Table 4
The average clustering performance and running time (s) of FTHMC equipped with different anchor selection strategies.

Dataset	MSRCv1			Hfeat			scene15			YouTubeFace20		
Metric	ACC	NMI	time	ACC	NMI	time	ACC	NMI	time	ACC	NMI	time
random	0.9876	0.9720	**0.1945**	0.8815	0.9484	**1.8584**	0.8586	0.8962	**8.3923**	0.7171	0.8016	**95.3385**
k-means	0.9962	0.9914	0.2460	0.9985	0.9959	2.2574	0.8580	0.8906	10.4261	0.7197	0.8191	114.1526
BKHK	**1.0000**	**1.0000**	0.1971	**0.9995**	**0.9986**	1.9698	**0.9353**	**0.9207**	8.4254	**0.7891**	**0.8344**	95.8800

Both parameters take values from the set $\{0.001, 0.01, 0.1, 1, 10, 100\}$. The analysis results across eight datasets are shown in Fig. 7. It can be observed that, except for the CCV dataset, the model performance changes very little with different settings of λ and p. Overall, for most datasets, the model can maintain good clustering performance across a wide range of parameter values. We believe this is related to the proposed tensor hyperbolic tangent-p norm (THT$_p$N). Regardless of the choice of p, the hyperbolic tangent function saturates as $x \to \infty$ $(\tanh(px) \approx 1)$, effectively approximating the rank function while avoiding excessive penalization of large singular values. This leads to robustness and demonstrates the superiority of THT$_p$N.

4.6. Algorithm convergence

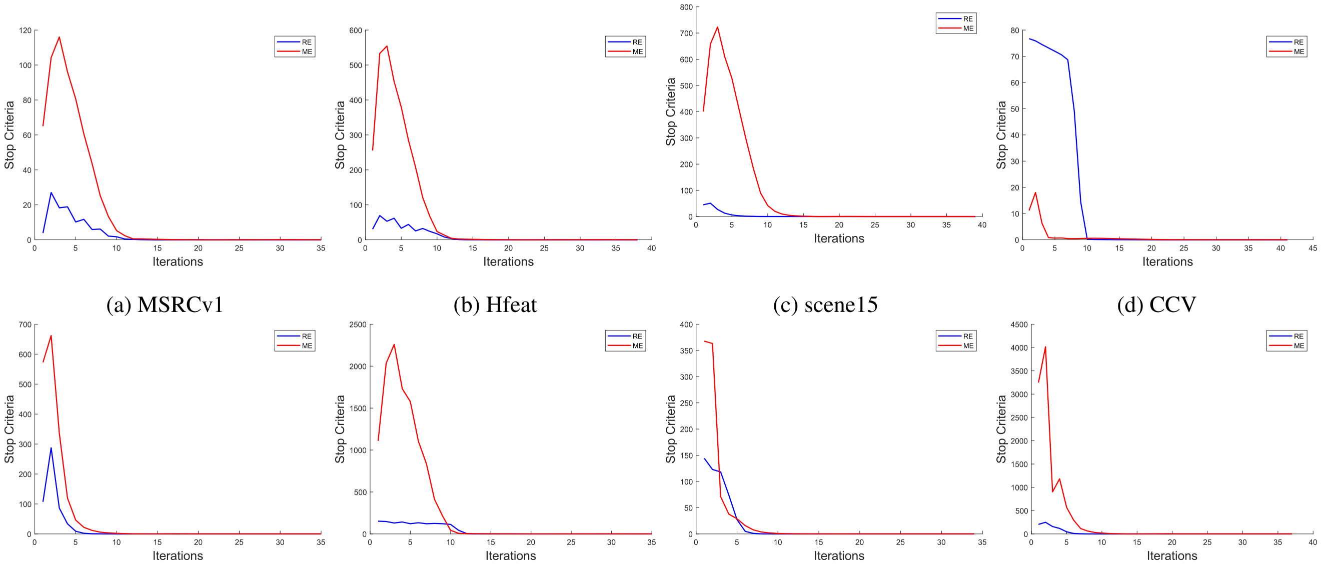
To analyze the convergence of FTHMC, we present the variations in the stopping criteria across eight datasets. The stopping criteria are represented as reconstruction error (RE): $RE = \max_v \left\| \mathbf{X}^{(v)} - \mathbf{A}^{(v)}\mathbf{Z}^{(v)} - \mathbf{E}^{(v)} \right\|_\infty$ and match error (ME): $ME = \|\mathcal{Z} - \mathcal{G}\|_\infty$.



(a) MSRCv1 (b) Hfeat (c) scene15 (d) CCV

(e) Caltech256-fea (f) cifar10 (g) YouTubeFace10 (h) YouTubeFace20

Fig. 7. The clustering ACC of FTHMC under different hyperparameters λ and p.



(a) MSRCv1 (b) Hfeat (c) scene15 (d) CCV

(e) Caltech256-fea (f) cifar10 (g) YouTubeFace10 (h) YouTubeFace20

Fig. 8. Convergence curves of FTHMC on eight datasets, with the X-axis representing the number of iterations.

As shown in Fig. 8, on multiple datasets, both RE and ME converge rapidly to 0 within 15 steps and remain stable, verifying the excellent convergence performance of the proposed FTHMC method.

5. Conclusion

In this paper, we introduced FTHMC, a fast multi-view clustering method. FTHMC significantly reduces the computational burden of tensor-based approaches for large-scale data by integrating efficient anchor selection method BKHK and constructing tensors from anchor representation matrices. We also introduce the tensor hyperbolic tangent-p norm (THT$_p$N), which provides a more robust and accurate approximation of the tensor rank function, enabling better extraction of multi-view consistency and complementarity. By combining the anchor acceleration strategy with THT$_p$N, FTHMC achieves improvements in both performance and efficiency.

However, in the current work we treat p as a fixed hyperparameter. Although experiments show that THT$_p$N exhibits strong stability and robustness across different p values, adopting an adaptive adjustment strategy for p-tailored to specific datasets or task scenarios-could further enhance the model's generalization and adaptability. Theoretically, THT$_p$N is fully differentiable, making it possible to learn p within the optimization process; future work might explore data-driven heuristic mechanisms (e.g., dynamically balancing punishment across singular values based on their distribution) or joint-learning strategies (treating p as a learnable parameter with appropriate regularization to ensure convergence and interpretability). Looking ahead, extending FTHMC to handle incomplete or noisy data in multi-view settings could unlock new opportunities in real-world applications, making it a promising direction for future research. Additionally, explicitly modeling data uncertainty and fuzziness within the framework represents a significant avenue for enhancing robustness in complex environments.

CRediT authorship contribution statement

Yongbo Yu: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Data curation; **Zhoumin Lu:** Writing – review & editing, Validation; **Jingjing Xue:** Writing – review & editing, Validation; **Rong Wang:** Writing – review & editing, Investigation; **Zongcheng Miao:** Validation, Resources, Project administration; **Feiping Nie:** Validation, Supervision, Funding acquisition.

Data availability

Data will be made available on request.

Declaration of competing interest

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

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62176212 and Grant 62236001, in part by the Natural Science Basic Research Program of Shaanxi under Program 2025JC-YBQN-837.

Supplementary material

Supplementary material associated with this article can be found in the online version at 10.1016/j.patcog.2025.112195.

References

[1] Y. Yu, Z. Lu, F. Nie, W. Yu, Z. Miao, X. Li, Pseudo-label guided bidirectional discriminative deep multi-view subspace clustering, IEEE Trans. Knowl. Data Eng. 37 (7) (2025) 4213–4224.

[2] Z. Long, C. Zhu, J. Chen, Z. Li, Y. Ren, Y. Liu, Multi-view MERA subspace clustering, IEEE Trans. Multimed. 26 (2024) 3102–3112.

[3] Y. Liu, J. Chen, Y. Lu, W. Ou, Z. Long, C. Zhu, Adaptively topological tensor network for multi-view subspace clustering, IEEE Trans. Knowl. Data Eng. 36 (11) (2024) 5562–5575.

[4] G. Griffin, A. Holub, P. Perona, et al., Caltech-256 object category dataset, Technical Report 7694, California Institute of Technology Pasadena, 2007.

[5] Q. Shen, Y. Chen, C. Zhang, Y. Tian, Y. Liang, Pick-and-place transform learning for fast multi-view clustering, IEEE Trans. Image Process. 33 (2024) 1272–1284.

[6] M. Brbić, I. Kopriva, Multi-view low-rank sparse subspace clustering, Pattern Recognit. 73 (2018) 247–258.

[7] Y. Yu, J. Wang, W. Yu, Z. Zhao, Z. Miao, F. Nie, Bidirectional fusion for deep contrastive multi-view clustering, Expert Syst. Appl. 287 (2025) 128193.

[8] H. Gao, F. Nie, X. Li, H. Huang, Multi-view subspace clustering, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4238–4246.

[9] D. Wu, X. Dong, F. Nie, R. Wang, X. Li, An attention-based framework for multi-view clustering on grassmann manifold, Pattern Recognit. 128 (2022) 108610.

[10] Y. Chen, X. Xiao, Y. Zhou, Multi-view subspace clustering via simultaneously learning the representation tensor and affinity matrix, Pattern Recognit. 106 (2020) 107441.

[11] Y. Du, G.-F. Lu, Joint local smoothness and low-rank tensor representation for robust multi-view clustering, Pattern Recognit. 157 (2025) 110944.

[12] Q. Shen, T. Xu, Y. Liang, Y. Chen, Z. He, Robust tensor recovery for incomplete multi-view clustering, IEEE Trans. Multimed. 26 (2024) 3856–3870.

[13] J. Guo, Y. Sun, J. Gao, Y. Hu, B. Yin, Logarithmic schatten-p norm minimization for tensorial multi-view subspace clustering, IEEE Trans. Pattern Anal. Mach. Intell. 45 (3) (2022) 3396–3410.

[14] Z. Kang, W. Zhou, Z. Zhao, J. Shao, M. Han, Z. Xu, Large-scale multi-view subspace clustering in linear time, in: Proceedings of the AAAI Conference on Artificial Intelligence, 34, 2020, pp. 4412–4419.

[15] X. Li, H. Zhang, R. Wang, F. Nie, Multiview clustering: a scalable and parameter-free bipartite graph fusion method, IEEE Trans. Pattern Anal. Mach. Intell. 44 (1) (2022) 330–344.

[16] W. Xia, Q. Gao, Q. Wang, X. Gao, C. Ding, D. Tao, Tensorized bipartite graph learning for multi-view clustering, IEEE Trans. Pattern Anal. Mach. Intell. 45 (4) (2022) 5187–5202.

[17] J. Li, Q. Gao, Q. WANG, M. Yang, W. Xia, Orthogonal non-negative tensor factorization based multi-view clustering, in: A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine (Eds.), Advances in Neural Information Processing Systems, 36, Curran Associates, Inc., 2023, pp. 18186–18202.

[18] S. Wang, X. Liu, X. Zhu, P. Zhang, Y. Zhang, F. Gao, E. Zhu, Fast parameter-free multi-view subspace clustering with consensus anchor guidance, IEEE Trans. Image Process. 31 (2021) 556–568.

[19] Y. Wen, S. Liu, X. Wan, S. Wang, K. Liang, X. Liu, X. Yang, P. Zhang, Efficient multi-view graph clustering with local and global structure preservation, in: Proceedings of the 31st ACM International Conference on Multimedia, 2023, pp. 3021–3030.

[20] S. Wang, X. Liu, S. Liu, J. Jin, W. Tu, X. Zhu, E. Zhu, Align then fusion: generalized large-scale multi-view clustering with anchor matching correspondences, 35, 2022, pp. 5882–5895.

[21] P. Zhang, S. Wang, L. Li, C. Zhang, X. Liu, E. Zhu, Z. Liu, L. Zhou, L. Luo, Let the data choose: flexible and diverse anchor graph fusion for scalable multi-view clustering, in: Proceedings of the AAAI Conference on Artificial Intelligence, 37, 2023, pp. 11262–11269.

[22] Q. Shen, Z. Guo, H. Wang, Y. Xu, Y. Chen, S. Wang, Y. Liang, Reliable entropy-induced anchor learning for incomplete multi-view subspace clustering, IEEE Trans. Circuits Syst. Video Technol. 35 (6) (2025) 5293–5306.

[23] Y. Xie, D. Tao, W. Zhang, Y. Liu, L. Zhang, Y. Qu, On unifying multi-view self-representations for clustering by tensor multi-rank minimization, Int. J. Comput. Vis. 126 (2018) 1157–1179.

[24] J. Wan, X. Li, P. Zhang, H. Chen, X. Ouyang, T. Li, K.C. Tan, FFS-MCC: fusing approximation and fuzzy uncertainty measures for feature selection with multi-correlation collaboration, Inf. Fusion 120 (2025) 103101.

[25] W. Sun, C. Li, Q. Li, X. Fang, J. He, L. Liu, Joint intra-view and inter-view enhanced tensor low-rank induced affinity graph learning, Pattern Recognit. 159 (2024) 111140.

[26] X. Li, Z. Ren, Q. Sun, Z. Xu, Auto-weighted tensor schatten p-norm for robust multi-view graph clustering, Pattern Recognit. 134 (2023) 109083.

[27] J. Ji, S. Feng, High-order complementarity induced fast multi-view clustering with enhanced tensor rank minimization, in: Proceedings of the 31st ACM International Conference on Multimedia, 2023, pp. 328–336.

[28] C. Zhang, H. Fu, S. Liu, G. Liu, X. Cao, Low-rank tensor constrained multiview subspace clustering, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1582–1590.

[29] Q. Gao, W. Xia, Z. Wan, D. Xie, P. Zhang, Tensor-SVD based graph learning for multi-view subspace clustering, in: Proceedings of the AAAI Conference on Artificial Intelligence, 34, 2020, pp. 3930–3937.

[30] B. Cai, G.-F. Lu, L. Yao, H. Li, High-order manifold regularized multi-view subspace clustering with robust affinity matrices and weighted TNN, Pattern Recognit. 134 (2023) 109067.

[31] M.E. Kilmer, K. Braman, N. Hao, R.C. Hoover, Third-order tensors as operators on matrices: a theoretical and computational framework with applications in imaging, SIAM J. Matrix Anal. Appl. 34 (1) (2013) 148–172.

[32] D. Cai, X. Chen, Large scale spectral clustering via landmark-based sparse representation, IEEE Trans. Cybern. 45 (8) (2014) 1669–1680.

[33] F. Nie, F. Xie, J. Wang, X. Li, Fast adaptively balanced min-cut clustering, Pattern Recognit. 158 (2025) 111027.

[34] F. Nie, W. Zhu, X. Li, Unsupervised large graph embedding based on balanced and hierarchical K-means, IEEE Trans. Knowl. Data Eng. 34 (4) (2020) 2008–2019.

[35] Y. Bengio, I. Goodfellow, A. Courville, Deep Learning, vol. 1, MIT Press, Cambridge, MA, USA 2017.

[36] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2012) 171–184.

[37] P.D. Tao, L.T.H. An, Convex analysis approach to DC programming: theory, algorithms and applications, Acta Math. Vietnam. 22 (1) (1997) 289–355.

[38] Z. Kang, C. Peng, Q. Cheng, Robust PCA via nonconvex rank approximation, in: 2015 IEEE International Conference on Data Mining, IEEE, 2015, pp. 211–220.

[39] J. Winn, N. Jojic, LOCUS: learning object classes with unsupervised segmentation, in: Tenth IEEE International Conference on Computer Vision (ICCV'05), vol. 1, 2005, pp. 756–763.

[40] R. Duin, Multiple features, 1998, UCI Machine Learning Repository. https://doi.org/10.24432/C5HC70.

[41] L. Fei-Fei, P. Perona, A Bayesian hierarchical model for learning natural scene categories, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 2, 2005, pp. 524–531.

[42] Y.-G. Jiang, G. Ye, S.-F. Chang, D. Ellis, A.C. Loui, Consumer video understanding: a benchmark database and an evaluation of human and machine performance, in: Proceedings of the 1st ACM International Conference on Multimedia Retrieval, 2011, pp. 1–8.

[43] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images, Toronto, ON, Canada, 2009.

[44] D. Huang, C.-D. Wang, J.-H. Lai, Fast multi-view clustering via ensembles: towards scalability, superiority, and simplicity, IEEE Trans. Knowl. Data Eng. 35 (11) (2023) 11388–11402.

[45] Y. Sun, X. Li, Q. Sun, M.-L. Zhang, Z. Ren, Improved weighted tensor schatten p-norm for fast multi-view graph clustering, in: ACM Multimedia 2024, 2024.