Noise-Aware Hierarchical Tensor Fusion for Multi-View Clustering

BO ZHONG, School of Mathematics and Computer Sciences, Institute of Metaverse, Jiangxi Key Laboratory of Virtual Reality, Nanchang University, Nanchang, China
JIAN-SHENG WU*, School of Mathematics and Computer Sciences, Institute of Metaverse, Jiangxi Key Laboratory of Virtual Reality, Nanchang University, Nanchang, China
WEIDONG MIN, School of Mathematics and Computer Sciences, Institute of Metaverse, Jiangxi Key Laboratory of Virtual Reality, Nanchang University, Nanchang, China

Recently, multi-view clustering methods based on deep matrix factorization have exhibited significant promise in learning hierarchical semantic representations. However, three critical limitations remain in existing approaches. First, due to the independent intra-view hierarchical decomposition, these methods rely on pairwise view correlations, thereby failing to capture high-order consistency. Second, the complementary information embedded in intermediate layers remains under-utilized. Third, they are susceptible to sensor noise and feature redundancy, which degrades the quality of learned representations. To address these limitations, this paper presents a novel approach named Noise-Aware Hierarchical Tensor Fusion (NAHTF). The proposed NAHTF integrates tensor dictionary learning with deep semi-nonnegative tensor decomposition, leveraging the tensor t-product to facilitate cross-view and cross-layer information fusion of multi-view data. Specifically, a noise-aware tensor dictionary learning module is designed to construct a robust tensor representation by combining Schatten p-norm-based low-rank tensor recovery for capturing high-order correlations and $l_{2,1}$-norm-based sparse noise modeling for suppressing adverse components. A deep semi-nonnegative tensor factorization module is then employed to hierarchically decompose the refined tensor representation using the tensor t-product, which facilitates the propagation of complementary information across all decomposition layers and thereby fully exploits both inter-view consistency and intra-view diversity. Moreover, a joint optimization framework is developed to simultaneously learn the tensor representation and perform deep tensor factorization, avoiding suboptimal representations caused by decoupled processes. Extensive experiments on six benchmark datasets, including text, face, and image data, demonstrate that NAHTF outperforms fourteen state-of-the-art multi-view clustering methods.

JAIR Associate Editor: Bo Han

JAIR Reference Format:
Bo Zhong, Jian-Sheng Wu, and Weidong Min. 2026. Noise-Aware Hierarchical Tensor Fusion for Multi-View Clustering. *Journal of Artificial Intelligence Research* 87, Article 27 (November 2026), 27 pages. DOI: 10.1613/jair.1.22933

1 Introduction

The rapid evolution of information technologies has enabled the collection of multi-view data through heterogeneous sensors or feature extraction methodologies. For example, visual data can be represented via diverse

*Corresponding Author.

Authors' Contact Information: Bo Zhong, ORCID: 0000-0002-5698-7073, zhongbo@bjtu.edu.cn, School of Mathematics and Computer Sciences, Institute of Metaverse, Jiangxi Key Laboratory of Virtual Reality, Nanchang University, Nanchang, Nanchang, Jiangxi, China; Jian-Sheng Wu, ORCID: 0000-0001-9029-357X, jianshengwu@ncu.edu.cn, School of Mathematics and Computer Sciences, Institute of Metaverse, Jiangxi Key Laboratory of Virtual Reality, Nanchang University, Nanchang, Nanchang, Jiangxi, China; Weidong Min, ORCID: 0000-0003-2526-2181, minweidong@ncu.edu.cn, School of Mathematics and Computer Sciences, Institute of Metaverse, Jiangxi Key Laboratory of Virtual Reality, Nanchang University, Nanchang, Nanchang, Jiangxi, China.



This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2026 Copyright held by the owner/author(s).
DOI: 10.1613/jair.1.22933

feature descriptors, including Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), Color Moments (CMT), Global Image Signatures (GIST), and Center-Symmetric Local Binary Patterns (CENTRIST). Similarly, textual content may exist in multilingual formats such as English, Chinese, and French. Multi-view data, compared to single-view data, encapsulate complementary perspectives of the same data objects, offering both consistency and diversity across views. Fully exploiting the complementary information (consistency and diversity) of multi-view data is a critical challenge in advancing multi-view learning (Kong et al. 2025; W. Liang, C. Tang, et al. 2024; W. Liang, S. Zhou, et al. 2020; Lin et al. 2022; J.-S. Wu et al. 2025). Multi-view clustering (MVC) (M.-S. Chen, Lai, et al. 2025; W. Liang, Zhu, et al. 2024; G.-Y. Zhang, D. Huang, et al. 2025, 2024; G.-Y. Zhang, Y.-D. Huang, et al. 2025), an important branch of multi-view learning, seeks to partition unlabeled datasets by exploiting inter-view consistency and intra-view diversity, and has demonstrated practical utility across diverse domains such as weather prediction (Sharma and Seal 2021), industrial fault detection (Sharma, Seal, et al. 2022), and social network community analysis (W. Tang et al. 2021).

When collecting data through sensors or describing data via various descriptors, multi-view data often exhibit redundant features that degrade clustering and classification performance. Therefore, learning discriminative low-dimensional representations for data that capture intrinsic data structures has emerged as a crucial preprocessing step (Lee and Seung 2000). Recently, nonnegative matrix factorization (NMF) (Lee and Seung 2000), an effective representation learning technique, has achieved prominence. It decomposes the nonnegative data matrix as a product of two nonnegative matrices: the basis matrix and the representation matrix. Owing to the non-negativity constraints, NMF learns interpretable part-based representations through additive combinations. To date, various variants of NMF have been proposed (Ding et al. 2008; B. Yang et al. 2020). The subsequent variant Semi-NMF (Ding et al. 2008) extends its applicability by relaxing non-negativity constraints on the input data and basis matrices while retaining interpretability. Recent years have witnessed the proliferation of NMF-based multi-view learning frameworks (J. Liu, C. Wang, et al. 2013; Luong and Nayak 2020), which are designed to derive discriminative low-dimensional representations for multi-view data by extracting complementary information.

However, practical applications frequently involve latent hierarchical attributes within data. For example, facial recognition systems must capture multi-level features ranging from low-level pose variations to high-level identity characteristics (Trigeorgis et al. 2014). Similarly, community detection in complex networks requires modeling complex hierarchical information due to the diverse and intricate topological structure (Ye et al. 2018). Mining the latent hierarchical features within data provides richer feature information for downstream tasks such as clustering.

Limited by shallow NMF architectures, traditional NMF methods face limitations in extracting such hierarchical features due to their linear separability assumptions and single-layer decomposition constraints (Trigeorgis et al. 2014). Recent advancements integrating deep learning architectures (Yoshua 2009) with NMF have yielded multi-view deep matrix factorization methods (Z. Chen et al. 2022; S. Wang et al. 2023; H. Zhao et al. 2017). These methods perform hierarchical decomposition to uncover multi-level data attributes with decreasing layer sizes while maintaining interpretability through layer-wise part-whole relationships. Despite gaining promising performance, three critical limitations persist: 1) **Limited High-Order Correlation Modeling**: Current methods mainly focus on pairwise view correlations, neglecting complex high-order dependencies, as hierarchical decomposition is performed independently within individual views; 2) **Top-Layer Representation Fusion**: Complementary information is exploited solely at the top decomposition layer, disregarding latent synergies at intermediate layers; and 3) **Noise Susceptibility**: Sensor noise and feature redundancy significantly degrade the quality of learned representations.

In this paper, a novel approach named Noise-Aware Hierarchical Tensor Fusion for multi-view clustering (NAHTF) is proposed to tackle these limitations. The NAHTF method integrates anchor-based tensor dictionary learning

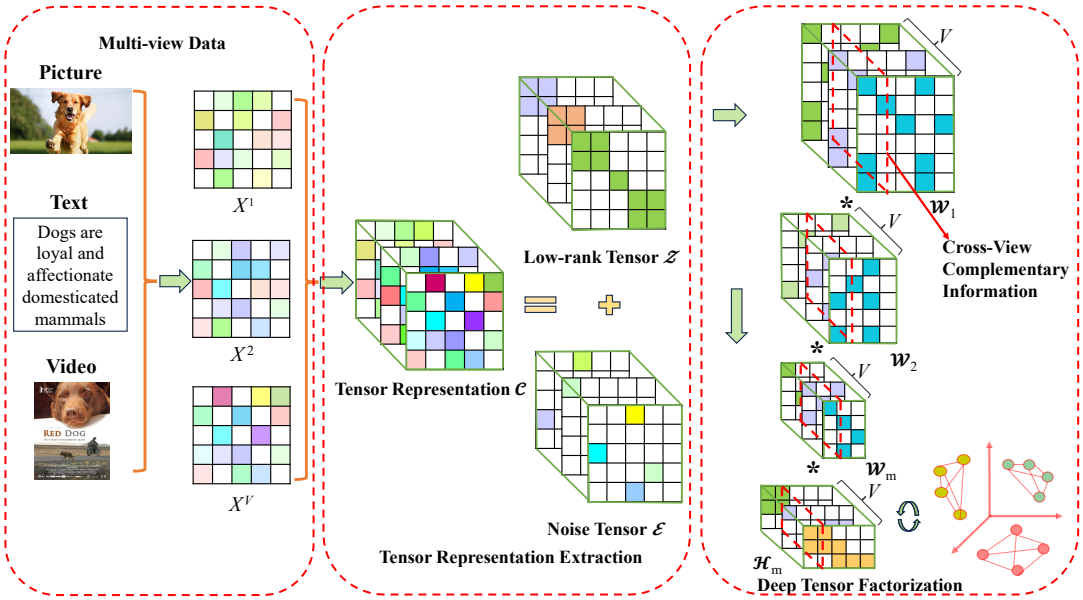


Fig. 1. Illustration of the proposed NAHTF. NAHTF simultaneously extracts a low-rank and high-quality tensor representation for data, explores the local and high-order consistency of multi-view data, and explores the complementary information of multi-view data throughout the multi-layer decomposition.

and deep semi-nonnegative tensor decomposition, leveraging tensor t-product to achieve cross-view and cross-layer fusion of multi-view data. As illustrated in Figure 1, NAHTF alternates between learning a high-quality tensor representation and performing deep tensor factorization. Given the significant impact of tensor quality on multi-view data clustering performance, NAHTF first constructs a tensor representation using an anchor-based tensor dictionary learning through Schatten p-norm-based low-rank tensor learning (Gao et al. 2020) and tensor $l_{2,1}$-norm-based adverse component modeling (J. Wu et al. 2019), in order to extract a noise-free tensor representation. The Schatten p-norm enables the exploration of high-order cross-view correlations, while the $l_{2,1}$-norm effectively removes noise and redundancy, ensuring that the extracted tensor representation is robust and captures the intrinsic data structures. Subsequently, a deep semi-nonnegative tensor factorization module is designed to directly decompose the tensor representation into multi-layer basis and representation tensors via the tensor t-product (Lu et al. 2019). Unlike traditional methods that rely on the top-layer fusion strategies and neglect complementary information at lower layers, NAHTF can fully explore the complementary information throughout the multi-layer decomposition, in which the tensor t-product operation circularly computes the multiplication between pairs of frontal slices of two tensors, facilitating the propagation of complementary information across all decomposition layers. Extensive experiments on benchmark datasets demonstrate the superiority of the proposed NAHTF over fourteen state-of-the-art multi-view clustering methods.

In summary, the model has the following contributions:

- **Hierarchical Representation Fusion**: NAHTF proposes a deep semi-nonnegative tensor factorization architecture that propagates complementary information across all decomposition layers, effectively addressing inter-layer view correlation isolation. This allows for a more comprehensive exploration of the hierarchical structure of multi-view data.

- **Noise-Robust Tensor Representation**: NAHTF proposes a tensor dictionary learning to learn a noise-robust tensor representation from multi-view data. It integrates sparse noise suppression via the tensor $l_{2,1}$-norm constraint and low-rank tensor recovery via the tensor Schatten p-norm constraint, effectively suppressing noise while preserving intrinsic data structures.
- **Joint Tensor-Factorization Learning**: A unified framework is developed to simultaneously optimize tensor construction and factorization, eliminating suboptimal representations caused by decoupled processes. This joint optimization ensures that the tensor representation better captures the characteristics of multi-view data.

The remainder of this paper are structured as follows. Section 2 offers an overview of matrix factorization-based clustering methods. Section 3 presents the proposed model, with its optimization procedure detailed in Section 4. Section 5 presents the evaluation experiments. Finally, Section 7 summarizes this paper and discusses future advancements.

2 Related Work

This section provides a review of recent progress in NMF methods relevant to the present work.

2.1 Single-view Deep NMF

Let $X = [x_1, x_2, ..., x_n] \in R_+^{d\times n}$ be a single-view data matrix containing n samples in the d-dimensional space. The NMF method seeks two non-negative low-rank matrices such that:

$$X \approx WH, \tag{1}$$

where $W \in R_+^{d\times r}$ and $H \in R_+^{r\times n}$ signify the basis and representation matrices, respectively. The non-negativity constraints facilitate the learning of interpretable part-based representations through additive combinations, adhering to the intuitive "parts-whole" paradigm.

Given that $r \ll d$, the NMF method is among the most prevalently used techniques for learning compact representations of high-dimensional data (S. Huang, Z. Xu, et al. 2017; Min et al. 2022). Nevertheless, in real-world applications, the original data often possess latent hierarchical attributes, rendering it challenging for shallow NMF methods to capture the intrinsic hierarchical semantics within the data. Inspired by the powerful capabilities of deep learning in nonlinear feature extraction, Deep Semi-NMF (Trigeorgis et al. 2014) integrates NMF and deep learning, extending shallow NMF into a multi-layer variant. This is accomplished by hierarchically decomposing the representation matrix to explore the hierarchical semantics of the original data:

$$X^v \approx W_1^v H_1^v, H_1^v \approx W_2^v H_2^v, \cdots, H_{m-1}^v \approx W_m^v H_m^v, \tag{2}$$

where $W_i^v \in \mathbb{R}^{r_{i-1}\times r_i}$ and $H_i^v \in \mathbb{R}_+^{r_i\times n}$ are the basis and representation matrices at the i-th layer of the hierarchical decomposition, respectively, and m is the number of layers. Here, $r_0 = d \geq r_1 \geq ... \geq r_m$. This architecture gradually reveals the latent hierarchical semantics of the data, progressing from low-level specific features to high-level abstract features (De Handschutter et al. 2021).

2.2 Multi-view Deep NMF

Recently, numerous multi-view learning methods based on NMF and its variants have been presented. For example, MultiNMF (J. Liu, C. Wang, et al. 2013) conducts NMF on each view separately and then aligns the representation matrix of each view with a consensus representation matrix. Given that the exploration of complementary information from multi-view data can enhance multi-view data processing, the 2CMV method (Luong and Nayak 2020) promotes the independence between the consensus matrix and the complementary matrix, which is derived from the view-specific representation matrices. However, these methods require post-processing, such as

k-means, on the new representations to cluster the data. Orthogonal NMF (B. Yang et al. 2020), on the other hand, directly decomposes the data matrices into a shared clustering indicator matrix and orthogonal basis matrices. Contrary to existing multi-view NMF methods that operate on each view independently, OrthNTF (J. Li, Gao, et al. 2023) proposes orthogonal tensor factorization to decompose the tensor formed by the anchor graphs of multi-view data, thus exploring high-order cross-view correlations. Nevertheless, it decouples the tensor representation construction from the tensor factorization, potentially resulting in a suboptimal tensor representation.

Due to their shallow architectures, the aforementioned methods cannot deal with real-world data, which typically exhibit complex and hierarchical characteristics (Trigeorgis et al. 2014). To address this issue, multi-view deep matrix factorization methods hierarchically decompose multi-view data into basis and representation matrices. To capture the cross-view consistency information, methods based on consistency information extraction (Cui et al. 2019; S. Huang, Kang, et al. 2020; H. Zhao et al. 2017) attempt to extract the same *representation* matrix for different data views. Unless otherwise specified below, the *italic "representation"* refers to the representation at the top layer of the decomposition. The multi-view deep semi-NMF method (H. Zhao et al. 2017) extends the single-view deep semi-NMF method (Trigeorgis et al. 2014) to multi-view data. To avoid introducing additional hyper-parameters, parameter-free multi-view deep NMF methods (Cui et al. 2019; S. Huang, Kang, et al. 2020) utilize an $l_{2,1}$-norm-based auto-weighted mechanism to automatically assign weights to each view.

In addition to consistency, diversity information is typically present in multi-view data. The diversity information extraction-based methods (Z. Chen et al. 2022; H. Huang, G. Zhou, N. Liang, et al. 2023) first decompose data views into different basis matrices and distinct representation matrices and then maximize the diversity between the *representations* of data views. While DDMF (Z. Chen et al. 2022) requires the *representation* matrices of different views to be orthogonal to each other to provide diverse *representations* for multi-view data, HDDMF (H. Huang, G. Zhou, N. Liang, et al. 2023) maximizes the diversity between *representation* pairs from different views.

To leverage both consistency and diversity information, multi-layer graph regularized multi-view deep NMF models (J. Li, G. Zhou, et al. 2020; Luong, Nayak, et al. 2022; Xue et al. 2019) extract consistency information from diverse *representation* matrices of data views after individually decomposing multi-view data. The distinction lies in how they approach the investigation of consistency. Specifically, DLRSE (Xue et al. 2019) extracts a consensus subspace from low-rank subspaces learned from *representations* in different views, MDGNMF (J. Li, G. Zhou, et al. 2020) aligns diverse *representation* matrices with a consensus representation matrix, and ODDNMF (Luong, Nayak, et al. 2022) expects the representations at each layer to preserve the same predefined manifold structure while making the *representation* matrices of different views orthogonal to each other. Alternatively, the models (S. Wang et al. 2023; W. Wu et al. 2024; C. Zhang et al. 2021) separate the *representation* matrix of each view into consensus and view-specific *representation* matrices and determine the diversity between views based on view-specific *representation pairs*. Moreover, DRDF (W. Wu et al. 2024) explores the intra-class compactness and inter-class separation of cluster pseudo-labels as self-supervised information. To benefit from the auto-encoder-like architecture (Ye et al. 2018), multi-view deep autoencoder-like NMF methods (H. Huang, G. Zhou, Q. Zhao, et al. 2023; L. Zhao et al. 2023) suggest investigating nonlinear hierarchical mappings between the original data and the *representations*. MGANMF (L. Zhao et al. 2023) preserves the local structure at the top layer in different views, whereas MDANMF (H. Huang, G. Zhou, Q. Zhao, et al. 2023) aligns the *representations* with the consensus ones.

From a broader perspective, existing factorization-based multi-view clustering methods can be roughly divided into three groups. The first group consists of shallow matrix factorization methods, which mainly learn consensus or complementary representations across views but have limited capability to characterize hierarchical semantics. The second group extends matrix factorization to deep architectures, allowing multi-level representations to be progressively extracted. However, most of these methods still perform hierarchical decomposition independently for individual views and introduce cross-view constraints or consensus learning only on the resulting representations. Consequently, their cross-view interaction is predominantly pairwise and is usually concentrated at the top or selected decomposition layers. The third group employs tensor representations to

Table 1. The notations used in this paper.

Notation	Description
V	The number of views
$X^v \in \mathbb{R}^{d_v \times n}$	The data matrix of the v-th view
n	The number of samples
d_v	The feature dimension of the v-th view
m	The number of layers
r_l	The size of the l-th layer
$\mathcal{Z} \in \mathbb{R}^{k \times n \times V}$	The essential tensor representation
$\mathcal{E} \in \mathbb{R}^{k \times n \times V}$	The error tensor
$\mathcal{W}_i \in \mathbb{R}^{r_{l-1} \times r_l \times V}$	The basic tensor stacked from all views at the l-th layer
$\mathcal{H}_m \in \mathbb{R}^{r_m \times n \times V}$	The representation tensor stacked from all views at the m-th layer
$S \in \mathbb{R}^{n \times n}$	The consensus similarity matrix of data shared by all views
$A^v \in \mathbb{R}^{d_v \times k}$	The dictionary matrix in the v-th view
$C^v \in \mathbb{R}^{k \times n}$	The dictionary representation matrix in the v-th view

preserve multi-way relationships among different views and has shown promising capability in modeling high-order correlations. Recent tensor-based multi-view clustering methods further demonstrate the effectiveness of tensor structures for exploiting cross-view dependencies (J. Li, Gao, et al. 2023). Nevertheless, tensor representation construction and subsequent factorization or clustering are often treated as separate stages, while the hierarchical propagation of complementary information across multiple decomposition layers has received relatively limited attention.

3 Methodology

Different from the aforementioned approaches, NAHTF bridges deep matrix factorization and tensor-based multi-view learning within a unified framework. Instead of independently decomposing each view followed by late-stage representation alignment, NAHTF first projects heterogeneous views into a common-dimensional tensor representation and explicitly separates the essential low-rank component from adverse components. The resulting tensor is then directly subjected to deep semi-nonnegative tensor factorization through the tensor t-product. In this manner, high-order cross-view dependencies are preserved in the tensor representation and complementary information participates in the hierarchical decomposition process rather than being introduced only at the final representation layer. Moreover, tensor representation learning and hierarchical factorization are jointly optimized instead of being performed as two isolated stages. Therefore, NAHTF is positioned as a noise-aware deep tensor factorization framework that simultaneously addresses high-order cross-view modeling, hierarchical information fusion, and robust representation learning.

3.1 Notations and Preliminaries

Before the formal presentation of the proposed NAHTF method, it is essential to establish the necessary notations and introduce fundamental preliminaries. In this work, tensors are denoted by calligraphic letters (e.g., $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$), matrices are represented by uppercase letters (e.g., $A \in \mathbb{R}^{n_1 \times n_2}$), and column vectors are signified by bold lowercase letters (e.g., $\mathbf{a} \in \mathbb{R}^{n_1}$). In the following, let $\mathcal{A}^k$ represent the k-th frontal slice of $\mathcal{A}$, i.e., $\mathcal{A}^k = \mathcal{A}(:, :, k)$. A comprehensive summary of commonly used notations is presented in Table 1.

To ensure the theoretical integrity of the work, the tensor t-product (Lu et al. 2019) and tensor Schatten p-norm (Xia et al. 2022) are introduced as follows:

Definition 1. *(**Tensor t-product**): Given two third-order tensors $\mathscr{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathscr{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$, denote the operations bvec($\mathscr{A}$), fold(bvec($\mathscr{A}$)), and bcirc($\mathscr{A}$) as bvec($\mathscr{A}$) $= [\mathscr{A}^1; \mathscr{A}^2; \cdots; \mathscr{A}^{n_3}]$, fold(bvec($\mathscr{A}$)) $= \mathscr{A}$, and*

$$bcirc(\mathscr{A}) = \begin{bmatrix} \mathscr{A}^1 & \mathscr{A}^{n_3} & \cdots & \mathscr{A}^2 \\ \mathscr{A}^2 & \mathscr{A}^1 & \cdots & \mathscr{A}^3 \\ \vdots & \ddots & \ddots & \vdots \\ \mathscr{A}^{n_3} & \mathscr{A}^{n_3-1} & \cdots & \mathscr{A}^1 \end{bmatrix},$$

respectively. Then, the tensor t-product $\mathscr{A} * \mathscr{B}$ is defined as:

$$\mathscr{C} = \mathscr{A} * \mathscr{B} = fold(bcirc(\mathscr{A})bvec(\mathscr{B})). \tag{3}$$

Definition 2. *(**Tensor Schatten p-norm**): For a tensor $\mathscr{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its tensor Schatten p-norm is given by:*

$$\|\mathscr{A}\|_{Sp}^p = \Big(\sum_{k=1}^{n_3} \sum_{j=1}^{\min(n_1,n_2)} \sigma_j\big(\bar{\mathscr{A}}^k\big)^p \Big)^{\frac{1}{p}}, \tag{4}$$

where $\bar{\mathscr{A}} = fft(\mathscr{A}, [], 3)$ denotes the fast Fourier transformation of $\mathscr{A}$ along the third dimension, $\sigma_j(\bar{\mathscr{A}}^k)$ is the j-th singular value of the k-th frontal slice $\bar{\mathscr{A}}^k$, and $p \in (0, 1]$ controls the singular value shrinkage.

3.2 Deep Semi-Nonnegative Tensor Factorization

Given a multi-view dataset $X = \{X^1, X^2, ..., X^V\}$ containing n samples from V sources, conventional multi-view deep NMF methods adopt layer-wise matrix decomposition in Eq. (2). Through this hierarchical decomposition, deep matrix factorization methods aim to identify layer-specific attributes of data. However, these methods perform deep matrix factorization on each view individually and rely on late fusion strategies at the top layer to exploit complementary information across views. They neglect the complementary information at lower layers, leading to a loss of such information since the top-layer representations are dependent on the previous layers. Moreover, these methods only leverage pairwise correlations between views, overlooking the high-order correlations inherent in multiple views, which capture more complex interactions involving all data views.

To fully explore the rich complementary information, tensor representation has been widely used in multi-view scenarios (Gao et al. 2020; J. Wu et al. 2019). Let $\mathscr{X} \in \mathbb{R}^{k \times n \times V}$ be a third-order tensor representation of the multi-view data, where k denotes the number of features. The deep semi-nonnegative tensor factorization is formulated using the tensor t-product as:

$$\mathscr{X} \approx \mathscr{W}_1 * \mathscr{H}_1, \mathscr{H}_1 \approx \mathscr{W}_2 * \mathscr{H}_2, \cdots, \mathscr{H}_{m-1} \approx \mathscr{W}_m * \mathscr{H}_m, \tag{5}$$

where $\mathscr{W}_i \in \mathbb{R}^{r_{i-1} \times r_i \times V}$ and $\mathscr{H}_i \in \mathbb{R}_+^{r_i \times n \times V}$ denote the basis and nonnegative representation tensors, respectively. Notably, $\mathscr{W}_i$ and $\mathscr{H}_i$ hold the basis and representation matrices at the i-th layer as their frontal slices. The t-product operation circularly computes the multiplication between frontal slices of the input tensors, thereby enabling hierarchical propagation of complementary information and facilitating full fusion of multi-view complementary information at the top layer. To fulfill deep semi-nonnegative tensor factorization, the proposed NAHTF optimizes the problem:

$$\min_{\mathscr{W}_i, \mathscr{H}_m} \|\mathscr{X} - \mathscr{W}_1 * \mathscr{W}_2 * \cdots * \mathscr{W}_m * \mathscr{H}_m\|_F^2, \ s.t. \ \mathscr{H}_m \geq 0. \tag{6}$$

3.3 Noise-Aware Tensor Dictionary Learning

In real-world applications, multi-view data $\{X^v\}_{v=1}^V$ intrinsically consist of heterogeneous feature spaces with varying dimensionalities, making it infeasible to directly stack them into a tensor. To address this challenge, the proposed NAHTF presents an anchor-based tensor dictionary learning method to project the data from different views into the subspaces of the same dimension.

Specifically, NAHTF makes use of the VDA method (Xia et al. 2022) to systematically select discriminative anchors from the original data. Let $A^v = \{a_1, a_2, \dots, a_k\}$ denote the anchor matrix. These anchors subsequently serve as the basis for constructing a tensor representation $\mathscr{C} \in \mathbb{R}^{k\times n\times V}$ via the anchor-sample correlation formulation:

$$\min_{C^v} \sum_{v=1}^{V} \|X^v - A^v C^v\|_F^2,\ s.t.\ \mathscr{C} = \Phi(C^1, C^2, \dots, C^V). \tag{7}$$

where $C^v = [c_1^v, c_2^v, ..., c_n^v] \in \mathbb{R}^{k\times n}$ denotes the learnable dictionary representation matrix corresponding to the v-th data view. Notably, the function $\Phi(\cdot)$ constructs a third-order tensor $\mathscr{C}$ by organizing C^v as the v-th frontal slice of $\mathscr{C}$.

However, the aforementioned tensor representation may be inadequate for two critical reasons. First, the data often contain noise and redundant features, contaminating the new representation. Second, in the process of anchor selection, VDA simply concentrates all data matrices into a single data matrix, limiting its ability to explore underlying complementary information. To provide a better tensor representation, the noise-aware tensor dictionary learning is formulated as:

$$\begin{aligned} \min_{\mathscr{Z},\mathscr{E}} \sum_{v=1}^{V} \|X^v - A^v C^v\|_F^2 + \lambda_2 \|\psi(\mathscr{Z})\|_{sp}^p + \lambda_3 \|\mathscr{E}\|_{2,1}, \\ s.t.\ \mathscr{C} = \Phi(C^1, C^2, \dots, C^V),\ \mathscr{C} = \mathscr{Z} + \mathscr{E}, \end{aligned} \tag{8}$$

where $\|\mathscr{E}\|_{2,1} = \sum_{v=1}^{V} \sqrt{\sum_{i=1}^{k}\sum_{j=1}^{n} \mathscr{E}(i,j,v)^2}$ (J. Wu et al. 2019), $\|\psi(\mathscr{Z})\|_{sp}^p$ is the Schatten p-norm of $\psi(\mathscr{Z})$ (Gao et al. 2020), and $\psi(\mathscr{Z})$ rotates $\mathscr{Z}$ into a tensor of $k\times V\times n$, i.e., $\psi(Z) \in \mathbb{R}^{k\times V\times n}$. This joint optimization framework promotes a synergistic interaction between tensor dictionary learning and deep tensor factorization. The learned tensor representation $\mathscr{C}$ improves the quality of the tensor representation $\mathscr{Z}$, while the low-rank structure of $\mathscr{Z}$ regularizes the dictionary learning process.

Optimizing the above problem offers two advantages. First, it reveals the low-rank structure and explores the high-order correlations of multi-view data through $\|\psi(\mathscr{Z})\|_{sp}^p$ (Xia et al. 2022), which is consistent with the motivation of tensor factorization. Second, it removes the adverse components from the data using $\|\mathscr{E}\|_{2,1}$ based on complementary information (J. Wu et al. 2019).

3.4 Local Consistency Exploration

The local structure is generally used as supervised information in unsupervised learning. To fully capture the local consistency across views, the local similarities S between data across views are derived based on the *representations* using a learning mechanism (S. Huang, Z. Xu, et al. 2017) as:

$$\min_{S} \sum_{v=1}^{V}\sum_{i\neq j}^{n} \|(h_m)_i^v - (h_m)_j^v\|_2^2 S_{ij}^2,\ s.t.\ S_{ij} \geq 0, \sum_{j=1}^{n} S_{ij} = 1, S_{ii} = 0, \tag{9}$$

where $(h_m)_i^v$ is the i-th column of the frontal slice $\mathscr{H}_m^v$.

3.5 Model

The proposed *Noise-Aware Hierarchical Tensor Fusion* model aims to fuse the complementary information of multi-view data through the propagation of high-order cross-view complementary information across the layers of hierarchical tensor factorization. This is achieved by alternating between extracting a high-quality tensor representation, performing deep semi-nonnegative tensor factorization, and capturing local consistency information. By combining Eqs. (6), (8), and (9), the proposed NAHTF is formulated as follows:

$$\begin{aligned}\min_{\Theta}\ & \underbrace{\|\mathcal{Z}-\mathcal{W}_1*\mathcal{W}_2*\cdots*\mathcal{W}_m*\mathcal{H}_m\|_F^2}_{\text{Deep semi-nonnegative tensor factorization}}+\underbrace{\frac{\lambda_1}{2}\sum_{v=1}^{V}\sum_{i\neq j}^{n}\|(h_m)_i^v-(h_m)_j^v\|_2^2 S_{ij}^2}_{\text{Local consistency exploration}} \\ & \underbrace{+\lambda_2\|\psi(\mathcal{Z})\|_{sp}^p+\lambda_3\|\mathcal{E}\|_{2,1}+\lambda_4\sum_{v=1}^{V}\|X^v-A^vC^v\|_F^2}_{\text{Noise-aware tensor dictionary learning}}, \\ & s.t.\ \mathcal{C}=\Phi(C^1,C^2,...,C^V),\mathcal{C}=\mathcal{Z}+\mathcal{E}, \\ & \mathcal{H}_m\geq 0, S_{ij}\geq 0, \sum_{j=1}^{n}S_{ij}=1, S_{ii}=0,\end{aligned} \tag{10}$$

where $\Theta=\{C^v,\mathcal{Z},\mathcal{E},\mathcal{W}_i,\mathcal{H}_m,S\}$ contains learnable variables.

4 Optimization

The optimization problem in Eq. (10) is non-jointly convex with respect to all variables. Therefore, this section presents an alternating algorithm based on the Alternating Direction Multiplier Method (ADMM) (Nishihara et al. 2015) to optimize this problem.

4.1 Augmented Lagrangian Function

Let $\{\mathcal{Y}_i\}_{i=1}^4$ and $\{\mu_i\}_{i=1}^4$ be Lagrangian multipliers and penalty parameters, respectively. Define the Laplacian matrix L_s on S as $L_s=D_s-\bar{S}$, where $\bar{S}_{ij}=\frac{S_{ij}^2+S_{ji}^2}{2}$ and D_s is a diagonal matrix with $(D_s)_{ii}=\sum_j\bar{S}_{ij}$. By introducing three auxiliary variables $\mathcal{J}_1$, $\mathcal{J}_2$, and $\mathcal{G}$ into Eq. (10) with $\mathcal{J}_1=\mathcal{J}_2=\mathcal{H}_m$ and $\mathcal{G}=\psi(\mathcal{Z})$, the augmented Lagrangian function is defined as:

$$\begin{aligned}\min\ & \|\mathcal{Z}-\mathcal{W}_1*\mathcal{W}_2*\cdots*\mathcal{W}_m*\mathcal{H}_m\|_F^2+\lambda_1\sum_{v=1}^{V}tr(\mathcal{J}_2^vL_s\mathcal{J}_2^{vT}) \\ & +\lambda_2\|\mathcal{G}\|_{sp}^p+\lambda_3\|\mathcal{E}\|_{2,1}+\sum_{v=1}^{V}\lambda_4\|X^v-A^vC^v\|_F^2+\frac{\mu_1}{2}\|\mathcal{C}-(\mathcal{Z}+\mathcal{E})+\frac{\mathcal{Y}_1}{\mu_1}\|_F^2 \\ & +\frac{\mu_2}{2}\|\mathcal{H}_m-\mathcal{J}_1+\frac{\mathcal{Y}_2}{\mu_2}\|_F^2+\frac{\mu_3}{2}\|\mathcal{H}_m-\mathcal{J}_2+\frac{\mathcal{Y}_3}{\mu_3}\|_F^2+\frac{\mu_4}{2}\|\psi(\mathcal{Z})-\mathcal{G}+\frac{\mathcal{Y}_4}{\mu_4}\|_F^2, \\ & s.t.\ \mathcal{C}=\Phi(C^1,C^2,...,C^V),\mathcal{J}_1\geq 0,S_{ij}\geq 0,\sum_{j=1}^{n}S_{ij}=1,S_{ii}=0.\end{aligned} \tag{11}$$

The subsequent subsection details the optimization process designed to optimize the augmented Lagrangian function.

4.2 Optimization Procedure

Update for $\mathscr{C}$: Fixing the other variables, the optimization of $\mathscr{C}$ can be derived by individually optimizing its frontal slices C^v as:

$$\begin{aligned} &\min \lambda_4 \|X^v - A^v C^v\|_F^2 + \frac{\mu_1}{2}\|C^v - (\mathscr{Z}^v + \mathscr{E}^v) + \frac{\mathscr{Y}_1^v}{\mu_1}\|_F^2, \\ &s.t.\ \mathscr{C} = \Phi(C^1, C^2, ..., C^V). \end{aligned} \tag{12}$$

Setting the derivative of Eq. (12) with respect to C^v to zero yields the closed-form update:

$$C^v = (2\lambda_4 A^{vT} A^v + \mu_1 I)^{-1}(2\lambda_4 A^{vT} X^v + \mu_1 \mathscr{Z}^v + \mu_1 \mathscr{E}^v - \mathscr{Y}_1^v). \tag{13}$$

Update for $\mathscr{Z}$: Let $\mathscr{F} = \mathscr{W}_1 * \mathscr{W}_2 * \cdots * \mathscr{W}_m * \mathscr{H}_m$. Then, with the other variables fixed, the optimal $\mathscr{Z}$ to Eq. (11) can be solved by optimizing its frontal slices $\mathscr{Z}^v$ as:

$$\begin{aligned} \min_{\mathscr{Z}^v} \|\mathscr{Z}^v - \mathscr{F}^v\|_F^2 + \frac{\mu_4}{2}\|\mathscr{Z}^v - \psi^{-1}(\mathscr{G})^v + \frac{\psi^{-1}(\mathscr{Y}_4)^v}{\mu_4}\|_F^2 \\ + \frac{\mu_1}{2}\|C^v - (\mathscr{Z}^v + \mathscr{E}^v) + \frac{\mathscr{Y}_1^v}{\mu_1}\|_F^2, \end{aligned} \tag{14}$$

where $\psi^{-1}(\cdot)$ represents the inverse operation of $\psi(\cdot)$, i.e., $\psi^{-1}(\mathscr{G})$ rotates $\mathscr{G}$ to a tensor of $k \times n \times V$.

Setting the derivative of Eq. (14) with respect to $\mathscr{Z}^v$ to 0, the optimal $\mathscr{Z}^v$ is deduced as:

$$\mathscr{Z}^v = \frac{2\mathscr{F}^v + \mu_1(C^v - \mathscr{E}^v) + \mathscr{Y}_1^v + \mu_4\psi^{-1}(\mathscr{G})^v - \psi^{-1}(\mathscr{Y}_4)^v}{2 + \mu_1 + \mu_4}. \tag{15}$$

Update for $\mathscr{W}_i$: Given the other variables, optimizing Eq. (11) concerning $\mathscr{W}_i$ can be conducted in the Fourier domain:

$$\min_{\bar{W}_i^v} \|\bar{\mathscr{Z}}^v - \bar{\mathscr{W}}_1^v \bar{\mathscr{W}}_2^v \cdots \bar{\mathscr{W}}_m^v \bar{\mathscr{H}}_m^v\|_F^2. \tag{16}$$

Setting its derivative with respect to $\bar{\mathscr{W}}_i^v$ to zero yields the optimal $\bar{\mathscr{W}}_i^v$ as:

$$\bar{\mathscr{W}}_i^v = (\bar{\mathscr{W}}_1^v \cdots \bar{\mathscr{W}}_{i-1}^v)^\dagger \bar{\mathscr{Z}}^v (\bar{\mathscr{W}}_{i+1}^v \cdots \bar{\mathscr{W}}_m^v \bar{\mathscr{H}}_m^v)^\dagger, \tag{17}$$

where $\bar{\mathscr{W}}_i = \text{fft}(\mathscr{W}_i, [], 3)$ represents the fast Fourier Transform (FFT) of $\mathscr{W}_i$ along the third dimension, and $(\cdot)^\dagger$ denotes the pseudo-inverse of a given matrix.

The optimal $\mathscr{W}_i$ is then obtained via inverse FFT as: $\mathscr{W}_i = \text{ifft}(\bar{\mathscr{W}}_i, [], 3)$.

Update for $\mathscr{H}_m$: Let $\mathscr{K} = \mathscr{W}_1 * \mathscr{W}_2 * \cdots * \mathscr{W}_m$. When the other variables are fixed, the solution of $\mathscr{H}_m$ to Eq. (11) can be derived in the Fourier domain by optimizing:

$$\min_{\bar{\mathscr{H}}_m^v} \|\bar{\mathscr{Z}}^v - \bar{\mathscr{K}}^v \bar{\mathscr{H}}_m^v\|_F^2 + \frac{\mu_2}{2}\|\bar{\mathscr{H}}_m^v - \bar{\mathscr{J}}_1^v + \frac{\bar{\mathscr{Y}}_2^v}{2}\| + \frac{\mu_3}{2}\|\bar{\mathscr{H}}_m^v - \bar{\mathscr{J}}_2^v + \frac{\bar{\mathscr{Y}}_3^v}{2}\|. \tag{18}$$

The optimal $\bar{\mathscr{H}}_m^v$ is given by:

$$\begin{aligned} \bar{\mathscr{H}}_m^v = (2\bar{\mathscr{K}}^{vT}\bar{\mathscr{K}}^v + (\mu_2 + \mu_3)I)^{-1} \\ (2\bar{\mathscr{K}}^{vT}\bar{\mathscr{Z}}^v + (\mu_2\bar{\mathscr{J}}_1^v - \bar{\mathscr{Y}}_2^v) + (\mu_3\bar{\mathscr{J}}_2^v - \bar{\mathscr{Y}}_3^v)). \end{aligned} \tag{19}$$

Similarly, $\mathscr{H}_m = \text{ifft}(\bar{\mathscr{H}}_m, [], 3)$ gives the optimal $\mathscr{H}_m$.

Update for $\mathcal{J}_1$: When the other variables are fixed, the optimization problem for $\mathcal{J}_1$ in Eq. (11) reduces to:

$$\min \ \|\mathcal{H}_m - \mathcal{J}_1 + \mathcal{Y}_2/\mu_2\|_F^2, \ s.t. \ \mathcal{J}_1 \geq 0. \tag{20}$$

The optimal solution is:

$$\mathcal{J}_1 = \max(\mathcal{H}_m + \mathcal{Y}_2/\mu_2, 0). \tag{21}$$

Update for $\mathcal{J}_2$: Given the other variables, the optimization problem for $\mathcal{J}_2$ can be resolved by individually optimizing its frontal slices:

$$\min \ \lambda_1 tr(\mathcal{J}_2^v L_s \mathcal{J}_2^{vT}) + \frac{\mu_3}{2}\|\mathcal{H}_m^v - \mathcal{J}_2^v + \frac{\mathcal{Y}_3^v}{\mu_3}\|_F^2. \tag{22}$$

The optimal $\mathcal{J}_2$ is given by:

$$\mathcal{J}_2^v = \left(\mu_3 \mathcal{H}_m^v + \mathcal{Y}_3^v\right)\left(2\lambda_1 L_s + \mu_3 I\right)^{-1}. \tag{23}$$

Update for S: Let $\beta_{ij} = \|(j_2)_i^v - (j_2)_j^v\|^2$, where $(j_2)_i^v$ is the i-th column of the frontal slice $\mathcal{J}_2^v$. With the other variables fixed, the optimization problem for S becomes:

$$\min \sum_{i,j=1}^{n} \beta_{ij} S_{ij}^2, s.t. \ S_{ij} \geq 0, \sum_{j=1}^{n} S_{ij} = 1, S_{ii} = 0. \tag{24}$$

By setting the derivative of its Lagrangian function with respect to S_{ij} to 0, the update rule for S is given by:

$$S_{ij} = \frac{(\beta_{ij})^{-1}}{\sum_{\substack{l=1 \\ l\neq i}}^{n} (\beta_{il})^{-1}}, \quad j \neq i, \qquad S_{ii} = 0. \tag{25}$$

when the nonnegative constraint $S_{ij} \geq 0$ is ignored.

Evidently, the S_{ij} given in Eq. (25) is nonnegative.

Update for $\mathcal{G}$: With the other variables fixed, the optimization problem for $\mathcal{G}$ is:

$$\min \ \frac{\lambda_2}{\mu_4}\|\mathcal{G}\|_{Sp}^p + \frac{1}{2}\|\mathcal{G} - (\psi(\mathcal{Z}) + \frac{\mathcal{Y}_4}{\mu_4})\|_F^2. \tag{26}$$

Theorem 1 (H. Yang et al. 2022) gives the optimal $\mathcal{G}$ to Eq. (11).

Theorem 1. *Given a third-order tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, let $\mathcal{A} = \mathcal{U} * \mathcal{D} * \mathcal{V}^T$ be the tensor Singular Value Decomposition (t-SVD) (J. Wu et al. 2019) of $\mathcal{A}$. Then, the solution to the following problem*

$$\min \sigma\|\mathcal{G}\|_{Sp}^p + \frac{1}{2}\|\mathcal{G} - \mathcal{A}\|_F^2 \tag{27}$$

is given by:

$$\mathcal{G}^* = \Gamma_{\sigma \cdot n_3}(\mathcal{A}) = \mathcal{U} * ifft(F_{\sigma \cdot n_3}(\bar{\mathcal{A}})) * \mathcal{V}^T, \tag{28}$$

where $F_{\sigma \cdot n_3}(\bar{\mathcal{A}}) \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an f-diagonal tensor with its diagonal elements determined using the generalized shrinkage thresholding algorithm (Zuo et al. 2013).

Update for $\mathcal{E}$: Let $E_{(3)}$ denote the matrix version of the tensor $\mathcal{E}$ along the mode-3 and $\mathcal{D} = \mathcal{C} - \mathcal{Z} + \frac{\mathcal{Y}_1}{\mu_1}$. Then, given the other variables, the optimization problem for $\mathcal{E}$ takes the form:

$$\min \ \lambda_3\|E_{(3)}\|_{2,1} + \frac{\mu_1}{2}\|E_{(3)} - D_{(3)}\|_F^2, \tag{29}$$

Algorithm 1: Algorithm for the proposed NAHTF

Input: Multi-view Data $\{X^v\}_{v=1}^V$, number of layers m, layer sizes $\{r_i\}_{i=1}^m$, number of anchors k, trade-off parameters $\{\lambda_i\}_{i=1}^4$.
Output: Representation matrices $\{H_m^v\}_{v=1}^V$.
Initialize $\mathcal{W}$,$\mathcal{H}$; $\mathcal{G}$,$\mathcal{J}_1$,$\mathcal{J}_2$,$\mathcal{Y}_1$, $\mathcal{Y}_2$, $\mathcal{Y}_3$, $\mathcal{Y}_4$ to 0; $\mathcal{Z} = \mathcal{C}$; $\mu_1 = \mu_2 = \mu_3 = \mu_4 = 1e-5$, and $\rho = 1.7$;
while *not converged* **do**
  Update C^v and $\mathcal{Z}$ via Eqs. (13) and (15), respectively;
  Update $\mathcal{W}_i$, and $\mathcal{H}_m$ after getting $\bar{\mathcal{W}_i}$ and $\bar{\mathcal{H}_m}$ via Eqs. (17) and (19), respectively;
  Update $\mathcal{J}_1$, $\mathcal{J}_2$, S, $\mathcal{G}$, and $\mathcal{E}$ via Eq. (21), Eq. (23), Eq. (25), Eq. (28), and Eq. (30), respectively;
  Update $\{\mathcal{Y}_i\}_{i=1}^4$ and $\{\mu_i\}_{i=1}^4$ using Eqs. (31)-(35).
end

where $\|E_{(3)}\|_{2,1} = \sum_i \|(e_{(3)})_i\|_2$. It has a closed-form solution (J. Wu et al. 2019):

$$(e_{(3)})_i = \begin{cases} \frac{\|(d_{(3)})_i\|_2 - \frac{\lambda_3}{\mu_1}}{\|(d_{(3)})_i\|_2}(d_{(3)})_i, & if \|(d_{(3)})_i\|_2 > \frac{\lambda_3}{\mu_1}, \\ 0, & otherwise. \end{cases} \tag{30}$$

After obtaining $E_{(3)}$, the tensor $\mathcal{E}$ can be retrieved by folding $E_{(3)}$ along the mode-3.

Update for $\{\mathcal{Y}_i\}_{i=1}^4$ and $\{\mu_i\}_{i=1}^4$: To ensure the convergence of the optimization process, the Lagrangian multipliers $\{\mathcal{Y}_i\}_{i=1}^4$ and penalty parameters $\{\mu_i\}_{i=1}^4$ are updated via the following rules:

$$\mathcal{Y}_1 = \mathcal{Y}_1 + \mu_1(\mathcal{C} - \mathcal{Z} - \mathcal{E}), \tag{31}$$
$$\mathcal{Y}_2 = \mathcal{Y}_2 + \mu_2(\mathcal{H}_m - \mathcal{J}_1), \tag{32}$$
$$\mathcal{Y}_3 = \mathcal{Y}_3 + \mu_3(\mathcal{H}_m - \mathcal{J}_2), \tag{33}$$
$$\mathcal{Y}_4 = \mathcal{Y}_4 + \mu_4(\psi(\mathcal{Z}) - \mathcal{G}), \tag{34}$$
$$\mu_i = \min(\rho\mu_i, 1e^{10}), i = 1, 2, 3, 4. \tag{35}$$

The overall optimization procedure is detailed in Algorithm 1. When Algorithm 1 converges, the consensus representation matrix is given by optimizing:

$$\min \sum_{v=1}^{V} \|H^* - H_m^v\|_F^2, \tag{36}$$

and the optimal H^* is given by $H^* = \frac{1}{V}\sum_{v=1}^V H_m^v$. Finally, the k-means algorithm is adopted to generate the final discrete clustering results.

4.3 Convergence and Complexity

Algorithm 1 optimizes Eq. (10) through the optimization of the augmented Lagrangian function Eq. (11), involving the primal variables $\{C^v, \mathcal{Z}, \mathcal{E}, \{\mathcal{W}_i\}_{i=1}^m, \mathcal{H}_m, S\}$, auxiliary variables $\{\mathcal{J}_1, \mathcal{J}_2, \mathcal{G}\}$, Lagrangian multipliers $\{\mathcal{Y}_i\}_{i=1}^4$, and penalty variables $\{\mu_i\}_{i=1}^4$. The optimization utilizes closed-form solutions to optimize the primal and auxiliary variables, thus reducing the objective function value of the augmented Lagrangian function Eq. (11). Additionally, the ADMM algorithm has been theoretically proven to converge to a local minimum. Consequently, despite occasional increases in the objective value due to the nature of the ADMM algorithm, Algorithm 1 can

Table 2. Details of benchmark datasets.

Dataset	#Samples	#Features	#Classes	#Views	Type
Yale	165	4096/3304/6750	15	3	Face
Orl	400	4096/3304/6750	40	3	Face
20Newsgroups	500	2000/2000/2000	5	3	Text
BBCSport	544	3183/3203	5	2	Text
BBC	685	4659, 4633/4665/4684	5	4	Text
Caltech101-7	1474	48/40/254/1984/512/928	7	6	Image

effectively optimize the model Eq. (10). The experimental results presented in the experiment section further demonstrate the convergence of the algorithm.

As previously mentioned, the primary operations in Algorithm 1 involve the optimization of the primal and auxiliary variables. Among these, the optimization of $\mathcal{J}_2$ is the most computationally expensive, as it requires calculating the inverse of the n-by-n matrix $2\lambda_1 L_s + \mu_3 I$, with a computational cost of $O(n^3)$. As a result, the overall computational complexity of the proposed method is $\mathcal{O}(n^3 T)$, where T is the number of iterations.

5 Experiment

In this section, the proposed NAHTF is compared with fourteen state-of-the-art multi-view clustering methods, including matrix factorization-based methods and tensor-based methods, on six widely used multi-view datasets to assess their clustering performance. Specifically, an introduction to the benchmark datasets, baseline methods, and evaluation metrics is presented. Subsequently, the experimental settings are detailed, followed by an analysis of the experimental results. Additionally, an investigation into parameter sensitivity is conducted, along with ablation experiments. Finally, the convergence of the algorithm is analyzed.

5.1 Datasets and Baseline Methods

The used benchmark datasets consist of three text datasets (i.e., **20Newsgroups**[1] (J. Li, G. Zhou, et al. 2020), **BBCSport**[2] (H. Huang, G. Zhou, Q. Zhao, et al. 2023), and **BBC**[2] (J. Li, G. Zhou, et al. 2020)), two face datasets (i.e., **Yale**[3] (Luong, Nayak, et al. 2022) and **ORL**[4] (M.-S. Chen, Z.-Y. Li, et al. 2024)), and one image dataset (i.e., **Caltech101_7**[5] (Kong et al. 2025)). Table 2 provides an overview of the statistics of these datasets.

In this paper, the proposed NAHTF is compared with fourteen state-of-the-art multi-view methods, which fall into four categories: shallow matrix factorization methods (i.e., **MultiNMF**[6] (J. Liu, C. Wang, et al. 2013) and **2CMV**[7] (Luong and Nayak 2020)), deep matrix factorization methods (i.e., **DMF-MVC**[8] (H. Zhao et al. 2017), **MvDGNMF**[9] (J. Li, G. Zhou, et al. 2020), **ODD-NMF**[10] (Luong, Nayak, et al. 2022), **DANMF-MRL**[11] (H. Huang,

[1] https://lig-membres.imag.fr/grimal/data.html.
[2] http://mlg.ucd.ie/datasets/segment.html.
[3] http://cvc.yale.edu/projects/yalefaces/yalefaces.html.
[4] http://www.uk.research.att.com/facedatabase.html.
[5] http://www.vision.caltech.edu/Image-Datasets/Caltech101/.
[6] https://jialu.info/.
[7] https://github.com/khanhluongds/Multi-view-Clustering-2CMV.
[8] https://github.com/hdzhao/DMF_MVC.
[9] https://github.com/chenglin-lcl/MvDGNMF.
[10] https://github.com/khanhluongds/Multi-view-Clustering-ODD-NMF.
[11] https://github.com/libertyhhn/DANMF-MRL.

Table 3. Clustering results (ACC±std %) and t-test results between NAHTF and baselines on the benchmark datasets.

Method	Yale	Orl	20Newsgroups	BBCSport	BBC	Caltech101_7	Avg. Rank
MultiNMF (J. Liu, C. Wang, et al. 2013)	56.67±3.74 (0.00)	69.30±3.09 (0.00)	73.74±1.60 (0.00)	78.60±0.40 (0.00)	55.78±5.52 (0.00)	44.65±2.92 (0.00)	11.33
2CMV (Luong and Nayak 2020)	56.24±3.32 (0.00)	65.43±2.96 (0.00)	66.60±1.58 (0.00)	78.46±1.32 (0.00)	73.53±0.06 (0.00)	63.51±4.88 (0.00)	10.00
DMF-MVC (H. Zhao et al. 2017)	60.55±3.29 (0.00)	74.58±2.58 (0.00)	49.58±4.12 (0.00)	75.88±6.78 (0.00)	53.05±4.69 (0.00)	63.77±4.94 (0.00)	10.17
MvDGNMF (J. Li, G. Zhou, et al. 2020)	54.67±5.11 (0.00)	61.18±4.10 (0.00)	91.04±3.08 (0.00)	84.85±6.99 (0.00)	76.48±7.03 (0.00)	57.89±5.22 (0.00)	8.67
ODD-NMF (Luong, Nayak, et al. 2022)	67.45±4.37 (0.00)	60.90±2.75 (0.00)	39.34±2.27 (0.00)	74.85±1.47 (0.00)	62.53±3.92 (0.00)	70.52±4.76 (0.00)	10.17
DANMF-MRL (H. Huang, G. Zhou, Q. Zhao, et al. 2023)	53.58±2.77 (0.00)	54.48±1.80 (0.00)	86.78±0.87 (0.00)	82.35±6.60 (0.00)	68.06±3.83 (0.00)	48.11±0.02 (0.00)	11.17
OS-DANMF-MRL (H. Huang, G. Zhou, Q. Zhao, et al. 2023)	45.64±1.02 (0.00)	66.33±3.42 (0.00)	87.94±0.96 (0.00)	72.54±1.37 (0.00)	75.52±1.31 (0.00)	45.45±0.52 (0.00)	11.50
Orth-NTF (J. Li, Gao, et al. 2023)	80.00±0.00 (-)	75.25±0.00 (-)	81.00±0.00 (-)	86.03±0.00 (-)	90.07±0.00 (-)	57.80±0.00 (-)	5.83
S2MVTC (Long et al. 2024)	78.79±0.00 (-)	77.50±0.00 (-)	79.00±0.00 (-)	79.60±0.00 (-)	77.52±0.00 (-)	69.06±0.00 (-)	5.83
DSTL (D. Xu et al. 2024)	47.70±1.85 (0.00)	74.88±1.95 (0.00)	93.22±0.06 (0.00)	94.94±0.07 (0.00)	76.37±0.12 (0.00)	62.91±0.38 (0.00)	7.17
SGTL (M.-S. Chen, Z.-Y. Li, et al. 2024)	55.94±3.73 (0.00)	85.05±2.72 (0.18)	97.80±4.75 (0.04)	97.54±3.66 (0.00)	63.26±6.99 (0.00)	67.37±5.77 (0.00)	5.50
TPCH (Z. Wang et al. 2025)	50.30±0.00 (0.00)	82.75±0.00 (0.00)	91.80±0.00 (0.00)	71.14±0.00 (0.00)	77.08±0.00 (0.00)	56.72±0.00 (0.00)	8.67
LMTC (J. Liu, X. Liu, et al. 2025)	66.18±2.91 (0.00)	78.13±2.95 (0.00)	94.28±0.08 (0.00)	90.26±0.18 (0.00)	79.66±0.19 (0.00)	44.46±2.90 (0.00)	5.83
TADMVC (Y. Chen et al. 2025)	63.82±2.25 (0.00)	70.25±3.55 (0.00)	97.94±0.13 (0.00)	87.00±0.80 (0.00)	70.22±0.56 (0.00)	53.47±3.55 (0.00)	7.17
NAHTF(Ours)	**84.48 ± 3.44**	**86.58 ± 2.90**	**98.18 ± 0.08**	**99.26 ± 0.08**	**96.04 ± 0.24**	**78.15 ± 2.51**	**1.00**

G. Zhou, Q. Zhao, et al. 2023), and **OS-DANMF-MRL**[11] (H. Huang, G. Zhou, Q. Zhao, et al. 2023)), tensor-based methods (i.e., **Orth-NTF**[12] (J. Li, Gao, et al. 2023), **S2MVTC**[13] (Long et al. 2024), **DSTL**[14] (D. Xu et al. 2024), **SGTL**[15] (M.-S. Chen, Z.-Y. Li, et al. 2024), **TPCH**[16] (Z. Wang et al. 2025), and **LMTC**[17] (J. Liu, X. Liu, et al. 2025)), and anchor-based method (i.e., **TADMVC**[18] (Y. Chen et al. 2025)).

5.2 Evaluation Metrics and Experiment Settings

In the experimental setup for the proposed NAHTF, the number of layers is configured as 3 throughout the experiments. The layer sizes from the first to the top layer are set to $10c$, $5c$, and c, respectively, where c represents the number of data classes. The quantity of anchors is fixed at $0.5n$, where n is the number of samples. The parameter p within the Schatten p-norm controls the rank of the tensor $\mathcal{Z}$. It is tuned from 0.1 to 1 with an increment of 0.1. The trade-off parameters λ_1, λ_2, λ_3, and λ_4 balance the contributions of local consistency exploration, tensor representation extraction, adverse component elimination, and dictionary representation learning within the

[12]https://github.com/xdjingli/Orth-NTF.
[13]https://github.com/longzhen520/S2MVTC.
[14]https://github.com/dengxu-nju/DSTL.
[15]https://github.com/ManshengChen/Code-for-SGTL-master.
[16]https://github.com/jankin-wang/TPCH.
[17]https://github.com/liujiyuan13/LMTC-code_release.
[18]https://github.com/whbdmu/TAD-MVC.

Table 4. Clustering results (NMI±std %) and t-test results between NAHTF and baselines on the benchmark datasets.

Method	Yale	Orl	20Newsgroups	BBCSport	BBC	Caltech101_7	Avg. Rank
MultiNMF (J. Liu, C. Wang, et al. 2013)	59.04±2.95 (0.00)	83.38±1.11 (0.00)	58.56±3.41 (0.00)	68.00±0.26 (0.00)	47.33±2.78 (0.00)	38.50±3.09 (0.00)	12.67
2CMV (Luong and Nayak 2020)	63.92±1.23 (0.00)	81.20±1.43 (0.00)	53.32±1.01 (0.00)	67.65±0.50 (0.00)	61.27±0.50 (0.00)	44.77±0.91 (0.00)	10.67
DMF-MVC (H. Zhao et al. 2017)	64.47±3.97 (0.00)	87.40±3.97 (0.00)	24.03±0.62 (0.00)	61.63±3.80 (0.00)	27.91±3.19 (0.00)	44.08±0.91 (0.00)	11.67
MvDGNMF (J. Li, G. Zhou, et al. 2020)	60.58±3.40 (0.00)	77.50±2.61 (0.00)	81.68±4.42 (0.00)	76.70±5.23 (0.00)	60.95±7.05 (0.00)	49.56±3.65 (0.00)	8.83
ODD-NMF (Luong, Nayak, et al. 2022)	69.70±4.24 (0.00)	77.27±1.54 (0.00)	18.45±3.69 (0.00)	64.76±4.50 (0.00)	48.10±4.15 (0.00)	45.45±5.02 (0.00)	11.67
DANMF-MRL (H. Huang, G. Zhou, Q. Zhao, et al. 2023)	61.20±2.10 (0.00)	73.61±0.83 (0.00)	72.53±5.08 (0.00)	74.50±6.92 (0.00)	52.55±6.03 (0.00)	43.86±0.16 (0.00)	11.33
OS-DANMF-MRL (H. Huang, G. Zhou, Q. Zhao, et al. 2023)	57.76±3.96 (0.00)	83.29±2.51 (0.00)	82.59±1.00 (0.00)	72.73±2.13 (0.00)	61.07±1.65 (0.00)	45.50±0.62 (0.00)	9.67
Orth-NTF (J. Li, Gao, et al. 2023)	83.53±0.00 (-)	87.27±0.00 (-)	70.53±0.00 (-)	74.37±0.00 (-)	84.33±0.00 (-)	54.03±0.00 (-)	6.33
S2MVTC (Long et al. 2024)	84.61±0.00 (-)	88.54±0.00 (-)	79.46±0.00 (-)	85.13±0.00 (-)	66.49±0.00 (-)	**68.46**±0.00 (-)	4.17
DSTL (D. Xu et al. 2024)	59.98±1.64 (0.00)	87.39±1.18 (0.00)	94.80±0.17 (0.00)	93.87±4.89 (0.00)	71.03±0.04 (0.00)	51.98±1.99 (0.00)	5.67
SGTL (M.-S. Chen, Z.-Y. Li, et al. 2024)	64.95±3.89 (0.00)	92.10±1.01 (0.18)	95.99±5.04 (0.00)	96.96±1.22 (0.00)	58.74±4.07 (0.00)	63.98±3.58 (0.00)	4.33
TPCH (Z. Wang et al. 2025)	52.62±0.00 (-)	90.94±0.00 (-)	81.07±0.00 (-)	73.12±0.00 (-)	66.49±0.00 (-)	59.35±0.00 (-)	7.50
LMTC (J. Liu, X. Liu, et al. 2025)	69.42±3.67 (0.00)	89.33±1.51 (0.00)	84.38±0.35 (0.00)	81.48±0.29 (0.00)	60.75±0.23 (0.00)	40.99±1.79 (0.00)	7.00
TADMVC (Y. Chen et al. 2025)	67.82±4.36 (0.00)	86.14±2.74 (0.00)	94.06±0.49 (0.27)	75.14±1.04 (0.00)	53.14±0.47 (0.00)	49.64±3.57 (0.00)	7.33
NAHTF(Ours)	**86.42**±1.59	**94.78**±1.26	**96.19**±0.22	**97.31**±0.16	**87.59**±0.55	67.96±3.33	**1.17**

NAHTF framework. These parameters are tuned through a grid search within $[10^0, 10^1, ..., 10^6]$. To achieve fairness in comparison, the parameter settings for the baseline methods are tuned following the recommendations provided by their respective authors, and the optimal performance of each baseline method is recorded.

All the compared methods are pre-trained using the basic NMF algorithm. The maximum number of pre-training iterations is set to 100. As a preprocessing step, each sample in every view is normalized to have unit l_2-norm. For the proposed NAHTF and the majority of the baseline methods, after obtaining the consensus representation matrix H^*, this matrix is fed to the k-means algorithm to generate clustering results. Notable exceptions are one-step methods that directly produce the clustering without any post-processing, along with SGTL, which employ spectral clustering instead. All methods are repeated 10 times with distinct initializations, and the average performance across these repetitions is reported in the subsequent subsections. All experiments are conducted on a desktop system with an Intel Core Ultra 7 265K CPU (3.90 GHz), 32 GB of RAM, running Windows 11, and using MATLAB R2021a (64-bit).

5.3 Experiment Results

To evaluate the clustering performance of the proposed NAHTF compared with the baseline methods, clustering experiments are conducted, with the performance measured using three commonly used evaluation metrics. The

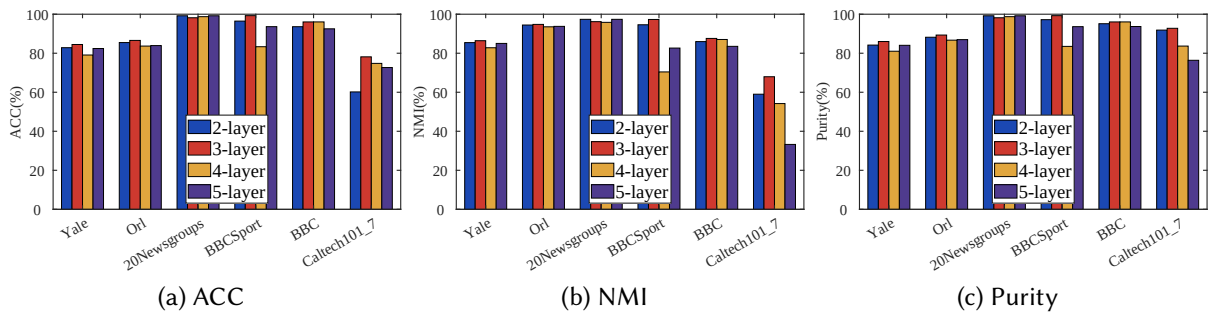
Table 5. Clustering results (Purity±std %) and t-test results between NAHTF and baselines on the benchmark datasets.

Method	Yale	Orl	20Newsgroups	BBCSport	BBC	Caltech101_7	Avg. Rank
MultiNMF (J. Liu, C. Wang, et al. 2013)	56.85±3.64 (0.00)	71.93±2.28 (0.00)	74.46±4.65 (0.00)	81.14±0.07 (0.00)	69.49±0.12 (0.00)	82.21±2.42 (0.00)	12.00
2CMV (Luong and Nayak 2020)	57.45±2.07 (0.00)	67.98±2.42 (0.00)	69.06±0.83 (0.00)	84.14±0.05 (0.00)	78.51±0.05 (0.00)	85.17±0.19 (0.00)	10.00
DMF-MVC (H. Zhao et al. 2017)	61.27±5.01 (0.00)	77.80±5.01 (0.00)	49.82±0.95 (0.00)	77.96±3.40 (0.00)	55.93±3.70 (0.00)	85.64±1.26 (0.00)	10.67
MvDGNMF (J. Li, G. Zhou, et al. 2020)	55.09±5.03 (0.00)	64.28±3.78 (0.00)	91.36±2.67 (0.00)	86.69±6.04 (0.00)	78.39±7.60 (0.00)	85.83±1.17 (0.00)	8.50
ODD-NMF (Luong, Nayak, et al. 2022)	68.00±4.80 (0.00)	63.88±2.80 (0.00)	40.58±3.26 (0.00)	79.08±3.91 (0.00)	67.47±1.29 (0.00)	83.43±6.10 (0.00)	12.17
DANMF-MRL (H. Huang, G. Zhou, Q. Zhao, et al. 2023)	54.06±2.38 (0.00)	56.28±1.58 (0.00)	87.48±5.90 (0.00)	85.85±5.00 (0.00)	73.46±5.74 (0.00)	84.68±0.03 (0.00)	11.00
OS-DANMF-MRL (H. Huang, G. Zhou, Q. Zhao, et al. 2023)	48.55±3.28 (0.00)	71.73±2.47 (0.00)	88.18±0.98 (0.00)	81.29±1.28 (0.00)	75.52±1.65 (0.00)	85.55±0.47 (0.00)	10.67
Orth-NTF (J. Li, Gao, et al. 2023)	<u>81.21</u>±0.00 (-)	77.50±0.00 (-)	81.00±0.00 (-)	86.03±0.00 (-)	<u>90.07</u>±0.00 (-)	90.37±0.00 (-)	5.33
S2MVTC (Long et al. 2024)	78.79±0.00 (-)	77.50±0.00 (-)	79.00±0.00 (-)	88.79±0.00 (-)	77.52±0.00 (-)	<u>91.52</u>±0.00 (-)	6.00
DSTL (D. Xu et al. 2024)	49.58±2.21 (0.00)	77.43±2.41 (0.00)	93.98±0.08 (0.00)	94.94±5.01 (0.00)	80.37±0.25 (0.00)	81.35±0.51 (0.00)	8.00
SGTL (M.-S. Chen, Z.-Y. Li, et al. 2024)	59.82±4.53 (0.00)	<u>87.35</u>±1.97 (0.23)	97.80±6.07 (0.00)	<u>98.68</u>±0.46 (0.00)	70.99±5.51 (0.00)	91.25±1.28 (0.04)	<u>5.00</u>
TPCH (Z. Wang et al. 2025)	50.91±0.00 (-)	84.25±0.00 (-)	91.80±0.00 (-)	85.11±0.00 (-)	77.66±0.00 (-)	89.89±0.00 (-)	7.33
LMTC (J. Liu, X. Liu, et al. 2025)	66.97±4.80 (0.00)	81.15±2.86 (0.00)	94.28±0.10 (0.00)	90.26±0.12 (0.00)	79.66±0.20 (0.00)	83.43±0.24 (0.00)	5.67
TADMVC (Y. Chen et al. 2025)	65.70±4.73 (0.00)	75.30±2.68 (0.00)	<u>97.94</u>±0.14 (0.02)	87.00±0.45 (0.00)	71.37±0.36 (0.00)	88.18±2.36 (0.09)	6.67
NAHTF(Ours)	**86.00**±2.89	**89.33**±2.17	**98.18**±0.06	**99.26**±0.05	**96.04**±0.27	**92.79**±0.11	**1.00**

average clustering results alongside stand derivations are presented in Tables 3-5, where the optimal and suboptimal results are highlighted in **bold** and <u>underlined</u>, respectively. The "Avg. Rank" signifies the average rank of each method across all datasets. The proposed NAHTF demonstrates optimal or near-optimal performance across all datasets and metrics. For example, on the BBC dataset, NAHTF exhibits improvements of 5.97% in ACC, 3.26% in NMI, and 5.97% in Purity compared to the suboptimal competitors. Moreover, NAHTF achieves near-perfect clustering on the 20Newsgroups, BBCSport, and BBC datasets, with ACC values approaching 100%. Although NAHTF ranks second on the Caltech101-7 dataset when measured by NMI, the NMI value is only 0.5% lower than that of the top performer. In addition, NAHTF achieves the highest Avg. Rank values, with reductions of 4.50, 3.0, and 4.0 for ACC, NMI, and Purity compared to the best baselines. These results collectively indicate that the proposed NAHTF significantly and consistently outperforms these baseline methods across diverse datasets.

Furthermore, a comprehensive analysis of the results reveals two primary insights:

- **Tensor-based Methods Superiority**: Tensor-based approaches, including Orth-NTF, S2MVTC, DSTL, SGTL, TPCH, and LMTC generally exhibit superiority over matrix factorization-based approaches. The Avg. Rank values obtained by these tensor-based methods are consistently higher than those of matrix factorization-based approaches across the evaluation metrics, with SGTL showing particularly notable



(a) ACC (b) NMI (c) Purity

Fig. 2. Clustering performance of NAHTF with different numbers of layers.

advantages. This advantage stems from the ability of low-rank tensors to capture rich complementary information.

- **Deep NMF Advantage**: Among matrix factorization approaches, deep NMF-based methods, especially MvDGNMF and OS-DANMF-MRL, generally outperform shallow NMF-based methods. For instance, MvDGNMF achieves significantly higher Avg. Rank values across evaluation metrics, attributed to its capacity to uncover hierarchical data attributes through layered decomposition.

Furthermore, to evaluate the statistical significance of the clustering improvements, a Student's t-test is conducted between NAHTF and baseline methods on each dataset at a significance level of 0.05. The resulting p-values reported in Tables 3-5 (in brackets) show $p < 0.05$ for nearly all comparisons, where "-" indicates that the t-test cannot be applied to deterministic methods (e.g., Orth-NTF, S2MVTC, and TPCH). These results confirm that the superior performance of NAHTF over state-of-the-art methods is statistically significant, thereby ruling out the possibility that it is an artifact of random initialization.

The above analysis confirms that deep tensor factorization can benefit from the deep architectures and low-rank tensor techniques to achieve overall superior performance. Consequently, the exceptional performance of NAHTF arises from four key advantages: (1) Adaptive tensor dictionary learning, which optimizes low-rank tensor representations by extracting high-quality components while eliminating adversarial noise; (2) Deep semi-nonnegative tensor factorization, which holistically propagates complementary information across all hierarchical decomposition layers; (3) Direct decomposition of the refined tensor representation, preserving high-order cross-view correlations inherent in multi-view data; and (4) Adaptive graph regularization, which dynamically captures the local consistency of data across views.

5.4 Parameter Analysis

Sensitivity analysis on the number of layers: The capacity of deep matrix methods for learning representations depends on the number of layers. To investigate the influence of the number of layers on the performance of the proposed method, numerical experiments are conducted by systematically varying the number of layers from 2 to 5. The corresponding layer sizes are set as $[5c, c]$, $[10c, 5c, c]$, $[15c, 10c, 5c, c]$, and $[20c, 15c, 10c, 5c, c]$. The results of NAHTF are presented in Figure 2, suggesting that NAHTF exhibits varying performance across these datasets with different numbers of layers. Evidently, the three-layer configuration shows optimal performance on the Yale, Orl, BBCSport, BBC, and Caltech101_7 datasets, whereas the five-layer configuration excels on the 20Newsgroup dataset. These results indicate that an increase in the number of layers does not necessarily ensure improved performance across all datasets. This is because a shallower configuration often leads to non-discriminative representations due to its limited depth, while a deeper configuration is prone to overfitting,

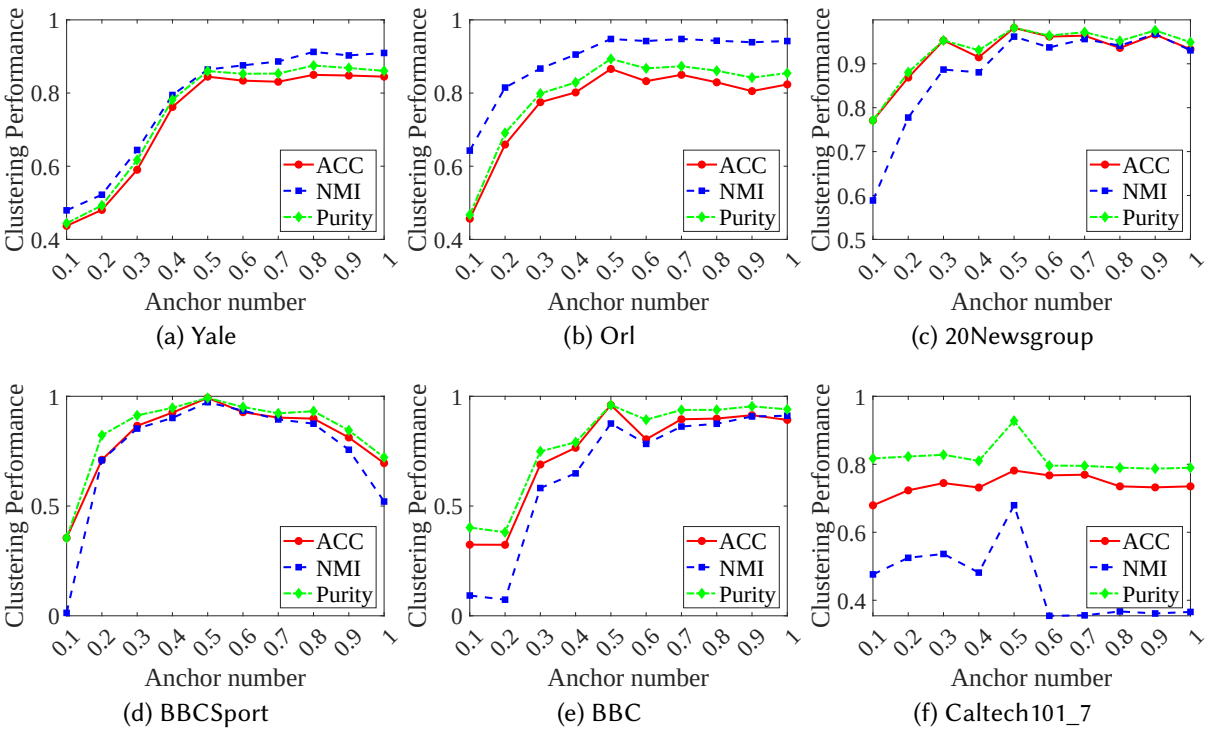


Fig. 3. Clustering performance of NAHTF with different numbers of anchors.

particularly on smaller datasets. Consequently, it is recommended to utilize fewer layers for small datasets and more layers for larger ones. Overall, a three-layer configuration strikes an effective balance, attaining significant performance while maintaining efficiency on benchmark datasets. Hence, the number of layers is set to 3 in this work.

Sensitivity analysis on the number of anchors: Figure 3 illustrates the performance of NAHTF with respect to the number of anchors across six datasets. The number of anchors is tuned from $0.1n$ to n (where n denotes the total data samples) in increments of $0.1n$. On the first five datasets (namely Yale, Orl, 20Newsgroup, BBCSport, and BBC), the proposed NAHTF exhibits consistent improvements with respect to all clustering metrics (i.e., ACC, NMI, and Purity) as the number of anchors increases from $0.1n$ to $0.5n$, followed by stable performance with further increases in the number of anchors. In contrast, on the Caltech101_7 dataset, NAHTF demonstrates anomalous behavior: its performance fluctuates irregularly across the range, with NMI and Purity showing unexpected drops when the number of anchors is increased from $0.5n$ to $0.6n$. This suggests potential challenges in handling highly heterogeneous multi-view features within this complex dataset. Moreover, NAHTF achieves optimal or near-optimal clustering performance on these benchmark datasets when the number of anchors is $0.5n$.

Sensitivity analysis on p: The parameter p in the Schatten p-norm controls the rank of the tensor representation $\mathcal{Z}$. Figure 4 depicts the performance of NAHTF on benchmark datasets as p is tuned from 0.1 to 1 with an interval of 0.1. The results demonstrate that NAHTF is relatively insensitive to p across most datasets, except

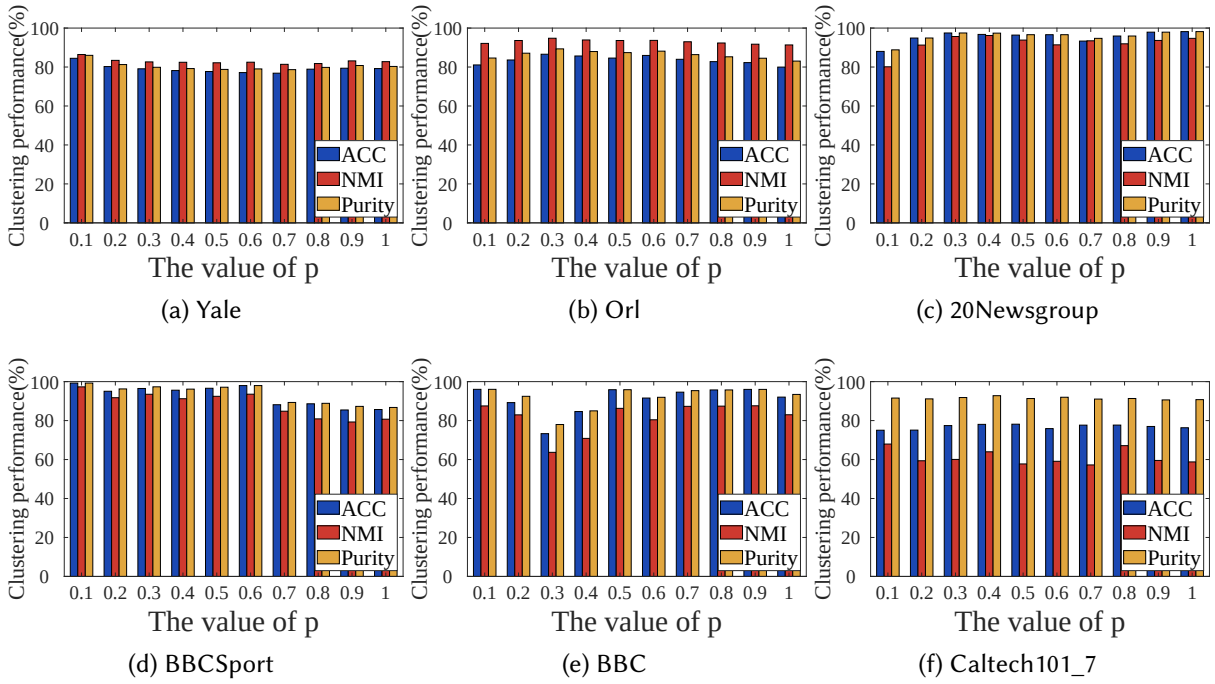


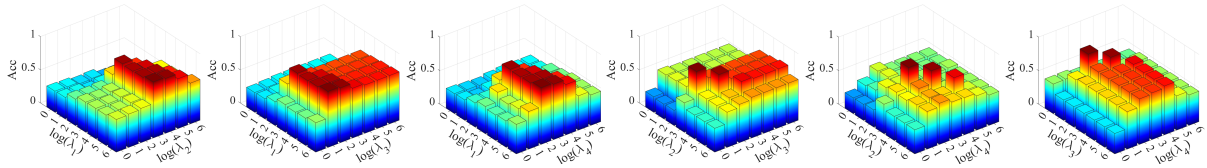
Fig. 4. Clustering performance of NAHTF with different values of p.

on the BBC dataset. Furthermore, it typically performs better when p is small, e.g., $p \leq 0.5$. In fact, the Schatten p-norm tends to yield a lower rank tensor when p is small.

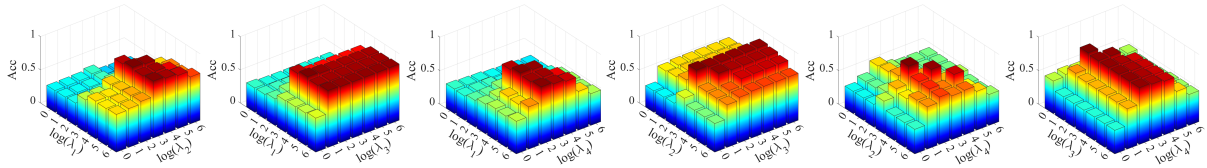
Sensitivity analysis on $\{\lambda_i\}_{i=1}^4$: These parameters act to balance the local consistency information exploration, Schatten p-norm-based high-order consistency exploration, tensor $l_{2,1}$-norm-based adverse component elimination, and dictionary learning respectively. The performance of NAHTF on these benchmark datasets is depicted in Figure 5 when these parameters are tuned in the range of $[10^0, 10^1, ..., 10^6]$ using a grid search strategy. It can be observed that NAHTF is sensitive to λ_1, λ_2, and λ_4, but slightly sensitive to λ_3. However, when λ_1 exceeds 10^2 or λ_2 is tuned from 10^2 to 10^4, NAHTF demonstrates more stable performance. Moreover, NAHTF performs better when λ_4 is less than 10^4. Accordingly, the following parameter ranges are recommended: $\lambda_1 \in [10^4, 10^6]$, $\lambda_2 \in [10^2, 10^4]$, $\lambda_3 \in [10^0, 10^2]$, and $\lambda_4 \in [10^0, 10^2]$. To ensure computational efficiency, a sequential search strategy is advised: initially anchoring $\lambda_3 = 10^2$ and $\lambda_4 = 10^1$, tuning λ_1 and λ_2 within their respective ranges, and subsequently refining λ_3 and λ_4 if necessary. This strategy greatly reduces the combinatorial search space while maintaining excellent clustering performance. The empirically optimal hyperparameter configurations across the benchmarks are detailed in Table 6.

5.5 Visualization Analysis

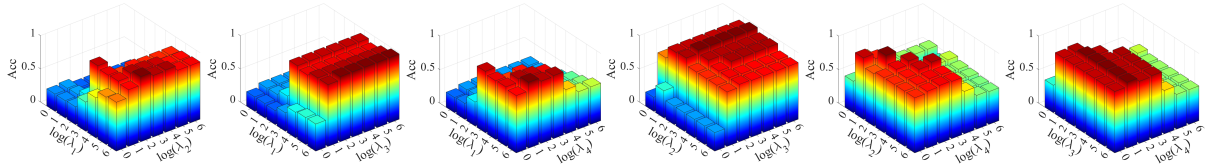
To visually demonstrate the effectiveness of the proposed hierarchical tensor factorization framework, t-SNE visualization is applied to the learned representations from the BBCSport dataset. The visualizations in Figure 6 yield the following observations. Although low-rank tensor techniques possess the advantages of exploring high-order consistency in multi-view data, representations from tensor-based baselines exhibit substantial intra-class



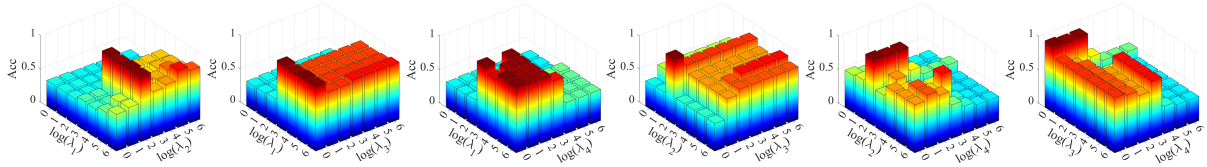
(a) Yale



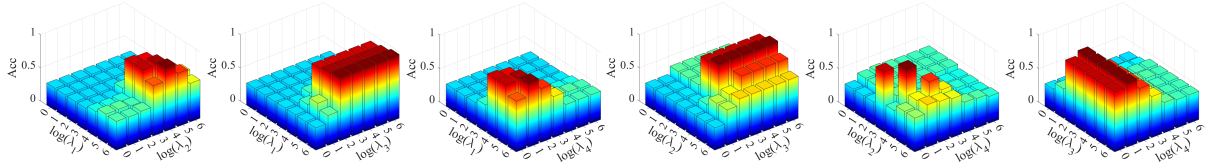
(b) Orl



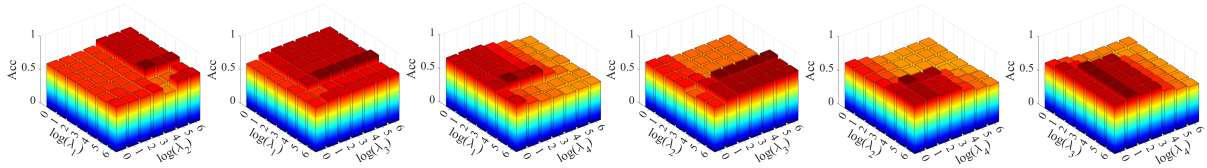
(c) 20Newsgroup



(d) BBCSport



(e) BBC



(f) Caltech101_7

Fig. 5. Clustering performance of the NAHTF, in terms of ACC, with different trade-off parameters $\{\lambda_i\}_{i=1}^4$.

Table 6. Hyperparameter configurations used in our experiments.

Dataset	λ_1	λ_2	λ_3	λ_4	p	ACC	NMI	Purity
Yale	10^6	10^3	10^1	10^2	0.1	84.48	86.42	86.00
Orl	10^4	10^3	10^1	10^2	0.3	86.58	94.78	89.33
20Newsgroups	10^6	10^2	10^2	10^1	0.3	98.18	96.19	98.18
BBCSport	10^5	10^2	10^0	10^0	0.1	99.26	97.31	99.26
BBC	10^6	10^4	10^2	10^1	0.1	96.04	87.59	96.04
Caltech101_7	10^4	10^4	10^2	10^1	0.4	78.15	67.96	92.79

dispersion and inter-class overlap, indicating that low-rank properties alone are insufficient for capturing intricate multi-view semantics. Specifically, Orth-NTF (Fig. 6a) fails to separate most classes. While SGTL (Fig. 6c) shows improved performance, it still suffers from pronounced inter-class overlap and dispersed intra-class samples. Conversely, our NAHTF (Fig. 6h) achieves clear separation of all classes, with optimal intra-class cohesion and minimal inter-class overlap. These results confirm that the proposed cross-layer complementary fusion mechanism yields more discriminative representations than shallow tensor decomposition approaches. Furthermore, the impact of decomposition depth on the representation learning capability of NAHTF is investigated. Employing a single layer (Fig. 6f) results in persistent inter-class overlap. Increasing the depth to two layers (Fig. 6g) enhances intra-class compactness and inter-class separability, although marginal overlap persists among adjacent classes. With the full three-layer hierarchical decomposition (Fig. 6h), NAHTF achieves complete separation of all classes and maximum compactness within each class. This progressive improvement demonstrates that the proposed deep tensor factorization module effectively captures hierarchical semantic features from multi-view data.

5.6 Ablation Experiments

The proposed NAHTF comprises three modules: deep semi-nonnegative tensor factorization, noise-aware tensor dictionary learning, and local consistency exploration. The tensor dictionary learning module can be further decomposed into the dictionary learning sub-module (i.e., $\sum_{v=1}^{V} \|X^v - A^v C^v\|_F^2$) and the tensor representation extraction sub-module (i.e., $\mathscr{C} = \mathscr{Z} + \mathscr{E}$). In this subsection, ablation experiments are conducted to analyze the contributions of these modules to the proposed model. Figure 7 presents the ablation experiment results concerning these modules on benchmark datasets. Specifically, in this figure, "Basic" represents the model containing only the deep semi-nonnegative tensor factorization module, "M1" represents the model excluding the tensor dictionary learning module, "M2" signifies the model excluding the local consistency exploration module, "M3" denotes the model excluding the dictionary learning sub-module but containing the tensor representation extraction sub-module, and "All" denotes the complete model NAHTF. Since "Basic", "M1", and "M3" require a tensor as input, dictionary learning is initially performed to derive view-specific representations $\{C^v\}_{v=1}^V$ for constructing this tensor. The results demonstrate the effectiveness of these modules in improving clustering performance. Through the integration of these modules into the joint framework, NAHTF achieves significant improvements in clustering performance.

5.7 Convergence Analysis

To optimize Eq. (10), NAHTF introduces three auxiliary variables $\mathscr{J}_1$, $\mathscr{J}_2$, and $\mathscr{G}$ and optimizes the corresponding augmented Lagrangian function Eq. (11). To determine the convergence of the optimization algorithm presented

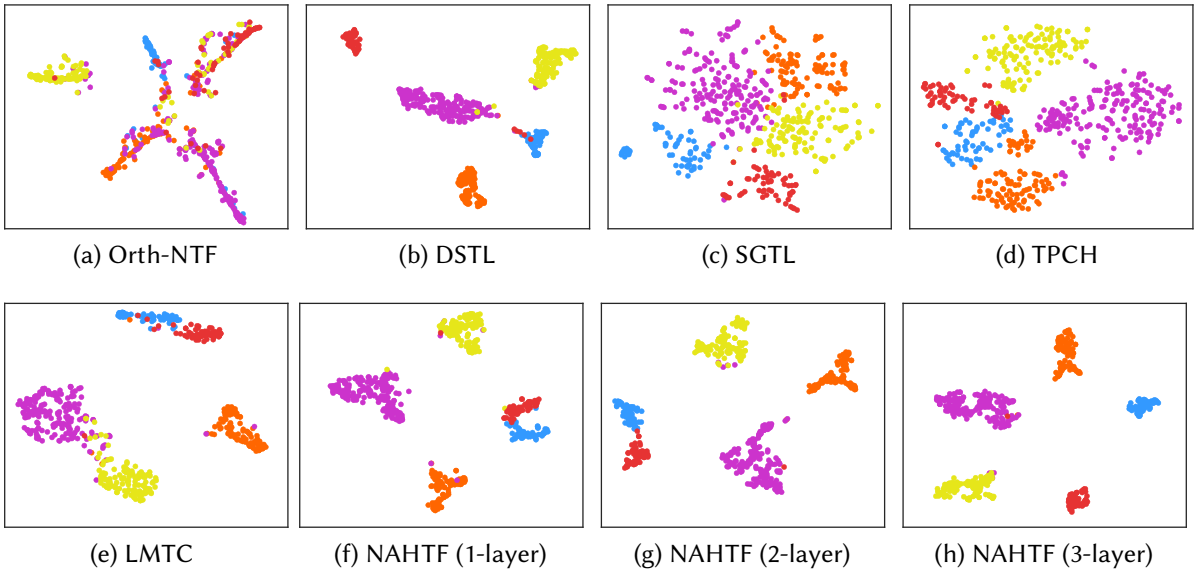


Fig. 6. t-SNE visualization of representation distributions on the BBCSport dataset.

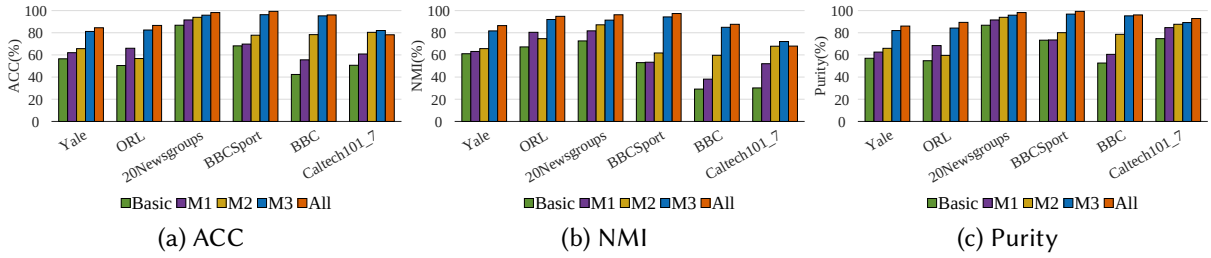


Fig. 7. Ablation experiment of NAHTF on benchmark datasets.

in Algorithm 1, the KKT conditions $\sum_{v=1}^{V} \|\mathcal{Z}^v - \mathcal{G}^v\|_\infty \leq \varepsilon$, $\sum_{v=1}^{V} \|\mathcal{H}_m^v - \mathcal{J}_1^v\|_\infty \leq \varepsilon$, and $\sum_{v=1}^{V} \|\mathcal{H}_m^v - \mathcal{J}_2^v\|_\infty \leq \varepsilon$ are employed, where ε is set to 10^{-13}. As demonstrated in Figure 8, Algorithm 1 achieves rapid convergence, i.e., all matching errors experience a significant decline within the first 30 iterations, followed by gradual stabilization until they converge to zero. These results empirically validate the convergence of the proposed optimization algorithm.

5.8 Running Time Analysis

To complement the theoretical complexity analysis of $\mathcal{O}(n^3T)$, Table 7 provides an empirical comparison of running time and theoretical complexity between our method and other methods. The results reveal the following observations. First, the empirical running time of the proposed NAHTF aligns with its theoretical cubic complexity. NAHTF requires more computational time than most linear or near-linear complexity methods, including Orth-NTF ($\mathcal{O}(nT)$), S2MVTC ($\mathcal{O}(nT)$), DSTL ($\mathcal{O}(n\log(n)T)$) and LMTC ($\mathcal{O}(nT)$). These four methods

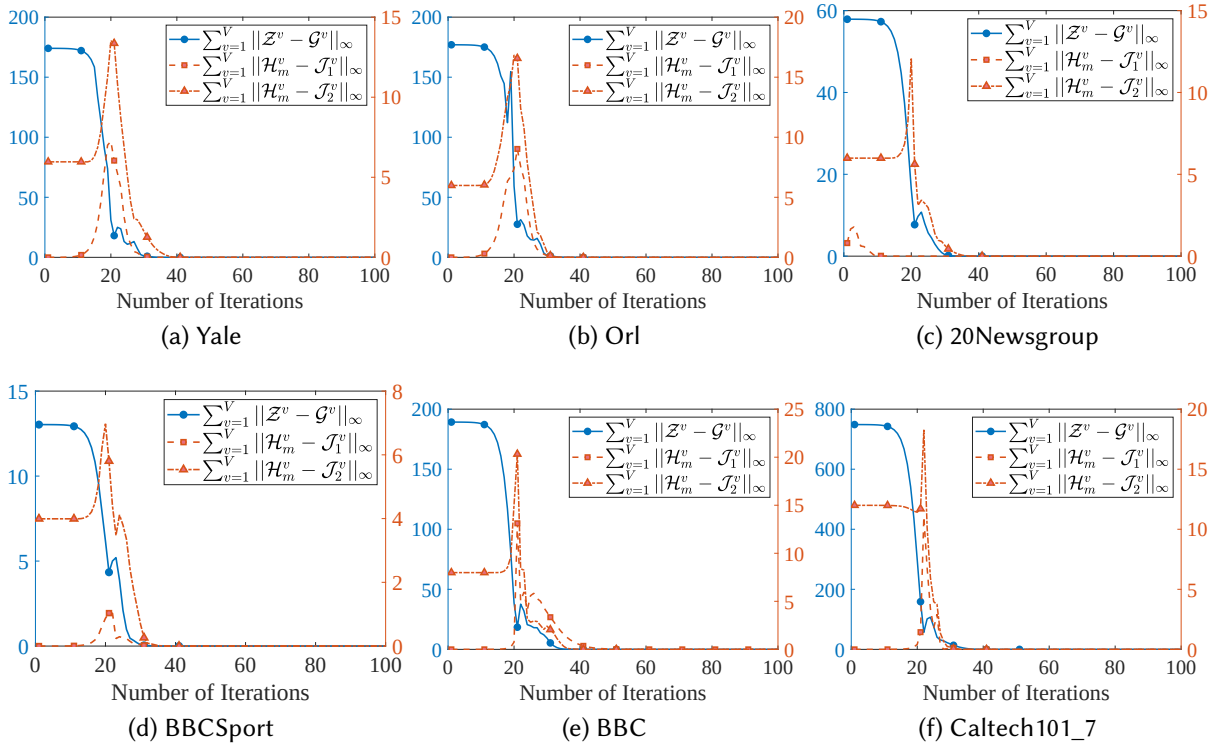


Fig. 8. Convergence behaviour of NAHTF.

run within sub-second ranges in most cases across all datasets. This is expected, as NAHTF incorporates deep semi-nonnegative tensor factorization and iterative tensor dictionary learning, requiring heavier computational resources to capture high-order correlations and hierarchical semantics. Second, and more importantly, NAHTF demonstrates significant efficiency advantages over methods of similar theoretical complexity, as NAHTF converges stably within 40 iterations on all benchmark datasets (as shown in Fig. 8). Specifically, compared to SGTL, which shares the same theoretical complexity of $\mathcal{O}(n^3T)$, NAHTF is substantially faster, particularly on larger datasets. For instance, on the BBC dataset, NAHTF takes 5.28 seconds versus 10.37 seconds for SGTL. This gap widens on the largest dataset, Caltech101-7, where NAHTF (32.70s) is approximately 2.7 times faster than SGTL (119.88s).

In summary, the empirical results confirm that the computational cost of NAHTF is justified, as it executes significantly faster than cubic-complexity competitors like SGTL while delivering substantially superior clustering performance (as demonstrated in Tables 3-5).

6 Applicability Analysis and Future Work

This section discusses the applicability and limitations of the proposed NAHTF framework and outlines several promising directions for future research.

Table 7. Running time (in seconds) comparison with other methods

Method	Yale	Orl	20Newsgroups	BBCSport	BBC	Caltech101_7	Time Complexity
MultiNMF	1.53	10.66	0.78	0.77	4.09	1.92	$\mathcal{O}(n^2T)$
2CMV	3.10	6.54	2.84	2.71	8.78	7.78	$\mathcal{O}(n^2T)$
DMF-MVC	5.33	7.08	3.82	4.02	11.93	10.23	$\mathcal{O}(n^2T)$
MvDGNMF	3.54	5.90	3.09	3.49	11.63	6.15	$\mathcal{O}(n^2T)$
ODD-NMF	3.43	5.48	3.08	3.12	9.62	10.53	$\mathcal{O}(n^2T)$
DANMF-MRL	3.50	4.05	1.23	1.48	5.88	1.75	$\mathcal{O}(n^2T)$
OS-DANMF-MRL	3.47	3.86	1.20	1.47	5.39	1.35	$\mathcal{O}(n^2T)$
Orth-NTF	**0.02**	**0.05**	**0.02**	**0.02**	**0.04**	**0.04**	$\mathcal{O}(nT)$
S2MVTC	0.09	0.23	0.20	0.17	0.31	0.72	$\mathcal{O}(nT)$
DSTL	**0.02**	**0.05**	**0.02**	**0.02**	0.05	**0.04**	$\mathcal{O}(nlog(n)T)$
SGTL	1.01	2.98	4.17	3.39	10.37	119.88	$\mathcal{O}(n^3T)$
TPCH	0.21	0.48	0.71	0.65	1.86	10.99	$\mathcal{O}(nlog(n)T)$
LMTC	0.53	0.82	0.11	0.12	0.55	0.37	$\mathcal{O}(nT)$
TADMVC	3.99	9.66	5.05	5.53	8.84	13.52	$\mathcal{O}(nT)$
NAHTF (Ours)	0.52	2.69	2.01	1.96	5.28	32.70	$\mathcal{O}(n^3T)$

6.1 Applicability Analysis

Given its structural advantages, the NAHTF framework is particularly applicable to: (1) Multi-view data with inherent hierarchical semantics (e.g., facial images and textual documents), where the deep tensor factorization module can hierarchically decompose tensor representations to capture low-level specific features in lower layers and abstract semantic features in higher layers; and (2) Multi-view data contaminated by sensor noise. Conversely, NAHTF is suboptimal for: (1) Single-view data, because its fundamental mechanism requires constructing a third-order tensor by stacking multi-view data, a process infeasible for single-view inputs, to leverage the tensor t-product for cross-view and cross-layer fusion; and (2) Large-scale data processing, due to its theoretical complexity of $\mathcal{O}(n^3)$, which incurs prohibitive computational overhead.

6.2 Future Work

Although the proposed NAHTF framework demonstrates superior clustering performance, several several avenues remain to enhance its applicability. First, extending NAHTF to address incomplete multi-view clustering (IMVC) is critical. The current methodology presupposes the completeness of data views. However, real-world multi-view data often exhibit missing instances, preventing the construction of third-order tensors for subsequent factorization. Inferring these missing instances by exploring underlying properties and cross-view correlations of multi-view data offers a promising solution. Consequently, integrating tensor completion techniques with a low-rank tensor recovery module within NAHTF is anticipated. This module recovers missing view information by leveraging the hierarchical semantic features of complete views and the global low-rank characteristics of multi-view data, thereby guiding the representation learning of the missing views. Second, the present approach employs a pre-selection strategy with a fixed number of anchors. To address this limitation, adaptive anchor selection techniques should be incorporated into the joint optimization framework. Specifically, discriminative anchors can be adaptively derived based on the underlying data characteristics of each view. Third, generalizing NAHTF for continuous (incremental) multi-view clustering on streaming data is imperative. The existing method targets static batch data and cannot accommodate continuously arriving views.

Retraining the entire model upon new arrivals incurs substantial computational overhead. To mitigate this issue, the current view representation can be further decoupled into semantically consistent and view-specific components. Upon new arrivals, knowledge transfer is effected by reusing the semantically consistent representation, thereby avoiding complete model retraining.

7 Conclusion

This paper presents the Noise-Aware Hierarchical Tensor Fusion (NAHTF) framework for multi-view clustering, addressing three critical limitations inherent in existing methods: namely, shallow representation learning, high-order correlation isolation, and noise susceptibility. The proposed model integrates tensor dictionary learning with deep semi-nonnegative tensor decomposition, leveraging the tensor t-product to achieve cross-view and cross-layer fusion of multi-view data. This approach enables the capture of high-order cross-view correlations and the propagation of complementary information across all hierarchical decomposition layers. This is a significant advancement over traditional methods, which typically concentrate on pairwise view correlations and the top-layer representation fusion. Additionally, the noise-aware tensor dictionary learning, which combines tensor Schatten p-norm regularization for low-rank tensor recovery and tensor $l_{2,1}$-norm regularization for sparse noise modeling, enhances the robustness of the tensor representation against data noise and feature redundancy. Experiments on various real-world datasets illustrate the effectiveness of the proposed model.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (No. 62466035), the Natural Science Foundation of Jiangxi Province, China (No. 20242BAB25106), and Jiangxi Provincial Key Laboratory of Virtual Reality (No. 2024SSY03151).

References

M.-S. Chen, P.-Y. Lai, D.-Z. Liao, C.-D. Wang, and J.-H. Lai. 2025. "Graph Prompt Clustering." *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

M.-S. Chen, Z.-Y. Li, J.-Q. Lin, C.-D. Wang, and D. Huang. 2024. "Sparse Graph Tensor Learning for Multi-View Spectral Clustering." *IEEE Transactions on Emerging Topics in Computational Intelligence*, 8, 5, 3534–3543.

Y. Chen, H. Wang, M. Yao, J. Peng, G. Jiang, and J. Zhang. 2025. "Scalable Multi-view Clustering based on Tight Anchor Distribution." In: *ACM MM*, 7739–7747.

Z. Chen, P. Lin, Z. Chen, D. Ye, and S. Wang. 2022. "Diversity embedding deep matrix factorization for multi-view clustering." *Information Sciences*, 610, 114–125.

B. Cui, H. Yu, T. Zhang, and S. Li. 2019. "Self-weighted multi-view clustering with deep matrix factorization." In: *ACML*, 567–582.

P. De Handschutter, N. Gillis, and X. Siebert. 2021. "A survey on deep matrix factorizations." *Computer Science Review*, 42, 100423.

C. H. Ding, T. Li, and M. I. Jordan. 2008. "Convex and semi-nonnegative matrix factorizations." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32, 1, 45–55.

Q. Gao, P. Zhang, W. Xia, D. Xie, X. Gao, and D. Tao. 2020. "Enhanced tensor RPCA and its application." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43, 6, 2133–2140.

H. Huang, G. Zhou, N. Liang, Q. Zhao, and S. Xie. 2023. "Diverse deep matrix factorization with hypergraph regularization for multiview data representation." *IEEE/CAA Journal of Automatica Sinica*, 10, 11, 2154–2167.

H. Huang, G. Zhou, Q. Zhao, L. He, and S. Xie. 2023. "Comprehensive multiview representation learning via deep autoencoder-like nonnegative matrix factorization." *IEEE Transactions on Neural Networks and Learning Systems*, 35, 5, 5953–5967.

S. Huang, Z. Kang, and Z. Xu. 2020. "Auto-weighted multi-view clustering via deep matrix decomposition." *Pattern Recognition*, 97, 107015.

S. Huang, Z. Xu, and F. Wang. 2017. "Nonnegative matrix factorization with adaptive neighbors." In: *IJCNN*, 486–493.

Z. Kong, Z. Fu, D. Chang, Y. Wang, and Y. Zhao. 2025. "Dual-space Co-training for Large-scale Multi-view Clustering." *Pattern Recognition*, 168, 111844.

D. Lee and H. S. Seung. 2000. "Algorithms for non-negative matrix factorization." In: *NIPS*, 556–562.

J. Li, G. Zhou, Y. Qiu, Y. Wang, Y. Zhang, and S. Xie. 2020. "Deep graph regularized non-negative matrix factorization for multi-view clustering." *Neurocomputing*, 390, 108–116.

J. Li, Q. Gao, Q. Wang, M. Yang, and W. Xia. 2023. "Orthogonal Non-negative Tensor Factorization based Multi-view Clustering." In: *NIPS*, 18186–18202.

W. Liang, C. Tang, X. Liu, Y. Liu, J. Liu, E. Zhu, and K. He. 2024. "On the consistency and large-scale extension of multiple kernel clustering." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46, 10, 6935–6947.

W. Liang, S. Zhou, J. Xiong, X. Liu, S. Wang, E. Zhu, Z. Cai, and X. Xu. 2020. "Multi-view spectral clustering with high-order optimal neighborhood laplacian matrix." *IEEE Transactions on Knowledge and Data Engineering*, 34, 7, 3418–3430.

W. Liang, E. Zhu, S. Yu, H. Xu, X. Zhu, and X. Liu. 2024. "Scalable multiple kernel clustering: Learning clustering structure from expectation." In: *ICML*.

J.-Q. Lin, X.-L. Li, M.-S. Chen, C.-D. Wang, and H. Zhang. 2022. "Incomplete data meets uncoupled case: A challenging task of multiview clustering." *IEEE Transactions on Neural Networks and Learning Systems*, 35, 6, 8097–8110.

J. Liu, C. Wang, J. Gao, and J. Han. 2013. "Multi-view clustering via joint nonnegative matrix factorization." In: *ICDM*, 252–260.

J. Liu, X. Liu, et al.. 2025. "Large-scale Multi-view Tensor Clustering with Implicit Linear Kernels." In: *CVPR*, 20727–20736.

Z. Long, Q. Wang, Y. Ren, Y. Liu, and C. Zhu. 2024. "S2MVTC: a Simple yet Efficient Scalable Multi-View Tensor Clustering." In: *CVPR*, 26213–26222.

C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan. 2019. "Tensor robust principal component analysis with a new tensor nuclear norm." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42, 4, 925–938.

K. Luong and R. Nayak. 2020. "A novel approach to learning consensus and complementary information for multi-view data clustering." In: *ICDE*, 865–876.

K. Luong, R. Nayak, T. Balasubramaniam, and M. A. Bashar. 2022. "Multi-layer manifold learning for deep non-negative matrix factorization-based multi-view clustering." *Pattern Recognition*, 131, 108815.

W. Min, T. Xu, X. Wan, and T.-H. Chang. 2022. "Structured Sparse Non-negative Matrix Factorization with L2, 1-Norm." *IEEE Transactions on Knowledge and Data Engineering*, 35, 8, 8584–8595.

R. Nishihara, L. Lessard, B. Recht, A. Packard, and M. Jordan. 2015. "A general analysis of the convergence of ADMM." In: *ICML*, 343–352.

K. K. Sharma and A. Seal. 2021. "Multi-view spectral clustering for uncertain objects." *Information Sciences*, 547, 723–745.

K. K. Sharma, A. Seal, A. Yazidi, and O. Krejcar. 2022. "A new adaptive mixture distance-based improved density peaks clustering for gearbox fault diagnosis." *IEEE Transactions on Instrumentation and Measurement*, 71, 1–16.

W. Tang, B. Hui, L. Tian, G. Luo, Z. He, and Z. Cai. 2021. "Learning disentangled user representation with multi-view information fusion on social networks." *Information Fusion*, 74, 77–86.

G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. Schuller. 2014. "A deep semi-nmf model for learning hidden representations." In: *ICML*, 1692–1700.

S. Wang, L. Chen, N. Zheng, L. Li, F. Peng, and J. Lu. 2023. "Shared and individual representation learning with Feature Diversity for Deep MultiView Clustering." *Information Sciences*, 647, 119426.

Z. Wang, X. Li, Y. Sun, Q. Sun, Y. Sun, H. Ling, J. Dai, and Z. Ren. 2025. "TPCH: tensor-interacted projection and cooperative hashing for multi-view clustering." In: *AAAI*, 21420–21428.

J.-S. Wu, J.-T. Yu, J.-Y. Wu, W. Min, and W.-S. Zheng. 2025. "High-order Aligned Deep Complementary and View-Specific Similarity Graphs for Unsupervised Multi-View Feature Selection." *Pattern Recognition*, 171, 112047.

J. Wu, Z. Lin, and H. Zha. 2019. "Essential tensor learning for multi-view spectral clustering." *IEEE Transactions on Image Processing*, 28, 12, 5910–5922.

W. Wu, X. Ma, Q. Wang, M. Gong, and Q. Gao. 2024. "Learning deep representation and discriminative features for clustering of multi-layer networks." *Neural Networks*, 170, 405–416.

W. Xia, Q. Gao, Q. Wang, X. Gao, C. Ding, and D. Tao. 2022. "Tensorized bipartite graph learning for multi-view clustering." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45, 4, 5187–5202.

D. Xu, C. Zhang, Z. Li, C. Chen, and H. Li. 2024. "Fast disentangled slim tensor learning for multi-view clustering." *IEEE Transactions on Multimedia*, 27, 1254–1265.

Z. Xue, J. Du, D. Du, and S. Lyu. 2019. "Deep low-rank subspace ensemble for multi-view clustering." *Information Sciences*, 482, 210–227.

B. Yang, X. Zhang, F. Nie, F. Wang, W. Yu, and R. Wang. 2020. "Fast multi-view clustering via nonnegative and orthogonal factorization." *IEEE Transactions on Image Processing*, 30, 2575–2586.

H. Yang, Q. Gao, W. Xia, M. Yang, and X. Gao. 2022. "Multiview spectral clustering with bipartite graph." *IEEE Transactions on Image Processing*, 31, 3591–3605.

F. Ye, C. Chen, and Z. Zheng. 2018. "Deep autoencoder-like nonnegative matrix factorization for community detection." In: *ICKM*, 1393–1402.

B. Yoshua. 2009. "Learning deep architectures for AI." *Foundations and Trends in Machine Learning*, 2, 1–127.

C. Zhang, S. Wang, J. Liu, S. Zhou, P. Zhang, X. Liu, E. Zhu, and C. Zhang. 2021. "Multi-view Clustering via Deep Matrix Factorization and Partition Alignment." In: *ACM MM*, 4156–4164.

G.-Y. Zhang, D. Huang, and C.-D. Wang. 2025. "Large-Scale Tensorized Multi-View Kernel Subspace Clustering." *ACM Transactions on Intelligent Systems and Technology*, 16, 4, 1–21.

G.-Y. Zhang, D. Huang, and C.-D. Wang. 2024. "Tensorized incomplete multi-view kernel subspace clustering." *Neural Networks*, 179, 106529.
G.-Y. Zhang, Y.-D. Huang, D. Huang, C.-D. Wang, Y. Liu, and E. Huang. 2025. "Structure-preserving contrastive graph clustering with dual-channel label alignment." *Neural Networks*, 107954.
H. Zhao, Z. Ding, and Y. Fu. 2017. "Multi-view clustering via deep matrix factorization." In: *AAAI*, 2921–2927.
L. Zhao, Z. Wang, Z. Wang, and Z. Chen. 2023. "Multi-View Graph Regularized Deep Autoencoder-Like NMF Framework." In: *ICASSP*, 1–5.
W. Zuo, D. Meng, L. Zhang, X. Feng, and D. Zhang. 2013. "A generalized iterated shrinkage algorithm for non-convex sparse coding." In: *ICCV*, 217–224.

Received 15 May 2026; accepted 22 October 2026