LOW-RANK SMOOTH UNIFIED ANCHOR TENSOR NONNEGATIVE ORTHOGONAL FACTORIZATION FOR MULTI-VIEW CLUSTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Anchor-based tensor learning provides an efficient framework for large-scale multi-view clustering by combining compact anchor representations with high-order cross-view modeling. However, existing methods often rely on predefined anchors, separating anchor construction from subsequent tensor learning and preventing their joint optimization. Moreover, independently learned anchors in heterogeneous feature spaces may lack explicit cross-view correspondence, while conventional low-rank tensor regularization tends to overlook local smoothness among samples. To address these limitations, we propose a **L**ow-Rank **S**mooth **U**nified **A**nchor **T**ensor **N**onnegative **O**rthogonal **F**actorization (**LS-UATNOF**) model for multi-view clustering. Specifically, heterogeneous views are projected into a common feature space, where a unified anchor matrix is adaptively learned and shared across all views. Unlike two-stage schemes, the unified anchors and tensor representations are jointly optimized, allowing the high-order structure revealed by tensor factorization to feed back into anchor learning. Tensor total variation Schatten-p regularization is further incorporated to capture global low-rankness and sample-wise local smoothness within a single term. An efficient ADMM-based optimization algorithm is developed. Extensive experiments on eight benchmark datasets demonstrate that **LS-UATNOF** achieves competitive or superior clustering performance with favorable convergence and computational efficiency.

1 INTRODUCTION

Multi-view data are ubiquitous in real-world applications, where the same objects are described by heterogeneous features from different sources or feature extractors Fang et al. (2023); Sun (2013); Tang et al. (2023). Multi-view clustering (MVC) aims to exploit the consistency and complementarity among these views to reveal the underlying sample structure Wang et al. (2019a;b). Early MVC methods are largely based on self-representation, where view-specific $n \times n$ representation matrices are learned to characterize pairwise relationships among all samples Gao et al. (2015); Zhang et al. (2017; 2020); Liu et al. (2024a). Despite their effectiveness, such formulations incur rapidly increasing computational and memory costs as the number of samples grows. Anchor-based MVC provides a scalable alternative by representing each sample through a small set of representative anchors, reducing the representation from $n \times n$ to $m \times n$ with $m \ll n$ Liu et al. (2024b); Zhang et al. (2024).

To further capture high-order correlations across views, recent studies stack view-specific anchor representations into a third-order tensor and exploit tensor low-rankness or factorization Li et al. (2022); Liu et al. (2024b); Song et al. (2026); Dong et al. (2026). Representative methods such as Orth-NTF Li et al. (2023a) demonstrate the effectiveness of this paradigm. Nevertheless, their anchor graphs are commonly constructed before tensor learning and then kept fixed Wu et al. (2019); Lv et al. (2023); consequently, the high-order structure discovered during tensor optimization cannot refine the anchors. This one-way pipeline motivates a jointly coupled formulation rather than a sequential combination of anchor construction and tensor learning.

Adaptive anchor learning provides a natural remedy Ji & Feng (2025); Zhang et al. (2025), but independently learned anchors in heterogeneous feature spaces lack explicit index correspondence.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

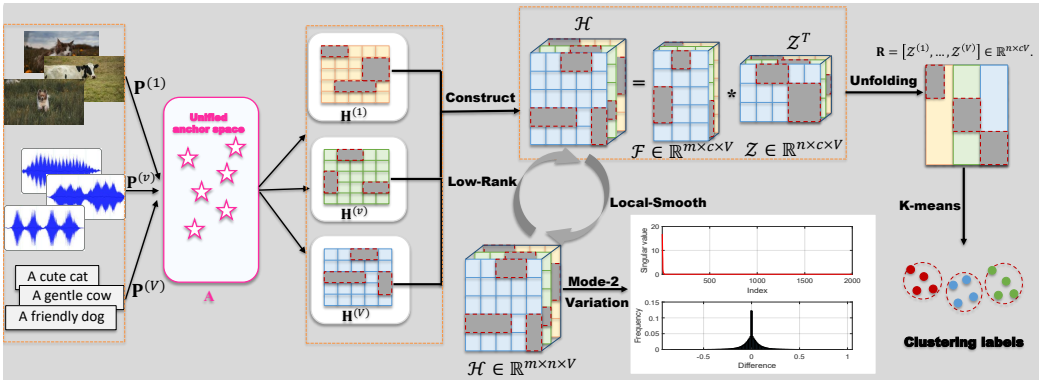


Figure 1: Overall framework of the proposed **LS-UATNOF**. Multi-view data are projected into a common latent space to learn unified anchor representations, which are regularized and factorized to produce the final representation for k-means clustering.

Consensus-anchor methods improve cross-view agreement Liu et al. (2022), yet they do not explicitly couple a shared adaptive anchor dictionary with nonnegative orthogonal tensor factorization. Meanwhile, existing tensor-based MVC methods mainly capture global low-rank correlations Wu et al. (2020); Wang et al. (2018); local sample smoothness is either overlooked or modeled by a separate penalty Du & Lu (2025). These gaps call for a framework that jointly learns semantically aligned anchors and tensor representations while integrating low-rankness and smoothness.

To address these issues, we propose a *Low-Rank Smooth Unified Anchor Tensor Nonnegative Orthogonal Factorization* (**LS-UATNOF**) model. Heterogeneous views are projected into a common latent space, where a shared anchor matrix and the view-specific anchor representations are learned together with the tensor factors. Because the anchor representations participate simultaneously in view reconstruction and tensor modeling, high-order cross-view information can feed back into the anchor-learning process instead of being used only after anchor construction. We further incorporate tensor total variation Schatten-p regularization to characterize global low-rank correlations and sample-wise local smoothness. The overall framework is illustrated in Fig. 1.

The main contributions of this work are summarized as follows:

- We propose a jointly coupled unified-anchor tensor learning framework. A shared anchor matrix is adaptively learned in a common latent space, while the learned tensor structure is fed back to refine the anchors and their sample representations, establishing explicit cross-view correspondence.
- We adapt tensor total variation Schatten-p regularization to the aligned anchor representation tensor, jointly capturing global low-rankness and sample-wise local smoothness without introducing a separate smoothness penalty or an additional trade-off parameter.
- We develop an efficient ADMM-based optimization algorithm with closed-form or efficiently computable updates. Extensive experiments on representative benchmark datasets demonstrate the effectiveness and efficiency of **LS-UATNOF**.

2 Preliminaries

For clarity, we summarize only the tensor tools that are essential for the proposed model. Additional notation and standard tensor definitions are provided in Appendix A.

Throughout this paper, scalars, vectors, matrices, and tensors are denoted by lowercase letters, bold lowercase letters, bold uppercase letters, and calligraphic letters, respectively. For a third-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathcal{X}^{(i)} = \mathcal{X}(:,:,i)$ denotes its i-th frontal slice. The discrete Fourier transform along the third mode is denoted by $\overline{\mathcal{X}} = \mathrm{fft}(\mathcal{X}, [], 3)$.

Definition 1 (Tensor Schatten-p norm Gao et al. (2020)). *For a third-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $0 < p \leq 1$, its tensor Schatten-p norm is defined as*

$$\|\mathcal{X}\|_{\circledS_p}^p = \sum_{i=1}^{n_3} \sum_{j=1}^{\min(n_1,n_2)} \sigma_j^p \left(\overline{\mathcal{X}}^{(i)}\right),$$

where $\overline{\mathcal{X}} = \mathrm{fft}(\mathcal{X}, [], 3)$ is the discrete Fourier transform of $\mathcal{X}$ along the third mode, and $\sigma_j(\overline{\mathcal{X}}^{(i)})$ denotes the j-th singular value of the i-th frontal slice in the Fourier domain. When $p = 1$, it reduces to the corresponding tensor nuclear norm Kilmer & Martin (2011).

3 Related Work

We review three lines of research closely related to our work: multi-view subspace clustering, anchor-based multi-view clustering, and anchor graph tensor factorization.

3.1 Multi-view Subspace Clustering

Multi-view subspace clustering (MVSC) aims to exploit the complementary and consistent information across multiple views to uncover the intrinsic clustering structure Gao et al. (2015); Luo et al. (2018). Most MVSC methods are built on the self-representation property, where each sample is expressed as a linear combination of other samples from the same underlying subspace Yan et al. (2026). For the v-th view, a typical formulation is $\mathbf{X}^{(v)} = \mathbf{X}^{(v)}\mathbf{Z}^{(v)} + \mathbf{E}^{(v)}$, where $\mathbf{Z}^{(v)} \in \mathbb{R}^{n \times n}$ characterizes pairwise sample relationships. By jointly regularizing the view-specific self-representation matrices, MVSC methods can capture shared and complementary structures across views.

Various extensions have incorporated latent representations Zhang et al. (2017) and low-rank tensor constraints Gao et al. (2020) to further exploit cross-view correlations. However, conventional MVSC still requires an $n \times n$ self-representation matrix for each view, which leads to high computational and memory costs and limits scalability to large datasets.

3.2 Anchor-based Multi-view Clustering

Anchor-based multi-view clustering improves scalability by replacing sample-to-sample relationships with sample-to-anchor relationships Liu et al. (2024b); Guo & Ye (2019). Given m anchors with $m \ll n$, each view is represented by an anchor representation matrix $\mathbf{H}^{(v)} \in \mathbb{R}^{m \times n}$, reducing the representation size from $n \times n$ to $m \times n$. A typical anchor-based model can be written as $\mathbf{X}^{(v)} \approx \mathbf{A}^{(v)}\mathbf{H}^{(v)}$, where $\mathbf{A}^{(v)}$ denotes the anchor matrix and $\mathbf{H}^{(v)}$ encodes sample-anchor relationships. This formulation substantially reduces computational and storage costs while preserving the essential structure of the data.

To exploit high-order correlations across views, recent methods stack $\{\mathbf{H}^{(v)}\}_{v=1}^{V}$ into a third-order tensor $\mathcal{H} \in \mathbb{R}^{m \times n \times V}$ and impose tensor low-rankness or tensor factorization Li et al. (2023b); Guo et al. (2026). Smoothness constraints have also been incorporated to preserve local structures Wang et al. (2026); Yang et al. (2025). However, consensus-anchor methods mainly enforce agreement among anchor representations, whereas many tensor methods exploit cross-view correlations only after separately constructed anchor graphs have been fixed. Neither strategy forms a direct feedback loop between a shared adaptive anchor dictionary and tensor factorization.

3.3 Anchor Graph Tensor Factorization

To model cross-view correlations throughout the learning process, recent methods directly organize multi-view anchor graphs into tensors and perform joint tensor factorization Li et al. (2023a); Feng et al. (2024). A representative example is Orth-NTF Li et al. (2023a), which factorizes the anchor representation tensor as

$$\mathcal{H} \approx \mathcal{F} * \mathcal{Z}^{\mathrm{T}},$$

while imposing nonnegative and orthogonal constraints on the learned tensor representation. Tensor Schatten-p regularization can be further introduced to exploit the low-rank structure of $\mathcal{Z}$, thereby capturing high-order cross-view correlations. This paradigm has inspired several extensions, including bi-orthogonal tri-factorization Wang et al. (2025) and deep tensor factorization Wu et al. (2026).

Despite their effectiveness, existing anchor graph tensor factorization methods commonly treat anchor construction and tensor learning as separate stages. In contrast, our shared anchor matrix is an optimization variable of the tensor model rather than a fixed input. The anchor representations connect view reconstruction with tensor factorization, enabling the cross-view structure learned in the tensor domain to refine the unified anchors and, in turn, allowing the updated anchors to reshape the tensor representations.

4 METHODOLOGY

In this section, we first introduce the motivation and model formulation of the proposed **LSUATNOF**. We then develop an efficient optimization algorithm for the resulting problem and analyze its computational complexity.

4.1 MOTIVATION AND MODEL FORMULATION

4.1.1 JOINT VIEW-SPECIFIC ANCHOR LEARNING AND TENSOR FACTORIZATION

Anchor-based tensor factorization replaces sample-level full graphs with compact sample-anchor representations and stacks them into a tensor to exploit high-order cross-view correlations. This design improves scalability, but its effectiveness depends strongly on how the anchors and their representations interact with subsequent tensor learning.

However, in many existing methods, anchors are selected or constructed before model optimization and remain fixed during the subsequent learning process. Consequently, the anchor structure cannot be adjusted according to the evolving representations or the cross-view information revealed during tensor optimization. If the predefined anchors are not sufficiently representative, the resulting structural deficiency may propagate to the anchor graphs and further affect tensor modeling.

To alleviate this limitation, we first consider a coupled scheme in which the anchor matrices and their representations are adaptively optimized together with nonnegative orthogonal tensor factorization. The anchor representations therefore serve as an interface between view reconstruction and tensor modeling rather than as fixed preprocessing outputs. Specifically, we formulate

$$
\begin{aligned}
\min_{\Theta} \quad & \|\mathcal{H}\|_{\circledast}^{p} + \lambda_1 \sum_{v=1}^{V} \|\mathbf{E}^{(v)}\|_{2,1} + \lambda_2 \|\mathcal{E}\|_{\mathrm{F}}^{2} \\
\text{s.t.} \quad & \mathbf{X}^{(v)} = \mathbf{A}^{(v)}\mathbf{H}^{(v)} + \mathbf{E}^{(v)}, \quad v = 1, \ldots, V, \mathcal{H} = \Phi\left(\mathbf{H}^{(1)}, \ldots, \mathbf{H}^{(V)}\right), \\
& \mathcal{H} = \mathcal{F} * \mathcal{Z}^{\mathrm{T}} + \mathcal{E}, \qquad \mathbf{A}^{(v)} \geq 0, \quad v = 1, \ldots, V, \\
& \mathcal{F} \geq 0, \qquad \mathcal{Z}^{\mathrm{T}} * \mathcal{Z} = \mathcal{I}_c.
\end{aligned}
$$

where $\Theta = \{\mathbf{A}^{(v)}, \mathbf{H}^{(v)}, \mathbf{E}^{(v)}\}_{v=1}^{V} \cup \{\mathcal{F}, \mathcal{Z}, \mathcal{E}\}$, and $\Phi(\cdot)$ stacks the view-specific anchor representations into $\mathcal{H} \in \mathbb{R}^{m \times n \times V}$, with $\mathbf{A}^{(v)} \in \mathbb{R}_{+}^{d_v \times m}$.

The first constraint reconstructs each view through an adaptive anchor representation, while the tensor terms regularize the stacked representations using their cross-view structure. Since $\mathbf{H}^{(v)}$ participates in both parts, tensor learning can affect $\mathbf{A}^{(v)}$ through the reconstruction constraint, rather than merely operating on a fixed anchor graph.

Nevertheless, the above formulation still learns $\{\mathbf{A}^{(v)}\}_{v=1}^{V}$ independently in different feature spaces. Since heterogeneous views generally have different dimensions and semantic descriptions, independently learned anchors do not necessarily possess explicit correspondence across views. Consequently, the same anchor index in different $\mathbf{H}^{(v)}$ may represent different semantic structures. Directly stacking these representations into $\mathcal{H}$ may therefore introduce structural inconsistency along the anchor mode and weaken the effectiveness of subsequent tensor factorization.

4.1.2 UNIFIED ANCHOR LEARNING

To establish explicit anchor correspondence across heterogeneous views, we further introduce a unified anchor learning strategy. Since the original views lie in different feature spaces, we first employ view-specific projection matrices to map them into a common latent space. A shared anchor

matrix is then learned in this common space and used by all views to generate their corresponding anchor representations.

Specifically, the unified anchor representation is modeled as

$$\begin{aligned} &\mathbf{P}^{(v)}\mathbf{X}^{(v)} = \mathbf{A}\mathbf{H}^{(v)} + \mathbf{E}^{(v)}, \quad v = 1, \ldots, V, \\ \text{s.t.} \quad &\mathbf{P}^{(v)}\big(\mathbf{P}^{(v)}\big)^{\mathrm{T}} = \mathbf{I}_d, \qquad \mathbf{A}\mathbf{A}^{\mathrm{T}} = \mathbf{I}_d. \end{aligned}$$

where $\mathbf{P}^{(v)} \in \mathbb{R}^{d\times d_v}$ projects the v-th view into a d-dimensional common space, $\mathbf{A} \in \mathbb{R}^{d\times m}$ denotes the unified anchor matrix, and $\mathbf{H}^{(v)} \in \mathbb{R}^{m\times n}$ is the corresponding sample-anchor representation.

The projection matrices remove the dimensional inconsistency among views, while the shared anchor matrix ensures that the same anchor index has a consistent meaning across different views. The orthogonality constraints on $\mathbf{P}^{(v)}$ and $\mathbf{A}$ reduce redundant information and help preserve the intrinsic structure of the projected data and the diversity of the learned anchors. As a result, the view-specific representations $\{\mathbf{H}^{(v)}\}_{v=1}^{V}$ become aligned along the anchor dimension.

This explicit anchor correspondence is particularly important for tensor modeling. After stacking the aligned representations into $\mathcal{H}$, each anchor-mode index has the same semantics across all views. Moreover, $\mathbf{A}$ and $\{\mathbf{H}^{(v)}\}_{v=1}^{V}$ are optimized jointly rather than sequentially, so the shared dictionary is continuously refined according to the tensor-structured representations.

4.1.3 Low-Rank and Smoothness-Aware Tensor Regularization

Although unified anchor learning improves the semantic consistency of $\mathcal{H}$, an additional issue remains in tensor regularization. Conventional tensor Schatten-p regularization mainly characterizes global low-rank correlations among the anchor representations. Such a global prior is effective for extracting shared structures across views, but it does not explicitly exploit the local smoothness inherent in the sample representations.

In multi-view clustering, samples with similar semantic structures are expected to exhibit similar representation patterns. Preserving such local variation can provide complementary structural information to global low-rankness. A straightforward strategy is to introduce an additional smoothness penalty, but this separates low-rankness and smoothness into two independent priors and introduces another trade-off parameter.

To jointly model these two properties, we employ the tensor total variation Schatten-p regularization Wang et al. (2026):

$$\|\mathcal{H}\|_{\mathfrak{D},\mathrm{TTV}}^{p} = \sum_{k}^{\Gamma} \|\mathcal{G}_k\|_{\mathfrak{D}}^{p}, \qquad \mathcal{G}_k = \nabla_k(\mathcal{H}),$$

where $\Gamma \subseteq \{1, 2, 3\}$ denotes the selected prior directions and $\nabla_k(\mathcal{H})$ is the difference operator along the k-th mode.

Instead of directly regularizing $\mathcal{H}$, the TTV-Schatten-p regularizer imposes the low-rank prior on its directional gradient tensors. The difference operator explicitly characterizes local variations, while the Schatten-p penalty captures the low-dimensional structure contained in these variations. Therefore, local smoothness and global low-rankness are integrated within a single regularization term rather than modeled independently.

Compared with introducing a separate smoothness penalty, this formulation avoids an extra trade-off parameter. In our framework, the regularizer is applied specifically to the aligned anchor tensor, so its low-rank and smooth structure participates directly in the joint refinement of the anchor representations rather than being imposed on a separately learned tensor.

4.1.4 Overall Formulation

By integrating unified anchor learning, nonnegative orthogonal tensor factorization, and low-rank and smoothness-aware tensor regularization into a unified framework, we arrive at the overall for-

mulation of the proposed **LS-UATNOF** model:

$$\begin{aligned}
\min_{\Theta} \quad & \|\mathcal{H}\|^p_{\circledS,\mathrm{TTV}} + \lambda_1 \sum_{v=1}^{V} \|\mathbf{E}^{(v)}\|_{2,1} + \lambda_2 \|\mathcal{E}\|_{\mathrm{F}}^2 \\
\text{s.t.} \quad & \mathbf{P}^{(v)}\mathbf{X}^{(v)} = \mathbf{A}\mathbf{H}^{(v)} + \mathbf{E}^{(v)}, \quad v = 1, \ldots, V, \mathcal{H} = \Phi\left(\mathbf{H}^{(1)}, \ldots, \mathbf{H}^{(V)}\right), \\
& \mathcal{H} = \mathcal{F} * \mathcal{Z}^{\mathrm{T}} + \mathcal{E}, \mathbf{A}\mathbf{A}^{\mathrm{T}} = \mathbf{I}_d, \quad \mathbf{P}^{(v)}\left(\mathbf{P}^{(v)}\right)^{\mathrm{T}} = \mathbf{I}_d, \quad v = 1, \ldots, V, \\
& \mathcal{F} \geq 0, \quad \mathcal{Z}^{\mathrm{T}} * \mathcal{Z} = \mathcal{I}_c,
\end{aligned} \tag{1}$$

where $\Theta = \{\mathbf{P}^{(v)}, \mathbf{H}^{(v)}, \mathbf{E}^{(v)}\}_{v=1}^{V} \cup \{\mathbf{A}, \mathcal{F}, \mathcal{Z}, \mathcal{E}\}$ collects all optimization variables. Here, $\mathcal{F} \in \mathbb{R}_{+}^{m \times c \times V}$ and $\mathcal{Z} \in \mathbb{R}^{n \times c \times V}$ denote the nonnegative basis tensor and the orthogonal clustering representation tensor, respectively, where c is the number of clusters. Importantly, equation 1 is not a sequential composition of anchor learning and tensor factorization. The tensor factors and TTV regularization shape $\mathcal{H}$; through the splitting constraint $\mathcal{H} = \mathcal{Q}$, this structure is transferred to $\{\mathbf{Q}^{(v)}\}_{v=1}^{V}$, which participate in updating $\mathbf{A}$ and $\{\mathbf{P}^{(v)}\}_{v=1}^{V}$. Conversely, the updated unified anchors reshape the representations entering $\mathcal{H}$. This bidirectional coupling permits anchor learning and high-order tensor modeling to refine each other during optimization.

After optimization, the final clustering labels are obtained from the learned tensor factor $\mathcal{Z}$; the detailed post-processing procedure is provided in Appendix C.

4.2 Optimization

By introducing auxiliary variables $\mathcal{G}_k = \nabla_k(\mathcal{H})$ and $\mathcal{H} = \mathcal{Q}$, where $\mathcal{Q} = \Phi(\mathbf{Q}^{(1)}, \ldots, \mathbf{Q}^{(V)})$, the augmented Lagrangian function associated with problem equation 1 can be formulated as

$$\begin{aligned}
\mathcal{L}(\Theta) = & \sum_{k}^{\Gamma} \|\mathcal{G}_k\|^p_{\circledS} + \lambda_1 \sum_{v=1}^{V} \|\mathbf{E}^{(v)}\|_{2,1} + \lambda_2 \|\mathcal{E}\|_{\mathrm{F}}^2 + \frac{\mu}{2} \sum_{v=1}^{V} \left\| \mathbf{P}^{(v)}\mathbf{X}^{(v)} - \mathbf{A}\mathbf{Q}^{(v)} - \mathbf{E}^{(v)} + \frac{\mathbf{J}^{(v)}}{\mu} \right\|_{\mathrm{F}}^2 \\
& + \frac{\theta}{2} \left\| \mathcal{H} - \mathcal{F} * \mathcal{Z}^{\mathrm{T}} - \mathcal{E} + \frac{\mathcal{Y}_1}{\theta} \right\|_{\mathrm{F}}^2 + \frac{\gamma}{2} \sum_{k}^{\Gamma} \left\| \mathcal{G}_k - \nabla_k(\mathcal{H}) + \frac{\mathcal{Y}_{2,k}}{\gamma} \right\|_{\mathrm{F}}^2 + \frac{\gamma}{2} \left\| \mathcal{H} - \mathcal{Q} + \frac{\mathcal{Y}_3}{\gamma} \right\|_{\mathrm{F}}^2,
\end{aligned}$$

where $\{\mathbf{J}^{(v)}\}_{v=1}^{V}$, $\mathcal{Y}_1$, $\{\mathcal{Y}_{2,k}\}_{k}^{\Gamma}$, and $\mathcal{Y}_3$ are the Lagrange multipliers, and μ, θ, and γ are the penalty parameters.

We employ the alternating direction method of multipliers (ADMM) Boyd et al. (2011) to optimize the above problem. The detailed derivations and explicit update rules of all variables are provided in Appendix C.

4.3 Complexity Analysis

Since the unified projected dimension d, the anchor number m, the number of clusters c, and the number of views V are usually much smaller than the sample size n, the dominant per-iteration computational complexity of **LS-UATNOF** can be simplified as $\mathcal{O}(n \sum_{v=1}^{V} d_v + n \log n)$, and further reduces to $\mathcal{O}(n \log n)$ when the feature dimensions are regarded as constants. The detailed time and space complexity analysis is provided in Appendix B.

5 Experiments

In this section, we evaluate the proposed **LS-UATNOF** on benchmark multi-view datasets. We compare it with state-of-the-art methods, conduct ablation, parameter sensitivity, and convergence studies, and further examine the effect of tensor total-variation regularization along different directions. Additional results are provided in Appendix D. All experiments are conducted on a computer with a 12th Gen Intel(R) Core(TM) i7-12650H CPU and 32 GB RAM.

5.1 Experimental Settings

Datasets. We evaluate the proposed method on six representative multi-view datasets of varying scales, including HW, BDGP, CCV, Animal, NUSWIDEOBJ, and NoisyMNIST. The detailed information of these datasets is summarized in Table 1. Additional results on two small-scale datasets are provided in Appendix D.

Table 1: Description of the benchmark datasets.

Dataset	Samples	Classes	Dim.
HW	2000	10	$216, 76, 64, 6, 240, 47$
BDGP	2500	5	$1000, 500$
CCV	6773	20	$20, 20, 20$
Animal	11673	20	$2689, 2000, 2001, 2000$
NUSWIDEOBJ	30000	31	$65, 226, 145, 74, 129$
NoisyMNIST	50000	10	$784, 784$

Evaluation metrics. To quantitatively evaluate the clustering performance, we adopt five widely used metrics, including accuracy (ACC), normalized mutual information (NMI), Purity, F-score, and adjusted rand index (ARI). For compact presentation, the main text reports ACC, NMI, and Purity, and the complete five-metric results are provided in Appendix D. For all metrics, a higher value indicates better clustering performance.

Parameter settings. For **LS-UATNOF**, λ_1 and λ_2 are selected from $\{10^{-3}, 5\times10^{-3}, 10^{-2}, 5\times10^{-2}, 10^{-1}, 5\times10^{-1}, 1\}$. We set $d = \min\{c, \min_v d_v\}$ and tune the anchor number $m \in \{d, 2d, 3d, 4d, 5d\}$. The TTV regularization is imposed along the sample mode with $\Gamma = \{2\}$, and $p \in \{0.1, 0.2, \ldots, 0.9, 1.0\}$. The maximum iteration number and tolerance are set to 200 and 10^{-6}, respectively. We run k-means 100 times with a fixed random seed. Parameters of the compared methods are tuned according to their recommended settings, and the best results are reported.

Compared methods. We compare the proposed **LS-UATNOF** with several state-of-the-art multi-view clustering methods, including t-SVD-MSC Xie et al. (2018), WTNNM Gao et al. (2020), Orth-NTF Li et al. (2023a), EOMSC-CA Liu et al. (2022), RCAGL Liu et al. (2024b), DSTL Xu et al. (2024), ESTMC Ji & Feng (2025), TLRLF4MVC Long et al. (2025), LMTC Liu et al. (2025), and ARIA Kong et al. (2026). The source codes of all compared methods are obtained from the corresponding authors or their official websites.

5.2 Experimental Results and Analysis

The main clustering results are reported in Tables 2 and 3. To keep the main text compact, we report ACC, NMI, and Purity in the main tables, while the complete results with F-score and ARI are provided in Appendix D. The best and second-best results are highlighted in bold and underlined, respectively.

Table 2: Clustering results on HW, BDGP, and CCV.

Method	HW			BDGP			CCV		
	ACC	NMI	Purity	ACC	NMI	Purity	ACC	NMI	Purity
t-SVD-MSC(IJCV'18)	**1.000**	**1.000**	**1.000**	0.432	0.280	0.447	0.478	0.413	0.498
WTNNM(AAAI'20)	0.945	0.928	0.945	0.496	0.373	0.514	0.476	0.502	0.468
EOMSC-CA(AAAI'22)	0.762	0.825	0.764	0.420	0.145	0.420	0.243	0.187	0.266
Orth-NTF(NeurIPS'23)	0.985	0.969	0.985	0.465	0.318	0.465	0.565	0.527	0.622
RCAGL(TKDE'24)	0.877	0.806	0.877	0.503	0.303	0.622	0.244	0.190	0.262
DSTL(TMM'24)	0.970	0.960	0.973	<u>0.986</u>	<u>0.957</u>	<u>0.986</u>	0.706	<u>0.750</u>	0.732
ESTMC(TPAMI'25)	<u>0.998</u>	<u>0.995</u>	<u>0.998</u>	0.984	0.950	0.984	<u>0.746</u>	0.736	<u>0.755</u>
TLRLF4MVC(TPAMI'25)	0.883	0.900	0.883	0.276	0.025	0.276	0.202	0.160	0.233
LMTC(CVPR'25)	0.942	0.873	0.942	0.541	0.336	0.567	0.237	0.190	0.272
ARIA(TKDE'26)	0.896	0.814	0.896	0.558	0.320	0.561	0.203	0.175	0.237
LS-UATNOF	**1.000**	**1.000**	**1.000**	**0.995**	**0.982**	**0.995**	**0.872**	**0.849**	**0.875**

Tables 2 and 3 report the main clustering results. Overall, **LS-UATNOF** achieves the best or tied-best performance in all 18 main evaluation cases. It obtains perfect performance on HW and shows more pronounced advantages on the challenging BDGP, CCV, Animal, NUSWIDEOBJ, and NoisyMNIST datasets. In terms of ACC, **LS-UATNOF** surpasses the second-best method by 0.9%, 12.6%, 5.0%, 7.9%, and 1.1% on BDGP, CCV, Animal, NUSWIDEOBJ, and NoisyMNIST, respectively.

Beyond ACC, **LS-UATNOF** consistently improves NMI and Purity. On CCV, it outperforms the second-best results by 9.9% and 12.0%, respectively. Moreover, it successfully handles

Table 3: Clustering results on Animal, NUSWIDEOBJ, and NoisyMNIST. OM denotes out of memory.

Method	Animal			NUSWIDEOBJ			NoisyMNIST		
	ACC	NMI	Purity	ACC	NMI	Purity	ACC	NMI	Purity
t-SVD-MSC(IJCV'18)	0.167	0.125	0.200	OM	OM	OM	OM	OM	OM
WTNNM(AAAI'20)	0.231	0.194	0.225	OM	OM	OM	OM	OM	OM
EOMSC-CA(AAAI'22)	0.188	0.152	0.206	0.195	0.129	0.237	0.547	0.470	0.562
Orth-NTF(NeurIPS'23)	0.573	0.664	0.602	0.178	0.146	0.205	0.701	0.729	0.747
RCAGL(TKDE'24)	0.160	0.135	0.230	0.192	0.125	0.361	0.520	0.463	0.560
DSTL(TMM'24)	0.764	0.825	0.808	0.130	0.175	0.221	0.871	0.816	0.901
ESTMC(TPAMI'25)	0.922	0.916	0.934	0.154	0.130	0.246	0.987	0.971	0.987
TLRLF4MVC(TPAMI'25)	0.165	0.138	0.200	0.168	0.190	0.304	OM	OM	OM
LMTC(CVPR'25)	0.177	0.137	0.210	0.123	0.100	0.216	0.606	0.593	0.657
ARIA(TKDE'26)	0.171	0.136	0.209	0.136	0.111	0.228	0.516	0.464	0.561
LS-UATNOF	**0.972**	**0.968**	**0.972**	**0.274**	**0.266**	**0.384**	**0.998**	**0.995**	**0.998**

Table 4: Ablation study of **LS-UATNOF** on six benchmark datasets.

Method	HW	BDGP	CCV	Animal	NUSWIDEOBJ	NoisyMNIST
w/o UA	**1.000**$_{0.00\%}$	0.991$_{\downarrow 0.40\%}$	0.866$_{\downarrow 0.69\%}$	0.958$_{\downarrow 1.44\%}$	0.247$_{\downarrow 9.85\%}$	0.991$_{\downarrow 0.70\%}$
w/o TTV	**1.000**$_{0.00\%}$	0.968$_{\downarrow 2.71\%}$	0.849$_{\downarrow 2.64\%}$	0.933$_{\downarrow 4.01\%}$	0.178$_{\downarrow 35.04\%}$	0.910$_{\downarrow 8.82\%}$
w/o NTF	0.993$_{\downarrow 0.70\%}$	0.989$_{\downarrow 0.60\%}$	0.718$_{\downarrow 17.66\%}$	0.932$_{\downarrow 4.12\%}$	0.232$_{\downarrow 15.33\%}$	**0.999**$_{\uparrow 0.10\%}$
LS-UATNOF	**1.000**	**0.995**	**0.872**	**0.972**	**0.274**	0.998

NUSWIDEOBJ and NoisyMNIST, on which several self-representation-based methods encounter out-of-memory issues, demonstrating its effectiveness and scalability.

5.3 Ablation Study

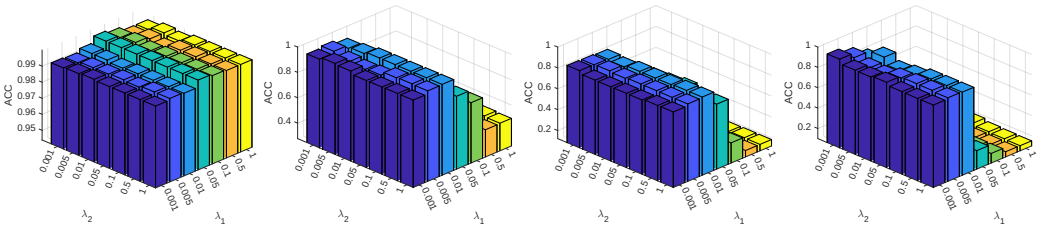
To investigate the contribution of each key component in **LS-UATNOF**, we construct the following three variants:

- **w/o UA**: The unified anchor matrix is replaced by view-specific anchor matrices, i.e., $\mathbf{P}^{(v)}\mathbf{X}^{(v)} = \mathbf{A}^{(v)}\mathbf{H}^{(v)} + \mathbf{E}^{(v)}$, to evaluate the effectiveness of unified anchor learning.
- **w/o TTV**: The $\|\mathcal{H}\|^p_{\circledS_p,\mathrm{TTV}}$ regularization is replaced by the conventional tensor Schatten-p regularization $\|\mathcal{H}\|^p_{\circledS_p}$ to verify the contribution of sample-wise local smoothness.
- **w/o NTF**: The nonnegative orthogonal tensor factorization constraint is removed, and clustering is directly performed on the learned anchor representations to assess the effectiveness of tensor factorization.

As shown in Table 4, the red and blue subscripts denote the relative performance decreases and increases with respect to the complete **LS-UATNOF** model, respectively. Overall, **LS-UATNOF** achieves the best or tied-best ACC on five of the six datasets. Replacing $\|\mathcal{H}\|^p_{\circledS_p,\mathrm{TTV}}$ with the conventional Schatten-p regularization $\|\mathcal{H}\|^p_{\circledS_p}$ or removing tensor factorization generally causes clear degradation, verifying the complementary contributions of unified anchor learning, $\|\mathcal{H}\|^p_{\circledS_p,\mathrm{TTV}}$ regularization, and tensor factorization.

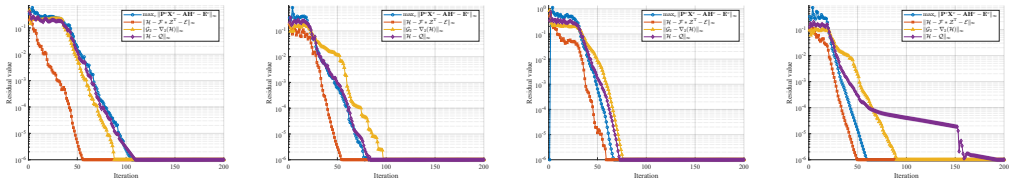
5.4 Parameter Sensitivity Analysis

We investigate the sensitivity of **LS-UATNOF** to the regularization parameters λ_1 and λ_2 on four representative datasets, i.e., HW, BDGP, CCV, and Animal. As shown in Fig. 2, **LS-UATNOF** maintains stable performance over a relatively broad range of λ_1 and λ_2 on HW, while appropriate combinations of the two parameters are more important for the relatively challenging BDGP, CCV, and Animal datasets. Additional sensitivity analyses of the anchor number m and the Schatten parameter p are provided in Appendix D.



(a) HW (b) BDGP (c) CCV (d) Animal

Figure 2: Parameter sensitivity of λ_1 and λ_2.



(a) HW (b) BDGP (c) CCV (d) Animal

Figure 3: Residual convergence curves of **LS-UATNOF** on four benchmark datasets.

5.5 Convergence Analysis

To empirically investigate the convergence behavior of **LS-UATNOF**, we report four primal residuals on HW, BDGP, CCV, and Animal: the reconstruction residual $r_{\mathrm{rec}} = \max_v \|\mathbf{P}^{(v)}\mathbf{X}^{(v)} - \mathbf{A}\mathbf{Q}^{(v)} - \mathbf{E}^{(v)}\|_\infty$, tensor-factorization residual $r_{\mathrm{ten}} = \|\mathcal{H} - \mathcal{F} * \mathcal{Z}^{\mathrm{T}} - \mathcal{E}\|_\infty$, gradient-consistency residual $r_{\mathrm{grad}} = \max_{k\in\Gamma} \|\mathcal{G}_k - \nabla_k(\mathcal{H})\|_\infty$, and splitting residual $r_{\mathrm{hq}} = \|\mathcal{H} - \mathcal{Q}\|_\infty$. As shown in Fig. 3, these residuals exhibit overall decreasing trends and gradually stabilize, with slight fluctuations in the early iterations. Additional clustering-performance curves over iterations are provided in Appendix D.

5.6 Effect of Tensor Total-Variation Directions

Table 5: Clustering performance (ACC) of **LS-UATNOF** with different tensor total-variation directions.

Direction Γ	HW	BDGP	CCV	Animal	NUSWIDEOBJ	NoisyMNIST
$\{1\}$ [m]	0.872$_{\downarrow 12.80\%}$	0.552$_{\downarrow 44.52\%}$	0.233$_{\downarrow 73.28\%}$	0.173$_{\downarrow 82.20\%}$	0.127$_{\downarrow 53.65\%}$	0.487$_{\downarrow 51.20\%}$
$\{3\}$ $[V]$	0.861$_{\downarrow 13.90\%}$	0.532$_{\downarrow 46.53\%}$	0.217$_{\downarrow 75.11\%}$	0.163$_{\downarrow 83.23\%}$	0.135$_{\downarrow 50.73\%}$	0.390$_{\downarrow 60.92\%}$
$\{1,2\}$ $[m,n]$	**1.000**$_{0.00\%}$	0.962$_{\downarrow 3.32\%}$	0.773$_{\downarrow 11.35\%}$	0.894$_{\downarrow 8.02\%}$	0.223$_{\downarrow 18.61\%}$	0.993$_{\downarrow 0.50\%}$
$\{2,3\}$ $[n,V]$	<u>0.999</u>$_{\downarrow 0.05\%}$	0.934$_{\downarrow 6.13\%}$	<u>0.836</u>$_{\downarrow 4.13\%}$	0.824$_{\downarrow 15.23\%}$	0.223$_{\downarrow 18.61\%}$	<u>0.997</u>$_{\downarrow 0.10\%}$
$\{2\}$ $[n]$	**1.000**$_{0.00\%}$	**0.995**$_{0.00\%}$	**0.872**$_{0.00\%}$	**0.972**$_{0.00\%}$	**0.274**$_{0.00\%}$	**0.998**$_{0.00\%}$

As shown in Table 5, the red subscripts report the relative percentage decreases from the best result in each column. Configurations containing the sample mode $\{2\}$ consistently outperform those using only the anchor mode $\{1\}$ or view mode $\{3\}$; hence, we set $\Gamma = \{2\}$ throughout the experiments.

6 Conclusion

In this paper, we proposed **LS-UATNOF**, a unified multi-view clustering framework that jointly learns latent projections, shared anchors, and tensor representations. The unified anchor matrix aligns sample-anchor correspondences across views, while nonnegative orthogonal tensor factorization yields discriminative clustering representations. Moreover, tensor total variation Schatten-p regularization integrates global low-rank correlations with sample-wise local smoothness. Extensive experiments on benchmark datasets demonstrate the effectiveness, scalability, and robustness of the proposed method.

AI USE STATEMENT

Generative AI tools were used to assist with language polishing and to improve the readability of the manuscript. All AI-assisted text was carefully reviewed and revised by the authors, who take full responsibility for the final content, technical claims, and conclusions.

REFERENCES

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

Runzong Dong, Jize Xue, Yongqiang Zhao, Tongle Wu, Ying Liu, and Jonathan Cheung-Wai Chan. Robust geo-sparse and graph-fusion tensor learning for incomplete multi-view clustering. *Information Fusion*, pp. 104716, 2026.

Yangfan Du and Gui-Fu Lu. Joint local smoothness and low-rank tensor representation for robust multi-view clustering. *Pattern Recognition*, 157:110944, 2025.

Uno Fang, Man Li, Jianxin Li, Longxiang Gao, Tao Jia, and Yanchun Zhang. A comprehensive survey on multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 35 (12):12350–12368, 2023.

Wei Feng, Dongyuan Wei, Qianqian Wang, Bo Dong, and Quanxue Gao. Multi-view clustering based on deep non-negative tensor factorization. In *Proceedings of the 32nd ACM international conference on multimedia*, pp. 1130–1138, 2024.

Hongchang Gao, Feiping Nie, Xuelong Li, and Heng Huang. Multi-view subspace clustering. In *Proceedings of the IEEE international conference on computer vision*, pp. 4238–4246, 2015.

Quanxue Gao, Wei Xia, Zhizhen Wan, Deyan Xie, and Pu Zhang. Tensor-svd based graph learning for multi-view subspace clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 3930–3937, 2020.

Jun Guo and Jiahui Ye. Anchors bring ease: An embarrassingly simple approach to partial multi-view clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 118–125, 2019.

Xi Guo, Hongmei Chen, Biao Xiang, Yong Mi, Zhong Yuan, Chuan Luo, and Tianrui Li. A multi-view clustering approach via anchoring high-order graphs into a low-rank tensor. *Information Fusion*, pp. 104317, 2026.

Jintian Ji and Songhe Feng. Anchors crash tensor: Efficient and scalable tensorial multi-view subspace clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(4):2660–2675, 2025.

Misha E. Kilmer and Carla D. Martin. Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, 435(3):641–658, 2011.

Zisen Kong, Dongxia Chang, Yiming Wang, Pengyuan Li, and Yao Zhao. Tensorial multi-view clustering via alternative rank minimization and inter-view alignment. *IEEE Transactions on Knowledge and Data Engineering*, 2026.

Jing Li, Quanxue Gao, Qianqian Wang, Ming Yang, and Wei Xia. Orthogonal non-negative tensor factorization based multi-view clustering. *Advances in neural information processing systems*, 36:18186–18202, 2023a.

Jing Li, Qianqian Wang, Ming Yang, Quanxue Gao, and Xinbo Gao. Efficient anchor graph factorization for multi-view clustering. *IEEE Transactions on Multimedia*, 26:5834–5845, 2023b.

Zhenglai Li, Chang Tang, Xiao Zheng, Xinwang Liu, Wei Zhang, and En Zhu. High-order correlation preserved incomplete multi-view subspace clustering. *IEEE Transactions on Image Processing*, 31:2067–2080, 2022.

Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):171–184, 2012.

Jiyuan Liu, Xinwang Liu, Chuankun Li, Xinhang Wan, Hao Tan, Yi Zhang, Weixuan Liang, Qian Qu, Yu Feng, Renxiang Guan, et al. Large-scale multi-view tensor clustering with implicit linear kernels. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20727–20736. IEEE, 2025.

Maoshan Liu, Vasile Palade, and Zhonglong Zheng. Learning the consensus and complementary information for large-scale multi-view clustering. *Neural Networks*, 172:106103, 2024a.

Suyuan Liu, Siwei Wang, Pei Zhang, Kai Xu, Xinwang Liu, Changwang Zhang, and Feng Gao. Efficient one-pass multi-view subspace clustering with consensus anchors. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 7576–7584, 2022.

Suyuan Liu, Qing Liao, Siwei Wang, Xinwang Liu, and En Zhu. Robust and consistent anchor graph learning for multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 36 (8):4207–4219, 2024b.

Zhen Long, Qiyuan Wang, Yazhou Ren, Yipeng Liu, and Ce Zhu. Tlrlf4mvc: Tensor low-rank and low-frequency for scalable multi-view clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(8):6900–6911, 2025.

Shirui Luo, Changqing Zhang, Wei Zhang, and Xiaochun Cao. Consistent and specific multi-view subspace clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Wei Lv, Chao Zhang, Huaxiong Li, Xiuyi Jia, and Chunlin Chen. Joint projection learning and tensor decomposition-based incomplete multiview clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 35(12):17559–17570, 2023.

Dan Song, Jing-Hua Yang, Heng-Chao Li, and Qian Du. Enhanced anchor multi-view clustering via projection learning and high-order correlation exploitation. *Information Fusion*, pp. 104297, 2026.

Shiliang Sun. A survey of multi-view machine learning. *Neural computing and applications*, 23(7): 2031–2038, 2013.

Shitao Tang, Fuyang Zhang, Jiacheng Chen, Peng Wang, and Yasutaka Furukawa. Mvdiffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion. *Advances in neural information processing systems*, 36:51202–51233, 2023.

Hailin Wang, Jiangjun Peng, Wenjin Qin, Jianjun Wang, and Deyu Meng. Guaranteed tensor recovery fused low-rankness and smoothness. *arXiv preprint arXiv:2302.02155*, 2023.

Hao Wang, Yan Yang, and Bing Liu. Gmc: Graph-based multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 32(6):1116–1129, 2019a.

Hao Wang, Yan Yang, Bing Liu, and Hamido Fujita. A study of graph-based system for multi-view clustering. *Knowledge-Based Systems*, 163:1009–1019, 2019b.

Rui Wang, Yuxuan Liu, Guangyu Yang, Quanxue Gao, and Cheng Deng. Bi-orthogonal non-negative tensor tri-factorization for tensorized label learning. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 2304–2312, 2025.

Yang Wang, Lin Wu, Xuemin Lin, and Junbin Gao. Multiview spectral clustering via structured low-rank matrix factorization. *IEEE transactions on neural networks and learning systems*, 29 (10):4833–4843, 2018.

Yapeng Wang, Quanxue Gao, Fangfang Li, Yu Yun, and Ming Yang. Unified view extraction with low-rankness and smoothness fusion for multi-view subspace clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 26634–26642, 2026.

Jian-Sheng Wu, Sang-Qi Zhao, Weidong Min, and Wei-Shi Zheng. Tensorized noise-aware anchor graph learning with deep complementary information propagation for scalable multi-view clustering. *IEEE Transactions on Multimedia*, 2026.

Jianlong Wu, Zhouchen Lin, and Hongbin Zha. Essential tensor learning for multi-view spectral clustering. *IEEE Transactions on Image Processing*, 28(12):5910–5922, 2019.

Jianlong Wu, Xingyu Xie, Liqiang Nie, Zhouchen Lin, and Hongbin Zha. Unified graph and low-rank tensor learning for multi-view clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 6388–6395, 2020.

Yuan Xie, Dacheng Tao, Wensheng Zhang, Yan Liu, Lei Zhang, and Yanyun Qu. On unifying multi-view self-representations for clustering by tensor multi-rank minimization. *International Journal of Computer Vision*, 126(11):1157–1179, 2018.

Deng Xu, Chao Zhang, Zechao Li, Chunlin Chen, and Huaxiong Li. Fast disentangled slim tensor learning for multi-view clustering. *IEEE Transactions on Multimedia*, 27:1254–1265, 2024.

Tinghe Yan, Qiang Guo, Jian-Xun Mi, Shuo Zhang, and Weisheng Li. Multi-view subspace clustering via latent representation learning and augmented tensorized self-representation. *Information Fusion*, pp. 104686, 2026.

Tianchuan Yang, Haiqiang Chen, Haoyan Yang, Man-Sheng Chen, Xiangcheng Li, Youming Sun, and Chang-Dong Wang. Smoothness-induced efficient incomplete multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

Changqing Zhang, Qinghua Hu, Huazhu Fu, Pengfei Zhu, and Xiaochun Cao. Latent multi-view subspace clustering. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 4333–4341. IEEE, 2017.

Chao Zhang, Xiuyi Jia, Zechao Li, Chunlin Chen, and Huaxiong Li. Learning cluster-wise anchors for multi-view clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 16696–16704, 2024.

Mengjiao Zhang, Xinwang Liu, Tianhao Han, Xiaofeng Qu, and Sijie Niu. Adaptive anchor-guided representation learning for efficient multi-view subspace clustering. *IEEE Transactions on Image Processing*, 2025.

Pei Zhang, Xinwang Liu, Jian Xiong, Sihang Zhou, Wentao Zhao, En Zhu, and Zhiping Cai. Consensus one-step multi-view subspace clustering. *IEEE Transactions on Knowledge and Data Engineering*, 34(10):4676–4689, 2020.

A Additional Preliminaries and Notations

Table 6: Summary of main notations used in this paper.

Notation	Description
V	Number of views
n	Number of samples
c	Number of clusters
m	Number of unified anchors
d	Unified projected dimension
$\mathbf{X}^{(v)} \in \mathbb{R}^{d_v \times n}$	Data matrix of the v-th view
$\mathbf{P}^{(v)} \in \mathbb{R}^{d \times d_v}$	Projection matrix of the v-th view
$\mathbf{A} \in \mathbb{R}^{d \times m}$	Unified anchor matrix
$\mathbf{H}^{(v)} \in \mathbb{R}^{m \times n}$	Anchor representation of the v-th view
$\mathbf{E}^{(v)} \in \mathbb{R}^{d \times n}$	View-specific reconstruction error
$\mathcal{H} \in \mathbb{R}^{m \times n \times V}$	Anchor representation tensor
$\mathcal{E} \in \mathbb{R}^{m \times n \times V}$	Tensor error term
$\mathcal{F} \in \mathbb{R}^{m \times c \times V}$	Nonnegative tensor factor
$\mathcal{Z} \in \mathbb{R}^{n \times c \times V}$	Cluster-indicator tensor factor

Definition 2 (Tensor t-product Kilmer & Martin (2011))**.** *Given two third-order tensors $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{Y} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$, their t-product $\mathcal{C} = \mathcal{X} * \mathcal{Y} \in \mathbb{R}^{n_1 \times n_4 \times n_3}$ can be computed in the Fourier domain by $\overline{\mathcal{C}}^{(i)} = \overline{\mathcal{X}}^{(i)} \overline{\mathcal{Y}}^{(i)}$, $i = 1, \ldots, n_3$, followed by inverse FFT along the third mode.*

Definition 3 (Tensor Total Variation Schatten-p Norm Wang et al. (2026)). *For a third-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, let $\Gamma \subseteq \{1, 2, 3\}$ denote the predefined set of prior modes. The tensor total variation Schatten-p norm is defined as*

$$\|\mathcal{X}\|^p_{\circledS_p,\mathrm{TTV}} := \sum_{k}^{\Gamma} \|\mathcal{G}_k\|^p_{\circledS_p},$$

where $\mathcal{G}_k = \nabla_k(\mathcal{X})$ denotes the finite-difference tensor of $\mathcal{X}$ along the k-th mode, and $\mathbf{D}_{n_k} \in \mathbb{R}^{n_k \times n_k}$ is the circulant first-order finite-difference matrix generated by $[-1, 1, 0, \ldots, 0]^{\mathrm{T}}$.

Definition 4 (Tensor transpose Kilmer & Martin (2011)). *For $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its transpose $\mathcal{X}^{\mathrm{T}} \in \mathbb{R}^{n_2 \times n_1 \times n_3}$ is obtained by transposing every frontal slice and then reversing the order of the transposed slices from the second to the last. Equivalently,*

$$(\mathcal{X}^{\mathrm{T}})^{(1)} = (\mathcal{X}^{(1)})^{\mathrm{T}}, \qquad (\mathcal{X}^{\mathrm{T}})^{(i)} = (\mathcal{X}^{(n_3-i+2)})^{\mathrm{T}}, \quad i = 2, \ldots, n_3.$$

Definition 5 (Identity tensor Kilmer & Martin (2011)). *The identity tensor $\mathcal{I} \in \mathbb{R}^{n \times n \times n_3}$ is a tensor whose first frontal slice is the identity matrix and whose remaining frontal slices are zero matrices.*

Definition 6 (Orthogonal tensor Kilmer & Martin (2011)). *A tensor $\mathcal{Q} \in \mathbb{R}^{n \times n \times n_3}$ is orthogonal if $\mathcal{Q}^{\mathrm{T}} * \mathcal{Q} = \mathcal{Q} * \mathcal{Q}^{\mathrm{T}} = \mathcal{I}$.*

Definition 7 (Tensor singular value decomposition (t-SVD) Kilmer & Martin (2011)). *For any tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its t-SVD is defined as $\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^{\mathrm{T}}$, where $\mathcal{U}$ and $\mathcal{V}$ are orthogonal tensors and $\mathcal{S}$ is an f-diagonal tensor.*

Definition 8 (Mode-k product Wang et al. (2023)). *Given a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_m}$ and a matrix $\mathbf{U} \in \mathbb{R}^{p \times n_k}$, the mode-$k$ product is denoted as $\mathcal{Y} = \mathcal{X} \times_k \mathbf{U}$, where $\mathcal{Y} \in \mathbb{R}^{n_1 \times \cdots \times n_{k-1} \times p \times n_{k+1} \times \cdots \times n_m}$.*

B Detailed Complexity Analysis

Space complexity. The storage cost of the original multi-view data $\{\mathbf{X}^{(v)}\}_{v=1}^V$ is $\mathcal{O}(\sum_{v=1}^V d_v n)$. During optimization, the projection matrices and unified anchor matrix occupy $\mathcal{O}(d\sum_{v=1}^V d_v + dm)$ memory, while the error variables, tensor variables, auxiliary variables, and multipliers require $\mathcal{O}(Vdn + (|\Gamma| + 1)mnV + mcV + ncV)$ memory. Thus, the overall space complexity is

$$\mathcal{O}\left(\sum_{v=1}^V d_v n + d\sum_{v=1}^V d_v + dm + Vdn + (|\Gamma| + 1)mnV + mcV + ncV\right).$$

Computational complexity. Updating $\{\mathbf{P}^{(v)}\}_{v=1}^V$ requires $\mathcal{O}(\sum_{v=1}^V (dnd_v + d^2 d_v))$, updating $\mathbf{A}$ requires $\mathcal{O}(\sum_{v=1}^V dnd_v + Vdnm + d^2 m)$, and updating $\{\mathbf{Q}^{(v)}\}_{v=1}^V$ and $\{\mathbf{E}^{(v)}\}_{v=1}^V$ requires $\mathcal{O}(\sum_{v=1}^V dnd_v + V(dmn + m^2 n + m^3))$ and $\mathcal{O}(Vdn)$, respectively. The updates of $\mathcal{H}$, $\mathcal{F}$, $\mathcal{Z}$, and $\mathcal{E}$ require $\mathcal{O}(mnV \log(mnV) + |\Gamma|mnV)$, $\mathcal{O}(mncV)$, $\mathcal{O}(V(mnc + nc^2))$, and $\mathcal{O}(mnV)$, respectively. For the tensor total-variation Schatten-p regularization, the finite difference is imposed along the sample mode, while the tensor Schatten-p shrinkage is computed by applying FFT along the view mode. Thus, each $\mathcal{G}_k$ update involves FFTs on an $m \times n \times V$ tensor and SVDs of V frontal slices of size $m \times n$, leading to $\mathcal{O}(mnV \log V + Vm^2 n)$ for each $k \in \Gamma$. Therefore, updating $\{\mathcal{G}_k\}_{k\in\Gamma}$ costs $\mathcal{O}(|\Gamma|(mnV \log V + Vm^2 n))$. Ignoring lower-order terms and constants, the per-iteration complexity can be simplified as

$$\mathcal{O}\left(n\sum_{v=1}^V d_v + n \log n\right),$$

which further becomes approximately $\mathcal{O}(n \log n)$ when $\{d_v\}_{v=1}^V$ are regarded as constants.

C Detailed Optimization Procedures

This appendix provides the detailed derivations of the ADMM updates used in Algorithm 1.

- **Clustering assignment**

Algorithm 1 Optimization of the proposed model

Input: $\{\mathbf{X}^{(v)}\}_{v=1}^{V}$, m, d, λ_1, λ_2, p, $\Gamma = \{2\}$, $\varepsilon = 10^{-6}$, and $T_{\max} = 200$.
Output: Clustering labels.

```
Normalize {X^(v)}_{v=1}^V and initialize {P^(v), Q^(v), E^(v)}_{v=1}^V, A, Z, F, H, G_k, and all multipliers.
for t = 1 to T_max do
    Update {P^(v)}_{v=1}^V by equation 3.
    Update A by equation 4.
    Update {Q^(v)}_{v=1}^V and {E^(v)}_{v=1}^V by equation 5 and equation 6.
    Construct Q = Φ(Q^(1), ..., Q^(V)).
    Update H, F, Z, and E by equation 9, equation 10, equation 12, and equation 13.
    Update {G_k}_{k∈Γ} by equation 15.
    Update the multipliers by equation 16.
    Update μ ← min(ρμ, μ_max), θ ← min(ρθ, θ_max), γ ← min(ργ, γ_max).
    if t > 5 and max(r_rec, r_ten, r_grad, r_hq) < ε then
        break
    end if
end for
Construct the final representation from Z and perform k-means.
```

Once the optimal clustering representation tensor $\mathcal{Z} \in \mathbb{R}^{n\times c\times V}$ is obtained, its frontal slices are concatenated to form the final sample representation:

$$\mathbf{R} = \left[\mathcal{Z}^{(1)}, \mathcal{Z}^{(2)}, \ldots, \mathcal{Z}^{(V)}\right] \in \mathbb{R}^{n\times cV}.$$

The rows of $\mathbf{R}$ are then normalized by their ℓ_2 norms, i.e., $\widehat{\mathbf{R}}_{i,:} = \mathbf{R}_{i,:}/\|\mathbf{R}_{i,:}\|_2$. Finally, k-means is applied to $\widehat{\mathbf{R}}$ with c clusters to obtain the final clustering assignments. In this manner, the information encoded in all frontal slices of $\mathcal{Z}$ is directly utilized for clustering without constructing an additional sample-level affinity matrix.

• Solving $\{\mathbf{P}^{(v)}\}_{v=1}^{V}$ subproblem

By fixing the other variables, the subproblem with respect to $\mathbf{P}^{(v)}$ can be formulated as follows:

$$\min_{\mathbf{P}^{(v)}(\mathbf{P}^{(v)})^{\mathrm{T}}=\mathbf{I}_d} \left\| \mathbf{P}^{(v)}\mathbf{X}^{(v)} - \mathbf{A}\mathbf{Q}^{(v)} - \mathbf{E}^{(v)} + \frac{\mathbf{J}^{(v)}}{\mu} \right\|_{\mathrm{F}}^{2}. \tag{2}$$

Letting $\mathbf{M}_1^{(v)} = \left(\mathbf{A}\mathbf{Q}^{(v)} + \mathbf{E}^{(v)} - \frac{\mathbf{J}^{(v)}}{\mu}\right)(\mathbf{X}^{(v)})^{\mathrm{T}}$, we approximate the update by an orthogonal Procrustes problem under the row-orthogonality constraint, and equation 2 can be rewritten as

$$\max_{\mathbf{P}^{(v)}(\mathbf{P}^{(v)})^{\mathrm{T}}=\mathbf{I}_d} \operatorname{tr}\left(\mathbf{P}^{(v)}(\mathbf{M}_1^{(v)})^{\mathrm{T}}\right).$$

Following Zhang et al. (2017), the resulting update for $\mathbf{P}^{(v)}$ can be obtained as

$$\mathbf{P}^{(v)} = \mathbf{U}_1^{(v)}(\mathbf{V}_1^{(v)})^{\mathrm{T}}, \tag{3}$$

where $\mathbf{M}_1^{(v)} = \mathbf{U}_1^{(v)}\mathbf{\Sigma}_1^{(v)}(\mathbf{V}_1^{(v)})^{\mathrm{T}}$ denotes the singular value decomposition of $\mathbf{M}_1^{(v)}$.

• Solving A subproblem

Similar to the optimization of $\mathbf{P}^{(v)}$, we employ the same orthogonal Procrustes approximation for $\mathbf{A}$, leading to

$$\max_{\mathbf{A}\mathbf{A}^{\mathrm{T}}=\mathbf{I}_d} \operatorname{tr}\left(\mathbf{A}\mathbf{M}_2^{\mathrm{T}}\right),$$

where

$$\mathbf{M}_2 = \sum_{v=1}^{V}\left(\mathbf{P}^{(v)}\mathbf{X}^{(v)} - \mathbf{E}^{(v)} + \frac{\mathbf{J}^{(v)}}{\mu}\right)(\mathbf{Q}^{(v)})^{\mathrm{T}}.$$

Then the resulting update for $\mathbf{A}$ can be obtained as

$$\mathbf{A} = \mathbf{U}_2\mathbf{V}_2^{\mathrm{T}}, \tag{4}$$

where $\mathbf{M}_2 = \mathbf{U}_2\mathbf{\Sigma}_2\mathbf{V}_2^{\mathrm{T}}$ denotes the singular value decomposition of $\mathbf{M}_2$.

• **Solving $\{\mathbf{Q}^{(v)}\}_{v=1}^{V}$ subproblem**

For each view, the terms involving $\mathbf{Q}^{(v)}$ are

$$\begin{aligned}\min_{\mathbf{Q}^{(v)}} \frac{\mu}{2}\left\|\mathbf{P}^{(v)}\mathbf{X}^{(v)} - \mathbf{A}\mathbf{Q}^{(v)} - \mathbf{E}^{(v)} + \frac{\mathbf{J}^{(v)}}{\mu}\right\|_{\mathrm{F}}^2 \\ + \frac{\gamma}{2}\left\|\mathbf{H}^{(v)} - \mathbf{Q}^{(v)} + \frac{\mathcal{Y}_3(:,:,v)}{\gamma}\right\|_{\mathrm{F}}^2.\end{aligned}$$

Setting its derivative to zero yields

$$\begin{aligned}\mathbf{Q}^{(v)} = (\mu\mathbf{A}^{\mathrm{T}}\mathbf{A} + \gamma\mathbf{I})^{-1}\Bigg[\mu\mathbf{A}^{\mathrm{T}}\left(\mathbf{P}^{(v)}\mathbf{X}^{(v)} - \mathbf{E}^{(v)} + \frac{\mathbf{J}^{(v)}}{\mu}\right) \\ +\gamma\mathbf{H}^{(v)} + \mathcal{Y}_3(:,:,v)\Bigg].\end{aligned} \tag{5}$$

After all views have been updated, they are stacked to obtain $\mathcal{Q}$.

• **Solving $\{\mathbf{E}^{(v)}\}_{v=1}^{V}$ subproblem**

The subproblem with respect to $\mathbf{E}^{(v)}$ can be written as

$$\min_{\mathbf{E}^{(v)}} \lambda_1\|\mathbf{E}^{(v)}\|_{2,1} + \frac{\mu}{2}\|\mathbf{M}_3^{(v)} - \mathbf{E}^{(v)}\|_{\mathrm{F}}^2,$$

where $\mathbf{M}_3^{(v)} = \mathbf{P}^{(v)}\mathbf{X}^{(v)} - \mathbf{A}\mathbf{Q}^{(v)} + \mathbf{J}^{(v)}/\mu$. Following Liu et al. (2012), its column-wise solution is

$$\mathbf{E}^{(v)}(:,i) = \max\left(1 - \frac{\lambda_1}{\mu\|\mathbf{M}_3^{(v)}(:,i)\|_2}, 0\right)\mathbf{M}_3^{(v)}(:,i). \tag{6}$$

• **Solving the $\mathcal{H}$ subproblem**

With the other variables fixed, the subproblem with respect to $\mathcal{H}$ is given by

$$\begin{aligned}\min_{\mathcal{H}} \quad & \frac{\theta}{2}\left\|\mathcal{H} - \mathcal{F} * \mathcal{Z}^{\mathrm{T}} - \mathcal{E} + \frac{\mathcal{Y}_1}{\theta}\right\|_{\mathrm{F}}^2 \\ & + \frac{\gamma}{2}\sum_{k}^{\Gamma}\left\|\mathcal{G}_k - \nabla_k(\mathcal{H}) + \frac{\mathcal{Y}_{2,k}}{\gamma}\right\|_{\mathrm{F}}^2 + \frac{\gamma}{2}\left\|\mathcal{H} - \mathcal{Q} + \frac{\mathcal{Y}_3}{\gamma}\right\|_{\mathrm{F}}^2.\end{aligned} \tag{7}$$

Taking the derivative of equation 7 with respect to $\mathcal{H}$ and setting it to zero yields

$$\begin{aligned}\left[(\theta+\gamma)\mathcal{I} + \gamma\sum_{k}^{\Gamma}\nabla_k^{\mathrm{T}}\nabla_k\right]\mathcal{H} = \mathcal{R}_H, \\ \mathcal{R}_H = \theta\left(\mathcal{F} * \mathcal{Z}^{\mathrm{T}} + \mathcal{E} - \frac{\mathcal{Y}_1}{\theta}\right) + \gamma\mathcal{Q} - \mathcal{Y}_3 \\ + \gamma\sum_{k}^{\Gamma}\nabla_k^{\mathrm{T}}\left(\mathcal{G}_k + \frac{\mathcal{Y}_{2,k}}{\gamma}\right).\end{aligned} \tag{8}$$

Under the periodic boundary condition, the finite-difference operators can be diagonalized in the Fourier domain. Therefore, following the FFT-based treatment of tensor total variation, we solve equation 8 efficiently in the frequency domain. Define

$$\begin{aligned}\mathcal{B}_H = \theta\left(\mathcal{F} * \mathcal{Z}^{\mathrm{T}} + \mathcal{E} - \frac{\mathcal{Y}_1}{\theta}\right) + \gamma\mathcal{Q} - \mathcal{Y}_3, \\ \widehat{\mathcal{C}}_H = \gamma\sum_{k}^{\Gamma}\mathrm{conj}(\mathrm{fftn}(\mathcal{D}_k)) \circ \mathrm{fftn}\left(\mathcal{G}_k + \frac{\mathcal{Y}_{2,k}}{\gamma}\right),\end{aligned}$$

where $\mathcal{D}_k$ denotes the finite-difference kernel associated with $\nabla_k(\cdot)$, and $\circ$ represents element-wise multiplication. Consequently, the optimal solution of $\mathcal{H}$ can be obtained as

$$\mathcal{H} = \text{ifftn}\left(\frac{\text{fftn}(\mathcal{B}_H) + \widehat{\mathcal{C}}_H}{\theta + \gamma + \gamma\sum_k^{\Gamma} |\text{fftn}(\mathcal{D}_k)|^2}\right), \tag{9}$$

where $\text{conj}(\cdot)$ denotes complex conjugation and the division is performed element-wise.

• **Solving $\mathcal{F}$ subproblem**

Define $\mathcal{R}_F = \mathcal{H} - \mathcal{E} + \mathcal{Y}_1/\theta$. The $\mathcal{F}$ subproblem is

$$\min_{\mathcal{F}\geq 0} \|\mathcal{R}_F - \mathcal{F} * \mathcal{Z}^{\mathrm{T}}\|_{\mathrm{F}}^2.$$

Using $\mathcal{Z}^{\mathrm{T}} * \mathcal{Z} = \mathcal{I}_c$, it is equivalent, up to a constant, to

$$\min_{\mathcal{F}\geq 0} \|\mathcal{F} - \mathcal{R}_F * \mathcal{Z}\|_{\mathrm{F}}^2.$$

Consequently, its closed-form solution is

$$\mathcal{F} = [\mathcal{R}_F * \mathcal{Z}]_+, \tag{10}$$

where $[x]_+ = \max(x, 0)$ denotes the element-wise projection onto the nonnegative orthant.

• **Solving the $\mathcal{Z}$ subproblem**

With the other variables fixed, the terms associated with $\mathcal{Z}$ lead to the following orthogonality-constrained subproblem:

$$\min_{\mathcal{Z}^{\mathrm{T}} * \mathcal{Z} = \mathcal{I}_c} \left\|\mathcal{R}_Z - \mathcal{F} * \mathcal{Z}^{\mathrm{T}}\right\|_{\mathrm{F}}^2, \tag{11}$$

where $\mathcal{R}_Z = \mathcal{H} - \mathcal{E} + \mathcal{Y}_1/\theta$. By applying the FFT along the third mode, equation 11 can be equivalently transformed into

$$\begin{aligned} \min_{\overline{\mathcal{Z}}} \quad & \sum_{v=1}^{V} \left\|\overline{\mathcal{R}}_Z^{(v)} - \overline{\mathcal{F}}^{(v)} (\overline{\mathcal{Z}}^{(v)})^{\mathrm{H}}\right\|_{\mathrm{F}}^2 \\ \text{s.t.} \quad & (\overline{\mathcal{Z}}^{(v)})^{\mathrm{H}} \overline{\mathcal{Z}}^{(v)} = \mathbf{I}_c, \quad v = 1, \ldots, V. \end{aligned}$$

Accordingly, the above problem can be solved independently for each frontal slice in the Fourier domain. Specifically, for the v-th frontal slice, we have

$$\min_{(\overline{\mathcal{Z}}^{(v)})^{\mathrm{H}} \overline{\mathcal{Z}}^{(v)} = \mathbf{I}_c} \left\|\overline{\mathcal{R}}_Z^{(v)} - \overline{\mathcal{F}}^{(v)} (\overline{\mathcal{Z}}^{(v)})^{\mathrm{H}}\right\|_{\mathrm{F}}^2.$$

The above formulation is a standard orthogonal Procrustes problem. Let

$$\overline{\mathcal{C}}^{(v)} = (\overline{\mathcal{R}}_Z^{(v)})^{\mathrm{H}} \overline{\mathcal{F}}^{(v)} = \mathbf{U}_Z^{(v)} \mathbf{\Sigma}_Z^{(v)} (\mathbf{V}_Z^{(v)})^{\mathrm{H}}$$

be the singular value decomposition of $\overline{\mathcal{C}}^{(v)}$. Then, the optimal solution for the v-th frontal slice is given by

$$\overline{\mathcal{Z}}^{(v)} = \mathbf{U}_Z^{(v)} (\mathbf{V}_Z^{(v)})^{\mathrm{H}}, \quad v = 1, \ldots, V.$$

Finally, the optimal $\mathcal{Z}$ is recovered by applying the inverse FFT along the third mode:

$$\mathcal{Z} = \text{ifft}(\overline{\mathcal{Z}}, [], 3). \tag{12}$$

• **Solving $\mathcal{E}$ subproblem**

Let $\mathcal{R}_{\mathcal{E}} = \mathcal{H} - \mathcal{F} * \mathcal{Z}^{\mathrm{T}} + \mathcal{Y}_1/\theta$. The corresponding subproblem is

$$\min_{\mathcal{E}} \lambda_2 \|\mathcal{E}\|_{\mathrm{F}}^2 + \frac{\theta}{2} \|\mathcal{E} - \mathcal{R}_{\mathcal{E}}\|_{\mathrm{F}}^2,$$

Table 7: Description of the two additional small-scale datasets.

Dataset	Samples	Classes	Dim.
MSRC	210	7	$24, 576, 512, 256, 254$
NGs	500	5	$2000, 2000, 2000$

Taking the derivative with respect to $\mathcal{E}$ and setting it to zero gives

$$2\lambda_2\mathcal{E} + \theta(\mathcal{E} - \mathcal{R}_{\mathcal{E}}) = 0.$$

Therefore, the closed-form solution is

$$\mathcal{E} = \frac{\theta}{\theta + 2\lambda_2}\mathcal{R}_{\mathcal{E}}. \tag{13}$$

• Solving the $\{\mathcal{G}_k\}_{k\in\Gamma}$ subproblem

With the other variables fixed, the subproblem with respect to $\mathcal{G}_k$ for each $k \in \Gamma$ is given by

$$\min_{\mathcal{G}_k} \frac{1}{\gamma}\|\mathcal{G}_k\|_{\circledS p}^p + \frac{1}{2}\left\|\mathcal{G}_k - \left(\nabla_k(\mathcal{H}) - \frac{\mathcal{Y}_{2,k}}{\gamma}\right)\right\|_{\mathrm{F}}^2. \tag{14}$$

Problem equation 14 is a standard tensor singular-value optimization problem. According to Lemma 1 in Gao et al. (2020), its closed-form solution can be obtained via the generalized tensor singular-value thresholding operator $\mathcal{S}_\tau^p(\cdot)$. Therefore, the optimal solution of $\mathcal{G}_k$ is given by

$$\mathcal{G}_k = \mathcal{S}_{1/\gamma}^p\left(\nabla_k(\mathcal{H}) - \frac{\mathcal{Y}_{2,k}}{\gamma}\right), \quad k \in \Gamma. \tag{15}$$

• Updating multipliers and penalty parameters

After updating all primal variables, the multipliers are updated by

$$\begin{aligned}
\mathbf{J}^{(v)} &\leftarrow \mathbf{J}^{(v)} + \mu(\mathbf{P}^{(v)}\mathbf{X}^{(v)} - \mathbf{A}\mathbf{Q}^{(v)} - \mathbf{E}^{(v)}),\\
\mathcal{Y}_1 &\leftarrow \mathcal{Y}_1 + \theta(\mathcal{H} - \mathcal{F} * \mathcal{Z}^{\mathrm{T}} - \mathcal{E}),\\
\mathcal{Y}_{2,k} &\leftarrow \mathcal{Y}_{2,k} + \gamma(\mathcal{G}_k - \nabla_k(\mathcal{H})),\\
\mathcal{Y}_3 &\leftarrow \mathcal{Y}_3 + \gamma(\mathcal{H} - \mathcal{Q}).
\end{aligned} \tag{16}$$

The penalty parameters can be adaptively increased as $\mu \leftarrow \min(\rho\mu, \mu_{\max})$, $\theta \leftarrow \min(\rho\theta, \theta_{\max})$, and $\gamma \leftarrow \min(\rho\gamma, \gamma_{\max})$, where $\rho > 1$. The iteration is terminated when all four primal residuals r_{rec}, r_{ten}, r_{grad}, and r_{hq} are below the prescribed tolerance $\varepsilon = 10^{-6}$ or the maximum iteration number $T_{\max} = 200$ is reached.

D Additional Experimental Results

This appendix reports supplementary experimental results omitted from the main text for compactness. In addition to the results on two small-scale datasets, we provide the complete five-metric clustering results for all benchmark datasets, as well as additional parameter, convergence, and visualization results.

Table 12 reports the running times on all eight benchmark datasets. Although **LS-UATNOF** is not always the fastest method, it achieves a favorable balance between clustering accuracy and computational cost, especially considering its joint unified-anchor learning and tensor regularization components.

Figure 4 shows the influence of the anchor number m. Overall, the clustering performance remains relatively stable when a moderate number of anchors is used, and further increasing m does not necessarily yield additional improvements.

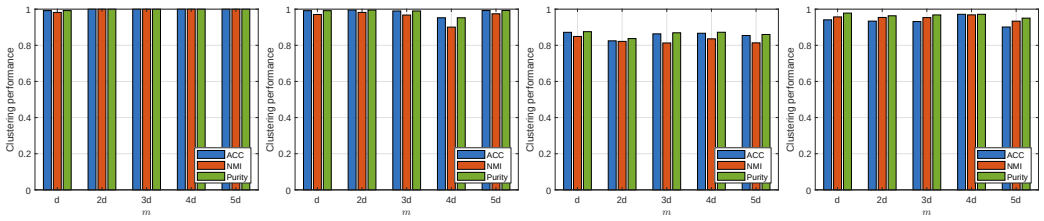
Figure 5 shows the influence of the Schatten parameter p. The results indicate that excessively small values of p may weaken the learned representation on some datasets, whereas moderate-to-large values generally lead to stable clustering performance.

Table 8: Complete clustering results on MSRC and NGs.

Method	MSRC					NGs				
	ACC	NMI	Purity	F-score	ARI	ACC	NMI	Purity	F-score	ARI
t-SVD-MSC(IJCV'18)	0.998	0.992	0.998	0.998	0.996	0.923	0.918	0.923	0.913	0.909
WTNNM(AAAI'20)	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.949	0.943	0.949	0.927	0.918
EOMSC-CA(AAAI'22)	0.671	0.560	0.671	0.547	0.469	0.630	0.478	0.648	0.542	0.413
Orth-NTF(NeurIPS'23)	0.990	0.978	0.990	0.981	0.978	0.963	0.955	0.963	0.921	0.909
RCAGL(TKDE'24)	0.800	0.688	0.800	0.675	0.623	0.666	0.567	0.812	0.610	0.501
DSTL(TMM'24)	0.470	0.435	0.495	0.395	0.272	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
ESTMC(TPAMI'25)	0.967	0.935	0.967	0.935	0.924	0.996	0.988	0.996	0.992	0.990
TLRLF4MVC(TPAMI'25)	0.852	0.889	0.852	0.825	0.794	0.964	0.895	0.964	0.928	0.911
LMTC(CVPR'25)	0.862	0.787	0.862	0.774	0.737	0.946	0.850	0.946	0.896	0.871
ARIA(TKDE'26)	0.828	0.759	0.828	0.742	0.699	0.992	0.972	0.992	0.984	0.980
LS-UATNOF	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.998	0.993	0.998	0.996	0.995

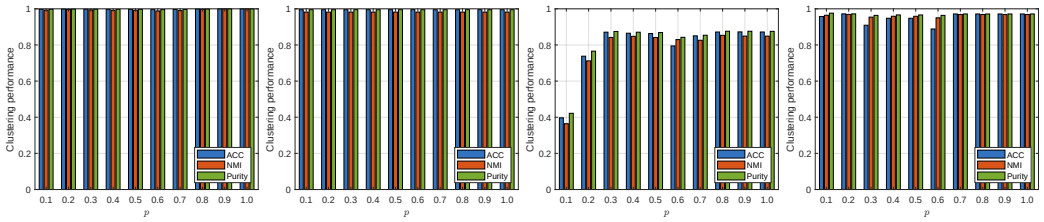
Table 9: Complete clustering results on HW and BDGP.

Method	HW					BDGP				
	ACC	NMI	Purity	F-score	ARI	ACC	NMI	Purity	F-score	ARI
t-SVD-MSC(IJCV'18)	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.432	0.280	0.447	0.356	0.278
WTNNM(AAAI'20)	0.945	0.928	0.945	0.903	0.899	0.496	0.373	0.514	0.404	0.296
EOMSC-CA(AAAI'22)	0.762	0.825	0.764	0.711	0.670	0.420	0.145	0.420	0.309	0.097
Orth-NTF(NeurIPS'23)	0.985	0.969	0.985	0.970	0.966	0.465	0.318	0.465	0.354	0.189
RCAGL(TKDE'24)	0.877	0.806	0.877	0.780	0.756	0.503	0.303	0.622	0.400	0.209
DSTL(TMM'24)	0.970	0.960	0.973	0.955	0.950	0.986	0.957	0.986	0.973	0.966
ESTMC(TPAMI'25)	0.998	0.995	0.998	0.996	0.996	0.984	0.950	0.984	0.967	0.959
TLRLF4MVC(TPAMI'25)	0.883	0.900	0.883	0.867	0.852	0.276	0.025	0.276	0.250	0.015
LMTC(CVPR'25)	0.942	0.873	0.942	0.887	0.875	0.541	0.336	0.567	0.423	0.276
ARIA(TKDE'26)	0.896	0.814	0.896	0.809	0.788	0.558	0.320	0.561	0.435	0.281
LS-UATNOF	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**0.995**	**0.982**	**0.995**	**0.990**	**0.988**



(a) HW (b) BDGP (c) CCV (d) Animal

Figure 4: Additional parameter sensitivity analysis of the anchor number m.



(a) HW (b) BDGP (c) CCV (d) Animal

Figure 5: Additional parameter sensitivity analysis of the Schatten parameter p.

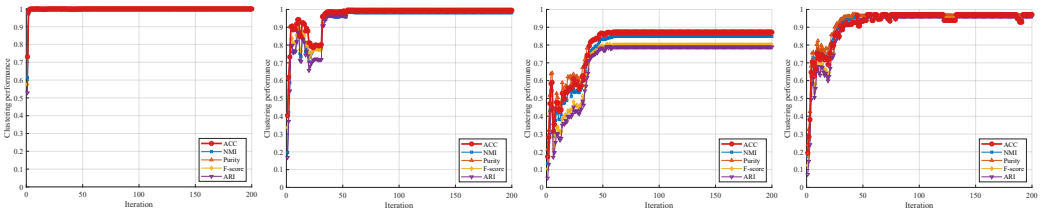
Figure 6 presents the evolution of clustering metrics during optimization. The curves show that ACC, NMI, Purity, F-score, and ARI rapidly improve and then become stable, indicating that reliable clustering structures are obtained within a limited number of iterations.

Table 10: Complete clustering results on CCV and Animal.

Method	CCV					Animal				
	ACC	NMI	Purity	F-score	ARI	ACC	NMI	Purity	F-score	ARI
t-SVD-MSC(IJCV'18)	0.478	0.413	0.498	0.334	0.296	0.167	0.125	0.200	0.104	0.045
WTNNM(AAAI'20)	0.476	0.502	0.468	0.433	0.392	0.231	0.194	0.225	0.188	0.161
EOMSC-CA(AAAI'22)	0.243	0.187	0.266	0.139	0.085	0.188	0.152	0.206	0.138	0.079
Orth-NTF(NeurIPS'23)	0.565	0.527	0.622	0.601	0.557	0.573	0.664	0.602	0.611	0.623
RCAGL(TKDE'24)	0.244	0.190	0.262	0.137	0.082	0.160	0.135	0.230	0.121	0.065
DSTL(TMM'24)	0.706	0.750	0.732	0.649	0.627	0.764	0.825	0.808	0.695	0.677
ESTMC(TPAMI'25)	0.746	0.736	0.755	0.660	0.640	0.922	0.916	0.934	0.890	0.844
TLRLF4MVC(TPAMI'25)	0.202	0.160	0.233	0.117	0.061	0.165	0.138	0.200	0.112	0.052
LMTC(CVPR'25)	0.237	0.190	0.272	0.137	0.086	0.177	0.137	0.210	0.105	0.052
ARIA(TKDE'26)	0.203	0.175	0.237	0.121	0.056	0.171	0.136	0.209	0.103	0.050
LS-UATNOF	**0.872**	**0.849**	**0.875**	**0.801**	**0.789**	**0.972**	**0.968**	**0.972**	**0.964**	**0.961**

Table 11: Complete clustering results on NUSWIDEOBJ and NoisyMNIST. OM denotes out of memory.

Method	NUSWIDEOBJ					NoisyMNIST				
	ACC	NMI	Purity	F-score	ARI	ACC	NMI	Purity	F-score	ARI
t-SVD-MSC(IJCV'18)	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
WTNNM(AAAI'20)	OM	OM	OM	OM	OM	OM	OM	OM	OM	OM
EOMSC-CA(AAAI'22)	0.195	0.129	0.237	0.137	0.067	0.547	0.470	0.562	0.451	0.384
Orth-NTF(NeurIPS'23)	0.178	0.146	0.205	0.157	**0.122**	0.701	0.729	0.747	0.728	0.713
RCAGL(TKDE'24)	0.192	0.125	0.361	0.134	0.058	0.520	0.463	0.560	0.441	0.376
DSTL(TMM'24)	0.130	0.175	0.221	0.118	0.063	0.871	0.816	0.901	0.822	0.779
ESTMC(TPAMI'25)	0.154	0.130	0.246	0.094	0.053	0.987	0.971	0.987	0.975	0.972
TLRLF4MVC(TPAMI'25)	0.168	0.190	0.304	0.126	0.086	OM	OM	OM	OM	OM
LMTC(CVPR'25)	0.123	0.100	0.216	0.078	0.036	0.606	0.593	0.657	0.537	0.484
ARIA(TKDE'26)	0.136	0.111	0.228	0.081	0.040	0.516	0.464	0.561	0.419	0.351
LS-UATNOF	**0.274**	**0.266**	**0.384**	**0.158**	0.119	**0.998**	**0.995**	**0.998**	**0.997**	**0.997**



(a) HW (b) BDGP (c) CCV (d) Animal

Figure 6: Additional clustering-performance curves of **LS-UATNOF** over iterations.

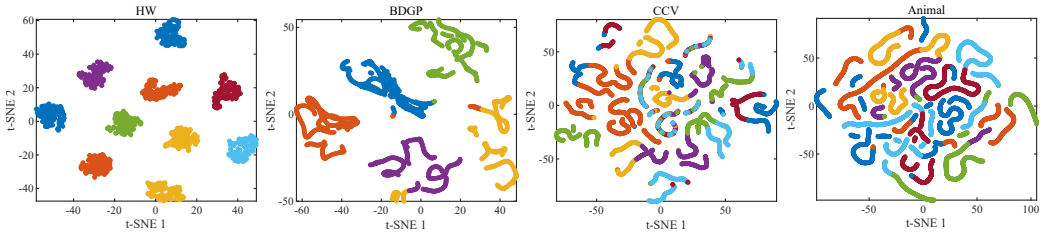
D.1 Visualization Results

As supplementary qualitative evidence, we visualize the final sample representations learned by **LS-UATNOF** on four representative datasets. Specifically, the representation matrix obtained by unfolding the tensor factor $\mathcal{Z}$ is projected onto a two-dimensional space using t-SNE. As shown in Fig. 7, samples from different classes form clearly separated clusters on HW and BDGP. Although CCV and Animal exhibit more complicated distributions and partial overlap, samples belonging to the same class still show evident local aggregation.

We further visualize the sample affinity matrices derived from the learned representations. Let $\widehat{\mathbf{R}}$ denote the row-normalized unfolded representation matrix; its affinity matrix is constructed as $\mathbf{S} = |\widehat{\mathbf{R}}\widehat{\mathbf{R}}^{\mathrm{T}}|$, with samples reordered according to their ground-truth labels for visualization. As illustrated in Fig. 8, the matrices exhibit visible block-diagonal structures, implying high within-class similarity and low between-class correlation.

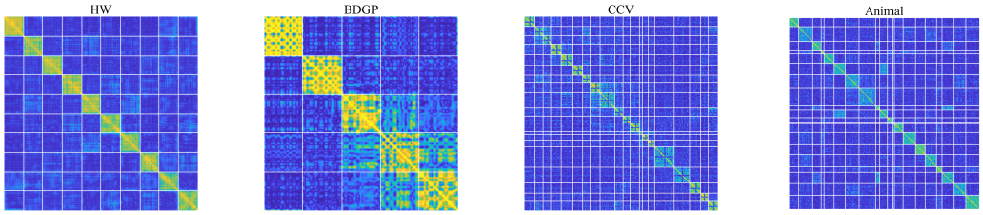
Table 12: Running time (seconds) of different methods on the eight benchmark datasets. OM denotes out of memory.

Method	MSRC	NGs	HW	BDGP	CCV	Animal	NUSWIDEOBJ	NoisyMNIST
t-SVD-MSC(IJCV'18)	1.06	5.37	54.93	59.63	2547.91	13670.28	OM	OM
WTNNM(AAAI'20)	2.58	13.15	132.52	191.41	3068.47	18579.02	OM	OM
EOMSC-CA(AAAI'22)	0.59	21.57	7.01	3.58	20.15	20.88	116.83	76.20
Orth-NTF(NeurIPS'23)	0.67	0.94	9.90	10.07	40.17	96.55	321.54	97.58
RCAGL(TKDE'24)	0.14	25.38	0.76	3.96	9.82	34.69	120.68	23.70
DSTL(TMM'24)	**0.07**	0.21	**0.40**	0.71	7.72	**20.37**	189.27	78.63
ESTMC(TPAMI'25)	0.35	3.21	2.69	4.06	5.24	137.51	112.04	129.28
TLRLF4MVC(TPAMI'25)	0.68	2.27	21.84	8.61	22.30	32.97	**62.57**	OM
LMTC(CVPR'25)	0.48	**0.16**	0.52	**0.49**	5.54	24.20	87.46	**23.54**
ARIA(TKDE'26)	0.27	0.95	2.18	1.75	**2.05**	63.32	70.29	40.52
LS-UATNOF	0.51	1.18	5.78	2.30	14.96	107.80	334.25	104.36



(a) HW (b) BDGP (c) CCV (d) Animal

Figure 7: t-SNE visualization of the representations learned by **LS-UATNOF** on four benchmark datasets. Different colors denote different ground-truth classes.



(a) HW (b) BDGP (c) CCV (d) Animal

Figure 8: Visualization of the affinity matrices derived from the representations learned by **LS-UATNOF**. Samples are reordered according to their ground-truth labels.