# Multi-View Bipartite Graph Clustering With Coupled Noisy Feature Filter

Liang Li, Junpu Zhang, Siwei Wang, Xinwang Liu, *Senior Member, IEEE*, Kenli Li, *Senior Member, IEEE*, and Keqin Li, *Fellow, IEEE*

***Abstract*—Unsupervised bipartite graph learning has been a hot topic in multi-view clustering, to tackle the restricted scalability issue of traditional full graph clustering in large-scale applications. However, the existing bipartite graph clustering paradigm pays little attention to the adverse impact of noisy features on learning process. To further facilitate this part of research, apart from simply reweighting features to depress the noisy ones, we take the first step towards analyzing the induced adverse impact via theoretical and experimental investigations. One crucial finding in this article is that the existence of noisy features will incur "anchor shift" phenomenon, which deviates from the potential representations of anchors and then degrades performance. To this end, we propose a coupled noisy feature filter mechanism with automatically finding feature importance to remedy the anchor shift issue in this article. Apart from leveraging features, we theoretically analyze the bounds of proposed feature-adaptive bipartite graph's fuzzy membership. Specifically, distinguishing features' discrimination will increase the fuzzy membership to achieve soft partitions against the potential inaccurate absolute relationships. With the afore-mentioned merits, our proposed multi-view bipartite graph clustering with coupled noisy feature filter model (MVBGC-NFF) provides novel and interesting insights on the feature level of anchor shift. The effectiveness and efficiency of MVBGC-NFF are demonstrated on synthetic and real-world datasets with improved clustering performance, increasing fuzzy membership, and filtering noisy features. The code is available on https://github.com/liliangnudt/MVBGC-NFF.**

***Index Terms*—Anchor shift, bipartite graph learning, multi-view clustering, noisy features.**

## I. Introduction

GRAPH learning has been a hot topic in machine learning, owing to promising capacity in discovering complex structures [1], [2], [3], [4], [5]. Graph clustering has attracted extensive attention and widely applied in handling unsupervised tasks [6], [7], [8], [9], [10]. To cope with multiple sources or multiple views real-world data, e.g., the features of an image can be extracted and described by Generalized Search Trees (GIST), Pyramid Histogram of Gradients (PHOG), and Local Binary Patterns (LBP), multi-view graph clustering (MVGC) evolves into an important branch of multi-view clustering (MVC), widely used for data mining and knowledge discovery [11], [12], [13], [14], [15], [16], [17].

Despite achieving great success, traditional MVGC [18], [19], [20] relies on building full graphs, i.e., computing fully-connected graph, requiring at least cubic computational complexity $\mathcal{O}(n^3)$ and quadratic space complexity $\mathcal{O}(n^2)$, which greatly restricts the scalability in large-scale tasks [21], [22], [23], [24], [25]. To avoid building pair-wise similarity of full graph, multi-view bipartite graph clustering (MVBGC) is proposed by constructing bipartite graph, i.e., merely building the relationship between representative anchors and samples to recover the full graph [26], [27], [28], [29], [30]. In this way, the computational complexity is reduced to $\mathcal{O}(n)$ and space complexity is reduced to $\mathcal{O}(nmv)$, where $n$, $m$, and $v$ denote the number of samples, anchors, and views, respectively. The linear complexities make it suitable for large-scale applications [31], [32], [33], [34], [35]. Naturally, the MVBGC paradigm requires a set of anchors to represent data distribution, which means the anchor plays an important role in constructing high-quality bipartite graphs. Currently, there are two anchor-selecting strategies, one is to utilize $k$-means [28], [36], random sampling [26] or heuristic methods [29], [37], [38] to select anchors, the other is to learn anchors via optimization [39], [40], [41]. Considering the randomness and inflexibility of sampling strategy, our model adopts the learning strategy to generate anchors.

However, the conventional MVBGC paradigm coupled anchor learning still encounters two deficiencies: 1) most variants focus on learning representative anchors in sample space while neglecting features' discrimination in feature space. 2) Current MVBGC paradigm pays little attention to the adverse impact of noisy features on bipartite graph clustering, making this part deserves further exploration. Fig. 1 shows a synthetic example. Fig. 1(a) shows the first two clean features, Fig. 1(b) shows

Manuscript received 15 September 2022; revised 14 February 2023; accepted 8 April 2023. Date of publication 27 April 2023; date of current version 8 November 2023. This work was supported in part by the National Key Research and Development Program of China under Grants 2020AAA0107703 and 2020AAA0107100, and in part by the National Natural Science Foundation of China under Grants 61922088, 61773392, 61976196, and 62276271. Recommended for acceptance by L. Zou. *(Liang Li and Junpu Zhang contributed equally to this work.) (Corresponding author: Xinwang Liu.)*

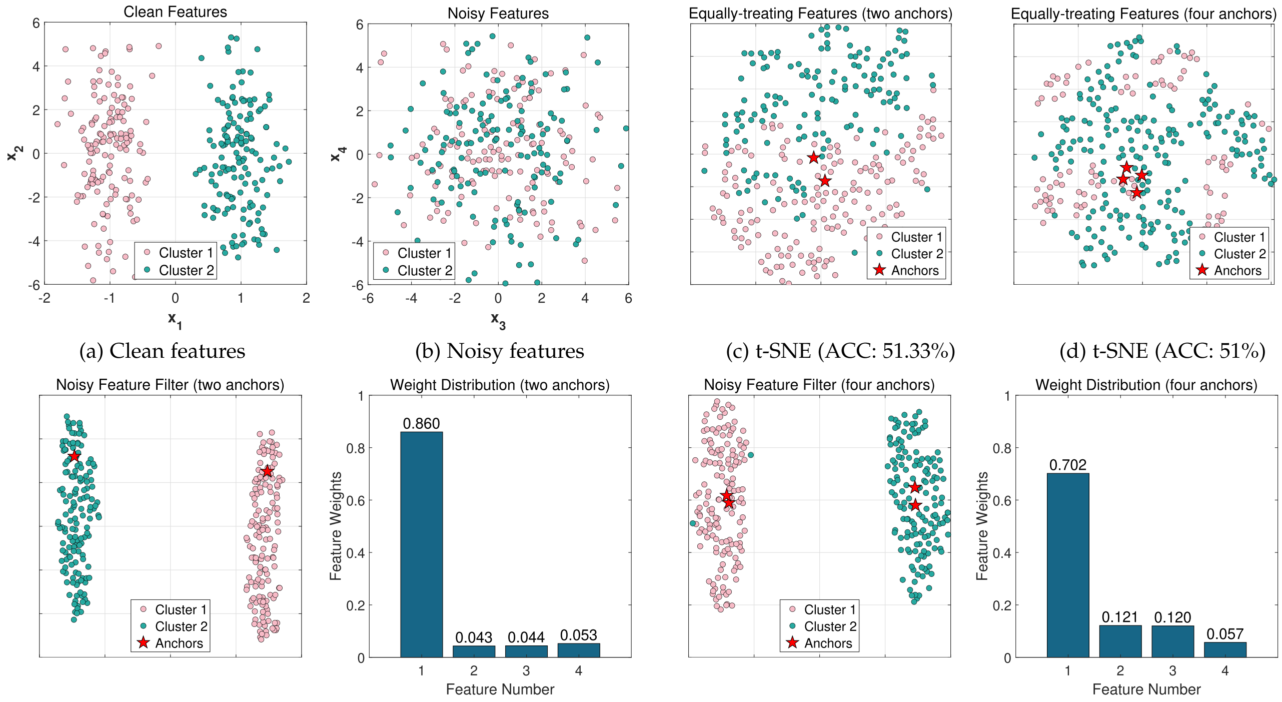
Liang Li, Junpu Zhang, Siwei Wang, and Xinwang Liu are with the School of Computer, National University of Defense Technology, Changsha, Hunan 410073, China (e-mail: liliang1037@gmail.com; zhangjunpu@nudt.edu.cn; wangsiwei13@nudt.edu.cn; xinwangliu@nudt.edu.cn).

Kenli Li is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, Hunan 410073, China, and also with the Supercomputing and Cloud Computing Institute, Hunan University, Changsha, Hunan 410073, China (e-mail: lkl@hnu.edu.cn).

Keqin Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TKDE.2023.3268215, provided by the authors.

Digital Object Identifier 10.1109/TKDE.2023.3268215



(a) Clean features

(b) Noisy features

(c) t-SNE (ACC: 51.33%)

(d) t-SNE (ACC: 51%)

(e) t-SNE (ACC: 100%)

(f) Feature weights

(g) t-SNE (ACC: 99.33%)

(h) Feature weights

Fig. 1. Visualization of the “anchor shift” phenomenon incurred by noisy features on a four-dimensional synthetic dataset. Simply enlarging anchor numbers is an ineffective solution, while depressing the noisy ones can alleviate this issue.

the rest noisy features, and Fig. 1(c) depicts t-SNE visualization (two anchors) [42] by equally treating features. Due to noisy disturbance, the learned anchors deviate from the intuitive centroid of clusters. Intuitively, enlarging anchor numbers can remedy the poor performance. However, as depicted in Fig. 1(d), simply enlarging anchor numbers (four anchors) on the sample level without depressing noisy features is not an effective solution. We summarize such a phenomenon as the “feature-induced anchor shift” phenomenon. Clearly, these anchors with poor discrimination induce inaccurate bipartite graph representation, and consequently ruin clustering performance. Similar results are further visualized on real-world datasets, illustrating the generality of “anchor shift” phenomenon in real-world applications.

To alleviate the adverse “anchor shift” phenomenon incurred by noisy features, an intuitive idea is to pay varing attention to features, e.g., when we are watching an image, the obvious parts will be gained more attention while the unimpressive background will be omitted. However, quantitatively measuring their discrimination is not easy, since acquiring prior knowledge is challenging in unsupervised scenarios. Alternatively, learning their importance via optimization is a reasonable strategy. Specifically, we propose a noisy feature filter to depress the noisy ones in a reweighting manner, the noisy features are filtered by imposing small or zero weights, while the discriminative ones are preserved by imposing large weights. Naturally, from Fig. 1(e), (f), and (g), one crucial empirical finding is that depressing noisy features can alleviate the “anchor shift” issue. Although similar feature reweighting or selection strategies have been applied in MVC [43], [44], [45] or MVBGC [36], [46], most works rarely dig deep into these adverse impacts.

Besides, another theoretical contribution is that we mathematically analyze the lower and upper bounds of the proposed bipartite graph’s fuzzy membership. A corollary is that reweighting features will increase the bipartite graph’s fuzzy membership. Compared to equally treating features that commonly incur sparse representation, increasing the fuzzy membership achieves soft partitions rather than hard clustering, making it fully explores entire anchors’ representation. Similar to the rationale of fuzzy $k$-means, such manner plays a role in remedying the potential inaccurate absolute relationships. We provide rigorous theoretical analysis and experimental validation to corroborate our findings.

The contributions are summarized as follows:

1) We take the first step towards analyzing and experimentally visualizing the adverse impact of noisy features on bipartite graph clustering, i.e., the “anchor shift” phenomenon, and propose a noisy feature filter mechanism to depress the noisy ones, which can effectively alleviate such issue.
2) Another theoretical contribution is that we mathematically analyze the bounds of bipartite graph’s fuzzy membership. A corollary is that distinguishing features’ discrimination will increase the fuzzy membership, which achieves soft partitions to remedy potential inaccurate absolute representation.
3) Extensive experiments and rigorous theoretical explanations demonstrate our theoretical findings on feature level,

TABLE I
MAIN NOTATIONS

| Notation | Explanation |
|---|---|
| $n$,$v$,$k$ | Number of samples, views, and clusters |
| $d_p$ | Feature dimension for the $p$-th view |
| $m$ | Number of anchors |
| $D$ | The minimal feature dimension |
| $\mathcal{L}$ | The loss function |
| $f$ | The function for the bound of $\|\mathbf{z}\|_0$ |
| $\boldsymbol{\alpha} \in \mathbb{R}^{v\times 1}$ | View weights |
| $\mathbf{X}_p \in \mathbb{R}^{d_p\times n}$ | Data matrix for the $p$-th view |
| $\boldsymbol{\Omega}_p \in \mathbb{R}^{d_p\times d_p}$ | Feature reweighting matrix for the $p$-th view |
| $\boldsymbol{\omega}_p \in \mathbb{R}^{d_p\times 1}$ | Feature reweighting vector for the $p$-th view |
| $\mathbf{A}_p \in \mathbb{R}^{d_p\times m}$ | Anchor matrix for the $p$-th view |
| $\mathbf{S}_p \in \mathbb{R}^{n\times n}$ | Full graph for the $p$-th view |
| $\mathbf{S} \in \mathbb{R}^{n\times n}$ | Consensus full graph |
| $\mathbf{Z}_p \in \mathbb{R}^{m\times n}$ | Bipartite graph for the $p$-th view |
| $\mathbf{Z} \in \mathbb{R}^{m\times n}$ | Consensus bipartite graph |
| $\mathbf{z}^* \in \mathbb{R}^{m\times 1}$ | $\mathbf{Z}_{:,j}$ learned by equally treating features |
| $\hat{\mathbf{z}}^* \in \mathbb{R}^{m\times 1}$ | $\hat{\mathbf{Z}}_{:,j}$ learned by proposed noisy feature filter |

which further facilitates this part of research in bipartite graph clustering.

## II. Related Work

This section briefly reviews several important research mostly related to our work. Table I records the main notations throughout this work.

### A. Multi-View Graph Clustering

Given multiple view data $\{\mathbf{X}_p\}_{p=1}^{v} \in \mathbb{R}^{d_p\times n}$ drawn from $k$ clusters, a representative paradigm of multi-view subspace clustering is formulated as

$$\min_{\mathbf{S}_p,\mathbf{S}} \sum_{p=1}^{v} \underbrace{\|\mathbf{X}_p - \mathbf{X}_p\mathbf{S}_p\|_F^2}_{Full\ Graph\ Learning} + \underbrace{\lambda\mathcal{L}(\mathbf{S}_p,\, \mathbf{S})}_{Full\ Graph\ Fusion},$$
$$\text{s.t.} \begin{cases} \operatorname{diag}(\mathbf{S}_{\mathrm{p}}) = \mathbf{0},\ \mathbf{S}_{\mathrm{p}}^{\top}\mathbf{1}_{\mathrm{n}} = \mathbf{1}_{\mathrm{n}},\ \mathbf{S}_{\mathrm{p}} \geq \mathbf{0}, \\ \operatorname{diag}(\mathbf{S}) = \mathbf{0},\ \mathbf{S}^{\top}\mathbf{1}_n = \mathbf{1}_n,\ \mathbf{S} \geq \mathbf{0}, \end{cases} \tag{1}$$

where the first term denotes graph learning for the $p$-th view, the second term denotes graph fusion across $\{\mathbf{S}_p\}_{p=1}^{v}$ to learn a consensus $\mathbf{S}$, $\lambda$ is a balanced hyper-parameter, $\mathcal{L}$ is the loss function. The learned partition can be computed by conducting spectral clustering on $\mathbf{S}$ [20], [21].

Many variants are developed along with this framework, aiming to explore diversity and consistent representations [21], [31], [47] or introduce various constraints [21], [23], [48]. However, this paradigm cannot avoid building a full graph, requiring at least cubic computational complexity $\mathcal{O}(n^3)$ and quadratic space complexity $\mathcal{O}(n^2)$, which greatly limits scalability in large-scale tasks.

### B. Multi-View Bipartite Graph Clustering

To alleviate the high computational and space complexity, multi-view bipartite graph clustering is proposed by merely building the relationship between representative anchors and samples, i.e., bipartite graph $\mathbf{Z}_p$, instead of constructing fully-connected graphs $\mathbf{S}_p$. A typical paradigm is formulated as

$$\min_{\mathbf{Z}_p,\mathbf{Z}} \sum_{p=1}^{v} \underbrace{\|\mathbf{X}_p - \mathbf{A}_p\mathbf{Z}_p\|_F^2}_{Bipartite\ Graph\ Learning} + \underbrace{\lambda\mathcal{L}(\mathbf{Z}_p,\, \mathbf{Z})}_{Bipartite\ Graph\ Fusion},$$
$$\text{s.t.} \begin{cases} \mathbf{Z}_p^{\top}\mathbf{1}_m = \mathbf{1}_n,\ \mathbf{Z}_p \geq \mathbf{0}, \\ \mathbf{Z}^{\top}\mathbf{1}_m = \mathbf{1}_n,\ \mathbf{Z} \geq \mathbf{0}, \end{cases} \tag{2}$$

where $\mathbf{A}_p$ is the anchor matrix for the $p$-th view, $\mathbf{Z}_p$ is the corresponding bipartite graph, and $\mathbf{Z}$ is the consensus one. The first term learns view-related bipartite graphs, and the second term achieves fusion. The learned partition can be computed by performing Singular Value Decomposition (SVD) on $\mathbf{Z}$ [28], [41]. In this way, the time and space complexities are reduced to $\mathcal{O}(n)$ and $\mathcal{O}(nmv)$, respectively.

Naturally, anchors play an essential role in learning high-quality bipartite graphs. Huang et al. [49] utilized random sampling to generate anchors. Kang et al. [28] and Yan et al. [36] generated anchors from $k$-means. Li et al. [29] and Huang et al. [38] utilized heuristic methods to select anchors. Note that sampling induces randomness and these anchors remain fixed during optimization. Liu et al. [41], Sun et al. [39], and Wang et al. [40], incorporated anchor selection into optimization, improving flexibility and scalability.

## III. Methodology

To alleviate the adverse "anchor shift" phenomenon incurred by noisy features, this section proposes an effective noisy feature filter mechanism. Moreover, we elaborately analyze the bounds of proposed bipartite graph's fuzzy membership, and a corollary is that distinguishing features' discrimination induce increasing of the fuzzy membership.

### A. The Proposed Formula

Although the existing MVBGC paradigm coupled anchor learning exhibits promising performance in real-world applications, it involves regularization that requires extra hyper-parameter tuning. Following the principle that multiple views share latent consensus clustering representation and keep view-related diversity, the conventional MVBGC paradigm can be compacted into the following,

$$\min_{\mathbf{A}_p,\mathbf{Z}} \sum_{p=1}^{v} \|\mathbf{X}_p - \mathbf{A}_p\mathbf{Z}\|_F^2,$$
$$\text{s.t.} \begin{cases} \mathbf{A}_p^{\top}\mathbf{A}_p = \mathbf{I}_m, \\ \mathbf{Z}^{\top}\mathbf{1}_m = \mathbf{1}_n,\ \mathbf{Z} \geq 0, \end{cases} \tag{3}$$

where $\mathbf{A}_p$ denotes view-related anchor matrix, $\mathbf{Z}$ denotes the consensus bipartite graph. Compared to the conventional MVBGC paradigm that separates bipartite graph learning and fusion, (3) unifies them into a compact form, which avoids to balance the regularization.

However, the conventional MVBGC paradigm and the above formulation still suffer two deficiencies:

1) Most variants focus on learning representative anchors while neglecting features' discrimination, i.e., equally treating all the features, conflicting the intuition that informative or expressive features should be emphasized while the irrelevant or noisy ones are filtered.
2) Existing paradigm pays little attention to the potential adverse impact of noisy features. Since the quality of bipartite graph is sensitive to anchors' representation, and equally treating features will bring the interference from irrelevant or noisy features, the induced impact on anchors' representation deserves further exploration.

The above analysis motivates us to facilitate this line of research, this article takes the first step toward investigating the adverse impact of noisy features.

*Remark 1.* One crucial experimental finding is that noisy features will incur the adverse "anchor shift" phenomenon, and simply increasing anchor numbers is an ineffective solution. Depressing the noisy ones can effectively alleviate such undesired issue.

For simplicity, the compact *MVBGC* with **E**qually **T**reating **F**eatures manner in (3) is termed as MVBGC-ETF.

Fig. 1(c) depicts the t-SNE visualization of MVBGC-ETF. Clearly, noisy features greatly affect anchor learning, the learned anchors exhibit limited even inaccurate representation, which directly induces mistaken partition with degraded clustering accuracy (51.33%) on such a simple dataset. Furthermore, as Fig. 1(d) shows, merely improving anchor numbers in sample space is an ineffective solution.

We summarize the "anchor shift" phenomenon as

> The existence of noisy features incurs the adverse "anchor shift" phenomenon, which deviates from the potential representations of anchors, inducing inaccurate bipartite representation and degraded performance.

To alleviate the undesired "anchor shift" issue, an intuition idea is to filter these noisy or weak expressive features and preserve the discriminative or expressive ones. Since directly measuring the discrimination is still a difficult issue in unsupervised scenarios, we introduce an effective noisy feature filter mechanism to adaptively re-weight their importance. In this way, the noisy features are depressed, and the discriminative ones are well preserved. Consequently, our MVGBC with coupled noisy feature filter model (termed MVBGC-NFF) is formulated as

$$\min_{\boldsymbol{\alpha},\boldsymbol{\Omega}_p,\mathbf{A}_p,\mathbf{Z}} \sum_{p=1}^{v} \alpha_p^2 \|\boldsymbol{\Omega}_p \mathbf{X}_p - \mathbf{A}_p \mathbf{Z}\|_F^2,$$
$$\text{s.t.} \begin{cases} \boldsymbol{\alpha}^\top \mathbf{1} = 1, \ \alpha_p \ge 0, \\ \boldsymbol{\Omega}_p = \operatorname{diag}(\boldsymbol{\omega}_p), \ \boldsymbol{\omega}_p^\top \mathbf{1} = 1, \ \boldsymbol{\omega}_p \ge \mathbf{0}, \\ \mathbf{A}_p^\top \mathbf{A}_p = \mathbf{I}_m, \\ \mathbf{Z}^\top \mathbf{1}_m = \mathbf{1}_n, \ \mathbf{Z} \ge 0, \end{cases} \tag{4}$$

where $\boldsymbol{\Omega}_p \in \mathbb{R}^{d_p \times d_p}$ denotes the diagonal feature reweighting matrix for the $p$-th view, $\boldsymbol{\omega}_p$ denotes the feature reweighting vector, and $\boldsymbol{\alpha}$ denotes view weight.

Fig. 1(e), (f), (g), and (h) presents the results by noisy feature filter mechanism. Obviously, depressing noisy features is a feasible solution to prevent anchor learning from suffering the "anchor shift" issue, the learned anchors exhibit stronger representation, which effectively guides the clustering process with 100% accuracy. Our noisy feature filter mechanism effectively detects the discriminative feature ($x_1$) and imposes a large weight (0.86) while filters useless ($x_2$) or noisy features ($x_3$ and $x_4$) by imposing small weights (0.043, 0.044, and 0.053), respectively. This is sufficient evidence to verify the importance of depressing noisy features.

### *B. Theoretical Contribution on Proposed Bipartite Graph's Fuzzy Membership*

*Remark 2.* One novel theoretical contribution is that we mathematically analyze the bounds of proposed bipartite graph's fuzzy membership. A corollary is that distinguishing features' discrimination will increase the fuzzy membership.

*Proposition 1.* Distinguishing features' discrimination will increase the bipartite graph's fuzzy membership. Specifically, the bounds of $\|\mathbf{z}^*\|_0$ (i.e., $\|\mathbf{Z}_{:,j}\|_0$) from equally treating manner (MVBGC-ETF) is $f(\mathrm{M}_2) \le \|\mathbf{z}^*\|_0 \le f(\mathrm{M}_1)$, and the bounds of $\|\hat{\mathbf{z}}^*\|_0$ from reweighting manner (MVBGC-NFF) is $f(\frac{\mathrm{M}_2}{D}) \le \|\hat{\mathbf{z}}^*\|_0 \le f(\frac{\mathrm{M}_1}{D})$, where $\|\mathbf{z}\|_0$ is the 0-norm of vector $\mathbf{z}$, $D = \min_p d_p$ is the minimal feature dimension, $\mathrm{M}_1 = \min_i \Delta b_i$, $\mathrm{M}_2 = \max_i \Delta b_i$, $\Delta b_i := b_{i+1} - b_i$, and $\mathbf{b}$ is a dataset-related vector defined in (6). The bound can be quantified by a unified and elegant function $f(t) = \min\{\lfloor\sqrt{\lceil \frac{2}{t} \rceil - \frac{3}{4}} + \frac{1}{2}\rfloor, m\}$.

Proposition 1 illustrates that reweighting features will increase the bipartite graph's fuzzy membership compared to the equally treating manner. Recall the "anchor shift" phenomenon incurred by noisy features shown in Fig. 1, the poor representation of anchors incurs much inaccurate relationship, making it insufficient to recover the latent clustering structures. Similar to the rationale of fuzzy $k$-means, inducing fuzzy membership plays a role in fully exploring the entire anchors' representation, which contributes to remedying the inaccurate membership during optimization.

The following part mathematically derives the bounds of the proposed feature-adaptive bipartite graph's fuzzy membership. As a reference, we firstly give the bounds of conventional equally-treating feature manner.

Considering the non-convexity, we employ alternative optimization strategy to solve one variable with others being fixed. According to $\mathbf{Z}$ optimization in Section III-C2, each column ($\mathbf{Z}_{:,j}$) is an independent Quadratic Programming (QP) problem with analytical solution in (22), we focus on $\mathbf{Z}_{:,j}$ to represent the fuzzy membership of bipartite graph $\mathbf{Z}$. Updating $\mathbf{Z}_{:,j}$ can be simplified by

$$\min_{\widetilde{\mathbf{z}}} \left\|\widetilde{\mathbf{z}} - \widetilde{\mathbf{b}}\right\|_2^2,$$
$$\text{s.t. } \widetilde{\mathbf{z}}^\top \mathbf{1}_m = 1, \widetilde{\mathbf{z}} \ge \mathbf{0}, \tag{5}$$

where $\widetilde{\mathbf{z}}$ denotes $\mathbf{Z}_{:,j}$ and $\widetilde{\mathbf{b}}$ denotes a dataset-related vector computed by other variables, i.e., $\hat{\mathbf{Z}}_{:,j}$ in (22).

Without loss of generality, we utilize 0-norm of $\widetilde{\mathbf{z}}$ to represent the fuzzy membership, which means that adjusting the element order of $\widetilde{\mathbf{z}}$ and $\widetilde{\mathbf{b}}$ will not change their sparsity. For simplicity, $\mathbf{z}$ and $\mathbf{b}$ denotes the sorted $\widetilde{\mathbf{z}}$ and $\widetilde{\mathbf{b}}$ in ascending order, respectively. (5) is converted to

$$\begin{aligned} \min_{\mathbf{z}} \ & \|\mathbf{z} - \mathbf{b}\|_2^2, \\ \text{s.t.} \ & \mathbf{z}^\top \mathbf{1}_m = 1, \mathbf{z} \geq \mathbf{0}. \end{aligned} \tag{6}$$

We define $\Delta b_i := b_{i+1} - b_i$, $\mathrm{M}_1 = \min_i \Delta b_i$, and $\mathrm{M}_2 = \max_i \Delta b_i$, where $b_i$ denotes the $i$-th smallest element of $\mathbf{b}$.

Replacing the original $\mathbf{b}$ by a specific arithmetic progression $\mathbf{b}^{(1)}$ with a common difference $\mathrm{M}_1$, Lemma 1 gives a bound of $\|\mathbf{z}\|_0$.

*Lemma 1.* Supposing an arithmetic progression $\mathbf{b}^{(1)} : b_i^{(1)} = b_m - (m-i)\mathrm{M}_1, 1 \leq i \leq m$, (6) is converted to

$$\begin{aligned} \min_{\mathbf{z}} \ & \|\mathbf{z} - \mathbf{b}^{(1)}\|_2^2, \\ \text{s.t.} \ & \mathbf{z}^\top \mathbf{1}_m = 1, \mathbf{z} \geq \mathbf{0}. \end{aligned} \tag{7}$$

The 0-norm of $\mathbf{z}$ is as follows,

$$\|\mathbf{z}^{(1)}\|_0 = \min\left\{ \left\lfloor \sqrt{\left\lceil \frac{2}{\mathrm{M}_1} \right\rceil - \frac{3}{4}} + \frac{1}{2} \right\rfloor, m \right\}, \tag{8}$$

where $\mathbf{z}^{(1)}$ is the optimal solution of (7), $\lfloor x \rfloor$ and $\lceil x \rceil$ donate "floor" and "ceil" operations imposed on $x$, respectively.

Similarly, replacing the original $\mathbf{b}$ by another arithmetic progression $\mathbf{b}^{(2)}$ with a common difference $\mathrm{M}_2$, Lemma 2 gives another bound of $\|\mathbf{z}\|_0$.

*Lemma 2.* Supposing an arithmetic progression $\mathbf{b}^{(2)} : b_i^{(2)} = b_m - (m-i)\mathrm{M}_2, 1 \leq i \leq m$, (6) is converted to

$$\begin{aligned} \min_{\mathbf{z}} \ & \|\mathbf{z} - \mathbf{b}^{(2)}\|_2^2, \\ \text{s.t.} \ & \mathbf{z}^\top \mathbf{1}_m = 1, \mathbf{z} \geq \mathbf{0}. \end{aligned} \tag{9}$$

The 0-norm of $\mathbf{z}$ is as follows

$$\|\mathbf{z}^{(2)}\|_0 = \min\left\{ \left\lfloor \sqrt{\left\lceil \frac{2}{\mathrm{M}_2} \right\rceil - \frac{3}{4}} + \frac{1}{2} \right\rfloor, m \right\}, \tag{10}$$

where $\mathbf{z}^{(2)}$ denotes the optimal solution of (9).

Lemma 1 and Lemma 2 provide the solution of $\|\mathbf{z}\|_0$ with specific arithmetic progression $\mathbf{b}^{(1)}$ and $\mathbf{b}^{(2)}$, respectively. The detailed proof is available in supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2023.3268215. Note that both (8) and (10) can be unified in a unified and elegant formulation $f$, i.e.,

$$f(t) = \min\left\{ \left\lfloor \sqrt{\left\lceil \frac{2}{t} \right\rceil - \frac{3}{4}} + \frac{1}{2} \right\rfloor, m \right\}. \tag{11}$$

Clearly, we have $f(\mathrm{M}_1) = \|\mathbf{z}^{(1)}\|_0$ and $f(\mathrm{M}_2) = \|\mathbf{z}^{(2)}\|_0$. Theorem 1 further gives the relationship of $f(\mathrm{M}_1)$, $f(\mathrm{M}_2)$, and $\|\mathbf{z}^*\|_0$, where $\mathbf{z}^*$ is the optimal solution of (6).

*Theorem 1.* $f(\mathrm{M}_1)$ and $f(\mathrm{M}_2)$ is the upper and lower bounds of $\|\mathbf{z}^*\|_0$ learned by equally treating features, i.e.,

$$f(\mathrm{M}_2) \leq \|\mathbf{z}^*\|_0 \leq f(\mathrm{M}_1). \tag{12}$$

The detailed proof is available in the supplementary material, available online.

Following a similar technology route, Theorem 2 gives the bound of $\|\hat{\mathbf{z}}^*\|_0$ learned by proposed reweighting features.

*Theorem 2.* $f(\frac{\mathrm{M}_1}{D})$ and $f(\frac{\mathrm{M}_2}{D})$ is the upper and lower bounds of $\|\hat{\mathbf{z}}^*\|_0$ learned by reweighting features, i.e.,

$$f\left(\frac{\mathrm{M}_2}{D}\right) \leq \|\hat{\mathbf{z}}^*\|_0 \leq f\left(\frac{\mathrm{M}_1}{D}\right), \tag{13}$$

where $D = \min_p d_p$ denotes the minimal feature dimension.

*Proof.* Compared to equally treating manner, the analysis of reweighting manner becomes complex due to multiple feature dimensions in the MVC scenario. For simplicity, we firstly consider single-view setting, supposing $\mathbf{\Omega} = \frac{1}{d}\mathbf{I}$, where $d$ is the feature dimension. (6) is transformed to

$$\begin{aligned} \min_{\mathbf{z}} \ & \|\mathbf{z} - \hat{\mathbf{b}}\|_2^2, \\ \text{s.t.} \ & \mathbf{z}^\top \mathbf{1}_m = 1, \mathbf{z} \geq \mathbf{0}, \end{aligned} \tag{14}$$

where $\hat{\mathbf{b}} = \frac{1}{d}\mathbf{b}$, $\hat{\mathrm{M}}_1 = \frac{1}{d}\mathrm{M}_1$, $\hat{\mathrm{M}}_2 = \frac{1}{d}\mathrm{M}_2$, and $\hat{\mathbf{z}}^*$ is the optimal solution.

According to Theorem 1, we have

$$f\left(\frac{\mathrm{M}_2}{d}\right) \leq \|\hat{\mathbf{z}}^*\|_0 \leq f\left(\frac{\mathrm{M}_1}{d}\right). \tag{15}$$

Furthermore, (15) is extended to MVC scenario,

$$\begin{aligned} f\left(\frac{\mathrm{M}_2}{d_1}\right) &\leq \|\hat{\mathbf{z}}^*\|_0 \leq f\left(\frac{\mathrm{M}_1}{d_1}\right) \\ &\vdots \\ f\left(\frac{\mathrm{M}_2}{d_m}\right) &\leq \|\hat{\mathbf{z}}^*\|_0 \leq f\left(\frac{\mathrm{M}_1}{d_m}\right) \end{aligned} \tag{16}$$

which can be unified into

$$\max_p f\left(\frac{\mathrm{M}_2}{d_p}\right) \leq \|\hat{\mathbf{z}}^*\|_0 \leq \min_p f\left(\frac{\mathrm{M}_1}{d_p}\right) \tag{17}$$

Considering the monotony of function $f$ in (11), the lower bound of $\hat{\mathbf{z}}^*$ will be gained by $d = \min_p d_p$ and the upper bound will be gained by $d = \max_p d_p$. Since the ceiling of $\|\mathbf{z}^*\|_0$ and $\|\hat{\mathbf{z}}^*\|_0$ is the anchor number $m$, we pay more attention to the lower bound, (17) is simplified by

$$f\left(\frac{\mathrm{M}_2}{D}\right) \leq \|\hat{\mathbf{z}}^*\|_0 \leq f\left(\frac{\mathrm{M}_1}{D}\right) \tag{18}$$

where $D = \min_p d_p$ denotes the minimal feature dimension.

This completes the proof. □

Consequently, based on Theorems 1 and 2, a corollary is that introducing noisy feature filter mechanism increases the bounds of bipartite graph's fuzzy membership.

In summary, this section gives the bounds of bipartite graph's fuzzy membership ($\|\mathbf{z}\|_0$), and theoretically analyzes the relationship of MVGBC-ETF ($\|\mathbf{z}^*\|_0$) and MVBGC-NFF ($\|\hat{\mathbf{z}}^*\|_0$). A corollary is that introducing noisy feature filter mechanism will increase their bounds. We emphasize that the proposed bound function $f$ is a general method, suitable for both single-view and multi-view models that utilize a similar manner in optimizing bipartite graph $\mathbf{Z}$.

### C. Optimization

Directly solving (4) is difficult due to non-convexity. Instead, we utilize alternative optimization manner.

*1) Update* $\mathbf{A}_p$*:* With $\boldsymbol{\alpha}$, $\boldsymbol{\Omega}_p$, and $\mathbf{Z}$ being fixed, each $\mathbf{A}_p$ can be solved by

$$\max_{\mathbf{A}_p} \operatorname{Tr}(\mathbf{A}_p^\top \mathbf{E}_p), \text{ s.t. } \mathbf{A}_p^\top \mathbf{A}_p = \mathbf{I}_m, \tag{19}$$

where $\mathbf{E}_p = \boldsymbol{\Omega}_p \mathbf{X}_p \mathbf{Z}^\top$. Theorem 3 gives the optimal solution of (19).

*Theorem 3.* Supposing the rank-$k$ SVD of $\mathbf{E}_p$ is $\mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{n\times k}$, $\boldsymbol{\Lambda} \in \mathbb{R}^{k\times k}$, and $\mathbf{V} \in \mathbb{R}^{k\times k}$. The optimal solution of (19) is as follows,

$$\mathbf{A}_p = \mathbf{U}\mathbf{V}^\top, \tag{20}$$

The details are available in the supplementary material, available online.

*2) Update* $\mathbf{Z}$*:* With $\boldsymbol{\alpha}$, $\boldsymbol{\Omega}_p$, and $\mathbf{A}_p$ being fixed, updating $\mathbf{Z}$ can be rewritten as $n$ column independent QP problems, i.e.,

$$\begin{aligned} &\min_{\mathbf{Z}_{:,j}} \frac{1}{2}\mathbf{Z}_{:,j}^\top \mathbf{O}\mathbf{Z}_{:,j} + \mathbf{h}^\top \mathbf{Z}_{:,j}, \\ &\text{s.t. } \mathbf{Z}_{:,j}^\top \mathbf{1} = 1,\ \mathbf{Z}_{:,j} \geq 0, \end{aligned} \tag{21}$$

where $\mathbf{O} = 2\sum_{p=1}^{v} \alpha_p^2 \mathbf{I}$, $\mathbf{h}^\top = -2\sum_{p=1}^{v} \alpha_p^2 \mathbf{X}_{p[:,j]}^\top \boldsymbol{\Omega}_p^\top \mathbf{A}_p$.

Furthermore, each QP problem is transformed to

$$\begin{aligned} &\min_{\mathbf{Z}_{:,j}} \frac{1}{2}\left\|\mathbf{Z}_{:,j} - \hat{\mathbf{Z}}_{:,j}\right\|_2^2, \\ &\text{s.t. } \mathbf{Z}_{:,j}^\top \mathbf{1} = 1,\ \mathbf{Z}_{:,j} \geq 0, \end{aligned} \tag{22}$$

where $\hat{\mathbf{Z}}_{:,j} = -\frac{\mathbf{h}}{2\sum_{p=1}^{v}\alpha_p^2}$. Mathematically, Theorem 4 gives the analytical solution of (22).

*Theorem 4.* The analytical solution of (22) is as follows,

$$\mathbf{Z}_{:,j} = \max\left\{\hat{\mathbf{Z}}_{:,j} + \beta_j \mathbf{1}_n, 0\right\}, \tag{23}$$

where $\beta_j$ can be efficiently computed by Newton's method.

The details are available in the supplementary material, available online.

*3) Update* $\boldsymbol{\Omega}_p$*:* With $\mathbf{A}_p$, $\mathbf{Z}$, and $\alpha$ being fixed, each $\boldsymbol{\Omega}_p$ can be solved by

$$\begin{aligned} &\min_{\boldsymbol{\Omega}_p} \|\boldsymbol{\Omega}_p \mathbf{X}_p - \mathbf{A}_p \mathbf{Z}\|_F^2, \\ &\text{s.t. } \boldsymbol{\Omega}_p = \operatorname{diag}(\boldsymbol{\omega}_p), \boldsymbol{\omega}_p^\top \mathbf{1} = 1, \boldsymbol{\omega}_p \geq \mathbf{0}. \end{aligned} \tag{24}$$

Furthermore, we have

$$\begin{aligned} &\min_{\boldsymbol{\Omega}_p} \operatorname{Tr}\left(\boldsymbol{\Omega}_p^\top \mathbf{X}_p \mathbf{X}_p^\top \boldsymbol{\Omega}_p - 2\mathbf{X}_p \mathbf{Z}^\top \mathbf{A}_p^\top \boldsymbol{\Omega}_p\right), \\ &\text{s.t. } \boldsymbol{\Omega}_p = \operatorname{diag}(\boldsymbol{\omega}_p), \boldsymbol{\omega}_p^\top \mathbf{1} = 1, \boldsymbol{\omega}_p \geq \mathbf{0}. \end{aligned} \tag{25}$$

**Algorithm 1:** MVBGC-NFF.

```
Input: {X_p}_{p=1}^v, k, and m.
Initialize Z, {Ω_p}_{p=1}^v, and α.
repeat
    Optimize {A_p}_{p=1}^v by computing (19).
    Optimize Z by computing (22).
    Optimize {Ω_p}_{p=1}^v by computing (26).
    Optimize α by computing (29).
until convergence
Compute partition matrix H by performing SVD on Z.
Output: Get the predicted clustering labels by
    performing k-means on H.
```

Recalling $\boldsymbol{\Omega}_p$ is diagonal matrix, (25) is equivalent to

$$\begin{aligned} &\min_{\boldsymbol{\omega}_p} \sum_{q=1}^{d_p} \left(\boldsymbol{\omega}_{p(q)} \mathbf{X}_{p[q,:]} \mathbf{X}_{p[q,:]}^\top \boldsymbol{\omega}_{p(q)} - 2\left(\mathbf{X}_p \mathbf{Z}^\top \mathbf{A}_p^\top\right)_{qq} \boldsymbol{\omega}_{p(q)}\right), \\ &\text{s.t. } \boldsymbol{\omega}_p^\top \mathbf{1} = 1, \boldsymbol{\omega}_p \geq \mathbf{0}. \end{aligned} \tag{26}$$

Denoting $\mathbf{X}_{p[q,:]}\mathbf{X}_{p[q,:]}^\top$ and $2(\mathbf{X}_p \mathbf{Z}^\top \mathbf{A}_p^\top)_{qq}$ as $a_q$ and $e_q$, respectively. The above formulation can be converted to

$$\begin{aligned} &\min_{\boldsymbol{\omega}_p} \sum_{q=1}^{d_p} \left(\boldsymbol{\omega}_{p(q)}^2 a_q - e_q \boldsymbol{\omega}_{p(q)}\right), \\ &\text{s.t. } \boldsymbol{\omega}_{p(q)} \geq 0, \boldsymbol{\omega}_p^\top \mathbf{1} = 1, \end{aligned} \tag{27}$$

which can be rewritten as

$$\begin{aligned} &\min_{\boldsymbol{\omega}_p} \|\boldsymbol{\omega}_p - \hat{\boldsymbol{\omega}}_p\|_2^2, \\ &\text{s.t. } \boldsymbol{\omega}_{p(q)} \geq 0, \boldsymbol{\omega}_p^\top \mathbf{1} = 1, \end{aligned} \tag{28}$$

where $\hat{\boldsymbol{\omega}}_{p(q)} = \frac{e_q}{2a_q}$. According to Theorem 4, $\boldsymbol{\omega}_p$ can be efficiently solved with an analytical solution.

*4) Update* $\alpha_p$*:* With $\boldsymbol{\Omega}_p$, $\mathbf{A}_p$, and $\mathbf{Z}$ being fixed, updating $\alpha_p$ can be simplified by

$$\min_{\alpha_p} \sum_{p=1}^{v} \alpha_p^2 \tau_p, \text{ s.t. } \boldsymbol{\alpha}^\top \mathbf{1} = 1, \boldsymbol{\alpha} \geq 0, \tag{29}$$

where $\tau_p = \|\boldsymbol{\Omega}_p \mathbf{X}_p - \mathbf{A}_p \mathbf{Z}\|_F^2$. The optimal $\alpha_p$ can be directly solved by Cauchy-Schwarz inequality, i.e.,

$$\alpha_p = \frac{1/\tau_p}{\sum_{p=1}^{v} 1/\tau_p}. \tag{30}$$

### D. Complexity and Convergence Analysis

*Time Complexity:* The computational complexity during optimization includes four parts. For simplicity, we set $g_1 = \sum_{p=1}^{v} d_p$ and $g_2 = \sum_{p=1}^{v} d_p^2$. Updating $\{\mathbf{A}_p\}_{p=1}^{v}$ requires $\mathcal{O}(nmg_1 + m^2 g_1)$ time. Updating $\mathbf{Z}$ requires $\mathcal{O}(nmg_1)$ time. Updating $\{\boldsymbol{\Omega}_p\}_{p=1}^{v}$ requires $\mathcal{O}(n(mg_2 + g_1))$ time. Updating $\boldsymbol{\alpha}$ requires $\mathcal{O}(n(g_2 + mg_1))$ time. Therefore, the time complexity of optimization process is $\mathcal{O}(nm(g_1 + g_2) + m^2 g_1)$ time.

TABLE II
REAL-WORLD DATASETS

| Datasets | Samples | Views | Clusters | Features |
|---|---|---|---|---|
| MSRCV1 | 210 | 6 | 7 | 1,302/48/512/100/256/210 |
| ORL | 400 | 4 | 40 | 256/256/256/256 |
| Flower17 | 1,360 | 7 | 17 | 5,376/512/5,376/5,376/1,239/5,376/5,376 |
| VGGFace2_50 | 16,936 | 4 | 50 | 944/576/512/640 |
| Caltech256 | 30,607 | 4 | 257 | 944/576/512/640 |
| VGGFace2_100 | 36,287 | 4 | 100 | 944/576/512/640 |
| CIFAR100_Train | 50,000 | 4 | 100 | 944/576/512/640 |
| CIFAR100_All | 60,000 | 4 | 100 | 944/576/512/640 |
| VGGFace2_200 | 72,283 | 4 | 200 | 944/576/512/640 |
| TinyImageNet | 100,000 | 4 | 200 | 944/576/512/640 |
| YoutubeFace sel | 101,499 | 5 | 31 | 512/64/647/838 |
| YouTubeFace50 | 126,054 | 4 | 50 | 944/576/512/640 |

TABLE III
COMPLEXITY ANALYSIS COMPARISON

| Method | Time Complexity | Space Complexity |
|---|---|---|
| RMKM | $\mathcal{O}(n)$ | $\mathcal{O}(n(g_1+k)+g_1k)$ |
| AMGL | $\mathcal{O}(n^3)$ | $\mathcal{O}(n^2v+n(g_1+k))$ |
| FMR | $\mathcal{O}(n^3)$ | $\mathcal{O}(3n^2+n(g_1+3k))$ |
| PMSC | $\mathcal{O}(n^3)$ | $\mathcal{O}(n^2(v+1)+n(g_1+kv+k))$ |
| MCLES | $\mathcal{O}(n^3)$ | $\mathcal{O}(2n^2+n(g_1+2k)+g_1k)$ |
| BMVC | $\mathcal{O}(n)$ | $\mathcal{O}(nl+\bar{d}l)$ |
| LMVSC | $\mathcal{O}(n)$ | $\mathcal{O}(n(g_1+mv)+g_1mv)$ |
| SMVSC | $\mathcal{O}(n)$ | $\mathcal{O}(nm+(g_1+m)k)$ |
| SFMC | $\mathcal{O}(n)$ | $\mathcal{O}(n(g_1+mv))$ |
| FMCNOF | $\mathcal{O}(n)$ | $\mathcal{O}(n(g_1+mv+k)+mk)$ |
| FPMVS-CAG | $\mathcal{O}(n)$ | $\mathcal{O}(nk+(g_1+k)k)$ |
| UoMVSC | $\mathcal{O}(n^3)$ | $\mathcal{O}(2n^2v+ng_1)$ |
| SDAFG | $\mathcal{O}(n)$ | $\mathcal{O}(n(g_1+2mv))$ |
| Proposed | $\mathcal{O}(n)$ | $\mathcal{O}(n(g_1+m)+g_1m)$ |

In post-processing, we conduct SVD on $\mathbf{Z}$ to compute the partition and utilize $k$-means to output the discrete clustering labels, which requires $\mathcal{O}(nm^2)$ computational complexity. Consequently, the total computational complexity of MVBGC-NFF is $\mathcal{O}(nm(m+g_1+g_2)+m^2g_1)$. The complexity is linear with sample number $n$, making it suitable for large-scale tasks.

Note that comparing with equally treating features, introducing noisy feature filter will induce extra complexity $\mathcal{O}(n(mg_2+g_1))$, which means the computational burden will be obvious in high-dimensional feature scenarios, especially when feature dimension exceeds sample number. Generally, the increased computational burden is affordable in most applications.

*Space Complexity:* The space complexity is mainly caused by storing matrices, i.e., $\mathbf{X}_p \in \mathbb{R}^{d_p \times n}$, $\mathbf{A}_p \in \mathbb{R}^{d_p \times m}$, $\mathbf{Z} \in \mathbb{R}^{m \times n}$, and $\mathbf{\Omega}_p \in \mathbb{R}^{d_p \times d_p}$. Since $\mathbf{\Omega}_p$ is a diagonal matrix, we just need to store $\boldsymbol{\omega}_p$, which saves memory consumption. The total space complexity is $\mathcal{O}(n(m+g_1)+mg_1)$. Therefore, the space complexity is linear with $n$, making it applicable on large-scale datasets.

*Convergence:* Since our MVBGC-NFF is non-convex, jointly solving all the variables is difficult, we adopt alternative optimization in this article. Note that each sub-optimization can achieve the optimum, as pointed in [50], the objective of Algorithm 1 decreases monotonically during optimization and converges to a local minimum bounded by 0.

## IV. EXPERIMENT

### A. Synthetic Dataset

To visualize the adverse impact of noisy features, we first conduct experiments on a synthetic single-view four-dimensional dataset with 300 samples drawn from two clusters, generated from Gaussian function, as shown in Fig. 1. The first dimension with $x_1$ is discriminative for distinguishing clustering structure, the second dimension with $x_2$ is useless, and the rest two dimensions are noisy features.

### B. Real-World Datasets

Table II lists twelve real-world MVC datasets. MSRCV1 [51] includes 210 images drawn from 7 clusters, i.e., airplane, bicycle, building, car, caw, face, and tree. ORL[1] is a face dataset containing 400 images from 40 distinct categories. Flower17[2] is an image dataset with each class includes 80 flower images. Caltech256[3] contains 30607 images spanning 257 object categories. VGGFace2_50, VGGFace2_100, and VGGFace2_200 are derived from VGGFace.[4] CIFAR100_Train and CIFAR100_All[5] are image datasets drawn from 100 classes. TinyImageNet[6] contains 100000 images drawn from 200 classes. YoutubeFace sel and YouTubeFace50 are face video datasets extracted from YouTube [38].

### C. Compared Algorithms

Thirteen existing MVGC or MVBGC algorithms are compared as baselines, including 1) Multi-view k-means Clustering on Big Data (RMKM) [18]. 2) Parameter-free Auto-weighted Multiple Graph Learning (AMGL) [52]. 3) Flexible Multi-View Representation Learning for Subspace Clustering (FMR) [19]. 4) Partition Level Multi-view Subspace Clustering (PMSC) [20]. 5) Multi-view Clustering in Latent Embedding Space (MCLES) [21]. 6) Binary Multi-View Clustering (BMVC) [33]. 7) Large-scale Multi-view Subspace Clustering in Linear Time (LMVSC) [28]. 8) Scalable Multi-view Subspace Clustering with Unified Anchors (SMVSC) [39]. 9) Multi-view clustering: a Scalable and Parameter-free Bipartite Graph Fusion Method (SFMC) [29]. 10) Fast Multiview Clustering via Nonnegative and Orthogonal Factorization (FMCNOF) [53]. 11) Fast Parameter-free Multi-view Subspace Clustering with Consensus Anchor Guidance (FPMVS-CAG) [40]. 12) Unified One-step Multi-view Spectral Clustering (UoMVSC) [54]. 13) Structure Diversity-Induced Anchor Graph Fusion for Multi-View Clustering (SDAFG) [37]. Table III compares the time and space complexities of baselines and ours. Due to space limitations, the experimental results of UoMVSC and SDAFG are provided in supplementary material, available online.

### D. Experimental Settings

The source codes of the compared baselines are directly downloaded from public websites without extra corrections, and the utilized hyper-parameters are directly tuned following

[1] https://cam-orl.co.UK/facedatabase.html

[2] https://www.robots.ox.ac.UK/ vgg/data/flowers/17/

[3] https://authors.library.caltech.edu/7694/

[4] https://www.robots.ox.ac.UK/ vgg/data/vgg_face/

[5] https://www.cs.toronto.edu/ kriz/cifar.html

[6] https://www.kaggle.com/c/tiny-imagenet

TABLE IV
CLUSTERING METRICS COMPARISON (ACC, NMI, PURITY, AND F-SCORE). THE BEST ONES ARE MARKED IN BOLD, THE SECOND-BEST ONES ARE IN ITALIC AND UNDERLINED, "N/A" MEANS TIME-OUT OR MEMORY-OVERFLOW ERRORS

| Datasets | RMKM | AMGL | FMR | PMSC | MCLES | BMVC | LMVSC | SMVSC | SFMC | FMCNOF | FPMVS-CAG | Proposed |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ACC (%) | | | | | | | | | | | |
| MSRCV1 | 71.43±0.00 | 76.44±6.30 | 77.48±6.40 | 47.45±4.23 | 60.86±3.45 | 26.67±0.00 | *83.73±7.20* | 70.51±4.98 | 60.48±0.00 | 47.14±0.00 | 71.95±5.36 | **86.19±7.59** |
| ORL | 47.00±0.00 | 59.73±2.78 | 25.21±1.10 | 21.48±1.02 | 29.18±1.99 | 43.25±0.00 | *61.50±2.96* | 47.76±2.36 | 37.00±0.00 | 21.50±0.00 | 54.63±1.49 | **64.53±3.55** |
| Flower17 | 23.24±0.00 | 9.70±1.53 | 33.43±1.75 | 20.82±0.74 | N/A | 26.99±0.00 | *34.47±1.86* | 27.13±0.84 | 7.57±0.00 | 17.43±0.00 | 25.99±1.83 | **36.61±1.96** |
| VGGFace2_50 | 8.23±0.00 | 2.95±0.35 | N/A | N/A | N/A | 10.30±0.00 | 10.56±0.26 | *13.36±0.60* | 3.64±0.00 | 5.51±0.00 | 12.06±0.36 | **14.55±0.56** |
| Caltech256 | 9.87±0.00 | N/A | N/A | N/A | N/A | 8.63±0.00 | 9.57±0.17 | *10.54±0.15* | 4.50±0.00 | 2.70±0.00 | 8.78±0.07 | **11.36±0.36** |
| VGGFace2_100 | 5.39±0.00 | N/A | N/A | N/A | N/A | 6.17±0.00 | 6.09±0.09 | *7.90±0.20* | 1.98±0.00 | 3.47±0.00 | 6.02±0.18 | **10.24±0.32** |
| CIFAR100_Train | *9.79±0.00* | N/A | N/A | N/A | N/A | 8.38±0.00 | 8.92±0.17 | 8.48±0.18 | 1.33±0.00 | 4.40±0.00 | 7.46±0.12 | **10.74±0.27** |
| CIFAR100_All | N/A | N/A | N/A | N/A | N/A | 8.32±0.00 | *9.53±0.15* | 8.34±0.17 | 1.18±0.00 | 3.66±0.00 | 7.29±0.11 | **10.76±0.27** |
| VGGFace2_200 | N/A | N/A | N/A | N/A | N/A | 3.99±0.00 | *4.31±0.06* | 4.25±0.06 | N/A | 0.90±0.00 | 3.41±0.05 | **6.44±0.18** |
| TinyImageNet | N/A | N/A | N/A | N/A | N/A | 4.09±0.00 | *4.28±0.05* | 0.04±0.04 | N/A | 0.50±0.00 | 2.83±0.02 | **5.08±0.10** |
| YoutubeFace sel | N/A | N/A | N/A | N/A | N/A | 17.88±0.00 | 16.34±0.54 | 10.09±0.33 | N/A | *23.27±0.00* | 12.38±0.16 | **27.15±0.67** |
| YouTubeFace50 | N/A | N/A | N/A | N/A | N/A | 66.00±0.00 | *68.32±2.45* | 2.46±2.46 | N/A | 21.66±0.00 | 64.24±2.97 | **68.48±3.12** |
| | NMI (%) | | | | | | | | | | | |
| MSRCV1 | 63.03±0.00 | 77.65±3.23 | 69.48±3.31 | 34.29±2.81 | 51.72±2.37 | 8.29±0.00 | *78.93±4.60* | 62.01±2.61 | 60.23±0.00 | 38.42±0.00 | 65.69±3.27 | **82.79±5.01** |
| ORL | 71.83±0.00 | *80.28±1.37* | 48.33±0.81 | 43.87±0.84 | 54.03±1.67 | 65.32±0.00 | 79.39±1.15 | 72.73±1.13 | 76.30±0.00 | 43.32±0.00 | 77.41±0.54 | **81.56±1.88** |
| Flower17 | 22.07±0.00 | 10.25±4.01 | 30.65±0.91 | 19.13±0.48 | N/A | 25.62±0.00 | *33.82±1.10* | 25.78±0.76 | 7.87±0.00 | 14.68±0.00 | 25.81±1.59 | **34.45±1.51** |
| VGGFace2_50 | 9.66±0.00 | 2.04±0.50 | N/A | N/A | N/A | 13.48±0.00 | 12.64±0.28 | *16.21±0.49* | 1.63±0.00 | 4.74±0.00 | 14.74±0.55 | **17.97±0.44** |
| Caltech256 | 31.01±0.00 | N/A | N/A | N/A | N/A | 31.83±0.00 | *31.96±0.11* | 28.27±0.24 | 5.67±0.00 | N/A | 22.97±0.21 | **33.60±0.18** |
| VGGFace2_100 | 11.14±0.00 | N/A | N/A | N/A | N/A | 14.26±0.00 | 11.92±0.11 | *14.80±0.23* | 0.91±0.00 | 5.81±0.00 | 12.33±0.29 | **18.47±0.28** |
| CIFAR100_Train | **17.80±0.00** | N/A | N/A | N/A | N/A | 15.12±0.00 | 15.49±0.15 | 14.83±0.22 | 3.78±0.00 | 8.49±0.00 | 13.87±0.21 | *17.14±0.19* |
| CIFAR100_All | N/A | N/A | N/A | N/A | N/A | 15.05±0.00 | *15.40±0.18* | 14.40±0.20 | 0.53±0.00 | 7.04±0.00 | 13.62±0.16 | **16.90±0.22** |
| VGGFace2_200 | N/A | N/A | N/A | N/A | N/A | *15.04±0.00* | 13.94±0.09 | 13.94±0.13 | N/A | N/A | 11.18±0.18 | **18.46±0.18** |
| TinyImageNet | N/A | N/A | N/A | N/A | N/A | *13.75±0.00* | 13.23±0.06 | 0.12±0.12 | N/A | N/A | 10.04±0.12 | **14.56±0.11** |
| YoutubeFace sel | N/A | N/A | N/A | N/A | N/A | 3.58±0.00 | *15.33±0.33* | 7.66±0.17 | N/A | 4.41±0.00 | 7.60±0.09 | **24.01±0.40** |
| YouTubeFace50 | N/A | N/A | N/A | N/A | N/A | 81.90±0.00 | *82.43±0.78* | 0.85±0.85 | N/A | 43.03±0.00 | 82.08±1.07 | **83.80±0.96** |
| | Purity (%) | | | | | | | | | | | |
| MSRCV1 | 74.76±0.00 | 80.45±4.29 | 79.01±4.16 | 49.91±3.78 | 61.57±2.91 | 27.14±0.00 | *85.25±5.56* | 71.51±4.02 | 62.86±0.00 | 50.48±0.00 | 72.33±5.01 | **87.30±6.09** |
| ORL | 53.00±0.00 | 66.92±2.07 | 26.48±1.14 | 23.90±1.08 | 31.96±1.78 | 47.50±0.00 | 65.68±2.53 | 51.91±2.16 | **78.00±0.00** | 21.75±0.00 | 58.97±1.39 | *68.41±3.03* |
| Flower17 | 24.49±0.00 | 10.76±1.53 | 34.74±1.38 | 22.20±0.62 | N/A | 29.41±0.00 | *35.89±1.54* | 27.88±0.78 | 10.29±0.00 | 17.57±0.00 | 26.38±1.85 | **38.15±1.93** |
| VGGFace2_50 | 9.28±0.00 | 3.17±0.37 | N/A | N/A | N/A | 11.44±0.00 | 11.41±0.25 | *13.90±0.58* | 3.86±0.00 | 5.67±0.00 | 12.27±0.36 | **15.65±0.54** |
| Caltech256 | 15.65±0.00 | N/A | N/A | N/A | N/A | 14.94±0.00 | *16.25±0.14* | 14.00±0.12 | 5.27±0.00 | 2.70±0.00 | 11.88±0.11 | **17.78±0.26** |
| VGGFace2_100 | 6.26±0.00 | N/A | N/A | N/A | N/A | 7.15±0.00 | 7.02±0.09 | *8.55±0.22* | 2.09±0.00 | 3.50±0.00 | 6.33±0.21 | **11.31±0.33** |
| CIFAR100_Train | *11.12±0.00* | N/A | N/A | N/A | N/A | 9.27±0.00 | 10.41±0.15 | 9.09±0.21 | 2.97±0.00 | 4.49±0.00 | 7.84±0.18 | **11.96±0.24** |
| CIFAR100_All | N/A | N/A | N/A | N/A | N/A | 9.33±0.00 | *10.90±0.20* | 8.93±0.17 | 1.26±0.00 | 3.68±0.00 | 7.64±0.13 | **12.02±0.23** |
| VGGFace2_200 | N/A | N/A | N/A | N/A | N/A | 4.66±0.00 | *4.98±0.07* | 4.62±0.08 | N/A | 0.90±0.00 | 3.56±0.06 | **7.20±0.18** |
| TinyImageNet | N/A | N/A | N/A | N/A | N/A | 4.69±0.00 | *4.93±0.04* | 0.04±0.04 | N/A | 0.50±0.00 | 2.95±0.02 | **5.71±0.11** |
| YoutubeFace sel | N/A | N/A | N/A | N/A | N/A | 26.62±0.00 | *29.55±0.40* | 26.62±0.00 | N/A | 26.62±0.00 | 26.81±0.06 | **35.85±0.46** |
| YouTubeFace50 | N/A | N/A | N/A | N/A | N/A | *73.64±0.00* | 73.21±2.18 | 2.61±2.61 | N/A | 22.83±0.00 | 66.84±3.02 | **75.04±2.33** |
| | F-score (%) | | | | | | | | | | | |
| MSRCV1 | 59.98±0.00 | 70.28±4.42 | 66.76±4.50 | 34.05±2.34 | 48.53±2.10 | 16.01±0.00 | *77.43±6.43* | 59.31±2.82 | 52.43±0.00 | 33.85±0.00 | 61.55±3.54 | **80.46±7.02** |
| ORL | 33.66±0.00 | 35.12±4.54 | 9.27±0.83 | 6.48±0.54 | 17.63±1.35 | 24.84±0.00 | *50.10±2.86* | 32.37±1.71 | 23.74±0.00 | 12.09±0.00 | 42.87±1.47 | **54.04±3.96** |
| Flower17 | 14.35±0.00 | 11.49±0.55 | 20.09±0.81 | 12.33±0.36 | N/A | 16.61±0.00 | *22.57±0.95* | 17.53±0.21 | 10.94±0.00 | 13.93±0.00 | 17.29±0.39 | **24.51±1.30** |
| VGGFace2_50 | 3.69±0.00 | 3.91±0.02 | N/A | N/A | N/A | 5.10±0.00 | 5.09±0.15 | *6.35±0.18* | 4.16±0.00 | 4.36±0.00 | 6.10±0.07 | **7.46±0.25** |
| Caltech256 | *7.28±0.00* | N/A | N/A | N/A | N/A | 6.25±0.00 | 6.00±0.28 | 5.37±0.20 | 1.17±0.00 | 1.17±0.00 | 3.22±0.02 | **9.24±0.86** |
| VGGFace2_100 | 2.11±0.00 | N/A | N/A | N/A | N/A | 2.77±0.00 | 2.47±0.04 | *3.45±0.06* | 2.12±0.00 | 2.39±0.00 | 3.14±0.01 | **4.33±0.11** |
| CIFAR100_Train | 3.99±0.00 | N/A | N/A | N/A | N/A | 4.17±0.00 | 3.52±0.06 | *4.46±0.07* | 1.98±0.00 | 2.74±0.00 | 3.74±0.01 | **4.62±0.12** |
| CIFAR100_All | N/A | N/A | N/A | N/A | N/A | 4.37±0.00 | 3.69±0.05 | *4.47±0.06* | 1.98±0.00 | 2.56±0.00 | 3.78±0.01 | **4.64±0.11** |
| VGGFace2_200 | N/A | N/A | N/A | N/A | N/A | 1.46±0.00 | 1.40±0.03 | *1.61±0.02* | N/A | 1.06±0.00 | 1.59±0.01 | **2.29±0.06** |
| TinyImageNet | N/A | N/A | N/A | N/A | N/A | *1.55±0.00* | 1.36±0.01 | 0.01±0.01 | N/A | 0.99±0.00 | 1.50±0.00 | **1.66±0.03** |
| YoutubeFace sel | N/A | N/A | N/A | N/A | N/A | *15.78±0.00* | 7.54±0.13 | 6.65±0.16 | N/A | **16.25±0.00** | 10.56±0.19 | 10.92±0.18 |
| YouTubeFace50 | N/A | N/A | N/A | N/A | N/A | 57.09±0.00 | **62.49±2.45** | 2.56±2.56 | N/A | 15.67±0.00 | 56.89±3.18 | *61.83±3.35* |

the authors' suggestion. For our proposed MVBGC-NFF, the anchor number $m$ varies in $[k, 2\,k, 3k, 4k]$. Since we impose the orthogonal constraint on anchor matrix $\mathbf{A}$, $\min_p d_p \leq m$ should be satisfied. To alleviate the randomness from $k$-means, we report mean ± std by repetitively running 50 times. Four metrics, including accuracy (ACC), normalized mutual information (NMI), purity, and F-score are to quantify performance. Experiments are performed on a computer with Intel Core i9 10900X CPU (3.5GHZ), 64 GB RAM, and Matlab 2020b (64 b) version.

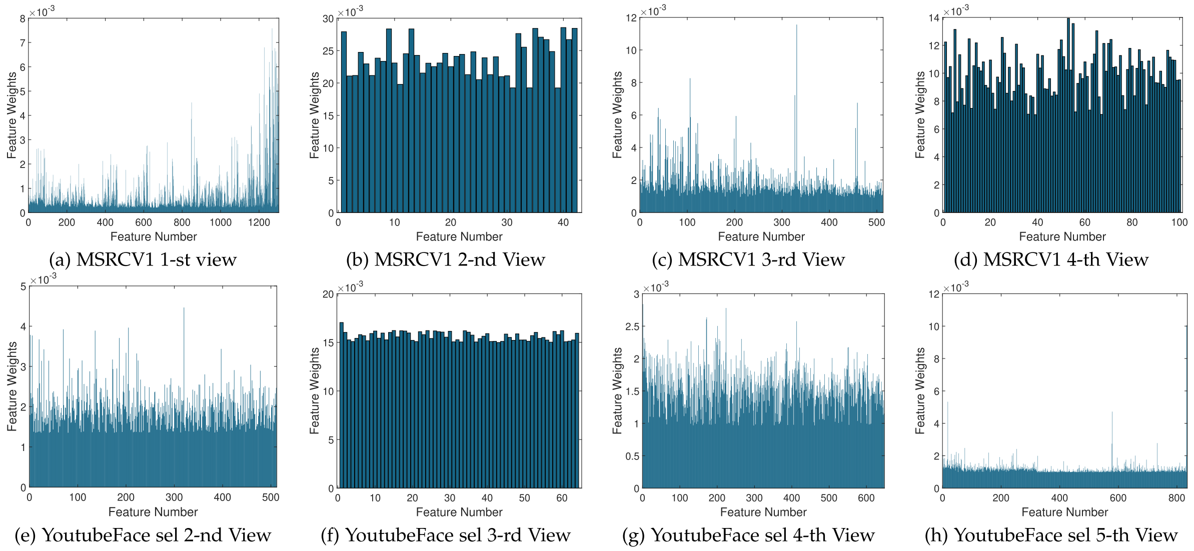
### E. Clustering Performance

Table IV reports four metrics comparisons. The best results are marked in bold, the second-best ones are in italic and underlined, and "N/A" denotes unavailable results due to time-out or memory-overflow errors. Based on the results, we observe that:

1) The proposed MVBGC-NFF achieves the best or second-best performance. Compared to the second-best ones, our MVBGC-NFF achieves 2.46%, 3.03%, 2.14%, 1.19%, 2.34%, 1.24%, 2.13%, and 3.89% improvement of ACC on eight datasets, respectively. Other metrics can demonstrate its superiority as well.
2) The traditional MVGC methods, i.e., RMKM, AMGL, FMR, PMSC, and MCLES, encounter scalability issues on large-scale datasets due to massive matrices computation and memory incurred by building the full graph. Our MVBGC-NFF outperforms them by 8.71%, 4.80%, 3.18%, 6.32%, 1.49%, 4.83%, and 0.95% of ACC on seven datasets, illustrating our effectiveness.
3) Compared to existing MVBGC methods, i.e., BMVC, LMVSC, SMVSC, SFMC, FMCNOF, and FPMVS-CAG, our MVBGC-NFF still achieves comparable or better performance. Especially, LMVSC can be regarded as the strongest competitor, our MVBGC-NFF exceeds it by large margins of 2.46%, 3.03%, 2.14%, 3.99%, 4.14%, 2.13%, and 10.81% on seven datasets, respectively, verifying our superiority.

### F. Feature Weight Distribution

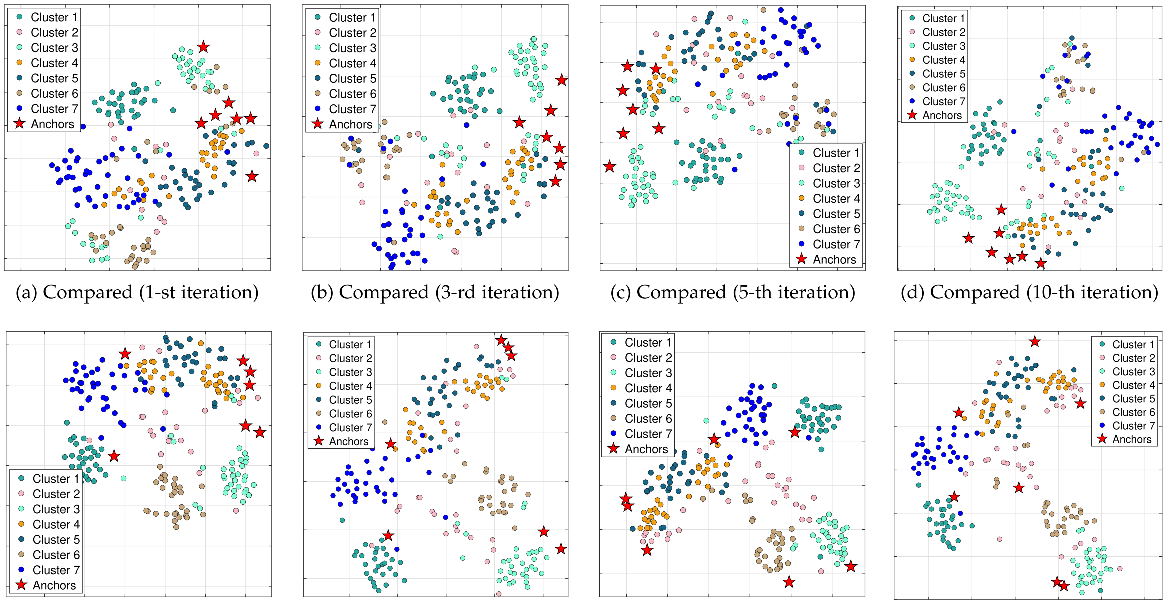
Fig. 2 depicts the learned feature weight distribution $\mathbf{\Omega}$ on MSRCV1 and YoutubeFace sel datasets. We observe that:



(a) MSRCV1 1-st view (b) MSRCV1 2-nd View (c) MSRCV1 3-rd View (d) MSRCV1 4-th View

(e) YoutubeFace sel 2-nd View (f) YoutubeFace sel 3-rd View (g) YoutubeFace sel 4-th View (h) YoutubeFace sel 5-th View

Fig. 2. Feature weights distribution of our MVBGC-NFF on MSRCV1 and YoutubeFace sel datasets.



(a) Compared (1-st iteration) (b) Compared (3-rd iteration) (c) Compared (5-th iteration) (d) Compared (10-th iteration)

(e) Proposed (1-st iteration) (f) Proposed (3-rd iteration) (g) Proposed (5-th iteration) (h) Proposed (10-th iteration)

Fig. 3. Comparing the t-SNE visualization of compared MVBGC-ETF (Compared) and our model on MSRCV1 dataset.

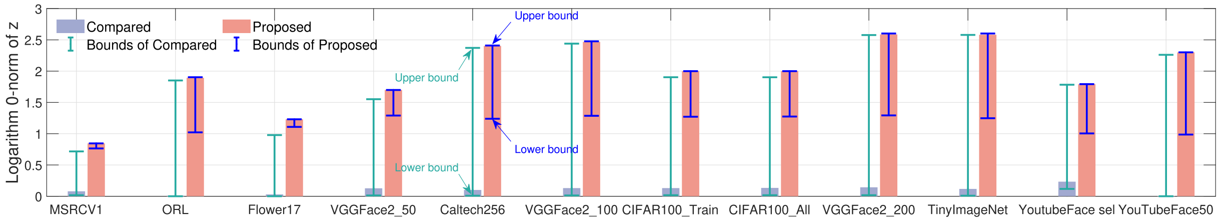
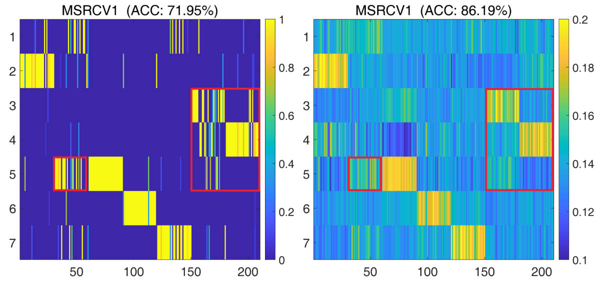


Fig. 4. Logarithm bipartite graph's fuzzy membership ($\|\mathbf{z}\|_0$) from equally-treating manner (Compared) and our noisy feature filter manner (Proposed). The upper and dot line denotes the predicted upper and lower bound, respectively.



(a) MVBGC-ETF (b) Proposed

Fig. 5. Visualization of bipartite graph from compared MVBGC-ETF and our MVBGC-NFF on MSRCV1 datasets.

1) The weight distributions vary across multiple views, with several views sparse while others smooth, illustrating the difference of features' representation.
2) The sparse weight distribution illustrates that inexpressive or noisy features indeed exist in real-world datasets, especially for the views with high-dimensional features. Conversely, low-dimensional features exhibit more smooth distribution relatively. A reasonable explanation is that there is a higher probability of validly extracting low-dimensional features while higher-dimensional features are likely to contain more noise.
3) Similar to Fig. 1(e) and (f) on synthetic dataset, our noisy feature filter mechanism achieves distinguishing features' discrimination on real-world datasets, i.e., the discriminative features are well preserved while the noisy or inexpressive ones are filtered, which contributes to effectively depressing the noisy ones.

### G. *Visualization of "Anchor Shift" Phenomenon*

To verify the generality of "anchor shift" phenomenon incurred by noisy features in real case, Fig. 3 plots the evolution of t-SNE visualization during optimization on MSRCV1 dataset, which shows the separation of clusters and anchors' distribution. Similarly, conventional equally-treating manner also incurs "anchor shift" issue: the learned anchors flock together, making it insufficient to represent the latent data distribution, which directly degrades the quality of bipartite graph and consequently ruins the clustering performance (ACC: 71.95%). In contrast, our MVBGC-NFF automatically measures their discrimination (refer to feature weight distribution shown in Fig. 2). Therefore, the undesired "anchor shift" issue is alleviated, and the learned anchors exhibit stronger representation, which improves clustering performance (ACC: 86.19%).

### H. *Validating the Theoretical Finding About Proposed Bipartite Graph's Fuzzy Membership*

This section validates our theoretical finding that distinguishing features' discrimination will increase the proposed bipartite graph's fuzzy membership.

Fig. 5 depicts the visualization of bipartite graph on MSRCV1. Clearly, compared to the sparse bipartite graph learned by MVBGC-ETF, introducing noisy feature filter mechanism can increase fuzzy membership. Recall the "anchor shift" phenomenon shown in Fig. 3, the potential absolute membership (hard clustering) preserves much inaccurate relationship (highlighted by red box), which directly degrades bipartite graph's quality (ACC: 71.95%). By contrast, introducing fuzzy probability membership (soft partition) can fully explore the entire anchors' representation, which contributes to remedying the inaccurate memberships (ACC: 86.19%).

Furthermore, Fig. 4 quantifies the logarithm experimental fuzzy membership ($\|\mathbf{z}\|_0$) and the predicted lower and upper bounds. We observe that:

1) The predicted bounds exhibit datasets-related results. The experimental $\|\mathbf{z}\|_0$ lies between the predicted lower and upper bounds, verifying our theoretical bounds.
2) Comparing to equally treating manner, introducing the reweighting manner increases the bounds of fuzzy membership on all datasets, especially for the lower bound, which accounts for the increasing fuzzy membership in experiments.
3) Generally, the experimental $\|\mathbf{z}\|_0$ from the compared MVBGC-ETF approaches the predicted lower bound, while the $\|\mathbf{z}\|_0$ from the proposed MVBGC-NFF nearly reaches the predicted upper bound.

### I. *Comparison With Equally-Treating Feature Manner*

Table V reports the comparison of clustering metrics and execution time. From the results, we observe that the noisy feature filter mechanism consistently exceeds equally-treating manner, with large margins of 14.24%, 9.90%, 9.42%, 2.13%, 2.58%, 4.07%, 3.28%, 3.48%, 3.04%, 14.77%, and 2.24% of ACC on eleven datasets, which is convincing evidence supporting the significance of filtering noisy features. The execution time comparison indicates that the extra computational burden is affordable in most applications.

### J. *Running Time Comparison*

Fig. 6 plots time consumption, we observe that:

1) Compared to full graph clustering, MVBGC methods significantly reduce the running time, owing to linear computational complexity.
2) Comparing to MVBGC coupled anchor learning, i.e., SMVSC and FPMVS-CAG, our MVBGC-NFF requires more running time, mainly due to our noisy feature filter mechanism. As pointed in Section III-D, the extra computational complexity increases with the feature dimension, which is evident on Flower17 (maximal dimension: 5376) over others (maximal dimension: 1302). Generally, we believe that extra computational burden is a worthwhile sacrifice in depressing noisy features.

### K. *Convergence and Sensitivity*

Fig. 7 shows the evolution of objective. Clearly, it decreases monotonically and converges to a local minimum, verifying the convergence.

Fig. 8 plots the sensitivity of anchor number varying in $[k, 2\,k, 3k, 4k]$. Since the constraint $\min_p d_p \leq m$ should be satisfied according to (4), both the maximum anchor

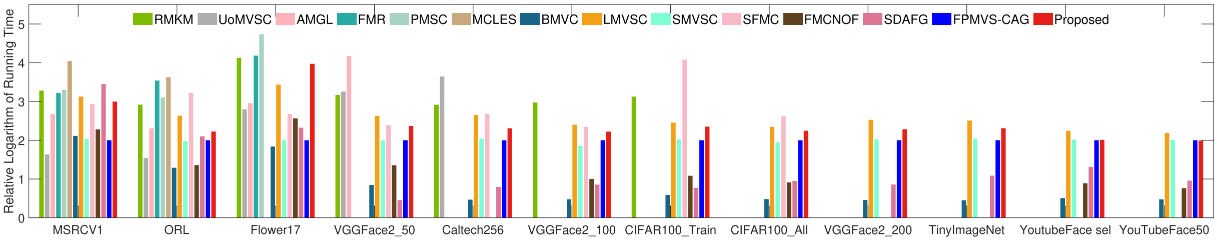
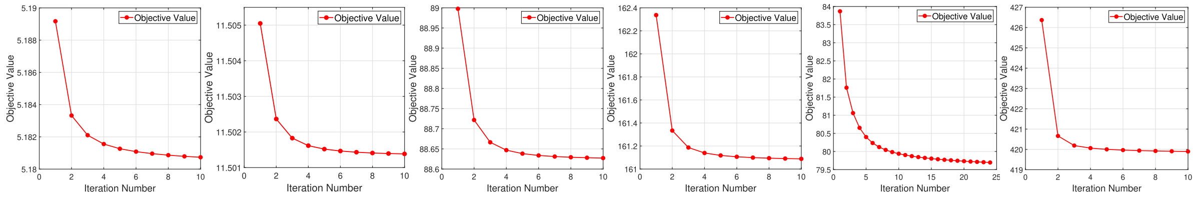
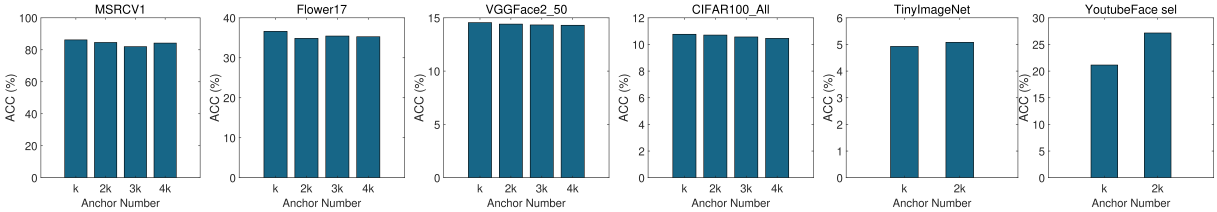


Fig. 6. Relative logarithm running time comparison. FPMVS-CAG is set as the baseline.



(a) MSRCV1 (b) Flower17 (c) VGGFace2_50 (d) CIFAR100_All (e) TinyImageNet (f) YoutubeFace sel

Fig. 7. Validating the convergence. More results are presented in supplementary material, available online.



(a) MSRCV1 (b) Flower17 (c) VGGFace2_50 (d) CIFAR100_All (e) TinyImageNet (f) YoutubeFace sel

Fig. 8. The sensitivity of anchor number. More results are presented in supplementary material, available online.

TABLE V
Performance Comparison With the Conventional Equally-Treating Features Manner (compared)

| Model | MSRCV1 | ORL | Flower17 | VGGFace2_50 | Caltech256 | VGGFace2_100 | CIFAR100_Train | CIFAR100_All | VGGFace2_200 | TinyImageNet | YoutubeFace sel | YouTubeFace50 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ACC (%) | | | | | | | | | | | |
| Compared | 71.95±5.36 | 54.63±1.49 | 27.18±1.57 | 12.42±0.61 | 8.78±0.07 | 6.17±0.20 | 7.46±0.12 | 7.29±0.11 | 3.41±0.05 | 2.84±0.02 | 12.38±0.16 | **68.89±2.72** |
| Proposed | **86.19±7.59** | **64.53±3.55** | **36.61±1.96** | **14.55±0.56** | **11.36±0.36** | **10.24±0.32** | **10.74±0.27** | **10.76±0.27** | **6.44±0.18** | **5.08±0.10** | **27.15±0.67** | 68.48±3.12 |
| | NMI (%) | | | | | | | | | | | |
| Compared | 65.69±3.27 | 77.41±0.54 | 25.16±1.65 | 15.20±0.65 | 22.97±0.21 | 12.49±0.28 | 13.87±0.21 | 13.62±0.16 | 11.18±0.18 | 10.20±0.11 | 7.60±0.09 | 83.55±1.22 |
| Proposed | **82.79±5.01** | **81.56±1.88** | **34.45±1.51** | **17.97±0.44** | **33.60±0.18** | **18.47±0.28** | **17.14±0.19** | **16.90±0.22** | **18.46±0.18** | **14.56±0.11** | **24.01±0.40** | **83.80±0.96** |
| | Purity (%) | | | | | | | | | | | |
| Compared | 72.33±5.01 | 58.97±1.39 | 28.07±1.56 | 12.78±0.61 | 11.88±0.11 | 6.45±0.21 | 7.84±0.18 | 7.64±0.13 | 3.56±0.06 | 2.91±0.02 | 26.81±0.06 | 71.40±2.81 |
| Proposed | **87.30±6.09** | **68.41±3.03** | **38.15±1.93** | **15.65±0.54** | **17.78±0.26** | **11.31±0.33** | **11.96±0.24** | **12.02±0.23** | **7.20±0.18** | **5.71±0.11** | **35.85±0.46** | **75.04±2.33** |
| | F-score (%) | | | | | | | | | | | |
| Compared | 61.55±3.54 | 42.87±1.47 | 16.41±0.44 | 6.13±0.09 | 3.22±0.02 | 3.12±0.02 | 3.74±0.01 | 3.78±0.01 | 1.59±0.01 | 1.55±0.00 | 10.56±4.91 | 59.26±4.91 |
| Proposed | **80.46±7.02** | **54.04±3.96** | **24.51±1.30** | **7.46±0.25** | **9.24±0.86** | **4.33±0.11** | **4.62±0.12** | **4.64±0.11** | **2.29±0.06** | **1.66±0.03** | **10.92±0.18** | **61.83±3.35** |
| | Time (s) | | | | | | | | | | | |
| Compared | 0.33 | 0.68 | 25.72 | 81.41 | 741.58 | 369.72 | 632.77 | 859.37 | 1822.58 | 2908.20 | 1104.67 | 1407.88 |
| Proposed | 0.58 | 0.71 | 56.83 | 104.81 | 876.77 | 493.56 | 807.75 | 1030.22 | 2187.04 | 3433.71 | 1141.22 | 1488.66 |

number available for TinyImageNet and YoutubeFace sel is $m = 2\,k$. Generally, our MVBGC-NFF exceeds most existing competitors.

## V. Conclusion

This article investigates the adverse impact of noisy features on bipartite graph clustering and proposes an effective alleviating solution. One crucial experimental insight is that noisy features incur "anchor shift" phenomenon, degrading the bipartite graph's quality and ruining the clustering performance. To remedy the deficiency of conventional MVBGC models that equally treat features, we propose a noisy feature filter to depress the noisy ones, which can alleviate the "anchor shift" issue. Another theoretical contribution is that we mathematically analyze the bounds of proposed feature-adaptive bipartite graph's fuzzy membership. A corollary is that distinguishing features' discrimination will increase the fuzzy membership. Such manner achieves soft partition to remedy the potential inaccurate absolute representation. The effectiveness and efficiency of our model are demonstrated on both synthetic and real-world datasets. This article provides novel and interesting feature-level findings, corroborated by rigorous theoretical analysis and extensive experiments, which facilitate this part of multi-view bipartite graph clustering research.

## References

[1] X. Li, M. Chen, and Q. Wang, "Adaptive consistency propagation method for graph clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 4, pp. 797–802, Apr. 2020.

[2] W. L. Hamilton, "Graph representation learning," *Synth. Lectures Artif. Intell. Mach. Learn.*, vol. 14, no. 3, pp. 1–159, 2020.

[3] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 249–270, Jan. 2022.

[4] J. Wen, N. Han, X. Fang, L. Fei, K. Yan, and S. Zhan, "Low-rank preserving projection via graph regularized reconstruction," *IEEE Trans. Cybern.*, vol. 49, no. 4, pp. 1279–1291, Apr. 2019.

[5] Y. Liu et al., "Deep graph clustering via dual correlation reduction," in *Proc. 36th AAAI Conf. Artif. Intell.*, Vancouver, Canada, 2022, pp. 7603–7611.

[6] Z. Kang, C. Peng, Q. Cheng, and Z. Xu, "Unified spectral clustering with optimal graph," in *Proc. 32th AAAI Conf. Artif. Intell.*, S. A. McIlraith and K. Q. Weinberger, Eds., New Orleans, Louisiana, USA, 2018, pp. 3366–3373.

[7] S. Huang, I. Tsang, Z. Xu, and J. C. Lv, "Measuring diversity in graph learning: A unified framework for structured multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 12, pp. 5869–5883, Dec. 2022.

[8] L. Li et al., "Local sample-weighted multiple kernel clustering with consensus discriminative graph," *IEEE Trans. Neural Netw.*, early access, Jul. 15, 2022, doi: 10.1109/TNNLS.2022.3184970.

[9] J. Zhang et al., "Multiple kernel clustering with dual noise minimization," in *Proc. 30th ACM Int. Conf. Multimedia*, ser. MM '22. New York, NY, USA, 2022, pp. 3440–3450.

[10] F. Nie, W. Zhu, and X. Li, "Unsupervised large graph embedding," in *Proc. 31th AAAI Conf. Artif. Intell.*, San Francisco, California, USA, 2017, pp. 2422–2428.

[11] F. Nie, X. Wang, C. Deng, and H. Huang, "Learning a structured optimal bipartite graph for co-clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 4129–4138.

[12] K. Zhan, C. Niu, C. Chen, F. Nie, C. Zhang, and Y. Yang, "Graph structure fusion for multiview clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 10, pp. 1984–1993, Oct. 2019.

[13] Z. Kang, Z. Lin, X. Zhu, and W. Xu, "Structured graph learning for scalable subspace clustering: From single view to multiview," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 8976–8986, Sep. 2022.

[14] K. Zhan, C. Zhang, J. Guan, and J. Wang, "Graph learning for multiview clustering," *IEEE Trans. Cybern.*, vol. 48, no. 10, pp. 2887–2895, Oct. 2018.

[15] C. Zhang, Y. Cui, Z. Han, J. T. Zhou, H. Fu, and Q. Hu, "Deep partial multi-view learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2402–2415, May 2022.

[16] J. Wen, Y. Xu, and H. Liu, "Incomplete multiview spectral clustering with adaptive graph learning," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1418–1429, Apr. 2020.

[17] F. Nie, J. Li, and X. Li, "Self-weighted multiview clustering with multiple graphs," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Melbourne, Australia, 2017, pp. 2564–2570.

[18] X. Cai, F. Nie, and H. Huang, "Multi-view k-means clustering on big data," in *Proc. 23th Int. Joint Conf. Artif. Intell.*, F. Rossi, Ed., Beijing, China, 2013, pp. 2598–2604.

[19] R. Li, C. Zhang, Q. Hu, P. Zhu, and Z. Wang, "Flexible multi-view representation learning for subspace clustering," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, S. Kraus, Ed., Macao, China, 2019, pp. 2916–2922.

[20] Z. Kang et al., "Partition level multiview subspace clustering," *Neural Netw.*, vol. 122, pp. 279–288, 2020.

[21] M. Chen, L. Huang, C. Wang, and D. Huang, "Multi-view clustering in latent embedding space," in *Proc. 34th AAAI Conf. Artif. Intell.*, New York, NY, USA, Feb. 7–12, 2020, pp. 3513–3520.

[22] S. Huang, Z. Kang, I. W. Tsang, and Z. Xu, "Auto-weighted multi-view clustering via kernelized graph learning," *Pattern Recognit.*, vol. 88, pp. 174–184, 2019.

[23] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The constrained Laplacian rank algorithm for graph-based clustering," in *Proc. 30th AAAI Conf. Artif. Intell.*, Phoenix, Arizona, USA, 2016, pp. 1969–1976.

[24] Z. Ren and Q. Sun, "Simultaneous global and local graph structure preserving for multiple kernel clustering," *IEEE Trans. Neural Netw.*, vol. 32, no. 5, pp. 1839–1851, May 2021.

[25] P. Zhang et al., "Consensus one-step multi-view subspace clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 4676–4689, Oct. 2022.

[26] D. Cai and X. Chen, "Large scale spectral clustering via landmark-based sparse representation," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1669–1680, Aug. 2015.

[27] Y. Li, F. Nie, H. Huang, and J. Huang, "Large-scale multi-view spectral clustering via bipartite graph," in *Proc. 39th AAAI Conf. Artif. Intell.*, B. Bonet and S. Koenig, Eds., Austin, Texas, USA, 2015, pp. 2750–2756.

[28] Z. Kang, W. Zhou, Z. Zhao, J. Shao, M. Han, and Z. Xu, "Large-scale multi-view subspace clustering in linear time," in *Proc. 34th Conf. Artif. Intell.*, New York, NY, USA, 2020, pp. 4412–4419.

[29] X. Li, H. Zhang, R. Wang, and F. Nie, "Multiview clustering: A scalable and parameter-free bipartite graph fusion method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 330–344, Jan. 2022.

[30] L. Li and H. He, "Bipartite graph based multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 7, pp. 3111–3125, Jul. 2022.

[31] F. Nie, W. Chang, R. Wang, and X. Li, "Learning an optimal bipartite graph for subspace clustering via constrained Laplacian rank," *IEEE Trans. Cybern.*, vol. 53, no. 2, pp. 1235–1247, Feb. 2023.

[32] C. Liu, F. Nie, R. Wang, and X. Li, "Scalable fuzzy clustering with anchor graph," *IEEE Trans. Knowl. Data Eng.*, early access, Aug. 22, 2022, doi: 10.1109/TKDE.2022.3200685.

[33] Z. Zhang, L. Liu, F. Shen, H. T. Shen, and L. Shao, "Binary multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1774–1782, Jul. 2019.

[34] F. Nie, C. Liu, R. Wang, Z. Wang, and X. Li, "Fast fuzzy clustering based on anchor graph," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 7, pp. 2375–2387, Jul. 2022.

[35] H. Chen, F. Nie, R. Wang, and X. Li, "Fast unsupervised feature selection with bipartite graph and l2, 0-norm constraint," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 5, pp. 4781–4793, May 2023.

[36] W. Yan, J. Xu, J. Liu, G. Yue, and C. Tang, "Bipartite graph-based discriminative feature learning for multi-view clustering," in *Proc. 30th ACM Int. Conf. Multimedia*, J. Magalh aes, A. D. Bimbo, S. Satoh, N. Sebe, X. Alameda-Pineda, Q. Jin, V. Oria, and L. Toni, Eds., Lisboa, Portugal, Oct. 10–14, 2022, pp. 3403–3411.

[37] X. Lu and S. Feng, "Structure diversity-induced anchor graph fusion for multi-view clustering," *ACM Trans. Knowl. Discov. Data*, vol. 17, pp. 1–18, 2022.

[38] D. Huang, C.-D. Wang, and J.-H. Lai, "Fast multi-view clustering via ensembles: Towards scalability, superiority, and simplicity," *IEEE Trans. Knowl. Data Eng.*, early access, Jan. 13, 2023, doi: 10.1109/TKDE.2023.3236698.

[39] M. Sun et al., "Scalable multi-view subspace clustering with unified anchors," in *Proc. 29th ACM Int. Conf. Multimedia*, ser. MM' 21. New York, NY, USA, 2021, pp. 3528–3536.

[40] S. Wang et al., "Fast parameter-free multi-view subspace clustering with consensus anchor guidance," *IEEE Trans. Image Process.*, vol. 31, pp. 556–568, 2022.

[41] S. Liu et al., "Efficient one-pass multi-view subspace clustering with consensus anchors," in *Proc. 36th AAAI Conf. Artif. Intell.*, Vancouver, Canada, 2022, pp. 7576–7584.

[42] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[43] C. Tang, X. Zheng, W. Zhang, X. Liu, and E. Zhu, "Unsupervised feature selection via multiple graph fusion and feature weight learning," *Sci. China Inf. Sci.*, vol. 66, pp. 1–152101, 2023.

[44] R. Wang, C. Zhang, J. Bian, Z. Wang, F. Nie, and X. Li, "Sparse and flexible projections for unsupervised feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 6362–6375, Apr. 2022.

[45] C. Tang et al., "Cross-view locality preserved diversity and consensus learning for multi-view unsupervised feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 4705–4716, Oct. 2022.

[46] Z. Wang, D. Wu, R. Wang, F. Nie, and F. Wang, "Joint anchor graph embedding and discrete feature scoring for unsupervised feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Nov. 23, 2022, doi: 10.1109/TNNLS.2022.3222466.

[47] M. Sun et al., “Projective multiple kernel subspace clustering,” *IEEE. Trans. Multimedia.*, vol. 24, pp. 2567–2579, 2022.
[48] F. Nie, Z. Wang, L. Tian, R. Wang, and X. Li, “Subspace sparse discriminative feature selection,” *IEEE Trans. Cybern.*, vol. 52, no. 6, pp. 4221–4233, Jun. 2022.
[49] D. Huang, C. Wang, J. Wu, J. Lai, and C. Kwoh, “Ultra-scalable spectral clustering and ensemble clustering,” *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1212–1226, Jun. 2020.
[50] J. C. Bezdek and R. J. Hathaway, “Convergence of alternating optimization,” *Neural Parallel Sci. Comput.*, vol. 11, no. 4, pp. 351–368, Aug. 2003.
[51] J. M. Winn and N. Jojic, “LOCUS: Learning object classes with unsupervised segmentation,” in *Proc. IEEE 10th Int. Conf. Comput. Vis.*, Beijing, China, 2005, pp. 756–763.
[52] F. Nie, J. Li, and X. Li, “Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification,” in *Proc. 25th Int. Joint Conf. Artif. Intell.*, New York, NY, USA, 2016, pp. 1881–1887.
[53] B. Yang, X. Zhang, F. Nie, F. Wang, W. Yu, and R. Wang, “Fast multiview clustering via nonnegative and orthogonal factorization,” *IEEE Trans. Image Process.*, vol. 30, pp. 2575–2586, 2020.
[54] C. Tang, Z. Li, J. Wang, X. Liu, W. Zhang, and E. Zhu, “Unified one-step multi-view spectral clustering,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 6449–6460, early access, May 2022.

**Liang Li** received the bachelor’s degree from the Huazhong University of Science and Technology, Wuhan, China, in 2018, and the master’s degree from the National University of Defense Technology, Changsha, China, in 2020, where he is currently working toward the PhD degree. His current research interests include graph learning, scalable clustering, and incomplete clustering.

**Junpu Zhang** received the bachelor’s degree from the Ocean University of China, Qingdao, China, in 2020. He is currently working toward the master’s degree with the National University of Defense Technology, Changsha, China. His current research interests include kernel learning, ensemble learning, and multiview clustering.

**Siwei Wang** is currently working toward the PhD degree with the National University of Defense Technology, Changsha, China. He has authored or coauthored and served as a reviewer for some highly regarded journals and conferences, such as *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Image Processing (TIP)*, *IEEE Transactions on Cybernetics (TYCB)*, *IEEE Transactions on Multimedia (TMM)*, International Conference on Machine Learning (ICML), Computer Vision and Pattern Recognition (CVPR), European Conference on Computer Vision (ECCV), International Conference on Computer Vision (ICCV), AAAI Conference on Artificial Intelligence (AAAI), and International Joint Conference on Artificial Intelligence (IJCAI). His current research interests include kernel learning, unsupervised multiple-view learning, scalable clustering, and deep unsupervised learning.

**Xinwang Liu** (Senior Member, IEEE) received the PhD degree from the National University of Defense Technology (NUDT), Changsha, China, in 2013. He is currently a full professor with the School of Computer, NUDT. He has authored or coauthored more than 80 peer-reviewed papers, including those in highly regarded journals and conferences, such as the *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, *IEEE Transactions on Image Processing (TIP)*, the *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Multimedia (TMM)*, *IEEE Transactions on Information Forensics and Security (TIFS)*, International Conference on Machine Learning (ICML), NeurIPS, International Conference on Computer Vision (ICCV), Computer Vision and Pattern Recognition (CVPR), AAAI Conference on Artificial Intelligence (AAAI), and International Joint Conference on Artificial Intelligence (IJCAI). His current research interests include kernel learning and unsupervised feature learning. He serves as an associated editor of the Information Fusion Journal and the *IEEE Transactions on Neural Networks and Learning Systems* journal. More information can be found at https://xinwangliu.github.io.

**Kenli Li** (Senior Member, IEEE) received the PhD degree in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 2003. He has authored or coauthored more than 200 research papers in international conferences and journals, such as the *IEEE Transactions on Computers*, *IEEE Transactions on Parallel and Distributed Systems*, and International Conference on Parallel Processing (ICPP). His current research interests include parallel computing, high-performance computing, and grid and cloud computing. He serves on the Editorial Board for *IEEE Transactions on Computers*.

**Keqin Li** (Fellow, IEEE) is currently a SUNY distinguished professor of computer science with the State University of New York, New Paltz, NY, USA. He is also a National distinguished professor with Hunan University, Changsha, China. He has authored or coauthored more than 860 journal articles, book chapters, and refereed conference papers. He holds more than 70 patents announced or authorized by the Chinese National Intellectual Property Administration. His current research interests include cloud computing, fog computing, mobile edge computing, energy-efficient computing and communications, embedded systems, cyber–physical systems, heterogeneous computing systems, Big Data computing, high-performance computing, CPU–GPU hybrid and cooperative computing, computer architectures and systems, computer networking, machine learning, and intelligent and soft computing. He received several best paper awards. He was the chair of many international conferences. He is currently an associate editor of the *ACM Computing Surveys* and the *CCF Transactions on High-Performance Computing*. He has served on the Editorial Board for *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Computers*, *IEEE Transactions on Cloud Computing*, *IEEE Transactions on Services Computing*, and *IEEE Transactions on Sustainable Computing*. He is among the world’s top 5 most influential scientists in parallel and distributed computing based on a composite indicator of the Scopus citation database.

# Appendix of "Multi-view Bipartite Graph Clustering with Coupled Noisy Feature Filter"

Liang Li, Junpu Zhang, Siwei Wang, Xinwang Liu, *Senior Member, IEEE,*
Kenli Li, *Senior Member, IEEE,* and Keqin Li, *Fellow, IEEE*

## 1 Proof of Lemma 1

**Lemma 1.** *Supposing an arithmetic progression* $\mathbf{b}^{(1)} : b_i^{(1)} = b_m - (m-i)\mathrm{M}_1$, $1 \leq i \leq m$, *Eq. (6) in the paper is converted to*

$$\begin{aligned} &\min_{\mathbf{z}} \|\mathbf{z} - \mathbf{b}^{(1)}\|_2^2, \\ &\text{s.t. } \mathbf{z}^\top \mathbf{1}_m = 1, \mathbf{z} \geq \mathbf{0}. \end{aligned} \tag{1}$$

*The* $0$*-norm of* $\mathbf{z}$ *is as follows,*

$$\|\mathbf{z}^{(1)}\|_0 = \min\left(\left\lfloor \sqrt{\left\lceil \frac{2}{\mathrm{M}_1} \right\rceil - \frac{3}{4}} + \frac{1}{2} \right\rfloor, m\right), \tag{2}$$

*where* $\mathbf{z}^{(1)}$ *is the optimal solution of Eq. (1),* $\lfloor x \rfloor$ *and* $\lceil x \rceil$ *donate "floor" and "ceil" operations imposed on* $x$*, respectively.*

*Proof.* According to Eq. (23) in the paper, the analytical solution of Eq. (1) can be simplified by

$$\mathbf{z}^{(1)} = \max\left(\mathbf{b}^{(1)} + c^{(1)}\mathbf{1}_m, 0\right). \tag{3}$$

Since $c^{(1)}$ is a constant related to Lagrangian function in Eq. (19), and $\mathbf{b}^{(1)}$ has been sorted in ascending order, the index of the smallest positive $b_i^{(1)} + c^{(1)}$ can be obtained by

$$i_1 = \min_i \left\{ i | b_i^{(1)} + c^{(1)} > 0 \right\}. \tag{4}$$

Note that $c^{(1)} > -b_{i_1}^{(1)}$. Moreover, it is clear that 1) $i < i_1$, we have $b_i^{(1)} + c^{(1)} \leq 0$, 2) $i \geq i_1$, we have $b_i^{(1)} + c^{(1)} > 0$. Recall the constraint $\mathbf{z}^\top \mathbf{1} = 1$, we have

$$\begin{aligned} \mathbf{z}^\top \mathbf{1} &= \sum_{i=1}^{m} \max\left(b_i^{(1)} + c^{(1)}, 0\right) \\ &> \sum_{i=1}^{m} \max\left(b_i^{(1)} - b_{i_1}^{(1)}, 0\right) \\ &= \sum_{i=i_1}^{m} \left(b_i^{(1)} - b_{i_1}^{(1)}\right) \\ &= \sum_{i=i_1}^{m} \left(i - i_1\right) \mathrm{M}_1 \\ &= \frac{1}{2}\left(m - i_1\right)\left(m - i_1 + 1\right) \mathrm{M}_1, \end{aligned} \tag{5}$$

- *Liang Li, Junpu Zhang, Siwei Wang, and Xinwang Liu are with School of Computer, National University of Defense Technology, Changsha 410073, China (E-mail: liangli@nudt.edu.cn; zhangjunpu@nudt.edu.cn; wangsiwei13@nudt.edu.cn; xinwangliu@nudt.edu.cn).*
- *Kenli Li is with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410073, China, and also with the Supercomputing and Cloud Computing Institute, Hunan University, Changsha 410073, China (E-mail: lkl@hnu.edu.cn).*
- *Keqin Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (E-mail: lik@newpaltz.edu).*

which is equivalent to

$$\begin{aligned} (m-i_1)^2+(m-i_1) &\leq \left\lceil \frac{2}{\mathrm{M}_1} \right\rceil - 1 \\ \left(m-i_1+\frac{1}{2}\right)^2 &\leq \left\lceil \frac{2}{\mathrm{M}_1} \right\rceil - \frac{3}{4} \\ i_1 &\geq m+\frac{1}{2}-\sqrt{\left\lceil \frac{2}{\mathrm{M}_1} \right\rceil - \frac{3}{4}} \end{aligned} \tag{6}$$

We derive that

$$i_1 = \max\left(\left\lceil m+\frac{1}{2}-\sqrt{\left\lceil \frac{2}{\mathrm{M}_1} \right\rceil - \frac{3}{4}} \right\rceil, 1\right). \tag{7}$$

Consequently, the $0$-norm of $\mathbf{z}$ can be formulated as

$$\|\mathbf{z}^{(1)}\|_0 = m-i_1+1 = \min\left(\left\lfloor \sqrt{\left\lceil \frac{2}{\mathrm{M}_1} \right\rceil - \frac{3}{4}} + \frac{1}{2} \right\rfloor, m\right). \tag{8}$$

This completes the proof. □

## 2 Proof of Lemma 2

**Lemma 2.** *Supposing an arithmetic progression $\mathbf{b}^{(2)}: b_i^{(2)} = b_m - (m-i)\mathrm{M}_2$, $1 \leq i \leq m$, Eq. (6) in the paper is converted to*

$$\begin{aligned} &\min_{\mathbf{z}} \|\mathbf{z}-\mathbf{b}^{(2)}\|_2^2, \\ &\text{s.t. } \mathbf{z}^\top \mathbf{1}_m = 1, \mathbf{z} \geq \mathbf{0}. \end{aligned} \tag{9}$$

*The $0$-norm of $\mathbf{z}$ is as follows,*

$$\|\mathbf{z}^{(2)}\|_0 = \min\left(\left\lfloor \sqrt{\left\lceil \frac{2}{\mathrm{M}_2} \right\rceil - \frac{3}{4}} + \frac{1}{2} \right\rfloor, m\right), \tag{10}$$

*where $\mathbf{z}^{(2)}$ denotes the optimal solution of Eq. (9).*

*Proof.* The detailed proof is similar to the aforementioned proof of Lemma 1. Note that the index of the smallest positive $b_i^{(2)} + c^{(2)}$ is as follows,

$$i_2 = \min_i \left\{ i | b_i^{(2)} + c^{(2)} > 0 \right\}. \tag{11}$$

We derive that

$$i_2 = \max\left(\left\lceil m+\frac{1}{2}-\sqrt{\left\lceil \frac{2}{\mathrm{M}_2} \right\rceil - \frac{3}{4}} \right\rceil, 1\right). \tag{12}$$

This completes the proof. □

## 3 Proof of Theorem 1

**Theorem 1.** *$f(\mathrm{M}_1)$ and $f(\mathrm{M}_2)$ is the upper and lower bounds of $\|\mathbf{z}^*\|_0$ learned by equally treating features, i.e.,*

$$f(\mathrm{M}_2) \leq \|\mathbf{z}^*\|_0 \leq f(\mathrm{M}_1). \tag{13}$$

*Proof.* This can be easily proofed by contradiction.

Similar to Eq. (4) and Eq. (11), we define the index of the smallest positive $b_i$ of Eq. (6) in the paper, i.e.,

$$i_0 = \min_i \{ i | b_i + c > 0 \}. \tag{14}$$

Naturally, we turn to proof $i_1 \leq i_0 \leq i_2$.

Supposing $i_0 < i_1$, i.e., $i_0 \leq i_1 - 1$ we have

$$b_{i_0} + c > 0 \Rightarrow b_{i_1-1} + c > 0 \Rightarrow c > -b_{i_1-1}, \tag{15}$$

Recall $i_1$ defined in Eq. (4), we have $b_{i_1-1}^{(1)} + c^{(1)} \leq 0$.

Furthermore, we have

$$\begin{aligned}\mathbf{z}^{*\top}\mathbf{1} &= \sum_{i=i_0}^{m}\left(b_i + c\right) > \sum_{i=i_1}^{m}\left(b_i + c\right) > \sum_{i=i_1}^{m}\left(b_i - b_{i_1-1}\right) \\ &\geq \sum_{i=i_1}^{m}\left(b_i^{(1)} - b_{i_1-1}^{(1)}\right) \geq \sum_{i=i_1}^{m}\left(b_i^{(1)} + c^{(1)}\right) = 1,\end{aligned} \tag{16}$$

which contradicts $\mathbf{z}^{*\top}\mathbf{1} = 1$, so $i_1 \leq i_0$, similarly, we derive $i_0 \leq i_2$. Consequently, we have $i_1 \leq i_0 \leq i_2$, which is equivalent to $f\left(\mathrm{M}_2\right) \leq \|\mathbf{z}^*\|_0 \leq f\left(\mathrm{M}_1\right)$.

This completes the proof. □

# 4 Proof of Theorem 3

**Theorem 3.** *Supposing the rank-$k$ SVD of $\mathbf{E}_p$ is $\mathbf{U\Lambda V}^\top$, where $\mathbf{U} \in \mathbb{R}^{n\times k}$, $\mathbf{\Lambda} \in \mathbb{R}^{k\times k}$, and $\mathbf{V} \in \mathbb{R}^{k\times k}$. The optimal solution of Eq. (19) in the paper is as follows,*

$$\mathbf{A}_p = \mathbf{UV}^\top \tag{17}$$

*Proof.* Supposing $\mathbf{R} = \mathbf{V}^\top\mathbf{A}_p^\top\mathbf{U}$, Eq. (19) in the paper can be rewritten as $\max_{\mathbf{A}_p} \mathrm{Tr}(\mathbf{R\Lambda})$. We have $\mathbf{RR}^\top = \mathbf{V}^\top\mathbf{A}_p^\top\mathbf{UU}^\top\mathbf{A}_p\mathbf{V} = \mathbf{I}$, which means $\mathbf{R}$ is orthogonal. Note that diagonal $\mathbf{\Lambda}$ is composed by the rank-$k$ non-negative singular values $\sigma_j$, we have $\mathrm{Tr}(\mathbf{R\Lambda}) \leq \sum_{j=1}^k \sigma_j$, and the equality holds when $\mathbf{R}$ is an identity matrix. Therefore, Eq. (17) holds.

This completes the proof.

□

# 5 Proof of Theorem 4

**Theorem 4.** *The analytical solution of Eq. (22) in the paper is as follows,*

$$\mathbf{Z}_{:,j} = \max\left(\hat{\mathbf{Z}}_{:,j} + \beta_j\mathbf{1}_n, 0\right), \tag{18}$$

*where $\beta_j$ can be computed by Newton's method efficiently.*

*Proof.* Lagrange multiplier can effectively solve Eq. (22) in the paper. For the $j$-th column of $\mathbf{Z}$, we have

$$\mathcal{L}\left(\mathbf{Z}_{:,j}, \beta_j, \boldsymbol{\eta}_j\right) = \frac{1}{2}\left\|\mathbf{Z}_{:,j} - \hat{\mathbf{Z}}_{:,j}\right\|_2^2 - \beta_j\left(\mathbf{Z}_{:,j}^\top\mathbf{1}_m - 1\right) - \boldsymbol{\eta}_j^\top\mathbf{Z}_{:,j}, \tag{19}$$

where $\beta_j$ and $\boldsymbol{\eta}_j$ are Lagrangian multipliers. According to Karush–Kuhn–Tucker (KKT) conditions,

$$\begin{cases}\mathbf{Z}_{:,j} - \hat{\mathbf{Z}}_{:,j} - \beta_j\mathbf{1}_m - \boldsymbol{\eta}_j = \mathbf{0}, \\ \boldsymbol{\eta}_j \odot \mathbf{Z}_{:,j} = \mathbf{0},\end{cases} \tag{20}$$

where $\odot$ denotes the Hadamard product.

We have

$$\mathbf{Z}_{:,j} = \max\left(\hat{\mathbf{Z}}_{:,j} + \beta_j\mathbf{1}_m, 0\right). \tag{21}$$

Note that $\mathbf{Z}_{:,j}^\top\mathbf{1}_m = 1$, according to Eq. (21), $\mathbf{Z}_{:,j}$ monotonically increases with regard to $\beta_j$, and $\beta_j$ can be computed by Newton's method efficiently.

This completes the proof. □

# 6 Experiment

## 6.1 Clustering Performance

Table 1 reports the metrics comparison of baselines UoMVSC, SDAFG, and ours. Although our model is inferior to UoMVSC on MSRCV1 and ORL in terms of ACC, our model still achieves better performance on most datasets, especially on large-scale datasets, illustrating our superiority.

## 6.2 Convergence and Sensitivity

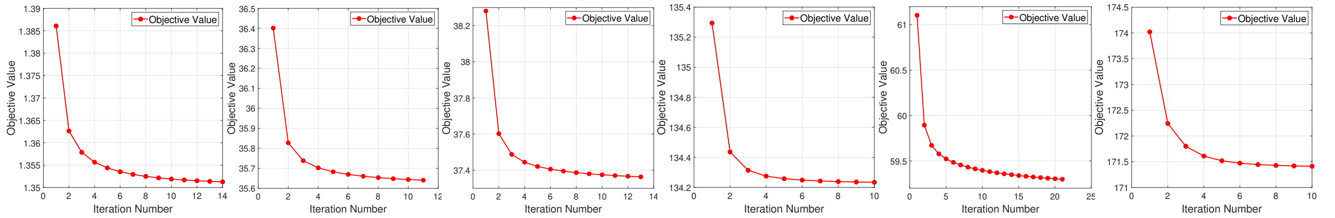
Fig. 1 shows the convergence. Fig. 2 shows sensitivity of anchor number.

## 6.3 The Learned View Weight

Fig. 3 plots the learned view weight $\boldsymbol{\alpha}$ on twelve datasets. We observe that the distribution is non-sparse and varies across multiple views. The results accord with the principle in MVC that multiple views share consistence and keep view-related diversity. Consequently, our MVBGC-NFF achieves view-level fusion.

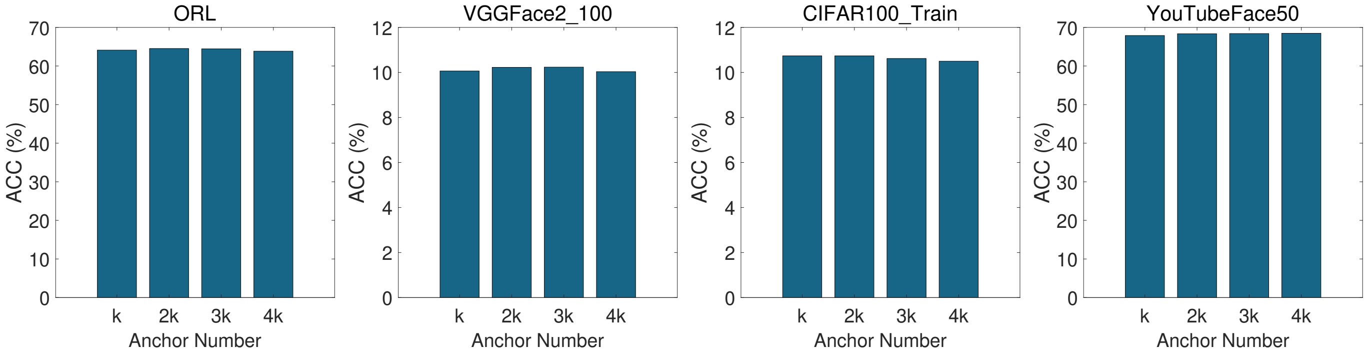
TABLE 1: Clustering metrics comparison. The best ones are marked in bold, “N/A” means memory-overflow errors.

| Datasets | UOMVSC | SDAFG | Proposed | UoMVSC | SDAFG | Proposed | UoMVSC | SDAFG | Proposed | UoMVSC | SDAFG | Proposed |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ACC (%) | | | NMI (%) | | | Purity (%) | | | F-score (%) | |
| MSRCV1 | **86.67±0.00** | 70.95±0.00 | 86.19±7.59 | 80.68±0.00 | 76.23±0.00 | **82.79±5.01** | 86.67±0.00 | 70.95±0.00 | **87.30±6.09** | 77.03±0.00 | 63.58±0.00 | **80.46±7.02** |
| ORL | **67.00±0.00** | 57.75±0.00 | 64.53±3.55 | **82.51±0.00** | 76.89±0.00 | 81.56±1.88 | **70.25±0.00** | 63.00±0.00 | 68.41±3.03 | **55.27±0.00** | 23.11±0.00 | 54.04±3.96 |
| Flower17 | 36.32±0.00 | 8.68±0.00 | **36.61±1.96** | **34.95±0.00** | 5.64±0.00 | 34.45±1.51 | 37.94±0.00 | 09.19±0.00 | **38.15±1.93** | 19.69±0.00 | 11.02±0.00 | **24.51±1.30** |
| VGGFace2_50 | 12.97±0.00 | 3.63±0.00 | **14.55±0.56** | 17.35±0.00 | 2.10±0.00 | **17.97±0.44** | 13.83±0.00 | 3.77±0.00 | **15.65±0.54** | 4.97±0.00 | 4.14±0.00 | **7.46±0.25** |
| Caltech256 | 10.57±0.00 | 6.30±0.00 | **11.36±0.36** | 32.77±0.00 | 13.43±0.00 | **33.60±0.18** | **18.50±0.00** | 7.63±0.00 | 17.78±0.26 | 2.63±0.00 | 1.62±0.00 | **9.24±0.86** |
| VGGFace2_100 | N/A | 2.30±0.00 | **10.24±0.32** | N/A | 3.03±0.00 | **18.47±0.28** | N/A | 2.38±0.00 | **11.31±0.33** | N/A | 2.12±0.00 | **4.33±0.11** |
| CIFAR100_Train | N/A | 2.43±0.00 | **10.74±0.27** | N/A | 5.06±0.00 | **17.14±0.19** | N/A | 2.64±0.00 | **11.96±0.24** | N/A | 2.02±0.00 | **4.62±0.12** |
| CIFAR100_All | N/A | 1.93±0.00 | **10.76±0.27** | N/A | 3.15±0.00 | **16.90±0.22** | N/A | 2.06±0.00 | **12.02±0.23** | N/A | 1.98±0.00 | **4.64±0.11** |
| VGGFace2_200 | N/A | 1.74±0.00 | **6.44±0.18** | N/A | 5.26±0.00 | **18.46±0.18** | N/A | 1.84±0.00 | **7.20±0.18** | N/A | 1.09±0.00 | **2.29±0.06** |
| TinyImageNet | N/A | 1.54±0.00 | **5.08±0.10** | N/A | 6.13±0.00 | **14.56±0.11** | N/A | 1.63±0.00 | **5.71±0.11** | N/A | 1.03±0.00 | **1.66±0.03** |
| YouTubeFace50 | N/A | 62.44±0.00 | **68.48±3.12** | N/A | 77.18±0.00 | **83.80±0.96** | N/A | 67.83±0.00 | **75.04±2.33** | N/A | 24.29±0.00 | **61.83±3.35** |
| YoutubeFace sel | N/A | 27.03±0.00 | **27.15±0.67** | N/A | 3.02±0.00 | **24.01±0.40** | N/A | 27.24±0.00 | **35.85±0.46** | N/A | **16.30±0.00** | 10.92±0.18 |



(a) ORL (b) Caltech256 (c) VGGFace2_100 (d) CIFAR100_Train (e) VGGFace2_200 (f) YouTubeFace50

Fig. 1: The convergence of our model on ORL, Caltech256, VGGFace2_100, CIFAR100_Train, VGGFace2_200, and YouTubeFace50 datasets.



(a) ORL (b) VGGFace2_100 (c) CIFAR100_Train (d) YouTubeFace50

Fig. 2: The sensitivity of anchor number on ORL, VGGFace2_100, CIFAR100_Train, and YouTubeFace50 datasets.

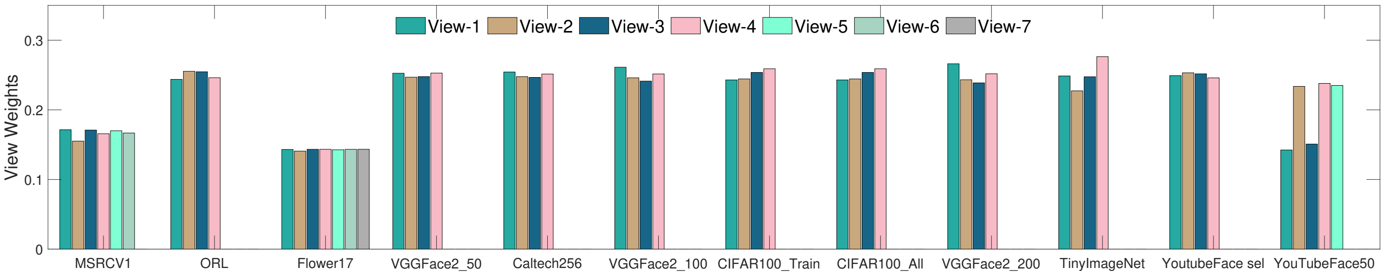


Fig. 3: The learned view weights.