# Self-Learning Symmetric Multi-view Probabilistic Clustering

Junjie Liu, Junlong Liu, Rongxin Jiang, Yaowu Chen, Chen Shen, Jieping Ye, *Fellow, IEEE*



***Abstract*—Multi-view Clustering (MVC) has achieved significant progress, with many efforts dedicated to learn knowledge from multiple views. However, most existing methods are either not applicable or require additional steps for incomplete MVC. Such a limitation results in poor-quality clustering performance and poor missing view adaptation. Besides, noise or outliers might significantly degrade the overall clustering performance, which are not handled well by most existing methods. In this paper, we propose a novel unified framework for incomplete and complete MVC named self-learning symmetric multi-view probabilistic clustering (SLS-MPC). SLS-MPC proposes a novel symmetric multi-view probability estimation and equivalently transforms multi-view pairwise posterior matching probability into composition of each view's individual distribution, which tolerates data missing and might extend to any number of views. Then, SLS-MPC proposes a novel self-learning probability function without any prior knowledge and hyper-parameters to learn each view's individual distribution. Next, graph-context-aware refinement with path propagation and co-neighbor propagation is used to refine pairwise probability, which alleviates the impact of noise and outliers. Finally, SLS-MPC proposes a probabilistic clustering algorithm to adjust clustering assignments by maximizing the joint probability iteratively without category information. Extensive experiments on multiple benchmarks show that SLS-MPC outperforms previous state-of-the-art methods.**

***Index Terms*—Complete and Incomplete Multi-view Clustering, Multi-view Pairwise Posterior Matching Probability, Probabilistic Clustering, Probability Estimation and Refinement.**

## I. Introduction

MULTI-VIEW clustering (MVC) [1] aims at exploiting both correlated and complementary information from multi-view data and improving clustering performance beyond single-view clustering. With the explosion of multi-source and multi-modal data, a great deal of effort has been put into MVC. Different methods have been proposed to handle multi-view data, trying to classify samples into various clusters. Co-Regularization [2], based on co-training, intends to learn classifiers in each view through forms of multi-view regularization. Large-Scale Bipartite Graph [3] fuses local manifold to integrate heterogeneous features and uses bipartite graphs to improve efficiency for large-scale MVC tasks. MKKM [4] proposes an effective matrix-induced regularization to enhance the diversity of the selected kernels, trying to maximize the kernel alignment. BMVC [5] first introduces a compact common binary code space for MVC task to optimize clusters in the hamming space with bit-operations. SMSC [6] seeks to learn the importance of different views and integrates anchor learning and graph construction into a unified framework to capture the complementary information from multiple views.

Despite previous progresses, MVC methods still face various challenges. Absence of partial views among data points [7], [8] might frequently take place in practice, while existing methods are either not applicable [6], [9] or require specific additional steps [10], [11] for these cases. Such a limitation results in poor-quality clustering performance and poor missing view adaptation. Besides, noise or outliers might significantly degrade the overall clustering performance, which are not handled well by most existing methods. Moreover, K-means [12] clustering and spectral [13] clustering are usually used for MVC tasks at the last step. Most of existing methods are less practical in real world cases because they have complex hyper-parameters and use extra information, including but not limited to the number of categories. This information plays an important role in their methods, and the absence of this information either causes their methods to fail or may degrade their clustering performance.

To address these issues, we propose a novel unified framework for incomplete and complete MVC named self-learning symmetric multi-view probabilistic clustering (SLS-MPC). It is difficult and complicated to learn a fusion similarity matrix in a linear or nonlinear manner based on the original similarity matrix. Thus, from a new perspective of probability, we utilize posterior probability to directly measure the probability that two samples belong to the same class. To obtain the posterior probability matrix, SLS-MPC mathematically decomposes it into the formulas of each views' distribution, which can extend to any number of views in an easy way. The proposed multi-view pairwise posterior matching probability is symmetric for each view and tolerates view missing in an intuitive way. Then, equipped with the consistency information excavation in single-view, cross-view and multi-view, a novel self-learning probability function is proposed to effectively learn each view's individual distribution without any prior knowledge and hyper-parameters. Next, SLS-MPC performs graph-context-aware probability refinement with path propagation and co-neighbor propagation, which can effectively alleviate the impact of noise and outliers. Finally, clusters are generated using

Junjie Liu is with the College of Biomedical Engineering and Instrument Science, Zhejiang University and Alibaba Cloud, Hangzhou, China. (e-mail: jumptoliujj@gmail.com).

Junlong Liu, Chen Shen and Jieping Ye are with the Alibaba Cloud, Hangzhou, China. (e-mail: pingwu.ljl@alibaba-inc.com, jason.sc@alibaba-inc.com and yejieping.ye@alibaba-inc.com).

Rongxin Jiang is with the Zhejiang University and the Zhejiang Provincial Key Laboratory for Network Multimedia Technologies (e-mail: rongxinj@zju.edu.cn).

Yaowu Chen is with the Zhejiang University and the Zhejiang University Embedded System Engineering Research Center, Ministry of Education of China (e-mail: cyw@mail.bme.zju.edu.cn).

Corresponding author: Chen Shen (e-mail: jason.sc@alibaba-inc.com).

This work was done when Junjie Liu was a research intern at Alibaba.

the proposed probabilistic clustering algorithm, which is more robustness to noise and does not require the prior knowledge of cluster numbers. Extensive experiments demonstrate that SLS-MPC significantly outperforms state-of-the-art methods.

In summary, the main novelties of this paper are as follows:

- A novel symmetric pairwise posterior matching probability is proposed and SLS-MPC equivalently transforms multi-view pairwise posterior matching probability into compositions of each view's individual distribution, which tolerates data missing and might extend to any number of views.
- To fully dig out the consistency information from multiple views in an unsupervised manner, a novel self-learning probability function is proposed to effectively learn each view's individual distribution without any prior knowledge and hyper-parameters.
- To further alleviate the impact of noise and outliers, a novel graph-context-aware refinement is proposed based on the aspect of graph context.
- Besides, a novel probabilistic clustering algorithm is proposed to generate clustering results in an unsupervised manner without any prior knowledge.
- Extensive experiments on multiple benchmarks for incomplete and complete MVC show that SLS-MPC significantly outperforms previous state-of-the-art methods.

## II. Related Work

A modern MVC method is usually composed of two parts, a consistent representation constructed from all views which is used to learn consensus from multi-view data and a clustering algorithm based on the consistent representation which is used to generate clustering result. Based on the mechanisms and principles used in learning consensus from multiple views, existing MVC algorithms can be grouped into several categories. The first category is based on graph clustering [9], [11], [14], [15]. As a typical graph clustering method, PIC [11] seeks to complete the similarity matrix and learn a consensus matrix and finally performs spectral clustering on the consensus laplacian matrix. GMC [9] weights each view's graph matrix to learn a unified graph matrix. The second one is based on matrix factorization [16]–[21]. This category seeks to learn a consensus representation by performing low-rankness to achieve clustering. For example, MIC [18], based on weighted non-negative matrix factorization and $L_{2,1}$-norm regularization, minimizes the consensus by learning the latent feature matrices for each view. The third one is multiple kernel learning [22]–[25]. In brief, this category seeks to combine different predefined kernels either linearly or non-linearly in order to arrive at a unified kernel. For example, OSLF [25] proposes to learn consensus cluster partition matrix by combing linearly-transformed base partitions obtained from single views. Besides, the methods like [26]–[28] are based on deep multi-view clustering and MCDCF [27] performs multi-layer concept factorization and derives a common consensus representation matrix from the hierarchical information. Moreover, some ensemble-based [29] MVC methods and scalable [6], [30] MVC methods are proposed to advance MVC understanding in new ways. Different from the aforementioned methods, we propose a novel self-learning probability function to effectively learn each view's individual distribution without any prior knowledge and hyper-parameters from the aspect of consistency in single-view, cross-view and multi-view and a novel method to adaptively estimate the posterior matching probability from multiple views without complicated hyper-parameters fine-tuning.

K-means clustering [12], spectral clustering [13], hierarchical clustering [31] and some other traditional clustering algorithms [32], [33] are usually used for clustering tasks. With a given number of clusters $K$, K-means clustering [12] is an iterative algorithm that tries to partition samples into $K$ clusters and makes the intra-cluster data points as similar as possible while also keeping the clusters as far as possible by minimizing the total intra-cluster variance. Spectral clustering [13] uses information from the eigenvalues of similarity matrix derived from the graph and seeks to choose appropriate eigenvectors to cluster different data points. Hierarchical clustering [31] seeks to create a hierarchical clustering tree in which the original data is at the bottom and the root node is at the top. The clustering performance of these algorithms is affected by the optimization parameters and the number of clusters. As one of effective clustering algorithms, probabilistic clustering algorithms [34], [35] are pioneered to incorporate pairwise relations and have achieved state-of-the-art performance in clustering tasks. The basic idea of probabilistic clustering is to maximize the intra-cluster similarities and minimize the inter-cluster similarities among the objects. Empirical functions and weighted confidence or preference are usually used to separate samples, which limits the final clustering performance. Moreover, the matching probability of all pairwise relations are taken into consideration in [34], [35] resulting in high computational complexity. Besides, the number of categories is used in optimization process in some methods and these information plays an important role in their methods [11], [25], without which either causing the failure of their methods or might degrade the performance. Thus, we propose a novel probabilistic clustering algorithm, which has no optimization parameters and generates clustering results in an unsupervised manner and an efficient way without category information.

This work is different from existing methodologies in several key aspects. First, almost all these methods [6], [9]–[11], [20], [21], [25], [27], [28], [30] contain complicated model design, which make them infeasible in real-world applications. In contrast, our SLS-MPC contains an intuitive and efficient clustering framework with multiple clear steps, including symmetric multi-view probability estimation, probability function self-learning, graph-context-aware refinement and probabilistic clustering. Second, different from the works like [9]–[11], our SLS-MPC seeks to adaptively handle multi-view data and missing data from a probabilistic perspective rather than fusing multi-view data using a set of weights, thus embracing higher explainability. In addition, this paper is extended from MPC [36] but differs in the following two aspects. First, multi-view probability estimation has been optimized from an asymmetric form to a symmetric form (Section III-A). This advancement eliminates the inherent issue of view order selection in the asymmetric form in MPC and ensures consistency in the prob-

ability form across all views. Second, MPC utilizes pseudo-labels to independently estimate each view's probability function. However, pseudo-labels may conflict across different views, making it difficult to ensure the consistency between the estimated probability functions. In contrast, our method proposes a novel self-learning probability function (Section III-B) to effectively learn each view's individual distribution from the perspective of consistency of probability function. The proposed self-learning probability function, in conjunction with the other components of our method, constitutes a more robust theoretical framework.

## III. Methodology

### A. Symmetric Multi-view Probability Estimation

Given a multi-view dataset of $N$ samples with $M$ views $S = \{V^{(1)}, V^{(2)}, ..., V^{(M)}\}$. $V^{(m)} \in R^{d^{(m)}*N}$ denotes the feature matrix in $m$-th view, where $d^{(m)}$ is the feature dimension of the $m$-th view. Let $W^{(m)} \in R^{N*N}$ calculated by $V^{(m)}$ using cosine similarity denotes the similarity matrix of the $m$-th view. Assuming that all views are conditionally independent similar to previous works [37]–[41], the pairwise posterior probability of sample $i$ and $j$ proposed in MPC [36] is:

$$\begin{aligned} P(i,j) &= P(e_{ij}=1|w_{ij}^{(1)}, w_{ij}^{(2)}, ..., w_{ij}^{(M)}) \\ &= \frac{(\prod_{m=2}^{M} P(w_{ij}^{(m)}|e_{ij}=1))P(e_{ij}=1|w_{ij}^{(1)})}{\sum_{l\in\{0,1\}}(\prod_{m=2}^{M} P(w_{ij}^{(m)}|e_{ij}=l))P(e_{ij}=l|w_{ij}^{(1)})} \end{aligned} \tag{1}$$

where $e_{ij}$ indicates that the two samples belong to the same class and $w_{ij}^{(m)}$ denotes the similarity of the two samples in $m$-th view. Eq. (1) is asymmetric for each view and has two types of probability function. Considering the consistent representation across multiple views, we further derive the Eq. (1). Let $d_m = w_{ij}^{(m)}, e_1 = (e_{ij}=1), e_0 = (e_{ij}=0)$ for short and Eq. (1) can be expressed as:

$$\begin{aligned} P(i,j) &= P(e_1|d_1, d_2, ..., d_M) \\ &= \frac{(\prod_{m=2}^{M} P(d_m|e_1))P(e_1|d_1)}{\sum_{e\in\{e_0,e_1\}}(\prod_{m=2}^{M} P(d_m|e))P(e|d_1)} \end{aligned} \tag{2}$$

Based on Bayesian formula, $P(d_m|e_1)$ and $P(d_m|e_0)$ can be expressed as:

$$\begin{aligned} P(d_m|e_1) &= \frac{P(e_1|d_m)P(d_m)}{P(e_1)} \\ P(d_m|e_0) &= \frac{P(e_0|d_m)P(d_m)}{P(e_0)} \end{aligned} \tag{3}$$

Naturally, Eq. (2) can be expressed as:

$$\begin{aligned} P(i,j) &= P(e_1|d_1, d_2, ..., d_M) \\ &= \frac{(\prod_{m=2}^{M} \frac{P(e_1|d_m)P(d_m)}{P(e_1)})P(e_1|d_1)}{\sum_{l\in\{0,1\}}(\prod_{m=2}^{M} \frac{P(e_l|d_m)P(d_m)}{P(e_l)})P(e_l|d_1)} \\ &= \frac{(\prod_{m=1}^{M} P(e_1|d_m))P(e_0)^{M-1}}{\sum_{l\in\{0,1\}}(\prod_{m=1}^{M} P(e_l|d_m))P(e_{1-l})^{M-1}} \end{aligned} \tag{4}$$

Thus, the pairwise probability of sample $i$ and $j$ can be expressed as:

$$P(i,j) = \frac{(\prod_{m=1}^{M} P(e_{ij}=1|w_{ij}^{(m)}))P_0}{(\prod_{m=1}^{M} P(e_{ij}=1|w_{ij}^{(m)}))P_0 + (\prod_{m=1}^{M} P(e_{ij}=0|w_{ij}^{(m)}))P_1} \tag{5}$$

where $P_0 = P(e_{ij}=0)^{M-1}$ and $P_1 = P(e_{ij}=1)^{M-1}$. Given sample $i$ and sample $j$ without any prior information, the two samples either belong to the same class or do not belong to the same class, which indicates $P(e_{ij}=0) = P(e_{ij}=1) = 0.5$. Finally, the pairwise probability of sample $i$ and $j$ can be expressed as:

$$P(i,j) = \frac{\prod_{m=1}^{M} P(e_{ij}=1|w_{ij}^{(m)})}{\prod_{m=1}^{M} P(e_{ij}=1|w_{ij}^{(m)}) + \prod_{m=1}^{M} P(e_{ij}=0|w_{ij}^{(m)})} \tag{6}$$

which is symmetric for each view.

### B. Self-Learning Probability Function

Eq. (6) defines the decomposition form and the probability function $P(e_{ij}=1|w_{ij}^{(m)})$ for each view needs to be estimated. A simple way to estimate the probability function is using isotonic regression to fit the pairwise relationship between samples based on pseudo labels (pseudo labels can be generated on each view by a simple clustering algorithm, such as K-means). The performance of the MVC task depends on the quality of the generated pseudo labels. Besides, this simple approach estimates the probability function on each single view, overlooking the important consistent information across multiple views. Thus, to fully dig out the consistency information from multiple views in an unsupervised manner, we propose a self-learning probability function to learn the $P(e_{ij}=1|w_{ij}^{(m)})$ from the aspect of consistency in single-view, cross-view and multi-view without any prior knowledge and hyper-parameters. Fig. 1 illustrates the detailed learning process of the proposed self-learning probability function. And, this section is structured as follows: (1) In Section III-B1, we first introduce the motivation behind the self-learning probability function and provide the definition of consistency. (2) Section III-B2 presents the definitions of multiple probability functions that need to be used in consistency learning defined in the first step. (3) In Section III-B3, we finally design the objective function to learn each view's individual distribution based on the definitions of consistency and multiple probability functions.

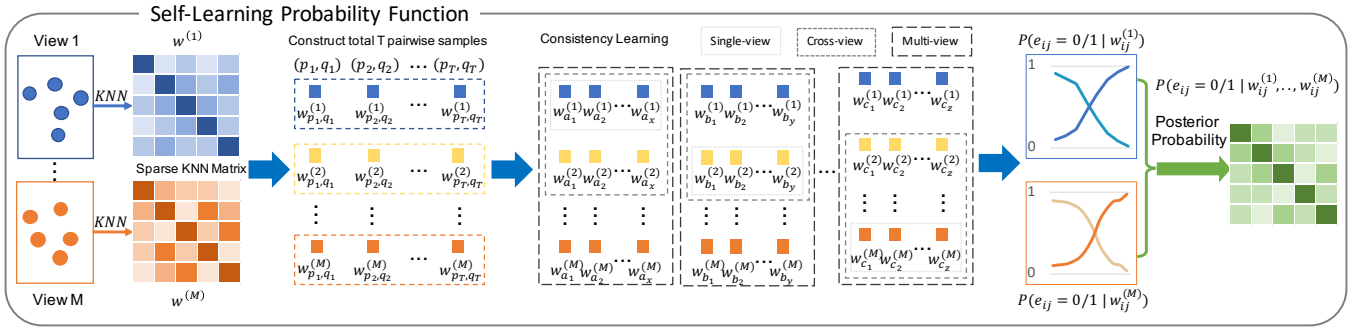



Fig. 1. Illustration of the self-learning probability function. Given a multi-view dataset of $N$ samples with $M$ views $S = \{V^{(1)}, V^{(2)}, ..., V^{(M)}\}$, $KNN^{(m)} \in R^{N*K}$ can be generated on the similarity matrix $W^{(m)} \in R^{N*N}$ of the $m$-th view. $KNN^{(m)}$ construct the training data including total $T$ pairwise samples $(p_t, q_t)$ and the corresponding similarity values $(w^{(1)}_{p_t,q_t}, w^{(2)}_{p_t,q_t}, ..., w^{(M)}_{p_t,q_t})$. We divide each view's data $\{w^{(m)}_{p_t,q_t}\}$ of total $T$ length into $I$ parts in the order of $\{w^{(m)}_{p_t,q_t}\}$ from small to large defined in Eq. (13). $a$, $b$ and $c$ are three specific parts in the total of $I$ parts from three specific views. The light gray dotted boxes represent the single-view forms from different views. The dark gray dotted box represent the cross-view forms from different views. And the black dotted box represent the multi-view forms from different views. The single-view, cross-view and multi-view probability functions are defined in Eq. (14), Eq. (15) and Eq. (17) and the consistency constraint is defined in Eq. (21). Then a self-learning probability function is proposed to learn the $P(e_{ij} = 1|w^{(m)}_{ij})$ from the aspect of consistency in single-view, cross-view and multi-view without any prior knowledge and hyper-parameters. Finally, a multi-view pairwise posterior matching probability matrix is generated from the composition of each view's individual distribution.

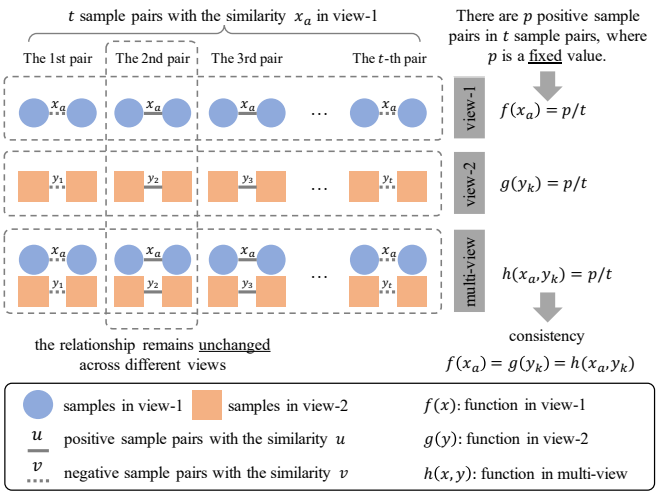



Fig. 2. Our basic observation and motivation of self-learning probability function. Given the condition that the original similarity between the sample pairs in the first view is $x_a$, there are $t$ sample pairs including fixed $p$ positive sample pairs. Fix these $t$ sample pairs and find the original similarity between the sample pairs in the second view ($\{y_k|k \in \{1, ..., t\}\}$). Due to the fixed sample pairs, the probability that the sample pairs belong to the same class in the first view ($f(x_a)$) and the second view ($g(y_k)$) should be consistent. In the same way, the probability that the sample pairs belong to the same class in the first view ($f(x_a)$) and multi-view ($h(x_a, y_k)$) should be also consistent.

*1) **Consistency Motivation and Definition**:* Firstly, we introduce the motivation behind the self-learning probability function. Taking two views as an example, we define the first view $P(e_{ij} = 1|w^{(1)}_{ij})$ as a continuous monotonic function $f(x)$ as:

$$\begin{aligned} &f(x) : P(e_{ij} = 1|w^{(1)}_{ij} = x) \\ &s.t.\ f(x_1) \leq f(x_2), x_1 < x_2, \\ &f(x_{min}) = 0,\ f(x_{max}) = 1 \end{aligned} \tag{7}$$

where $x \in \{w^{(1)}_{i,j}\}$, $x_{min} = min(w^{(1)}_{i,j})$ and $x_{max} = max(w^{(1)}_{i,j})$. In the same way, we define the second view $P(e_{ij} = 1|w^{(2)}_{ij})$ as a continuous monotonic function $g(y)$ and $g(y)$ has the same constraints as $f(x)$ including range and monotonicity. And, the multi-view function $h(x, y)$ based on Eq. (6) is defined as:

$$h(x, y) = \frac{f(x)g(y)}{f(x)g(y) + (1 - f(x))(1 - g(y))} \tag{8}$$

As illustrated in Fig. 2, $f(x_a)$ indicates the probability that the sample pairs belong to the same class given the similarity $x_a$ in the first view. A subset of pairwise samples $s = \{(i, j)|w^{(1)}_{ij} = x_a\}$ contains all pairs of samples with similarity $x_a$ in the first view and the proportion of pairwise samples of the same class in the subset $s$ is a fixed value. Then from the perspective of the second view, $\{(f(x_a), g(y_k))|y_k = w^{(2)}_{ij}, (i, j) \in s\}$ contains the probability that the sample pairs belong to the same class in the subset $s$ from the first view and second view. Due to the fixed number of positive sample pairs in the subset $s$, the probability that the sample pairs belong to the same class in the first view and the second view should be consistent. Thus, we present the cross-view consistency as follows.

***Definition 1:*** The cross-view consistency from the first view ($f(x)$) to the second view ($g(y)$) can be mathematically expressed as:

$$L_{f-g} = D(f(x), \frac{\int g(y)p(x, y)dy}{\int p(x, y)dy}) \tag{9}$$

where $D$ is the distance function and $p(x, y)$ is the similarity distribution between the first view and second view.

***Definition 2:*** The cross-view consistency from the second view ($g(y)$) to the first view ($f(x)$) can be expressed as:

$$L_{g-f} = D(g(y), \frac{\int f(x)p(x, y)dx}{\int p(x, y)dx}) \tag{10}$$

Furthermore, $\{(f(x_a), h(x_a, y_k))|y_k = w^{(2)}_{ij}, (i, j) \in s\}$ contains the probability that the sample pairs belong to the same class in the subset $s$ from the perspective of multi-view.

As illustrated in Fig. 2, the probability that the sample pairs belong to the same class in single-view and multi-view should be also consistent. Thus, we present the multi-view consistency as follows.

***Definition 3:*** The multi-view consistency from the single-view ($f(x)$ and $g(y)$) to multi-view ($h(x,y)$) can be mathematically expressed as:

$$\begin{aligned} L_{f-h} &= D(f(x), \frac{\int h(x,y)p(x,y)dy}{\int p(x,y)dy}) \\ L_{g-h} &= D(g(y), \frac{\int h(x,y)p(x,y)dx}{\int p(x,y)dx}) \end{aligned} \tag{11}$$

Finally, based on *Definition 1-3*, we present the consistency constraint as follows.

***Definition 4:*** To constraint the function $f(x)$ and $g(y)$, consistency constraint can be mathematically expressed as:

$$\begin{aligned} L_{consistency} &= L_{f-g} + L_{g-f} + L_{f-h} + L_{g-h} \\ f, g &= \arg\min_{f,g} L_{consistency} \end{aligned} \tag{12}$$

where $L_{f-g}$ and $L_{g-f}$ are the cross-view constraints, $L_{f-h}$ and $L_{g-h}$ are the multi-view constraints.

*2) **Definitions of Probability Functions**:* Next, we present definitions of multiple probability functions that need to be used in the above consistency definition. As mentioned in Section III-A, given a multi-view dataset of $N$ samples with $M$ views $S = \{V^{(1)}, V^{(2)}, ..., V^{(M)}\}$, $KNN^{(m)} \in R^{N*K}$ can be generated on the similarity matrix $W^{(m)} \in R^{N*N}$ of the $m$-th view. Then $KNN^{(m)}$ construct the training data including total $T$ pairwise samples $(p_t, q_t)$ and the corresponding similarity values $(w^{(1)}_{p_t,q_t}, w^{(2)}_{p_t,q_t}, ..., w^{(M)}_{p_t,q_t})$, $t = 1, 2, ..., T$. Due to the complexity of solution in Eq. (9), Eq. (10) and Eq. (11), we simply use monotonic increasing piecewise function $f^{(m)}(x)$ instead of continuous monotonic function defined in Eq. (7) for approximate solution and the function $f^{(m)}(x)$ is designed as below:

$$\begin{aligned} & f^{(m)}(x): \ (x^{(m)}_{i_m}, f^{(m)}_{i_m}) \\ s.t. \ & x^{(m)}_1 < x^{(m)}_2 < ... < x^{(m)}_I, \\ & f^{(m)}_1 \le f^{(m)}_2 \le ... \le f^{(m)}_I, \\ & f^{(m)}_1 = 0, f^{(m)}_I = 1, \\ & |z^{(m)}_{i_m}| = length(r^{(m)}_{i_m}) = T/I \end{aligned} \tag{13}$$

where $r^{(m)}_{i_m} = \{w^{(m)}_{p_t,q_t} | (x^{(m)}_{i_m} - \Delta_{l^{(m)}_{i_m}}) \le w^{(m)}_{p_t,q_t} < (x^{(m)}_{i_m} + \Delta_{r^{(m)}_{i_m}}), t = 1, 2, ..., T\}$ is the similarity set of the $i_m$-th segment in the $m$-th view, $x^{(m)}_{i_m} = mean(r^{(m)}_{i_m})$, $x^{(m)}_{i_m} + \Delta_{r^{(m)}_{i_m}} = x^{(m)}_{i_m+1} - \Delta_{l^{(m)}_{i_m+1}}$, $m \in [1, 2, ..., M]$, $i_m \in [1, 2, ..., I]$, $I$ is the total segments of piecewise function and we divide the data $\{w^{(m)}_{p_t,q_t}\}$ of total $T$ length into $I$ equal parts in the order of $\{w^{(m)}_{p_t,q_t}\}$ from small to large. With the above definitions, we propose three types of functions including single-view function $Fsingle^{(m)}(x): \ (x^{(m)}_{i_m}, Fsingle^{(m)}_{i_m})$, cross-view function $Fcross^{(m)}(x): \ (x^{(m)}_{i_m}, Fcross^{(m)}_{i_m})$ and multi-view function $Fmulti^{(m)}(x): \ (x^{(m)}_{i_m}, Fmulti^{(m)}_{i_m})$, where $i_m \in [1, 2, ..., I]$.

***Definition 5:*** The single-view function is designed as:

$$Fsingle^{(m)}_{i_m} = f^{(m)}_{i_m} = f^{(m)}(x^{(m)}_{i_m}) \tag{14}$$

where $i_m \in [1, 2, .., I]$ and $m \in [1, 2, ..., M]$.

***Definition 6:*** The cross-view function is designed as below to measure the similarity distribution of another cross view ($m$-th view's cross view $b$):

$$\begin{aligned} Fcross^{(m)-(b)}_{i_m} &= \frac{1}{|z^{(m)}_{i_m}|} \sum_{x \in r^{(m)}_{i_m}, x^{(b)}_{i_b} \in r^{(m)-(b)}_{i_m}} f^{(b)}(x^{(b)}_{i_b} | x) \\ &= \frac{1}{|z^{(m)}_{i_m}|} \sum_{x \in r^{(m)}_{i_m}, x^{(b)}_{i_b} \in r^{(m)-(b)}_{i_m}} f^{(b)}_{i_b} \end{aligned} \tag{15}$$

where $r^{(m)}_{i_m} = \{w^{(m)}_{p_t,q_t} | (x^{(m)}_{i_m} - \Delta_{l^{(m)}_{i_m}}) \le w^{(m)}_{p_t,q_t} < (x^{(m)}_{i_m} + \Delta_{r^{(m)}_{i_m}}), t = 1, 2, ..., T\}$ is the similarity set of the $i_m$-th segment in the $m$-th view, $r^{(m)-(b)}_{i_m} = \{x^{(b)}_{i_x} | (x^{(m)}_{i_m} - \Delta_{l^{(m)}_{i_m}}) \le w^{(m)}_{p_t,q_t} < (x^{(m)}_{i_m} + \Delta_{r^{(m)}_{i_m}}), (x^{(b)}_{i_x} - \Delta_{l^{(b)}_{i_x}}) \le w^{(b)}_{p_t,q_t} < (x^{(b)}_{i_x} + \Delta_{r^{(b)}_{i_x}}), t = 1, 2, ..., T\}$ is the segment set to which the pairwise samples in the $i_m$-th segment in the $m$-th view belongs in the $b$-th view, $|z^{(m)}_{i_m}| = length(r^{(m)}_{i_m})$, $i_m, i_b \in [1, 2, ..., I]$ and $m, b \in [1, 2, ..., M]$.

As designed in Eq. (6), given the pairwise similarity $(x^{(1)}_{i_1}, x^{(2)}_{i_2}, ..., x^{(M)}_{i_M})$ of $M$ views, the joint probability is defined as:

$$\begin{aligned} & Fjoint(x^{(1)}_{i_1}, x^{(2)}_{i_2}, ..., x^{(M)}_{i_M}) \\ &= \frac{\prod_{m=1}^{M} f^{(m)}(x^{(m)}_{i_m})}{\prod_{m=1}^{M} f^{(m)}(x^{(m)}_{i_m}) + \prod_{m=1}^{M} (1 - f^{(m)}(x^{(m)}_{i_m}))} \\ &= \frac{\prod_{m=1}^{M} f^{(m)}_{i_m}}{\prod_{m=1}^{M} f^{(m)}_{i_m} + \prod_{m=1}^{M} (1 - f^{(m)}_{i_m})} \end{aligned} \tag{16}$$

***Definition 7:*** The multi-view function is designed as below to measure the similarity distribution of multiple views:

$$Fmulti^{(m)}_{i_m} = \frac{1}{|z^{(m)}_{i_m}|} \sum_{x \in r^{(m)}_{i_m}} Fjoint(x^{(1)}_{i_1}, x^{(2)}_{i_2}, ..., x^{(M)}_{i_M} | x) \tag{17}$$

where $r^{(m)}_{i_m} = \{w^{(m)}_{p_t,q_t} | (x^{(m)}_{i_m} - \Delta_{l^{(m)}_{i_m}}) \le w^{(m)}_{p_t,q_t} < (x^{(m)}_{i_m} + \Delta_{r^{(m)}_{i_m}}), t = 1, 2, ..., T\}$ is the similarity set of the $i_m$-th segment in the $m$-th view, $x^{(b)}_{i_b} \in r^{(m)-(b)}_{i_m} = \{x^{(b)}_{i_x} | (x^{(m)}_{i_m} - \Delta_{l^{(m)}_{i_m}}) \le w^{(m)}_{p_t,q_t} < (x^{(m)}_{i_m} + \Delta_{r^{(m)}_{i_m}}), (x^{(b)}_{i_x} - \Delta_{l^{(b)}_{i_x}}) \le w^{(b)}_{p_t,q_t} < (x^{(b)}_{i_x} + \Delta_{r^{(b)}_{i_x}}), b \ne m, t = 1, 2, ..., T\}$ is the segment set to which the pairwise samples in the $i_m$-th segment in the $m$-th view belongs in the $b$-th view, $|z^{(m)}_{i_m}| = length(r^{(m)}_{i_m})$, $i_m, i_b \in [1, 2, ..., I]$ and $m, b \in [1, 2, ..., M]$.

*3) **Objective Function**:* With the above definitions of consistency and multiple probability functions, we propose the following objective function to learn each view's individual distribution:

$$L = \lambda L_{consistency} + L_{constraint} \tag{18}$$

where $L_{consistency}$ is consistency loss and $L_{constraint}$ is constraint loss. The parameter $\lambda$ is the balanced factor on $L_{consistency}$ and $L_{constraint}$.

**Consistency Loss.** The consistency loss aims to learn the consistency from multiple views between $Fsingle_{i_m}^{(m)}$, $Fcross_{i_m}^{(m)-(b)}$ and $Fmulti_{i_m}^{(m)}$. Based on Eq. (12), $L_{consistency}$ is defined as:

$$\begin{aligned} L_{consistency} &= \frac{1}{M}\sum_{m}(\frac{1}{I}\sum_{i_m} D(Fsingle_{i_m}^{(m)}, Fmulti_{i_m}^{(m)})) \\ &+ \frac{1}{M}\sum_{m}(\frac{1}{I}\sum_{i_m}\sum_{b\neq m} D(Fsingle_{i_m}^{(m)}, Fcross_{i_m}^{(m)-(b)})) \end{aligned} \tag{19}$$

where $i_m \in [1, 2, .., I]$ and $m \in [1, 2, ..., M]$. Due to the difficulty of consistency loss in Eq. (19) in which $Fsingle$ needs to be constrained by both $Fmulti$ and $Fcross$, the mix function is designed as below for fusion and learning instead of directly learning the consistency between $Fsingle_{i_m}^{(m)}$, $Fcross_{i_m}^{(m)-(b)}$ and $Fmulti_{i_m}^{(m)}$:

$$Fmix_{i_m}^{(m)} = \sqrt{Fmulti_{i_m}^{(m)} \frac{1}{M}(Fsingle_{i_m}^{(m)} + \sum_{b\neq m} Fcross_{i_m}^{(m)-(b)})} \tag{20}$$

where $i_m \in [1, 2, .., I]$ and $m \in [1, 2, ..., M]$. The mix function $Fmix_{i_m}^{(m)}$ is used to constrain the value of $Fsingle_{i_m}^{(m)}$ and $Fcross_{i_m}^{(m)-(b)}$. Lastly, the consistency loss is mathematically designed as:

$$\begin{aligned} L_{consistency1} &= \frac{1}{M}\sum_{m}(\frac{1}{I}\sum_{i_m} D(Fsingle_{i_m}^{(m)}, Fmix_{i_m}^{(m)})) \\ L_{consistency2} &= \frac{1}{M}\sum_{m}(\frac{1}{I}\sum_{i_m}\sum_{b\neq m} D(Fcross_{i_m}^{(m)-(b)}, Fmix_{i_m}^{(m)})) \\ L_{consistency} &= \frac{1}{M}(L_{consistency1} + L_{consistency2}) \end{aligned} \tag{21}$$

where $D$ is the distance function and we use $D(x, y) = (x - y)^2$ in our experiments. The detailed experiments on consistency loss with Eq. (19) and Eq. (21) are listed in Table IX.

**Constraint Loss.** As defined in Eq. (13), the value range of the probability function is 0 to 1 and the constraint loss aims to limit the range of the functions, including single-view function $Fsingle^{(m)}(x) : (x_{i_m}^{(m)}, Fsingle_{i_m}^{(m)})$, cross-view function $Fcross^{(m)}(x) : (x_{i_m}^{(m)}, Fcross_{i_m}^{(m)})$ and multi-view function $Fmulti^{(m)}(x) : (x_{i_m}^{(m)}, Fmulti_{i_m}^{(m)})$. Mathematically, the constraint loss is designed as below to limit the values of functions at the beginning and the end:

$$\begin{aligned} L_{constraint} = \sum_{m}( & \sum_{i_m\in r_i} D(Fmulti_{i_m}^{(m)}, 0) \\ & + \sum_{j_m\in r_j} D(Fmulti_{j_m}^{(m)}, 1) \\ & + \sum_{i_m\in r_i} D(Fsingle_{i_m}^{(m)}, 0) \\ & + \sum_{j_m\in r_j} D(Fsingle_{j_m}^{(m)}, 1) \\ & + \sum_{b\neq m}\sum_{i_m\in r_i} D(Fcross_{i_m}^{(m)-(b)}, 0) \\ & + \sum_{b\neq m}\sum_{j_m\in r_j} D(Fcross_{j_m}^{(m)-(b)}, 1)) \end{aligned} \tag{22}$$

where $r_i = [1, 2, .., indi]$ and $r_j = [I-indj, ..., I-1, I]$. $indi$ and $indj+1$ are the limit width and the detailed parameters are listed in Table II. Specially, there is a monotonic constraint in Eq. (13) which is not included in $L_{constraint}$. For monotonic constraint, we use mandatory constraint to ensure that the functions satisfy monotonicity in the process of iteration.

### C. Graph-context-aware Refinement

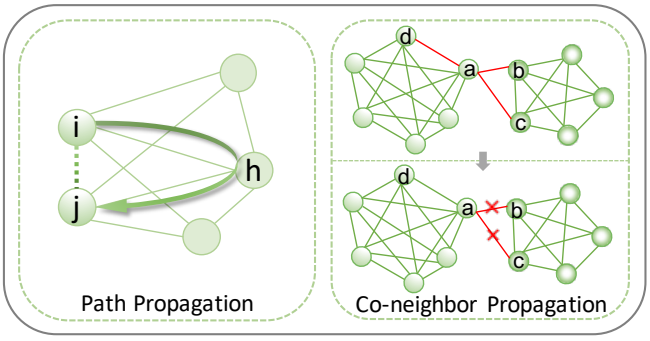



Fig. 3. Illustration of the proposed graph-context-aware refinement including path propagation and co-neighbor propagation. As shown in path propagation, taken probability consistency information into consideration, $h$ sets up the probability path between $i$ and $j$ and the probability between $i$ and $j$ can be enhanced by finding the path with the maximum probability. Besides, in co-neighbor propagation, $b$ and $c$ are the noise in k-nearest-neighbors of $a$. Based on the number of common neighbours and the proportion of the common probabilities, co-neighbor propagation refinement adjusts the probability between $a$ and $b$ and the probability between $a$ and $c$ to a small value and the small value indicates that they are not linked. The probability between $a$ and $d$ can be further adjusted and enhanced.

The probability estimation in Eq. (6) is calculated based on the aspect of sample relationship, overlooking the aspect of graph context which contains rich information. Thus, we perform graph-context-aware refinement with path propagation and co-neighbor propagation to further alleviate the impact of noise and outliers.

Due to the data perturbation of each view, there exists a few outliers in dataset which may affect the clustering performance in the final step. The probability estimation of outliers can not be calculated accurately by using Eq. (6), we therefore try to fine-tune them with path propagation. Inspired by the message passing, where the information among nodes is transmissible, the proposed path propagation passes probabilities between samples like follows:

$$P(i, j) = \max\left(P(i, j), P(i, h) \times P(h, j)\right) \tag{23}$$

where $j \in knn_i$, $h \in knn_{ij}$, $knn_i = \{\cup knn_i^m\}$, $knn_j = \{\cup knn_j^m\}$, $knn_{ij} = \{knn_i \cap knn_j\}$ and $knn_i^m \in R^k$ is the k-nearest-neighbors of sample $i$ in $m$-th view. Fig. 3 shows an intuitive path propagation case, in which sample $h$ sets up the path between sample $i$ and sample $j$ and the probability between sample $i$ and sample $j$ can be enhanced by finding the path with the maximum probability. From the aspect of probability, given three samples (sample $i$, $j$, $h$) and let $a = P(i, j)$, $b = P(i, h), c = P(j, h)$ for short, the probability that sample $i$ and sample $j$ belong to one class is defined as $q = q_p/q_a$, where $q_p = abc+a(1-b)(1-c)$, $q_a = abc+a(1-b)(1-c)+(1-a)(1-b)(1-c)+(1-a)(1-b)c+(1-a)b(1-c)$.

In the formula, $q_a$ denotes the sum of all possibilities and $q_p$ denotes the sum of all possibilities that sample $i$ and sample $j$ belong to one class. Simply given $P(i,j)=0.5$ for a fuzzy probability, it's natural to prove:

$$q=\frac{q_p}{q_a}=\frac{bc+\frac{1}{2}(1-b-c)}{\frac{1}{2}bc+1-\frac{1}{2}b-\frac{1}{2}c}\geq bc \tag{24}$$

where $0<b,c<1$. Using path propagation, the probability consistency information between the outliers and their neighbors is taken into consideration, in which the outliers can be detected and the pairwise probabilities between the outliers and their neighbors can be enhanced.

Besides, the probability estimation is calculated in Euclidean space while the visual features usually lie in low-dimensional manifolds [42]. Only using the information in Euclidean space, overlooking the graph context, may result in inaccuracy of the actual pairwise posterior probabilities between samples. To take advantage of the graph context, the co-neighbor propagation is defined as:

$$P(i,j)=\frac{\sum_{h\in knn_{ij}}(P(i,h)+P(j,h))}{\sum_{h_i\in knn_i}P(i,h_i)+\sum_{h_j\in knn_j}P(j,h_j)} \tag{25}$$

where $knn_i\in R^k$ is the k-nearest-neighbors of sample $i$ calculated by $P(i,j)$ and $knn_{ij}=\{knn_i\cap knn_j\}$. Fig. 3 shows an intuitive co-neighbor propagation case, in which the local graph is constructed by the k-nearest-neighbors of two samples. We take both the number of common neighbours and the proportion of the common probabilities into consideration to further refine the probability based on the local graph. As shown in Eq. (25), the available graph-based probability information can be mined to dig out as much manifold-like distribution information as possible. Using co-neighbor propagation, the noise in k-nearest-neighbors can be detected and the outliers can be further enhanced in an efficient way.

### D. Probabilistic Clustering

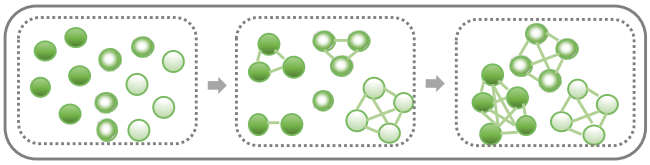


Fig. 4. Illustration of the proposed probabilistic clustering. Each sample is assigned to its own clustering set at the beginning and each sample is moved to the neighbour clustering set in random sequential order by maximizing joint probability iteratively. Finally, a good clustering result can be generated in a convergent way.

Given the estimated self-learning probability function, we can utilize Eq. (6) to calculate the multi-view pairwise posterior matching probability $P(i,j)$ and we can utilize graph-context-aware refinement to further refine the probability $P(i,j)$. Finally, to cluster samples in an unsupervised manner, the probabilistic clustering algorithm is introduced to generate clustering result without any prior knowledge based on the probability $P(i,j)$. Given $N$ samples with the clustering set $\pi:[z_1,z_2,...,z_N]$, the optimization goal of probabilistic clustering can be mathematically expressed as:

$$\begin{aligned}\pi_{opt}&=\underset{\pi}{\operatorname{argmax}}P(X|\pi)=\underset{\pi}{\operatorname{argmax}}\frac{P(X,\pi)}{P(\pi)}\\ s.t.\ P(X,\pi)&=\frac{\prod_{i,j}(\frac{P(e_{ij}=1)}{P(e_{ij}=0)})^{\delta(z_i,z_j)}P(e_{ij}=0)}{\Omega}\end{aligned} \tag{26}$$

where $\delta$ is the Kronecker function and $\Omega$ is the normalization parameter. Besides, there exists an easy-to-understand formula for probabilistic clustering and $P(X,\pi)$ can be mathematically expressed as:

$$P(X,\pi)=\frac{\prod_{i,j}P(e_{ij}=1)^{\delta(z_i,z_j)}P(e_{ij}=0)^{1-\delta(z_i,z_j)}}{\Omega} \tag{27}$$

The basic idea of probabilistic clustering is to maximize the intra-cluster similarities and minimize the inter-cluster similarities among the samples and Eq. (26) and Eq. (27) are equivalent. With the above definitions, the objective optimization function $L=-logP(X|\pi)$ can be expressed as:

$$L=\sum_{i,j}(\delta(z_i,z_j)(logP(e_{ij}=0)-logP(e_{ij}=1)))+c \tag{28}$$

where $c=-\sum_{i,j}(logP(e_{ij}=0))-logP(\pi)-log\Omega$ is a constant. Only the probabilities within the class need to be calculated in Eq. (28), which reduces the computational complexity. The whole probabilistic clustering optimization procedure is outlined in Algorithm 1 and Fig. 4 shows an intuitive clustering process. In the first step, k-nearest-neighbors is constructed using refined multi-view pairwise posterior matching probability. In the second step, each sample is assigned to its own clustering set. Then, in random sequential order, each sample is moved to the neighbour clustering set that results in the minimum value using Eq. (28). The moving procedure is repeated for every sample until no moving steps. With this algorithm, a good clustering result can be generated in a convergent way.

### E. Summary of SLS-MPC

In this section, we summarize the whole framework of SLS-MPC. Firstly, SLS-MPC proposes a self-learning probability function to learn $P(e_{ij}=1|w_{ij}^{(m)})$ using Eq. (18), Eq. (21) and Eq. (22). Then the pairwise posterior probability $P(e_{ij}=0/1)$ of sample $i$ and $j$ is estimated using the proposed symmetric multi-view probability estimation formula in Eq. (6). Next SLS-MPC uses Eq. (23) and Eq. (25) to further refine the pairwise probability based on the the aspect of graph context. Finally, the refined pairwise probability $P(e_{ij}=0/1)$ is used as input to the probabilistic clustering optimization procedure to generate clustering results.

## IV. Experiments

### A. Experimental Settings

**Datasets.** The experimental comparisons are experimentally evaluated on several multi-view datasets. **(1) Handwritten** [43] contains 2000 samples of 10 digits (i.e., digits '0-9'),

**Algorithm 1:** Probabilistic Clustering Optimization Procedure

Input: $P(e_{ij}=1)$ and $P(e_{ij}=0)$;
Construct KNN $nbrs \in R^{n*k}$ by $P(e_{ij}=1)$;
Initialization: $listn = [1, 2, ..., n]$, $it = 0$, $maxiter = 20$, $z = [z_1, z_2, ..., z_n] = [1, 2, .., n]$;
**while** $it < maxiter$ **do**
  $count = 0$
  random shuffle $listn$
  **for** $i$ *in* $listn$ **do**
    find $z_{find}$ in $z[nbrs[i]]$ with minimum objective value denoted by Eq. (28)
    **if** $z_i \,!= z_{find}$ **then**
      update $z_i = z_{find}$
      $count = count + 1$
    **end**
  **end**
  **if** $count == 0$ **then**
    break
  **end**
  $it = it + 1$
**end**
Output: $z$;

**Algorithm 2:** Summary of SLS-MPC

Input: a multi-view dataset of $N$ samples with $M$ views $S = \{V^{(1)}, V^{(2)}, ..., V^{(M)}\}$;
Solution:
1. Construct $KNN^{(m)} \in R^{N*K}$ based on the similarity matrix $W^{(m)} \in R^{N*N}$ of the $m$-th view; Construct the training data including total $T$ pairwise samples $(p_t, q_t)$ and the corresponding similarity values $(w^{(1)}_{p_t,q_t}, w^{(2)}_{p_t,q_t}, ..., w^{(M)}_{p_t,q_t})$;
2. Using Eq. (18), Eq. (21) and Eq. (22) to learn probability function $P(e_{ij}=1|w^{(m)}_{ij})$;
3. Using Eq. (6) to estimate the pairwise posterior probability $P(e_{ij}=0/1)$ of sample $i$ and $j$;
4. Using Eq. (23) and Eq. (25) to further refine the pairwise probability $P(e_{ij}=0/1)$;
5. Using Algorithm 1 to perform probabilistic clustering based on the refined pairwise probability $P(e_{ij}=0/1)$ and generate clustering results $z$;
Output: $z$;

TABLE I
SUMMARY OF THE DATASETS. $\{M, C, N, d^{(m)}\}$ DENOTES THE NUMBER OF {VIEWS, CLUSTERS, SAMPLES, FEATURES} IN EACH VIEW, RESPECTIVELY.

| Datasets | $M$ | $C$ | $N$ | $d^{(m)}(m=1,...,M)$ |
|---|---|---|---|---|
| Handwritten | 4 | 10 | 2000 | 240,76,47,64 |
| 100Leaves | 2 | 100 | 1600 | 64,64 |
| Humbi240 | 2 | 240 | 13440 | 256,256 |
| BUAA | 2 | 150 | 1350 | 100,100 |
| BBCSport | 2 | 5 | 544 | 3181,3202 |

TABLE II
THE DETAILED SETTINGS OF $I, indi, indj$ AND $\lambda$.

| Datasets | $I$ | $indi$ | $indj+1$ | $\lambda$ |
|---|---|---|---|---|
| Handwritten view1-4 | 1000 | 10 | 4 | 80 |
| Handwritten view1-2 | 1000 | 10 | 4 | 20 |
| 100Leaves | 200 | 10 | 2 | 2 |
| Humbi240 | 1000 | 10 | 4 | 20 |
| BUAA | 200 | 10 | 4 | 20 |
| BBCSport | 200 | 10 | 4 | 20 |

covering four kinds of features, which are average pixels features, Fourier coefficient features, Zernike moments features and Karhunen-Love coefficient features. **(2) 100Leaves** [44] contains 1600 samples from 100 plant species. For each sample, a shape descriptor and texture histogram are given. **(3) Humbi240**, a subset of Humbi [45] dataset, contains 13440 samples of 240 persons covering face features extracted by face recognition model[1] and body features extracted by person reID model[2]. **(4) BUAA-visnir face dataset (BUAA)** [46] contains 1350 visual images and 1350 near infrared images of the 150 volunteers. **(5) BBCSport**[3] contains 544 samples of 5 categories. The feature dimensions of the two views used in experiments are 3181 and 3202 respectively. The datasets are summarized in Table I. To evaluate the clustering performance on incomplete data, we select $c\%$ ($c = 90, 70, 50, 30$) samples as the paired samples that have full views. For the remaining samples, half of them miss the first view, while the second view of the other half is removed. The missing rate is defined as $\eta = 1 - c$.

**Evaluation Metrics.** In the experiments, several widely-used clustering metrics including BCubed Fmeasure, Pairwise Fmeasure [47], Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI) are used as the evaluation metrics. A higher value of these metrics indicates a better clustering performance.

**Implementation Details.** We implement our SLS-MPC in PyTorch 1.2 [48] and perform all evaluations on a standard Linux OS with 16 2.50GHz Intel Xeon Platinum 8163 CPUs. The self-learning probability function of each view is initialized as a uniform line from 0 to 1 and the self-learning probability function is trained by SGD with a learning rate of 0.001, a momentum of 0.9 and a weight decay of 0.00005. The detailed settings of $I, indi, indj$ and $\lambda$ are listed in Table II. The setting of $I$ takes into account the size of training data $T$.

### *B. Compared Methods*

We compare our method with SOTA multi-view clustering algorithms. SMSC [6], GMC [9], MCDCF [27] and SFMC [30] could only handle complete multi-view data and thus we fill the missing data with the mean values of the same view following previous work [28] for incomplete clustering cases. PIC [11], OSLF [25], EEIMC [7], UEAF [10], IMCCP [28] and MPC [36] are six compared methods for complete and incomplete clustering cases. For all methods, we download

[1]https://github.com/XiaohangZhan/face_recognition_framework
[2]https://github.com/layumi/Person_reID_baseline_pytorch
[3]http://mlg.ucd.ie/datasets/segment.html

TABLE III
THE CLUSTERING PERFORMANCE COMPARISONS ON THREE DATASETS. MVC INDICATES COMPLETE MULTI-VIEW CLUSTERING; IMVC INDICATES INCOMPLETE MULTI-VIEW CLUSTERING WITH 0.5 MISSING RATE.

| Type | Methods | Handwritten | | | | 100Leaves | | | | Humbi240 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $F_P$ | $F_B$ | NMI | ARI | $F_P$ | $F_B$ | NMI | ARI | $F_P$ | $F_B$ | NMI | ARI |
| MVC | MCDCF [27] | 54.92 | 59.32 | 64.90 | 49.45 | 51.04 | 58.14 | 82.20 | 50.52 | 53.16 | 67.99 | 88.91 | 52.91 |
| | SMSC [6] | 67.48 | 69.20 | 72.54 | 63.83 | 25.88 | 42.12 | 72.59 | 24.77 | 26.59 | 44.37 | 74.09 | 26.13 |
| | SFMC [30] | 72.70 | 73.72 | 77.35 | 69.66 | 29.97 | 61.31 | 80.97 | 28.94 | 51.78 | 91.19 | 95.47 | 51.50 |
| | IMCCP [28] | 76.56 | 80.96 | 83.86 | 73.73 | 22.91 | 36.20 | 69.94 | 21.78 | 49.68 | 58.43 | 88.42 | 49.37 |
| | GMC [9] | 74.84 | 80.47 | 82.20 | 71.75 | 36.40 | 78.98 | 88.75 | 35.47 | 87.99 | 96.05 | 98.57 | 87.94 |
| | OSLF [25] | 78.24 | 78.55 | 79.32 | 75.82 | 65.55 | 69.59 | 87.68 | 65.20 | 90.35 | 93.62 | 98.20 | 90.31 |
| | EEIMC [7] | 78.86 | 79.13 | 80.80 | 76.51 | 74.10 | 77.53 | 91.18 | 73.84 | 91.45 | 94.45 | 98.54 | 91.41 |
| | UEAF [10] | 80.61 | 80.92 | 81.43 | 78.46 | 64.54 | 72.81 | 89.18 | 64.16 | 86.36 | 90.36 | 97.11 | 86.30 |
| | PIC [11] | 76.61 | 77.88 | 80.23 | 73.94 | 78.04 | 81.49 | 92.76 | 77.82 | 94.34 | 96.29 | 98.95 | 94.32 |
| | MPC [36] | 84.57 | 84.45 | 85.60 | 83.04 | 84.18 | 85.65 | 94.40 | 84.04 | 95.49 | 97.03 | 99.07 | 95.47 |
| | SLS-MPC | **87.03** | **86.51** | **87.62** | **85.73** | **85.46** | **86.39** | **95.03** | **85.34** | **98.12** | **98.77** | **99.62** | **98.11** |
| IMVC | MCDCF [27] | 20.84 | 22.99 | 25.38 | 11.38 | 23.84 | 30.61 | 68.36 | 23.06 | 29.91 | 41.78 | 71.44 | 29.53 |
| | SMSC [6] | 62.83 | 63.26 | 65.65 | 58.65 | 17.51 | 30.59 | 63.26 | 16.27 | 18.69 | 31.59 | 64.42 | 18.17 |
| | SFMC [30] | 54.81 | 67.30 | 71.99 | 47.53 | 22.67 | 51.94 | 73.81 | 21.50 | 7.61 | 71.73 | 81.66 | 6.88 |
| | IMCCP [28] | 58.52 | 71.10 | 72.68 | 52.71 | 17.08 | 24.75 | 60.84 | 15.99 | 37.20 | 42.66 | 80.93 | 36.84 |
| | GMC [9] | 53.56 | 73.19 | 73.56 | 46.05 | 3.55 | 47.35 | 56.76 | 1.76 | 2.55 | 52.86 | 65.28 | 1.75 |
| | OSLF [25] | 53.86 | 54.06 | 58.51 | 48.73 | 33.86 | 39.04 | 71.84 | 33.19 | 70.72 | 73.40 | 89.41 | 70.59 |
| | EEIMC [7] | 68.80 | 69.48 | 70.26 | 65.33 | 52.65 | 56.74 | 81.11 | 52.18 | 80.94 | 86.24 | 94.84 | 80.86 |
| | UEAF [10] | 68.94 | 69.48 | 72.55 | 65.48 | 38.47 | 45.87 | 75.62 | 37.82 | 86.04 | 89.96 | 96.81 | 85.98 |
| | PIC [11] | 75.65 | 76.03 | 76.67 | 72.95 | 50.79 | 55.61 | 80.72 | 50.30 | 83.30 | 85.74 | 94.64 | 83.23 |
| | MPC [36] | 77.44 | 77.65 | 78.52 | 75.13 | 58.31 | 61.19 | 83.39 | 57.94 | 90.10 | 91.56 | 96.53 | 90.06 |
| | SLS-MPC | **77.80** | **78.65** | **79.62** | **75.46** | **59.91** | **62.87** | **84.16** | **59.56** | **92.69** | **94.02** | **97.55** | **92.66** |

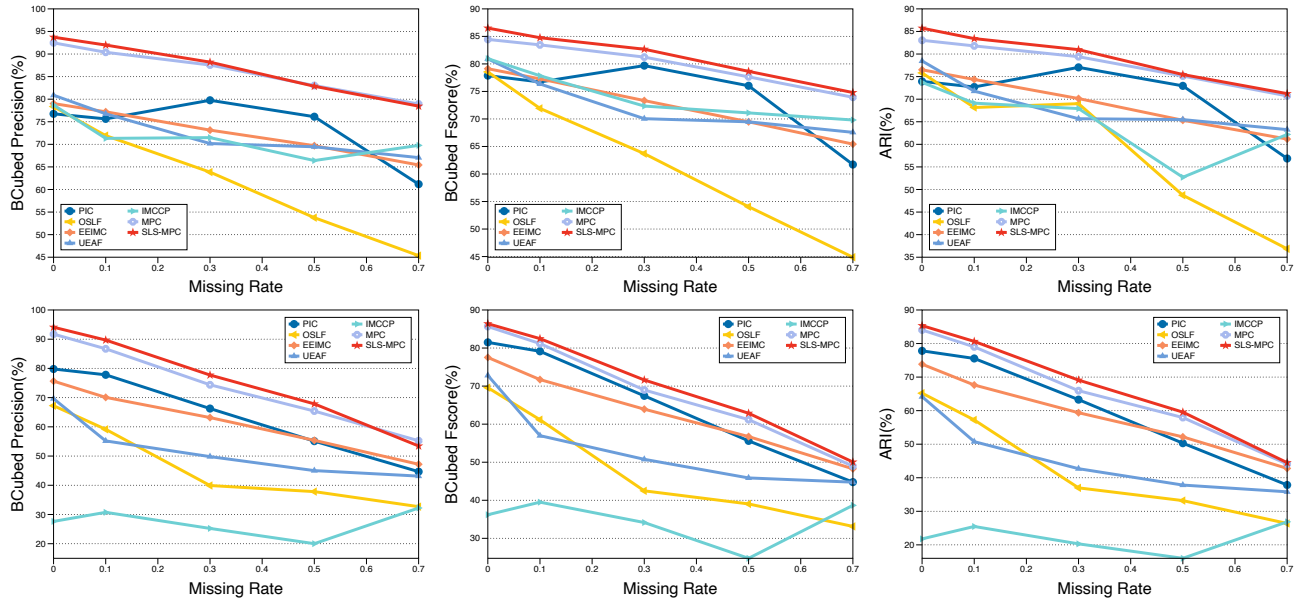



Fig. 5. The clustering performance comparisons on Handwritten and 100Leaves with different missing rates. Three comparisons in the first row are experiments on Handwritten. Three comparisons in the second row are experiments on 100Leaves.

their released codes and tune the hyper-parameters by grid search to generate the best possible results on each dataset.

**Performance Comparison with Two Views.** Table III lists the experimental results of different methods on Handwritten, 100Leaves and Humbi240. In the complete cases, our proposed SLS-MPC achieves the best performance and surpasses the best baseline by 2.69% on Handwritten, 1.30% on 100Leaves and 2.64% on Humbi240 in terms of ARI. Moreover, in the incomplete cases, SLS-MPC surpasses the SOTA by 0.33% on Handwritten, 1.62% on 100Leaves and 2.60% on Humbi240 in terms of ARI. Table V lists the experimental results of different methods on BUAA and BBCSport and our method surpasses almost all tested baselines in terms of BCubed Precision and Fscore. Furthermore, the incomplete multi-view clustering performance with different missing rates on Handwritten and 100Leaves are shown in Fig. 5. From these experimental results, we can observe the following points: (1) our proposed SLS-MPC outperforms all the tested baselines with different missing rates, which demonstrates SLS-MPC's adaptability to different missing rates; (2) SLS-MPC achieves the best precision with almost different missing rates, which further proves the accuracy of self-learning probability function and

TABLE IV
THE CLUSTERING PERFORMANCE COMPARISONS ON HANDWRITTEN WITH 4 VIEWS. VIEW 1 AND VIEW 2 ARE COMPLETE AND VIEW 3 AND VIEW 4 ARE 50% MISSING IN THE INCOMPLETE CASES.

| Type | Methods | Pairwise Fmeasure | | | BCubed Fmeasure | | | NMI | ARI |
|---|---|---|---|---|---|---|---|---|---|
| | | Precision | Recall | Fscore | Precision | Recall | Fscore | | |
| MVC | OSLF [25] | 76.23 | 76.58 | 76.40 | 76.28 | 76.70 | 76.49 | 76.51 | 73.79 |
| | EEIMC [7] | 75.33 | 76.39 | 75.86 | 76.53 | 76.51 | 76.52 | 78.28 | 73.17 |
| | PIC [11] | 80.76 | 80.91 | 80.84 | 81.28 | 81.01 | 81.14 | 83.26 | 78.72 |
| | UEAF [10] | 81.59 | 82.25 | 81.92 | 82.57 | 82.34 | 82.45 | 83.00 | 79.91 |
| | IMCCP [28] | - | - | - | - | - | - | - | - |
| | MPC [36] | 95.85 | 85.12 | 90.17 | 94.89 | 85.19 | 89.78 | 89.77 | 89.15 |
| | SLS-MPC | **96.51** | **90.25** | **93.28** | **95.85** | **90.30** | **92.99** | **92.13** | **92.56** |
| IMVC | OSLF [25] | 62.25 | 67.05 | 64.56 | 64.61 | 67.21 | 65.88 | 69.75 | 60.48 |
| | EEIMC [7] | 73.93 | 78.60 | 78.26 | 78.88 | 78.71 | 78.79 | 79.53 | 75.85 |
| | PIC [11] | 77.24 | 79.72 | 78.46 | 78.83 | 79.82 | 79.32 | 81.34 | 76.04 |
| | UEAF [10] | 81.31 | 81.77 | 81.54 | 81.90 | 81.86 | 81.88 | 82.39 | 79.49 |
| | IMCCP [28] | - | - | - | - | - | - | - | - |
| | MPC [36] | 95.42 | 83.84 | 89.26 | 94.09 | 83.93 | 88.72 | 88.70 | 88.16 |
| | SLS-MPC | **96.77** | **87.18** | **91.73** | **96.00** | **87.25** | **91.42** | **90.92** | **90.86** |

symmetric multi-view probability estimation in our proposed method.

TABLE V
THE CLUSTERING PERFORMANCE OF BCUBED PRECISION $Pre_B$ AND FSCORE $F_B$ COMPARISONS ON BUAA AND BBCSPORT. MVC INDICATES COMPLETE MULTI-VIEW CLUSTERING; IMVC INDICATES INCOMPLETE MULTI-VIEW CLUSTERING WITH 0.5 MISSING RATE.

| Type | Methods | BUAA | | BBCSport | |
|---|---|---|---|---|---|
| | | Precision | Fscore | Precision | Fscore |
| MVC | IMCCP [28] | 39.29 | 39.74 | 28.67 | 35.42 |
| | OSLF [25] | 23.39 | 24.75 | 86.04 | 86.01 |
| | EEIMC [7] | 34.09 | 34.49 | 76.87 | 73.71 |
| | UEAF [10] | 28.46 | 29.59 | 82.69 | 83.88 |
| | PIC [11] | 44.25 | 43.65 | 90.41 | 90.39 |
| | MPC [36] | 58.36 | 44.52 | **95.52** | 93.84 |
| | SLS-MPC | **79.22** | **49.50** | 95.04 | **94.68** |
| IMVC | IMCCP [28] | 32.50 | 32.94 | 25.13 | 34.20 |
| | OSLF [25] | 30.55 | 31.08 | 66.00 | 63.75 |
| | EEIMC [7] | 32.33 | 32.73 | 76.63 | 74.88 |
| | UEAF [10] | 29.02 | 30.05 | 87.51 | 87.20 |
| | PIC [11] | 35.02 | 35.46 | 86.80 | 86.96 |
| | MPC [36] | 40.56 | 36.84 | 88.45 | 88.34 |
| | SLS-MPC | **44.88** | **39.25** | **91.01** | **90.44** |

**Performance Comparison with Four Views.** For the Handwritten dataset, additional incomplete case is constructed in which all samples have two complete views (the first view and the second view) and half of them miss the third view, while the other half of the samples remove the fourth view. As shown in Table IV, SLS-MPC significantly outperforms these state-of-the-art methods and SLS-MPC surpasses the best baseline by 3.41% and 2.70% in terms of ARI in complete case and incomplete case, respectively. The encouraging performance demonstrates SLS-MPC's capacity of extending to multiple views and self-learning capacity of probability function in multi-view information excavation. IMCCP can only handle two views, so the result of IMCCP is not listed in Table IV. Specially in this case, view completion is introduced to handle data missing $Fjoint(x_{i_1}^{(1)}, x_{i_2}^{(2)})$ with only two views, $Fjoint(x_{i_1}^{(1)}, x_{i_2}^{(2)}, x_{i_3}^{(3)})$ and $Fjoint(x_{i_1}^{(1)}, x_{i_2}^{(2)}, x_{i_4}^{(4)})$ with only three views. The pairwise probability is defined as in this case:

$$Fjoint(x_{i_1}^{(1)}, x_{i_2}^{(2)}, x_{i_3}^{(3)}) = \frac{f_{i_1}^{(1)} f_{i_2}^{(2)} f_{i_3}^{(3)} f_c^{(4)}}{f_{i_1}^{(1)} f_{i_2}^{(2)} f_{i_3}^{(3)} f_c^{(4)} + (1-f_{i_1}^{(1)})(1-f_{i_2}^{(2)})(1-f_{i_3}^{(3)})(1-f_c^{(4)})} \tag{29}$$

$$Fjoint(x_{i_1}^{(1)}, x_{i_2}^{(2)}, x_{i_4}^{(4)}) = \frac{f_{i_1}^{(1)} f_{i_2}^{(2)} f_c^{(3)} f_{i_4}^{(4)}}{f_{i_1}^{(1)} f_{i_2}^{(2)} f_c^{(3)} f_{i_4}^{(4)} + (1-f_{i_1}^{(1)})(1-f_{i_2}^{(2)})(1-f_c^{(3)})(1-f_{i_4}^{(4)})} \tag{30}$$

$$Fjoint(x_{i_1}^{(1)}, x_{i_2}^{(2)}) = \frac{f_{i_1}^{(1)} f_{i_2}^{(2)} f_c^{(3)} f_c^{(4)}}{f_{i_1}^{(1)} f_{i_2}^{(2)} f_c^{(3)} f_c^{(4)} + (1-f_{i_1}^{(1)})(1-f_{i_2}^{(2)})(1-f_c^{(3)})(1-f_c^{(4)})} \tag{31}$$

where $f_c^{(3)} = \sqrt{Fcross_{i_1}^{(1)-(3)} Fcross_{i_2}^{(2)-(3)}}$ and $f_c^{(4)} = \sqrt{Fcross_{i_1}^{(1)-(4)} Fcross_{i_2}^{(2)-(4)}}$ are the completion views constructed from cross-view functions. The detailed view completion experiments are listed in Table VIII. Equipped with view completion, the clustering performance has been improved by about 0.6%-0.8%, proving the effectiveness of consistency learning and view completion.

### C. Ablation Studies And Parameter Analysis

In this section, we conduct some studies on several datasets in the following.

**Ablation on Probability Estimation.** In the probability estimation, we use Eq. (6) to fuse the probability information of each view. In Table VI, we compare the formula with different aggregation functions on Handwritten with two views and four views. And the aggregation function is expressed as: $P(i,j) = Aggregation(P(e_{ij}=1|w^{(1)}), P(e_{ij}=1|w^{(2)}), ..., P(e_{ij}=1|w^{(M)}))$, where aggregation functions include mean, max, min and multiply. The mean function treats multiple views as equally important and cannot generate good clustering result. Compared with the naive max function, SLS-MPC using the formula in Eq. (6) can significantly boost the ARI from 78.25

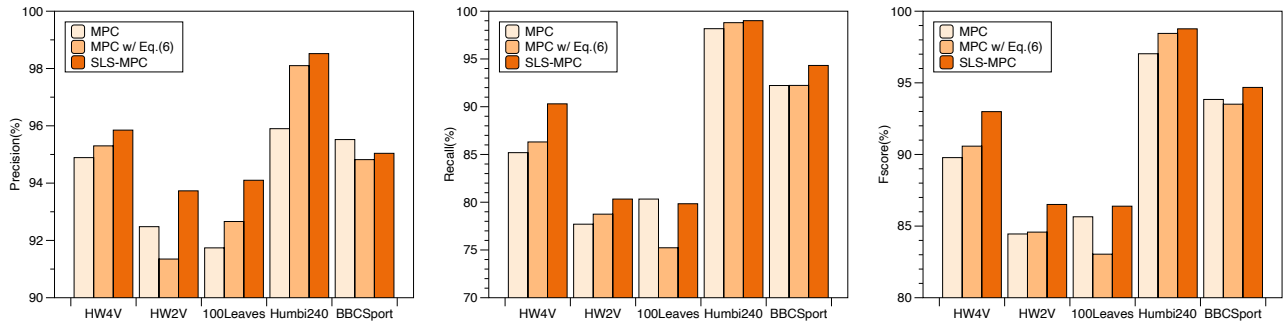



Fig. 6. Ablation study of our method. Comparison on probability estimation between MPC, MPC w/ Eq. (6) and SLS-MPC.

to 92.56 on handwritten with four views. It further proves that Eq. (6) can adaptively estimate the posterior matching probability from multiple views. From the perspective of multi-view probability estimation, we compare our method with MPC and MPC using Eq. (6) in Fig. 6. The performance of MPC using Eq. (6) is about 0.80% higher than that of MPC on Handwritten with four views in terms of BCubed Fscore. And the performance of SLS-MPC is about 2.41% higher than that of MPC using Eq. (6) on Handwritten with four views in terms of BCubed Fscore. These experimental results prove that our formula proposed in Eq. (6) can adaptively fuse multi-view probability information in an efficient way, which plays a major role in performance improvement.

TABLE VI
ABLATION STUDY OF OUR METHOD. COMPARISON BETWEEN THE FORMULA AND THE DIFFERENT AGGREGATION FUNCTIONS ON HUMBI240 AND HANDWRITTEN.

| Datasets | Methods | FP | FB | NMI | ARI |
|---|---|---|---|---|---|
| Humbi240 | max | 87.67 | 89.67 | 96.39 | 87.62 |
| | mean | 92.14 | 93.46 | 97.71 | 92.11 |
| | min | 97.2 | 98.04 | 99.37 | 97.19 |
| | multiply | 97.26 | 98.03 | 99.35 | 97.25 |
| | formula | **98.12** | **98.77** | **99.62** | **98.11** |
| Handwritten view 1-2 | max | 73.86 | 74.33 | 80.18 | 71.45 |
| | mean | 80.6 | 80.35 | 83.52 | 78.79 |
| | min | 83.83 | 82.99 | 84.42 | 82.24 |
| | multiply | 84.16 | 83.45 | 85.01 | 82.61 |
| | formula | **87.03** | **86.51** | **87.62** | **85.73** |
| Handwritten view 1-4 | max | 80.15 | 79.76 | 83.56 | 78.25 |
| | mean | 88.96 | 88.53 | 88.46 | 87.83 |
| | min | 87.21 | 86.64 | 86.80 | 85.90 |
| | multiply | 90.75 | 90.23 | 89.69 | 89.78 |
| | formula | **93.28** | **92.99** | **92.13** | **92.56** |

**Ablation on Similarity Measures.** To keep consistent with previous works MPC [36], PIC [11] and UEAF [10], we use cosine metric to estimate the similarity matrix. As listed in Table VII and Table VIII, we report the clustering performance in complete cases and incomplete cases obtained using similarity metric $L_p$, where $L_p(x_i, x_j) = (\sum_{l=1}^{n} |x_i^{(l)} - x_j^{(l)}|^p)^{\frac{1}{p}}, x_i = (x_i^{(1)}, ..., x_i^{(n)})$. Overall, SLS-MPC is robust to the choice of metric and the performance using cosine metric is more stable than that of $L_p$.

**Ablation on Consistency Loss.** As described in Section Self-Learning Probability Function, consistency loss Eq. (19) and Eq. (21) are introduced in self-learning to learn probability

TABLE VII
ABLATION STUDY OF OUR METHOD. COMPARISON BETWEEN DIFFERENT SIMILARITY MEASURES IN THE COMPLETE CASES.

| Datasets | Methods | FP | FB | NMI | ARI |
|---|---|---|---|---|---|
| 100Leaves | $L_1$ | **89.65** | **90.60** | **96.49** | **89.56** |
| | $L_2$ | 84.35 | 85.70 | 94.83 | 84.22 |
| | $L_3$ | 79.35 | 80.64 | 92.96 | 79.18 |
| | Cosine | 85.46 | 86.39 | 95.03 | 85.34 |
| Humbi240 | $L_1$ | 96.94 | 98.09 | 99.44 | 96.93 |
| | $L_2$ | 97.75 | 98.60 | 99.59 | 97.74 |
| | $L_3$ | 97.79 | 98.63 | 99.60 | 97.78 |
| | Cosine | **98.12** | **98.77** | **99.62** | **98.11** |
| Handwritten view 1-2 | $L_1$ | 86.08 | 85.55 | 87.04 | 84.69 |
| | $L_2$ | 86.91 | 86.44 | 87.47 | 85.59 |
| | $L_3$ | **87.07** | **86.69** | **87.66** | **85.76** |
| | Cosine | 87.03 | 86.51 | 87.62 | 85.73 |
| Handwritten view 1-4 | $L_1$ | **93.32** | **93.03** | **92.15** | **92.60** |
| | $L_2$ | 92.32 | 92.06 | 91.35 | 91.51 |
| | $L_3$ | 91.76 | 91.49 | 91.18 | 90.89 |
| | Cosine | 93.28 | 92.99 | 92.13 | 92.56 |

TABLE VIII
ABLATION STUDY OF OUR METHOD. COMPARISON BETWEEN DIFFERENT SIMILARITY MEASURES IN THE INCOMPLETE CASES. VP INDICATES VIEW COMPLETION PROPOSED IN EQ. (29), EQ. (30) AND EQ. (31).

| Datasets | Methods | FP | FB | NMI | ARI |
|---|---|---|---|---|---|
| Handwritten view 1-4 | $L_1$ | 89.06 | 88.97 | 89.06 | 87.94 |
| | $L_2$ | 88.17 | 88.18 | 88.36 | 86.97 |
| | $L_3$ | 89.47 | 89.15 | 89.04 | 88.39 |
| | Cosine | **91.73** | **91.42** | **90.92** | **90.86** |
| Handwritten view 1-4 VP | $L_1$ | 90.04 | 89.95 | 89.83 | 89.01 |
| | $L_2$ | 89.51 | 89.45 | 89.29 | 88.44 |
| | $L_3$ | 90.32 | 90.04 | 89.81 | 89.33 |
| | Cosine | **92.48** | **92.18** | **91.56** | **91.69** |

TABLE IX
ABLATION STUDY OF OUR METHOD. COMPARISON ON CONSISTENCY LOSS.

| Dataset | Consistency Loss | FP | FB | NMI | ARI |
|---|---|---|---|---|---|
| Handwritten view 1-2 | w/ Eq. (19) | 77.82 | 77.95 | 82.64 | 75.66 |
| | w/ Eq. (21) | **87.03** | **86.51** | **87.62** | **85.73** |
| Handwritten view 1-4 | w/ Eq. (19) | 80.43 | 80.34 | 82.41 | 78.71 |
| | w/ Eq. (21) | **93.28** | **92.99** | **92.13** | **92.56** |

TABLE X
ABLATION STUDY OF OUR METHOD. COMPARISON ON LOSS COMPONENT.

| Dataset | Component | FP | FB | NMI | ARI |
|---|---|---|---|---|---|
| Handwritten view 1-2 | w/o $L_{consistency1}$ | 83.63 | 83.00 | 84.60 | 82.03 |
| | w/o $L_{consistency2}$ | 86.15 | 85.95 | 87.39 | 84.77 |
| | w/o $L_{constraint}$ | 80.81 | 80.41 | 83.97 | 79.06 |
| | SLS-MPC | **87.03** | **86.51** | **87.62** | **85.73** |
| Handwritten view 1-4 | w/o $L_{consistency1}$ | 90.13 | 89.74 | 89.63 | 89.12 |
| | w/o $L_{consistency2}$ | 85.61 | 85.44 | 86.17 | 84.25 |
| | w/o $L_{constraint}$ | 83.50 | 83.44 | 84.75 | 81.98 |
| | SLS-MPC | **93.28** | **92.99** | **92.13** | **92.56** |

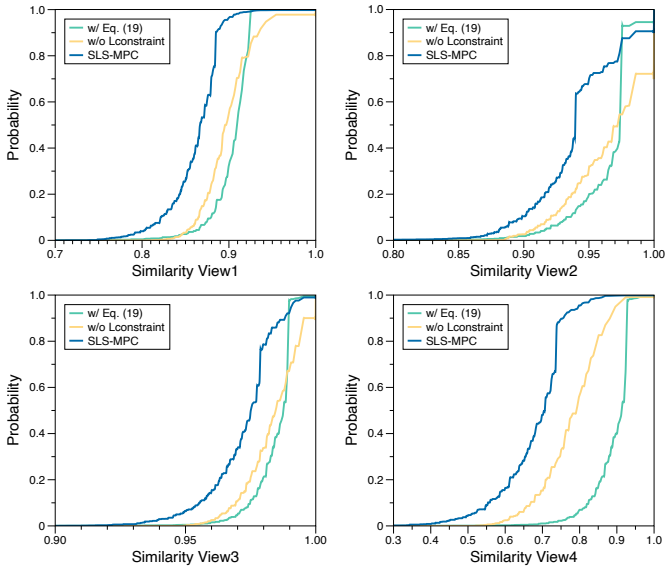



Fig. 7. The visualization of self-learning probability function in Handwritten with four views.

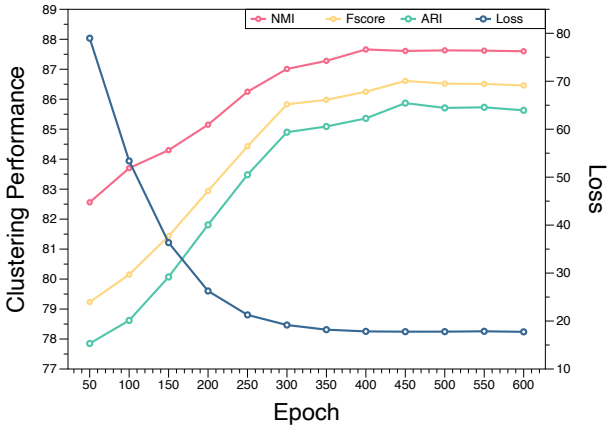



Fig. 8. The clustering performance of SLS-MPC with increasing epoch on Handwritten. The x-axis denotes the epoch in iteration, the left and right y-axis denote the clustering performance and corresponding loss value, respectively.

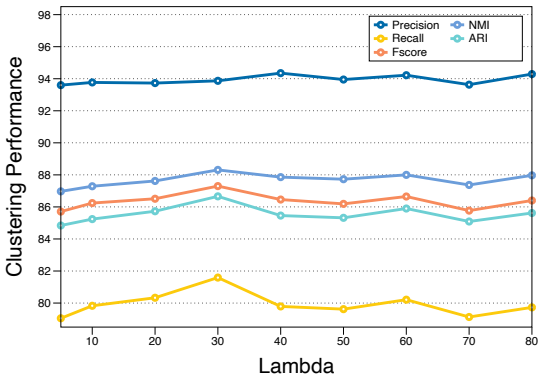



Fig. 9. The analysis of parameter $\lambda$ on Handwritten.

function. As shown in Table IX, using Eq. (19) results in poor clustering performance, which demonstrates that $Fsingle$ is confused by $Fmulti$ and $Fcross$ in the consistency learning process and the successful introduction of Eq. (21) enables the learning of a better probability function. Moreover, as shown in Fig. 7, using Eq. (19) causes the probability function to shift to the right. The probability function is relatively steep and the value of the probability function is low and inaccurate. Specifically, in the fourth view, the value of the probability function reaches 1.0 only when the similarity arrives at about 0.92. And, the value of the probability function varies greatly when the similarity fluctuates around 0.9.

**Ablation on Loss Component.** As described in Eq. (18), consistency loss and constraint loss are introduced in self-learning to learn probability function. As shown in Table X, all loss terms play indispensable roles in SLS-MPC. Moreover, as shown in Fig. 7, optimizing without $L_{constraint}$ makes the range of the probability function unconstrained. The maximum value of the probability function is about 0.7 and 0.9 in the second view and the third view respectively. It should be pointed out that optimizing without $L_{constraint}$ results in poor clustering performance, which demonstrates the importance of range constraint.

**Analysis of Convergence.** In this sub-section, we analyze the convergence of SLS-MPC by reporting the loss value and the corresponding clustering performance with increasing epochs. As shown in Fig. 8, the loss value remarkably decreases in the first 300 epochs, and meanwhile NMI, Fscore, and ARI continuously increase. And then the clustering performance keeps stable in the last moving epochs.

**Analysis of Parameter $\lambda$.** According to Eq. (18), objective function contains a balanced factor $\lambda$ on $L_{consistency}$ and $L_{constraint}$. We choose nine values from 5 to 80 to study how it affects the clustering performance on Handwritten with two views. As shown in Fig. 9, the clustering performance is robust when the factor $\lambda$ changes and precision is stable when the factor $\lambda$ is around 20, which is the value we used for reporting performance in the above results on Handwritten with two views. The detailed factor $\lambda$ used in our experiments is listed in Table II.

**Analysis of Multi-view Probability.** We use multi-view probability generated from MPC and our proposed SLS-MPC to replace the kernel matrix in EEIMC [7] and the similarity matrix in PIC [11]. The clustering results are listed in Table XI. Compared with origin kernel matrix and similarity matrix, the performance of EEIMC and PIC using multi-view probability are improved which further demonstrates that the accuracy of multi-view probability is better than that of origin similarity

TABLE XI
THE CLUSTERING PERFORMANCE OF EEIMC AND PIC WITH MPC AND SLS-MPC.

| Methods | Handwritten | 100Leaves | Humbi240 |
|---|---|---|---|
| EEIMC [7] | 76.51 | 73.84 | 91.41 |
| EEIMC w/ MPC | +9.69 | +11.63 | -0.70 |
| EEIMC w/ SLS-MPC | **+10.81** | **+14.73** | **+0.48** |
| PIC [11] | 73.94 | 77.82 | 94.32 |
| PIC w/ MPC | +14.87 | +8.78 | +1.62 |
| PIC w/ SLS-MPC | **+17.24** | **+12.66** | **+2.62** |

and using SLS-MPC works better which demonstrates the effectiveness of symmetry and self-learning in SLS-MPC.

## V. CONCLUSION

In this paper, we propose self-learning symmetric multi-view probabilistic clustering (SLS-MPC) to tackle the challenges: i) lack of unified framework for incomplete and complete MVC, ii) lack of emphasis on noise and outliers and iii) dependence on category information and complex hyper-parameters. SLS-MPC proposes a novel self-learning probability function to effectively learn each view's individual distribution without any prior knowledge and hyper-parameters from the aspect of consistency in single-view, cross-view and multi-view and a novel method to adaptively estimate the posterior matching probability from multiple views without complicated hyper-parameters fine-tuning, which tolerates incomplete views. Besides, equipped with graph-context-aware probability refinement, SLS-MPC takes noise and outliers into consideration. Moreover, SLS-MPC proposes a novel probabilistic clustering algorithm, which has no optimization parameters and generates clustering results in an unsupervised manner and an efficient way without category information. Extensive experiments on multiple benchmarks for incomplete and complete MVC show that our proposed SLS-MPC performs markedly better than SOTA methods.

## ACKNOWLEDGMENTS

This work was supported in part by Zhejiang Provincial Natural Science Foundation of China under Grant No. LDT23F01013F01; in part by the Fundamental Research Funds for the Central Universities; in part by Alibaba Group through Alibaba Research Intern Program.

## REFERENCES

[1] Y. Yang and H. Wang, "Multi-view clustering: A survey," *Big Data Mining and Analytics*, vol. 1, no. 2, pp. 83–107, 2018.
[2] V. Sindhwani, "A co-regularized approach to semi-supervised learning with multiple views," in *Proc. of the 22th ICML workshop on Learning with Multiple views, 2008*, 2008.
[3] Y. Li, F. Nie, H. Huang, and J. Huang, "Large-scale multi-view spectral clustering via bipartite graph," in *AAAI Conference on Artificial Intelligence*, 2015.
[4] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, "Multiple kernel k-means clustering with matrix-induced regularization," in *AAAI Conference on Artificial Intelligence*, 2016.
[5] Z. Zhang, L. Liu, F. Shen, H. T. Shen, and L. Shao, "Binary multi-view clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1774–1782, 2019.
[6] M. Sun, P. Zhang, S. Wang, S. Zhou, W. Tu, X. Liu, E. Zhu, and C. Wang, "Scalable multi-view subspace clustering with unified anchors," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 3528–3536.
[7] X. Liu, M. Li, C. Tang, J. Xia, J. Xiong, L. Liu, M. Kloft, and E. Zhu, "Efficient and effective regularized incomplete multi-view clustering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 8, pp. 2634–2646, 2020.
[8] S. Xiang, L. Yuan, W. Fan, Y. Wang, P. M. Thompson, and J. Ye, "Multi-source learning with block-wise missing data for alzheimer's disease prediction," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013, p. 185–193.
[9] H. Wang, Y. Yang, and B. Liu, "Gmc: Graph-based multi-view clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 6, pp. 1116–1129, 2020.
[10] J. Wen, Z. Zhang, Y. Xu, B. Zhang, L. Fei, and H. Liu, "Unified embedding alignment with missing views inferring for incomplete multi-view clustering," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 5393–5400, Jul. 2019.
[11] H. Wang, L. Zong, B. Liu, Y. Yang, and W. Zhou, "Spectral perturbation meets incomplete multi-view data," in *International Joint Conference on Artificial Intelligence*, 7 2019, pp. 3677–3683.
[12] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.
[13] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.
[14] F. Nie, J. Li, and X. Li, "Self-weighted multiview clustering with multiple graphs," in *International Joint Conference on Artificial Intelligence*, 2017, p. 2564–2570.
[15] X. Zhu, S. Zhang, W. He, R. Hu, C. Lei, and P. Zhu, "One-step multi-view spectral clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 10, pp. 2022–2034, 2019.
[16] D. D. Lee and H. S. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, no. 7, 1999.
[17] J. Liu, W. Chi, G. Jing, and J. Han, *Multi-view clustering via joint nonnegative matrix factorization*. Proceedings of the 2013 SIAM International Conference on Data Mining, 2013.
[18] W. Shao, L. He, and P. S. Yu, "Multiple incomplete views clustering via weighted nonnegative matrix factorization with $l_{2,1}$ regularization," in *Machine Learning and Knowledge Discovery in Databases*, 2015, pp. 318–334.
[19] J. Wang, F. Tian, H. Yu, C. H. Liu, K. Zhan, and X. Wang, "Diverse non-negative matrix factorization for multiview data representation," *IEEE Transactions on Cybernetics*, vol. 48, no. 9, pp. 2620–2632, 2018.
[20] Y. Wang, L. Wu, X. Lin, and J. Gao, "Multiview spectral clustering via structured low-rank matrix factorization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 4833–4843, 2018.
[21] Y. Wang and L. Wu, "Beyond low-rank representations: Orthogonal clustering basis reconstruction with optimized graph structure for multi-view spectral clustering," *Neural Networks*, vol. 103, pp. 1–8, 2018.
[22] B. Zhao, J. T. Kwok, and C. Zhang, "Multiple kernel clustering," in *Proceedings of the SIAM International Conference on Data Mining*, 2009, pp. 638–649.
[23] S. Wang, X. Liu, E. Zhu, C. Tang, and J. Yin, "Multi-view clustering via late fusion alignment maximization," in *International Joint Conference on Artificial Intelligence*, 7 2019, pp. 3778–3784.
[24] S. Zhou, X. Liu, M. Li, E. Zhu, L. Liu, C. Zhang, and J. Yin, "Multiple kernel clustering with neighbor-kernel subspace segmentation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 4, pp. 1351–1362, 2020.
[25] Y. Zhang, X. Liu, S. Wang, J. Liu, S. Dai, and E. Zhu, "One-stage incomplete multi-view clustering via late fusion," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 2717–2725.
[26] C. Zhang, Y. Liu, and H. Fu, "Ae2-nets: Autoencoder in autoencoder networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2577–2585.
[27] S. Chang, J. Hu, T. Li, H. Wang, and B. Peng, "Multi-view clustering via deep concept factorization," *Knowledge-Based Systems*, vol. 217, p. 106807, 2021.
[28] Y. Lin, Y. Gou, Z. Liu, B. Li, J. Lv, and X. Peng, "Completer: Incomplete multi-view clustering via contrastive prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 174–11 183.

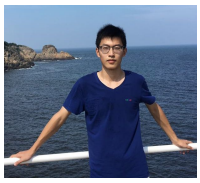
[29] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, "Marginalized multiview ensemble clustering," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 2, pp. 600–611, 2019.
[30] X. Li, H. Zhang, R. Wang, and F. Nie, "Multiview clustering: A scalable and parameter-free bipartite graph fusion method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 330–344, 2020.
[31] R. Sibson, "Slink: An optimally efficient algorithm for the single-link cluster method," *The Computer Journal*, vol. 16, no. 1, pp. 30–34, 01 1973.
[32] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *science*, vol. 315, no. 5814, pp. 972–976, 2007.
[33] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *KDD*, vol. 96, no. 34, 1996, pp. 226–231.
[34] Z. Lu and T. K. Leen, "Semi-supervised learning with penalized probabilistic clustering," in *NIPS*, 2004, pp. 849–856.
[35] ——, "Penalized probabilistic clustering," *Neural Computation*, vol. 19, no. 6, pp. 1528–1567, 2007.
[36] J. Liu, J. Liu, S. Yan, R. Jiang, X. Tian, B. Gu, Y. Chen, C. Shen, and J. Huang, "Mpc: Multi-view probabilistic clustering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9509–9518.
[37] S. Abney, "Bootstrapping," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 2002, p. 360–367.
[38] M. White, Y. Yu, X. Zhang, and D. Schuurmans, "Convex multi-view subspace learning." in *NIPS*, 2012, pp. 1682–1690.
[39] N. Chen, J. Zhu, and E. P. Xing, "Predictive subspace learning for multi-view data: a large margin approach," in *NIPS*, 2010, pp. 361–369.
[40] S. Bickel and T. Scheffer, "Multi-view clustering." in *ICDM*, vol. 4, no. 2004, 2004, pp. 19–26.
[41] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan, "Multi-view clustering via canonical correlation analysis," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 129–136.
[42] U. Cohen, S. Chung, D. D. Lee, and H. Sompolinsky, "Separability and geometry of object manifolds in deep neural networks," *Nature communications*, vol. 11, no. 1, pp. 1–13, 2020.
[43] D. Dua and C. Graff, "UCI machine learning repository," 2017, available: http://archive.ics.uci.edu/ml.
[44] C. Mallah, J. Cope, J. Orwell *et al.*, "Plant leaf classification using probabilistic integration of shape, texture and margin features," *Signal Processing, Pattern Recognition and Applications*, vol. 5, no. 1, 2013.
[45] Z. Yu, J. S. Yoon, I. K. Lee, P. Venkatesh, J. Park, J. Yu, and H. S. Park, "Humbi: A large multiview dataset of human body expressions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
[46] J. S. Di Huang and Y. Wang, "The buaa-visnir face database instructions," *Technical report*, 2012.
[47] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo, "A comparison of extrinsic clustering evaluation metrics based on formal constraints," *Information retrieval*, vol. 12, no. 4, pp. 461–486, 2009.
[48] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

**Junjie Liu** was born in Jiangsu Province, China, in 1995. He received the B.Sc. and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 2017 and 2024. He is currently an algorithm engineer at Alibaba Cloud, Hangzhou, China. His research interests include image processing, computer vision and machine learning.

**Junlong Liu** received the B.S. degree in computer science from Beihang University and the M.S. degree in machine learning from University of Science and Technology of China, China, in 2014 and 2017. He is currently an algorithm engineer at Alibaba Cloud, Hangzhou, China. His research interests include image processing, computer vision and machine learning.

**Rongxin Jiang** was born in Hunan Province, China, in 1982. He received the B.Sc. and Ph.D. degrees in computer vision from Zhejiang University, Hangzhou, China, in 2002 and 2008, respectively. He is currently an Associate Professor of Zhejiang University. His major research fields are computer vision and networking.

**Yaowu Chen** was born in Heilongjiang Province, China, in 1963. He received the Ph.D. degree in embedded system from Zhejiang University, Hangzhou, China, in 1998. He is currently a Professor and the Director of the Institute of Advanced Digital Technologies and Instrumentation, Zhejiang University. His major research fields are embedded system, multimedia system, and networking.



**Chen Shen** received his B.S. degree and Ph.D. in Electrical Engineering at Zhejiang University, China, in 2012 and 2018. Now he is a Senior algorithm Engineer at Alibaba Cloud, Hangzhou, China. His research interests include deep learning, data mining and large language models.

**Jieping Ye** is a VP of Alibaba Cloud. His research interests include big data, machine learning, and artificial intelligence with applications in transportation, smart city, and biomedicine. He has served as a Senior Program Committee/Area Chair/Program Committee Vice Chair of many conferences including NeurIPS, ICML, KDD, IJCAI, ICDM, and SDM. He has served as an Associate Editor of Data Mining and Knowledge Discovery, IEEE Transactions on Knowledge and Data Engineering, and IEEE Transactions on Pattern Analysis and Machine Intelligence. He won the NSF CAREER Award in 2010. His papers have been selected for the outstanding student paper at ICML in 2004, the KDD best research paper runner up in 2013, and the KDD best student paper award in 2014. He has also won the first place in 2019 INFORMS Daniel H. Wagner Prize, one of the top awards in operation research practice. Dr. Ye was elevated to an IEEE Fellow in 2019 and named an ACM Distinguished Scientist in 2020 for his contributions to the methodolog