GENERALIZED STRUCTURE-AWARE MISSING VIEW COMPLETION NETWORK FOR INCOMPLETE MULTI-VIEW CLUSTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

In recent years, incomplete multi-view clustering has been widely regarded as a challenging problem. The missing views inevitably damage the effective information of the multi-view data itself. To date, existing methods for incomplete multi-view clustering usually bypass invalid views according to prior missing information, which is considered as a second-best scheme based on evasion. Other methods that attempt to recover missing information are mostly applicable to specific two-view datasets. To handle these problems, we design a general structure-aware missing view completion network (SMVC) for incomplete multi-view clustering. Concretely, we build a two-stage autoencoder network with the self-attention structure to synchronously extract high-level semantic representations of multiple views and recover the missing data. In addition, we develop a recurrent graph reconstruction mechanism that cleverly leverages the restored views to promote the representation learning and the further data reconstruction. Sufficient experimental results confirm that our SMVC has obvious advantages over other top methods.

1 INTRODUCTION

It is well known that multi-view data depicts the observed objects from different perspectives (Tsai et al., 2020; Han et al., 2020; Li et al., 2021; Wang et al., 2015). Compared with traditional single-view data, this heterogeneous data retains multi-level semantic information (Wen et al., 2020a). In the past few years, multi-view clustering, as a novel representation learning method, has aroused extensive research enthusiasm and has been widely used in related fields such as data analysis (Zhang et al., 2019). However, conventional multi-view clustering methods usually assume that obtaining complete multi-view data is a matter of course, which obviously goes against practical experience. Therefore, a number of incomplete multi-view clustering (IMC) approaches have been developed to fit the increasingly common incomplete multi-view datasets, which is also the focus of our paper.

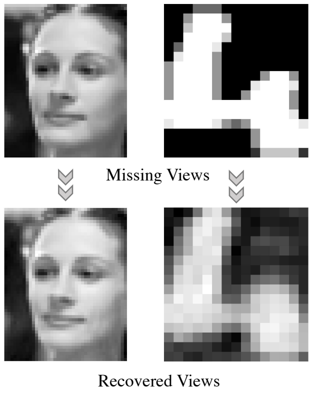


Figure 1: Missing view completion

In the area of incomplete multi-view learning, there are two main technical approaches to missing multi-view data, *i.e.*, *detour*–skip unavailable views by prior missing information or *completion*–recover missing data according to existing information. In general, it is more difficult to get good recovery quality because the information content is indeed incomplete, so most methods choose to skip these missing data to avoid a bad impact. Classical partial multi-view clustering (PMVC) (Li et al., 2014) is one of the most representative IMC methods, which tries to connect different view spaces employing the samples with all views. On this basis, IMG (incomplete multi-modality grouping) (Zhao et al., 2016) constructs a complete Laplacian graph with the common representation in the latent space to provide the global property to subspace learning. However, both of these two grouping methods require at least one complete sample in the multi-view database. Online multi-view clustering (OMVC) (Shao et al., 2016) and One-pass

IMC (OPIMVC) (Hu & Chen, 2019b) introduce a weight matrix with missing prior information to perform multi-view weighted matrix factorization, assigning low weights to missing views to reduce the negative impact caused by mean padding or zero padding.

However, such detour or neglect is only the second-best solution to the incomplete issue. Some researchers attempt to recover missing information in various ways to perform complete multi-view clustering. Rai et al. (2010) proposed the kernel canonical correlation analysis with incomplete views (KCCA-IV) to restore the kernel matrix of incomplete view based on the complete kernel, which requires at least one view to be intact. The multiple kernel k-means with incomplete kernels (MKKM-IK-MKC) (Liu et al., 2019) fuses kernel completion and clustering into one framework. Wen et al. (2019) proposed an IMC framework named unified embedding alignment framework (UEAF), which emphasizes the importance of recovering missing views, and introduces both forward and reverse graph regularization to facilitate view recovery and cross-view consistent representation learning.

Considering the above issues, in this paper, we aim to propose a general IMC framework, called **S**tructure-aware **M**issing **V**iew **C**ompletion network (SMVC), which can handle arbitrary view-missing situations and enhance the performance of representation learning through efficient view completion. Meanwhile, inspired by the successful application of deep learning, especially the transformer (Vaswani et al., 2017), we combine the characteristics of the multi-head self-attention mechanism and multi-view learning to design a transformer-style cross-view autoencoder network. Compared with a simple linear encoder, it can extract high-level semantic features and support cross-view information interaction, which is conducive to mining the complementarity of multiple views. At the same time, we skillfully integrate multi-view fusion representation learning and missing view recovery into a unified framework. More importantly, we propose a structure-aware module (recurrent graph constraint) to push the reconstructed data inversely to participate in the representation learning process, where they are allowed to collaborate with each other to achieve better clustering. Finally, our contributions are summarized as follows:

- We design a general IMC framework named SMVC, which includes an encoder module that integrates cross-view information interaction and high-level semantic feature extraction, and a multi-view reconstruction and recovery module based on controlled coding.
- We propose an innovative recurrent graph embedding constraint whose core, an approximately complete graph generated from imputation data, cyclically facilitates reliable feature extraction and view recovery.
- A cluster-friendly two-stage training strategy is presented in detail, and extensive experiments and intuitive visualization results demonstrate the effectiveness of our SMVC.

2 Preliminary

2.1 Problem definition and notations

For ease of expression, we first give a formal definition of the investigated problem. Given the multi-view data $\{\boldsymbol{X}^{(v)} \in \mathbb{R}^{n \times d_v}\}_{v=1}^{m}$ with m views and n samples, our goal is to divide them into c cluster centroids. d_v is the dimension of view v and the dimension of embedding feature is d_e. In our method, a missing indicator matrix is introduced, *i.e.*, $\boldsymbol{W}$, whose element $\boldsymbol{W}_{i,j} = 1$ denotes the j-th view of i-th sample is available, otherwise $\boldsymbol{W}_{i,j} = 0$. $\bar{\boldsymbol{X}}^{(v)} \in \mathbb{R}^{n \times d_v}$ represents the reconstructed view v including missing instances and $\boldsymbol{X}'^{(v)} \in \mathbb{R}^{n \times d_v}$ is the imputation view v filled with recovered data. $\mathbf{Z} \in \mathbb{R}^{n \times m \times d_e}$ is the extracted embedding tensor and its fusion representation $\bar{\boldsymbol{Z}} \in \mathbb{R}^{n \times d_e}$ is our objective matrix. Following Goodfellow et al. (2016), all subscript representations of matrices or tensors conform to the recommended criteria.

2.2 Related work: UEAF

In this subsection, we simply introduce a related IMC method: UEAF (Wen et al., 2019). Like most IMC methods, UEAF aims to learn a consensus representation ($\boldsymbol{P}$ in this work) for all views. To do this, Wen et al. developed a complex framework that integrates view recovery, consensus learning,

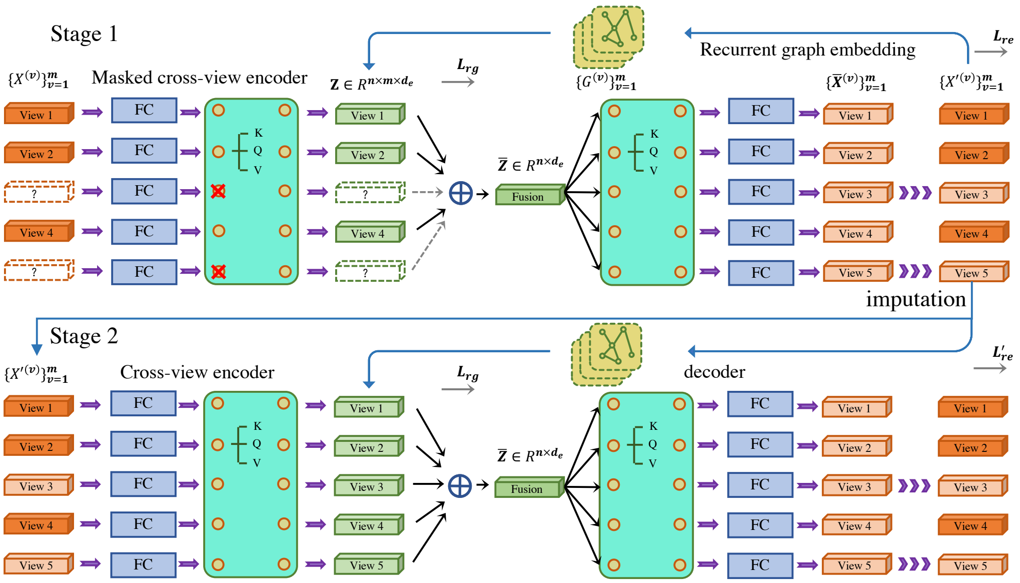


Figure 2: Main framework of our model. The FC module means Fully Connected layer.

and two graph constraint terms into one model:

$$\min_{\Upsilon} \sum_{v=1}^{m} (\alpha_v)^r \begin{pmatrix} \|\boldsymbol{X}^{(v)} + \boldsymbol{E}^{(v)}\boldsymbol{M}^{(v)} - \boldsymbol{U}^{(v)}\boldsymbol{P}^{(v)}\|_F^2 \\ + \lambda_1 \operatorname{Tr}(\boldsymbol{E}^{(v)T}\boldsymbol{L}^{(v)}\boldsymbol{E}^{(v)}) \\ + \frac{\lambda_2}{2} \sum_{i,j}^{n} \|\boldsymbol{U}^{(v)}\boldsymbol{P}_{:,i} - \boldsymbol{U}^{(v)}\boldsymbol{P}_{:,i}\|_2^2 \boldsymbol{S}_{i,j}^2 \end{pmatrix}, s.t. \begin{cases} \boldsymbol{U}^{(v)T}\boldsymbol{U}^{(v)} = \boldsymbol{I} \\ \sum_{v=1}^{m} \alpha_v = 1, \alpha_v \geq 0 \\ \forall i, \boldsymbol{S}_{i,:}\mathit{1} = \mathit{1} \end{cases} \tag{1}$$

Such a complex model contains many variables, so we only explain the core parts in detail. $\boldsymbol{E}^{(v)}$ is the error matrix used to model the missing instances and $\boldsymbol{M}$ is the transition matrix from missing instances to complete views. $\boldsymbol{U}^{(v)}$ and $\boldsymbol{P}^{(v)}$ are the basis matrix and low-dimensional representation of view v. The most commendable part is the two Laplacian constraint terms, *i.e.*, the Laplacian matrix $\boldsymbol{L}^{(v)}$ constructed by the incomplete graph is introduced to constrain the inference of missing views and another nearest neighbor graph $\boldsymbol{S}$ is utilized reversely to align all recovered incomplete views. Υ is the optimization objective concerning the above variables.

3 METHOD

3.1 MOTIVATION

As mentioned in the introduction, recovering the missing data must be based on existing information. We consider this question from the key property of multi-view data. As we know, different views enjoy same high-level semantic information in the clustering task, *i.e.*, they're different descriptions of the same abstract target. If we can capture the shared high-level semantic information, then it becomes possible to infer the missing information backwards based on the learned patterns. From another perspective, missing data inference can be regarded as a generation task, which is usually implemented via autoencoder networks. Inspired by above analysis, we design a cross-view autoencoder as our main framework, whose encoder learns the high-level semantic representations and decoder attempts to recover the missing views from a fusion representation.

Another thing that should not be ignored is that the intrinsic structure of the data is crucial for unsupervised learning, which has been demonstrated by numerous studies. The classical nearest neighbor graph constraint, widely used in various traditional machine learning methods, enables the extracted semantic representation to maintain the original topology of data, which not only facilitates the learning of clustering structure to a large extent but also drives the model to 'guess' the missing data in a more reasonable direction. Nevertheless, it should be noted that it is hard for us to directly obtain a complete graph from the incomplete data unless we can provide approximately intact data. So far, re-examining the above motivations, our idea to missing data imputation is gradually blossomed, that is, fusing approximate graph construction and missing view recovery in an unified autoencoder framework.

At the same time, it must be considered that our downstream task is unsupervised multi-view clustering, which means that view imputation serves the complete multi-view clustering. To do this, a 2-Stage training strategy is applied in our network, *i.e.*, *Stage 1*: missing view recovery and *Stage 2*: complete multi-view clustering based on recovered data. In Stage 1, the noisy data in missing positions must be addressed to avoid the negative influence, but in Stage 2, the new imputed data (approximately complete data), generated by the recovered views of Stage 1 and raw incomplete data, is used as the input, *i.e.*, $\boldsymbol{X}'^{(v)} = \mathrm{diag}(\boldsymbol{W}_{:,v})\boldsymbol{X}^{(v)} + \mathrm{diag}(1-\boldsymbol{W}_{:,v})\bar{\boldsymbol{X}}^{(v)}$, so the entire model can be treated as a common multi-view clustering network without any extra measures in Stage 2. It is worth noting that the input to Stage 2, $\boldsymbol{X}'^{(v)}$, is fixed after the training is completed in Stage 1 because frequent changes of input make it difficult for the model to converge. The remainder of this section elaborates on the details of our model and the main framework is shown in Figure 2.

3.2 Cross-view encoder

It is well known that the feature dimensions of raw data are diverse, so it is difficult for a model to fuse multi-view features in the original feature space. Traditional approaches usually utilize the autoencoders, mainly composed of Fully Connected (FC) layers, to extract view-specific features, which not only aligns the different dimensions in a common space, but also maps high-dimensional features to a relatively low-dimensional space to facilitate subsequent representation learning. However, such high-level features only extracted from their corresponding raw views lack information interaction among views. Specifically, different views describe the objects from different perspectives, so each view can be considered both unique and biased. These differences (*i.e.*, complementary information) naturally make multi-view data more expressive than single-view data, and how to make full use of complementary information is always one of the key problems in the field of multi-view learning. To do this, we design a transformer-style cross-view encoder with multi-view information aggregation. At first, we need a group of low-level feature extractors to project various views into a common feature space, which allows subsequent modules to process the representations of all views in parallel. For simplicity, we select m Fully Connected layers as the low-level feature extractors: $\left\{\Phi_v(\boldsymbol{X}^{(v)};\boldsymbol{\theta}_v) = \widehat{\boldsymbol{X}^{(v)}} \in \mathbb{R}^{n\times d_e}\right\}_{v=1}^{m}$, where $\boldsymbol{\theta}_v$ denotes the parameters of extractor Φ_v. Notably, $\widehat{\boldsymbol{\mathsf{X}}} = \left\{\widehat{\boldsymbol{X}^{(v)}} \in \mathbb{R}^{n\times d_e}\right\}_{v=1}^{m} \in \mathbb{R}^{n\times m\times d_e}$ is the input tensor of the subsequent self-attention module with no information interaction. And then, the transformer-style view aggregation module is defined as follows: Given the number of heads h and the embedding $\widehat{\boldsymbol{\mathsf{X}}}_{i,:,:}$ of sample i, we utilize h groups of linear layers with weights $\left\{\boldsymbol{W}^{q_t}, \boldsymbol{W}^{k_t}, \boldsymbol{W}^{v_t}\right\}_{t=1}^{h}$ to obtain corresponding query, key, and value maps $\{\boldsymbol{Q}^t, \boldsymbol{K}^t, \boldsymbol{V}^t\}_{t=1}^{h} \in \mathbb{R}^{m\times d_h}$. An entire embedding feature is split into h segments with dimension $d_h = d_e/h$. For head t of sample i, we compute their score matrix as follows:

$$\boldsymbol{S}^t = \mathit{softmax}(\boldsymbol{Q}^t(\boldsymbol{K}^t)^T/\sqrt{d_h}). \tag{2}$$

We need to highlight that this $\boldsymbol{S}^t \in \mathbb{R}^{m\times m}$ is for the complete views used in Stage 2. As for the incomplete views in missing view inference stage (*i.e.* Stage 1), we define the masked attention scores $\widehat{\boldsymbol{S}}^t \in \mathbb{R}^{m\times m}$ by:

$$\widehat{\boldsymbol{S}}^t = \mathit{softmax}(\mathit{zerofill}(\boldsymbol{Q}^t(\boldsymbol{K}^t)^T/\sqrt{d_h}) \odot \boldsymbol{W}_{i,:}^T\boldsymbol{W}_{i,:}), \tag{3}$$

where the *zerofill* represents the operation to fill zero value with -1e9 and $\odot$ is the Hadamard product. This aims to ignore the missing views in the computation of cross-view attentions. Then, we aggregate all views by:

$$\boldsymbol{A}^t = \begin{cases} \widehat{\boldsymbol{S}}^t\boldsymbol{V}^t, \text{Stage 1} \\ \boldsymbol{S}^t\boldsymbol{V}^t, \text{Stage 2} \end{cases}, \tag{4}$$

where $\boldsymbol{A}^t \in \mathbb{R}^{m\times d_h}$ is the new embedding features with information interaction of head t. Similar to the multi-head transformer, the final embedding features of m views can be calculated as $\boldsymbol{A} = Concat(\boldsymbol{A}^1, \boldsymbol{A}^2, ..., \boldsymbol{A}^h) \in \mathbb{R}^{m\times d_e}$. Besides, we sequentially input the $\boldsymbol{A}$ into a linear layer, layer norm module, and a multilayer perceptrons to get final embedding features $\boldsymbol{Z} \in \mathbb{R}^{m\times d_e}$ for each sample ($\boldsymbol{\mathsf{Z}} \in \mathbb{R}^{n\times m\times d_e}$ for all samples), *i.e.*, cross-view encoder $\mathcal{E} : \{\boldsymbol{X}^{(v)}\}_{v=1}^{m} \rightarrow \boldsymbol{\mathsf{Z}}$.

3.3 View recovery based on controlled encoding

As mentioned in the previous subsection, all views exchange information in the extraction of high-level semantic representations. However, no appropriate constraint be imposed in the extraction of encoder, which means that the encoding process is uncontrolled. To solve this problem, we exploit a simple but effective approach, named multi-view weighted fusion, to obtain a common representation of all views, which is expected to comprehensively characterize the sample across views. Specifically, our multi-view weighted fusion is formulated as follows:

$$\bar{\boldsymbol{Z}}_{i,:} = \sum_{v=1}^{m} \frac{\mathbf{Z}_{i,v,:}\boldsymbol{W}_{i,v}}{\sum_{v} \boldsymbol{W}_{i,v}}, \tag{5}$$

where vector $\bar{\boldsymbol{Z}}_{i,:}$ denotes the i-th sample of the fusion matrix $\bar{\boldsymbol{Z}} \in \mathbb{R}^{n \times d_e}$. Obviously, $\bar{\boldsymbol{Z}}$ in Eq. (5) is designed for Stage 1, and $\bar{\boldsymbol{Z}}$ can be simply calculated by $\bar{\boldsymbol{Z}}_{i,:} = \frac{1}{m}\sum_{v} \mathbf{Z}_{i,v,:}$ in Stage 2. Furthermore, the common representation $\bar{\boldsymbol{Z}}$ is required to contain all information of multiple views. To the end, a symmetrical cross-view decoder module $\mathcal{D}$ is concatenated to $\bar{\boldsymbol{Z}}$ to reconstruct all the data including missing views, *i.e.*, $\mathcal{D} : \{\bar{\boldsymbol{Z}}\}^{m} \in \mathbb{R}^{n \times m \times d_e} \rightarrow \{\bar{\boldsymbol{X}}^{(v)} \in \mathbb{R}^{n \times d_v}\}_{v=1}^{m}$. But in fact, due to the lack of supervisory information to directly discriminate the recovered data, we can only leverage the available original data to impose a partial reconstruction constraint. In other words, this recovery is a natural but necessary byproduct of the proposed autoencoder framework that aims to learn the common representation shared by the available views. As a result, we introduce a weighted reconstruction loss $\mathcal{L}_{re}$:

$$\mathcal{L}_{re} = \frac{1}{mn}\sum_{v=1}^{m}\sum_{i=1}^{n}\frac{1}{d_v}\left\|\bar{\boldsymbol{X}}_{i,:}^{(v)} - \boldsymbol{X}_{i,:}^{(v)}\right\|_2^2 \boldsymbol{W}_{i,v}, \tag{6}$$

where $\bar{\boldsymbol{X}}_{i,:}$ is the i-th sample of the reconstructed data $\bar{\boldsymbol{X}}$. And in Stage 2, the $\mathcal{L}_{re}$ degenerates into

$$\mathcal{L}'_{re} = \frac{1}{mn}\sum_{v=1}^{m}\sum_{i=1}^{n}\frac{1}{d_v}\left\|\bar{\boldsymbol{X}}_{i,:}^{(v)} - \boldsymbol{X}_{i,:}^{\prime(v)}\right\|_2^2. \tag{7}$$

3.4 Recurrent graph constraint

In recent years, researchers have been accustomed to adding graph constraints to traditional multi-view learning methods, which help preserve the original intrinsic structure of data by constructing a prior adjacency matrix. This is based on this basic manifold assumption: if two samples are close to each other in original feature space, then they are also close in the embedding space. But in the case of incomplete data, some existing methods simply skip the missing views to construct the adjacency graph, which is obviously biased, especially on the databases with lager missing rates. Therefore, we expect to obtain an approximately complete adjacency graph to guide the encoder for the extraction of high-level semantic features. On the other hand, more discriminative semantic features can also facilitate the recovery of missing views and thus help to construct a more realistic graph. Combining these two points, we innovatively propose the recurrent graph constraint:

$$\mathcal{L}_{rg} = \frac{1}{mn^2}\sum_{v=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n}\left\|\mathbf{Z}_{i,v,:}|_{k} - \mathbf{Z}_{j,v,:}|_{k}\right\|_2^2 \mathcal{G}_{i,j}^{(v)}|_{k-1}, \tag{8}$$

where $\left\{\mathcal{G}^{(v)} \in \mathbb{R}^{n \times n}\right\}_{v=1}^{m}$ are the approximately complete graphs generated by imputation data of last epoch, *i.e.*, $\mathcal{G}^{(v)}|_{k-1} = \mathit{knn}(\boldsymbol{X}^{\prime(v)}|_{k-1}, K)$ and '$|_k$' denotes the k-th epoch. K is the number of nearest neighbors. $\mathcal{G}_{i,j}^{(v)} = 1$ means instance j is one of the K-neighbors of instance j in view v. The $\mathcal{L}_{rg}$ is executed only if $k > 0$. In fact, deep learning models are usually trained in mini-batch iterations to reduce memory overhead, and ours is no exception. But in doing so, the neighbor graph contains only a small batch of samples, which means that the graph constraint is local rather than global. In order to balance the mini-batch training approach and the intact graph constraint, we rewrite Eq. (8) in the case of mini-batch training as follows:

$$\mathcal{L}_{rg}^{batch} = \frac{1}{mnb}\sum_{v=1}^{m}\sum_{i=1}^{b}\sum_{j=1}^{n}\left\|\mathbf{Z}_{i,v,:}^{batch}|_{k} - \mathbf{Z}_{j,v,:}|_{k-1}\right\|_2^2 \mathcal{G}_{i,j}^{(v)}|_{k-1}, \tag{9}$$

where the size of mini-batch is b, and $\mathbf{Z}^{batch} \in \mathbb{R}^{b \times m \times d_e}$ is the output of the cross-view encoder corresponding to each mini-batch data. In the computation of graph loss, to maximize the use of global structural information, we preserve all the embedded features $\mathbf{Z}$ obtained in the last epoch and update its corresponding part after the current batch is processed. Similarly, in Stage 2, the graph $\mathcal{G}$ from Stage 1 is fixed rather than updated by last output, *i.e.*, $\mathcal{G}^{(v)}|_{k-1} = \mathcal{G}^{(v)}|_{k}$.

3.5 Overall loss function and clustering

To sum up, our overall loss function in Stage 1 is:

$$\mathcal{L} = \mathcal{L}_{re} + \beta \mathcal{L}_{rg}, \tag{10}$$

β is penalty parameter to balance the two losses. And the loss function in Stage 2 is:

$$\mathcal{L}' = \mathcal{L}'_{re} + \beta \mathcal{L}_{rg}. \tag{11}$$

As mentioned above, we conduct complete multi-view clustering in Stage 2, and the fusion embedding feature $\bar{\boldsymbol{Z}} \in \mathbb{R}^{n \times d_e}$ obtained in Stage 2 is regarded as our clustering indicator matrix. For simplicity, we perform *K*-means (MacQueen, 1967) on the $\bar{\boldsymbol{Z}}$ to obtain final clustering results $\boldsymbol{p} \in \mathbb{R}^{c}$.

4 Experiments

4.1 Experimental settings

Databases: In order to evaluate the performance of our model, we adopt five popular databases in our comparison experiments: (1) ***Handwritten digit*** (Asuncion & Newman, 2007), a dataset widely used in various fields, contains 2000 handwritten digital images with ten classes from ’0’ to ’9’. There are five categories of features as 5 views selected in our experiments. The first view represents a linear combination of pixels from the original picture with the size of 16×15. (2) ***Caltech7*** (Cai et al., 2013) is a subset of the Caltech101 database (Fei-Fei et al., 2004), and we select 1474 images covering seven categories from it. Each image is extracted with 6 types of features, i.e, LBP, Gist, Hog, Cenhist, Gabor, and wavelet-moments. (3) ***NH_face*** (Cao et al., 2015), as a subset of the NH database, is composed of 4660 images belonging to five persons in the movie ’Notting Hill’ (Wu et al., 2013). 3 views in terms of Gabor, gray pixels (size of 40×50), and LBP are included in the NH_face database. (4) ***Animal*** (Fei-Fei et al., 2004; Zhang et al., 2019) is a larger dataset with up to 10158 images and 50 categories, whose features are extracted by DECAF (Krizhevsky et al., 2017) and Vgg19 (Simonyan & Zisserman, 2014). All samples and features are adopted in our experiments. (5) ***Aloi_deep*** is a new multi-view database proposed by this paper, which is derived from the Aloi database (Geusebroek et al., 2005). The original Aloi database contains 110250 images of 1000 small objects. We select 100 objects as 100 categories, each with about 108 images, for a total of 10800 images. And we utilize three typical deep neural networks, *i.e.*, ResNet50 (He et al., 2016), Vgg16 (Simonyan & Zisserman, 2014), and Inception-v3 (Szegedy et al., 2016) with pre-trained weights, to extract the three-view features. Detailed information about the five datasets is listed in Table 7.

Preprocessing of incomplete datasets: To generate the incomplete datasets to simulate the missing-view case, following Wen et al. (2020b), we randomly disable [10%,30%,50%,70%] of the instances of each view but keep at least one view available for each sample. As for the Animal dataset with only two views, we randomly select [10%,30%,50%] of all samples as the paired samples with two views. The first view is removed for half of the remaining samples, and the second view is removed for the other half.

Comparison methods: In our experiments, eight state-of-the-art methods are selected to evaluate the performance of the proposed SMVC, of which **OMVC** (Shao et al., 2016), **OPIMC** (Hu & Chen, 2019b), **MKKM-IK-MKC** (Liu et al., 2019), and **UEAF** (Wen et al., 2019) have been described in the introduction. The other four comparison methods are as follows: (1) **BSV** (Zhao et al., 2016), a simple baseline method, fills missing views with the average vector and performs *K*-means on each view to obtain the best result. (2) **Concat** (Zhao et al., 2016) is another popular baseline method, which aligns all views with the same imputation strategy as BSV and simply concatenates them to conduct single-view clustering. (3) **MIC** (Shao et al., 2015), based on non-negative matrix factorization technology, also introduces the prior missing information to help learn robust latent representations. (4) **DAIMC** (Hu & Chen, 2019a) not only introduces a prior matrix to avoid the negative impact of missing instances, but also additionally introduces a regression coefficient matrix to align the basis matrix of individual views in the latent space.

Evaluation: Following Wen et al. (2021; 2019); Liu et al. (2020), we still select the clustering accuracy (ACC), normalized mutual information (NMI), and purity as our three metrics to evaluate these methods. The higher the values of the three metrics, the better the clustering performance. Besides, all comparison methods are performed multiple times to reduce randomness and their parameters are set as suggested in their papers or codes for a fair comparison.

4.2 Experimental results and analysis

Table 1: Results on **Handwritten** database with different incomplete rates. The 1st and 2nd best results are marked in **bold** and underline.

Method	**ACC (%)**			**NMI (%)**			**Purity (%)**		
	30%	50%	70%	30%	50%	70%	30%	50%	70%
BSV	51.49±2.29	38.24±2.25	27.15±1.31	47.01±1.71	32.21±1.00	19.48±0.69	53.69±1.54	39.54±2.04	27.76±1.09
Concat	55.48±1.57	42.19±0.99	28.31±0.75	51.66±0.99	38.24±1.59	23.50±0.95	57.32±1.15	44.21±0.98	30.45±0.80
MIC	73.29±3.41	61.27±3.16	41.34±2.69	65.39±2.08	52.95±1.33	34.71±2.11	74.31±3.15	62.89±3.08	43.25±2.86
DAIMC	86.73±0.79	81.92±0.88	60.44±6.87	76.65±1.07	68.77±0.99	47.10±4.79	86.73±0.79	81.92±0.88	61.24±0.42
OMVC	55.00±5.06	36.40±4.93	29.80±4.63	44.99±4.56	35.16±4.62	25.83±8.37	55.89±4.72	38.51±4.87	31.95±5.22
OPIMC	76.45±5.15	69.50±6.54	56.66±10.06	73.74±3.42	66.57±4.18	51.86±7.97	78.96±3.37	72.00±6.39	58.16±10.35
MKKM-IK-MKC	69.07±0.73	66.08±3.25	55.55±1.39	65.42±0.61	59.04±2.69	47.36±1.78	73.12±0.61	66.58±3.26	56.26±1.07
UEAF	76.11±7.74	65.39±5.09	61.11±1.41	69.37±3.31	55.09±2.05	50.56±1.11	76.51±7.17	66.49±4.18	61.60±1.09
Ours	**93.07±0.41**	**91.74±0.43**	**84.43±1.18**	**86.12±0.64**	**83.39±0.91**	**72.23±1.08**	**93.07±0.41**	**91.74±0.43**	**84.43±1.18**

Table 2: Results on **Caltech7** database with different incomplete rates. The 1st and 2nd best results are marked in **bold** and underline.

Method	**ACC (%)**			**NMI (%)**			**Purity (%)**		
	10%	30%	50%	10%	30%	50%	10%	30%	50%
BSV	43.89±1.37	39.06±1.26	38.31±1.68	39.66±2.23	31.63±1.51	26.81±1.38	84.08±1.23	75.25±0.71	68.97±0.49
Concat	41.25±1.67	40.55±1.89	38.06±0.88	43.48±0.92	37.99±2.17	30.28±0.66	84.91±0.50	82.54±1.12	77.56±0.98
MIC	44.07±4.97	38.01±2.12	35.80±2.34	33.71±2.66	27.35±1.69	20.44±0.98	78.12±1.76	73.31±0.72	68.26±1.40
DAIMC	48.29±6.76	47.46±3.42	44.89±4.88	44.61±3.88	38.45±2.88	36.28±2.34	83.32±1.31	76.83±3.23	75.50±1.17
OMVC	40.88±1.54	36.82±1.65	33.28±4.40	28.13±2.54	25.32±1.03	18.76±4.22	79.21±1.77	77.73±1.35	74.05±4.74
OPIMC	49.24±2.89	48.34±4.36	44.12±5.85	42.98±1.02	41.54±2.38	35.98±2.77	84.89±0.69	83.70±1.80	80.64±2.06
MKKM-IK-MKC	36.54±0.51	34.87±1.53	36.05±0.45	24.09±0.98	23.45±0.52	22.91±0.67	72.98±0.80	73.82±0.53	72.52±1.55
UEAF	50.82±4.05	42.71±0.84	36.32±4.22	39.44±2.07	31.07±1.99	24.02±1.37	81.49±1.78	78.26±2.12	76.29±1.93
Ours	**53.13±2.65**	**51.42±1.57**	**51.23±3.32**	**54.56±2.68**	**52.48±0.83**	**50.29±1.37**	**85.17±0.91**	**84.02±0.69**	**83.79±0.45**

Table 3: Results on **NH_face** database with different incomplete rates. The 1st and 2nd best results are marked in **bold** and underline.

Method	**ACC (%)**			**NMI (%)**			**Purity (%)**		
	10%	30%	50%	10%	30%	50%	10%	30%	50%
BSV	69.09±4.76	56.82±2.28	46.54±1.90	56.26±4.07	39.29±2.63	26.20±1.09	73.59±2.96	60.13±1.52	50.15±1.28
Concat	85.87±2.64	63.14±2.78	52.99±1.84	81.46±1.70	59.12±1.14	47.42±1.29	87.39±1.57	87.39±1.57	62.21±1.04
MIC	78.83±4.07	77.22±0.76	75.77±4.05	73.04±2.78	66.82±0.80	62.84±3.20	82.54±1.66	78.83±0.64	77.40±3.48
DAIMC	87.42±4.15	85.35±3.44	84.57±3.49	78.37±3.42	74.71±2.91	70.09±5.08	87.03±2.74	85.66±2.91	84.66±3.41
OMVC	75.35±2.11	72.85±3.17	70.61±2.77	68.45±3.22	65.44±2.89	63.34±4.36	80.89±3.05	77.96±2.33	74.52±3.55
OPIMC	79.82±8.32	74.57±3.81	71.25±6.27	69.92±6.36	66.87±1.86	64.65±6.94	81.56±5.12	79.02±1.27	78.21±4.01
MKKM-IK-MKC	74.34±0.34	75.92±0.93	71.22±1.19	65.21±0.32	66.83±1.24	65.27±1.66	78.96±0.07	79.18±0.16	79.94±1.03
UEAF	80.36±0.10	71.22±0.68	64.37±1.13	67.11±0.52	55.52±2.55	47.97±1.50	81.67±0.13	73.32±0.70	68.49±1.21
Ours	**97.10±1.02**	**96.40±2.17**	**95.40±4.82**	**93.62±1.49**	**92.03±3.80**	**92.01±4.24**	**97.10±1.02**	**96.40±2.17**	**95.48±4.66**

As shown in Table 1-5, the values of ACC, NMI, and Purity are listed and corresponding standard deviations are given after the sign '±'. We report the nine methods on each dataset with different incomplete rates or paired rates in these tables and mask the best or second-best results in bold or underline. Looking closely at these data, we can easily get a few points: **(1)** Our approach shines brightly, beating other state-of-the-art methods in almost all metrics. For instance, our SMVC exceeds the second-best DAIMC in the ACC metric by approximately 10, 11, and 11 percentage points on NH_face dataset with three different incomplete rates, respectively. The good performance of our method on the clustering demonstrates that the high-level semantic representation extracted by it is effective and solid. **(2)** Comparing all results horizontally, we conclude that the harm of

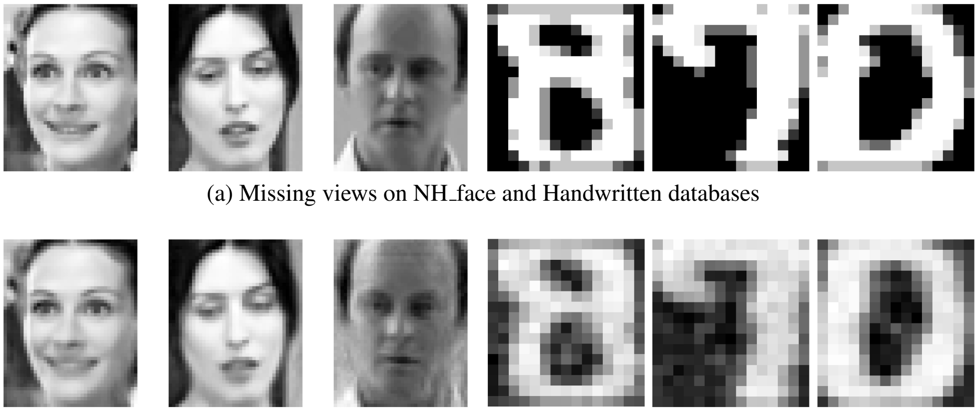
Table 4: Results on **Animal** database with different paired rates. The 1st and 2nd best results are marked in **bold** and underline.

Method	ACC (%)			NMI (%)			Purity (%)		
	30%	50%	70%	30%	50%	70%	30%	50%	70%
BSV	42.05±1.20	48.63±1.89	56.22±1.20	48.16±0.44	55.91±0.58	63.99±0.38	45.20±0.88	52.26±1.19	60.31±0.78
Concat	42.79±0.67	49.34±1.39	53.99±0.99	55.46±0.16	59.31±0.38	63.88±0.35	48.12±0.45	53.24±0.88	59.26±0.81
MIC	43.38±0.63	45.88±0.34	49.15±0.88	52.79±0.77	55.69±0.36	59.30±0.54	49.21±0.78	52.31±0.34	55.33±0.64
DAIMC	50.18±2.18	53.87±1.36	56.42±1.37	55.03±1.03	59.36±1.16	62.76±0.46	54.82±1.57	59.51±1.65	62.12±1.04
OMVC	42.51±0.89	43.98±0.77	46.39±1.02	50.77±0.63	53.11±0.83	55.38±0.46	47.33±0.66	50.42±0.91	52.97±0.76
OPIMC	46.33±2.14	53.14±1.38	53.88±1.26	52.34±0.69	58.51±0.46	62.04±0.26	49.49±1.41	56.23±1.20	57.91±0.43
MKKM-IK-MKC	51.77±0.48	**57.75±0.38**	**61.18±0.59**	56.54±0.33	61.66±0.22	66.28±0.27	56.14±0.48	**62.14±0.41**	**66.40±0.53**
UEAF	45.73±12.9	51.86±6.48	58.19±3.04	51.61±12.87	58.43±7.53	64.92±3.95	49.10±0.27	55.36±0.36	63.02±0.47
Ours	**52.90±0.50**	56.00±0.72	59.20±0.43	**59.65±0.30**	**63.32±0.92**	**66.49±0.77**	**57.78±0.11**	61.28±0.71	64.12±1.07

Table 5: Results on **Aloi_deep** database with different incomplete rates. The 1st and 2nd best results are marked in **bold** and underline.

Method	ACC (%)			NMI (%)			Purity (%)		
	10%	30%	50%	10%	30%	50%	10%	30%	50%
BSV	64.14±1.23	50.63±2.20	37.37±1.34	81.29±0.65	63.15±0.62	45.36±0.59	69.89±1.13	55.10±1.76	40.18±1.07
Concat	71.07±2.90	59.60±1.26	41.39±1.30	89.75±1.24	77.47±0.46	68.63±0.97	76.52±2.39	64.44±0.88	44.99±1.54
MIC	43.69±2.30	35.54±1.19	27.96±1.39	72.18±2.29	66.16±2.68	59.10±2.82	44.77±0.31	36.30±0.25	28.35±0.18
DAIMC	84.07±1.27	81.99±1.32	69.00±2.75	95.66±0.38	94.78±0.23	87.70±1.63	87.65±0.85	85.64±0.82	72.61±0.63
OMVC	63.13±1.43	51.02±1.45	35.18±0.62	80.99±1.37	69.54±0.87	57.91±0.80	67.58±1.30	55.59±1.36	39.37±0.72
OPIMC	47.09±1.77	35.07±1.99	33.97±1.64	77.56±1.01	69.05±0.79	67.62±1.71	51.17±0.37	36.51±0.29	34.73±0.28
MKKM-IK-MKC	83.23±1.16	83.80±1.86	83.56±1.54	95.52±0.26	95.44±0.43	95.03±0.46	86..90±1.06	87.06±1.56	86.58±1.42
UEAF	82.74±1.44	75.69±2.00	72.11±1.91	93.92±0.28	87.45±0.51	88.87±0.49	85.91±0.82	78.71±0.59	75.85±0.65
Ours	**93.03±0.36**	**91.53±0.74**	**90.89±1.15**	**98.54±0.07**	**98.19±0.11**	**97.72±0.19**	**94.78±0.33**	**93.60±0.44**	**92.88±0.77**

missing views to multi-view learning is definite and the higher missing rates lead to worse learning outcomes in most cases, which is intuitive and comprehensible. Besides, different IMC methods have different immunity to data incompleteness, such as, UEAF and our approach are relatively insensitive to the missing views due to its powerful view recovery capability.



(a) Missing views on NH_face and Handwritten databases

(b) Recoverd views on NH_face and Handwritten databases

Figure 3: The visual example pairs about missing views and their restoration results. The (a) shows the missing views sampling from the NH_face and Handwritten databases; The (b) denotes corresponding views recovered by our SMVC.

In Figure 3, we give the six visual recovery results by reshaping the pixel feature in the first view of Handwritten database and the second view of NH_face database. The first row is the raw missing views, and the second row denotes the recovery results of our method. As we can see, our SMVC has an amazing recovery effect on missing views, which is very beneficial for the subsequent generation of approximately complete graph.

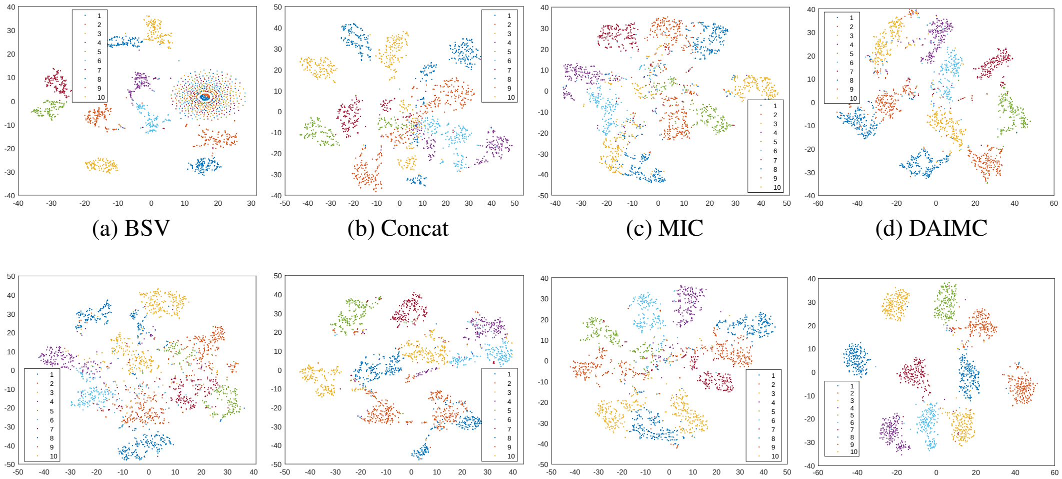
4.3 Ablation study

To confirm the effectiveness of each component of our SMVC, we respectively remove the cross-view en-decoder module leaving only the FC layers, recurrent graph constraint, and missing indicator matrix to form new models. Results of these methods on Handwritten and NH_face databases with a 50% missing ratio are shown in Table 6. From the table, it is clear that the full version of SMVC achieves the best performance. Furthermore, we find that the introduction of missing prior matrix is crucial for the recovery phase of SMVC, which well avoids the negative influence of missing data.

Table 6: The ablation experiments on two datasets with a 50% missing ratio. $\mathcal{B}$ is the backbone with only linear layers; $\mathcal{C}$ denotes the cross-view en-decoder module; $\mathcal{G}$ represents the current graph constraint; and incomplete mask means the missing prior matrix introduced in SMVC.

method	Handwritten			NH_face		
	ACC	NMI	PUR	ACC	NMI	PUR
$\mathcal{B}$	85.80	77.43	85.80	49.61	27.70	54.29
$\mathcal{B}+\mathcal{C}$	89.85	80.02	89.85	82.81	77.24	84.70
$\mathcal{B}+\mathcal{G}$	88.05	77.99	88.05	79.98	73.05	82.85
$\boldsymbol{\mathcal{B}+\mathcal{C}+\mathcal{G}}$	**91.74**	**83.39**	**91.74**	**95.40**	**92.01**	**95.48**
w/o incomplete mask	55.40	47.92	55.50	58.78	48.62	64.29

4.4 T-SNE VISUALIZATION RESULTS



(a) BSV (b) Concat (c) MIC (d) DAIMC

(e) OMVC (f) MKKM-IK-MKC (g) UEAF (h) SMVC

Figure 4: Feature space visualization of final clustering representations of different methods via t-SNE on the Handwritten dataset with a 30% incomplete rate.

In Figure 4, we show the final clustering representation of different methods on the Handwritten database with a 30% missing rate via the t-SNE (Van der Maaten & Hinton, 2008) technology. The result of OPIMC is ignored in the figure because it obtains the clustering result directly without producing any clustering indicator matrix. Comparing all t-SNE results, it's easy to find that our SMVC enjoys the best discrimination performance than other state-of-the-art methods.

5 CONCLUSION

In this paper, we propose a general IMC model with a two-stage training strategy that can handle all kinds of random missing datasets. Distinct from most existing methods, our approach focuses on cleverly recovering missing views and performing complete multi-view clustering. To do this, we design a transformer-style cross-view autoencoder and propose a structure-aware recurrent graph constraint that circularly promotes the restoration of incomplete views and the preservation of geometry structure within the views, which help to obtain more discriminative semantic fusion information. Sufficient experimental results confirm that our SMVC has obvious advantages over other top methods. At the same time, our model can be easily extended to other multi-view classification or regression models, which can provide more solid data support for incomplete multi-view learning just by inputting the consistent representation into the classifier or regression layer.

REFERENCES

Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

Xiao Cai, Feiping Nie, and Heng Huang. Multi-view k-means clustering on big data. In *International Joint conference on artificial intelligence*. Citeseer, 2013.

Xiaochun Cao, Changqing Zhang, Chengju Zhou, Huazhu Fu, and Hassan Foroosh. Constrained multi-view video face clustering. *IEEE Transactions on Image Processing*, 24(11):4381–4393, 2015.

Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *IEEE Conference on Computer Vision and Pattern Recognition workshop*, pp. 178–178. IEEE, 2004.

Jan-Mark Geusebroek, Gertjan J Burghouts, and Arnold WM Smeulders. The amsterdam library of object images. *International Journal of Computer Vision*, 61(1):103–112, 2005.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT Press, 2016.

Zongbo Han, Changqing Zhang, Huazhu Fu, and Joey Tianyi Zhou. Trusted multi-view classification. In *International Conference on Learning Representations*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Menglei Hu and Songcan Chen. Doubly aligned incomplete multi-view clustering. *arXiv preprint arXiv:1903.02785*, 2019a.

Menglei Hu and Songcan Chen. One-pass incomplete multi-view clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 3838–3845, 2019b.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

Shao-Yuan Li, Yuan Jiang, and Zhi-Hua Zhou. Partial multi-view clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28, 2014.

Zhenglai Li, Chang Tang, Xinwang Liu, Xiao Zheng, Wei Zhang, and En Zhu. Consensus graph learning for multi-view clustering. *IEEE Transactions on Multimedia*, 24:2461–2472, 2021.

Xinwang Liu, Xinzhong Zhu, Miaomiao Li, Lei Wang, En Zhu, Tongliang Liu, Marius Kloft, Dinggang Shen, Jianping Yin, and Wen Gao. Multiple kernel k k-means with incomplete kernels. *IEEE transactions on pattern analysis and machine intelligence*, 42(5):1191–1204, 2019.

Xinwang Liu, Miaomiao Li, Chang Tang, Jingyuan Xia, Jian Xiong, Li Liu, Marius Kloft, and En Zhu. Efficient and effective regularized incomplete multi-view clustering. *IEEE transactions on pattern analysis and machine intelligence*, 43(8):2634–2646, 2020.

J MacQueen. Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*, pp. 281–297, 1967.

Piyush Rai, Anusua Trivedi, Hal Daumé III, and Scott L DuVall. Multiview clustering with incomplete views. In *Proceedings of the NIPS Workshop on Machine Learning for Social Computing*. Citeseer, 2010.

Weixiang Shao, Lifang He, and Philip S Yu. Multiple incomplete views clustering via weighted nonnegative matrix factorization with l 2, 1 regularization. In *Proceedings of the 2015th European Conference on Machine Learning and Knowledge Discovery in Databases-Volume Part I*, pp. 318–334, 2015.

Weixiang Shao, Lifang He, Chun-ta Lu, and S Yu Philip. Online multi-view clustering with incomplete views. In *2016 IEEE International conference on big data (Big Data)*, pp. 1012–1017. IEEE, 2016.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

Yao-Hung Hubert Tsai, Yue Wu, Ruslan Salakhutdinov, and Louis-Philippe Morency. Self-supervised learning from a multi-view perspective. In *International Conference on Learning Representations*, 2020.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Yang Wang, Xuemin Lin, Lin Wu, Wenjie Zhang, Qing Zhang, and Xiaodi Huang. Robust subspace clustering for multi-view data by exploiting correlation consensus. *IEEE Transactions on Image Processing*, 24(11):3939–3949, 2015.

Jie Wen, Zheng Zhang, Yong Xu, Bob Zhang, Lunke Fei, and Hong Liu. Unified embedding alignment with missing views inferring for incomplete multi-view clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 5393–5400, 2019.

Jie Wen, Zheng Zhang, Zhao Zhang, Lunke Fei, and Meng Wang. Generalized incomplete multiview clustering with flexible locality structure diffusion. *IEEE transactions on cybernetics*, 51(1):101–114, 2020a.

Jie Wen, Zheng Zhang, Zhao Zhang, Zhihao Wu, Lunke Fei, Yong Xu, and Bob Zhang. Dimc-net: Deep incomplete multi-view clustering network. In *Proceedings of the 28th ACM international conference on multimedia*, pp. 3753–3761, 2020b.

Jie Wen, Zheng Zhang, Zhao Zhang, Lei Zhu, Lunke Fei, Bob Zhang, and Yong Xu. Unified tensor framework for incomplete multi-view clustering and missing-view inferring. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 10273–10281, 2021.

Baoyuan Wu, Yifan Zhang, Bao-Gang Hu, and Qiang Ji. Constrained clustering and its application to face clustering in videos. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 3507–3514, 2013.

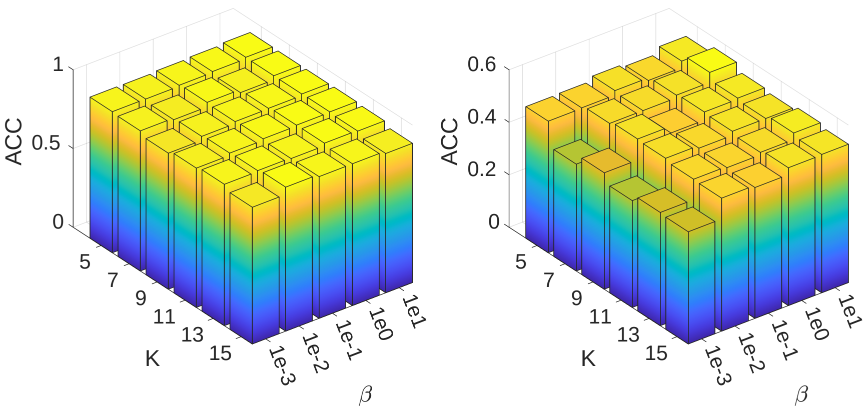
Changqing Zhang, Zongbo Han, Huazhu Fu, Joey Tianyi Zhou, Qinghua Hu, et al. Cpm-nets: Cross partial multi-view networks. *Neural Information Processing Systems*, 32, 2019.

Handong Zhao, Hongfu Liu, and Yun Fu. Incomplete multi-modal visual data grouping. In *IJCAI*, pp. 2392–2398, 2016.

A Appendix

A.1 Hyperparameters sensitivity study

In our SMVC, there are two key hyperparameters that need to be set, *i.e.*, β and K. In order to study the sensitivity of our SMVC to the two parameters, we perform a mesh search in different parameter combinations and plot the results in Figure 5. From the figure, it is not difficult to choose a pair of appropriate parameters for our SMVC. For instance, we can select parameters β and K from the range of $[10^{-3}, 10]$ and $[5, 15]$ separately for the Handwritten database, and the range of $[1, 10]$ and $[5, 15]$ separately for the Caltech7 database.



(a) Handwritten

(b) Caltech7

Figure 5: Results involving different parameter combinations on the Handwritten dataset and Caltech7 dataset with a 50% missing rate.

Algorithm 1 Training process of SMVC

Input: Incomplete multi-view data $\left\{\boldsymbol{X}^{(v)}\right\}_{v=1}^{m}$ with view-presence information, hyperparameters β and K.

Initialization: Construct $\left\{\boldsymbol{W}^{(v)}\right\}_{v=1}^{m}$ according to the view-presence information. Initialize parameters of model. Set training epochs e_1 and e_2 for two stages.

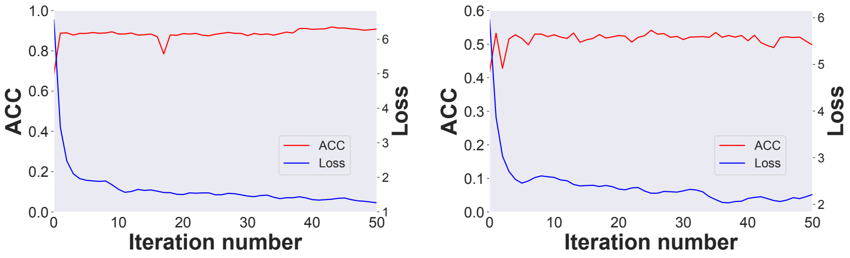
```
while $k \leq e_1$ do                                                        ▷ Stage 1
    Compute multi-view embedding tensor $\mathbf{Z}$.
    if $k > 1$ then
        Compute graph loss $\mathcal{L}_{rg}$ by Eq. (9).
    end if
    Compute reconstructed data $\left\{\bar{\boldsymbol{X}}^{(v)}\right\}_{v=1}^{m}$ by decoder and reconstruction loss $\mathcal{L}_{re}$ by Eq. (6).
    Impute miss data in $\left\{\boldsymbol{X}^{(v)}\right\}_{v=1}^{m}$ with $\left\{\bar{\boldsymbol{X}}^{(v)}\right\}_{v=1}^{m}$ to get new data $\left\{\boldsymbol{X}'^{(v)}\right\}_{v=1}^{m}$.
    Generate graph $\{\mathcal{G}^{(v)}\}_{v=1}^{m}$ using $\left\{\boldsymbol{X}'^{(v)}\right\}_{v=1}^{m}$.
    Compute total loss $\mathcal{L}$ by Eq. (10) and update network parameters.
end while
Update $\left\{\boldsymbol{X}^{(v)}\right\}_{v=1}^{m}$ with $\left\{\boldsymbol{X}'^{(v)}\right\}_{v=1}^{m}$.
while $k \leq e_2$ do                                                        ▷ Stage 2
    Compute multi-view embedding tensor $\mathbf{Z}$.
    Compute graph loss $\mathcal{L}_{rg}$ by Eq. (9).
    Compute fusion representation $\bar{\boldsymbol{Z}}$ by Eq. (5).
    Compute reconstructed data $\left\{\bar{\boldsymbol{X}}^{(v)}\right\}_{v=1}^{m}$ by decoder and reconstruction loss $\mathcal{L}'_{re}$ by Eq. (7).
    Impute miss data in $\left\{\boldsymbol{X}^{(v)}\right\}_{v=1}^{m}$ with $\left\{\bar{\boldsymbol{X}}^{(v)}\right\}_{v=1}^{m}$ to get new data $\left\{\boldsymbol{X}'^{(v)}\right\}_{v=1}^{m}$.
    Compute total loss $\mathcal{L}'$ by Eq. (11) and update network parameters.
end while
Run $K$-means on $\bar{\boldsymbol{Z}}$ to obtain final prediction $\boldsymbol{p}$.
```

Output: $\boldsymbol{p}$

A.2 Algorithm

A.3 Convergence study

In order to study the convergence of our SMVC, in Figure 6, we plot two ACC-Loss curves on the Handwritten dataset and Caltech7 dataset with a 50% missing rate for Stage 2. From the figure, thanks to good data recovery, the loss value decreases steadily while the ACC keeps a slow upward trajectory, which demonstrates that our model has a good convergence.



(a) Handwritten

(b) Caltech7

Figure 6: The ACC and Loss curves on the Handwritten dataset and Caltech7 dataset with a 50% missing rate in Stage 2.

A.4 Detailed information of databases

Table 7: Detailed information about five multi-view databases.

Database	# Class	# View	# Samples	# Features
Handwritten	10	5	2000	76/216/64/240/47
Caltech7	7	6	1474	48/40/254/1984/512/928
NH_face	5	3	4660	6750/2000/3304
Animal	50	2	10158	4096/4096
Aloi_deep	100	3	10800	2048/4096/2048

A.5 Implementation details

In this subsection, we mainly present the implementation details of our model and experimental environment. The two transformer-style cross-view en-decoders are set as one layer with 4 heads. The learning rate is 0.001 and we select the Adaptive Moment Estimation (Adam) as our optimizer. For all databases, we set 50 epochs for each of the two stages. All experiments are performed on a personal computer with an Intel 10700k CPU, RTX2080s GPU, Ubuntu 20.04, PyTorch 1.12.1, and python 3.10.4.