DeConFCluster: Deep Convolutional Transform Learning based Multiview Clustering Fusion Framework

Pooja Gupta, Anurag Goel, Angshul Majumdar, Emilie Chouzenoux, Giovanni Chierchia

DeConFCluster: Deep Convolutional Transform Learning based Multiview Clustering Fusion Framework

Pooja Gupta *
Indraprastha Institute of Information Technology, Okhla Industrial Estate, Phase III, New Delhi, India - 110020
poojag@iiitd.ac.in

Anurag Goel
Delhi Technological University, Shahbad, Daulatpur, New Delhi, India - 110042
Indraprastha Institute of Information Technology, Okhla Industrial Estate, Phase III, New Delhi, India - 110020
anuragg@iiitd.ac.in

Angshul Majumdar
TCG CREST, Sector 5, Salt Lake, Kolkata, West Bengal, India - 700091
angshul.majumdar@tcgcrest.org

Emilie Chouzenoux
Université Paris-Saclay, CentraleSupélec, Inria, CVN, Gif-sur-Yvette, France - 91190
emilie.chouzenoux@inria.fr

Giovanni Chierchia
Université Gustave Eiffel, ESIEE Paris, CNRS, LIGM, Noisy-le-Grand, France – 93162
giovanni.chierchia@esiee.fr

*corresponding author: Pooja Gupta, Indraprastha Institute of Information Technology, Delhi, India; Email: poojag@iiitd.ac.in

DeConFCluster: Deep Convolutional Transform Learning based Multiview Clustering Fusion Framework

Pooja Gupta*[a], Anurag Goel*[a,b], Angshul Majumdar[c], Emilie Chouzenoux[d], Giovanni Chierchia[e]

[a]*Indraprastha Institute of Information Technology, Delhi, India*
[b]*Delhi Technological University, Delhi, India*
[c]*TCG CREST, Kolkata, West Bengal, India*
[d]*Université Paris-Saclay, CentraleSupélec, Inria, CVN, Gif-sur-Yvette, France*
[e]*LIGM, Université Gustave Eiffel, CNRS, ESIEE Paris, Noisy-le-Grand, France*

Abstract

Multi-view data clustering is essential for discovering patterns and exploiting information from different sources. In this context, we propose DeConFCluster, an unsupervised multi-view clustering fusion framework based on Deep Convolutional Transform Learning (CTL). Our approach has the advantage that it does not require an additional decoder network during the training phase. This makes our model less prone to overfitting in data-constrained scenarios, as opposed to several recent studies based on the encoder-decoder framework. Furthermore, our method incorporates a loss function inspired by K-Means, which enables it to learn more effective representations for the clustering task. Finally, we evaluate our framework on five standard multi-view clustering datasets, and show that it outperforms the state-of-the-art multi-view deep clustering techniques.

Keywords: Multiview Clustering, Convolutional Transform Learning, K-Means Clustering, Information Fusion

1. Introduction

Multiview data represents the data gathered from a singular data source but from varying viewpoints or perspectives. For instance, identical news stories may be disseminated or presented across different media platforms with diverse content; identical statements may be categorized with differing tags by distinct individuals; and identical images may be captured utilizing various features. Multiview data is denser and more enlightening, yet concurrently more intricate compared to single-view data. Within multiview data, each viewpoint contains data pertaining to distinct contexts along with supplementary information. Previously, numerous studies have proposed techniques based on multiview learning, which involve utilizing multiview data for classification tasks [1–3]. However, clustering is emerging as another significant application area for multiview data analysis [4–7]. Thus, we delve deeper into discussing techniques related to multiview clustering (MVC) that have been developed for this purpose.

One of the emerging trends in multiview data analysis is graph-based multiview clustering (MVC). The study [8] proposes a Graph-based Multi-view Clustering (GMC), which fuses the data graph matrices of multiple views into a unified graph matrix by generating the similarity induced graph matrices for all the available views. The rank constraint is then applied to the graph Laplacian matrix, and the number of connected components from the unified graph results in the final number of clusters. Another work [9] constructs a joint graph Laplacian containing individual views' denoised cluster information. In [10], the model explores a common joint graph across multiple views.

In recent years, deep unsupervised learning has gained the attention of researchers, as neural network based clustering algorithms usually perform better than standard clustering algorithms. The study [11] presents a comprehensive survey of multiview data clustering approaches relying on the representation learning framework. Summarily, in this survey, the authors have highlighted that deep-representation based learning tools which utilize deep learning models are more successful for MVC. The reason is their inherent ability to learn more complex non-linear functions compared to shallow architectures. In deep clustering, the clustering layer is applied to the representations learned from the deep neural networks. In [4], Deep Clustering Network (DCN) is proposed in which the K-Means clustering is embedded in deep neural networks in a piecemeal manner. First, the representations are learned using deep neural networks, and then the K-Means clustering is applied to the learned representations. Several deep clustering approaches

are based on an autoencoder framework. The clustering layer is embedded into the latent space representation after the encoder layer and trained in a piecemeal fashion. In [12], the authors address the problem of piecemeal training approach of autoencoder framework and clustering module and propose Deep K-Means (DKM). The latter embeds the K-Means clustering module in the latent space after the encoder layer and trains the autoencoder and K-Means in a joint end-to-end fashion instead. The DKM joint formulation outperforms piecemeal versions. Later on, several other deep clustering approaches have been proposed, which embed the clustering module in convolutional [13] or variational [14] autoencoders.

In general, multiview data clustering algorithms target both the complimentary as well as the consensus information contained in multiple views. The work in [15] proposed Deep Embedded Multiview Clustering with collaborative training (DEMVC), by employing deep autoencoders to use the multiple views' complimentary and consensus information and collaboratively learn the deep latent feature representations and clustering assignments. Further, there are recent works that utilize deep learning and graph based MVC together. In [16], multiple autoencoders are employed for multiview data to generate multiple latent representations and apply heterogeneous graph learning to fuse the generated latent representations. The K-Means network is used on the fused multiview latent representations to obtain the final clusters. Though convolutional filters have shown a lot of promises in supervised deep learning, CNNs require labelled data in abundance for training. The workaround is convolutional autoencoders [13]. The problem with autoencoders is that the autoencoders need to learn twice the number of parameters, i.e., both encoder and decoder parameters. Also, CNNs do not guarantee distinctive feature learning. This may lead to redundant representations/features and then may reduce the performance on the task to be performed.

In this proposed work, we present a novel framework -DeConFCluster[2] [18, 19] based on CTL to solve the multiview clustering problem that overcomes all the challenges discussed above. It is an end-to-end multiview framework having multiple channels with one channel per view that learns the view's contribution through channel-wise diverse and interpretable representations. Further, we fuse each view's learned representations via transform learning to learn the cross-channel information as a common representation. Finally, this common representation is passed to the K-Means clustering, and we get the clusters as the end output. The proposed framework is based joint and global optimization, thus, leveraging the advantage of learning representations/features not only from a diversity and interpretability perspective but also from a clustering perspective. The code for the proposed architecture is available on Github platform, in Python language using Pytorch framework[3]. The performance of the proposed framework has been compared with five state-of-the-arts on multiple MVC datasets, namely 100leaves, ALOI, Mfeat, WebKB and Caltech-5V. The contributions of this work are summarized as follows:

- We propose a novel multiview clustering fusion framework DeConFCluster that leverages DCTL based DeConFuse architecture and K-Means clustering. The proposed framework is less prone to overfitting since it does not require the learning of additional deconvolution/decoder layers.
- The proposed framework jointly trains DeConFuse framework and K-Means clustering in an end-to-end fashion and thus, achieves better results than a piecemeal training strategy, as shown in Table 6 of Section 5.
- The performance of DeConFCluster is evaluated on five MVC datasets, namely 100leaves, ALOI, Mfeat, WebKB and Caltech-5V. The proposed framework shows higher clustering performance, especially in data-constrained scenarios where data comprises less number of samples but high number of clusters.

In the following paper, we first discuss the existing literature presented in Section 2. Thereafter, we explain the proposed model and datasets in Section 3. Then we carry on the experiments that includes an ablation study in Section 4 succeeded by the results and analysis in Section 5. Finally, we present concluding discussion about our work in Section 6.

[2] A limited version of this work has been published in the proceedings of a conference [17]. It involved single view clustering without any fusion.

[3] https://github.com/pooja290992/DeConFCluster.git

2. Related Work

Multiview clustering clusters subjects into subgroups using multiview data and has gained significant attention rapidly as it caters in solving real-world problems that fall under big data analytics. Recently many solutions have been proposed to perform the same. These are broadly classified into two categories generative and discriminative approaches. Generative methodologies aim to comprehend the inherent distribution of data by employing generative models, where each model embodies a distinct perspective and subsequently identifies a clustering solution. Conversely, discriminative approaches strive to enhance an objective function by considering pairwise similarities, aiming to minimize the average similarity within each cluster while maximizing it between different clusters. The former typically involves techniques such as expectation maximization and mixture models, while the latter, more diverse in nature, encompasses various sub-categories like multiview spectral clustering, multiview non-negative matrix factorization clustering, multi-kernel clustering, Canonical Correlation Analysis (CCA), among others [20].

In generative methodologies, the study by [21] works under the assumption of independent views and applies a multinomial distribution to tackle the document clustering issue. Similarly, various iterations of the multiview Expectation-Maximization (EM) algorithm for finite mixture models are introduced in [22], each tailored to specific assumptions and criteria. Additionally, by leveraging Convex Mixture Models (CMMs) for single-view clustering, the multiview adaptation outlined in [23] successfully identifies the global optimum while circumventing the initialization and local optima challenges inherent in standard mixture models, which necessitate multiple executions of EM algorithms. The major issue with EM-based algorithms is their sometimes slow convergence, and convergence to local optima. Another limitation in EM-based algorithms is that in some scenarios, the E-step and M-step could be unmanageable analytically since it requires both forward and backward probabilities versus the numerical optimization that requires only forward probability.

Next, in discriminative approaches, we first discuss multiview spectral clustering method [24–28]. This technique achieves a unified clustering outcome by presuming that an identical or similar eigenvector matrix is shared across all perspectives. There are two characteristic methods namely co-training spectral clustering [15, 24–26] and co-regularized spectral clustering [27, 28]. The former is applicable when both labeled and unlabelled data are available while the latter is a semi-supervised learning technique that minimizes the difference between the predictor functions of the two views.

Some methods for multiview data processing are based on subspace clustering [29–33]. Subspace clustering finds the underlying low dimensional common subspace from each view which is, in general, obtained by making each of the view's coefficient matrix as similar as possible. A unified multiview clustering framework is proposed in [32] which simultaneously learns a graph for each view, a partition for each view and a consensus partition. Another study [33] proposes framework for adaptive multi-view subspace clustering. In this, firstly, a low-order representations using self-representation subspace learning are extracted. Next, these are combined with adaptive weights to capture high-order representations in a rotated tensor, effectively handling noise. Lastly, the low-rank tensor using invertible linear transforms based tensor nuclear norm is approximated.

Matrix Factorization (MF) is also used extensively in solving multiview clustering problems. In [34], the geometrical structure of the data is extracted via graph regularization of each view. Next, the virtual label is used to guide matrix factorization. Lastly, the joint framework combining the clustering process and the consensus latent representation learning of data is implemented. Under MF, specifically, several studies propose Non-Negative Matrix Factorization (NMF) [35–39], which aims to decompose matrices into two non-negative factors known as basis and indicator matrices. In the context of MVC, some works suggest learning a shared indicator matrix across multiple views [35, 36] for NMF. Additionally, certain works advocate for utilizing multiview K-Means clustering to manage extensive datasets, leveraging the computational efficiency of K-Means over eigen-decomposition methods. For instance, in [40], the authors propose a multiview K-Means clustering approach that employs a common indicator matrix across different views. Another variant [37] of the previous work introduced a projection matrix for each view's data and then administered MVC by enforcing the common indicator matrix. A novel approach for clustering multi-view data is proposed in [41] by integrating centric graph regularization with log-norm sparsity into NMF. This method enhances clustering performance by preserving local geometric structures and ensuring sparse representations of data across multiple views. A MF-based one-step multi-view clustering with diverse representation (OMVCDR) method is introduced in [42]. OMVCDR projects data matrices into diverse latent spaces and directly utilizes comprehensive knowledge to get clustering results. In addition to NMF, the researchers in [38] introduced a categorical utility func-

tion to assess the similarity between the indicator matrix of each view and the common indicator matrix, presenting a consensus-oriented approach to MVC.

There are set of methods in which direct view combination via a kernel is used as a common approach to perform MVC. Usually, each view is defined by a kernel and then these kernels are combined in a convex combination [7, 43, 44]. Another method, CCA, integrates multiple views following projection [5, 45]. While the previously mentioned techniques have demonstrated satisfactory performance for clustering tasks, handling high-dimensional feature data with nonlinear properties might pose a challenge for these methods. This is because they predominantly rely on shallow and linear embedding functions to unveil the intrinsic structure of multiview data.

Lately, graph-based MVC has emerged as a notable trend [8, 46, 47]. In [8], the authors introduced an approach where the graph matrices from multiple views are combined into a unified graph matrix by generating Similarity Induced Graph (SIG) matrices for each available view. Then the number of clusters obtained is same as the number of connected components produced from the unified graph with the rank constraint. Another work [46] uses geometric mean of the network Laplacian matrices to aggregate different layers of network information into a graph representation. It then employs a neural net to learn a feature embedding and finally obtain clusters. In [48], a framework was proposed that combines the CCA with an autoencoder built upon Graph Convolutional Neural Network layers. Though this study does not perform clustering task but handles multiview datasets and worked with missing views which is believed to be extendible to the clustering task as well. [49] combines multiple perspectives through a multi-view ensemble clustering method, leveraging a joint affinity matrix for enhanced clustering accuracy and robustness by integrating diverse data views into a unified framework. The work [50] proposed a low-rank tensor regularized graph fuzzy learning (LRTGFL) method for multi-view data processing. LRTGFL replaced Euclidean distance with Jensen-Shannon divergence for obtaining more completely nonlinear structures. LRTGFL adopted fuzzy learning to make graph clustering be a soft clustering method.

Deep learning has emerged as a widely employed approach to address various real-world challenges, including MVC. In [16], multiple autoencoders are employed to process multiview data, generating diverse latent representations, which are then fused using heterogeneous graph learning. Subsequently, the K-Means algorithm is applied to obtain the final clusters. Moreover, in [15], a method called Deep Embedded Multi-View Clustering with collaborative training (DEMVC) is proposed, leveraging autoencoders. DEMVC utilizes complementary and consensus information from multiple views, facilitating collaborative learning of deep latent feature representations and clustering assignments.

[51] introduced an approach that integrates deep learning with fuzzy c-means clustering to improve the accuracy and robustness of classification tasks. A collaborative feature-weighted multiview fuzzy c-means clustering approach was introduced in [52] that incorporates multiple views of data and assigns varying weights to features to improve the overall clustering accuracy and robustness. Recently, Federated Multiview Fuzzy C-Means clustering (FedMVFCM) was proposed in [53] that integrates multiple views of data in a federated learning framework and realize the federated optimization procedure. A cluster-based optimization approach that utilizes a parallel bi-objective real-coded genetic algorithm was proposed in [54] for enhancing the efficiency and safety of evacuation processes.

Graph-based learning based methods are also gaining momentum. A Graph Neural Network (GNN) is integrated into deep representation-based MVC to fully exploit the features embedded within the attributed multiview graph data [55]. Furthermore, in [56], Graph Convolutional Network (GCN) serves as an encoder, taking the most reliable view as input. In a separate study, multiple GCN decoders capture the view-consistent low-dimensional feature representation across different views [57]. However, a concern arises regarding the additional weight training introduced by the decoder network, which may lead to overfitting in data-constrained scenarios [17]. Additionally, existing solutions face a limitation with CNNs, potentially resulting in a trivial solution without output. Incorporating Deconvolutional layers is the only approach to mitigate this issue, albeit with the possibility of encountering overfitting even with this solution. Further, CNNs do not guarantee the diversity among learnt filters. This may lead to redundant representations/features which may reduce the performance of the task at hand.

Lately established frameworks - Convolutional Transform Learning (CTL) [58] and Deep CTL (DCTL) [59] can learn convolutional filters in an unsupervised fashion and ensure non-trivial solution. Further, these are extended to a multi-channel fusion framework DeConFuse [60]. Recently, the works [61] and [17] embedded K-Means clustering in Transform Learning and DCTL frameworks respectively. In this work, we propose a multiview multi-channel clustering fusion model named as DeConFCluster that extends DCTL based DeConFuse architecture by embedding K-Means clustering in a joint end-to-end fashion. The proposed framework overcomes the aforementioned shortcomings.

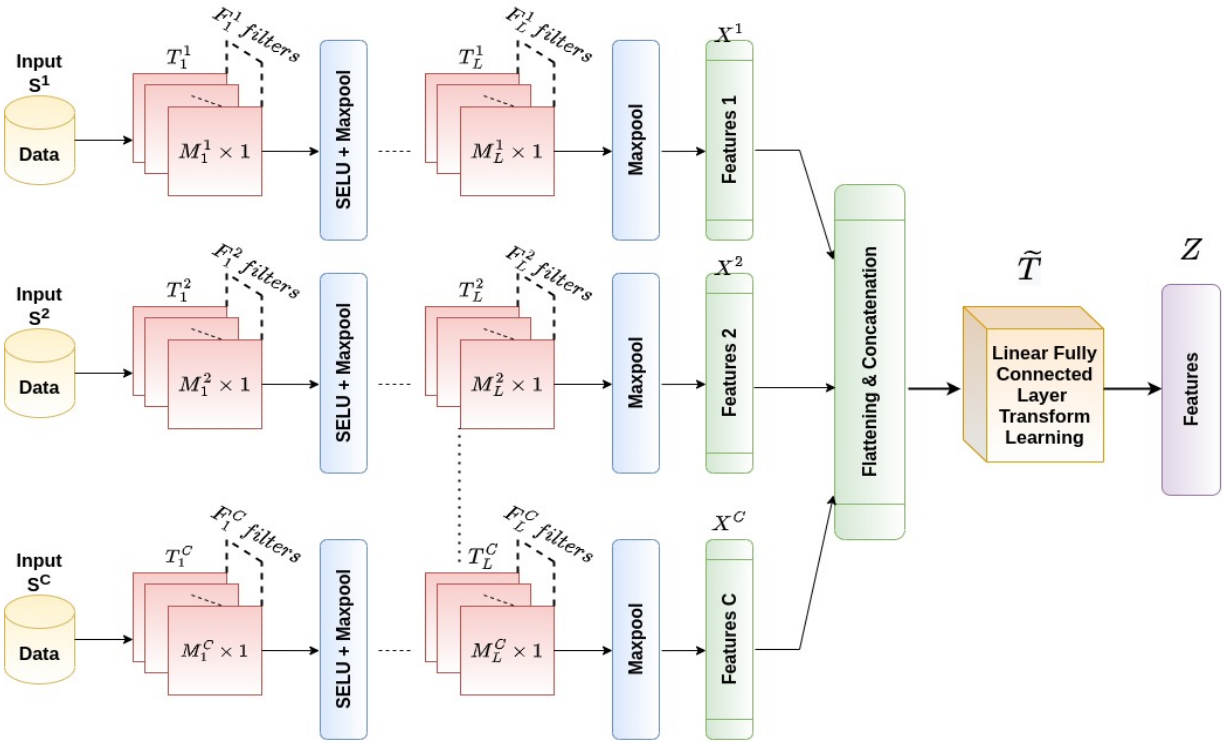

Figure 1: General view of the DeConFuse architecture. C represents number of DCTL network based channels, L is the number of DCTL layers, M_ℓ^c is the filter size and F_ℓ^c is the number of filters of the respective layer ℓ and channel c.

3. Methodology and Preliminaries

Now, we discuss our proposed framework in this section. We propose an unsupervised multi-channel fusion framework that we call DeConFCluster to perform MVC. It combines our previously established works Deep CTL based K-Means clustering framework [17] utilized for single-view clustering and DeConFuse [60] framework and extends it for MVC. We briefly discuss the prior methods mentioned and then explain our proposed formulation in subsequent sections.

3.1. Preliminaries

3.1.1. DeConFuse

We first introduce Deep CTL (DCTL) based architecture for representation learning. Many convolutional layers are stacked to produce features, as shown in Fig. 1. The input S is convoluted with a series of filters $T_1, \ldots, T_L$. The output X is the learned representation/feature corresponding to the convoluted output. The quadratic loss is introduced to measure the discrepancy of the learnt features:

$$\widehat{F}_{\text{conv}}(T_1, \ldots, T_L, X \,|\, S) = \frac{1}{2}\|T_L * \phi_{L-1}(T_{L-1} * \ldots \phi_1(T_1 * S)) - X\|_F^2 \tag{1}$$

where ϕ is an element wise activation function (e.g., SELU and optionally maxpool operation), and L is the number of CTL layers (typically, $L = 3$).

This objective is however not sufficient alone for learning rich and diverse features, as it is trivially minimized by setting all variables to zero. Therefore, we introduce regularization terms that discourage trivial solutions:

$$F_{\text{conv}}(T_1, \ldots, T_L, X \,|\, S) = \frac{1}{2}\|T_L * \phi_{L-1}(T_{L-1} * \ldots \phi_1(T_1 * S)) - X\|_F^2 + \iota_+(X) + \sum_{\ell=1}^{L} (\mu\|T_\ell\|_F^2 - \lambda \log\det(T_\ell)) \tag{2}$$

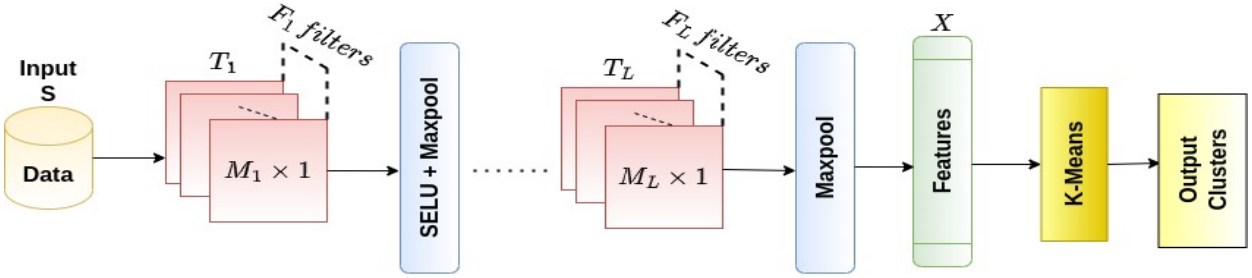

Figure 2: DCKM architecture. L represents number of DCTL layers, M_l^c - filter size and F_l^c - #filters of the respective layer l and channel c.

The term $\iota_+(X)$ is an indicator function of a non-negativity constraint on X that is 0 for negative values of X and $+\infty$ otherwise. The penalization term "$\mu\|\cdot\|_F^2 - \lambda \log\det$" promotes diversity and non degeneracy of the learned filters. The learnable variables are optimized in an end-to-end manner.

DCTL can then be extended into a fusion network where separate DCTL networks/channels represent each view in a multiview dataset to give features $X = (X^{(c)})_{1\le c\le C}$ and then fused to give the common representation Z. The representation Z is learned via linear transforms without convolution as learned in the Transform Learning (TL) technique [62] originally. The learning in this first part of the framework facilitates reduction of the fusion loss. Therefore, the objective function becomes :

$$F_{\text{fusion}}(\widetilde{T}, Z, X) = \frac{1}{2}\Big\|Z - \sum_{c=1}^{C} \text{flat}(X^{(c)})\widetilde{T}_c\Big\|_F^2 + \iota_+(Z) + \sum_{c=1}^{C}\Big(\mu\|\widetilde{T}_c\|_F^2 - \lambda \log\det(\widetilde{T}_c)\Big), \tag{3}$$

Here, "flat" is an operator transforming $X^{(c)}$ into a matrix with every row containing the "flattened" features of the sample. In summary, the DeConFuse architecture is a two-segment framework with first having DCTL based multiple channels and second segment involving fusion. Its training amounts to solving the joint optimization problem:

$$\underset{T,X,\widetilde{T},Z}{\text{minimize}} \;\; F_{\text{fusion}}(\widetilde{T}, Z, X) + \sum_{c=1}^{C} F_{\text{conv}}(T_1^{(c)}, \ldots, T_L^{(c)}, X^{(c)} \,|\, S^{(c)}) \tag{4}$$

DeConFuse architecture is given in Fig. 1. It has noteworthy advantages with the first one enabling to use automatic differentiation [63] and Stochastic Gradient Descent (SGD) techniques to solve (4). Secondly, in (2), advanced activation functions like SELU [64] can be used.

3.1.2. Deep CTL Based K-Means Clustering Framework

The methodology outlined in Section 3.1.1, known as DCTL, has the potential for extension by integrating joint training and optimization with K-Means to conduct single-view clustering as described in [17]. The loss formulation following embedding with K-Means clustering loss is as follows [65]:

$$\underset{T_1,\ldots,T_L,X,H}{\text{minimize}} \;\; \underbrace{F_{\text{conv}}(T_1, \ldots, T_L, X \,|\, S)}_{\text{DCTL loss}} + \beta \underbrace{\|X - XH^\top (HH^\top)^{-1} H\|_F^2}_{\text{K-Means loss}}. \tag{5}$$

Here, X represents the learned representation, S denotes the input, $\beta > 0$ stands for the regularization weight linked with the K-Means clustering loss, and H represents the matrix of binary indicator variables. In this matrix, an entry $h_{ij} = 1$ indicates that x_j belongs to cluster i, and 0 otherwise. The architecture is illustrated in Fig. 2.

3.2. Proposed Formulation

We are now ready to propose our new unsupervised fusion framework that we call DeConFCluster. Previously, the DCKM framework [17] integrated DCTL [59] with K-Means for Single View Clustering (SVC). In contrast, our focus here is on a multiview clustering task. Therefore, DeConFCluster represents a multi-channel clustering framework that extends DeConFuse Network [60] based on DCTL by incorporating the K-Means clustering loss.

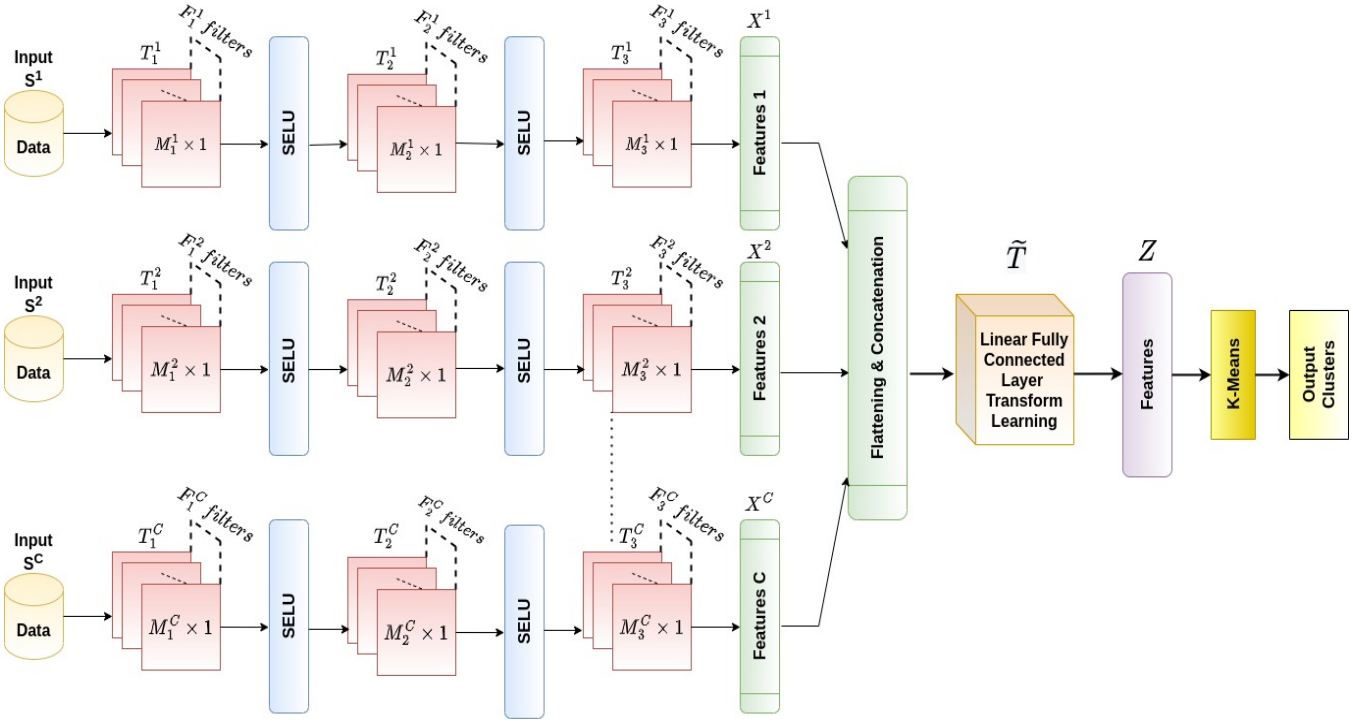

Figure 3: Overview of the proposed DeConFCluster architecture. C represents the number of DeepCTL networks/channels, L is the number of DCTL layers, M_ℓ^c is the filter size and F_ℓ^c is the number of filters of the respective layer ℓ and channel c.

Unlike DCKM, fusion occurs here, where DeConFuse Network and the K-Means module are jointly trained and globally optimized. Number of channels corresponds to the number of views in any of the considered datasets, i.e., $C = V$. The DCTL network processes each channel individually, resulting in unique transforms $(T_c)_{1\le c\le C}$ and, consequently, diverse and interpretable representations $(X_c)_{1\le c\le C}$ for each channel input $(S_c)_{1\le c\le C}$. These channel-wise representations are subsequently fused using Transfer Learning (TL) [62] to learn a common representation Z and transform $\widetilde{T}$, completing the first module of the architecture. The learned representations are then inputted into the second part of the framework, the K-Means clustering module, which produces the clustering results. Thus, the learned representations are also influenced by the K-Means loss. The learning problem is formulated as follows:

$$\underset{T,X,\widetilde{T},Z,H}{\text{minimize}} \; F_{\text{fusion}}(\widetilde{T}, Z, X) + \sum_{c=1}^{C} F_{\text{conv}}(T_1^{(c)}, \ldots, T_L^{(c)}, X^{(c)} \,|\, S^{(c)}) + \beta \|Z - ZH^\top (HH^\top)^{-1} H\|_F^2 \tag{6}$$

The complete architecture of the DeConFCluster is summarized in the Fig. 3.

3.3. Optimization

All variables are trained and optimized in an end-to-end manner. Specifically, we employ alternating minimization in our approach, where at each iteration the network is updated under the assumption that the clustering loss is fixed, and the clustering is updated assuming the network remains constant. The paper [66] demonstrates the convergence of such an alternating minimization procedure for nested non-convex problems like ours. Stochastic Gradient Descent (SGD) is utilized to optimize all variables except H. This particular variable can be directly updated through K-Means clustering [65] at each iteration, utilizing the current estimate of Z as input. We leveraged automatic differentiation provided by the PyTorch framework for all stochastic gradient updates.

4. Experimental Setup

In this section, we illustrate the performance of our approach on various multiview clustering datasets listed below:

- 100leaves: It contains one hundred plant species each of which have 16 samples per specie. Thus, there are 100 clusters and 1600 total samples. Out of these species, few species have leaves with very similar appearances while others show significant variation within the same specie. Hence, it is a challenging dataset to pick due to this added complexity within the same classes. Here, for each sample, shape descriptor, fine scale margin and texture histogram are given [8].
- Amsterdam Library of Object Images (ALOI). ALOI dataset consists of 11025 images of 100 small objects. In this dataset, many images have controlled backgrounds, some variations include more complex backgrounds, which introduce noise that makes the identification and prediction tasks challenging and difficult. Further, every image is represented with multiple features namely Color similarity, HSV, RGB, and Haralick features [67] i.e., constituting four views per image.
- Mfeat: Mfeat dataset is from the UCI repository that contains 2000 samples of handwritten digits (0-9). Each image of this dataset is represented using six different features [8]. Since there are six different views/features, it is a high dimensional dataset where relationships between the different types of features and the digit classes can be highly non-linear.
- WebKB: It consists of 203 web pages with four classes collected from computer science departments of various universities. Each web page is attributed by the page's content, hyperlink's anchor text of the hyperlink and its title text [8]. Since the data belongs to different sources/universities, it becomes difficult to generalize across such diversity for any model. Thus, there could be content variability that might lead to difficulty in prediction and, therefore, makes it a challenging dataset to handle.
- Caltech-5V: It consists of 1400 RGB images with five views namely - WM, CENTRIST, LBP, GIST, and HOG [68, 69]. Here, the images, from which views that have been extracted, have complex and cluttered backgrounds and there are variations in the scales of the objects, lighting and occlusion conditions. Though a balanced dataset, but the aforesaid conditions and the limited data per class makes it a difficult dataset for application.

The complete statistics of all the above mentioned datasets can be referred from Table 1.

Table 1: Statistics of the considered MVC datasets

Datasets	#Samples	#Classes	#Views
100leaves	1600	100	3
WebKB	203	4	3
Mfeat	2000	10	6
ALOI	11025	100	4
Caltech-5V	1400	7	5

4.1. Processing Details

The network's architecture comprises multiple channels, with each channel dedicated to one of the views in the multiview dataset. Initially, each channel's data is normalized using maximum value present in the respective channel's/view's data. Then, representations are learned from the networks associated with these channels, capturing the individual contributions of each view. Subsequently, these representations are flattened and concatenated before passing through a fully connected layer, which is trained via Transform Learning (TL). This step facilitates learning a shared representation across all channels, capturing cross-channel information or shared features from each view. Finally, clusters are obtained by inputting the resulting representation into the K-Means module. The schematic of this flow pipeline is depicted in Fig. 4 and is explained step wise in algorithm 1.

For optimization, Stochastic Gradient Descent (SGD) is employed as the optimizer with $\lambda = 0.01$, $\mu = 0.0001$, and a weight decay of 0.001 for all datasets. Additionally, a hyperparameter named feature_ratio is defined, indicating the percentage of features retained in the final representation Z. The values of all other hyperparameters are determined

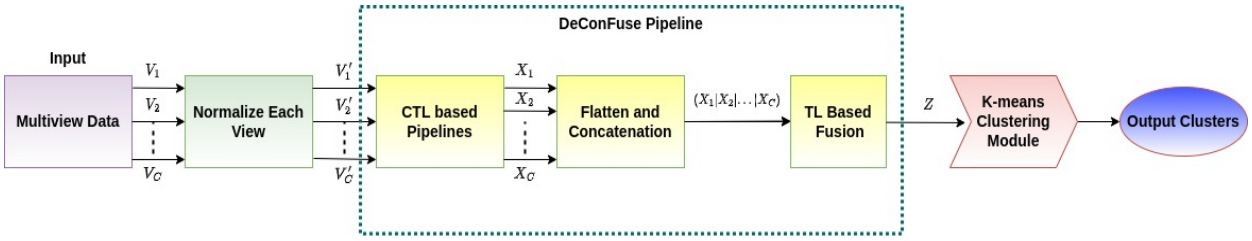

Figure 4: Overview of the proposed DeConFCluster flow diagram. C represents the number of DeepCTL networks/channels and $c \in 1, \ldots, C$, V_c is the individual view data, V_c' is the corresponding normalized view data, X_c is the representation learned channel-wise (view-wise) based on CTL pipelines and Z is the fused representation across all the views' representations learned via TL.

Algorithm 1 DeConFCluster Method

Input : MVD - Array of views belonging to one Multiview Dataset
K - #Samples
C - #Views
Output : OutClusters - clusters for input Z corresponding to input MVD

for $k = 0, \ldots, K-1$ **do**
  for $i = 0, \ldots, C-1$ **do**
    Views[i] = Normalize(MVD[i]) // *Normalizing each view using Max Scaling and storing each view in array Views*
    $X[i]$ = ExtractCTLFeatures(Views[i]) // *extract CTL based features for each view*
  end for
  X_k = FlattenConcatenate(X) // *flatten and concatenate X for each sample*
  Z_k = TLFusion(X_k) // *obtain common representations Z for k^{th} sample*
  $Z = Z.append(Z_k)$ // *append each k^{th} sample fused features to Z*
end for
OutClusters = KMeansClustering(Z) // *fused features of all K samples input to K-means algorithm and returns clusters*
return OutClusters

Table 2: DeConFCluster hyperparameters for MVC Datasets

Parameter	100leaves	WebKB	Mfeat	ALOI	Caltech-5V
Batch size	1600	203	128	11025	1400
Epochs	25	25	40	25	25
Learning Rate	5e-6	1e-4	1e-4	5e-6	1e-4
Kernel Sizes[1]	(3,3,3)	(3,3,3)	(5,3,3)	(3,3,3)	(3,3,3)
#Filters[2]	(4,8,16)	(4,8,16)	(2,4,8)	(4,8,16)	(2,4,8)
feature_ratio	0.15	0.15	0.25	0.25	0.10
λ, μ	$10^{-2}, 10^{-4}$	$10^{-2}, 10^{-4}$	$10^{-2}, 10^{-4}$	$10^{-2}, 10^{-4}$	$10^{-2}, 10^{-4}$
β[3]	1.0	0.5	0.8	0.5	1.0

[1] Kernel sizes for DCTL layers 1,2,3
[2] #Filters for DCTL layers 1,2,3
[3] K-Means loss regularizer

via grid search, with the optimal values selected as the final ones. These final hyperparameter values are summarized in Table 2.

We have compared our results with five state-of-art works. We briefly describe them here that are as follows:

- MCGL: This method is based on graph learning. Initial graphs are constructed from the data points of different

views, which are then optimized by imposing a rank constraint on the Laplacian matrix. Subsequently, these refined graphs are merged into a unified global graph. This global graph is learned while adhering to the same rank constraint on its Laplacian matrix. Cluster indicators are directly derived from the global graph without resorting to any graph cut techniques or employing K-Means clustering. [70].

- GMC: In this method, each view is assigned a weight, and both the SIG matrices and the unified graph matrix are simultaneously learned [8]. The unified graph matrix is derived by fusing the graph matrices from each individual view.
- DEMVC: This approach introduces a framework centered around autoencoders. It leverages complementary and consensus information from various views, facilitating collaborative learning of deep latent feature representations and clustering assignments [15].
- RRA-MVC: This approach introduces a basic baseline model called SiMVC, which aims to align the distributions of different views. Additionally, it incorporates a contrastive module and selective views alignment by prioritizing certain views, enhancing the performance of the baseline model. This improved version is referred to as the CoMVC framework [71]. Therefore, we only conducted experiments using the CoMVC framework, as it yielded the most favorable results.
- MFLVC: It is a framework for multi-level feature learning that facilitates contrastive multi-view clustering [69]. This framework encompasses the learning of features at both low and high levels, as well as semantic labels or features, without the need for fusion. This approach aids in attaining reconstruction and consistency objectives across various feature spaces.

4.2. Software and Hardware Implementation Details

Our entire architecture is programmed in Python 3.6 language with the help of Sklearn, NumPy packages and Pandas libraries. It also builds upon the Pytorch Framework for achieving convolution and optimization related operations. The hardware machine utilized for conducting the experiments is a Dell T30, Xeon E3-1225V5 3.3GHz with GeForce GT 730, and Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz 16GB RAM, 200GB HDD, equipped with Nvidia 1080 8GB and Ubuntu OS.

5. Results and Analysis

We have used three metrics, namely Accuracy, Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI) to evaluate our framework's performance. The results of our model and benchmark algorithms on all five datasets are presented in Table 3. It is evident from the table that across all datasets, our proposed model exhibits better performance compared to state-of-the-art methods, particularly notable in the cases of the 100leaves, WebKB, and Caltech-5V datasets. Specifically, our model demonstrates substantial improvements of 10.62%, 5.81%, and 10.17% in accuracy over the second best performing benchmark for the 100leaves, WebKB, and Caltech-5V datasets, respectively. On the Mfeat dataset, our model surpasses all benchmarks in terms of accuracy and ARI metrics, although its NMI value slightly lags behind that of benchmark models such as MCGL [70] and GMC [8]. For the ALOI dataset, our model outperforms all state-of-the-art methods in terms of accuracy, while it ranks second in terms of NMI and ARI metrics following the RRA-MVC benchmark [71]. This discrepancy may be attributed to the limited variability of features across classes in the Mfeat and ALOI datasets. Notably, our model performs admirably on the challenging 100leaves and WebKB datasets, both of which have a scarcity of samples per class. For instance, the 100leaves dataset only contains 16 images per class, posing a significant challenge for feature representation learning with limited samples. Despite lagging marginally in performance on certain datasets, our proposed method demonstrates overall strong performance across all datasets.

We have additionally generated convergence plots for several datasets, which can be found in Fig. 5. By employing Stochastic Gradient Descent (SGD) as an optimizer, it is evident that our method converges to a stable point. The specific SGD parameters, including mini-batch size and learning rate, are listed in Table 2 for all datasets under consideration.

Table 3: Clustering Results

Dataset	Metric	MCGL	GMC	DEMVC	RRA-MVC	MFLVC	Proposed
100leaves	**Acc**	81.06	82.38	6.69	73.25	11.38	**91.13**
	NMI	91.30	92.92	24.53	92.56	56.89	**96.59**
	ARI	51.50	49.74	0.60	71.58	10.61	**88.01**
WebKB	**Acc**	54.19	76.35	49.75	40.89	45.32	**80.79**
	NMI	8.60	41.64	10.05	13.43	12.41	**54.98**
	ARI	4.01	42.80	8.43	9.22	11.70	**52.02**
Mfeat	**Acc**	85.30	88.20	46.45	81.20	85.95	**95.00**
	NMI	**90.55**	90.50	37.53	83.19	86.40	89.22
	ARI	83.13	85.02	24.59	74.36	80.71	**89.89**
ALOI	**Acc**	46.25	57.05	13.52	55.22	24.33	**58.95**
	NMI	66.57	73.50	41.30	**80.79**	56.54	79.75
	ARI	4.41	43.05	8.45	**49.35**	16.87	46.84
Caltech-5V	**Acc**	32.93	34.07	39.14	74.00	80.4	**88.14**
	NMI	46.89	48.40	26.50	68.34	70.3	**78.76**
	ARI	21.33	22.21	18.80	62.14	70.27	**75.01**

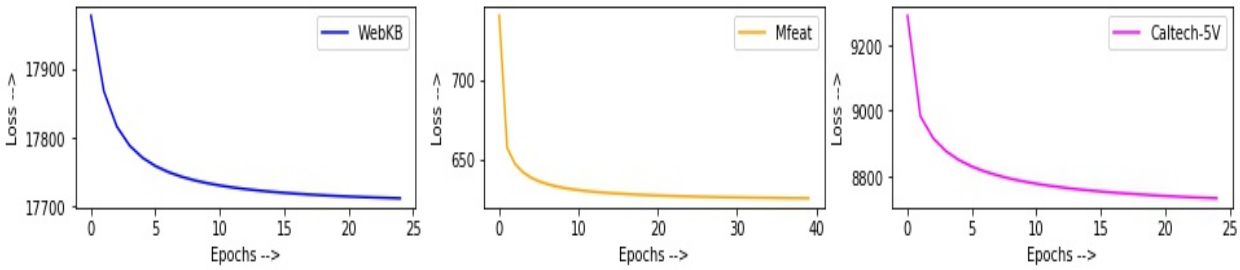

Figure 5: Loss Plots

5.1. Ablation studies

In this section, we have shown the results corresponding to the three ablation studies performed for all the datasets. Firstly, we conducted experiments by varying the values of the regularizers λ and μ, which are associated with the penalty terms log-det and Frobenius norms in both the CTL and TL equations (Equations 2 and 3 respectively). The combinations of values tested for both penalty regularizers included $\{(10^{-2}, 10^{-4}), (10^{-2}, 10^{-3}), (10^{-3}, 10^{-4}), (10^{-3}, 10^{-5}), (10^{-4}, 10^{-5})\}$. This set of values are taken from the past experience. The results of these experiments are provided in Table 4, and they are visually represented for all three metrics (Accuracy, NMI, and ARI) in Fig. 6. The analysis of the impact of regularization weights on performance across various datasets reveals that the importance of these regularizers varies by dataset. For four datasets, performance remains consistent despite changes in regularization weights, indicating that these weights have minimal impact. This robustness is likely due to the high inter-class variability in these datasets, which prevents trivial solutions and reduces reliance on regularizers.

In contrast, the Mfeat dataset shows a significant deterioration in performance with lower regularization weights. This underscores the critical role of regularizers in learning better representations for Mfeat. The Mfeat dataset's lower inter-class variability makes it more susceptible to overfitting or trivial solutions without adequate regularization. Therefore, penalization terms are essential in this context to maintain performance by promoting better generalization and representation learning.

In summary, while the role of regularizers may seem less crucialsignificant for datasets with high inter-class variability, they are neededcrucial for datasets like Mfeat that need additional regularization to achieve optimal performance. This highlights the importance of considering dataset characteristics when tuning regularization parameters in deep learning models.

Table 4: Ablation Studies Results on λ, μ

Dataset	Metric	$(10^{-2}, 10^{-4})$	$(10^{-2}, 10^{-3})$	$(10^{-3}, 10^{-4})$	$(10^{-3}, 10^{-5})$	$(10^{-4}, 10^{-5})$
100leaves	**Acc**	91.13	91.13	91.13	91.13	91.13
	NMI	96.59	96.59	96.59	96.59	96.59
	ARI	88.01	88.01	88.01	88.01	88.01
WebKB	**Acc**	80.79	80.79	80.79	80.79	80.79
	NMI	54.98	54.98	54.98	54.98	54.98
	ARI	52.02	52.02	52.02	52.02	52.02
Mfeat	**Acc**	95.00	95.00	94.70	94.70	91.80
	NMI	89.22	89.22	89.49	89.49	86.24
	ARI	89.89	89.89	88.66	88.66	83.74
ALOI	**Acc**	58.95	58.95	58.95	58.95	58.95
	NMI	79.75	79.75	79.75	79.75	79.75
	ARI	46.84	46.84	46.84	46.84	46.84
Caltech-5V	**Acc**	88.14	88.14	88.14	88.14	88.14
	NMI	78.76	78.76	78.76	78.76	78.76
	ARI	75.01	75.01	75.01	75.01	75.01

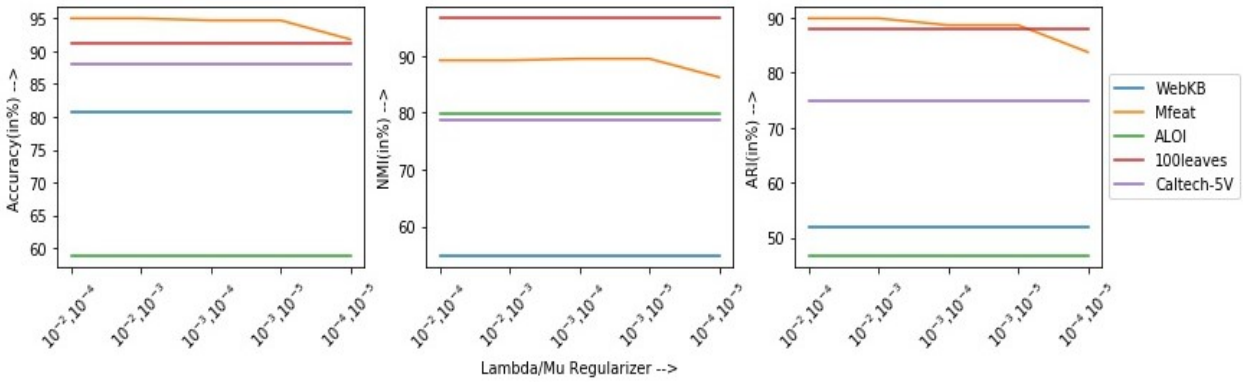

Figure 6: Ablation Studies Result Plots on λ, μ

Next, we conducted experiments involving the regularizer β associated with the K-Means clustering loss in Equation 6. The values for β ranged from 0 to 1, specifically $\{0.0, 0.1, 0.3, 0.5, 0.8, 1.0\}$. We present the results both numerically and graphically, which can be found in Table 5 and Fig. 7, respectively. For most datasets examined, the implementation with K-means regularizer $\beta \geq 0.5$ demonstrated superior performance. This underscores the importance of the K-Means loss term in the final loss function. It plays a crucial role in facilitating the learning of robust representations, thereby contributing to enhanced clustering performance.

The inference drawn from the second experiment is further confirmed by the third experiment, where we conduct a piecemeal version of our model. Specifically, we first learn representations from the DeConFuse network separately and then pass these learned representations through the K-Means clustering module to obtain the final clusters, implying $\beta = 0$. The results of this experiment are provided in Table 6. It is evident from these results that the joint optimization of the DeConFuse and K-Means clustering modules yields better performance compared to the piecemeal approach.

6. Conclusion

In this work, we have proposed a novel unsupervised multi-channel fusion clustering model based on Deep Convolutional Transform Learning named DeConFCluster. The proposed framework jointly trains the DCTL based DeConFuse and K-Means clustering modules in an end-to-end fashion. It does not have the additional overhead of learning

Table 5: Ablation Studies Results on K-Means Regularizer

Dataset	Metric	$\beta = 0.0$	$\beta = 0.1$	$\beta = 0.3$	$\beta = 0.5$	$\beta = 0.8$	$\beta = 1.0$
100leaves	**Acc**	89.56	89.69	87.75	88.69	86.56	**91.13**
	NMI	96.17	**96.62**	96.05	95.91	95.76	96.59
	ARI	86.50	87.26	84.96	85.30	84.24	**88.01**
WebKB	**Acc**	77.83	77.83	77.83	**80.79**	80.79	80.79
	NMI	44.47	44.47	44.47	**54.98**	52.05	51.32
	ARI	52.57	52.57	52.57	52.02	52.81	**53.24**
Mfeat	**Acc**	91.10	94.65	94.45	91.35	**95.00**	81.60
	NMI	85.37	89.39	89.18	85.57	**89.22**	85.62
	ARI	82.15	88.51	88.08	82.79	**89.89**	78.46
ALOI	**Acc**	55.27	54.15	53.72	**58.95**	54.20	55.31
	NMI	78.34	78.41	77.87	**79.75**	78.61	79.40
	ARI	41.16	41.49	38.65	**46.84**	40.80	42.89
Caltech-5V	**Acc**	84.64	84.57	88.00	87.93	87.50	**88.14**
	NMI	75.71	75.50	**78.97**	78.08	77.49	78.76
	ARI	69.45	69.43	74.90	74.77	73.92	**75.01**

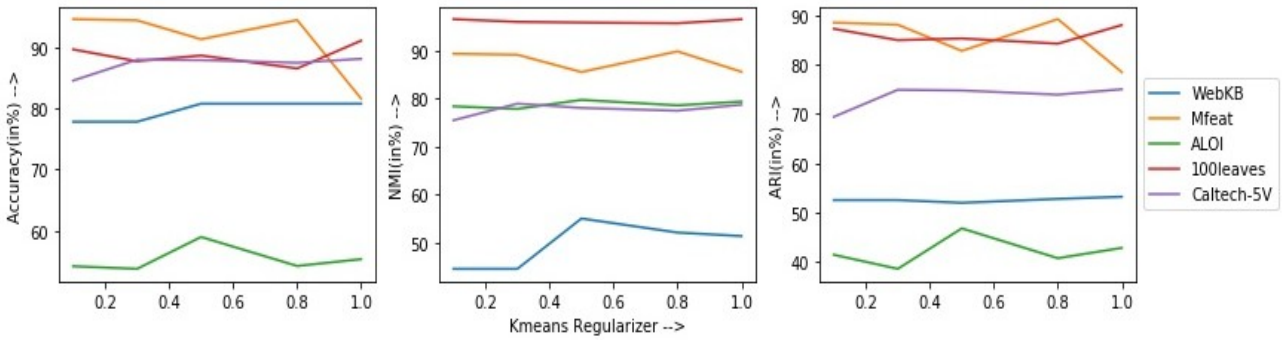

Figure 7: Ablation Studies Result Plots on K-Means Regularizer

the weights of decoder or deconvolutional layers, which is the case in existing multi-view clustering approaches. The proposed framework avoids overfitting in data-constrained scenarios where the number of data instances is low and the number of classes is high. The proposed framework encourages diversity among filters, thereby facilitating the learning of more interpretable filters, which are then guided by the K-Means loss. Therefore, due to these advantages, the proposed framework DeConFCluster, evaluated on the five standard multi-view datasets, demonstrated higher clustering scores as compared to the current state-of-the-art MVC frameworks. Currently, the proposed framework is applied to the multi-view clustering datasets that has views that are similar in nature. As a future work, it could be further extended to multi-modal datasets that have different type of views including images, text, audio, etc.

References

[1] Z. Wang, Z. Shen, H. Zou, P. Zhong, Y. Chen, Retargeted multi-view classification via structured sparse learning, Signal Processing 197 (2022) 108538. doi:https://doi.org/10.1016/j.sigpro.2022.108538.
URL https://www.sciencedirect.com/science/article/pii/S0165168422000858

[2] Y. Zhang, X. Guo, H. Ren, L. Li, Multi-view classification with semi-supervised learning for sar target recognition, Signal Processing 183 (2021) 108030. doi:https://doi.org/10.1016/j.sigpro.2021.108030.
URL https://www.sciencedirect.com/science/article/pii/S0165168421000694

[3] Q. Lin, Z. Wang, Y. Chen, P. Zhong, Supervised multi-view classification via the sparse learning joint the weighted elastic loss, Signal Processing 191 (2022) 108362. doi:https://doi.org/10.1016/j.sigpro.2021.108362.
URL https://www.sciencedirect.com/science/article/pii/S0165168421003996

Table 6: Ablation Studies Results on Piecemeal and Proposed Formulation

Dataset	Metric	Piecemeal	Proposed
100leaves	**Acc**	89.56	**91.13**
	NMI	96.17	**96.59**
	ARI	86.50	**88.01**
WebKB	**Acc**	77.83	**80.79**
	NMI	44.47	**54.98**
	ARI	**52.57**	52.02
Mfeat	**Acc**	91.10	**95.00**
	NMI	85.37	**89.22**
	ARI	82.15	**89.89**
ALOI	**Acc**	55.27	**58.95**
	NMI	78.34	**79.75**
	ARI	41.16	**46.84**
Caltech-5V	**Acc**	84.64	**88.14**
	NMI	75.71	**78.76**
	ARI	69.45	**75.01**

[4] B. Yang, X. Fu, N. D. Sidiropoulos, M. Hong, Towards k-means-friendly spaces: Simultaneous deep learning and clustering, in: International Conference on Machine Learning, 2017, pp. 3861–3870.

[5] M. B. Blaschko, C. H. Lampert, Correlational spectral clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008), 2008, pp. 1–8.

[6] X. Zhao, N. Evans, J.-L. Dugelay, A subspace co-training framework for multi-view clustering, Pattern Recognition Letters 41 (2014) 73–82, supervised and Unsupervised Classification Techniques and their Applications. doi:https://doi.org/10.1016/j.patrec.2013.12.003.
URL https://www.sciencedirect.com/science/article/pii/S0167865513004686

[7] G. Chao, S. Sun, Multi-kernel maximum entropy discrimination for multi-view learning, Intelligence Data Analysis 20 (3) (2016) 481–493.

[8] H. Wang, Y. Yang, B. Liu, Gmc: Graph-based multi-view clustering, IEEE Transactions on Knowledge and Data Engineering 32 (6) (2019) 1116–1129.

[9] A. Khan, P. Maji, Multi-manifold optimization for multi-view subspace clustering, IEEE Transactions on Neural Networks and Learning Systems 33 (8) (2022) 3895–3907. doi:10.1109/TNNLS.2021.3054789.

[10] S. Shi, F. Nie, R. Wang, X. Li, Multi-view clustering via nonnegative and orthogonal graph reconstruction, IEEE Transactions on Neural Networks and Learning Systems 34 (1) (2023) 201–214. doi:10.1109/TNNLS.2021.3093297.

[11] M.-S. Chen, J.-Q. Lin, X.-L. Li, B.-Y. Liu, C.-D. Wang, D. Huang, J.-H. Lai, Representation learning in multi-view clustering: A literature review, Data Science and Engineering (2022) 1–17.

[12] M. M. Fard, T. Thonet, E. Gaussier, Deep k-means: Jointly clustering with k-means and learning representations, Pattern Recognition Letters 138 (2020) 185–192.

[13] X. Guo, X. Liu, E. Zhu, J. Yin, Deep clustering with convolutional autoencoders, in: International Conference on Neural Information Processing, 2017, pp. 373–382.

[14] K.-L. Lim, X. Jiang, C. Yi, Deep clustering with variational autoencoder, IEEE Signal Processing Letters 27 (2020) 231–235.

[15] J. Xu, Y. Ren, G. Li, L. Pan, C. Zhu, Z. Xu, Deep embedded multi-view clustering with collaborative training, Information Sciences 573 (2021) 279–290.

[16] X. Yang, C. Deng, Z. Dang, D. Tao, Deep multiview collaborative clustering, IEEE Transactions on Neural Networks and Learning Systems (2021).

[17] A. Goel, A. Majumdar, E. Chouzenoux, G. Chierchia, Deep convolutional k-means clustering, in: Proceedings of the IEEE International Conference on Image Processing (ICIP 2022), Bordeaux, France, 2022, pp. 211–215.

[18] P. Gupta, Information fusion using convolutional transform learning, Phd thesis, Indraprastha Institute of Information Technology Delhi, New Delhi, India, available at https://repository.iiitd.edu.in/xmlui/handle/123456789/1307 (September 2023).

[19] A. Goel, Deep clustering, Phd thesis, Indraprastha Institute of Information Technology Delhi, New Delhi, India, available at https://repository.iiitd.edu.in/xmlui/handle/123456789/1313 (November 2023).

[20] G. Chao, S. Sun, J. Bi, A survey on multiview clustering, IEEE Transactions on Artificial Intelligence 2 (2) (2021) 146–168.

[21] S. Bickel, T. Scheffer, Multi-view clustering, in: Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM 2004), 2004, pp. 19–26. doi:10.1109/ICDM.2004.10095.

[22] X. Yi, Y. Xu, C. Zhang, Multi-view em algorithm for finite mixture models, in: S. Singh, M. Singh, C. Apte, P. Perner (Eds.), Pattern Recognition and Data Mining, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 420–425.

[23] D. Lashkari, P. Golland, Convex clustering with exemplar-based models, in: J. Platt, D. Koller, Y. Singer, S. Roweis (Eds.), Advances in Neural Information Processing Systems, Vol. 20, Curran Associates, Inc., 2007, pp. 825–832.

[24] A. Kumar, H. D. III, A co-training approach for multi-view spectral clustering, in: Proceedings of the 28th International Conference on

International Conference on Machine Learning (ICML 2011), Omnipress, Madison, WI, USA, 2011, p. 393–400.
[25] T. Liu, Guided co-training for large-scale multi-view spectral clustering, CoRR abs/1707.09866 (2017). arXiv:1707.09866.
URL http://arxiv.org/abs/1707.09866
[26] W. Cai, H. Zhou, L. Xu, A multi-view co-training clustering algorithm based on global and local structure preserving, IEEE Access 9 (2021) 29293–29302. doi:10.1109/ACCESS.2021.3056677.
[27] A. Kumar, P. Rai, H. Daume, Co-regularized multi-view spectral clustering, in: J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, K. Weinberger (Eds.), Advances in Neural Information Processing Systems, Vol. 24, Curran Associates, Inc., 2011, pp. 1413–1421.
[28] Y. Ye, X. Liu, J. Yin, E. Zhu, Co-regularized kernel k-means for multi-view clustering, in: Proceedings of the 23rd International Conference on Pattern Recognition (ICPR 2016), 2016, pp. 1583–1588.
[29] C. Zhang, H. Fu, Q. Hu, X. Cao, Y. Xie, D. Tao, D. Xu, Generalized latent multi-view subspace clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 42 (1) (2020) 86–99. doi:10.1109/TPAMI.2018.2877660.
[30] J. Tan, Y. Shi, Z. Yang, C. Wen, L. Lin, Unsupervised multi-view clustering by squeezing hybrid knowledge from cross view and each view, IEEE Transactions on Multimedia 23 (2021) 2943–2956. doi:10.1109/TMM.2020.3019683.
[31] F. Lei, Q. Li, Sequential multi-view subspace clustering, Neural Networks 155 (2022) 475–486.
[32] Z. Kang, X. Zhao, C. Peng, H. Zhu, J. T. Zhou, X. Peng, W. Chen, Z. Xu, Partition level multiview subspace clustering, Neural Networks 122 (2020) 279–288.
[33] Y. Su, Z. Hong, X. Wu, C. Lu, Invertible linear transforms based adaptive multi-view subspace clustering, Signal Processing 209 (2023) 109014. doi:https://doi.org/10.1016/j.sigpro.2023.109014.
URL https://www.sciencedirect.com/science/article/pii/S0165168423000889
[34] X. Liu, P. Song, Incomplete multi-view clustering via virtual-label guided matrix factorization, Expert Systems with Applications 210 (2022) 118408. doi:https://doi.org/10.1016/j.eswa.2022.118408.
URL https://www.sciencedirect.com/science/article/pii/S0957417422015159
[35] S. Yu, L. Tranchevent, X. Liu, W. Glanzel, J. A. Suykens, B. De Moor, Y. Moreau, Optimized data fusion for kernel k-means clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (5) (2012) 1031–1039. doi:10.1109/TPAMI.2011.255.
[36] X. Chen, X. Xu, J. Z. Huang, Y. Ye, Tw-k-means: Automated two-level variable weighting clustering algorithm for multiview data, IEEE Transactions on Knowledge and Data Engineering 25 (4) (2013) 932–944. doi:10.1109/TKDE.2011.262.
[37] J. Xu, J. Han, F. Nie, X. Li, Re-weighted discriminatively embedded k -means for multi-view clustering, IEEE Transactions on Image Processing 26 (6) (2017) 3016–3027. doi:10.1109/TIP.2017.2665976.
[38] H. Liu, Y. Fu, Consensus guided multi-view clustering, ACM Trans. Knowl. Discov. Data 12 (4) (apr 2018). doi:10.1145/3182384.
URL https://doi.org/10.1145/3182384
[39] Y. Dong, H. Che, M.-F. Leung, C. Liu, Z. Yan, Centric graph regularized log-norm sparse non-negative matrix factorization for multi-view clustering, Signal Processing 217 (2024) 109341. doi:https://doi.org/10.1016/j.sigpro.2023.109341.
URL https://www.sciencedirect.com/science/article/pii/S0165168423004152
[40] X. Cai, F. Nie, H. Huang, Multi-view k-means clustering on big data, in: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI 2013), AAAI Press, 2013, p. 2598–2604.
[41] Y. Dong, H. Che, M.-F. Leung, C. Liu, Z. Yan, Centric graph regularized log-norm sparse non-negative matrix factorization for multi-view clustering, Signal Processing 217 (2024) 109341.
[42] X. Wan, J. Liu, X. Gan, X. Liu, S. Wang, Y. Wen, T. Wan, E. Zhu, One-step multi-view clustering with diverse representation, IEEE Transactions on Neural Networks and Learning Systems (2024).
[43] T. Joachims, N. Cristianini, J. Shawe-Taylor, Composite kernels for hypertext categorisation, in: Proceedings of the 18th International Conference on Machine Learning (ICML 2001), 2001, pp. 250–257.
[44] T. Zhang, A. Popescul, B. Dom, Linear prediction models with graph regularization for web-page categorization, in: Proceedings of the 12th ACM International Conference Knowledge Discovery Data Mining (SIGKDD 2006), 2006, p. 821–826.
[45] W. Shao, L. He, C.-t. Lu, P. S. Yu, Online multi-view clustering with incomplete views, in: Proceedings of the IEEE International Conference on Big Data (Big Data 2016), 2016, pp. 1012–1017. doi:10.1109/BigData.2016.7840701.
[46] M. El Gheche, G. Chierchia, P. Frossard, Orthonet: Multilayer network data clustering, IEEE Transactions on Signal and Information Processing over Networks 6 (2020) 152–162. doi:10.1109/TSIPN.2020.2970313.
[47] S. Yu, S. Wang, Z. Dong, W. Tu, S. Liu, Z. Lv, P. Li, M. Wang, E. Zhu, A non-parametric graph clustering framework for multi-view data, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38(15), 2024, pp. 16558–16567.
[48] Y. Kaloga, P. Borgnat, S. P. Chepuri, P. Abry, A. Habrard, Variational graph autoencoders for multiview canonical correlation analysis, Signal Processing 188 (2021) 108182. doi:https://doi.org/10.1016/j.sigpro.2021.108182.
URL https://www.sciencedirect.com/science/article/pii/S0165168421002206
[49] X. Niu, C. Zhang, X. Zhao, L. Hu, J. Zhang, A multi-view ensemble clustering approach using joint affinity matrix, Expert Systems with Applications 216 (2023) 119484.
[50] B. Pan, C. Li, H. Che, M.-F. Leung, K. Yu, Low-rank tensor regularized graph fuzzy learning for multi-view data processing, IEEE Transactions on Consumer Electronics (2023).
[51] M. Yeganejou, S. Dick, Classification via deep fuzzy c-means clustering, in: 2018 IEEE international conference on fuzzy systems (FUZZ-IEEE), IEEE, 2018, pp. 1–6.
[52] M.-S. Yang, K. P. Sinaga, Collaborative feature-weighted multi-view fuzzy c-means clustering, Pattern Recognition 119 (2021) 108064.
[53] X. Hu, J. Qin, Y. Shen, W. Pedrycz, X. Liu, J. Liu, An efficient federated multi-view fuzzy c-means clustering method, IEEE Transactions on Fuzzy Systems (2023).
[54] A. S. Akopov, L. A. Beklaryan, A. L. Beklaryan, Cluster-based optimization of an evacuation process using a parallel bi-objective real-coded genetic algorithm, Cybernetics and information technologies 20 (3) (2020) 45–63.
[55] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE Transactions on Neural Networks 20 (1) (2009) 61–80. doi:10.1109/TNN.2008.2005605.

[56] H. Zhang, G. Lu, M. Zhan, B. Zhang, Semi-supervised classification of graph convolutional networks with laplacian rank constraints, Neural Processing Letters 54 (2022) 1–12. doi:10.1007/s11063-020-10404-7.
[57] S. Fan, X. Wang, C. Shi, E. Lu, K. Lin, B. Wang, One2multi graph autoencoder for multi-view graph clustering, in: In Proceedings of The Web Conference 2020 (WWW '20), 2020, pp. 3070–3076. doi:10.1145/3366423.3380079.
[58] J. Maggu, E. Chouzenoux, G. Chierchia, A. Majumdar, Convolutional transform learning, in: Proceedings of the International Conference on Neural Information Processing (ICONIP 2018), 2018, pp. 162–174.
[59] J. Maggu, A. Majumdar, E. Chouzenoux, G. Chierchia, Deep convolutional transform learning, in: Proceedings of the International Conference on Neural Information Processing (ICONIP 2000), 2020, pp. 300–307.
[60] P. Gupta, J. Maggu, A. Majumdar, E. Chouzenoux, G. Chierchia, Deconfuse: a deep convolutional transform-based unsupervised fusion framework, EURASIP Journal on Advances in Signal Processing 2020 (1) (2020) 1–32.
[61] A. Goel, A. Majumdar, Transformed k-means clustering, in: Proceedings of the 29th European Signal Processing Conference (EUSIPCO 2021), 2021, pp. 1526–1530.
[62] S. Ravishankar, Y. Bresler, Learning sparsifying transforms, IEEE Transactions on Signal Processing 61 (5) (2013) 1072–1086. doi:https://www.doi.org/10.1109/TSP.2012.2226449.
[63] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch, NIPS Autodiff Workshop (2017).
[64] G. Klambauer, T. Unterthiner, A. Mayr, S. Hochreiter, Self-normalizing neural networks, Advances Neural Information Processing Systems 30 (2017) 971—-980.
[65] C. Bauckhage, K-means clustering is matrix factorization, arXiv preprint arXiv:1512.07548 (2015).
[66] E. Gur, S. Sabach, S. Shtern, Convergent nested alternating minimization algorithms for nonconvex optimization problems, Mathematics of Operations Research 48(1) (2022) 53–57. doi:https://doi.org/10.1287/moor.2022.1256.
[67] X. Zhang, L. Zhao, L. Zong, X. Liu, H. Yu, Multi-view clustering via multi-manifold regularized nonnegative matrix factorization, in: Proceedings of the IEEE International Conference on Data Mining (ICDM 2014), 2014, pp. 1103–1108. doi:10.1109/ICDM.2014.19.
[68] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories, Computer Vision and Image Understanding 106 (1) (2007) 59–70, special issue on Generative Model Based Vision. doi:https://doi.org/10.1016/j.cviu.2005.09.012.
URL https://www.sciencedirect.com/science/article/pii/S1077314206001688
[69] J. Xu, H. Tang, Y. Ren, L. Peng, X. Zhu, L. He, Multi-level feature learning for contrastive multi-view clustering, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 16051–16060.
[70] K. Zhan, C. Zhang, J. Guan, J. Wang, Graph learning for multiview clustering, IEEE Transactions on Cybernetics 48 (10) (2018) 2887–2895. doi:10.1109/TCYB.2017.2751646.
[71] D. J. Trosten, S. Lokse, R. Jenssen, M. Kampffmeyer, Reconsidering representation alignment for multi-view clustering, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2021), Los Alamitos, CA, USA, 2021, pp. 1255–1265. doi:10.1109/CVPR46437.2021.00131.