Reconsidering Representation Alignment for Multi-view Clustering

Daniel J. Trosten Sigurd Løkse Robert Jenssen Michael Kampffmeyer
Department of Physics and Technology, UiT The Arctic University of Norway*

Abstract

Aligning distributions of view representations is a core component of today's state of the art models for deep multi-view clustering. However, we identify several drawbacks with naïvely aligning representation distributions. We demonstrate that these drawbacks both lead to less separable clusters in the representation space, and inhibit the model's ability to prioritize views. Based on these observations, we develop a simple baseline model for deep multi-view clustering. Our baseline model avoids representation alignment altogether, while performing similar to, or better than, the current state of the art. We also expand our baseline model by adding a contrastive learning component. This introduces a selective alignment procedure that preserves the model's ability to prioritize views. Our experiments show that the contrastive learning component enhances the baseline model, improving on the current state of the art by a large margin on several datasets[1].

1. Introduction

Several kinds of real world data are gathered from different points of view, or by using a collection of different sensors. Videos, for instance, contain both visual and audible components, while captioned images include both the raw image data and a descriptive text. In both of these examples, the low-level content of the views are vastly different, but they can still carry the same high-level cluster structure. The objective of multi-view clustering is to discover this common clustering structure, by learning from all available views simultaneously.

Learning from multiple sources at once is not a trivial task [6]. However, the introduction of deep learning [33], has led to the development of several promising deep multi-view clustering models [1, 36, 48, 61, 64]. These models efficiently learn from multiple views by transforming each view with a view-specific encoder network. The resulting representations are fused to obtain a common representation for all views, which can then be clustered by a subsequent clustering module.

The current state of the art methods for deep multi-view clustering use adversarial training to align the representation distributions from different views [36, 64].

Aligning distributions leads to view invariant representations, which can be beneficial for the subsequent fusion of views, and the clustering module [64]. View invariant representations preserve the information present in all views, while discarding information that only exists in a subset of views. If the view-specific information is irrelevant to the clustering objective, it will be advantageous for the clustering module that the encoders learn to remove it. Moreover, aligning representation distributions introduces an auxiliary task, which regularizes the encoders, and helps preserve the local geometric structure of the input space. This has been shown to improve single-view deep clustering models [21].

Despite these advantages however, we identify three important drawbacks of distribution alignment for multi-view clustering:

Aligning representations prevents view-prioritization in the representation space. Views are not necessarily equally important to the clustering objective. The model should therefore be able to adaptively prioritize views, based on the information contained in the view representations. However, aligning representation distributions makes it harder for the model to prioritize views in the representation space, by making these distributions as similar as possible.

One-to-one alignment of clusters is only attainable when encoders can separate all clusters in all views. When the clustering structure is only partially present in the individual views, alignment causes clusters to merge together in the representation space. This makes the clustering task more difficult for the subsequent clustering module.

Aligning representation distributions can make it harder to discriminate between clusters. Since adversarial alignment only considers the representation distributions, a given cluster from one view might be aligned with a different cluster from another view. This misalignment of label distributions has been shown to have a negative impact on discriminative models in the representation space [62].

The End-to-end Adversarial-attention network for Multi-

*UiT Machine Learning Group, machine-learning.uit.no

[1]The source code for the experiments performed in this paper is available at https://github.com/DanielTrosten/mvc

modal Clustering (EAMC) [64] represents the current state of the art for deep multi-view clustering. EAMC aligns the view representations by optimizing an adversarial objective on the encoder networks. The resulting representations are fused with a weighted average, with weights produced by passing the representations through an attention network. Following our reasoning above, we hypothesize that the alignment done by the adversarial module may defeat the purpose of the attention mechanism. Thus inhibiting view prioritization, and leading to less separable clusters after fusion. Our hypothesis is supported by the empirical results of EAMC [64], where the fusion weights are close to uniform for all datasets. Equal fusion weights cause all views to contribute equally to the fused representation, regardless of their contents. Moreover, the fusion weights produced by the attention network depend on all the samples within the current batch. Out-of-sample inference is therefore impossible with EAMC, without making additional modifications to the attention mechanism.

In this work, we seek to alleviate the problems that can arise when aligning distributions of representations in deep multi-view clustering. To this end, we make the following key contributions:

- We highlight pitfalls of aligning representation distributions in deep multi-view clustering, and show that these pitfalls limit previous state of the art models.
- We present Simple Multi-View Clustering (SiMVC) – a new and simple baseline model for deep multi-view clustering, without any form of alignment. Despite its simplicity compared to existing methods, our experiments show that this baseline model performs similar to – and in some cases, even better than – current state of the art methods. SiMVC combines representations of views using a learned linear combination – a simple but effective mechanism for view-prioritization. We empirically demonstrate that this mechanism allows the model to suppress uninformative views and emphasize views that are important for the clustering objective.
- In order to leverage the advantages of alignment – *i.e.* preservation of local geometric structure, and view invariance – while simultaneously avoiding the pitfalls, we attach a selective contrastive alignment module to SiMVC. The contrastive module aligns angles between representations at the sample level, circumventing the problem of misaligned label distributions. Furthermore, in the case that one-to-one alignment is not possible, we make the model capable of ignoring the contrastive objective, preserving the model's ability to prioritize views. We refer to this model as Contrastive Multi-View Clustering (CoMVC).

2. Pitfalls of distribution alignment in multi-view clustering

Here, we consider an idealized version of the multi-view clustering problem. This allows us to investigate and formalize our observations on alignment of representation distributions in multi-view clustering. By assuming that, for each view, all samples within a cluster are located at the same point in the input space, we develop the following proposition[2]:

Proposition 1. *Suppose our dataset consists of V views and k ground truth clusters, and we wish to cluster the data according to this ground truth clustering. Furthermore, we make the following assumptions:*

1. *For each view, all observations that belong to the same ground truth cluster, are located at the same point in the input space.*
2. *For a given view v, $v \in \{1, \dots, V\}$, the number of unique points (*i.e. *distinct/separable clusters) in the input space is k_v.*
3. *The views are mapped to representations using view-specific encoders, and subsequently fused according to a linear combination with unique weights.*

Then, the maximum number of unique clusters after fusion is

$$\kappa^{fused}_{aligned} = \min\left\{k, \left(\min_{v=1,\dots,V}\{k_v\}\right)^V\right\} \tag{1}$$

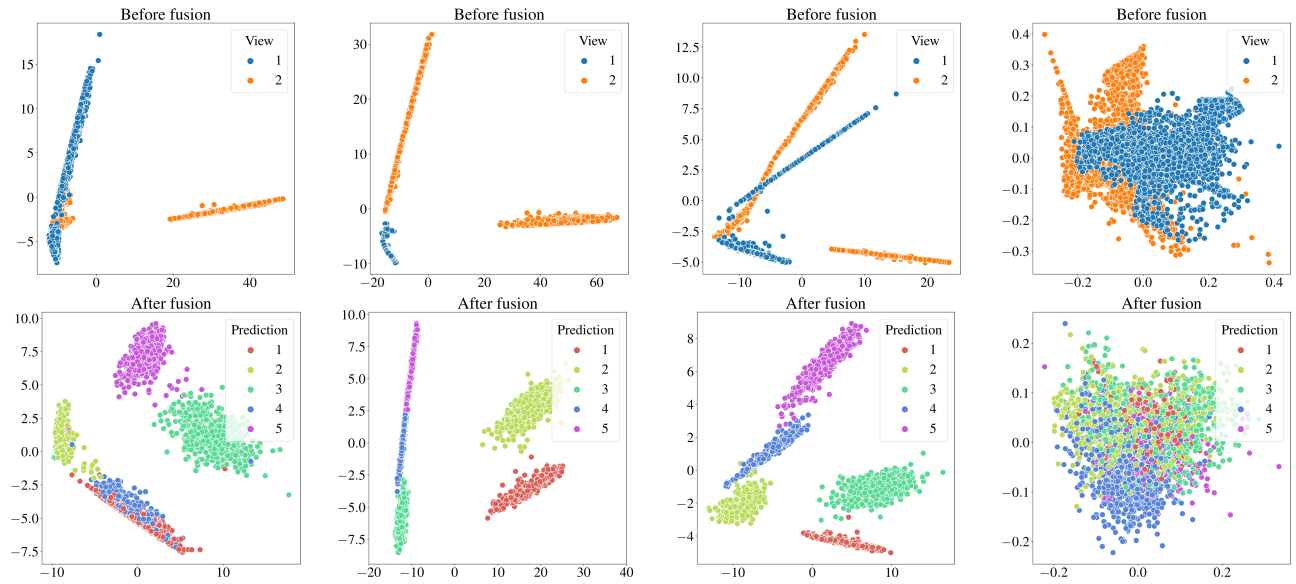
if the distributions of representations from different views are perfectly aligned, and

$$\kappa^{fused}_{not\ aligned} = \min\left\{k, \prod_{v=1}^{V} k_v\right\} \tag{2}$$

if no alignment is performed.

Implications of Proposition 1. $\kappa^{\text{fused}}_{\cdot}$ in Proposition 1 controls how well the clustering module is able to cluster the fused representations. If $\kappa^{\text{fused}}_{\cdot} < k$, it means that some of the clusters are located at the same point after fusion, making it impossible for the clustering module to discriminate between these clusters. In the extreme case that one of the views groups all the clusters together (*i.e.* $k_v = 1$), it follows that $\kappa^{\text{fused}}_{\text{aligned}} = 1$. This happens because all other views are aligned to the uninformative view (for which $k_v = 1$), collapsing the cluster structure in the representation space. Alignment thus prevents the suppression of this view, and makes it harder to discriminate between clusters in the representation space.

[2] We provide a proof sketch for Proposition 1 in the supplementary.



(a) SiMVC +Adv. ACC = 0.80 (b) SiMVC. ACC = 0.99. (c) CoMVC. ACC = 0.99. (d) EAMC. ACC = 0.368.

Figure 1: Representations for SiMVC with and without adversarial alignment, CoMVC, and EAMC on our toy dataset.

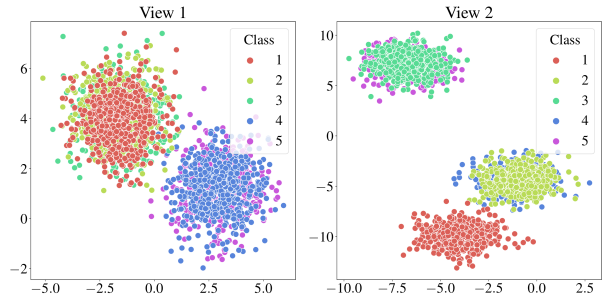


Figure 2: Toy dataset. View 1: Classes (1-3) and (4,5) overlap. View 2: Class 1 is isolated, and classes (2,4) and (3,5) overlap.

However, if we are able to discriminate between all clusters in all views, we have $k_v = k$ for all views, resulting in $\kappa^{\text{fused}}_{\text{aligned}} = \kappa^{\text{fused}}_{\text{not aligned}} = k$. In this case it is possible for both alignment-based models and non-alignment-based models to perfectly cluster the data, provided that the clustering module is sufficiently capable. Alignment-based models can thus benefit from the advantages of alignment, while still being able to separate clusters after fusion.

Experiments on toy data. Proposition 1 makes the simplification that all samples within a cluster are located at the same point, for each view. In order to demonstrate the potential negative impact of aligning representation distributions in a less idealistic setting, and to further motivate the problem, we create a simple two-view dataset. The dataset is shown in Figure 2, and contains five elliptical clusters in two two-dimensional views[3].

We fit SiMVC and SiMVC with adversarial alignment (SiMVC +Adv.) to this dataset, in order to demonstrate the effects of aligning distributions, in a controlled setting. Additionally, we fit our CoMVC and the current state of the art, EAMC, to evaluate more advanced alignment procedures. Note that, for all of these models, the fusion is implemented as a weighted average of view representations, as in Proposition 1. The remaining details on SiMVC and CoMVC are provided in the next section.

Figures 1a and 1b show that attempting to align distributions with adversarial alignment prevents SiMVC from separating between clusters 1 and 4. By adding the adversarial alignment to SiMVC, the number of visible clusters after fusion is reduced from 5 to 4. This is in line with Proposition 1, since we have $\kappa^{\text{fused}}_{\text{aligned}} = 4$ and $\kappa^{\text{fused}}_{\text{not aligned}} = 5$ for this dataset. Figure 1c shows that CoMVC, which relies on the cosine similarity, aligns the angles between the majority of observations. This alignment does not cause classes to overlap in the fused representation. EAMC attempts to align the distributions of view representations (Figure 1d), resulting in a fused representation where the classes are hard to separate. Interestingly, the resulting fused representation exhibits a single group of points, which is significantly worse than the upper bound $\kappa^{\text{fused}}_{\text{aligned}} = 4$ in the analogous idealistic setting. We hypothesize that this is due to EAMC's fusion weights, which we observed to be almost equal for this experiment – thus breaking assumption 3 in Proposition 1.

[3] We repeat this experiment for a 3-cluster dataset in the supplementary.

3. Methods

3.1. Simple Multi-View Clustering (SiMVC)

Suppose our dataset consists of n objects observed from V views. Let $x_i^{(v)}$ be the observation of object i from view v. The objective of our models is then to assign the set of views for each object, $\{x_i^{(1)}, \ldots, x_i^{(V)}\}$, to one of k clusters.

To achieve this, we first transform each $x_i^{(v)}$ to its representation $z_i^{(v)}$ according to

$$z_i^{(v)} = f^{(v)}(x_i^{(v)}) \tag{3}$$

where $f^{(v)}$ denotes the encoder network for view v. We then compute the fused representation as a weighted average

$$z_i = \sum_{v=1}^{V} w_v z_i^{(v)} \tag{4}$$

where $w_1, \ldots, w_v$ are the *fusion weights*, satisfying $w_v > 0$ for $v = 1, \ldots, V$ and $\sum_{v=1}^{V} w_v = 1$. We enforce these constraints by keeping a set of unnormalized weights, from which we obtain $w_1, \ldots, w_V$ using the softmax function. We let the unnormalized weights be trainable parameters – a design choice which has the following advantages: (i) During training, the model has a simple and interpretable way to prioritize views according to its clustering objective. By not relying on an auxiliary attention network, we also make the model more efficient – both in terms of memory consumption and training time[4]. (ii) In inference, the weights act as any other model parameters, meaning that out-of sample inference can be done with arbitrary batch sizes, without any modifications to the trained model. Fixed fusion weights also result in deterministic predictions, which are independent of any other samples within the batch.

To obtain the final cluster assignments, we pass the fused representation through a fully connected layer, producing the hidden representation h_i. This is processed by another fully connected layer with a softmax activation, to obtain the k-dimensional vector of soft cluster assignments, α_i.

Loss function. We adopt the Deep Divergence-based Clustering (DDC) [30] loss, which has shown state of the art performance in single-view image clustering [30]. This is also the clustering loss used by EAMC [64] – the current state of the art method for multi-view clustering.

The clustering loss consists of three terms, enforcing cluster separability and compactness, orthogonal cluster assignments, and closeness of cluster assignments to simplex corners, respectively. The first loss term is derived from the multiple-density generalization of the Cauchy-Schwarz divergence [28], and requires clusters to be separable and compact in the space of hidden representations:

$$\mathcal{L}_1 = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{\binom{k}{2}^{-1} \sum_{a=1}^{n} \sum_{b=1}^{n} \alpha_{ai} \kappa_{ab} \alpha_{bj}}{\sqrt{\sum_{a=1}^{n} \sum_{b=1}^{n} \alpha_{ai} \kappa_{ab} \alpha_{bi} \sum_{a=1}^{n} \sum_{b=1}^{n} \alpha_{aj} \kappa_{ab} \alpha_{bj}}} \tag{5}$$

where k denotes the number of clusters, $\kappa_{ij} = \exp(-||h_i - h_j||^2/(2\sigma^2))$, and σ is a hyperparameter.

The second term encourages the cluster assignment vectors for different objects to be orthogonal:

$$\mathcal{L}_2 = \binom{n}{2}^{-1} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \alpha_i^T \alpha_j. \tag{6}$$

Finally, the third term pushes the cluster assignment vectors close to the standard simplex in $\mathbb{R}^k$:

$$\mathcal{L}_3 = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{\binom{k}{2}^{-1} \sum_{a=1}^{n} \sum_{b=1}^{n} m_{ai} \kappa_{ab} m_{bj}}{\sqrt{\sum_{a=1}^{n} \sum_{b=1}^{n} m_{ai} \kappa_{ab} m_{bi} \sum_{a=1}^{n} \sum_{b=1}^{n} m_{aj} \kappa_{ab} m_{bj}}} \tag{7}$$

where $m_{ij} = \exp(-||\alpha_i - e_j||^2)$, and e_j is corner j of the standard simplex in $\mathbb{R}^k$.

The final clustering loss which we minimize during training of SiMVC is the sum of these three terms:

$$\mathcal{L}_{\text{cluster}} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3. \tag{8}$$

3.2. Contrastive Multi-View Clustering (CoMVC)

Contrastive learning offers a way to align representations from different views at the sample level, forcing the label distributions to be aligned as well. Our hypothesis is therefore that a *selective* contrastive alignment will allow the model to learn common representations that are well suited for clustering – while simultaneously avoiding the previously discussed pitfalls of distribution alignment. Self-supervised contrastive models have shown great potential for a large variety of downstream computer vision tasks [5, 12, 13, 20, 22, 40, 49]. These models learn image representations by requiring that representations from *positive* pairs are mapped close together, while representations from *negative* pairs are mapped sufficiently far apart. In multi-view learning, each object has a set of observations from different views associated with it. This admits a natural definition of pairs: Let views of the *same* object be positive pairs, and views of *different* objects be negative pairs.

[4] Average training times for SiMVC, CoMVC, and EAMC are given in the supplementary.

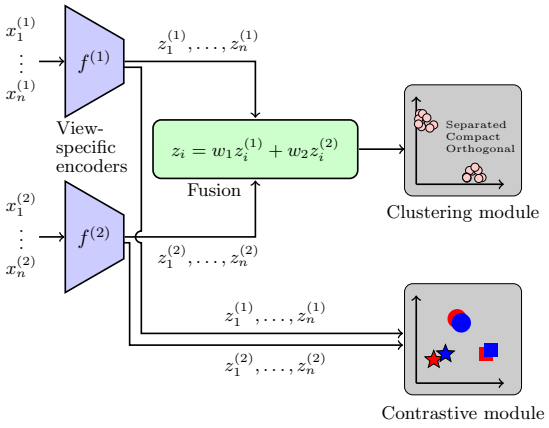


Figure 3: Overview of our proposed models for a two-view dataset. In both SiMVC and CoMVC, the views are first encoded by the view-specific encoder networks $f^{(1)}$ and $f^{(2)}$. The resulting representations are fused with a weighted mean, and then clustered by the clustering module. CoMVC includes an additional contrastive module.

Following [12], we compute the similarity of two representations $z_i^{(v)}$ and $z_j^{(u)}$ as the cosine similarity:

$$s_{ij}^{(vu)} = \frac{z_i^{(v)T} z_j^{(u)}}{||z_i^{(v)}|| \cdot ||z_j^{(u)}||}. \tag{9}$$

Note that in [12], they show that the addition of a projection head between the representations and the similarity, results in better representations – in terms of linear classification accuracy on the learned representations. We found that this was not the case for our model, so we chose to compute the similarity on the representations directly. Experiments comparing versions of our model with and without the projection head can be found in the supplementary.

In order to define a contrastive loss for an arbitrary number of views, we introduce the following generalized version of the NT-Xent loss [12]:

$$\mathcal{L}_{\text{contrastive}} = \frac{1}{nV(V-1)} \sum_{i=1}^{n} \sum_{v=1}^{V} \sum_{u=1}^{V} \mathbb{1}_{\{u \neq v\}} \ell_i^{(uv)} \tag{10}$$

where $\mathbb{1}_{\{u \neq v\}}$ evaluates to 1 when $u \neq v$ and 0 otherwise, and

$$\ell_i^{(uv)} = -\log \frac{e^{s_{ii}^{(uv)}/\tau}}{\sum_{s' \in \text{Neg}(z_i^{(v)}, z_i^{(u)})} e^{s'/\tau}}. \tag{11}$$

Here, τ is a hyperparameter[5], and $\text{Neg}(z_i^{(v)}, z_i^{(u)})$ denotes the set of similarities for negative pairs corresponding to the positive pair $(z_i^{(v)}, z_i^{(u)})$.

[5]We set $\tau = 0.1$ for all experiments, following [12].

A straightforward way to construct $\text{Neg}(z_i^{(v)}, z_i^{(u)})$ would be to include the similarity between all views of object i, and all views of all the other objects within the current batch. However, minimizing Eq. (11) will result in negative samples having a low similarity score. This is indeed the objective of ordinary contrastive learning, but it might be counteractive to the clustering objective, where we want objects from the same cluster to be grouped together in the representation space, and thus be similar to each other. To prevent the contrastive loss from breaking this group structure, we construct $\text{Neg}(z_i^{(v)}, z_i^{(u)})$ in the following manner: First, we define the set

$$\mathcal{N}_i = \{s_{ij}^{(uv)} : j = 1, \dots, n,\ j \neq i, u, v = 1, \dots, V, \\ \arg\max \alpha_i \neq \arg\max \alpha_j\}, \tag{12}$$

which consists of all similarities between all views of object i, and all views of all other objects *that were assigned to a different cluster than object* i. We then construct $\text{Neg}(z_i^{(v)}, z_i^{(u)})$ by sampling a fixed number of similarities from $\mathcal{N}_i$. This procedure ensures that we only repel representations of objects that were assigned to different clusters by the clustering module.

CoMVC is the result of adding this contrastive learning framework to SiMVC. Figure 3 shows a schematic overview of the model for a dataset containing two views.

The loss we use to train CoMVC is

$$\mathcal{L} = \mathcal{L}_{\text{cluster}} + \delta \cdot \min\{w_1, \dots, w_V\} \mathcal{L}_{\text{contrastive}} \tag{13}$$

where $\mathcal{L}_{\text{cluster}}$ is the clustering loss defined in Eq. (8), and δ is a hyperparameter which influences the strength of the contrastive loss. $w_1, \dots w_V$ are the fusion weights from SiMVC[6].

Minimizing the contrastive loss results in representations that have high cosine similarities. The contrastive alignment is therefore (i) *approximate*, since only the angles between representations, and not the representations themselves, are considered; and (ii) *at the sample level*, preventing misaligned label distributions. Furthermore, multiplying the contrastive loss with the smallest fusion weight automatically adjusts the strength of the contrastive loss, according to the weight of the least informative view. The alignment is therefore *selective*: If the model learns to discard a view by setting its fusion weight to 0, it will simultaneously disable the alignment procedure. By adapting the alignment weight and not relying on adversarial training, CoMVC can benefit from the advantages of aligning representations, while circumventing both the drawbacks of adversarial alignment, and possible difficulties with min-max optimization [4, 19].

[6]Note that we do not propagate gradients through the min operation, in order to avoid the trivial solution of setting the smallest fusion weight to 0.

4. Related work

In this section we will give a brief summary of the existing work on multi-view clustering, as well as related work discussing modality alignment in multi-modal learning. Existing methods for multi-view clustering can be divided into two categories: Traditional (non-deep learning based) methods and deep learning based methods.

Traditional methods. Two-stage methods first learn a common representation from all the views, before clustering them using a single-view clustering algorithm [7, 11]. However, recent work shows that letting the learned representation adapt to the clustering algorithm leads to better clusterings [56]. In order to avoid this drawback of two-stage approaches, non-negative matrix factorization [8, 15, 24, 57, 63] has been used to compute the cluster assignment matrix directly from the data matrices. Similarly, subspace methods assume that observations can be represented by one or more self-representation matrices [5, 10, 37, 41, 55, 58, 59, 61] and use the self-representation matrices to identify linear subspaces of the vector space spanned by all the observations, that represent distinct clusters. Alternative popular approaches include methods based on graphs [44, 48, 50, 51, 60, 65] and kernels [16, 18, 35, 39], which both assume that the data can be represented with one or more kernel (or affinity) matrices such that respective clusterings can be found based on these matrices.

Deep learning based methods. Deep learning based two-stage methods [2, 43, 52] work similarly to the two-stage methods described above, but instead use deep neural networks to learn the common representation. However, the two-stage methods are regularly outperformed by deep end-to-end methods that adapt their representation learning networks to the subsequent clustering module. Deep graph-based methods [14, 25, 26, 34] for instance, use affinity matrices together with graph neural networks to directly cluster the data. Similarly, deep subspace methods [1, 3] make the same subspace assumption as above, but compute the self-representation matrix from an intermediate representation in their deep neural network. Lastly, adversarial methods [36, 64] use generators and discriminators to align distributions of hidden representations from different views. These adversarial methods have outperformed the previous approaches to multi-view clustering, yielding state of the art clustering performance on several multi-view datasets.

Distribution alignment. Outside the field of multi-view clustering, the problem of naïvely aligning distributions has recently found increasing attention [62, 53], and led to more efficient fusion techniques [23, 9, 46]. However, this effort has largely been restricted to supervised multi-modal learning frameworks and domain adaptation approaches.

5. Experiments

5.1. Setup

Implementation. Our models are implemented in the PyTorch [45] framework. We train the models for 100 epochs on mini-batches of size 100, using the Adam optimization technique [31] with default parameters. We observe that 100 epochs is sufficient for the training to converge. Training is repeated 20 times, and we report the results from the run resulting in the lowest value of $\mathcal{L}_1$ in the clustering loss. The σ hyperparameter was set to 15% of the median pairwise distance between hidden representations within a mini-batch, following [30]. For the contrastive model, we set the number of negative pairs per positive pair to 25 for all experiments. We set $\delta = 0.1$ for the two-view datasets, and $\delta = 20$ for the three-view datasets. We observe that the three-view datasets benefit from stronger contrastive alignment. Our implementation and a complete overview of the architecture details can be found in the supplementary.

Datasets. We evaluate our models using six well-known multi-view datasets [36, 64], containing both raw image data, and vector data. These are: (i) *PASCAL VOC 2007 (VOC)* [17]. We use the version provided by [27], which contains GIST features and word frequency counts for manually tagged natural images. (ii) *Columbia Consumer Video (CCV)* [29], which consists of SIFT, STIP and MFCC features from internet videos. (iii) *Edge-MNIST (E-MNIST)* [38], which is a version of the ordinary MNIST dataset where the views contain the original digit, and an edge-detected version, respectively. (iv) *Edge-FashionMNIST (E-FMNIST)* [54], which consists of grayscale images of clothing items. We synthesize a second view by running the same edge-detector as the one used to create E-MNIST. (v) *COIL-20* [42], which contains grayscale images of 20 items, depicted from different angles. We create a three-view dataset by randomly grouping the images for an item into groups of three. (vi) *SentencesNYU v2 (RGB-D)* [32], which consists of images of indoor scenes along with descriptions of each image. Following [64], we use image features from a ResNet-50 without the classification head, pre-trained on the ImageNet dataset, as the first view. Embeddings of the image descriptions using a pre-trained doc2vec model on the Wikipedia dataset constitute the second view[7].

Note that, for the datasets with multiple labels, we select the objects with exactly one label. See Table 1 for more information on the evaluation datasets.

Baseline models. We compare our models to an extensive set of baseline methods, which represent the current state of the art for multi-view clustering: (i) Spectral Clustering (SC) [47] on each view, and the concatenation of all views SC(con); (ii) Robust Multi-view K-means Clustering (RMKMC) [8]; (iii) tensor-based Representation Learn-

[7] We provide the details of these pre-trained models in the supplementary.

Dataset	Objs.	Cats.	Views	Dims.
VOC	5649	20	2	512, 399
CCV	6773	20	3	5000, 5000, 4000
E-MNIST	60000	10	2	28×28
E-FMNIST	60000	10	2	28×28
COIL-20	480	20	3	128×128
RGB-D	1449	13	2	2048, 300

Table 1: Summary of the datasets used for evaluation. *Objs.* and *Cats.* denote the number of objects and categories present in the dataset, respectively. *Views* and *Dims.* denote the number of views, and the dimensionality of each view, respectively. Note that for E-MNIST, E-FMNIST, and COIL-20, the input dimensionality is the same for all views.

ing Multi-view clustering tRLMvc [15]; (iv) Consistent and Specific Multi-view Subspace Clustering (CSMSC) [41]; (v) Weighted Multi-view Spectral Clustering (WMSC) [65]; (vi) Multi-view Consensus Graph Clustering (MCGC) [60]; (vii) Deep Canonical Correlation Analysis (DCCA) [2]; (viii) Deep Multimodal Subspace Clustering (DMSC) [1]; (ix) Deep Adversarial Multi-view Clustering (DAMC) [36]; and (x) End-to-end Adversarial attention network for Multi–modal Clustering (EAMC) [64].

Evaluation protocol. To ensure a fair comparison, we report the baseline results over multiple runs, following [64][8]. To assess the models' clustering performance, we use the unsupervised clustering accuracy (ACC) and normalized mutual information (NMI). For both these metrics, higher values correspond to better clusterings.

5.2. Results

Quantitive results on VOC, CCV and E-MNIST are shown in Table 2. The results illustrate that not aligning representations can have a significant improvement (relative gain in ACC larger than 29% on E-MNIST) compared to adversarial alignment, while selective alignment always improves performance. Note that entries for E-MNIST in Table 2 are missing as the number of samples makes the traditional approaches computationally infeasible.

Table 3 compares SiMVC and CoMVC to the previous state of the art, EAMC on E-FMNIST, COIL-20 and RGB-D. Again, we observe that naïvely aligning feature representations tends to worsen performance. This highlights the importance of being considerate when aligning representations in multi-view clustering.

Ablation study. We perform an ablation study in order to evaluate the effects of the different components in the contrastive loss[9]. Specifically, we train CoMVC with and without the proposed negative pair sampling and the adaptive weight factor ($\min\{w_1, \dots, w_V\}$), on E-MNIST and VOC. When we remove the negative sampling, we construct $\mathrm{Neg}(z_i^{(v)}, z_i^{(u)})$ by including the similarities between all views of object i, and all views of all the other objects within the current batch.

[8]The details of the evaluation protocol are given in the supplementary.

[9]We include an ablation study with the DDC loss in the supplementary.

Dataset	VOC		CCV		E-MNIST	
Metric	ACC	NMI	ACC	NMI	ACC	NMI
SC(1)	38.4	39.2	10.2	0.5		
SC(2)	40.2	41.1	18.8	17.3		
SC(3)			11.3	0.8		
SC(con)	37.2	38.7	9.3	7.4		
RMKMC	45.8	46.9	17.6	16.5		
tRLMvc	53.4	54.7	21.2	22.6		
CSMSC	48.8	49.6	19.4	18.6		
WMSC	47.1	46.2	20.5	19.6		
MCGC	52.7	54.6	22.4	21.6		
DCCA	39.7	42.5	17.3	18.2	47.6	44.3
DMSC	54.1	53.8	18.3	19.4	65.3	61.4
DAMC	56.0	55.2	24.3	23.1	64.6	59.4
EAMC	**60.7**	**61.5**	**26.1**	**26.6**	66.8	62.8
SiMVC	55.1 (-5.6)	**61.5** (+0.0)	14.4 (-11.7)	11.2 (-15.4)	**86.2** (+19.4)	**82.6** (+19.8)
CoMVC	**61.9** (+1.2)	**67.5** (+6.0)	**29.5** (+3.4)	**28.7** (+2.1)	**95.5** (+28.7)	**90.7** (+27.9)

Table 2: Clustering metrics [%] on VOC, CCV, and E-MNIST. The best and second best are highlighted in bold. The differences between our models and the best baseline model are shown in parentheses. Green differences indicate improvements. Baseline results are taken from [64].

Dataset	E-FMNIST		COIL-20		RGB-D	
Metric	ACC	NMI	ACC	NMI	ACC	NMI
EAMC	55.2	**62.5**	69.0	75.3	32.3	20.7
SiMVC	**56.8** (+1.6)	50.7 (-11.8)	**77.5** (+8.5)	**91.8** (+16.5)	**39.6** (+7.3)	**35.6** (+14.9)
CoMVC	**59.5** (+4.3)	52.3 (-10.2)	**89.4** (+20.4)	**95.7** (+20.4)	**41.3** (+9.0)	**40.5** (+19.8)

Table 3: Clustering metrics [%] on E-FMNIST, COIL-20 and RGB-D. Same formatting as in Table 2.

	Neg. samp.	Ad. weight	ACC [%]	NMI [%]
E-MNIST	–	–	87.4	86.8
	–	✓	94.7	89.5
	✓	–	87.5	86.6
	✓	✓	**95.5**	**90.7**
VOC	–	–	54.7	61.3
	–	✓	55.3	60.7
	✓	–	58.5	67.4
	✓	✓	**61.9**	**67.5**

Table 4: Ablation study results for CoMVC on E-MNIST and VOC.

	EAMC			SiMVC			CoMVC		
View	1	2	3	1	2	3	1	2	3
VOC	48	52		47	53		64	36	
CCV	26	38	36	32	35	33	1	75	24
E-MNIST	48	52		95	05		67	33	
E-FMNIST	53	47		78	22		99	1	
COIL-20	32	32	36	33	35	32	34	32	34
RGB-D	53	47		59	41		59	41	

Table 5: Fusion weights [%] for EAMC, SiMVC, and CoMVC. For EAMC, we split the entire dataset into batches of size 100 and report the average weight over these batches.

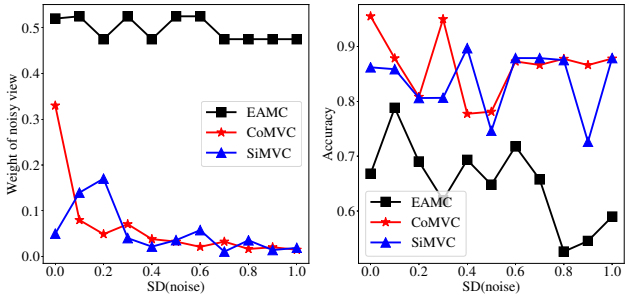


Figure 4: Fusion weights and clustering accuracy (ACC) on E-MNIST, with increasing levels of Gaussian noise added to the second view.

Results of the ablation study (Table 4) show that dropping the adaptive weighting and the negative sampling strategy both have a negative impact on CoMVC's performance. This justifies their inclusion in the final contrastive loss.

View prioritization. Table 5 shows the weight parameters that are obtained for EAMC, SiMVC and CoMVC for all datasets. EAMC always produces close to uniform weight distributions, while SiMVC and CoMVC are able to suppress uninformative views. Note, for datasets, such as COIL-20, where views are assumed equally important[10], we do also observe close to uniform weight distributions for SiMVC and CoMVC.

To further assess our models' capabilities to prioritize views, we corrupt the edge-view (view 2) in E-MNIST with additive Gaussian noise, and record the models' performance as the standard deviation of the noise increases. We also repeat the experiment for the EAMC model, as it represents the current state of the art. Figure 4 shows the resulting fusion weights for the noisy view and the clustering accuracies, for different noise levels. For SiMVC and CoMVC, we observe that the weight of the noisy view decreases as the noise increases. The mechanism for prioritizing views thus works as expected. SiMVC and CoMVC can therefore produce accurate clusterings, regardless of the noise level. Conversely,

[10]Since views in COIL-20 refer to objects depicted from random angles.

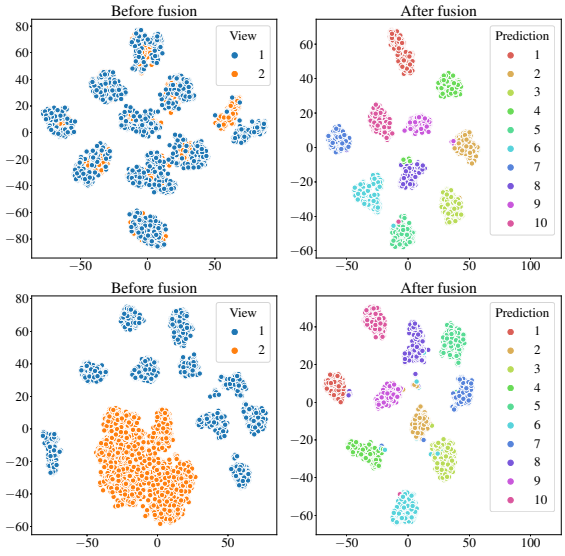


Figure 5: Learned representations before and after fusion for regular (top) and noisy ($\sigma = 1$) E-MNIST (bottom). Projected to 2-D using T-SNE.

we observe that the attention mechanism in EAMC is unable to produce fusion weights that suppress the noisy view. This results in a significant drop in clustering accuracy, as the noise increases.

Selective alignment in CoMVC. Figure 5 demonstrates the selective alignment in CoMVC, for the noise-free and noisy variants of the E-MNIST dataset. In the noise-free case, CoMVC aligns the representations, resulting in clusters that are well separated. When the second view has been corrupted by noise however, it is discarded by the view prioritization mechanism, by setting its fusion weight to 0. This simultaneously disables the alignment procedure, preventing the fused representation from being corrupted by the noisy view, thus preserving the cluster structure.

6. Conclusion

Our work highlights the importance of considering representation alignment when performing multi-view clustering. Comparing the results of our SiMVC to previous results illustrates that naïvely aligning distributions using adversarial learning can prevent the model from learning good clusterings, while CoMVC illustrates the benefit of selective alignment, leveraging the best of both worlds.

Acknowledgements. This work was financially supported by the Research Council of Norway (RCN), through its Centre for Research-based Innovation funding scheme (Visual Intelligence, grant no. 309439), and Consortium Partners. The work was further funded by RCN FRIPRO grant no. 315029, RCN IKTPLUSS grant no. 303514, and the UiT Thematic Initiative "Data-Driven Health Technology"

References

[1] Mahdi Abavisani and Vishal M. Patel. Deep Multimodal Subspace Clustering Networks. *IEEE Journal of Selected Topics in Signal Processing*, 12(6), 2018.

[2] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep Canonical Correlation Analysis. In *International Conference on Machine Learning*, 2013.

[3] Aluizio F. R. Araújo, Victor O. Antonino, and Karina L. Ponce-Guevara. Self-organizing subspace clustering for high-dimensional and multi-view data. *Neural Networks*, 2020.

[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein Generative Adversarial Networks. In *International Conference on Machine Learning*, 2017.

[5] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning Representations by Maximizing Mutual Information Across Views. In *Neural Information Processing Systems*, 2019.

[6] Tadas Baltrusaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal Machine Learning: A Survey and Taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2), 2019.

[7] M. B. Blaschko and C. H. Lampert. Correlational spectral clustering. In *Computer Vision and Pattern Recognition*, 2008.

[8] Xiao Cai, Feiping Nie, and Heng Huang. Multi-View K-Means Clustering on Big Data. In *International Joint Conference on Artificial Intelligence*, 2013.

[9] Cătălina Cangea, Petar Veličković, and Pietro Liò. Xflow: Cross-modal deep neural networks for audiovisual classification. *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[10] Xiaochun Cao, Changqing Zhang, Huazhu Fu, Si Liu, and Hua Zhang. Diversity-induced Multi-view Subspace Clustering. In *Computer Vision and Pattern Recognition*, 2015.

[11] Kamalika Chaudhuri, Sham Kakade, K. Livescu, and Karthik Sridharan. Multi-View Clustering via Canonical Correlation Analysis. In *International Conference on Machine Learning*, 2009.

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *International Conference on Machine Learning*, 2020.

[13] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big Self-Supervised Models are Strong Semi-Supervised Learners. *arXiv:2006.10029 [cs, stat]*, 2020.

[14] Jiafeng Cheng, Qianqian Wang, Quanxue Gao, Deyan Xie, and Zhiqiang Tao. Multi-View Attribute Graph Convolution Networks for Clustering. In *International Joint Conference on Artificial Intelligence*, 2020.

[15] Miaomiao Cheng, Liping Jing, and Michael K. Ng. Tensor-Based Low-Dimensional Representation Learning for Multi-View Clustering. *IEEE Transactions on Image Processing*, 28(5), 2019.

[16] Liang Du, Peng Zhou, Lei Shi, Hanmo Wang, Mingyu Fan, Wenjian Wang, and Yi-Dong Shen. Robust multiple kernel k-means using l21-norm. In *AAAI Conference on Artificial Intelligence*, 2015.

[17] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. In *International Journal of Computer Vision*, 2010.

[18] Mehmet Gönen and Adam A. Margolin. Localized Data Fusion for Kernel k-Means Clustering with Application to Cancer Biology. In *Neural Information Processing Systems*, 2014.

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Neural Information Processing Systems*, 2014.

[20] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised Learning. *arXiv:2006.07733 [cs, stat]*, 2020.

[21] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. Improved Deep Embedded Clustering with Local Structure Preservation. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, Melbourne, Australia, 2017.

[22] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *Computer Vision and Pattern Recognition*, 2020.

[23] Ming Hou, Jiajia Tang, Jianhai Zhang, Wanzeng Kong, and Qibin Zhao. Deep multimodal multilinear fusion with high-order polynomial pooling. In *Advances in Neural Information Processing Systems*, pages 12136–12145, 2019.

[24] Aiping Huang, Tiesong Zhao, and Chia-Wen Lin. Multi-View Data Fusion Oriented Clustering via Nuclear Norm Minimization. *IEEE Transactions on Image Processing*, 29, 2020.

[25] Shudong Huang. Auto-weighted multi-view clustering via deep matrix decomposition. *Pattern Recognition*, 2020.

[26] Shuning Huang, Kaoru Ota, Mianxiong Dong, and Fanzhang Li. MultiSpectralNet: Spectral Clustering Using Deep Neural Network for Multi-View Data. *IEEE Transactions on Computational Social Systems*, 6(4), 2019.

[27] Sung Ju Hwang and Kristen Grauman. Accounting for the relative importance of objects in image retrieval. In *British Machine Vision Conference*, 2010.

[28] Robert Jenssen, Jose C. Principe, Deniz Erdogmus, and Torbjørn Eltoft. The Cauchy–Schwarz divergence and Parzen windowing: Connections to graph theory and Mercer kernels. *Journal of the Franklin Institute*, 343(6), 2006.

[29] Yu-Gang Jiang, Guangnan Ye, Shih-Fu Chang, Daniel Ellis, and Alexander C. Loui. Consumer video understanding: A benchmark database and an evaluation of human and machine performance. In *International Conference on Multimedia Retrieval*, 2011.

[30] Michael Kampffmeyer, Sigurd Løkse, Filippo M. Bianchi, Lorenzo Livi, Arnt-Børre Salberg, and Robert Jenssen. Deep

divergence-based approach to clustering. *Neural Networks*, 113, 2019.

[31] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015.

[32] Chen Kong, Dahua Lin, Mohit Bansal, Raquel Urtasun, and Sanja Fidler. What Are You Talking About? Text-to-Image Coreference. In *Computer Vision and Pattern Recognition*, Columbus, OH, USA, 2014.

[33] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep Learning. *Nature*, 521(7553), 2015.

[34] Jianqiang Li, Guoxu Zhou, Yuning Qiu, Yanjiao Wang, Yu Zhang, and Shengli Xie. Deep graph regularized non-negative matrix factorization for multi-view clustering. *Neurocomputing*, 390, 2020.

[35] Miaomiao Li, Xinwang Liu, Lei Wang, Yong Dou, Jianping Yin, and En Zhu. Multiple kernel clustering with local kernel alignment maximization. In *International Joint Conference on Artificial Intelligence*, 2016.

[36] Zhaoyang Li, Qianqian Wang, Zhiqiang Tao, Quanxue Gao, and Zhaohua Yang. Deep Adversarial Multi-view Clustering Network. In *International Joint Conference on Artificial Intelligence*, 2019.

[37] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust Recovery of Subspace Structures by Low-Rank Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1), 2013.

[38] Ming-Yu Liu and Oncel Tuzel. Coupled Generative Adversarial Networks. In *Neural Information Processing Systems*, 2016.

[39] Xinwang Liu. Multiple Kernel k-Means Clustering with Matrix-Induced Regularization. In *AAAI Conference on Artificial Intelligence*, 2016.

[40] Sindy Löwe, Peter O'Connor, and Bastiaan Veeling. Putting An End to End-to-End: Gradient-Isolated Learning of Representations. In *Neural Information Processing Systems*, 2019.

[41] Shirui Luo, Changqing Zhang, Wei Zhang, and Xiaochun Cao. Consistent and Specific Multi-View Subspace Clustering. In *AAAI Conference on Artificial Intelligence*, 2018.

[42] Sameer A. Nene, Shree K. Nayar, and Hiroshi Murase. Columbia Object Image Library (COIL-20). Techincal Report CUCS-006-96, 1996.

[43] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Ng. Multimodal Deep Learning. In *International Conference on Machine Learning*, 2011.

[44] Feiping Nie, Jing Li, and Xuelong Li. Self-weighted Multiview Clustering with Multiple Graphs. In *International Joint Conference on Artificial Intelligence*, 2017.

[45] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Neural Information Processing Systems*, 2019.

[46] Juan-Manuel Pérez-Rúa, Valentin Vielzeuf, Stéphane Pateux, Moez Baccouche, and Frédéric Jurie. Mfas: Multimodal fusion architecture search. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pages 6966–6975, 2019.

[47] Jianbo Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), 2000.

[48] Zhiqiang Tao, Hongfu Liu, Sheng Li, Zhengming Ding, and Yun Fu. Marginalized Multiview Ensemble Clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 31(2), 2020.

[49] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv:1807.03748 [cs, stat]*, 2019.

[50] Beilei Wang, Yun Xiao, Zhihui Li, Xuanhong Wang, Xiaojiang Chen, and Dingyi Fang. Robust Self-Weighted Multi-View Projection Clustering. In *AAAI Conference on Artificial Intelligence*, 2020.

[51] R. Wang, F. Nie, Z. Wang, H. Hu, and X. Li. Parameter-Free Weighted Multi-View Projected Clustering with Structured Graph Learning. *IEEE Transactions on Knowledge and Data Engineering*, 32(10), 2020.

[52] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On Deep Multi-View Representation Learning. In *International Conference on Machine Learning*, 2015.

[53] Yifan Wu, Ezra Winston, Divyansh Kaushik, and Zachary Lipton. Domain adaptation with asymmetrically-relaxed distribution alignment. In *International Conference on Machine Learning*, pages 6872–6881, 2019.

[54] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv:1708.07747 [cs, stat]*, 2017.

[55] Deyan Xie, Xiangdong Zhang, Quanxue Gao, Jiale Han, Song Xiao, and Xinbo Gao. Multiview Clustering by Joint Latent Representation and Similarity Learning. *IEEE Transactions on Cybernetics*, 2019.

[56] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised Deep Embedding for Clustering Analysis. In *International Conference on Machine Learning*, 2016.

[57] Chang Xu, D. Tao, and Chao Xu. Multi-view Self-Paced Learning for Clustering. In *International Joint Conference on Artificial Intelligence*, 2015.

[58] Zhiyong Yang, Qianqian Xu, Weigang Zhang, Xiaochun Cao, and Qingming Huang. Split Multiplicative Multi-View Subspace Clustering. *IEEE Transactions on Image Processing*, 28(10), 2019.

[59] Ming Yin, Junbin Gao, Shengli Xie, and Yi Guo. Multiview Subspace Clustering via Tensorial t-Product Representation. *IEEE Transactions on Neural Networks and Learning Systems*, 30(3), 2019.

[60] Kun Zhan, Feiping Nie, Jing Wang, and Yi Yang. Multiview Consensus Graph Clustering. *IEEE Transactions on Image Processing*, 28(3), 2019.

[61] Changqing Zhang, Huazhu Fu, Qinghua Hu, Xiaochun Cao, Yuan Xie, Dacheng Tao, and Dong Xu. Generalized Latent Multi-View Subspace Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(1), 2020.

[62] Han Zhao, Remi Tachet des Combes, Kun Zhang, and Geoffrey J. Gordon. On Learning Invariant Representation for Domain Adaptation. *arXiv:1901.09453 [cs, stat]*, 2019.

[63] Handong Zhao, Zhengming Ding, and Yun Fu. Multi-View Clustering via Deep Matrix Factorization. In *AAAI Conference on Artificial Intelligence*, 2017.

[64] Runwu Zhou and Yi-Dong Shen. End-to-End Adversarial-Attention Network for Multi-Modal Clustering. In *Computer Vision and Pattern Recognition*, 2020.

[65] Linlin Zong, Xianchao Zhang, Xinyue Liu, and Hong Yu. Weighted Multi-View Spectral Clustering Based on Spectral Perturbation. In *AAAI Conference on Artificial Intelligence*, 2018.