On the Effects of Self-supervision and Contrastive Alignment in Deep Multi-view Clustering

Daniel J. Trosten[*], Sigurd Løkse[*], Robert Jenssen[*†‡§], Michael C. Kampffmeyer[*†]
Department of Physics and Technology, UiT The Arctic University of Norway
firstname[.middle initial].lastname@uit.no

Abstract

Self-supervised learning is a central component in recent approaches to deep multi-view clustering (MVC). However, we find large variations in the development of self-supervision-based methods for deep MVC, potentially slowing the progress of the field. To address this, we present DeepMVC, a unified framework for deep MVC that includes many recent methods as instances. We leverage our framework to make key observations about the effect of self-supervision, and in particular, drawbacks of aligning representations with contrastive learning. Further, we prove that contrastive alignment can negatively influence cluster separability, and that this effect becomes worse when the number of views increases. Motivated by our findings, we develop several new DeepMVC instances with new forms of self-supervision. We conduct extensive experiments and find that (i) in line with our theoretical findings, contrastive alignments decreases performance on datasets with many views; (ii) all methods benefit from some form of self-supervision; and (iii) our new instances outperform previous methods on several datasets. Based on our results, we suggest several promising directions for future research. To enhance the openness of the field, we provide an open-source implementation of DeepMVC, including recent models and our new instances. Our implementation includes a consistent evaluation protocol, facilitating fair and accurate evaluation of methods and components[1].

1. Introduction

Multi-view clustering (MVC) generalizes the clustering task to data where the instances to be clustered are observed through multiple views, or by multiple modalities. In recent years, deep learning architectures have seen widespread adoption in MVC, resulting in the *deep MVC* subfield. Methods developed within this subfield have shown state-of-the-art clustering performance on several multi-view datasets [14, 19–21, 29, 33], largely outperforming traditional, non-deep-learning-based methods [33].

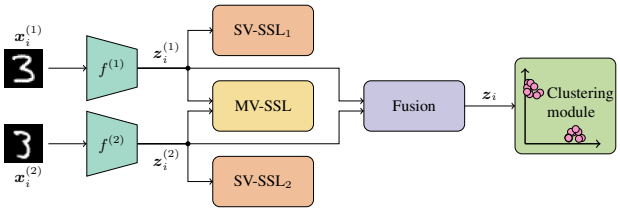

Figure 1. Overview of the DeepMVC framework for a two-view dataset. Different colors denote different components. The framework is generalizable to an arbitrary number of views by adding more view specific encoders (f) and SV-SSL blocks.

Despite these promising developments, we identify significant drawbacks with the current state of the field. Self-supervised learning (SSL) is a crucial component in many recent methods for deep MVC [14, 19–21, 29, 33]. However, the large number of methods, all with unique components and arguments about how they work, makes it challenging to identify clear directions and trends in the development of new components and methods. Methodological research in deep MVC thus lacks foundation and consistent directions for future advancements. This effect is amplified by large variations in implementation and evaluation of new methods. Architectures, data preprocessing and data splits, hyperparameter search strategies, evaluation metrics, and model selection strategies all vary greatly across publications, making it difficult to properly compare methods from different papers. To address these challenges, we present a unified framework for deep MVC, coupled with a rigorous and consistent evaluation protocol, and an open-source implementation. Our main contributions are summarized as follows:

(1) DeepMVC framework. Despite the variations in the development of new methods, we recognize that the majority of recent methods for deep MVC can be decomposed

[*]UiT Machine Learning group (machine-learning.uit.no) and Visual Intelligence Centre (visual-intelligence.no).
[†]Norwegian Computing Center (nr.no).
[‡]Department of Computer Science, University of Copenhagen.
[§]Pioneer Centre for AI (aicentre.dk).
[1]Code: https://github.com/DanielTrosten/DeepMVC

into the following fixed set of components: (i) view-specific encoders; (ii) single-view SSL; (iii) multi-view SSL; (iv) fusion; and (v) clustering module. The DeepMVC framework (Figure 1) is obtained by organizing these components into a unified deep MVC model. Methods from previous work can thus be regarded as *instances* of DeepMVC.

(2) Theoretical insight on alignment and number of views. Contrastive alignment of view-specific representations is an MV-SSL component that has demonstrated state-of-the-art performance in deep MVC [19]. We study a simplified case of deep MVC, and find that contrastive alignment can only decrease the number of separable clusters in the representation space. Furthermore, we show that this potential negative effect of contrastive alignment becomes worse when the number of views in the dataset increases.

(3) New instances of DeepMVC. Inspired by initial findings from the DeepMVC framework, and our theoretical findings on contrastive alignment, we develop 6 new instances of DeepMVC, which outperform current state-of-the-art methods on several multi-view datasets. The new instances include both novel and well-known types of self-supervision, fusion and clustering modules.

(4) Open-source implementation of DeepMVC and evaluation protocol. We provide an open-source implementation of DeepMVC, including several recent methods and our new instances. The implementation includes a shared evaluation protocol for all methods, and all datasets used in the experimental evaluation. By making the datasets and our implementation openly available, we aim to facilitate simpler development of new methods, as well as rigorous and accurate comparisons between methods and components.

(5) Evaluation of methods and components. We use the implementation of DeepMVC to evaluate and compare several recent state-of-the-art methods and SSL components – both against each other, and against our new instances. In our experiments, we both provide a consistent evaluation of methods in deep MVC, and systematically analyze several SSL-based components – revealing how they behave under different experimental settings.

The main findings from our work are:

- We show that aligning view-specific representations can have a negative impact on cluster separability, especially when the number of views becomes large. In our experiments, we find that contrastive alignment of view-specific representations works well for datasets with few views, but *significantly degrades performance when the number of views increases*. Conversely, we find that maximization of mutual information performs well with many views, while not being as strong with fewer views.
- All methods included in our experiments benefit from at least one form of SSL. In addition to contrastive alignment for few views and mutual information maximization for many views, we find that autoencoder-style reconstruction improves overall performance of methods.
- Properties of the datasets, such as class (im)balance and the number of views, heavily impact the performance of current MVC approaches. There is thus not a single "state-of-the-art" – it instead depends on the datasets considered.
- Results reported by the original authors differ significantly from the performance of our re-implementation for some baseline methods, illustrating the necessity of a unified framework with a consistent evaluation protocol.

2. DeepMVC framework

In this section we present the DeepMVC framework, its components and their purpose, and how they fit together. This allows us to, in the next section, summarize recent work on deep MVC, and illustrate that the majority of recent methods can be regarded as instances of DeepMVC.

Suppose we have a multi-view dataset consisting of n instances and V views, and let $\boldsymbol{x}_i^{(v)}$ be the observation of instance i through view v. The task of the DeepMVC framework is then to cluster the instances into k clusters, and produce cluster membership indicators $\alpha_{ic} \in [0, 1]$, $c = 1, \ldots, k$. The framework is illustrated in Figure 1. It consists of the following components.

View-specific encoders. The framework is equipped with V deep neural network encoders $f^{(1)}, \ldots, f^{(V)}$, one for each view. Their task is to produce the view-specific representations $\boldsymbol{z}_i^{(v)} = f^{(v)}(\boldsymbol{x}_i^{(v)})$ from the input data.

Single-view self-supervised learning (SV-SSL). The SV-SSL component consists of a set of pretext tasks (auxiliary objectives) that are designed to aid the optimization of the view-specific encoders. Specifically, the tasks should be designed to help the encoders learn representations that simplify the clustering task. Each pretext task is specific to its designated view, and is isolated from all other views.

Multi-view self-supervised learning (MV-SSL). MV-SSL is similar to SV-SSL – they are both self-supervised modules whose goals are to help the encoders learn representations that are suitable for clustering. However, MV-SSL leverages all views simultaneously in the pretext tasks, allowing the model to exploit information from all views simultaneously to learn better features.

Fusion. This component combines view-specific representations into a shared representation for all views. Fusion is typically done using a (weighted) average [11, 19], or by concatenation [5, 26, 29]. More complex fusion modules using *e.g.* attention mechanisms [33], are also possible.

Clustering module (CM). The CM is responsible for determining cluster memberships based on view-specific or fused representations. The CM can consist of a traditional clustering method, such as k-means [13] or Spectral Clustering [16]. Such CMs are applied to the fused representations after other components have been trained, resulting in a two-stage method that first learns fused representations, and then applies a clustering algorithm to these representations.

Alternatively, the CM can be integrated into the model [19, 33], allowing it to be trained alongside other components, resulting in fused representations that are better suited for clustering.

Loss functions and training. The loss functions for the models are specified by the SV-SSL, MV-SSL, and CM components. To train the model, the terms arising from the different components can be minimized simultaneously or they can be minimized in an alternating fashion. It is also possible with pre-training/fine-tuning setups where the model is pre-trained with one subset of the losses and fine-tuned with another subset of the losses.

We note that DeepMVC is a conceptual framework, and that a model is not necessarily completely described by a list of its DeepMVC components. Consequently, it is possible for two models with similar DeepMVC components to have slightly different implementations. This illustrates the importance of our open-source implementation of DeepMVC, which allows the implementation of a model to be completely transparent.

3. Previous methods as instances of DeepMVC

Table 1 shows selected recent methods for deep MVC (the full table can be found in the supplementary), categorized by its DeepMVC components, allowing for systematic comparisons between models[2].

View-specific encoders. As can be seen in Table 1, all models use view-specific encoders to encode views into view-specific embeddings. Multi-layer perceptrons (MLPs) are usually used for vector data, while convolutional neural networks (CNNs) are used for image data.

SV-SSL and MV-SSL. Alongside the encoder network, many methods use decoders to reconstruct the original views from either the view-specific representations or the fused representation. The reconstruction task is the most common self-supervised pretext task, both for SV-SSL and for MV-SSL. In SV-SSL, the views are reconstructed from their respective view-specific representations, without any influence from the other views [1, 17, 18, 22, 28, 31, 32, 35]. In MV-SSL, it is common to either do (i) cross view reconstruction, where all views are reconstructed from all view-specific representations [34]; or (ii) fused view reconstruction, where all views are reconstructed from the fused representation [11, 20, 30, 34].

Aligning distributions of view-specific representations is another MV-SSL pretext task that has been shown to produce representations suitable for clustering [33]. However, [19] demonstrate that the alignment of representation distributions can be detrimental to the clustering performance – especially in the presence of noisy or non-informative views. To avoid these drawbacks, they propose Simple MVC (SiMVC) and Contrastive MVC (CoMVC). In the former, the alignment is dropped altogether, whereas the latter includes a contrastive learning module that aligns the view-specific representations at the instance level, rather than at the distribution level.

Clustering modules. Many deep MVC methods use subspace-based clustering modules [1, 17, 21, 34]. These methods assume that representations, either view-specific or fused, can be decomposed into linear combinations of each other. Once determined, the self-representation matrix containing the coefficients for these linear combinations is used to compute an affinity matrix, which in turn is used as input to spectral clustering. This requires the full $n \times n$ self-representation matrix available in memory, which is computationally prohibitive for datasets with a large number of instances.

Other clustering modules have also been adapted to deep MVC. The clustering module from Deep Embedded Clustering (DEC) [25], for instance, is used in several models [2, 11, 20, 26, 28]. Recently, the Deep Divergence-Based Clustering (DDC) [7] clustering module has been used in several state-of-the-art deep MVC models [19, 33]. In addition, some methods treat either the encoder output or the fused representation as cluster membership vectors [14, 31].

Lastly, some methods adopt a two-stage approach, where they first use the SSL components to learn representations, and then apply a traditional clustering method, such as k-means [4, 5, 29, 32, 35], a Gaussian mixture model [30], or spectral clustering [22], on the trained representations.

4. Contrastive alignment in deep MVC

As can be seen in Table 1, SSL components are crucial in recent state-of-the-art methods for deep MVC. Recent works have focused on aligning view-specific representations [19, 33], and in particular, contrastive alignment [19]. We study a simplified setting where, for each view, all observations in a cluster are located at the same point. This allows us to prove that aligning view-specific representations has a negative impact on the cluster separability after fusion. This is the same starting point as in [19], but we extend the analysis to investigate contrastive alignment when the number of views increases.

Proposition 1 (Adapted from [19])**.** *Suppose the dataset consists of n instances, V views, and k ground-truth clusters, and that view-specific representations are computed with view-specific encoders as $\boldsymbol{z}_i^{(v)} = f^{(v)}(\boldsymbol{x}_i^{(v)})$. Furthermore, assume that:*

- *For all $v \in \{1, \dots V\}$ and $j \in \{1, \dots, k\}$,*

$$\forall i \in \mathcal{C}_j, \boldsymbol{x}_i^{(v)} = \boldsymbol{c}^{(v)} \in \{\boldsymbol{c}_1^{(v)}, \dots, \boldsymbol{c}_{k_v}^{(v)}\} \quad (1)$$

[2]Note, here we limit our discussion to MVC approaches without missing data. While most of the theoretical and empirical results also generalize to the emerging incomplete MVC setting [12, 23, 27], we consider it out of scope of this work.

Model	Pub.	Enc.	SV-SSL	MV-SSL	Fusion	CM
DCCAE [22]	ICML'15	MLP	Reconstruction	CCA	1st view	SC
DMSC [1]	J. STSP'18	CNN	Reconstruction	–	Affinity fusion	SR, SC
MvSCN [5]	IJCAI'19	MLP	Sp. Emb.	MSE Al.	Concat.	k-means
DAMC [11]	IJCAI'19	MLP	–	Reconstruction	Average	DEC
SGLR-MVC [30]	AAAI'20	MLP	Variational Reconstruction	Variational Reconstruction	Weighted sum	GMM
EAMC [33]	CVPR'20	MLP	–	Distribution Al., Kernel Al.	Attention	DDC
SiMVC [19]	CVPR'21	MLP/CNN	–	–	Weighted sum	DDC
CoMVC [19]	CVPR'21	MLP/CNN	–	Contrastive Al.	Weighted sum	DDC
Multi-VAE [29]	ICCV'21	CNN	–	Variational Reconstruction	Concat.	Gumbel, k-means
DMIM [14]	IJCAI'21	MLP	Min. superflous information	Max. shared information	?	Encoder output
Model	**Category**	**Enc.**	**SV-SSL**	**MV-SSL**	**Fusion**	**CM**
AE–KM	Simple	MLP/CNN	Reconstruction	–	Concat.	k-means
AE–DDC	Simple	MLP/CNN	Reconstruction	–	Weighted sum	DDC
AECoKM	Contrastive Al.	MLP/CNN	Reconstruction	Contrastive Al.	Concat.	k-means
AECoDDC	Contrastive Al.	MLP/CNN	Reconstruction	Contrastive Al.	Weighted sum	DDC
InfoDDC	Mutual info.	MLP/CNN	–	Max. mutual info.	Weighted sum	DDC
MV-IIC	Mutual info.	MLP/CNN	–	IIC Overclustering	–	IIC, k-means

Table 1. Overview of selected methods from previous work (top) and proposed new instances (bottom), and their DeepMVC components. The complete table of previous methods is included in the supplementary. **Abbreviations:** "–" = Not included, "?" = Not specified, Al. = Alignment, Concat. = Concatenate, CCA = Canonical correlation analysis, DDC = Deep divergence-based clustering, DEC = Deep embedded clustering, SC = Spectral clustering, Sp. Emb. = Spectral Embedding, SR = Self-representation

where $\mathcal{C}_j$ is the set of indices for instances in cluster j, and $k_v \in \{1, \dots, k\}$ is the number of separable clusters in view v.

- *Representations are fused as $\boldsymbol{z}_i = \sum_{v=1}^{V} w_v \boldsymbol{z}_i^{(v)}$ where $w_1, \dots, w_V$ are all unique.*

- *For all $j \in \{1, \dots, k\}$,*

$$\forall i \in \mathcal{C}_j, \boldsymbol{z}_i = \boldsymbol{z}^\star \in \{\boldsymbol{z}_1^\star, \dots, \boldsymbol{z}_\kappa^\star\} \tag{2}$$

Then if $\boldsymbol{z}_i^{(1)} = \cdots = \boldsymbol{z}_i^{(V)}$ (perfectly aligned view-specific representations),

$$\kappa = \min\{k, (\min_{v=1,\dots,V} \{k_v\})^V\} \tag{3}$$

Proof. See [19]. □

According to Proposition 1, when the view-specific representations are perfectly aligned, the number of separable clusters after fusion, κ, depends on the number of separable clusters in the *least informative view* – the view with the lowest k_v. The following propositions show what happens to $\min\{k_v\}$ when the number of views increases[3].

Proposition 2. *Suppose $k_v, v \in \mathbb{N}$ are random variables taking values in $\{1, \dots, k\}$. Then, for any $V \geq 1$,*

$$\mathbb{P}\{\min_{v=1,\dots,V+1} \{k_v\} \leq \min_{v=1,\dots,V} \{k_v\} \mid k_1, \dots, k_V\} = 1 \tag{4}$$

[3]The proofs of Propositions 2 and 3 are given in the supplementary

Proposition 3. *Suppose $k_v, v \in \mathbb{N}$ are iid. random variables taking values in $\{1, \dots, k\}$. Then, for any $V \geq 1$,*

$$\mathbb{E}(\min_{v=1,\dots,V+1} \{k_v\}) \leq \mathbb{E}(\min_{v=1,\dots,V} \{k_v\}) \tag{5}$$

Assuming the view-specific representations are perfectly aligned, Propositions 2 and 3 show that: (i) Given a number of views, adding another view will, with probability 1, not increase $\min\{k_v\}$. (ii) Among two datasets with the same distribution for the k_v, the dataset with the *smallest number of views* will have the highest expected value of $\min\{k_v\}$.

In summary, we have shown that contrastive alignment-based models perform worse when the number of views in a dataset increases. These findings are supported by the experimental results in Figure 2 and Table 2 which show that, when the number of views increases, the contrastive alignment-based model is outperformed by the model without any alignment.

Alignment as a pretext task. In contrast to our theoretical findings in the simplified case, Figure 2 and Table 2 show that contrastive alignment can sometimes be beneficial for the performance, particularly when the number of views is small. This is because alignment might be a good pretext task that helps the encoders learn informative representations, by learning to represent the information that is shared across views. However, we emphasize that this is only true when the number of views is small (≤ 4 in Figure 2), meaning that alignment should be used with caution when the number of views increases beyond this point.

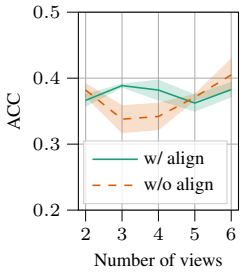

Figure 2. Clustering accuracy for an increasing number of views on Caltech7.

Dataset	w/o align	w/ align
Edge-MNIST (2 views)	0.89	**0.97**
Caltech7 (6 views)	**0.41**	0.38
Patched-MNIST (12 views)	**0.84**	0.73

Table 2. Clustering accuracies on datasets with varying number of views.

5. New instances of DeepMVC

With our new instances of DeepMVC, we aim to further analyze and address the many-views-issue with contrastive alignment highlighted above, as well as to investigate the effect of other SSL components. In addition to alignment of view-specific representations [2, 19, 33], we identify reconstruction [1, 11, 22] and mutual information maximization [6, 21] to be promising directions for the new instances. Maximizing mutual information is particularly interesting, as it enables the view-specific encoders to represent the information which is shared across views, without explicitly forcing the view-specific representations to be aligned. Furthermore, we recognize that simple baselines with few or no SSL components – exemplified by SiMVC [19] – might perform similarly to more complicated methods, while being significantly easier to implement and faster to train. It is therefore crucial to include such methods in an experimental evaluation, in order to properly determine whether additional SSL-based components are beneficial for the models' performance. Finally, our overview of recent work shows that both traditional clustering modules (*e.g.* k-means) and deep learning-based clustering modules (*e.g.* DDC) are commonly used in deep MVC.

In total, we develop 6 new DeepMVC instances in 3 categories. The new instances are summarized in Table 1. Evaluating these instances and several methods from recent work, allows us to accurately evaluate methods and components, and investigate how they behave for datasets with varying characteristics.

Simple baselines: **AE–KM** has view-specific autoencoders (AEs) with a mean-squared-error (MSE) loss

$$\mathcal{L}^{\text{SV}}_{\text{Reconstruction}} = \frac{1}{nV}\sum_{i=1}^{n}\sum_{v=1}^{V}||\boldsymbol{x}_i^{(v)} - \hat{\boldsymbol{x}}_i^{(v)}||^2 \tag{6}$$

as its SV-SSL task. The views are fused by concatenation and the concatenated representations are clustered using k-means after the view-specific autoencoders have been trained. **AE–DDC** uses view-specific autoencoders with an MSE loss (Eq. (6)) as its SV-SSL task. The views are fused using a weighted sum and the fused representations are clustered using the DDC clustering module [7].

Contrastive alignment-based: **AECoKM** extends AE–KM with a contrastive loss on the view-specific representations. We use the multi-view generalization of the NT-Xent (contrastive) loss by Trosten *et al*. [19], without the "other clusters" negative sampling

$$\mathcal{L}^{\text{MV}}_{\text{Contrastive}} = \frac{1}{nV(V-1)}\sum_{i=1}^{n}\sum_{v=1}^{V}\sum_{u=1}^{V}\mathbb{1}_{\{u\neq v\}}\,\ell_i^{(uv)}, \tag{7}$$

$$\ell_i^{(uv)} = -\log\frac{\exp(s_{ii}^{(uv)})}{\sum_{s'\in\text{Neg}(\boldsymbol{z}_i^{(u)},\boldsymbol{z}_i^{(v)})}\exp(s')} \tag{8}$$

and $s_{ij}^{(uv)} = \frac{1}{\tau}\frac{\boldsymbol{z}_i^u\cdot\boldsymbol{z}_j^{(v)}}{||\boldsymbol{z}_i^u||\cdot||\boldsymbol{z}_j^{(v)}||}$ denotes the cosine similarity between $\boldsymbol{z}_i^u$ and $\boldsymbol{z}_j^v$. The set $\text{Neg}(\boldsymbol{z}_i^{(u)}, \boldsymbol{z}_i^{(v)})$ is the set of similarities of negative pairs for the positive pair $(\boldsymbol{z}_i^{(u)}, \boldsymbol{z}_i^{(v)})$, which consists of $s_{ij}^{(uv)}$, $s_{ij}^{(uu)}$, and $s_{ij}^{(vv)}$, for all $j \neq i$. τ is a hyperparameter, which we set to 0.1 for all experiments. **AECoDDC** extends AE–DDC using the same generalized NT-Xent contrastive loss on the view-specific representations.

Mutual information-based: **InfoDDC** maximizes the mutual information (MI) between the view-specific representations, using the MI loss from Invariant Information Clustering (IIC) [6][4]. The MI maximization is regularized by also maximizing the entropy of view-specific representations. The view-specific representations are fused using a weighted sum, and the fused representations are clustered using DDC [7]. **MV-IIC** is a multi-view generalization of IIC [6], where cluster assignments are computed for each of the view-specific representations. The MI between pairs of these view-specific cluster assignments is then maximized using the information maximization loss from IIC. In order to get a final shared cluster assignment for all views, the view-specific cluster assignments are concatenated and clustered using k-means. As in IIC, this model includes 5 over-clustering heads as its MV-SSL task. In both InfoDDC and MV-IIC, we generalize the loss from IIC to an arbitrary number of views:

$$\mathcal{L}^{\text{MV}}_{\text{MI}} = \frac{2}{V(V-1)}\sum_{u=1}^{V-1}\sum_{v=u+1}^{V} -\Big(\underbrace{I(\boldsymbol{Z}^{(u)},\boldsymbol{Z}^{(v)})}_{\text{mutual information}} + (\lambda-1)\underbrace{(H(\boldsymbol{Z}^{(u)})+H(\boldsymbol{Z}^{(v)}))}_{\text{entropy regularization}}\Big) \tag{9}$$

[4]The supplementary includes a brief overview of the connection between InfoDDC and contrastive alignment.

where the summands are computed as

$$I(\boldsymbol{Z}^{(u)}, \boldsymbol{Z}^{(v)}) + (\lambda - 1)(H(\boldsymbol{Z}^{(u)}) + H(\boldsymbol{Z}^{(v)})) = -\sum_{a=1}^{D}\sum_{b=1}^{D} \boldsymbol{P}_{ab}^{(uv)} \log \frac{\boldsymbol{P}_{ab}^{(uv)}}{(\boldsymbol{P}_{a}^{(u)}\boldsymbol{P}_{b}^{(v)})^{\lambda}}, \quad (10)$$

where D denotes the dimensionality of the view-specific representations. λ is a hyperparameter that controls the strength of the entropy regularization. We set $\lambda = 10$ for InfoDDC, and $\lambda = 1.5$ for MV-IIC. The joint distribution $\boldsymbol{P}^{(uv)}$ is estimated by first computing $\tilde{\boldsymbol{P}}^{(uv)} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{z}_i^{(u)}(\boldsymbol{z}_i^{(v)})^{\top}$, and then symmetrizing it $\boldsymbol{P}^{(uv)} = \frac{1}{2}(\tilde{\boldsymbol{P}}^{(uv)} + (\tilde{\boldsymbol{P}}^{(uv)})^{\top})$. We assume that each view-specific representation is normalized such that its elements sum to one, and are all non-negative. The marginals $\boldsymbol{P}^{(u)}$ and $\boldsymbol{P}^{(v)}$ are obtained by summing over the rows and columns of $\boldsymbol{P}^{(uv)}$, respectively.

6. Experiments

In this section we provide a rigorous evaluation of methods and their DeepMVC components. Inspired by the initial findings in Section 4 and our overview of recent methods in Section 3, we focus mainly on the SSL and CM components in our evaluation. We found these components to be most influential on the methods' performance. For completeness, we include experiments with different fusion and CM components in the supplementary.

6.1. Setup

Baselines. In addition to the new instances presented in Section 5, we include 6 baseline models from previous work in our experiments. The following baseline models were selected to include a diverse set of framework components in the evaluation: (i) Deep Multimodal Subspace Clustering (DMSC) [1]; (ii) Multi-view Spectral Clustering Network (MvSCN) [5]; (iii) End-to-end Adversarial-attention Multimodal Clustering (EAMC) [33]; (iv) Simple Multi-View Clustering (SiMVC) [19]; (v) Contrastive Multi-View Clustering (CoMVC) [19]; (vi) Multi-view Variational Autoencoder (Multi-VAE) [29].

As can be seen in Table 1, this collection of models includes both reconstruction-based and alignment-based SSL, as well as traditional (k-means and spectral) and deep learning-based CMs. They also include several fusion strategies and encoder networks. Section 6.3 includes an ablation study that examines the influence of SSL components in these models.

Datasets. We evaluate the baselines and new instances on 8 widely used benchmark datasets for deep MVC. We prioritize datasets that were also used in the original publications for the selected baselines. Not only does this result in a diverse collection of datasets common in deep MVC – it also allows us to compare the performance of our implementations to what was reported by the original authors. The results of this comparison are given in the supplementary.

The following datasets are used for evaluation: (i) **NoisyMNIST / NoisyFashion**: A version of MNIST [9] / FashionMNIST [24] where the first view contains the original image, and the second view contains an image sampled from the same class as the first image, with added Gaussian noise ($\sigma = 0.2$). (ii) **EdgeMNIST / EdgeFashion**: Another version of MNIST / FashionMNIST where the first view contains the original image, and the second view contains an edge-detected version of the same image. (iii) **COIL-20**: The original COIL-20 [15] dataset, where we randomly group the images of each object into groups of size 3, resulting in a 3-view dataset. (iv) **Caltech7 / Caltech20**: A subset of the Caltech101 [3] dataset including 7 / 20 classes. We use the 6 different features extracted by Li *et al*. [10], resulting in a 6-view dataset[5]. (v) **PatchedMNIST**: A subset of MNIST containing the first three digits, where views are extracted as 7×7 non-overlapping patches of the original image. The corner patches are dropped as they often contain little information about the digit, resulting in a dataset with 12 views. Each patch is resized to 28×28.

All views are individually normalized so that the values lie in $[0, 1]$. Following recent work on deep MVC, we train and evaluate on the full datasets [14, 19, 29, 33]. More dataset details are provided in the supplementary.

Hyperparameters. The baselines use the hyperparameters reported by the original authors, because (i) it is not feasible for us to tune hyperparameters individually for each model on each dataset; and (ii) it is difficult to tune hyperparameters in a realistic clustering setting due to the lack of labeled validation data. For each method, the same hyperparameter configuration is used for all datasets.

New instances use the same hyperparameters as for the baselines wherever possible[6]. Otherwise, we set hyperparameters such that loss terms have the same order of magnitude, and such that the training converges. We refrain from any hyperparameter tuning that includes the dataset labels to keep the evaluation fair and unsupervised. We include a hyperparameter sweep in the supplementary, in order to assess the new instances' sensitivity to changes in their hyperparameter. However, we emphasize that the results of this sweep were *not* used to select hyperparameters for the new instances. All models use the same encoder architectures and are trained for 100 epochs with the Adam optimizer [8].

Evaluation protocol. We train each model from 5 different initializations. Then we select the run that resulted in the lowest value of the loss and report the performance metrics from that run, following [7, 19]. This evaluation protocol is both fully unsupervised, and is not as impacted by poorly performing runs, as for instance the mean performance of all

[5]The list of classes and feature types is included in the supplementary.

[6]Hyperparameters for all models are listed in the supplementary.

(a) **All datasets**			(b) **Random pairings**			(c) **Many views**				(d) **Balanced vs. imbalanced**			
Model	BL	$\bar{Z}$	Model	CA	$\bar{Z}$	Model	MI	CA	$\bar{Z}$	Model	DDC	$\bar{Z}_{\text{bal}}$	$\bar{Z}_{\text{imb}}$
MvSCN	✘	−2.23	MvSCN	✘	−2.49	MvSCN	✘	✘	−1.78	MvSCN	✘	−2.41	−1.78
AECoKM	✘	−0.32	DMSC	✘	−0.54	AECoKM	✘	✓	−0.83	DMSC	✘	−0.39	0.45
EAMC	✘	−0.22	InfoDDC	✘	−0.41	EAMC	✘	✘	−0.75	InfoDDC	✓	−0.13	1.18
DMSC	✘	−0.11	EAMC	✘	−0.17	AE–DDC	✘	✘	−0.36	EAMC	✓	0.00	−0.75
AE–KM	✓	0.16	MV-IIC	✘	0.05	CoMVC	✘	✓	−0.33	MV-IIC	✘	0.01	1.06
InfoDDC	✘	0.20	AE–KM	✘	0.11	SiMVC	✘	✘	−0.12	AE–KM	✘	0.03	0.56
AE–DDC	✓	0.26	Multi-VAE	✘	0.32	AE–KM	✘	✘	0.23	AECoKM	✘	0.08	−1.54
SiMVC	✓	0.27	SiMVC	✘	0.35	AECoDDC	✘	✓	0.28	CoMVC	✓	0.30	0.25
MV-IIC	✘	0.27	AE–DDC	✘	0.56	Multi-VAE	✘	✘	0.38	SiMVC	✓	0.31	0.16
CoMVC	✘	0.29	AECoKM	✓	0.59	DMSC	✘	✘	0.45	AE–DDC	✓	0.33	0.06
Multi-VAE	✘	0.43	CoMVC	✓	0.63	MV-IIC	✓	✘	0.98	Multi-VAE	✘	0.42	0.47
AECoDDC	✘	0.65	AECoDDC	✓	0.82	InfoDDC	✓	✘	1.15	AECoDDC	✓	0.92	−0.13

Table 3. Aggregated evaluation results for the dataset groups. Models are sorted from lowest to highest by average Z-score for each group. Higher Z-scores indicate better clusterings. Our new instances are underlined. **Abbreviations:** BL = Simple baseline, CA = Contrastive alignment, DDC = Deep divergence-based clustering MI = Mutual information, $\bar{Z}$ = Average Z-score for group.

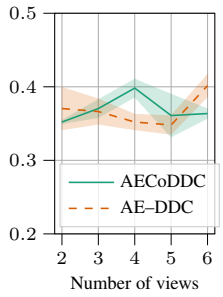

Figure 3. Accuracies on Caltech7 with increasing number of views.

runs. The uncertainty of the performance metric under this model selection protocol is estimated using bootstrapping[7]. We measure clustering performance with the accuracy (ACC) and normalized mutual information (NMI). Both metrics are bounded in $[0, 1]$, and higher values correspond to better performing models, with respect to the ground truth labels.

6.2. Evaluation results

To emphasize the findings from our experiments, we compute the average Z-score for each model, for 4 groups of datasets[8]. Z-scores are calculated by subtracting the mean and dividing by the standard deviation of results, per dataset and per metric. Table 3 shows Z-scores for the groups: (i) **All datasets**. (ii) **Random pairings:** Datasets generated by randomly pairing within-class instances to synthesize multiple views (NoisyMNIST, NoisyFashion, COIL-20). (iii) **Many views:** Datasets with many views (Caltech7, Caltech20, PatchedMNIST). (iv) **Balanced vs. imbalanced:** Datasets with balanced classes (NoisyMNIST, NoisyFashion, EdgeMNIST, EdgeFashion, COIL-20, PatchedMNIST) vs. datasets with imbalanced classes (Caltech7, Caltech20). Our main experimental findings are:

Dataset properties significantly impact the performance of methods. We observe that the ranking of methods varies significantly based on dataset properties, such as the number of views (Table 3c) and class (im)balance (Table 3d). Hence, there is not a single "state-of-the-art" for all datasets.

Our new instances outperform previous methods. In Table 3a we see that the simple baselines perform remarkably well, when compared to the other, more complex methods. This highlights the importance of including simple baselines like these in the evaluation. Table 3a shows that AECoDDC overall outperforms the other methods, and on datasets with many views (Table 3c) we find that InfoDDC and MV-IIC outperform the others by a large margin.

[7]Details on uncertainty computations are included in the supplementary.
[8]Results for all methods/datasets are included in the supplementary.

Maximization of mutual information outperforms contrastive alignment on datasets with many views. Contrastive alignment-based methods show good overall performance, but they struggle when the number of views becomes large (Table 3c). This holds for both baseline methods (as observed in Section 4), and the new instances. As in Section 4, we hypothesize that this is due to issues with representation alignment, where the presence of less informative views is more likely when the number of views becomes large. Contrastive alignment attempts to align view-specific representations to this less informative view, resulting in clusters that are harder to separate in the representation space. This is further verified in Figures 2 and 3, illustrating a decrease in performance on Caltech7 for contrastive alignment-based models with 5 or 6 views. Models based on maximization of mutual information do not have the same problem. We hypothesize that this is because maximizing mutual information still allows the view-specific representations to be different, avoiding the above issues with alignment. The MI-based models also include regularization terms that maximize the entropy of view-specific representations, preventing the representations from collapsing to a single value.

Contrastive alignment works particularly well on datasets consisting of random pairings (Table 3b). In these datasets, the class label is the only thing the views have in common. Contrastive alignment, *i.e.* learning a shared representation for all pairs within a class, thus asymptotically amounts to learning a unique representation for each class, making it easier for the CM to separate between classes.

The DDC CM performs better than the other CMs on balanced datasets. With the DDC CM, the models are end-to-end trainable – jointly optimizing all components in the model. The view-specific representations can thus be adapted to suit the CM, potentially improving the clustering result. DDC also has an inherent bias towards balanced clusters [7], which helps produce better clusterings when the ground truth classes are balanced.

Model	NoisyMNIST w/o SV-SSL	NoisyMNIST w/ SV-SSL	Caltech7 w/o SV-SSL	Caltech7 w/ SV-SSL
DMSC	0.54	0.66 (+0.12)	0.35	0.50 (+0.15)
AE–DDC	1.00	1.00 (0.00)	0.41	0.40 (0.00)
AE–KM	0.67	0.74 (+0.07)	0.39	0.44 (+0.05)
AECoDDC	1.00	1.00 (0.00)	0.38	0.36 (−0.02)
AECoKM	0.56	1.00 (+0.44)	0.22	0.20 (−0.02)
Model	**w/o MV-SSL**	**w/ MV-SSL**	**w/o MV-SSL**	**w/ MV-SSL**
EAMC	1.00	0.83 (−0.17)	0.36	0.44 (+0.08)
Multi-VAE	0.52	0.98 (+0.46)	0.31	0.47 (+0.15)
CoMVC	1.00	1.00 (0.00)	0.41	0.38 (−0.02)
AECoDDC	1.00	1.00 (0.00)	0.40	0.36 (−0.04)
AECoKM	0.74	1.00 (+0.26)	0.44	0.20 (−0.24)
InfoDDC	1.00	0.90 (−0.10)	0.41	0.51 (+0.10)
MV-IIC	0.52	0.52 (0.00)	0.53	0.53 (0.00)

Table 4. Accuracies from ablation studies with SSL components.

Reproducibility of original results. During our experiments we encountered issues with reproducibility with several of the methods from previous work. In the supplementary we include a comparison between our results and those reported by the original authors of the methods from previous work. We find that most methods use different network architectures and evaluation protocols in the original publications, making it difficult to accurately compare performance between methods and their implementations. This illustrates the difficulty of reproducing and comparing results in deep MVC, highlighting the need for a unified framework with a consistent evaluation protocol and an open-source implementation.

6.3. Effect of SSL components

Table 4 shows the results of ablation studies with the SV-SSL and MV-SSL components. These results show that having at least one form of SSL is beneficial for the performance of all models, with the exception being AE–DDC/AECoDDC, which on Caltech7 performs best without any self-supervision. We suspect that this particular result is due to the issues with many views and class imbalance discussed in Section 6.2. Further, we observe that having both forms of SSL is not always necessary. For instance is there no difference with and without SV-SSL for AECoDDC and AECoKM, both of which include contrastive alignment-based MV-SSL. Lastly, we note that contrastive alignment-based MV-SSL decreases performance on Caltech7 for most models. This is consistent with our theoretical findings in Section 4, as well as the results in Section 6.2 and in Figures 2 and 3 – illustrating that contrastive alignment is not suitable for datasets with a large number of views.

7. Conclusion

We investigate the role of self-supervised learning (SSL) in deep MVC. Due to its recent success, we focus particularly on contrastive alignment, and prove that it can be detrimental to the clustering performance, especially when the number of views becomes large. To properly evaluate models and components, we develop DeepMVC – a new unified framework for deep MVC, including the majority of recent methods as instances. By leveraging the new insight from our framework and theoretical findings, we develop 6 new DeepMVC instances with several promising forms of SSL, which perform remarkably well compared to previous methods. We conduct a thorough experimental evaluation of our new instances, previous methods, and their DeepMVC components – and find that SSL is a crucial component in state-of-the-art methods for deep MVC. In line with our theoretical analysis, we observe that contrastive alignment worsens performance when the number of views becomes large. Further, we find that performance of methods depends heavily on dataset characteristics, such as number of views, and class imbalance. Developing methods that are robust towards changes in these properties can thus result in methods that perform well over a wide range of multi-view clustering problems. To this end, we make the following recommendations for future work in deep MVC:

Improving contrastive alignment or maximization of mutual information to handle both few and many views. Addressing pitfalls of alignment to improve contrastive alignment-based methods on many views, is a promising direction for future research. Similarly, we believe that improving the methods based on maximization of mutual information on few views, will result in better models.

Developing end-to-end trainable clustering modules that are not biased towards balanced clusters. The performance of the DDC clustering module illustrates the potential of end-to-end trainable clustering modules, which are capable of adapting the representations to produce better clusterings. Mitigating the bias towards balanced clusters thus has the potential to produce models that perform well, both on balanced and imbalanced datasets.

Proper evaluation and open-source implementations. Finally, we emphasize the importance of evaluating new methods on a representative collection of datasets, *e.g.* many views and few views, paired, imbalanced, *etc*. Also, in the reproducibility study (see supplementary), we find that original results can be difficult to reproduce. We therefore encourage others to use the open-source implementation of DeepMVC, as open code and datasets, and consistent evaluation protocols, are crucial to properly evaluate models and facilitate further development of new methods and components.

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. It 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):1601–1614, 2018. 3, 4, 5, 6

[2] Guowang Du, Lihua Zhou, Yudi Yang, Kevin Lü, and Lizhen Wang. Deep Multiple Auto-Encoder-Based Multi-view Clustering. *Data Science and Engineering*, 6(3):323–338, 2021. 3, 5

[3] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007. 6

[4] 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):749–760, 2019. 3

[5] Zhenyu Huang, Joey Tianyi Zhou, Xi Peng, Changqing Zhang, Hongyuan Zhu, and Jiancheng Lv. Multi-view Spectral Clustering Network. In *IJCAI*, 2019. 2, 3, 4, 6

[6] Xu Ji, Andrea Vedaldi, and Joao Henriques. Invariant Information Clustering for Unsupervised Image Classification and Segmentation. In *ICCV*, 2019. 5

[7] Michael Kampffmeyer, Sigurd Løkse, Filippo M. Bianchi, Lorenzo Livi, Arnt-Børre Salberg, and Robert Jenssen. Deep Divergence-based Approach to Clustering. *Neural Networks*, 113:91–101, 2019. 3, 5, 6, 7

[8] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015. 6

[9] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 6

[10] Yeqing Li, Feiping Nie, Heng Huang, and Junzhou Huang. Large-Scale Multi-View Spectral Clustering via Bipartite Graph. In *AAAI*, 2015. 6

[11] Z. Li, Q. Wang, Z. Tao, Q. Gao, and Z. Yang. Deep Adversarial Multi-view Clustering Network. In *IJCAI*, 2019. 2, 3, 4, 5

[12] Yijie Lin, Yuanbiao Gou, Zitao Liu, Boyun Li, Jiancheng Lv, and Xi Peng. COMPLETER: Incomplete Multi-View Clustering via Contrastive Prediction. In *CVPR*, 2021. 3

[13] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, 1967. 2

[14] Yiqiao Mao, Xiaoqiang Yan, Qiang Guo, and Yangdong Ye. Deep Mutual Information Maximin for Cross-Modal Clustering. In *IJCAI*, 2021. 1, 3, 4, 6

[15] S. A. Nene, S. K. Nayar, and H. Murase. Columbia Object Image Library (COIL-20). Technical report, 1996. 6

[16] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 2

[17] Xiukun Sun, Miaomiao Cheng, Chen Min, and Liping Jing. Self-Supervised Deep Multi-View Subspace Clustering. In *ACML*, 2019. 3

[18] Xiaoliang Tang, Xuan Tang, Wanli Wang, Li Fang, and Xian Wei. Deep Multi-view Sparse Subspace Clustering. In *ICNCC*, 2018. 3

[19] Daniel J. Trosten, Sigurd Løkse, Robert Jenssen, and Michael Kampffmeyer. Reconsidering Representation Alignment for Multi-view Clustering. In *CVPR*, 2021. 1, 2, 3, 4, 5, 6

[20] Qianqian Wang, Zhiqiang Tao, Wei Xia, Quanxue Gao, Xiaochun Cao, and Licheng Jiao. Adversarial Multiview Clustering Networks With Adaptive Fusion. *IEEE transactions on neural networks and learning systems*, PP, 2022. 3

[21] Shiye Wang, Changsheng Li, Yanming Li, Ye Yuan, and Guoren Wang. Self-Supervised Information Bottleneck for Deep Multi-View Subspace Clustering. *arXiv:2204.12496 [cs]*, 2022. 1, 3, 5

[22] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On Deep Multi-View Representation Learning. In *ICML*, 2015. 3, 4, 5

[23] Jie Wen, Zheng Zhang, Guo-Sen Xie, Lunke Fei, Bob Zhang, and Yong Xu. CDIMC-net: Cognitive Deep Incomplete Multi-view Clustering Network. In *IJCAI*, 2020. 3

[24] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv:1708.07747 [cs, stat]*, 2017. 6

[25] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised Deep Embedding for Clustering Analysis. In *ICML*, 2016. 3

[26] Bowen Xin, Shan Zeng, and Xiuying Wang. Self-Supervised Deep Correlational Multi-View Clustering. In *IJCNN*, 2021. 2, 3

[27] Cai Xu, Ziyu Guan, Wei Zhao, Hongchang Wu, Yunfei Niu, and Beilei Ling. Adversarial Incomplete Multi-view Clustering. In *IJCAI*, 2019. 3

[28] Jie Xu, Yazhou Ren, Guofeng Li, Lili Pan, Ce Zhu, and Zenglin Xu. Deep embedded multi-view clustering with collaborative training. *Information Sciences*, 573:279–290, 2021. 3

[29] Jie Xu, Yazhou Ren, Huayi Tang, Xiaorong Pu, Xiaofeng Zhu, Ming Zeng, and Lifang He. Multi-VAE: Learning Disentangled View-Common and View-Peculiar Visual Representations for Multi-View Clustering. In *ICCV*, 2021. 1, 2, 3, 4, 6

[30] Ming Yin, Weitian Huang, and Junbin Gao. Shared

Generative Latent Representation Learning for Multi-View Clustering. In *AAAI*, 2020. 3, 4

[31] Xianchao Zhang, Jie Mu, Linlin Zong, and Xiaochun Yang. End-To-End Deep Multimodal Clustering. In *ICME*, 2020. 3

[32] Xianchao Zhang, Xiaorui Tang, Linlin Zong, Xinyue Liu, and Jie Mu. Deep Multimodal Clustering with Cross Reconstruction. In *PAKDD*, 2020. 3

[33] Runwu Zhou and Yi-Dong Shen. End-to-End Adversarial-Attention Network for Multi-Modal Clustering. In *CVPR*, 2020. 1, 2, 3, 4, 5, 6

[34] Pengfei Zhu, Binyuan Hui, Changqing Zhang, Dawei Du, Longyin Wen, and Qinghua Hu. Multi-view Deep Subspace Clustering Networks. *arXiv:1908.01978 [cs]*, 2019. 3

[35] Linlin Zong, Faqiang Miao, Xianchao Zhang, and Bo Xu. Multimodal Clustering via Deep Commonness and Uniqueness Mining. In *ICIKM*, 2020. 3