Partially View-aligned Representation Learning with Noise-robust Contrastive Loss

Mouxing Yang[1], Yunfan Li[1], Zhenyu Huang[1], Zitao Liu[2], Peng Hu[1], Xi Peng[1]*
[1] College of Computer Science, Sichuan University.
[2] TAL Education Group, Beijing China.
{yangmouxing, yunfanli.gm, zyhuang.gm, zitao.jerry.liu, penghu.ml, pengx.gm}@gmail.com

Abstract

In real-world applications, it is common that only a portion of data is aligned across views due to spatial, temporal, or spatiotemporal asynchronism, thus leading to the so-called Partially View-aligned Problem (PVP). To solve such a less-touched problem without the help of labels, we propose simultaneously learning representation and aligning data using a noise-robust contrastive loss. In brief, for each sample from one view, our method aims to identify its within-category counterparts from other views, and thus the cross-view correspondence could be established. As the contrastive learning needs data pairs as input, we construct positive pairs using the known correspondences and negative pairs using random sampling. To alleviate or even eliminate the influence of the false negatives caused by random sampling, we propose a noise-robust contrastive loss that could adaptively prevent the false negatives from dominating the network optimization. To the best of our knowledge, this could be the first successful attempt of enabling contrastive learning robust to noisy labels. In fact, this work might remarkably enrich the learning paradigm with noisy labels. More specifically, the traditional noisy labels are defined as incorrect annotations for the supervised tasks such as classification. In contrast, this work proposes that the view correspondence might be false, which is remarkably different from the widely-accepted definition of noisy label. Extensive experiments show the promising performance of our method comparing with 10 state-of-the-art multi-view approaches in the clustering and classification tasks. The code will be publicly released at https://pengxi.me.

1. Introduction

Multi-view Representation Learning (MvRL) [2, 16, 24, 37, 44] aims at learning consistent representations from multi-view/modal data to facilitate the downstream tasks including but not limited to clustering, classification, and retrieval. The success of all existing works [2, 37, 44] heavily relies on two assumptions, *i.e.*, the completeness of data and the consistency of views. To be specific, the completeness assumption requires that the instances are presented in all views, and the consistency assumption requires that the data from different views must be strictly aligned. When one of these two assumptions is unsatisfied, it is impossible to perform MvRL. In practice, however, the two assumptions could be easily violated in data collection or transmission,

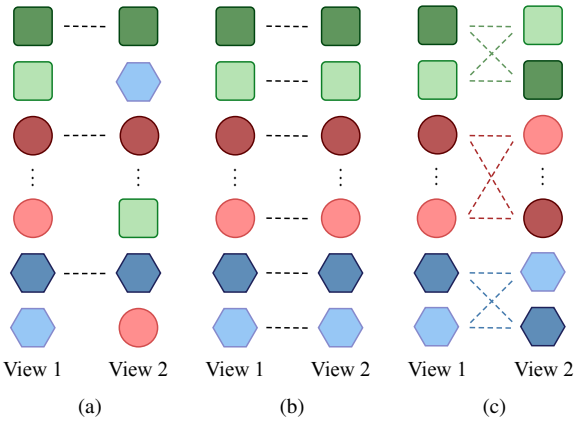


Figure 1. The motivation of this paper. In the figure, different colors denote different instances which will present in multiple views, different shapes indicate different categories, and the dotted line denotes the desired correspondence. (a) Partially View-aligned problem: only a portion of data is with the known correspondence due to complexity of data collection and transmission; (b) Instance-level alignment: it aims to establish the correspondence between two cross-view samples of the same instance; (c) Category-level alignment: each pair consists of the samples belonging to the same category. Considering the downstream tasks such as clustering and classification, category-level alignment is more desirable than instance-level alignment due to its higher accessibility and scalability.

*Corresponding author

thus resulting in *Partially Data-missing Problem* (PDP) and *Partially View-aligned Problem* (PVP, see Fig. 1(a)). More specifically, PDP happens when some data is missed in some views, thus leading to data incompleteness. PVP refers to the case when only a portion of data is aligned, thus leading to data inconsistency. Recently, several works have made remarkable progress on PDP [15, 27, 40], but only a few studies have been conducted to solve PVP.

In this paper, we try to solve PVP without the help of data annotations. Our observation and motivation are shown in Fig. 1. Ideally, the data is highly expected to be fully aligned at the instance level as shown in Fig. 1(b). To achieve this goal, a straightforward solution is using the Hungarian algorithm as a preprocessing step to build the correspondence of two views and then passing the aligned data into a standard multi-view method to learn representation. However, the performance of such a two-stage learning paradigm is sub-optimal because the Hungarian algorithm i) cannot be applied to the multi-view raw spaces which are heterogeneous; and ii) will not utilize the known correspondence in data. In very recent, Partially View-aligned Clustering (PVC) [18] proposed a differentiable neural module of the Hungarian algorithm, and thus data alignment and representation learning could be achieved in a one-stage manner. However, both the vanilla Hungarian algorithm and PVC aim to achieve instance-level alignment which might be over-sufficient to multi-view clustering and classification. Different from the one-to-one mapping tasks like retrieval [8, 17], the essence of clustering and classification is a one-to-many mapping. Therefore, the category-level alignment is more desirable than the instance-level alignment for clustering and classification thanks to its higher accessibility and scalability. Intuitively, for a given cross-view instance, it has a random probability of $1/N$ and $1/K$ to be aligned at the instance- and category-level correctly, where N and K are the number of instances and categories, and $K \ll N$. In other words, the category-level alignment enjoys higher accessibility. On the other hand, the computational complexity of the instance-level alignment methods such as the Hungarian algorithm is $O(N^3)$ which prohibits it from handling large-scale datasets.

Based on the above observation and motivation, we solve PVP by trying to achieve the category- instead of instance-level alignment as shown in Fig. 1(c). To the end, we propose a novel partially view-aligned representation learning method, termed Multi-view Contrastive Learning with Noise-robust loss (MvCLN). Our basic idea is reformulating the view alignment problem as an identification task. Specifically, taking bi-view data as a showcase, for each sample from one view, MvCLN aims to identify its counterparts that belong to the same category from the other view. To train MvCLN, we construct the positive pairs using the available aligned data and the Negative Pairs (NP) using random sampling. To alleviate or even eliminate the influence of the False-Negative Pairs (FNP) caused by random sampling, our MvCLN is with a novel noise-robust contrastive loss. The contributions of this work could be summarized as follows:

- To facilitate one-to-many mapping tasks like clustering, we propose solving PVP by establishing category- rather than instance-level alignment. Such a task-specific alignment enjoys higher accessibility and scalability as shown in the above analysis and the following experiments;
- We reformulate the alignment problem as a view identification task which is further performed under the contrastive learning framework. To the best of our knowledge, this could be one of the first works by employing contrastive learning to achieve the category-level alignment;
- To establish the view correspondence using contrastive learning, we propose a novel noise-robust contrastive loss which could alleviate or even eliminate the influence of noisy labels (*i.e.*, FNP) introduced during pair construction. As far as we know, this could be the first contrastive learning method with the capacity of handling noisy labels. It should be pointed out that, **the traditional noisy labels are defined as incorrect annotations for the supervised tasks such as classification. In contrast, this work proposes that the view correspondence might be false, which is remarkably different from the traditional definition.** As a result, our study might enrich the learning paradigm with noisy labels.

2. Related Works

In this section, we give a brief review of some recent developments that are related to this work.

2.1. Multi-view Representation Learning

Generally, most existing MvRL methods highly rely on the completeness and consistency assumptions of multi-view data. As introduced in Section 1, most multi-view representation learning methods cannot handle the partially data-missing problem (PDP) and partially view-aligned problem (PVP). From this perspective, the existing multi-view learning works can be grouped into three categories. Namely, the vanilla multi-view learning methods [2, 4, 32, 36, 37, 41, 42, 44, 47] which aim at exploiting the homogeneous and complementary information of different views to learn representation; Incomplete multi-view learning methods [15, 26, 27, 40] which utilize the complete views to predict the missing views; Partially view-aligned representation learning methods [18, 21, 39] which establish

the correspondence of unaligned data and almost all existing studies achieve the correspondence at the instance level.

Among the aforementioned studies, [18, 21, 39] are most related to this work. Different from them, we perform alignment at the category rather than instance level. To be specific, two arbitrary cross-view samples are defined to be aligned in our study *i.f.f.* they belong to the same category. Considering the downstream tasks including clustering and classification, such a category-level alignment scheme is more desirable than instance-level alignment which is more expensive in accessibility and scalability. Furthermore, the method based on metric learning [39], which achieve alignment under the scenario of supervised learning, is less challenging than our setting because the category-level alignment could be naturally derived from annotations.

2.2. Contrastive Learning

Contrastive Learning, which is a recently proposed unsupervised learning paradigm [5, 6, 10, 12, 23, 31], has achieved state of the arts in a variety of tasks. The main difference of them lies in the used data augmentation strategy and contrastive loss. In brief, most contrastive learning methods first construct positive and negative pairs at the instance-level through a series of data augmentation. After that, different contrastive losses such as Triplet [33], NCE [11], and NT-Xent [6], could be used to maximize the similarity between positive pairs while minimizing those of the negatives.

The difference of this work with the existing methods [5, 6, 10–12, 31, 33] are given below. First, this study aims to handle multi- instead of single-view data. In other words, these contrastive learning methods cannot be directly used to handle multi-view data, especially, when PVP happens. Second, we do not employ data augmentation to build data pairs. Instead, we directly use the available aligned data as positives and perform random sampling on the observed data to build negatives, which leading to a noisy label problem. Third, our method is with a novel contrastive loss which is robust against the noisy labels. As far as we know, the contrastive learning with noisy labels has not been touched so far.

2.3. Learning with Noisy Labels

In recent, some studies [13, 14, 28, 34, 38] have been conducted to enable neural networks robust against noisy labels, which have attracted interests from the community. In general, these existing works aim at handling the incorrect annotations for the supervised tasks such as classification. Different from them, this work proposes that the view correspondence might be false, and strive to solve such a special noisy label issue.

3. Method

In this section, we propose a partially view-aligned representation learning method to solve PVP, termed Multi-view Contrastive Learning with Noise-robust loss (Mv-CLN). This section is organized as follows. First, Section 3.1 introduces how to reformulate the alignment problem as a category-level identification task which is further achieved through contrastive learning. Section 3.2 elaborates on the proposed noise-robust contrastive loss to alleviate or even eliminate the influence of noisy pairs which are inevitable for the unsupervised pair construction. Section 3.3 presents the necessity of our two-stage optimization from the theoretical and experimental perspectives. Finally, Section 3.4 presents the implementation details of our model.

3.1. Problem Formulation

Let $\{\mathbf{X}^i\}_{i=1}^v = \{\mathbf{x}_1^i, \mathbf{x}_2^i, \ldots, \mathbf{x}_N^i\}_{i=1}^v$ be a partially view-aligned dataset, *i.e.*, $\{\mathbf{X}^i\}_{i=1}^v = \{\mathbf{A}^i, \mathbf{U}^i\}_{i=1}^v$, where v refers to the number of views, the aligned and unaligned data are denoted by $\{\mathbf{A}^i\}_{i=1}^v = \{\mathbf{a}_1^i, \mathbf{a}_2^i, \ldots, \mathbf{a}_{N_1}^i\}_{i=1}^v$ and $\{\mathbf{U}^i\}_{i=1}^v = \{\mathbf{u}_1^i, \mathbf{u}_2^i, \ldots, \mathbf{u}_{N_2}^i\}_{i=1}^v$, respectively. We aim to align $\{\mathbf{U}^i\}_{i=1}^v$ by utilizing $\{\mathbf{A}^i\}_{i=1}^v$ and simultaneously learn a common representation for the whole dataset.

Without loss of generality, we take $v = 2$ as a showcase. A data set is aligned at the category-level when $\mathbf{x}_k^1$ and $\mathbf{x}_k^2$ belong to the same category, *i.e.*,

$$C(\mathbf{x}_k^1) = C(\mathbf{x}_k^2), \forall k \in [1, N], \tag{1}$$

where $C(x)$ denotes the category of x. The category-level alignment can be achieved by solving an identification task which aims to identify the counterpart $\mathbf{x}_k^2$ for $\mathbf{x}_k^1$ to satisfy the above objective.

To complete the identification task, the category-level contrastive learning [12], which aims at increasing the similarity of positive pairs while minimizing those of the negatives, could be used. However, it is impossible to directly use contrastive learning for performing the identification task due to the following limitations. On the one hand, our setting only contains the positive pairs $\{\mathbf{A}^i\}_{i=1}^v$, and thus it is necessary to construct the negative pairs from data. On the other hand, without the help of labeled data, it is inevitable to obtain some noisy negative pairs despite the used data pair construction method. Therefore, we propose using random sampling to generate the negatives for simplicity. To be specific, we randomly choose two samples $\mathbf{a}_i^1$ and $\mathbf{a}_j^2$ from $\{\mathbf{A}^i\}_{i=1}^v$ as a negative pair, where $i \neq j$. Intuitively, the constructed pairs have a probability of $1/K$ to be noisy when the categories are uniformly distributed, where K is the class number. Therefore, our goal becomes making contrastive learning robust to noisy labels (*i.e.*, false negatives).

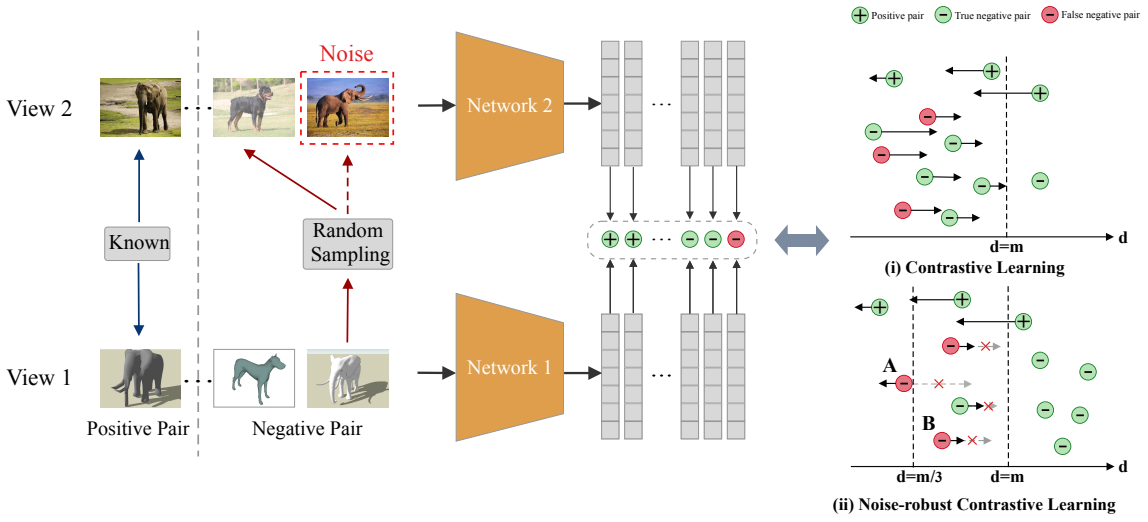


Figure 2. To learn common representation from a partially view-aligned dataset, the proposed MvCLN consists of pair construction and noise-robust optimization. To obtain the data pairs, MvCLN takes the aligned data $\{A^i\}_{i=1}^2$ as positive pairs and all samples in $\{A^1\}$ as anchors. With each anchor together, MvCLN randomly chooses M samples from $\{A^2\}$ to form M negative pairs. In such a random sampling process, some positive pairs will be wrongly regarded as negative pairs, thus leading to a noisy label problem. Based on our analysis (see Section 3.3), MvCLN solves this problem by employing a two-stage optimization strategy. More specifically, (i) contrastive learning: it aims to increase the distance of true negatives over a data-adaptive margin m so that the discrimination between true and false negatives is maximized. After (i), the distances of most true negatives are greater than m, that of some negatives ranges into $(0, m/3)$, and that of the other negatives will fall into $(m/3, m)$. Then, we switch to (ii) noise-robust contrastive learning: it will alleviate the influence of the false negatives by reducing the magnitude of the gradient (see point B) or even reversing the direction of the gradient (see point A). In (i) and (ii), the direction and length of the arrows refer to the direction and magnitude of gradients of the loss, respectively.

3.2. Noise-robust Contrastive Loss

To alleviate or even eliminate the influence of false negatives, we propose the following loss function:

$$\mathcal{L} = \frac{1}{2N}\sum_{i=1}^{N}\left(P\mathcal{L}_i^{pos} + (1-P)\mathcal{L}_i^{neg}\right), \tag{2}$$

where N denotes the number of data pairs, and $P = 1/0$ for positive/negative pairs. Clearly, $\mathcal{L}_i^{pos}$ will take effects when the pairs are positive, and $\mathcal{L}_i^{neg}$ acts on negative pairs.

For the positive cross-view samples $\mathbf{a}_i^1$ and $\mathbf{a}_i^2$, we aim to minimize their distance in a latent space by minimizing

$$\mathcal{L}_i^{pos} = d(\mathbf{a}_i^1, \mathbf{a}_i^2), \tag{3}$$

where

$$d(\mathbf{a}_i^1, \mathbf{a}_i^2) = \left\| f_1(\mathbf{a}_i^1) - f_2(\mathbf{a}_i^2) \right\|_2^2, \tag{4}$$

and f_1 and f_2 denote two parametrized neural networks which project two views (the data pairs) into a latent space, respectively.

If only minimizing the distance of positive pairs, all samples might collapse to one point. To avoid the trivial solution, the following contrastive term could be helpful:

$$\mathcal{L}_i^{ctr} = \max(m - d(\mathbf{a}_i^1, \mathbf{a}_j^2), 0)^2, \tag{5}$$

where m is a margin to enforce the distance of negatives to be moderately large, $(\mathbf{a}_i^1, \mathbf{a}_j^2)$ is a negative pair. This is the loss of the well-known SIAMESE network [12].

As the above loss does not explicitly embrace the robustness to noisy labels, it would mix up the true- and false-negative pairs, thus obtaining degraded performance as demonstrated in Fig. 3(c) and 3(d). Therefore, to enjoy the robustness against the false negatives, we propose the following noise-robust loss:

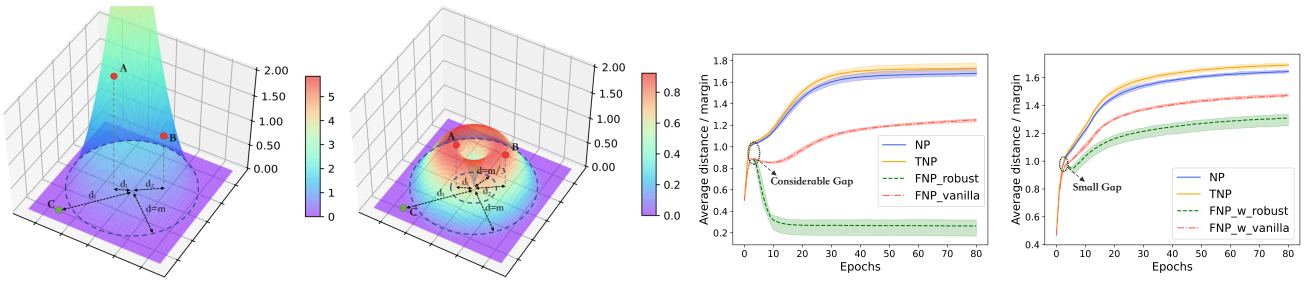
$$\mathcal{L}_i^{neg} = \frac{1}{m}\max(md^{\frac{1}{2}}(\mathbf{a}_i^1, \mathbf{a}_j^2) - d^{\frac{3}{2}}(\mathbf{a}_i^1, \mathbf{a}_j^2), 0)^2 \tag{6}$$

where m is the margin computed only once at the initial state through

$$m = \frac{1}{N_p}\sum d(\mathbf{a}_i^1, \mathbf{a}_i^2) + \frac{1}{N_n}\sum d(\mathbf{a}_i^1, \mathbf{a}_j^2), \tag{7}$$

where N_p and N_n are the number of positives and negatives, respectively.

Thanks to the formulation of Eq. 6, MvCLN could prevent the networks from fitting false negatives or even correct the wrong optimization direction as shown in Fig. 3(c) and 3(d). The detailed analysis will be presented in the next



(a) (b) (c) (d)

Figure 3. Mathematical and experimental analysis of our noise-robust contrastive loss. (a-b) The loss value of Eq. 5 and Eq. 6 w.r.t. the distance of data pairs. To better illustrate why our loss could be robust against the noisy labels, we consider all three possible cases on the performance surface. Namely, A, B, and C refer to the false-negative pairs (FNP) with the distance of $d_1 < m/3$, of $m/3 < d_2 < m$, and the true negative pairs (TNP) with the distance of $d_3 > m$. (a) shows that the vanilla contrastive loss (Eq. 5), which will increase the distance of all negatives including A, B and C, cannot handle the noisy labels. In contrast, (b) shows that our loss could decrease the distance of A and slowly increasing that of B, thus enjoying the robustness against the noisy labels. (c-d) The ratio of average distance to margin with increasing epoch on NoisyMNIST and Reuters datasets, where NP, FNP_robust, and FNP_vanilla denotes negative pairs, false-negative pairs optimized by our loss (Eq. 6), and false-negative pairs optimized by the vanilla loss (Eq. 5). The colored areas denote the variances of five network initializations. One could observe that with more training epochs, our loss will remarkably enlarge the distance gap between TNP and FNP. In fact, on NoisyMNIST, our loss could even correct the gradient direction of the noisy labels, *i.e.*, FNP could be treated as true positives as desired.

section to explain why our loss could enjoy the above desirable properties from mathematical and experimental perspectives.

3.3. Analysis on the Proposed Loss

In this section, we carry out mathematical and experimental analysis to show why the proposed loss function could be robust to the noisy labels and why our model adopts a two-stage optimization strategy.

Let the gradient of $\mathcal{L}^{neg}$ w.r.t. the distance of negatives d be zero and we only need to consider $d \leq m$, *i.e.*,

$$\begin{aligned}\frac{\partial \mathcal{L}^{neg}}{\partial d} &= \frac{\partial(\frac{1}{m}d^3 - 2d^2 + md)}{\partial d} \\ &= \frac{3}{m}d^2 - 4d + m,\end{aligned} \tag{8}$$

then $d = m/3$ or $d = m$. As a result, the performance surface will be divided into two areas, namely, $0 < d < m/3$ and $m/3 < d < m$.

To visually illustrate the above theoretical result, we show the loss surface w.r.t. the distance of a given negative pairs in Figs. 3(a)–3(b). From the results, one could observe that comparing with the vanilla loss (Eq. 5), the optimization of our noise-robust loss (Eq. 6) will not monotonically increase the distance of negative pairs, thus enjoying the following two characteristics:

Reverse optimization ($0 < d < m/3$): For the negative pairs locating into the hole area (see A for example), the gradient of our loss will be reversed, and thus the distance of negative pairs will decrease.

Slow optimization ($m/3 < d < m$): For the pairs locating into the area of $m/3 < d < m$ (see B for example), the optimization speed of our loss will be slower than that of the vanilla loss because the gradient is always with the negative value and the gradient of the former is greater than that of the latter. Mathematically,

$$\begin{aligned}\Delta &= \frac{\partial \mathcal{L}^{neg}}{\partial d} - \frac{\partial \mathcal{L}^{ctr}}{\partial d} \\ &= \frac{\partial(\frac{1}{m}d^3 - 2d^2 + md)}{\partial d} - \frac{\partial(d^2 - 2dm + m^2)}{\partial d} \\ &= \frac{\partial(d-m)^2}{\partial d} \geq 0.\end{aligned} \tag{9}$$

Clearly, the first characteristic could be used to eliminate the influence of the false negatives if the false negatives are constrained into $0 < d < m/3$. Alternatively, the second one could be used to alleviate the influence of the false negatives by constraining the false negatives into $m/3 < d < m$. However, the problem, which has become how to distinct the false negatives from the true ones, is a daunting task in practice.

Fortunately, Bengio *et al*. [3] have empirically found that the neural networks apt to fit the simple patterns first, which provide a motivation to us. To be specific, we propose that TNP could be regarded as simple patterns and FNP could be treated as the complex ones. As a result, it is reasonable to conjecture that the neural networks with the vanilla contrastive loss will fit TNP faster than FNP, as verified in Fig. 3(c) and 3(d). More specifically, the figures illustrate that there is a gap between TNP and FNP_vanilla in the early training stage due to the faster fitting speed of TNP.

Thanks to the above observation, we propose adopting a two-stage optimization strategy to distinct FNP from TNP. In short, this first stage will employ the vanilla contrastive loss (Eq. 5) to optimize our model until the average distance of all negative pairs larger than m. As a result, most TNP and FNP will locate into the areas of $d > m$ and $d < m$, respectively, due to the faster fitting speed of TNP. Then, our model will switch into the second optimization stage with the noise-robust contrastive loss, i.e., Eq. 6. In this stage, as most FNP will locate into either $m/3 < d < m$ or $0 < d < m/3$, the distance of FNP will either increase slowly (see FNP_robust in Fig. 3(d)) or decrease (see FNP_robust in Fig. 3(c)), thus alleviating or even eliminating the influence of noisy labels. Meanwhile, it only imposes negligible effects on true negative pairs since most of their distances are larger than m so far.

3.4. Implement Details

In this section, we first elaborate on the implementation details of the proposed MvCLN and then show how to conduct MvCLN for category-level alignment while learning common representation for different views.

As shown in Fig. 2, MvCLN will first construct data pairs by simply treating the available aligned data $\{\mathbf{A}\}_{i=1}^{2}$ as positive pairs and performing random sampling to obtain the negative pairs. More specifically, MvCLN takes each sample in $\{\mathbf{A}^1\}$ as an anchor and randomly samples M samples from $\{\mathbf{A}^2\}$ to form M negative pairs. In other words, the ratio of positives to negatives is $1/M$.

After obtaining the data pairs, MvCLN will pass them into two neural networks (f_1 and f_2) that are with the dimensionality of D-1024-1024-1024-10, where D is the dimension of inputs. All the layers are densely connected followed by a Batch Normalization [20], a ReLU [29], and a Dropout layer [35].

As elaborated in Section 3.3, our model adopts a two-stage optimization strategy. In short, MvCLN is optimized using the vanilla contrastive loss (Eq. 5) with SGD until the average distance of all NP reaches the margin m. After that, MvCLN is continuously optimized using the noise-robust contrastive loss (Eq. 6).

Once our model converges, the category-level alignment could be achieved through the following two steps:

- Step 1 (distance calculation): obtaining the representation $f_1(\mathbf{X}^1)$ and $f_2(\mathbf{X}^2)$ and computing the their distance matrix $\mathbf{D} \in \mathbb{R}^{N\times N}$. In our implementation, we simply adopt the Euclidean distance.

- Step 2 (alignment): For each sample $\mathbf{x}_i^1$ in one view, its correspondence in another view is with the smallest $\mathbf{D}_{ij}$.

After establishing the correspondence across views, we concatenate the representations of the aligned data for the downstream tasks.

4. Experiments

We carry out experiments on four widely-used multi-view datasets and evaluate the learned representation through clustering and classification tasks. Due to the space limitation, we present the classification results and more details in the supplementary material. To verify the effectiveness of our MvCLN in clustering, we use 10 start-of-the-art multi-view clustering methods as baselines and ACC, NMI, and ARI as performance metrics.

4.1. Experimental Configurations

We implement MvCLN in PyTorch 1.5.0 and carry out all evaluations on a standard Ubuntu-16.04 OS with an NVIDIA 2080Ti GPU. To optimize MvCLN, the Adam optimizer [22] with an initial learning rate of 0.001 is adopted, and no scheduler or weight decay is used. The batch size is fixed to 1024 for all the datasets.

In the experiments, four multi-view datasets are used, Namely, Scene-15 [7, 9] and Caltech-101 [25, 45] with two image features extracted as views, Reuters [1, 19] with the first two languages (English and French) as two views, and NoisyMNIST [37] with 30,000 randomly selected samples since the baselines cannot handle the original large-scale dataset. More details are describe in the supplementary material. Unless otherwise specified, for each dataset $\{\mathbf{X}^v\}_{v=1}^{2}$, we randomly divide it into two partitions with the equal size, namely $\{\mathbf{A}^v\}_{v=1}^{2}$ and $\{\mathbf{U}^v\}_{v=1}^{2}$. In training, only $\{\mathbf{A}^v\}_{v=1}^{2}$ is used as the positive pairs and the negative pairs are obtained by performing random sampling as elaborated in Section 3.4.

4.2. Comparisons with State of The Arts

In this section, we compare the proposed MvCLN with 10 multi-view clustering methods, including CCA [36], KCCA [4], DCCA [2], DCCAE [37], LMSC [43], MvC-DMF [46], SwMC [30], BMVC [45], AE^2-Nets [44], and PVC [18]. For all the baselines, we tune the parameters as suggested in the original paper to achieve their best performance. For our MvCLN, we fix the negative/positive ratio M to 30 for all datasets. To achieve clustering, k-means is conducted on the representations learned by all tested methods except Mvc-DMF, SwMC and BMVC.

As only PVC and our MvCLN could solve PVP, for fair comparisons, the results of the other tested methods are reported under the following two settings:

- Setting 1 (Partially View-aligned Data): We first use PCA to project the raw data into a latent space whose dimensionality is the same as MvCLN so that the Hungarian algorithm could be applied to establish corre-

Table 1. Clustering comparisons on four widely-used multi-view datasets, where the best result for each setting is in bold and "–" indicates that the method cannot obtain the result due to over-high time or memory cost.

Aligned	Methods	Scene-15			Caltech-101			Reuters			NoisyMNIST		
		ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
Fully	CCA (NeurIPS'03)	36.37	36.91	19.82	20.25	45.41	16.34	44.31	20.34	14.52	71.31	52.60	48.46
	KCCA (JMLR'02)	37.93	37.42	21.38	21.45	45.58	17.62	**50.87**	22.34	**20.61**	**96.85**	**92.10**	**93.23**
	DCCA (ICML'13)	36.61	39.20	21.03	27.60	47.84	**30.86**	47.95	**26.57**	12.71	89.64	88.33	83.95
	DCCAE (ICML'15)	34.58	39.01	19.65	19.84	45.05	14.57	41.98	20.30	8.51	78.00	81.24	68.15
	LMSC (CVPR'17)	38.46	35.50	20.54	26.87	48.80	18.06	38.56	20.12	15.48	–	–	–
	MvC-DMF (AAAI'17)	30.99	31.35	15.68	24.35	44.98	14.82	33.83	14.89	12.59	74.39	63.22	49.79
	SwMC (IJCAI'17)	33.89	32.98	11.78	**30.74**	36.07	7.75	33.65	16.02	5.90	–	–	–
	BMVC (TPAMI'18)	**40.74**	**41.67**	**24.19**	27.59	46.43	21.28	42.39	21.86	15.14	88.31	77.01	76.58
	AE^2-Nets (CVPR'19)	37.17	40.47	22.24	20.79	45.01	15.89	42.39	19.76	14.87	42.11	43.38	30.42
Partially	CCA (NeurIPS'03)	32.73	34.24	18.80	20.06	41.56	**16.62**	40.87	**15.82**	12.68	34.46	29.83	17.89
	KCCA (JMLR'02)	33.09	31.43	16.35	12.57	31.36	7.65	40.08	11.80	11.27	26.57	18.19	10.55
	DCCA (ICML'13)	34.27	36.55	18.83	12.52	32.13	7.63	39.71	13.83	**14.38**	29.22	20.24	11.08
	DCCAE (ICML'15)	33.62	36.56	18.54	11.75	30.54	6.60	**41.42**	12.82	13.61	27.61	19.45	10.00
	LMSC (CVPR'17)	26.27	20.45	10.93	**21.54**	**40.26**	15.51	32.17	11.34	7.19	–	–	–
	MvC-DMF (AAAI'17)	28.49	24.31	11.22	9.54	23.41	3.84	32.58	12.36	11.08	27.34	22.96	6.85
	SwMC (IJCAI'17)	31.03	30.39	12.94	19.03	22.75	3.73	31.92	11.03	5.40	–	–	–
	BMVC (TPAMI'18)	**36.81**	**36.55**	**20.20**	12.13	31.33	7.11	38.15	11.57	12.07	28.47	24.69	14.19
	AE^2-Nets (CVPR'19)	28.56	26.58	12.96	10.45	29.51	7.90	35.49	10.61	8.07	**38.25**	**34.32**	**22.02**
Partially	PVC (NeurIPS'20)	37.88	39.12	20.63	22.11	**47.82**	17.98	42.07	20.43	16.95	81.84	82.29	82.03
	MvCLN (Mean)	**38.53**	**39.90**	**24.26**	**30.09**	43.07	**38.34**	**50.16**	**30.65**	**24.90**	**91.05**	**84.15**	**83.56**
	MvCLN (Best)	39.87	40.47	24.83	35.72	45.25	51.44	56.62	33.62	27.37	94.51	86.77	88.42

spondence of the partially view-aligned data. After that, we conduct these baselines on the realigned data. For PVC and MvCLN, we simply run them on the partially view-aligned data.

- Setting 2 (Fully View-aligned Data): We directly run all methods except PVC and MvCLN on the original data which is fully aligned.

To avoid performance changing due to randomness, we run all methods five times and report the average performance in terms of three performance metrics, *i.e.*, ACC, NMI, and ARI. Note that, due to the difference in the hardware and software environments, the results of some baselines in our experiments are slightly different from that reported in [18]. From Table 1, one could observe that:

- In the first setting, our MvCLN remarkably outperforms all tested methods by a considerable margin. Particularly, MvCLN achieves an ARI improvement of 113.2% and 46.9% on Caltech-101 and Reuters, respectively, comparing with the best baseline. This verifies our claim and motivation that the category-level alignment is more desirable than instance-level alignment;
- In the second setting, MvCLN still achieves competitive results even though it is conducted on partially view-aligned data whereas the baselines are conducted on fully view-aligned data.

Table 2. Ablation studies on NoisyMNIST. "✔" represents MvCLN with the component and "✗" denotes MvCLN without the component.

$\mathcal{L}_{neg}$	ACC	NMI	ARI	CAR
✗	88.2	75.73	75.89	84.48
✔	**92.17**	**85.57**	**84.51**	**88.24**

4.3. Ablation Studies and Parameter Analysis

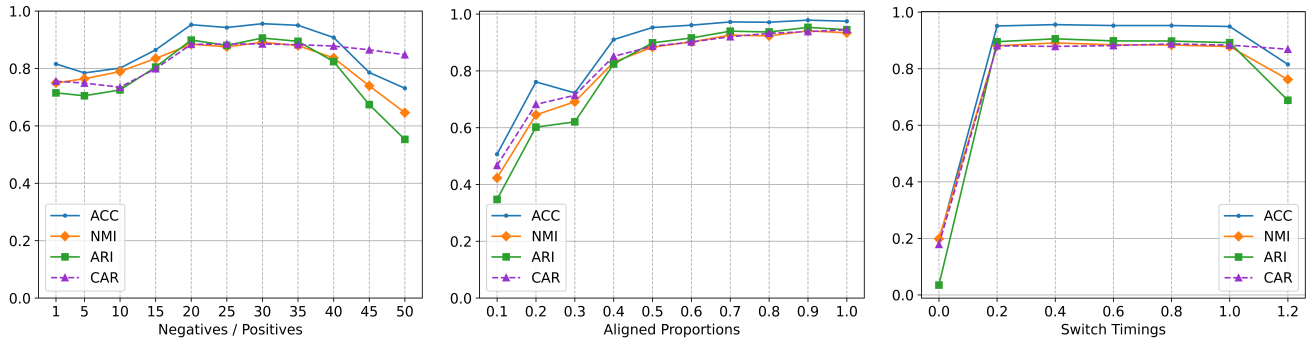
In this section, we conduct the following experimental analyses on NoisyMNIST, *i.e.*, the ablation study, the influence of the ratio of positives to negatives, the influence of aligned proportion, and the influence of the switch timing between two optimization stages.

Besides the used clustering performance metrics, we introduce a metric termed Category-level Alignment Rate (CAR) to measure the ratio of the category-level alignment. Mathematically,

$$CAR = \frac{\sum_{i=1}^{N} \delta(C(\hat{\mathbf{x}}_i^1), C(\hat{\mathbf{x}}_i^2))}{N}, \tag{10}$$

where $(\hat{\mathbf{x}}_i^1, \hat{\mathbf{x}}_i^2)$ denotes the realigned pairs, δ is the Dirichlet function, and N is the number of data pairs.

Effectiveness of Noise-robust Contrastive Loss: To prove the effectiveness of the proposed noise-robust contrastive loss, we replace it with the vanilla contrastive loss, *i.e.*, Eq. 5. As shown in Table 2, although the vanilla contrastive loss, which could also achieve some promising re-



(a) (b) (c)

Figure 4. Performance analysis on the NoisyMNIST dataset. (a) Performance with negatives/positives ratio (M); (b) Performance with varying aligned proportions; (c) Performance with different switching times of two optimization stages.

sults, is remarkably worse than MvCNL.

Influence of the Positive/Negative Pairs Ratio: Our method uses the aligned data as positives and randomly chosen data as negatives. In other words, it is easy to obtain the negatives. However, an over-high negatives/positive pairs ratio (M) will lead to imbalanced data distribution. Hence, intuitively, it is crucial to determine the value of M. In Fig. 4(a), we investigate the performance of MvCLN by increasing M from 1 to 50 with an interval of 5. From the results, one could have the following observations. On the one hand, the increasing M will enhance the performance of MvCLN within a quite large value range. On the other hand, MvCLN performs stably when M ranges into $[20, 40]$, which shows its robustness to the parameter.

Influence of Different Aligned Proportions: To investigate the performance of MvCLN on the data with different aligned proportions, we increase the aligned proportion from 10% to 100% with an interval of 10%. From the results, one could observe that: i) with more available aligned data, MvCLN achieves better results; ii) when the aligned proportion increases from 70% to 100%, the performance of MvCLN slightly increase. The possible reason is that 70% aligned data is sufficient to enable MvCLN to learn the alignment patterns.

Influence of the Switching Time between Two Optimization Stages: Our method consists of two optimization stages which are automatically switched in a data-driven way as elaborated in Section 3.4. In this section, we experimentally investigate the influence of the following seven switching criteria, *i.e.*, switching to Stage 2 when the mean distance of negative pairs reaches $0.0m$, $0.2m$, $0.4m$, $0.6m$, $0.8m$, $1.0m$, and $1.2m$, where the margin m is automatically determined from data. As shown in Fig. 4(c), MvCLN will achieve a stable result within the range of $[0.2m, 1.0m]$. Without Stage 1 (*i.e.*, $0.0m$), MvCLN will achieve an inferior result which is consistent with the analysis in Fig. 3(b). When the switching time is too late ($1.2m$), the distance of most false-negative pairs may approach or even surpass m, and thus the false and true negative pairs will be mixed.

5. Conclusion

In this paper, we proposed MvCLN which handles the partially view-aligned problem by endowing contrastive learning with the robustness against noisy labels. Different from the existing solutions, our MvCLN aims to achieve the category- instead of instance-level alignment. Extensive experiments verify the effectiveness and efficiency of our learning paradigm. In addition, we theoretically and experimentally show why our model could be robust against the noisy labels. To the best of our knowledge, the proposed method could be regarded as the first study of enabling contrastive learning robust to noisy labels (*i.e.*, false-negative corresponding pairs). What is more important, this work might remarkably enrich the learning paradigm with noisy labels by treating the view correspondence as a special noisy label issue. In the future, we plan to explore a more general solution for the situation wherein both positive and negative pairs would be contaminated by noise. Such a solution is valuable to a variety of applications including but not limited to ReID, object tracking, face identification, graph matching, image translation and restoration.

6. Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant 2020YFB1406702 and 2020AAA0104500; in part by the Fundamental Research Funds for the Central Universities under Grant YJ201949; in part by NFSC under Grant U19A2081, 61625204, 61836006, U19A2078; in part by the Fund of Sichuan University Tomorrow Advancing Life; and in part by Beijing Nova Program (Z201100006820068) from Beijing Municipal Science & Technology Commission.

References

[1] Massih-Reza Amini, Nicolas Usunier, and Cyril Goutte. Learning from multiple partially observed views - an application to multilingual text categorization. In *NeruIPS*, pages 28–36, 2009. 6

[2] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *ICML*, pages 1247–1255, 2013. 1, 2, 6

[3] Devansh Arpit, Stanislaw K Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron C Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *ICML*, pages 233–242, 2017. 5

[4] Francis R Bach and Michael I Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3(Jul):1–48, 2002. 2, 6

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv:2006.09882*, 2020. 3

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv:2002.05709*, 2020. 3

[7] Dengxin Dai and Luc Van Gool. Ensemble projection for semi-supervised image classification. In *ICCV*, pages 2072–2079, 2013. 6

[8] Cheng Deng, Zhaojia Chen, Xianglong Liu, Xinbo Gao, and Dacheng Tao. Triplet-based deep hashing network for cross-modal retrieval. *IEEE Transactions on Image Processing*, 27(8):3893–3903, 2018. 2

[9] Li Fei-Fei and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, volume 2, pages 524–531, 2005. 6

[10] 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, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv:2006.07733*, 2020. 3

[11] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, pages 297–304, 2010. 3

[12] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, volume 2, pages 1735–1742, 2006. 3, 4

[13] Bo Han, Jiangchao Yao, Gang Niu, Mingyuan Zhou, Ivor Tsang, Ya Zhang, and Masashi Sugiyama. Masking: A new perspective of noisy supervision. *arXiv:1805.08193*, 2018. 3

[14] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *arXiv:1804.06872*, 2018. 3

[15] Menglei Hu and Songcan Chen. Doubly aligned incomplete multi-view clustering. *arXiv:1903.02785*, 2019. 2

[16] Peng Hu, Dezhong Peng, Yongsheng Sang, and Yong Xiang. Multi-view linear discriminant analysis network. *IEEE Transactions on Image Processing*, 28(11):5352–5365, 2019. 1

[17] Peng Hu, Xi Peng, Hongyuan Zhu, Jie Lin, Liangli Zhen, and Dezhong Peng. Joint versus independent multiview hashing for cross-view retrieval. *IEEE Transactions on Cybernetics*, 2020. 2

[18] Zhenyu Huang, Peng Hu, Joey Tianyi Zhou, Jiancheng Lv, and Xi Peng. Partially view-aligned clustering. *NeurIPS*, 33, 2020. 2, 3, 6, 7

[19] Zhenyu Huang, Joey Tianyi Zhou, Xi Peng, Changqing Zhang, Hongyuan Zhu, and Jiancheng Lv. Multi-view spectral clustering network. In *IJCAI*, pages 2563–2569, 2019. 6

[20] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv:1502.03167*, 2015. 6

[21] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, pages 3128–3137, 2015. 2, 3

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 6

[23] Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. Contrastive clustering. In *AAAI*, 2021. 3

[24] Yingming Li, Ming Yang, and Zhongfei Zhang. A survey of multi-view representation learning. *IEEE Transactions on Knowledge and Data Engineering*, 31(10):1863–1883, 2018. 1

[25] F Li Fei-Fei, M Andreetto, MA Ranzato, and P Perona. Caltech101. *Computational Vision Group, California Institute of Technology*, 2003. 6

[26] Yijie Lin, Yuanbiao Gou, Zitao Liu, Boyun Li, Jiancheng Lv, and Xi Peng. Completer: Incomplete multi-view clustering via contrastive prediction. In *CVPR*, 2021. 2

[27] Xinwang Liu, Miaomiao Li, Chang Tang, Jingyuan Xia, Jian Xiong, Li Liu, Marius Kloft, and En Zhu. Efficient and effective regularized incomplete multi-view clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2

[28] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *ICML*, pages 6543–6553, 2020. 3

[29] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010. 6

[30] Feiping Nie, Jing Li, Xuelong Li, et al. Self-weighted multiview clustering with multiple graphs. In *IJCAI*, pages 2564–2570, 2017. 6

[31] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv:1807.03748*, 2018. 3

[32] Xi Peng, Zhenyu Huang, Jiancheng Lv, Hongyuan Zhu, and Joey Tianyi Zhou. Comic: Multi-view clustering without parameter selection. In *ICML*, pages 5092–5101, 2019. 2

[33] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015. 3

[34] Hwanjun Song, Minseok Kim, Dongmin Park, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *arXiv preprint arXiv:2007.08199*, 2020. 3

[35] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 6

[36] Alexei Vinokourov, Nello Cristianini, and John Shawe-Taylor. Inferring a semantic representation of text via cross-language correlation analysis. In *NeurIPS*, pages 1497–1504, 2003. 2, 6

[37] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *ICML*, pages 1083–1092, 2015. 1, 2, 6

[38] Xiaobo Wang, Shuo Wang, Jun Wang, Hailin Shi, and Tao Mei. Co-mining: Deep face recognition with noisy labels. In *CVPR*, pages 9358–9367, 2019. 3

[39] Jiwei Wei, Xing Xu, Yang Yang, Yanli Ji, Zheng Wang, and Heng Tao Shen. Universal weighting metric learning for cross-modal matching. In *CVPR*, pages 13005–13014, 2020. 2, 3

[40] Chang Xu, Dacheng Tao, and Chao Xu. Multi-view learning with incomplete views. *IEEE Transactions on Image Processing*, 24(12):5812–5825, 2015. 2

[41] Ming Yin, Weitian Huang, and Junbin Gao. Shared generative latent representation learning for multi-view clustering. In *AAAI*, pages 6688–6695, 2020. 2

[42] Ming Yin, Wei Liu, Mingsuo Li, Taisong Jin, and Rongrong Ji. Cauchy loss induced block diagonal representation for robust multi-view subspace clustering. *Neurocomputing*, 427:84–95, 2021. 2

[43] Changqing Zhang, Qinghua Hu, Huazhu Fu, Pengfei Zhu, and Xiaochun Cao. Latent multi-view subspace clustering. In *CVPR*, pages 4279–4287, 2017. 6

[44] Changqing Zhang, Yeqing Liu, and Huazhu Fu. Ae2-nets: Autoencoder in autoencoder networks. In *CVPR*, pages 2577–2585, 2019. 1, 2, 6

[45] Zheng Zhang, Li Liu, Fumin Shen, Heng Tao Shen, and Ling Shao. Binary multi-view clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1774–1782, 2018. 6

[46] Handong Zhao and Zhengming Ding. Multi-view clustering via deep matrix factorization. In *AAAI*, pages 2921–2927, 2017. 6

[47] Tao Zhou, Changqing Zhang, Xi Peng, Harish Bhaskar, and Jie Yang. Dual shared-specific multiview subspace clustering. *IEEE Transactions on Cybernetics*, 50(8):3517–3530, 2019. 2

Supplementary Material for Partially View-aligned Representation Learning with Noise-robust Contrastive Loss

Mouxing Yang[1], Yunfan Li[1], Zhenyu Huang[1], Zitao Liu[2], Peng Hu[1], Xi Peng[1*]
[1] College of Computer Science, Sichuan University.
[2] TAL Education Group, Beijing China.
{yangmouxing, yunfanli.gm, zyhuang.gm, zitao.jerry.liu, penghu.ml, pengx.gm}@gmail.com

1. Introduction

In this supplementary material, we elaborate on the details of the datasets used and conduct additional experiments to verify the effectiveness of MvCLN.

2. Additional Experiments

In this section, we carry out additional experiments including classification and ablation studies to further show the effectiveness of the proposed MvCLN.

2.1. Details of the Datasets

- **Scene-15** [5]: The dataset consists of 4,485 images distributed over 15 indoor and outdoor scene categories. Similar to [4], two image features are extracted as views, *i.e.*, 20-dim GIST feature and 59-dim PHOG feature;

- **Caltech-101** [8]: The dataset consists of 9,144 images associated with 101 object categories, as well as an additional background category. Following [16], two features are used as views, *i.e.*, 1,984-dim HOG feature and 512-dim GIST feature;

- **Reuters** [1]: A subset, which consists of 18,758 samples of six classes, is used. Similar to [7], we use the first two languages (English and French) as two views and apply a standard autoencoder to project the data into a 10-dim space for faster speed;

- **NoisyMNIST** [13]: The dataset consists of 70,000 samples from 10 classes. As the baselines cannot handle a large-scale dataset, we randomly select 30,000 samples for evaluation.

*Corresponding author

2.2. Classification Performance

To further verify the effectiveness of MvCLN, we perform classification on the learned representations with a comparison of nine multi-view learning methods. The used baselines include CCA [11], KCCA [3], DCCA [2], DCCAE [13], LMSC [14], MvC-DMF [17], BMVC [16], AE^2-Nets [15], and PVC [6]. Note that, SwMC [9] cannot be used in this task since it directly obtains the clustering assignments and does not explicitly learn representations for data. For CCA, KCCA, DCCA, DCCAE, and PVC, we concatenate the obtained representations for classification. For graph-based methods (LMSC and MvC-DMF), we use the spectral representations for classification. For BMVC and AE^2-Nets, we use the common representations for classification. For our MvCLN, we fix the dimensionality of representations to 32. Results with other dimensionalities are shown in Section 2.4.

To achieve classification, we use the SVM classifier contained in the Scikit-Learn package [10] with the default configurations. The representations learned are divided into training and testing sets with different proportions, denoted as $Tr_{train\_ratio}/Te_{test\_ratio}$, where Tr_{train_ratio} indicates the proportion of the training set and Te_{test_ratio} indicates the proportion of the testing set. To avoid randomness due to data partition, we perform the classification 20 times and report the mean classification accuracy.

The results are reported in Table 1, from which one could observe that

- In the partially view-aligned setting, our MvCLN remarkably outperforms all baselines by a considerable margin under different Tr/Te. Particularly, MvCLN achieves an improvement of 26.9% and 12.6% on Caltech101 and Reuters when "Tr/Te" is "8/2", comparing with the best baseline. This further verifies the effectiveness of our MvCLN.

- In the fully view-aligned setting, our MvCLN still

Table 1. Classification performance comparisons on four widely-used multi-view datasets, where the best result for each setting is in bold and "Tr / Te" denotes the size ratio of the training set to testing set. "–" indicates that the method cannot obtain the result due to over-high time or memory cost.

Aligned	Methods	Scene-15			Caltech-101			Reuters			NoisyMNIST		
		8 / 2	5 / 5	2 / 8	8 / 2	5 / 5	2 / 8	8 / 2	5 / 5	2 / 8	8 / 2	5 / 5	2 / 8
Partially	CCA (NeurIPS'03)	**52.49**	**51.52**	**47.95**	35.72	34.56	31.47	64.73	64.70	63.91	65.53	65.05	64.07
	KCCA (JMLR'02)	50.49	48.82	45.75	32.87	31.29	28.90	64.06	63.95	62.83	57.08	56.63	56.06
	DCCA (ICML'13)	51.68	50.64	46.85	35.72	33.97	31.20	**65.92**	**65.80**	**65.15**	60.95	60.90	60.16
	DCCAE (ICML'15)	46.24	45.37	43.75	31.95	30.75	28.14	61.88	61.58	60.67	47.42	47.17	46.26
	LMSC (CVPR'17)	39.15	38.10	36.88	**45.21**	**43.51**	**38.02**	45.03	44.76	44.49	–	–	–
	MvC-DMF (AAAI'17)	36.74	36.42	34.71	20.78	20.08	18.93	41.59	41.34	41.27	33.04	32.64	32.03
	BMVC (TPAMI'18)	50.35	49.83	46.39	33.56	32.83	30.09	64.69	64.20	63.27	72.49	72.03	70.92
	AE^2-Nets (CVPR'19)	48.19	47.64	42.61	23.30	22.65	20.61	62.74	62.40	60.65	**76.58**	**75.87**	**73.75**
Fully	CCA (NeurIPS'03)	57.44	56.21	51.07	37.70	36.14	32.79	69.13	68.67	67.07	87.85	86.09	82.06
	KCCA (JMLR'02)	50.19	50.18	47.26	38.50	36.95	33.72	64.75	64.63	64.63	**97.20**	**97.18**	**97.08**
	DCCA (ICML'13)	63.61	61.72	57.3	38.89	37.23	33.75	71.92	72.33	71.54	96.22	96.34	96.08
	DCCAE (ICML'15)	50.42	48.84	46.48	38.61	37.53	34.03	72.00	71.65	70.63	96.45	96.37	96.08
	LMSC (CVPR'17)	51.28	51.08	48.99	53.92	51.25	42.80	56.09	55.53	54.99	–	–	–
	MvC-DMF (AAAI'17)	43.07	42.45	40.48	48.27	46.71	40.53	42.97	43.08	76.45	75.83	74.05	49.79
	BMVC (TPAMI'18)	66.32	65.16	61.73	**58.57**	**55.69**	**49.92**	**78.65**	**78.20**	**77.73**	92.45	92.47	92.05
	AE^2-Nets (CVPR'19)	**72.03**	**69.76**	**64.66**	35.24	34.38	31.72	65.47	64.82	63.28	89.74	89.33	87.90
Partially	PVC (NeurIPS'20)	48.77	45.97	40.46	36.78	36.50	35.54	72.63	72.08	71.11	93.09	93.12	93.06
	MvCLN (Mean)	**57.93**	**57.15**	**55.52**	**46.69**	**45.89**	**43.87**	**81.77**	**81.63**	**81.11**	**96.19**	**96.18**	**96.15**

achieves competitive results even though the baselines are with ground-truth alignment whereas our method does not. Note that, MvCLN is even better than all the baselines on the Reuters dataset. The possible reason is that the category-level alignment may be more helpful to performance improvement.

Table 2. Clustering performance comparison on the whole NoisyMNIST data. The best result is in bold.

Alignment Type	Method	ACC	NMI	ARI
Partially	MvCLN	**97.50**	**93.09**	**94.57**
Fully	CCA	70.89	52.03	47.91
	KCCA	83.43	88.29	82.59
	DCCA	89.34	91.40	86.87
	DCCAE	89.09	91.37	87.82
	BMVC	91.59	83.48	83.79
	AE^2-Nets	50.83	53.14	40.55

2.3. Clustering on the Whole Dataset

As aforementioned in the manuscript, we carry out all the tested methods on a subset of NoisyMNIST due to the over-high computational cost of the Hungarian algorithm and PVC on the whole dataset. In this section, we carry out our MvCLN on the whole NoisyMNIST in the partially view-aligned setting and conduct some cost-feasible baselines on the same dataset in the fully view-aligned setting for comparison. As shown in Table 2, our MvCLN performs better on the full dataset comparing to the case of the subset, which indicates that more data could do a favor to our method. Besides, MvCLN remarkably outperforms all baselines which are even with fully ground-truth alignment.

2.4. Influence of the Dimensionality

In this section, we investigate the classification performance of representations with different dimensionalities. As shown in Table 3, a higher dimensionality often give better classification performance because more information is contained in the latent space, while giving a high computational complexity. In our implementations, we fix the dimensionality of representations to 32 for all the datasets on the classification task.

2.5. Comparison on the Time Cost

In this section, we quantitatively compare our MvCLN method with the Hungarian algorithm and PVC in terms of the time cost. In the experiments, we employ the package contained in [12] to implement the Hungarian algorithm. As shown in Table 4, our method performs remarkably better than the Hungarian algorithm and PVC in terms of time cost, which verify higher accessibility and scalability of our category-level alignment strategy comparing to the instance-level ones.

2.6. Convergence Analysis

In this section, we investigate the convergence of MvCLN by reporting its loss value, CAR, and clustering performance with the increasing training epoch. From Fig. 1, one could observe that the loss value drops fast in the first 10 epochs, and then slowly decreases until convergence. For

Table 3. Ablation studies on the dimensionality. The best result for each setting is in bold and "Tr / Te" denotes the size ratio of training set to testing set.

Dataset	Dimensionality	8/2	5/5	2/8
Scene-15	10d	47.05	46.60	45.30
	32d	57.93	57.15	55.52
	64d	58.30	57.29	**54.71**
	128d	**58.77**	**57.74**	54.65
Caltech-101	10d	33.78	33.32	32.10
	32d	46.69	45.89	43.87
	64d	**47.17**	**46.45**	**44.01**
	128d	46.97	45.98	43.28
Reuters	10d	75.24	75.10	74.84
	32d	81.77	81.63	81.11
	64d	83.04	82.87	82.32
	128d	**83.94**	**83.88**	**83.19**
NoisyMNIST	10d	95.90	95.89	95.87
	32d	96.19	96.18	96.15
	64d	96.55	96.46	96.41
	128d	**96.58**	**96.62**	**96.55**

Table 4. Time cost comparisons. The best result for each setting is in bold and "–" indicates the method does not involve this phase.

Dataset	Method	training time (s)	inferring time (s)
Scene-15	Hungarian	–	2.69
	PVC	10,907.27	2.01
	MvCLN	**155.53**	**0.72**
Caltech-101	Hungarian	–	48.87
	PVC	11,839.74	7.2
	MvCLN	**388.58**	**1.75**
Reuters	Hungarian	–	289.82
	PVC	18,715.34	30.36
	MvCLN	**790.30**	**3.48**
NoisyMNIST	Hungarian	–	3,778.39
	PVC	53,070.87	34.36
	MvCLN	**1,202.77**	**5.76**

CAR and clustering metrics, they continuously increase in the first 10 epoch and then stay at a high level.

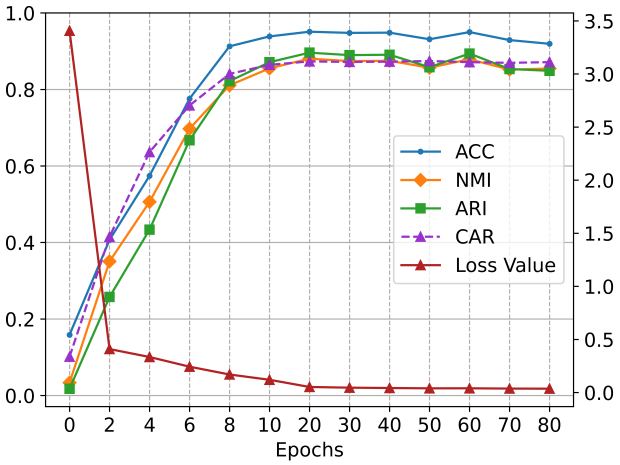


Figure 1. Convergence of MvCLN on the NoisyMNIST dataset. The left and right y-axis denote the performance results and the loss value, respectively.

References

[1] Massih-Reza Amini, Nicolas Usunier, and Cyril Goutte. Learning from multiple partially observed views - an application to multilingual text categorization. In *NeruIPS*, pages 28–36, 2009. 1

[2] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *ICML*, pages 1247–1255, 2013. 1

[3] Francis R Bach and Michael I Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3(Jul):1–48, 2002. 1

[4] Dengxin Dai and Luc Van Gool. Ensemble projection for semi-supervised image classification. In *ICCV*, pages 2072–2079, 2013. 1

[5] Li Fei-Fei and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, volume 2, pages 524–531, 2005. 1

[6] Zhenyu Huang, Peng Hu, Joey Tianyi Zhou, Jiancheng Lv, and Xi Peng. Partially view-aligned clustering. *NeurIPS*, 33, 2020. 1

[7] Zhenyu Huang, Joey Tianyi Zhou, Xi Peng, Changqing Zhang, Hongyuan Zhu, and Jiancheng Lv. Multi-view spectral clustering network. In *IJCAI*, pages 2563–2569, 2019. 1

[8] F Li Fei-Fei, M Andreetto, MA Ranzato, and P Perona. Caltech101. *Computational Vision Group, California Institute of Technology*, 2003. 1

[9] Feiping Nie, Jing Li, Xuelong Li, et al. Self-weighted multiview clustering with multiple graphs. In *IJCAI*, pages 2564–2570, 2017. 1

[10] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011. 1

[11] Alexei Vinokourov, Nello Cristianini, and John Shawe-Taylor. Inferring a semantic representation of text via cross-language correlation analysis. In *NeurIPS*, pages 1497–1504, 2003. 1

[12] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature Methods*, 17(3):261–272, 2020. 2

[13] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *ICML*, pages 1083–1092, 2015. 1

[14] Changqing Zhang, Qinghua Hu, Huazhu Fu, Pengfei Zhu, and Xiaochun Cao. Latent multi-view subspace clustering. In *CVPR*, pages 4279–4287, 2017. 1

[15] Changqing Zhang, Yeqing Liu, and Huazhu Fu. Ae2-nets: Autoencoder in autoencoder networks. In *CVPR*, pages 2577–2585, 2019. 1

[16] Zheng Zhang, Li Liu, Fumin Shen, Heng Tao Shen, and Ling Shao. Binary multi-view clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1774–1782, 2018. 1

[17] Handong Zhao and Zhengming Ding. Multi-view clustering via deep matrix factorization. In *AAAI*, pages 2921–2927, 2017. 1