Deep Spectral Representation Learning From Multi-View Data

Zhenyu Huang, Joey Tianyi Zhou, Hongyuan Zhu, *Member, IEEE*, Changqing Zhang, *Member, IEEE*, Jiancheng Lv, *Senior Member, IEEE*, and Xi Peng, *Member, IEEE*

***Abstract*— Multi-view representation learning (MvRL) aims to learn a consensus representation from diverse sources or domains to facilitate downstream tasks such as clustering, retrieval, and classification. Due to the limited representative capacity of the adopted shallow models, most existing MvRL methods may yield unsatisfactory results, especially when the labels of data are unavailable. To enjoy the representative capacity of deep learning, this paper proposes a novel multi-view unsupervised representation learning method, termed as Multi-view Laplacian Network (MvLNet), which could be the first deep version of the multi-view spectral representation learning method. Note that, such an attempt is nontrivial because simply combining Laplacian embedding (i.e., spectral representation) with neural networks will lead to trivial solutions. To solve this problem, MvLNet enforces an orthogonal constraint and reformulates it as a layer with the help of Cholesky decomposition. The orthogonal layer is stacked on the embedding network so that a common space could be learned for consensus representation. Compared with numerous recent-proposed approaches, extensive experiments on seven challenging datasets demonstrate the effectiveness of our method in three multi-view tasks including clustering, recognition, and retrieval. The source code could be found at www.pengxi.me.**

***Index Terms*— Unsupervised multi-view representation learning, multi-view clustering, cross-modal retrieval.**

I. Introduction

REPRESENTATION learning is crucial to a variety of tasks including recognition, clustering, and retrieval. Over the past decades, a great number of methods have been proposed and achieved promising results [1]–[4]. Among them, manifold learning has taken a dominant position in a long time. The typical methods include locally linear embedding (LLE) [5], Laplacian Eigenmaps (LE) [6], t-SNE [7], low rank representation (LRR) [3] and subspace learning [8]–[11]. The basic idea of these methods is learning a low-dimensional representation by using local/global invariance as an affinity. Despite the success of these methods, they do not explore the latent relations across different views and thus may get inferior performance when handling multi-view data or called multimodal data. Therefore, it is highly expected to develop multi-view representation learning (MvRL) methods to overcome the challenges accompanied by the booming data.

The key point to MvRL is how to exploit the diverse and complementary information contained in different views. The existing multi-view representation learning methods could be roughly classified into three categories, namely, unsupervised methods [12]–[20], supervised methods [21]–[24] and semi-supervised methods [25]–[27]. A comprehensive survey could refer to [28].

In this paper, we only focus on the unsupervised MvRL which learns a shared/consensus representation from different views by exploring the latent data structure without using label information. Unsupervised MvRL methods could be further partitioned into two categories, *i.e.*, shallow models and deep models. One of the most effective shallow MvRL methods is canonical correlation analysis (CCA) [29], [30] which projects different views into one common space by maximizing the correlation of pairwise modalities. Another typical method is the multi-view subspace learning (MvSL) [26], [31]–[37] which learns view-specific representations under the graph Laplacian framework [6] and a common representation by enforcing the view-specific representations as similar as possible. The main difference among the existing works is the formulation of the cross-view consistency or within-view similarity.

Benefited from the powerful nonlinear parametric mapping capacity, deep neural networks (DNNs) have made huge progress in numerous single-modality applications [38], especially, in the scenario of supervised learning. Inspired by the success of DNNs, some deep multi-view approaches have been proposed to learn a common representation, e.g., deep canonical correlation analysis (DCCA) [12], deep canonically correlated autoencoder (DCCAE) [39], multi-view deep matrix factorization (MvDMF) [40], multi-view adversarial

Manuscript received October 6, 2019; revised November 2, 2020 and January 27, 2021; accepted May 10, 2021. Date of current version June 4, 2021. This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant YJ201949; in part by National Natural Science Foundation of China under Grant 61625204, Grant 61836006, Grant 61971296, Grant U19A2078, and Grant 61836011; and in part by the Agency for Science, Technology and Research (A*STAR) through the AME Programmatic Funding Scheme under Project A18A1b0045 and Project A18A2b0046. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Baoxin Li. *(Corresponding author: Xi Peng.)*

Zhenyu Huang, Jiancheng Lv, and Xi Peng are with the College of Computer Science, Sichuan University, Chengdu 610017, China (e-mail: zyhuang.gm@gmail.com; lvjiancheng@scu.edu.cn; pengx.gm@gmail.com).

Joey Tianyi Zhou is with the Institute of High Performance Computing, Agency for Science, Technology and Research, Singapore 138632 (e-mail: joey.tianyi.zhou@gmail.com).

Hongyuan Zhu is with the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore 138632 (e-mail: zhuh@i2r.a-star.edu.sg).

Changqing Zhang is with the School of Computer Science and Technology, Tianjin University, Tianjin 300222, China (e-mail: zhangchangqing@tju.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TIP.2021.3083072, provided by the authors.

Digital Object Identifier 10.1109/TIP.2021.3083072

learning [41]–[43], and multi-view deep hashing methods [44], [45], to name a few. The shared idea of these methods is employing a neural network to learn view-specific representations which are further merged into a common representation by referring to the traditional MvRL paradigm.

Although more and more works have recognized the effectiveness of DNNs in multi-view representation learning, there are no attempts to develop deep multi-view subspace learning approaches so far. In fact, this is not strange due to the following technical challenge. In brief, most MvRL methods always require solving an eigenvalue decomposition (ED) problem (Eq. 2) with the orthogonal constraint, thus making it difficult even impossible in back-propagating the gradient through a neural network. To overcome this challenge, we recast the orthogonal constraint as a neural layer with a theoretical guarantee. In other words, we reformulate a non-differentiable matrix decomposition problem as a differentiable neural layer that could be plugged into existing neural networks. Note that, the orthogonal layer is quite general, which could be used as a surrogate for similar problems.

In this paper, we propose the Multi-view Laplacian Network (MvLNet) which learns a common space from multi-view data using a parametric deep model. Specifically, MvLNet consists of four modules, i.e., 1) a Siamese Network [46] that is used to learn view-specific affinity which could incorporate discrimination with the help of the synthesized positive and negative pairs, 2) an embedding network that is used to learn view-specific representation by preserving the graph structure based on the Laplacian matrix, 3) an orthogonal layer that is used to guarantee the orthogonality of the view-specific representation to avoid the aforementioned gradient back-propagation issue and trivial solutions, and 4) a constraint that is used to learn a common space in which the view-consistency is enforced. In addition, we propose a new training algorithm called alternative orthogonal and embedding (AOE) to train our network in a coordinate descent fashion. To the best of our knowledge, the proposed MvLNet could be the first effective deep extension of multi-view spectral representation learning by solving the gradient back-propagation issue, which is complementary to the classical MvSL and unsupervised deep learning. From the view of the classical MvSL, our work may provide a promising way to boost the performance and revive it in the era of big data and deep learning. Moreover, the constructed orthogonal layer is actually the neural network implementation of the orthogonality constraint, which is pluggable into existing networks to enjoy the benefits from orthogonality. Note that, the orthogonality has been theoretically and experimentally proved effective to representation learning [47]. Extensive experiments on real-world datasets show the superior performance of MvLNet in the task of clustering, classification, and retrieval. Fig. 1 presents the visualization results of MvLNet to show its effectiveness and fast convergence.

The main contributions of our work are summarized as follows:

- We proposed a new multi-view unsupervised representation learning method named MvLNet which integrates the local invariance and the cross-view consistency to learn the common representation progressively. To the best of our knowledge, MvLNet could be the first deep version of the multi-view spectral representation learning method by solving the optimization problem caused by eigen-decomposition with neural networks.
- Instead of using the Euclidean distance like traditional MvSL approaches did, we adopt the Siamese network [46] to better characterize the local manifold structure.
- Unlike traditional MvSL approaches that explicitly optimize the objective function to achieve orthogonality, our MvLNet reformulates the orthogonality constraint as an independent layer which could be plugged into existing neural networks. Moreover, to train the proposed network efficiently, we propose an optimization algorithm (i.e., AOE) which works in a coordinate descent fashion.
- Extensive experiments on three different tasks demonstrate the effectiveness of the proposed model compared to the state-of-the-art unsupervised methods.

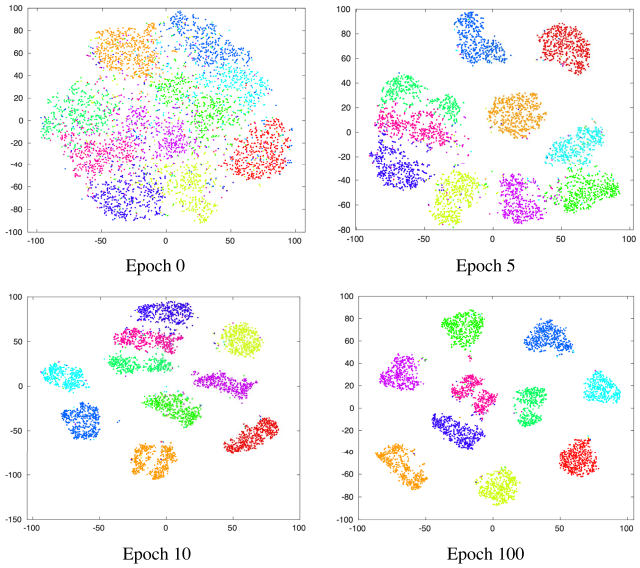


Fig. 1. The visualization on **Noisy MNIST** w.r.t. increasing training epoch, where t-SNE is used to visualize our learned representation. Different colors denote different digits. As shown, our method separates the data into different clusters with growing training epochs, while converging quickly.

Notations: For ease of presentation, we first define the used mathematical notations through this paper. In brief, the lower-case letters denote scalars, the lower-case bold letters denote vectors, and the upper-case bold ones denote matrices. Note that, $\mathbf{I}$ denotes the identity matrix. And for a given matrix $\mathbf{A}$, $Tr(\mathbf{A})$ denotes the trace of $\mathbf{A}$.

II. Related Works

In this section, we briefly review some works in multi-view representation learning, including multi-view spectral representation methods and deep approaches proposed in recent years.

A. Multi-View Spectral Representation

Firstly we give a brief introduction to one pioneer work on representation learning, i.e., spectral representation or called

Laplacian Eigenmaps (LE) [6]. For a given dataset $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ distributed over c classes, LE first builds an affinity matrix or called similarity graph of which each vertex represents a data point and any two data points are connected in the graph i.i.f. one of them is among k nearest neighbors of the other. To construct the affinity matrix $\mathbf{W}$, there are numerous methods are proposed during past decade, e.g. sparsity [31], [48], [49], low-rankness [3], [50], [51], denseness [52], [53], and so on. Among these choices, the vanilla LE method adopts the Euclidean distance with the Gaussian kernel as below:

$$W_{ij} = \begin{cases} \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}), & \mathbf{x}_i, \mathbf{x}_j \text{ are connected.} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

where $W_{ij} \in \mathbf{W}$ is the connection weight between the i-th and the j-th data point.

With the precomputed $\mathbf{W}$, the objective function of SC is defined by:

$$\begin{aligned} \arg\min_{\mathbf{Y}} \quad & Tr(\mathbf{Y}^\top \mathbf{L}\mathbf{Y}) \\ s.t. \quad & \mathbf{Y}^\top\mathbf{Y} = \mathbf{I} \end{aligned} \tag{2}$$

where $\mathbf{Y}$ is the final representation, $\mathbf{L}$ is a Laplacian matrix defined by $\mathbf{L} = \mathbf{D} - \mathbf{W}$, $\mathbf{D}$ is a diagonal matrix defined by $D_{ii} = \sum_j W_{ij}$. The optimal solution to Eq. 2 consists of c eigenvectors corresponding to c smallest eigenvalues of $\mathbf{L}$.

For ease of presentation, let $\{\mathbf{X}^{(\upsilon)}\}_{\upsilon=1}^m$ ($m \geq 2$) be the dataset consisting of υ views. For example, $\mathbf{x}_i^{(1)}$ and $\mathbf{x}_i^{(2)}$ denote the same object $\mathbf{x}_i$ in the first and second views. A general formulation of MvSL can be found in [28], [54] as below,

$$\begin{aligned} \arg\min_{\mathbf{Y}, a^{(\upsilon)}} \quad & \sum_{\upsilon}^{m} (a^{(\upsilon)})^r Tr(\mathbf{Y}^\top \mathbf{L}^{(\upsilon)} \mathbf{Y}) \\ s.t. \quad & \mathbf{Y}^\top\mathbf{Y} = \mathbf{I}, \sum_{\upsilon}^{m} a^{(\upsilon)} = 1, a^{(\upsilon)} > 0 \end{aligned} \tag{3}$$

where $a^{(\upsilon)}$ is the non-negative normalized variable for reflecting the contribution/importance of the υ-th view and r is a scalar to control the distribution of different weights on different views. More details could refer to [54], [55].

B. Deep Multi-View Representation Learning

Thanks to the success of DNNs in a variety of applications, some recent works have been devoted to extending the traditional multi-view learning methods to deep ones. Among these works, DCCA incorporates deep neural networks with CCA [29]. For the given data $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$, DCCA uses two neural networks (f_{θ_1}, f_{θ_2}) to extract features for each view and maximizes the canonical correlation between the extracted features as follows:

$$\begin{aligned} \arg\max_{\theta_1,\theta_2,\mathbf{U},\mathbf{V}} \quad & \frac{1}{N} Tr(\mathbf{U}^\top f_{\theta_1}(\mathbf{X}^{(1)}) f_{\theta_2}(\mathbf{X}^{(2)})^\top \mathbf{V}) \\ s.t. \quad & \mathbf{U}^\top (\frac{1}{N} f_{\theta_1}(\mathbf{X}^{(1)}) f_{\theta_1}(\mathbf{X}^{(1)})^\top + r_x \mathbf{I})\mathbf{U} = \mathbf{I} \\ & \mathbf{V}^\top (\frac{1}{N} f_{\theta_2}(\mathbf{X}^{(2)}) f_{\theta_2}(\mathbf{X}^{(2)})^\top + r_y \mathbf{I})\mathbf{V} = \mathbf{I} \\ & \mathbf{u}_i^\top f_{\theta_1}(\mathbf{X}^{(1)}) f_{\theta_2}(\mathbf{X}^{(2)})^\top \mathbf{v}_j = 0, \forall i \neq j, \end{aligned} \tag{4}$$

where $\mathbf{U} = \{\mathbf{u}_1, \cdots, \mathbf{u}_L\}$ and $\mathbf{V} = \{\mathbf{v}_1, \cdots, \mathbf{v}_L\}$ are two CCA directions that maximize the correlation among the neural network outputs, and r_x and r_y are regularization parameters with positive value. Once the network converged, DCCA uses the final projection mapping $\mathbf{U}^\top f_{\theta_1}$ and $\mathbf{V}^\top f_{\theta_2}$ to obtain the final view-specific representations and apply them for the downstream tasks.

Different from DCCA, DCCAE [39] employs an autoencoder rather than a simple feedforward neural network to learn the view-specific representation. The objective function of DCCAE is as follows:

$$\begin{aligned} \arg\min_{\substack{\theta_1,\theta_2,\mathbf{w}_1 \\ \mathbf{w}_2,\mathbf{U},\mathbf{V}}} \quad & \frac{\lambda}{N} \sum_{i=1}^{N} \|\mathbf{x}_i^{(1)} - g_{\mathbf{w}_1}(\mathbf{x}_i^{(1)})\|^2 + \|\mathbf{x}_i^{(2)} - g_{\mathbf{w}_2}(\mathbf{x}_i^{(2)})\|^2 \\ & - \frac{1}{N} Tr(\mathbf{U}^\top f_{\theta_1}(\mathbf{X}^{(1)}) f_{\theta_2}(\mathbf{X}^{(2)})^\top \mathbf{V}) \\ s.t. \quad & \mathbf{U}^\top (\frac{1}{N} f_{\theta_1}(\mathbf{X}^{(1)}) f_{\theta_1}(\mathbf{X}^{(1)})^\top + r_x \mathbf{I})\mathbf{U} = \mathbf{I} \\ & \mathbf{V}^\top (\frac{1}{N} f_{\theta_2}(\mathbf{X}^{(2)}) f_{\theta_2}(\mathbf{X}^{(2)})^\top + r_y \mathbf{I})\mathbf{V} = \mathbf{I} \\ & \mathbf{u}_i^\top f_{\theta_1}(\mathbf{X}^{(1)}) f_{\theta_2}(\mathbf{X}^{(2)})^\top \mathbf{v}_j = 0, \forall i \neq j, \end{aligned} \tag{5}$$

where $\lambda > 0$ is a trade-off parameter. $g_{\mathbf{w}_1}$ and $g_{\mathbf{w}_2}$ are two autoencoders with weights collection $\mathbf{w}_1$ and $\mathbf{w}_2$.

III. Multi-View Laplacian Network

In this section, we propose a deep multi-view learning model, termed *Multi-view Laplacian Network* (MvLNet). Different from the existing multi-view learning methods, MvLNet is a deep MvRL which implements the deep neural networks as a parametric function $f_\theta : \mathbb{R}^d \to \mathbb{R}^c$, where d denotes the data dimension, c is the class number, and θ denotes the parametric model. Once the representations are obtained with the well-trained θ, the final representations are used for the downstream tasks such as clustering, recognition, retrieval, and so on.

A. Multi-View Laplacian Loss

To deeply perform multi-view representation learning, we enforce the within-view local invariance and cross-view consistency with the following objective function:

$$\mathcal{L} = (1-\lambda) \sum_{\upsilon=1}^{m} \mathcal{L}_1^{(\upsilon)} + \lambda \mathcal{L}_2, \tag{6}$$

where the within-view invariance loss $\mathcal{L}_1^{(\upsilon)}$ enforces similar points as close as possible in each single view and the cross-view consistency loss $\mathcal{L}_2$ aims to learn a common space in which the discrepancy among different views is minimized. $\lambda \in [0, 1]$ is a scalar to balance the contribution of these two losses.

In our objective function, $\mathcal{L}_1^{(\upsilon)}$ encapsulates local invariance on manifold based on the widely-used manifold assumption [6] as follows:

$$\mathcal{L}_1^{(\upsilon)} = \frac{1}{n^2} \sum_{i,j}^{n} W_{ij}^{(\upsilon)} \|\mathbf{y}_i^{(\upsilon)} - \mathbf{y}_j^{(\upsilon)}\|_2^2, \tag{7}$$

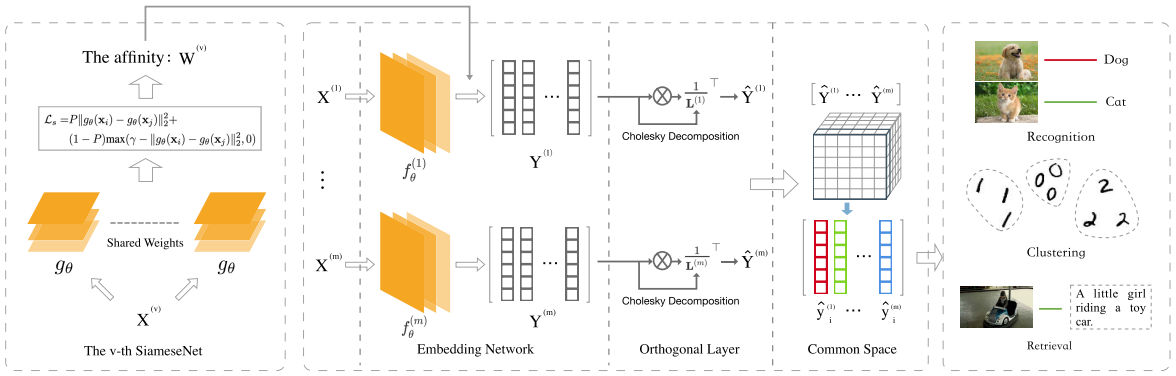


Fig. 2. **The architecture of the proposed MvLNet**. Our model consists m embedding networks $\{f_\theta^{(1)}, \cdots, f_\theta^{(m)}\}$, which output the representations of original data $\{\mathbf{X}^{(i)}\}_{i=1}^m$ from different views. First we train the Siamese network for each view as the left panel showed. After all the Siamese networks are well trained, the affinity matrix $\mathbf{W}$ are used for the embedding network. In order to obtain the orthogonal representations $\{\hat{\mathbf{Y}}^{(i)}\}_{i=1}^m$, each embedding network is followed by an orthogonal layer which performs the QR decomposition to achieve the orthogonality. Once we obtain the representation of the batch data, we compute the objective function and update the network weight using the gradients.

where $W_{ij}^{(v)}$ denotes a precomputed similarity graph and $\mathbf{y}_i^{(v)}$ denotes the output of a neural network w.r.t. the input $\mathbf{x}_i^{(v)}$. Namely, $\mathbf{y}_i^{(v)} = f_\theta^{(v)}(\mathbf{x}_i^{(v)})$, where $f_\theta^{(v)}$ is the v-th sub-network that is used to handle the v-th view. By minimizing the $\sum_v^m \mathcal{L}_1^{(v)}$, the yielded embeddings could remain the local structure of the original data on the subspaces via DNNs.

Regarding $\mathcal{L}_2$, we try to learn the consistency across views by learning a common representation.

$$\mathcal{L}_2 = \frac{1}{nm^2} \sum_{v \neq p}^{m} \sum_{i}^{n} \|\mathbf{y}_i^{(v)} - \mathbf{y}_i^{(p)}\|_2^2, \tag{8}$$

Note that, we do not explicitly learn a common representation that is close to different view-specific representations. Instead, we learn a common space in which the view-specific representations are as close as possible and obtain the final representations. The advantage of such an approach is two-fold. One major advantage is that fewer variables need optimization, thus remarkably decreasing the optimization complexity. The other advantage is close to the downstream task. Taking retrieval as an example, it aims to retrieval one modality from another modality, e.g., to retrieval the images from the description text. Clearly, such a task needs view-specific representations instead of a single common representation which only plays a role in keeping view-consistency.

Although our network with the above objective function could be easily optimized by the back-propagation algorithm, it will lead to a trivial solution that maps all inputs to the same point into the common space, i.e.

$$\mathbf{y}_i^{(v)} = \mathbf{y}, \forall(i, v), \tag{9}$$

It is easy to find that our objective will obtain the minimum of 0 with Eq.9. In other words, all the data points will collapse into the same point $\mathbf{y}$. Clearly, such a solution is undesirable for the downstream tasks. In order to avoid the trivial solution, a constraint is used to orthogonalize all view-specific representation via

$$(\mathbf{Y}^{(v)})^\top \mathbf{Y}^{(v)} = \mathbf{I}_{n \times n}, \tag{10}$$

where $\mathbf{Y}^{(v)}$ is a $n \times d$ matrix in which i-th denotes the $\mathbf{y}_i^{(v)}$. The orthogonal constraint has shown effectiveness in numerous theoretical and experimental studies [6], [47], [56], [57]. This is why we adopt it rather than other relaxed constraints such as low-rankness in our method.

To achieve orthogonality, the orthogonal constraint is usually incorporated into Eq. 6 as a regularization term, which has been widely adopted by plenty of shallow model [3], [31], [53], [55], [56]. However, such a learning paradigm has suffered from two limitations. On the one hand, a new parameter has to be introduced to determine the contribution of the orthogonal term, whose optimal value cannot be determined in a data-driven way. On the other hand, the involved optimization cannot guarantee the strict orthogonality of $\mathbf{Y}^{(v)}$. Thus we choose to achieve the orthogonality by recasting the constraint as an independent layer with the following theorem:

Theorem 1: Given a matrix $\mathbf{A}$ and suppose $\mathbf{A}^\top\mathbf{A}$ is full rank, $\mathbf{Q}$ is an orthogonal matrix which is defined as:

$$\mathbf{Q} = \mathbf{A}(\mathbf{L}^{-1})^\top \tag{11}$$

where $\mathbf{L}$ is obtained by Cholesky decomposition as $\mathbf{A}^\top\mathbf{A} = \mathbf{L}\mathbf{L}^\top$ and $\mathbf{L}$ is a lower triangular matrix.

Proof: For a matrix $\mathbf{A}$ that $\mathbf{A}^\top\mathbf{A}$ is full rank, one could perform the Cholesky decomposition as below:

$$\mathbf{A}^\top\mathbf{A} = \mathbf{L}\mathbf{L}^\top \tag{12}$$

where $\mathbf{L}$ is a lower triangular matrix. Thus $\mathbf{L}^{-1}$ is lower triangular and $(\mathbf{L}^{-1})^\top$ is upper triangular accordingly. For $\mathbf{Q} = \mathbf{A}(\mathbf{L}^{-1})^\top$, it is easy to find that $\mathbf{Q}$ is an orthogonal matrix as

$$\mathbf{Q}^\top\mathbf{Q} = \mathbf{L}^{-1}\mathbf{A}^\top\mathbf{A}(\mathbf{L}^{-1})^\top = \mathbf{L}^{-1}\mathbf{L}\mathbf{L}^\top(\mathbf{L}^{-1})^\top = \mathbf{I}. \tag{13}$$

The proof is complete. □

With Theorem 1, we construct a new layer to implement the orthogonal constraint to enforce the strong orthogonality on the final representations. To be specific, the orthogonal layer first performs Cholesky decomposition on $(\mathbf{Y}^{(v)})^\top\mathbf{Y}^{(v)}$ to obtain $\mathbf{L}^{(v)}$ and then obtains the orthogonal representation via $\hat{\mathbf{Y}}^{(v)} = \mathbf{Y}^{(v)}((\mathbf{L}^{(v)})^{-1})^\top$. Note that, the full rankness of $(\mathbf{Y}^{(v)})^\top\mathbf{Y}^{(v)}$ could be easily guaranteed by adding a sufficiently small number (e.g., 10^{-5}) at the diagonal elements without loss of generality. In addition, the orthogonality aims

at avoiding trivial solution into a batch of samples. In other words, we do not require the orthogonality across different batches during optimizing our neural network.

B. Affinity Learning

In our objective function, $\mathcal{L}_1$ aims to preserve the local invariance on the manifold, where the local invariance is usually formulated as an affinity matrix as introduced in Section II. Most subspace learning methods define the local invariance using Euclidean distance with a Gaussian kernel [56] or self-expressive representation [3], [31], [48], [49], [53], [58]. Although these methods have shown impressive results, they may obtain inferior performance when the data distribution is complex. For example, when the data is insufficient sampling, either of them might not give a desirable affinity matrix as pointed out in [59].

Different from the aforementioned affinity building methods, we employ Siamese network [46] to learn the affinity for each single view. Siamese network is one of the most effective metric learning methods, which has achieved impressive performance in numerous tasks [14], [60]. Given pairs of positive (similar) or negative (dissimilar) samples $(\mathbf{x}_i, \mathbf{x}_j)$, Siamese network learns a parametric model $g_\theta(\cdot)$ by minimizing the distance of positive pairs while maximizing the distance of negative pairs under the help of the ground-truth. Note that, for ease of presentation, we discard the superscript of $\mathbf{x}_i^{(v)}$ in this section, which will not cause misunderstandings. Formally, the objective function of the Siamese network is defined as:

$$\begin{aligned}\mathcal{L}_s = & P\|g_\theta(\mathbf{x}_i) - g_\theta(\mathbf{x}_j)\|_2^2 \\ & +(1-P)\max(\gamma - \|g_\theta(\mathbf{x}_i) - g_\theta(\mathbf{x}_j)\|_2^2, 0),\end{aligned} \tag{14}$$

where the ground-truth $P = 0/1$ if the pair $(\mathbf{x}_i, \mathbf{x}_j)$ is negative/positive, g_θ is a neural network to embed the input $\mathbf{x}_i$ into a latent space, and γ denotes the margin which is fixed to 1.0.

As in the unsupervised settings, the ground-truth is unavailable. To solve this issue, we construct a collection of positive and negative pairs using the k-NN graph. Specifically, $(\mathbf{x}_i, \mathbf{x}_j)$ is a positive pair if $\mathbf{x}_j$ falls into the k-neighborhood of $\mathbf{x}_i$. To construct the negative pairs, we use $\mathbf{x}_i$ and its k non-neighbors that are randomly selected. Note that, the positive pairs and the negative pairs are of equal size. Once the Siamese network convergences, all data points are passed through the network $g_\theta(\cdot)$ and the affinity is computed via

$$W_{ij} = \begin{cases} \exp(-\frac{\|g_\theta(\mathbf{x}_i) - g_\theta(\mathbf{x}_i)\|_2^2}{2\sigma^2}), & \mathbf{x}_i, \mathbf{x}_j \text{ are connected.} \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

C. Alternative Orthogonal and Embedding Training

In this section, we elaborate on the structure and training procedure of our model. As shown in Fig. 3, the proposed MvLNet consists of two stages involving two networks. The first network learns the affinity matrix for each view using a Siamese network. The second one passes the original data of each view into an embedding network and further projects

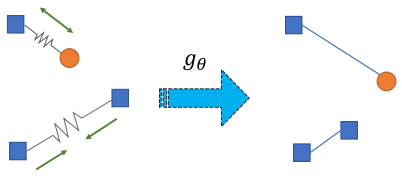


Fig. 3. The Siamese network aims to minimize the distance between the positive pairs (two blue rectangles) and maximizes the distance between negative ones (blue and orange) at the same time.

the view-specific representations into a common space to achieve view-consistency. These two networks are with only one difference in structure. To be specific, the second network replaces the output layer of the first network with a fully connected layer consisting of c neurons followed by the orthogonal layer, where c is the cluster number.

According to Theorem 1, we implement the orthogonal constraint as a pluggable layer, which makes back-propagating the gradient possible. We propose the alternative orthogonal and embedding (AOE) algorithm to train our network in a coordinate descent fashion. More specifically, we alternatively perform the following two optimization steps in each training iteration with different mini-batch sampled uniformly from the training set until our network converges:

- Orthogonal step: We use the Cholesky decomposition as shown in Eq. 11 to obtain the orthogonal representations and update the orthogonal layer weights for each view accordingly.
- Embedding step: We use the standard back-propagation to optimize the parameters of the embedding network.

Once our MvLNet convergences, we forward the original data and obtain the final representations for the downstream tasks. A detailed algorithm of MvLNet is presented in the supplementary material.

D. Discussion

In brief, our method is the deep version of the multi-view spectral clustering. Here we present a detailed discussion on the relationship between the proposed model and some well-known related works. First, compared to the CCA-based approaches (including the shallow version [29] and deep version [12], [39]), our method incorporates the within-view manifold invariance and cross-view consistency, i.e. $\mathcal{L}_1$ and $\mathcal{L}_2$, while the CCA-based approaches aim to maximize the canonical correlation between two views based on the cross-covariance matrix. Second, different from the other multi-view learning methods including the above mentioned, we employ the Siamese network to learn the better affinity, boosting the representation learning performance in various downstream tasks.

IV. Experiments

In this section, we evaluate the performance of the proposed MvLNet for three typical multi-view applications including clustering, classification, and retrieval. To improve the performance and reduce the computational cost, we use an auto-encoder to preprocessing the raw data by reducing the

dimensionality. To investigate the contributions of components of our model, we also report the performance of the following three variants, namely,

- **MvLNet (k-NN)**: it uses the k-NN graph to compute the affinity matrix $\mathbf{W}$ in the raw features. This baseline is used to show the effectiveness of affinity learning based on the Siamese network.
- **MvLNet (w.o. AE)**: it passes the raw data through the embedding network, which is used to show the effectiveness of our autoencoder based learning paradigm.
- **MvLNet** ($\lambda = 0$) discards the $\mathcal{L}_2$ term, which is used to show the importance of our constraint of cross-view consistency.

All the following experiments are implemented using Keras+TensorFlow on a standard Ubuntu-16.04 OS with an NVIDIA 1080Ti GPU. Due to space limitation, the training and network structure detail is presented in the attached supplementary materials.

A. Clustering Performance

Clustering aims to group a set of data points so that the data in the same cluster are as similar as possible, while the data in the different clusters are dissimilar to each other. As multi-view clustering favors a common representation rather than view-specific representations, we obtain the final representation by concatenating all view-specific representations as [31], [53] did. With the final representation, we perform k-means to obtain the clustering assignments.

1) Experimental Setting: For comparison, we evaluate the MvLNet with 10 state-of-the-art clustering methods including three single-view methods and seven multi-view clustering methods. To be exact, the single view methods include spectral clustering (SC) [56] or equivalently LE [6] + k-Means, LRR [3], SNet [60]. The multi-view clustering methods include CCA [29], DCCA [12], DCCAE [39], DiMSC [61], LMSC [53], MvDMF [40], SwMC [62], BMVC [63]. For the single-view methods, we report their results by concatenating the feature vectors of all views.

We carry out experiments on four popular multi-view datasets, namely, Noisy MNIST,[1] Caltech101-20,[2] Reuters[3] and NUS-WIDE-OBJ.[4] To be specific,

- **Noisy MNIST**: The dataset is generated using the MNIST dataset [5] and we adopt the setting used in [39]. Specifically, we use the original dataset as view 1 and randomly select within-class images with additive noise as view 2. Thus, we obtain a binary-view dataset consisting of 70K samples for each view.
- **Caltech101-20**: The dataset consists of 2386 images of 20 subjects selected from the original Caltech101 dataset. We follow the setting used in [40] to extract six handcrafted features as six views, including Gabor feature (48D), Wavelet Moments (40D), CENTRIST feature (254D), HOG feature (1984D), GIST feature (512D), and LBP feature (928D).
- **Reuters**: We use a subset of the Reuters database which consists of the English version and the translations in four different languages, i.e., French, German, Spanish and Italian. The used subset consists of 18,758 samples from six classes.
- **NUS-WIDE-OBJ**: This dataset consists of 30K images distributed over 31 classes. We use five features provided by NUS, i.e., Color Histogram (65D), Color Moments (226D), Color Correlation (145D), Edge Distribution (74D) and wavelet texture (129D).

Note that, we use 10K samples randomly selected from Noisy MNIST, Reuters and NUSWIDEOBJ in experiments since most of the baselines are inefficient to handle large-scale datasets. For a fair comparison, we randomly split the dataset into two partitions of equal size, one partition is used to tune parameters for all the methods and the other partition is used for evaluation. For a comprehensive investigation, we adopt Accuracy (**ACC**), normalized mutual information (**NMI**), adjusted mutual information (**AMI**), and F-measure (**F-mea**) as the performance measurement. A higher value indicates better performance for all metrics.

2) Clustering Results: Tables I– II show the quantitative comparison with 10 state-of-the-art methods on four datasets. Note that, as DCCA/DCCAE can only handle bi-view dataset, we report their performance on the best two views accordingly. From Table I, one could observe that our MvLNet outperforms the other tested methods in terms of all the evaluation metrics. Specifically our model achieves 99.18% on **Noisy MNIST**, which is the best performance to the best of our knowledge. Moreover, our model outperforms the other methods with a large margin on **Caltech101-20**. Similarly, Table II shows that MvLNet outperforms the other methods on **Reuters** and **NUSWIDEOBJ** in most cases and LMSC achieves a competitive result in terms of NMI and AMI. Regarding the ablation studies with MvLNet (k-NN), MvLNet (w.o. AE) and MvLNet ($\lambda = 0$), the proposed model outperforms them in all tests. This demonstrates the effectiveness of the proposed method. Especially, when the data is complex (Caltech101-20, Reuters), MvLNet remarkably outperforms the MvLNet (k-NN), indicating the superiority of affinity learning based on the Siamese network.

B. Recognition Performance

Besides boosting the clustering performance, the representation learned by MvLNet could also be applied to recognition task. Different from clustering, recognition needs label information. To investigate the performance of MvLNet on the recognition task, we evaluate it with the aforementioned baselines on three datasets, i.e., Caltech101-20, Reuters, and NUSWIDEOBJ. Note that, we cannot report the result of SwMC since it does not explicitly learn a representation for this task. In addition, as SC could be regarded as identical with LE + k-means, we replace SC with LE in this experiment.

[1] http://ttic.uchicago.edu/∼wwang5/dccae.html, createMNIST.m
[2] http://www.vision.caltech.edu/Image Datasets/Caltech101/
[3] https://archive.ics.uci.edu/ml/datasets.html
[4] http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm
[5] http://yann.lecun.com/exdb/mnist/

TABLE I

CLUSTERING PERFORMANCE COMPARISON USING NOISY MNIST AND CALTECH101-20 DATASETS

	Noisy MNIST				Caltech101-20			
Methods	ACC (%)	F-mea (%)	NMI (%)	AMI (%)	ACC (%)	F-mea (%)	NMI (%)	AMI (%)
SC (NIPS'02)	66.26	66.42	61.36	61.08	42.50	34.15	62.41	53.63
LRR (TPAMI'13)	56.96	55.06	65.84	64.19	39.15	29.83	59.53	51.09
SNet (ICLR'18)	84.68	82.21	90.14	88.97	51.05	36.91	64.55	57.71
CCA (BS'92)	51.36	50.75	49.57	49.15	23.39	15.57	24.32	17.90
DCCA (ICML'13)	95.50	95.46	89.47	89.42	42.83	37.60	62.03	53.77
DCCAE (ICML'15)	94.92	94.87	88.45	88.40	44.76	38.87	61.19	52.61
DiMSC (CVPR'15)	47.24	50.25	34.84	33.84	21.46	16.59	24.70	17.30
LMSC (CVPR'17)	66.88	66.79	61.94	61.62	38.14	30.06	57.02	47.89
MvDMF (AAAI'17)	75.26	75.00	67.12	66.63	35.96	26.23	47.25	38.89
SwMC (IJCAI'17)	98.98	98.96	97.14	97.12	49.87	35.53	62.32	59.17
BMVC (TPAMI'18)	90.40	85.98	93.47	90.80	36.55	25.70	56.19	48.24
MvLNet (k-NN)	98.12	98.08	95.63	95.60	50.38	42.01	67.10	61.52
MvLNet (w.o. AE)	70.24	65.54	79.99	77.88	53.98	40.59	66.09	59.37
MvLNet (λ = 0)	84.08	81.97	89.77	88.67	45.26	36.06	63.88	56.00
MvLNet	**99.18**	**99.16**	**97.76**	**97.75**	**58.84**	**44.30**	**68.55**	**62.30**

TABLE II

CLUSTERING PERFORMANCE COMPARISON USING REUTERS AND NUSWIDEOBJ DATASETS

	Reuters				NUSWIDEOBJ			
Methods	ACC (%)	F-mea (%)	NMI (%)	AMI (%)	ACC (%)	F-mea (%)	NMI (%)	AMI (%)
SC (NIPS'02)	45.94	38.17	22.26	22.11	15.32	10.33	15.58	12.46
LRR (TPAMI'13)	41.52	27.26	26.37	24.78	13.94	10.73	14.16	10.91
SNet (ICLR'18)	46.64	29.45	24.66	23.63	15.38	11.52	15.19	11.96
CCA (BS'92)	43.48	36.26	16.87	16.18	13.08	8.21	11.01	8.12
DCCA (ICML'13)	29.40	25.54	6.73	10.92	16.00	8.83	11.34	8.19
DCCAE (ICML'15)	30.28	25.21	8.87	19.43	14.76	8.55	11.65	8.72
DiMSC (CVPR'15)	40.50	37.38	13.51	13.24	9.28	7.49	7.53	4.39
LMSC (CVPR'17)	40.06	33.20	**28.89**	**28.46**	15.40	**12.14**	16.30	12.90
MvDMF (AAAI'17)	45.78	24.93	24.69	18.91	12.04	7.49	7.53	9.70
SwMC (IJCAI'17)	32.84	20.59	23.00	14.54	13.84	4.53	9.58	2.25
BMVC (TPAMI'18)	46.96	34.82	22.10	20.96	14.12	9.95	12.57	9.43
MvLNet (k-NN)	31.02	13.58	5.46	2.61	16.08	9.67	15.24	12.52
MvLNet (w.o. AE)	-	-	-	-	16.24	11.13	13.54	12.43
MvLNet (λ = 0)	47.84	40.86	25.14	24.22	13.64	9.42	14.39	11.29
MvLNet	**48.86**	**43.45**	26.75	25.64	**16.56**	12.02	**16.73**	**13.51**

TABLE III

RECOGNITION PERFORMANCE COMPARISON USING CALTECH101-20, REUTERS AND NUSWIDEOBJ DATASETS

	Caltech101-20			Reuters			NUSWIDEOBJ		
Methods	ACC (%)	F-mea (%)	Precision (%)	ACC (%)	F-mea (%)	Precision (%)	ACC (%)	F-mea (%)	Precision (%)
LE (NIPS'02)	84.32	84.01	84.24	74.20	74.32	74.47	20.26	20.48	20.87
LRR (TPAMI'13)	82.90	82.92	83.43	58.00	57.99	58.07	21.26	21.49	21.85
SNet (ICLR'18)	82.06	80.50	80.45	73.44	73.43	73.46	22.52	22.56	23.04
CCA (BS'92)	57.16	56.26	58.82	73.58	74.03	75.15	15.30	15.58	16.08
DCCA (ICML'13)	82.29	81.71	81.79	74.24	74.34	74.59	16.62	16.59	16.62
DCCAE (ICML'15)	83.23	82.69	82.97	74.06	74.14	74.32	14.90	15.02	15.19
DiMSC (CVPR'15)	58.00	56.61	56.15	50.48	50.45	50.49	13.16	13.25	13.55
LMSC (CVPR'17)	83.57	83.00	82.80	58.48	58.41	58.43	22.68	22.61	22.90
MvDMF (AAAI'17)	66.72	68.81	73.19	40.56	40.79	41.67	7.42	7.06	8.53
BMVC (TPAMI'18)	83.65	83.55	83.72	71.62	71.60	71.77	12.86	12.93	13.14
MvLNet (k-NN)	83.65	82.05	82.17	50.04	49.73	50.38	22.18	22.36	23.65
MvLNet (w.o. AE)	83.24	82.07	83.06	-	-	-	22.28	22.37	22.76
MvLNet (λ = 0)	83.15	82.20	83.13	74.62	74.67	74.75	21.94	22.17	22.98
MvLNet	**84.49**	**83.57**	**84.29**	**75.50**	**75.50**	**75.53**	**23.14**	**23.19**	**23.63**

1) Experimental Setting: Similar to the experiment setting in the clustering task, we only use 10K samples of Reuters and NUSWIDEOBJ in this evaluation. In each test, we split the dataset into two partitions of equal size, one for training and one for testing. For all methods, we first obtain the representations with them and then perform the nearest neighbor classifier (1-NN classifier) to obtain the classification result. In the experiments, we adopt the accuracy (ACC), F-measure (F-mea), and precision for evaluation.

2) Recognition Results: Table III reports the recognition results on the three datasets. One could observe that the proposed MvLNet outperforms the tested methods on all the datasets in terms of all the evaluation metrics. The results demonstrate the effectiveness of MvLNet in the

TABLE IV
RETRIEVAL PERFORMANCE COMPARISON USING WIKIPEDIA, PASCAL AND NUSWIDE-10K DATASETS IN TERMS OF MAP SCORE

Methods	Wikipedia			Pascal			NUSWIDE-10K		
	Image - Text	Text - Image	AVG	Image - Text	Text - Image	AVG	Image - Text	Text - Image	AVG
CCA (BS'92)	0.178	0.176	0.177	0.110	0.116	0.113	0.159	0.189	0.174
CFA (ICME'03)	0.306	0.330	0.318	0.341	0.308	0.325	0.299	0.301	0.300
KCCA (NC'04)	0.328	0.357	0.382	0.312	0.329	0.321	0.295	0.162	0.229
DCCA (ICML'13)	0.355	0.409	0.382	0.312	0.311	0.311	0.384	0.382	0.383
DCCAE (ICML'15)	0.344	0.393	0.368	0.456	0.467	0.461	0.360	0.351	0.332
Bimodal AE (ICML'11)	0.267	0.301	0.284	0.404	0.447	0.426	0.234	0.376	0.305
Multi-DBN (ICML'12)	0.145	0.204	0.175	0.438	0.363	0.401	0.178	0.144	0.161
Corr-AE (ICM'14)	0.357	0.373	0.365	0.411	0.475	0.443	0.306	0.340	0.323
JRL (TCSVT'14)	0.353	0.408	0.381	0.416	0.377	0.397	0.410	0.444	0.427
LGCFL (TMM'15)	0.360	0.416	0.388	0.381	0.435	0.408	0.408	0.374	0.391
CMDN (IJCAI'16)	0.364	0.409	0.387	0.458	0.444	0.451	0.410	0.450	0.430
Deep-SM (TCYB'17)	0.345	0.458	0.402	0.440	0.414	0.427	0.389	**0.496**	0.443
ACMR (ACM ICM'17)	0.361	0.439	0.400	0.434	0.416	0.425	0.445	0.473	0.459
MvLNet (k-NN)	0.362	0.421	0.391	0.435	0.465	0.450	0.470	0.454	0.462
MvLNet (w.o. AE)	0.168	0.178	0.173	0.483	0.491	0.487	0.473	0.442	0.457
MvLNet ($\lambda = 0$)	0.105	0.131	0.118	0.097	0.121	0.109	0.148	0.111	0.129
MvLNet	**0.387**	**0.444**	**0.415**	**0.497**	**0.507**	**0.502**	**0.477**	0.457	**0.467**

recognition task. Regarding the three alternative baselines (MvLNet(k-NN)), MvLNet(w.o. AE) and MvLNet($\lambda = 0$), the proposed model outperforms them in all cases. Again, the comparison verifies the effectiveness of the proposed model as a whole.

C. Retrieval Performance

A typical application of multi-view analysis is the retrieval task, i.e. retrieving the related samples across different views/modalities based on the learned final view-specific representation. A representative example is to retrieval the expected images from a gallery with a given description text sequence, which needs two views at least. Since MvLNet can learn the view-specific representations, we perform the following retrieval experiment on three wildly used datasets as elaborate later.

1) Experimental Setting: For comparison, we evaluate the MvLNet with 13 state-of-the-art multi-view retrieval methods, including CCA [29], CFA [64], KCCA [30], DCCA [12], DCCAE [39], Bimodal AE [65], Multi-DBN [66], CorrAE [67], JRL [68], LGCFL [69], CMDN [41], Deep-SM [70], ACMR [42], GSPH [71].

In this evaluation, we carry out experiments on three popular multi-view retrieval datasets including Wikipedia [72], Pascal Sentences [73], and NUSWIDE-10K.[6] All of these datasets contain at least two modalities. The datasets are with following details:

- **Wikipedia**: The dataset is generated from the featured articles of Wikipedia. It contains 2,866 image/text pairs of 10 classes. In each pair, the text is descriptions of the corresponding image. The classes are of high-level semantics, such as history and warfare. We split the dataset into three parts by following [43], where the training, testing, and validation set consists of 2,173, 462, and 231 pairs, respectively.
- **Pascal Sentences**: The dataset consists of 1,000 images each with 5 corresponding description sentences. This dataset is categorized into 20 categories. We follow the setting in [43] to split the data into three parts like Wikipedia, i.e., 800 pairs for training, 100 pairs for testing and 100 pairs for validation.
- **NUSWIDE-10K**: This dataset is constructed by randomly selecting 10,000 image/tag pairs from 10 largest classes in NUS-WIDE dataset. Therefore, each class consists of 1,000 pairs. Note that NUS-WIDE only contains the tag description instead of the text description. Similar to other datasets, we split it into three parts, namely, 8,000 pairs for training, 1,000 pairs for testing and 1,000 pairs for validation.

Following the setting in [43], we use the AlexNet pre-trained on ImageNet to extract image features for these three datasets. As a result, the image feature is 4,096D. Regarding the text modality, a 3,000D BoW feature is extracted from the Wikipedia, and a 1,000D BoW feature is extracted from the NUS-WIDE and Pascal Sentences datasets.

To evaluate the performance, we report the mean average precision (MAP) scores of cross-view retrieval tasks for all methods. Note that, all the retrieval results are considered when computing the MAP score.

2) Retrieval Results: Table IV shows the experimental results in terms of the MAP score on all query results. For the Wikipedia dataset, our method gives the best retrieval performance with 0.387 in **image** $\rightarrow$ **text** and 0.444 in **text** $\rightarrow$ **image** with an average performance gain of 0.013. For the Pascal dataset, our method also obtains the best result with 0.497 in **image** $\rightarrow$ **text** and 0.507 in **text** $\rightarrow$ **image** with an average performance margin of 0.041. As for the NUSWIDE-10K dataset, MvLNet again achieves the best result of 0.477 in **image** $\rightarrow$ **text**. As for MvLNet ($\lambda = 0$), it achieves an inferior result because of the specific characteristics of the retrieval task. In brief, it aims to seek the most similar samples in different modalities, rather than separate samples into different classes like classification and clustering. Without $\mathcal{L}_2$, MvLNet will only learn the view-specific representations for each modality and the cross-view information is ignored during learning. As a result, it is impossible to retrieve the

[6]http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm

TABLE V
VIEW-SPECIFIC VS. THE COMMON REPRESENTATION

Dataset	View	ACC (%)	F-mea (%)	NMI (%)
Noisy MNIST	View 1	94.28	94.23	87.60
	View 2	94.72	94.65	87.89
	Multi-view	**99.18**	**99.16**	**97.76**
Caltech101-20	View 1	33.61	25.51	34.74
	View 2	49.79	33.42	52.36
	View 3	45.68	33.13	48.58
	View 4	**59.60**	**45.83**	67.67
	View 5	57.00	42.75	65.53
	View 6	53.06	36.51	60.20
	Multi-view	58.68	42.88	**68.81**
Reuters	View 1	40.94	35.25	18.96
	View 2	42.54	35.34	21.26
	View 3	43.40	38.46	21.26
	View 4	46.80	38.41	22.69
	View 5	39.76	34.89	15.9
	Multi-view	**48.86**	**43.45**	**26.75**
NUSWIDEOBJ	View 1	11.80	9.04	11.23
	View 2	11.92	9.61	11.51
	View 3	11.92	9.54	12.87
	View 4	12.26	9.08	14.22
	View 5	13.38	10.57	14.17
	Multi-view	**16.56**	**12.02**	**16.73**

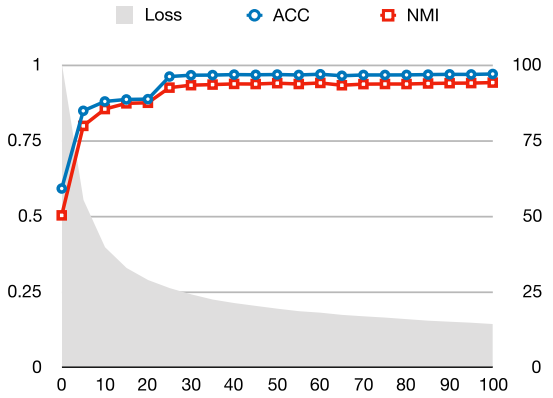


Fig. 4. The influence of training epoch w.r.t. the loss and clustering performance on **Noisy MNIST**, where the left y-axis denotes the normalized loss and the right y-axis corresponds to the clustering performance.

samples across different modalities due to the aforementioned loss of the cross-view consistency.

D. Analysis Experiment

In this section, we conduct some experiments to further investigate the performance of MvLNet. The experiments aim to investigate the view-specific representation concatenating effects in the clustering task and training convergence.

1) View-Specific Representation vs. Common Representation: To investigate the effectiveness of the common representation learned by the adopted concatenating representation, we conduct k-means on each view-specific representation and the common representation given by our method. From Table V, one could find that the common representation learned by our method remarkably outperforms the view-specific case with a large margin on Noisy MNIST, Reuters and NUSWIDEOBJ. In other words, the learned common representation is more discriminative and informative than any view-specific representation, which verifies that our method could explore the complementary information underlying different views.

2) Convergence Analysis: In addition, we also investigate the convergence performance w.r.t. training epochs in Fig. 4. As shown, the loss consistently decreases with more training epochs, which declines quickly in the first 30 epochs. In terms of ACC and NMI, they first quickly increase in the first 30 epochs, and then increase smoothly and slowly.

V. CONCLUSION

In this paper, we proposed a deep unsupervised multi-view representation method, termed as *Multi-view Laplacian Network* (MvLNet). Thanks to the collaboration of the within-view invariance, the cross-view consistency, the non-linear embedding network, and the orthogonal layer, MvLNet could effectively learn a common space to facilitate the performance on the downstream tasks including clustering, recognition, and retrieval. Extensive experiments are conducted on multiple challenging datasets to show the efficacy of MvLNet compared to the state-of-the-art MvRL methods in these three tasks.

ACKNOWLEDGMENT

The authors would like to thank the Associate Editor Prof. Baoxin Li and the anonymous reviewers for their constructive comments and valuable suggestions that help to remarkably improve this paper.

REFERENCES

[1] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
[2] X. Peng, J. Lu, Z. Yi, and R. Yan, "Automatic subspace learning via principal coefficients embedding," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3583–3596, Nov. 2017.
[3] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.
[4] Z. Huang, H. Zhu, J. T. Zhou, and X. Peng, "Multiple marginal Fisher analysis," *IEEE Trans. Ind. Electron.*, vol. 66, no. 12, pp. 9798–9807, Dec. 2019.
[5] S. T. Roweis, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.
[6] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 585–591.
[7] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
[8] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2691–2698.
[9] N. Kulkarni and B. Li, "Discriminative affine sparse codes for image classification," in *Proc. CVPR*, Jun. 2011, pp. 1609–1616.
[10] X. Peng, J. Feng, S. Xiao, W.-Y. Yau, J. T. Zhou, and S. Yang, "Structured autoencoders for subspace clustering," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5076–5086, Oct. 2018.
[11] X. Peng, H. Zhu, J. Feng, C. Shen, H. Zhang, and J. T. Zhou, "Deep clustering with sample-assignment invariance prior," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4857–4868, Nov. 2020.
[12] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *Proc. Int. Conf. Mach. Learn.*, May 2013, pp. 1247–1255.
[13] Z. Ding and Y. Fu, "Robust multi-view subspace learning through dual low-rank decompositions," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2016, pp. 1–7.
[14] J. Hu, J. Lu, and Y.-P. Tan, "Sharable and individual multi-view metric learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 9, pp. 2281–2288, Sep. 2018.

[15] X. Liu, L. Huang, C. Deng, J. Lu, and B. Lang, "Multi-view complementary hash tables for nearest neighbor search," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1107–1115.

[16] X. Liu *et al.*, "Late fusion incomplete multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2410–2423, Oct. 2019.

[17] X. Peng, Z. Huang, J. Lv, H. Zhu, and J. T. Zhou, "COMIC: Multi-view clustering without parameter selection," in *Proc. Int. Conf. Mach. Learn.*, vol. 97, Jun. 2019, pp. 5092–5101.

[18] C. Xu, D. Tao, and C. Xu, "Multi-view intact space learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 12, pp. 2531–2544, Dec. 2015.

[19] Z. Huang, P. Hu, J. T. Zhou, J. Lv, and X. Peng, "Partially view-aligned clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, Dec. 2020, pp. 2892–2902.

[20] M. Yang, Y. Li, Z. Huang, Z. Liu, P. Hu, and X. Peng, "Partially view-aligned representation learning with noise-robust contrastive loss," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021.

[21] C. Deng, Z. Lv, W. Liu, J. Huang, D. Tao, and X. Gao, "Multi-view matrix decomposition: A new scheme for exploring discriminative information," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, Jun. 2015, pp. 3438–3444.

[22] M. Kan, S. Shan, and X. Chen, "Multi-view deep network for cross-view classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4847–4855.

[23] M. Kan, S. Shan, H. Zhang, S. Lao, and X. Chen, "Multi-view discriminant analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 188–194, Jan. 2016.

[24] A. Sharma, A. Kumar, H. Daume, and D. W. Jacobs, "Generalized multiview analysis: A discriminative latent space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2160–2167.

[25] J. Zhang, Y. Peng, and M. Yuan, "SCH-GAN: Semi-supervised cross-modal hashing by generative adversarial network," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 489–502, Feb. 2020.

[26] C. Zhang *et al.*, "Generalized latent multi-view subspace clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 86–99, Jan. 2020.

[27] X. Xu, W. Li, D. Xu, and I. W. Tsang, "Co-labeling for multi-view weakly labeled learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 6, pp. 1113–1125, Jun. 2016.

[28] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," 2013, *arXiv:1304.5634*. [Online]. Available: https://arxiv.org/abs/1304.5634

[29] H. Hotelling, "Relations between two sets of variates," in *Breakthroughs in Statistics*. New York, NY, USA: Springer, 1992, pp. 162–190.

[30] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Comput.*, vol. 16, no. 12, pp. 2639–2664, Dec. 2004.

[31] C. Lu, S. Yan, and Z. Lin, "Convex sparse spectral clustering: Single-view to multi-view," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2833–2843, Jun. 2016.

[32] S. Li, M. Shao, and Y. Fu, "Person re-identification by cross-view multi-level dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2963–2977, Dec. 2018.

[33] Q. Wang, M. Chen, F. Nie, and X. Li, "Detecting coherent groups in crowd scenes by multiview clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 46–58, Jan. 2020.

[34] C. Zhang, H. Fu, Q. Hu, P. Zhu, and X. Cao, "Flexible multi-view dimensionality co-reduction," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 648–659, Feb. 2017.

[35] T. Zhou, C. Zhang, C. Gong, H. Bhaskar, and J. Yang, "Multiview latent space learning with feature redundancy minimization," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1655–1668, Apr. 2020.

[36] C. Xu, D. Tao, and C. Xu, "Multi-view learning with incomplete views," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5812–5825, Dec. 2015.

[37] Z. Huang, J. T. Zhou, X. Peng, C. Zhang, H. Zhu, and J. Lv, "Multi-view spectral clustering network," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 2563–2569.

[38] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[39] Z. Jiao and C. Xu, "Deep multi-view robust representation learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 1083–1092.

[40] H. Zhao, Z. Ding, and Y. Fu, "Multi-view clustering via deep matrix factorization," in *Proc. Conf. AAAI Artif. Intell.*, Feb. 2017, pp. 2921–2927.

[41] Y. Peng, X. Huang, and J. Qi, "Cross-media shared representation by hierarchical learning with multiple deep networks," in *Proc. Int. Joint Conf. Artifi. Intell.*, Jul. 2016, pp. 3846–3853.

[42] B. Wang, Y. Yang, X. Xu, A. Hanjalic, and H. T. Shen, "Adversarial cross-modal retrieval," in *Proc. 25th ACM Int. Conf. Multimedia*, Oct. 2017, pp. 154–162.

[43] X. Huang, Y. Peng, and M. Yuan, "MHTN: Modal-adversarial hybrid transfer network for cross-modal retrieval," *IEEE Trans. Cybern.*, vol. 50, no. 3, pp. 1047–1059, Mar. 2020.

[44] C. Deng, Z. Chen, X. Liu, X. Gao, and D. Tao, "Triplet-based deep hashing network for cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3893–3903, Aug. 2018.

[45] E. Yang, C. Deng, W. Liu, X. Liu, D. Tao, and X. Gao, "Pairwise relationship guided deep hashing for cross-modal retrieval," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2017, pp. 1–8.

[46] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New York, NY, USA, Jun. 2006, pp. 1735–1742.

[47] N. Bansal, X. Chen, and Z. Wang, "Can we gain more from orthogonality regularizations in training deep networks?" in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates, 2018, pp. 4261–4271.

[48] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.

[49] Y. Yang, J. Feng, N. Jojic, J. Yang, and T. S. Huang, "Subspace learning by ℓ^0-induced sparsity," *Int. J. Comput. Vis.*, vol. 126, no. 10, pp. 1138–1156, Jul. 2018.

[50] C. Zhang, H. Fu, S. Liu, G. Liu, and X. Cao, "Low-rank tensor constrained multiview subspace clustering," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1582–1590.

[51] B. Li, R. Liu, J. Cao, J. Zhang, Y.-K. Lai, and X. Liu, "Online low-rank representation learning for joint multi-subspace recovery and clustering," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 335–348, Jan. 2018.

[52] C. Lu, J. Feng, Z. Lin, T. Mei, and S. Yan, "Subspace clustering by block diagonal representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 487–501, Feb. 2019.

[53] C. Zhang, Q. Hu, H. Fu, P. Zhu, and X. Cao, "Latent multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4279–4287.

[54] Y. Li, F. Nie, H. Huang, and J. Huang, "Large-scale multi-view spectral clustering via bipartite graph," in *Proc. Conf. AAAI Artif. Intell.*, Feb. 2015, pp. 2750–2756.

[55] A. Kumar, P. Rai, and H. Daume, "Co-regularized multi-view spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1413–1421.

[56] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 849–856.

[57] Y. Sun, L. Zheng, W. Deng, and S. Wang, "SVDNet for pedestrian retrieval," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3800–3808.

[58] Z. Yang, Q. Li, L. Wenyin, and J. Lv, "Shared multi-view data representation for multi-domain event detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 5, pp. 1243–1256, May 2019.

[59] Z. Ding, S. Ming, and Y. Fu, "Latent low-rank transfer subspace learning for missing modality recognition," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2014, pp. 1–7.

[60] U. Shaham, K. Stanton, H. Li, B. Nadler, R. Basri, and Y. Kluger, "SpectralNet: Spectral clustering using deep neural networks," 2018, *arXiv:1801.01587*. [Online]. Available: https://arxiv.org/abs/1801.01587

[61] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang, "Diversity-induced multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 586–594.

[62] F. Nie, J. Li, and X. Li, "Self-weighted multiview clustering with multiple graphs," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2564–2570.

[63] Z. Zhang, L. Liu, F. Shen, H. T. Shen, and L. Shao, "Binary multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1774–1782, Jul. 2019.

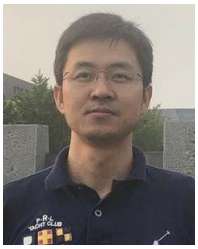
[64] D. Li, N. Dimitrova, M. Li, and I. K. Sethi, "Multimedia content processing through cross-modal association," in *Proc. 11th ACM Int. Conf. Multimedia*, Nov. 2003, pp. 604–611.

[65] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 689–696.
[66] N. Srivastava and R. Salakhutdinov, "Learning representations for multimodal data with deep belief nets," in *Proc. Int. Conf. Mach. Learn.*, vol. 79, 2012, p. 3.
[67] F. Feng, X. Wang, and R. Li, "Cross-modal retrieval with correspondence autoencoder," in *Proc. 22nd ACM Int. Conf. Multimedia*, Nov. 2014, pp. 7–16.
[68] X. Zhai, Y. Peng, and J. Xiao, "Learning cross-media joint representation with sparse and semisupervised regularization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 6, pp. 965–978, Jun. 2014.
[69] C. Kang, S. Xiang, S. Liao, C. Xu, and C. Pan, "Learning consistent feature representation for cross-modal multimedia retrieval," *IEEE Trans. Multimedia*, vol. 17, no. 3, pp. 370–381, Mar. 2015.
[70] Y. Wei *et al.*, "Cross-modal retrieval with CNN visual features: A new baseline," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 449–460, Feb. 2017.
[71] D. Mandal, K. N. Chaudhury, and S. Biswas, "Generalized semantic preserving hashing for N-label cross-modal retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4076–4084.
[72] N. Rasiwasia *et al.*, "A new approach to cross-modal multimedia retrieval," in *Proc. Int. Conf. Multimedia*, Oct. 2010, pp. 251–260.
[73] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier, "Collecting image annotations using Amazon's mechanical Turk," in *Proc. NAACL HLT Workshop Creating Speech Lang. Data Amazon's Mech. Turk*, Jun. 2010, pp. 139–147.

Zhenyu Huang received the bachelor's degree in computer science from Sichuan University in July 2018, where he is currently pursuing the Ph.D. degree with the College of Computer Science. He has published several articles on NeurIPS, ICML, CVPR, IJCAI, and IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS (TIE). His research interests include deep learning and clustering.

Joey Tianyi Zhou is currently a Scientist and a PI with the Institute of High Performance Computing (IHPC), Research Agency for Science, Technology and Research (A*STAR), Singapore. His research interests include transfer learning and sparse coding. He was awarded the NIPS 2017 Best Reviewer Award, the Best Paper Award at the BeyondLabeler Workshop on IJCAI 2016, the Best Paper Nomination at ECCV 2016, and the Best Poster Honorable Mention at ACML 2012.

Hongyuan Zhu (Member, IEEE) received the B.S. degree in software engineering from the University of Macau, Macau, in 2010, and the Ph.D. degree in computer engineering from Nanyang Technological University, Singapore, in 2014. He is currently a Research Scientist with the Institute for Infocomm Research, A*STAR, Singapore. His research interests include multimedia content analysis and segmentation.



Changqing Zhang (Member, IEEE) was a Research Fellow with the University of North Carolina at Chapel Hill (UNC-CH) from March 2017 to March 2018. He is currently an Assistant Professor with the School of Artificial Intelligence, College of Intelligence and Computing, Tianjin University, China. He has published more than 50 articles on journals and conferences, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS), IEEE TRANSACTIONS ON MEDICAL IMAGING (TMI), CVPR, ICCV, AAAI, IJCAI, and ICDM. His research interests include machine learning, computer vision, and medical image analysis.

Jiancheng Lv (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2006. He is currently a Professor with the Data Intelligence and Computing Art Laboratory, College of Computer Science, Sichuan University, Chengdu. His research interests include neural networks, machine learning, and big data.

Xi Peng (Member, IEEE) is currently a Full Professor with the College of Computer Science, Sichuan University. His current research interests include machine learning and data mining. On these areas, he has authored more than 60 articles in the tier 1 conferences and journals. He has served as an associate editor for four journals, such as IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS and a guest editor for three journals, such as IEEE TRANSACTIONS ON NEURAL NETWORK AND LEARNING SYSTEMS.