Autoencoder in Autoencoder Networks

Changqing Zhang, *Member, IEEE*, Yu Geng, Zongbo Han, Yeqing Liu, Huazhu Fu, *Senior Member, IEEE*, and Qinghua Hu, *Senior Member, IEEE*

***Abstract*— Modeling complex correlations on multiview data is still challenging, especially for high-dimensional features with possible noise. To address this issue, we propose a novel unsupervised multiview representation learning (UMRL) algorithm, termed autoencoder in autoencoder networks (AE^2-Nets). The proposed framework effectively encodes information from high-dimensional heterogeneous data into a compact and informative representation with the proposed bidirectional encoding strategy. Specifically, the proposed AE^2-Nets conduct encoding in two directions: the inner-AE-networks extract view-specific intrinsic information (forward encoding), while the outer-AE-networks integrate this view-specific intrinsic information from different views into a latent representation (backward encoding). For the nested architecture, we further provide a probabilistic explanation and extension from hierarchical variational autoencoder. The forward–backward strategy flexibly addresses high-dimensional (noisy) features within each view and encodes complementarity across multiple views in a unified framework. Extensive results on benchmark datasets validate the advantages compared to the state-of-the-art algorithms.**

***Index Terms*— Bidirectional encoding, complete representation, multiview representation learning.**

I. Introduction

MULTIVIEW learning has shown its effectiveness in real-world applications due to the capacity in exploiting the complementarity among these different views. In medical data [9], there are usually multiple types of examinations producing multiview data where they capture different characteristics (e.g., anatomical/functional structures in medical images). For social networks, there are usually relationships (i.e., network) among subjects and identity-specific information (e.g., text or images) [42], [47]. For video surveillance, multimodal sensors provide complementary information for accurate moving person detection [4] and activity recognition [16]. In methodology, multiview learning approaches have attracted intensive attention to jointly exploit multiple modalities or multiple types of features characterizing data from different perspectives [8], [27], [37].

Compared with multiview classification [5], [13] and multiview clustering [6], [21], [50]–[52], unsupervised multiview representation learning (UMRL) tries to integrate heterogeneous views into unified comprehensive representations. Then, the unified representations can be naturally utilized on off-the-shelf methods for downstream tasks. The representative approach for UMRL is canonical correlation analysis (CCA) [15], which searches for linear projections to obtain maximum correlation among different views. In order to deal with the nonlinearity in more general cases, kernelized CCA (KCCA) [1] and deep CCA (DCCA) [2] were proposed to explore nonlinearity with kernel techniques and neural networks, respectively. CCA and its variants both aim to maximize the underlying correlations among different views. From different perspectives, partial least squares (PLS) [35] conducts regression from one view to another, and the multiview dimensionality co-reduction (MDcR) algorithm [53] applies the Hilbert–Schmidt independence criterion to maximize the correlations among different views in the kernel space.

Although UMRL is quite important, it is also very challenging since it is difficult to model complex correlations among different views, especially for the high-dimensional and noisy features. First, existing UMRL methods usually assume that correlations should be maximized, which overemphasizes correlation (consistence) but neglects independence (complementarity). Accordingly, most existing algorithms tend to maximize the underlying correlation [2], [21] among different views. Second, the original data of each view are usually high-dimensional and possibly contain noise. It is critical to effectively extract compact and intrinsic information from each view for integration. Therefore, in this article, we propose a novel UMRL algorithm termed autoencoder in autoencoder networks (AE^2-Nets) to collaboratively encode the intrinsic information from each view into a unified comprehensive representation and is adaptive to the underlying correlation and independence among different views [Fig. 1(a)]. Specifically, the proposed AE^2-Nets conduct encoding in two directions: the inner-AE-networks extract view-specific intrinsic information (forward encoding), while the outer-AE-networks integrate this view-specific intrinsic information from different views into a latent representation (backward encoding). The proposed model can be approximately explained as hierarchical variational autoencoder (VAE), and thus, structured

Manuscript received 24 September 2021; revised 27 January 2022 and 17 April 2022; accepted 30 June 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB2101900; in part by the National Natural Science Foundation of China under Grant 61976151, Grant 61925602, and Grant 61732011; in part by the Natural Science Foundation of Tianjin under Grant 19JCYBJC15200; and in part by the AISG Tech Challenge Funding under Grant AISG2-TC-2021-003. *(Corresponding author: Qinghua Hu.)*

Changqing Zhang and Qinghua Hu are with the Tianjin Key Laboratory of Machine Learning and the College of Intelligence and Computing, Tianjin University, Tianjin 300350, China (e-mail: zhangchangqing@tju.edu.cn; huqinghua@tju.edu.cn).

Yu Geng, Zongbo Han, and Yeqing Liu are with the College of Intelligence and Computing, Tianjin University, Tianjin 300350, China (e-mail: gengyu@tju.edu.cn; hanzongbo.mail@gmail.com; yeqing@tju.edu.cn).

Huazhu Fu is with the Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A*STAR), Singapore 138632 (e-mail: hzfu@ieee.org).

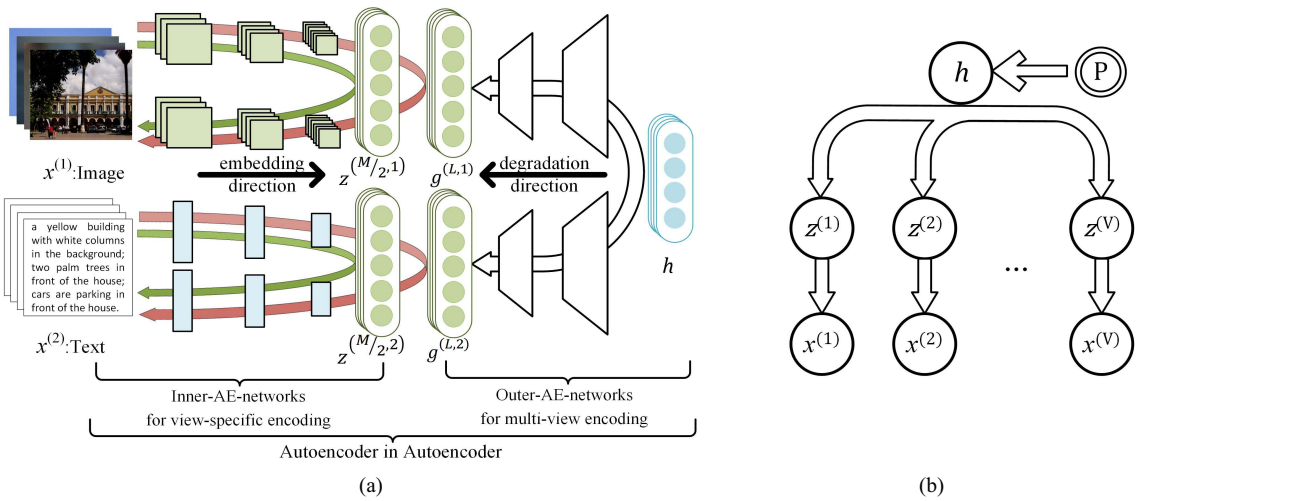

Fig. 1. (a) Overview of AE^2-Nets. The nested autoencoder architecture is composed of inner-AE-networks (shown as the pathway with red arrows) for forward encoding and outer-AE-networks[1] (shown as the pathway with white arrows) for backward encoding. Forward encoding automatically extracts intrinsic information from each view, while backward encoding degrades the intact latent representation to each view through a degradation process. Accordingly, the intrinsic information from multiple views is encoded into the learned intact latent representation. (b) Approximate explanation with probabilistic graphical model. P in double-line circle represents prior on latent representation. See Section III-C for detail.

representation could be achieved with proper prior distribution [Fig. 1(b)]. The main advantages of the proposed algorithm are summarized as follows.

1) A novel UMRL framework—AE^2-Nets—is proposed to deal with multiview heterogeneous data, which can effectively integrate multiple views into a complete representation.
2) The bidirectional learning manner jointly conducts intraview and interview encoding—the inner-AE-networks extract intrinsic information from each view, while the outer-AE-networks integrate this intrinsic information into a complete representation.
3) We provide the theoretical derivation to show that our model is essentially a form of hierarchical VAE, which endows interpretability and flexibility in structuring the unified representations for further performance promotion.
4) The AE^2-Nets can be easily extended for nonnegative multiview representation learning under the framework of deep semi-nonnegative matrix factorization (NMF), bringing performance improvement and interpretability due to the nonnegativity characteristics.
5) Extensive experimental results on diverse benchmark datasets verify the effectiveness of the proposed AE^2-Nets for both classification and clustering tasks.

A preliminary version of this work was published earlier [54]. The present work adds to the initial version in significant ways. First, we extend the AE^2-Nets to nonnegative multiview representation learning. Second, we provide probabilistic explanation and demonstrate that the proposed model is essentially a form of hierarchical VAE. Furthermore, the learned representations are significantly improved with proper prior. For experiment, sufficient ablation studies validate the necessity and advantages of key components.

[1]Note that, since the degradation process (the pathway with white arrows) can also automatically (i.e., without label or other constraint) conduct encoding, i.e., automatically encoding intrinsic information into an intact representation from different views, it is in analogy to autoencoder and termed as outer-AE-Networks.

II. Related Work

Multiview learning has recently attracted intensive attention. For supervised learning, late fusion strategy [17], [29]–[31], [48] trains a classifier on each view and classification is obtained by combiinng these view-specific classifiers. Multimodal metric learning [18], [55], [56] searches multiple metrics for different modalities to maximize the correlations. Hierarchical multimodal metric learning (HM3L) [56] considers the modality-specific and modality-shared metrics; therefore, metric for each modality is a product of these two parts. For nonlinearity in classification, multiview deep network (MvDN) [19] employs unidirectional networks to integrate different views and uses the Rayleigh quotient for discrimination. From probabilistic perspective, the algorithm [46] searches latent representations and metrics from different modalities by using multiwing harmonium (MWH) learning. With specific assumptions, there are theoretical results [7], [10] advocating the effectiveness of integrating multiple views. For multiview clustering, co-regularized [21] and co-training [6], [11], [24], [43], [49] exploit self-representing property among data points on original views and constrain the obtained subspace representations under different assumptions for complementarity.

UMRL [14], [44] is quite challenging since there is no target information to guide the integration of multiple views. The most representative algorithms are CCA-based, which maximize the correlation between two views by projecting original features into low-dimensional space. The CCA extension in kernel space has been widely used for integrating multiview features and dimensionality reduction. DCCA [2] learns a deep neural network for each view. The autoencoder-based

model [27] learns shared representation that requires the network to reconstruct both modalities given only one, which essentially maximizes the consistence between two views. A flexible MDcR method [53] was proposed to search each view independently for the correlations and jointly maximize the dependence among different views in kernel space. The NMF was extended to obtain hierarchical semantics from multiview data [57], where the shared representations for all views are enforced to be the same. Considering the noise inherent in different views, some methods have studied the weights of views to extract the essential information [12], [28], [40].

Autoencoder networks [3], [34] are widely used in unsupervised representation learning (URL), the goal of which is to map the high-dimensional data into a low-dimensional representation, where the original observations can be reconstructed with minimum distortion from the low-dimensional representation. For robustness, stacked denoising autoencoders (SDAEs) [39] are trained to reconstruct a clean input from noisy data generated from a corruption process. Beyond utilizing a Kullback–Leibler (KL) divergence penalty to impose a prior distribution on the latent representation from the autoencoder, the Adversarial AutoEncoder (AAE) [26] employs generative adversarial networks (GANs) (i.e., an adversarial training procedure) for variational inference. Theoretical analysis [22] has been provided to support that autoencoders could be considered as an unsupervised regularizer for improving generalization performance. VAEs [20], [33] learn a parametric generative model by maximizing the marginal log likelihood of training data. Recently, some studies are focused on using VAE for multiview data. The multimodal variational autoencoders (MVAEs) [45] jointly learn posterior in a product of experts (PoE) manner. Furthermore, Shi *et al.* [36] imposed a mixture of experts (MoE) variational posterior over individual modalities to promisingly satisfy some criteria. From the probability perspective, our model can be approximately regarded as a hierarchical multiview VAE, which is different from PoE/MoE obtaining the joint posterior with a product/mixture of individual posteriors in a flat way.

III. Proposed Method

In this section, we present the AE^2-Nets architecture to obtain latent representations from a set of multiview samples $\mathcal{X} = \{\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(V)}\}$, where $\mathbf{X}^{(v)} \in \mathbb{R}^{d_v \times n}$ is feature matrix of the vth view, with v, n, and d_v denote the number of views, the number of samples, and the dimensionality of feature space of the vth view, respectively.

A. Autoencoder in Autoencoder Networks

The goal of our model [shown in Fig. 1(a)] is to obtain an intact space to effectively reveal the underlying structure across multiple views. The AE^2-Nets jointly extract a compact representation encoding intrinsic information for each single view and an intact representation encoding all these intrinsic information as well. To this end, we employ inner-AE-networks to extract intrinsic information from each single view, and outer-AE-networks are regarded as a degradation process to encode complementary intrinsic information into the latent representation. It is noteworthy that the model is endowed with the ability for handling general relationships across different views due to the degradation strategy with neural networks.

The necessities of using autoencoder networks are given as follows. First, there is no supervised information (e.g., class labels) to guide learning; hence, autoencoder networks rather than general neural networks (e.g., for classification) provide a reconstruction constraint in learning. Second, in conventional MRL algorithms, integration processes are usually based on high-dimensional features, which is risky due to redundancy and potential noise involved. The introduced encoding networks can automatically extract intrinsic information for the latent multiview representation, which is superior to the original high-dimensional/noisy features.

The inner-AE-network of the vth view is defined as $f(\mathbf{X}^{(v)}; \mathbf{\Theta}_{\text{ae}}^{(v)})$, where $\mathbf{\Theta}_{\text{ae}}^{(v)} = \{\mathbf{W}_{\text{ae}}^{(m,v)}, \mathbf{b}_{\text{ae}}^{(m,v)}\}_{m=1}^{M}$, which consists of M nonlinear layers ($M+1$ being the total number of layers of each inner-AE-network). Specifically, the first $M/2$ hidden layers (also known as encoder) map an input into an compact representation, and the last $M/2$ layers (also known as decoder) reconstruct the input from the low-dimensional representation. For convenience of description, we use $\mathbf{z}_i^{(0,v)} = \mathbf{x}_i^{(v)} \in \mathbb{R}^{d_v}$ to denote an input vector, and then, the output of the mth layer is

$$\mathbf{z}_i^{(m,v)} = a\big(\mathbf{W}_{\text{ae}}^{(m,v)}\mathbf{z}_i^{(m-1,v)} + \mathbf{b}_{\text{ae}}^{(m,v)}\big), \quad m = 1, 2, \ldots, M \tag{1}$$

where $d_{(m,v)}$ is the number of nodes at the mth layer in the vth view ($\mathbf{z}_i^{(m,v)} \in \mathbb{R}^{d_{(m,v)}}$). $\mathbf{W}_{\text{ae}}^{(m,v)} \in \mathbb{R}^{d_{(m,v)} \times d_{(m-1,v)}}$ and $\mathbf{b}_{\text{ae}}^{(m,v)} \in \mathbb{R}^{d_{(m,v)}}$ denote the weights and bias associated at the mth layer, respectively. $a(\cdot)$ is a nonlinear activation function. Then, given the feature matrix $\mathbf{X}^{(v)} = [\mathbf{x}_1^{(v)}, \mathbf{x}_2^{(v)}, \ldots, \mathbf{x}_n^{(v)}] \in \mathbb{R}^{d_v \times n}$ for the vth view, the corresponding reconstructed counterpart is denoted as

$$\mathbf{Z}^{(M,v)} = \big[\mathbf{z}_1^{(M,v)}, \mathbf{z}_2^{(M,v)}, \ldots, \mathbf{z}_n^{(M,v)}\big] \tag{2}$$

where $\mathbf{z}_i^{(M,v)}$ is the reconstructed representation of the ith sample in the vth view. To obtain low-dimensional representations $\mathbf{Z}^{((M/2),v)}$, we minimize the following reconstruction loss:

$$\min_{\{\mathbf{\Theta}_{\text{ae}}^{(v)}\}_{v=1}^{V}} \frac{1}{2}\sum_{v=1}^{V}\big\|\mathbf{X}^{(v)} - \mathbf{Z}^{(M,v)}\big\|_F^2. \tag{3}$$

The inner-AE-networks forwardly extract low-dimensional view-specific representations $\mathbf{Z}^{((M/2),v)}$. Afterward, we target to encode them into intact representations, $\mathbf{H} \in \mathbb{R}^{k \times n}$, where k is the dimensionality of the latent space, to preserve the intrinsic information from different views. To this end, the outer-AE-networks backwardly conduct degradation process from a comprehensive (or intact) common representation to each single view. Fully connected neural networks (FCNNs) are employed to model the degradation process, as shown in Fig. 1(a). Specifically, we map $\mathbf{H}$ onto the view-specific representation $\mathbf{Z}^{((M/2),v)}$ with the degradation networks $g(\mathbf{H}; \mathbf{\Theta}_{\text{dg}}^{(v)})$, where $\mathbf{\Theta}_{\text{dg}}^{(v)} = \{\mathbf{W}_{\text{dg}}^{(l,v)}, \mathbf{b}_{\text{dg}}^{(l,v)}\}_{l=1}^{L}$ with $L+1$ being the number of layers of degradation networks.

Accordingly, we have $\mathbf{G}^{(0,v)} = \mathbf{H}$ as the input of the degradation networks and $\mathbf{G}^{(l,v)} = [\mathbf{g}_1^{(l,v)}, \ldots, \mathbf{g}_n^{(l,v)}]$, with $\mathbf{g}_i^{(l,v)} = a(\mathbf{W}_{\text{dg}}^{(l,v)}\mathbf{g}_i^{(l-1,v)} + \mathbf{b}_{\text{dg}}^{(l,v)})$. Then, the objective of degradation networks is defined as

$$\min_{\left\{\mathbf{\Theta}_{\text{dg}}^{(v)}\right\}_{v=1}^{V}} \frac{1}{2}\sum_{v=1}^{V}\left\|\mathbf{Z}^{\left(\frac{M}{2},v\right)} - \mathbf{G}^{(L,v)}\right\|_F^2. \tag{4}$$

The proposed AE^2-Nets jointly extract the intrinsic information for each view (with the inner-AE-networks) and learn an intact latent representation (with the outer-AE-networks) in a unified framework. The objective of our bidirectional collaborative encoding is induced as

$$\min_{\left\{\mathbf{\Theta}_{\text{ae}}^{(v)},\mathbf{\Theta}_{\text{dg}}^{(v)}\right\}_{v=1}^{V},\mathbf{H}} \frac{1}{2}\sum_{v=1}^{V}\Bigg(\left\|\mathbf{X}^{(v)} - \mathbf{Z}^{(M,v)}\right\|_F^2 + \lambda\left\|\mathbf{Z}^{\left(\frac{M}{2},v\right)} - \mathbf{G}^{(L,v)}\right\|_F^2\Bigg) \tag{5}$$

where the tradeoff factor $\lambda > 0$ balances view-specific reconstruction and cross-view reconstruction (from latent representation to each single view). For all views, $\mathbf{G}^{(L,v)}$s are degraded from intact latent representation $\mathbf{H}$. The proposed model forwardly conducts view-specific encoding and backwardly conducts multiview intact encoding in a seamless way. It is noteworthy that although bidirectional collaborative encoding is an unsupervised MRL algorithm, it is easy to extend it for specific tasks (e.g., classification or clustering). Moreover, the proposed model is not limited to data with only two views.

Remarks: For clarification, we emphasize the following aspects. First, our work is a novel framework for UMRL, where the target is to flexibly integrate multiple heterogeneous views into one intact representation, rather than improvement of the original autoencoder. The autoencoder is only utilized as a unit in our framework. Second, we design the autoencoder in autoencoder structure to jointly learn both view-specific ($\mathbf{Z}^{(M/2,v)}$) and common intact representation ($\mathbf{H}$). This unified structure allows intrinsic information from multiple views to be encoded and consistency and complementarity to be balanced. Furthermore, we provide an overview explanation for autoencoder in autoencoder in terms of objective function in (5). In the degradation networks, $\mathbf{h}$ acts as an input and each $\mathbf{z}^{(M/2,v)}$ is the associated output, i.e., $\mathbf{h}$ degrading (parameterized with degradation networks) into representation of each single view ($\mathbf{z}^{(M/2,v)}$). In this way, the intact representation $\mathbf{h}$ is constrained by each view and, accordingly, can encode the intrinsic information from multiple views. In practice, $\mathbf{h}$ is randomly initialized and iteratively updated.

B. Nonnegative Multiview Representation

Surprisingly, the proposed model can be naturally extended to nonnegative multiview representation learning without explicit constraint. NMF [23] is particularly effective due to the nonnegativity constraint imposed on the factors, which automatically extracts compact features from a set of nonnegative data vectors. Since the data matrix is usually not strictly nonnegative, Semi-NMF imposes nonnegativity constraints only on the second factor, allowing for clustering/classification interpretability. The objective of Semi-NMF is given as

$$\min_{\mathbf{P},\mathbf{H}\geq 0} \|\mathbf{X} - \mathbf{P}\mathbf{H}\|_F^2 \tag{6}$$

where $\mathbf{H} \geq 0$ is the nonnegative representation. Furthermore, to hierarchically learn the hidden representations and exploit the nonlinearity layer-by-layer, the deep Semi-NMF [38] is proposed as

$$\begin{aligned}
&\mathbf{X} \approx \mathbf{P}_1\mathbf{H}_1^+\\
&\mathbf{X} \approx \mathbf{P}_1 a\left(\mathbf{P}_2\mathbf{H}_2^+\right)\\
&\ldots\\
&\mathbf{X} \approx \mathbf{P}_1 a\left(\cdots a\left(\mathbf{P}_L\mathbf{H}_L^+\right)\right)
\end{aligned} \tag{7}$$

where a nonlinear function $a(\cdot)$ can be introduced as $\mathbf{H}_l = a(\mathbf{P}_{l+1}\mathbf{H}_{l+1})$. Since our model targets at handling multiview data, the objective is induced as

$$\begin{aligned}
&\mathbf{Z}^{(M/2,v)} \approx \mathbf{W}_{\text{dg}}^{(1,v)}\left(a\left(\mathbf{W}_{\text{dg}}^{(2,v)}\mathbf{H}_2^+\right)\right)\\
&\quad\ldots\\
&\mathbf{Z}^{(M/2,v)} \approx \mathbf{W}_{\text{dg}}^{(1,v)} a\left(\cdots a\left(\mathbf{W}_{\text{dg}}^{(L,v)}\mathbf{H}_L^+\right)\right).
\end{aligned} \tag{8}$$

The final nonnegative representation $\mathbf{H}_L^+$ is shared with all views, and it is learned directly on the view-specific representation $\mathbf{Z}^{(M/2,v)}$ (from the inner-AE-networks) instead of the original data matrix. Then, the objective is

$$\begin{aligned}
&C = \sum_{v=1}^{V}\left\|\mathbf{Z}^{(M/2,v)} - \mathbf{W}_{\text{dg}}^{(1,v)} a\left(\cdots a\left(\mathbf{W}_{\text{dg}}^{(L,v)}\mathbf{H}_L^+\right)\right)\right\|_F^2\\
&\text{s.t. } \mathbf{H}_l \geq 0.
\end{aligned} \tag{9}$$

To ensure that the constraint of nonnegativity is satisfied, we add one additional layer equipped with rectified linear unit (ReLU). Specifically, we have $\mathbf{H}_L^+ = r(\mathbf{H})$, where the final common representation $\mathbf{H}_L^+$ acts as the output of the first mapping layer with ReLU ($r(\cdot)$) being the activation function. Accordingly, the final representation is guaranteed to be nonnegative. Then, we can choose appropriate activation functions (e.g., sigmoid or ReLU) to ensure that $\mathbf{H}_l \geq 0$, where $l = 1, \ldots, L-1$.

Algorithm 1 Optimization Algorithm of AE^2-Nets

Input: multi-view data $\mathcal{X} = \{\mathbf{X}^{(v)}\}_{v=1}^{V}$, dimensionality k of latent representation $\mathbf{H}$.
Initialize: randomly initialize $\{\mathbf{\Theta}_{ae}^{(v)}, \mathbf{\Theta}_{dg}^{(v)}\}$ and $\mathbf{H}$.
while *not converged* **do**
  for *each of V views* **do**
    update the parameters of inner-AE-networks
  end
  for *each of V views* **do**
    update the parameters of the outer-AE-networks
  end
  update $\mathbf{H}$
end
Output: latent representation $\mathbf{H}$.

C. Probabilistic Perspective of AE^2-Nets

Our AE^2-Nets model can be approximately explained from probability perspective and strengthened with structured representations for more promising performance. As shown in Fig. 1(a), the inner-AE-networks encode $\mathbf{x}^{(v)}$ to obtain view-specific latent vector $\mathbf{z}^{(v)}$ and outer-AE-networks employ the degradation process [i.e., from latent vector $\mathbf{h}$ to $\mathbf{z}^{(v)}$ ($v = 1, 2, \ldots, V$)] in order to integrate information from each view. Under the learning process mentioned above, we have the following observations.

1) The connection between observation $\mathbf{x}^{(v)}$ and latent representation $\mathbf{h}$ is through the view-specific latent embedding $\mathbf{z}^{(v)}$ (hierarchical realtionship).
2) The view-specific latent embedding $\mathbf{z}^{(v)}$ is conditionally independent with each other ($\mathbf{z}^{(w)}$) given the latent embedding $\mathbf{h}$ (outer-AE-networks).
3) The view-specific latent embedding $\mathbf{z}^{(v)}$ is conditioned on the observation $\mathbf{x}^{(v)}$ and vice versa (inner-AE-networks).

From the perspective of probabilistic generative model, the corresponding graphical model is induced, as shown in Fig. 1(b). Accordingly, the traditional autoencoder is replaced with the probabilistic latent variable models—VAEs [20]. The graphical model shows that AE^2-Nets are essentially a form of hierarchical multiview variational autoencoder (HMVAE). Then, the joint probability of observation and latent variables is given as follows:

$$\begin{aligned} p(\mathbf{x}^{(1:V)}, \mathbf{z}^{(1:V)}, \mathbf{h}) &= p(\mathbf{x}^{(1:V)}|\mathbf{z}^{(1:V)}, \mathbf{h})p(\mathbf{z}^{(1:V)}|\mathbf{h})p(\mathbf{h}) \\ &= p(\mathbf{x}^{(1:V)}|\mathbf{z}^{(1:V)})p(\mathbf{z}^{(1:V)}|\mathbf{h})p(\mathbf{h}) \\ &= \prod_{v=1}^{V} p(\mathbf{x}^{(v)}|\mathbf{z}^{(v)})p(\mathbf{z}^{(v)}|\mathbf{h})p(\mathbf{h}). \end{aligned} \tag{10}$$

According to the proposed model, the evidence lower bound (ELBO) of our proposed hierarchical multiview model is induced as

$$\begin{aligned} \text{ELBO}(\mathbf{x}) &= \mathbb{E}_{q(\mathbf{h},\mathbf{z}^{(1:V)}|\mathbf{x}^{(1:V)})}\left[\log \frac{p(\mathbf{h}, \mathbf{z}^{(1:V)}, \mathbf{x}^{(1:V)})}{q(\mathbf{h}, \mathbf{z}^{(1:V)}|\mathbf{x}^{(1:V)})}\right] \\ &= \sum_{v=1}^{V} \mathbb{E}_{q(\mathbf{z}^{(v)}|\mathbf{x}^{(v)})}[\log p(\mathbf{x}^{(v)}|\mathbf{z}^{(v)})] \\ &\quad - \mathbb{E}_{q(\mathbf{h})}\left[\sum_{v=1}^{V} D_{\text{KL}}(q(\mathbf{z}^{(v)}|\mathbf{x}^{(v)})||p(\mathbf{z}^{(v)}|\mathbf{h}))\right] \\ &\quad - D_{\text{KL}}(q(\mathbf{h})||p(\mathbf{h})). \end{aligned} \tag{11}$$

The three terms in (11) could be well explained as follows.

1) The first term can be considered as the data matching term [similar to (3)].
2) The second term is the expectation of KL divergence that measures the distance between view-specific latent embedding $\mathbf{z}^{(v)}$ encoded from $\mathbf{x}^{(v)}$ and $\mathbf{z}^{(v)}$ generated from $\mathbf{h}$. Since it is not our goal to model the distribution of $\mathbf{z}^{(v)}$ for data generation, we implement $D_{\text{KL}}(q(\mathbf{z}^{(v)}|\mathbf{x}^{(v)})||p(\mathbf{z}^{(v)}|\mathbf{h}))$ using the Euclidean distance as (4), which elegantly avoids explicitly modeling distributions. Changing divergence may alter the empirical behavior of the model, but replacing with any strict divergence is still correct [58] (strict divergence is defined as $D(q, p) = 0$ iff $q = p$).
3) The third term enforces the learned latent representation $\mathbf{h}$ to obey the prior distribution $p(\mathbf{h})$, which endows the model with flexibility in incorporating prior. We can choose priors such as Gaussian to endow the model with ability of generation or mixture of Gaussian (MoG) to further boost latent representations (see the ablation study).

D. Complexity

We provide a complexity analysis of the main steps in our model as follows. For convenience of description, the architectures of the networks used in our experiments are specified and stochastic gradient descent (SGD) is used for optimization. Specifically, we consider a set with n samples and a five-layer AE net (inner-AE-networks) with the numbers of node being $[d_{(0,v)}, d_{(1,v)}, d_{(2,v)}, d_{(1,v)}, d_{(0,v)}]$, where $d_{(0,v)} = d_v$ is the dimensionality of the vth view. The computational complexity is basically $O(nd_{(1,v)}d_{(2,v)} + nd_{(0,v)}d_{(1,v)})$ for each iteration in updating the inner-AE-networks. Similarly, the computational complexity for updating a three-layer degradation net with nodes $[k, d_{(3,v)}, d_{(2,v)}]$ is $O(n(d_{(3,v)}k + d_{(2,v)}d_{(3,v)}))$, where k denotes the dimensionality of the latent representation. For updating $\mathbf{H}$, the complexity is $O(nd_{(2,v)}d_{(3,v)} + nkd_{(3,v)})$. Accordingly, under the condition $d_i = \max(\{d_{(i,v)}\}_{v=1}^{V})$ ($i = \{1, 2, 3\}$) and $d = \max(\{d_{(0,v)}\}_{v=1}^{V})$, the complexity of AE^2-Nets is basically $O(nd_1d_2 + ndd_1 + nd_2d_3 + nkd_3)$. Since the dimensionality of a view-specific representation ($\mathbf{z}^{(v)}$) is usually lower than that of the original features, i.e., $d_{(2,v)} < d_{(0,v)}$, the total complexity is $O(ndd_1 + nkd_3 + nd_2d_3)$. The time complexity is basically linear with respect to the number of samples (n) and the dimensionality of the original feature space (d).

IV. Experiments

In this section, we compare the proposed AE^2-Nets with state-of-the-art multiview representation learning methods on real-world datasets with multiple views. Furthermore, sufficient ablation studies are presented to investigate the rationality and effectiveness of each component. The results are evaluated on clustering and classification tasks.

A. Experimental Settings

1) Datasets: We conduct experiments on the following datasets: **handwritten**[2] contains 2000 images of ten classes from digits "0" to "9." Two different types of descriptors, i.e., pix (240 pixel averages in 2 × 3 windows) and fac (216 profile correlations), are used as two views. **Caltech101-7**[3] contains a subset of images from Caltech101. There are seven categories selected with 1474 images: faces, motorbikes, dollar-bill, garfield, snoopy, stop-sign, and windsor-chair.

[2]https://archive.ics.uci.edu/ml/datasets/Multiple+Features
[3]http://www.vision.caltech.edu/Image_Datasets/Caltech101/

The HOG and GIST descriptors are used. **ORL**[4] contains ten different images for each of 40 distinct subjects. **COIL-20**[5] contains 1440 images from 20 object categories. Each image is normalized to 32×32 with 256 gray levels per pixel. For ORL and COIL-20, the intensity of gray level and Gabor descriptors is used. Caltech-UCSD Birds (**CUB**)[6] contains 11 788 bird images associated with text descriptions [32] from 200 different categories. Each image is described with ten sentences, which are formed as a document. A subset of images from ten categories (i.e., Bobolink, Fish_Crow, Shiny_Cowbird, Frigatebird, Pomarine_Jaeger, California_Gull, Tropical_Kingbird, Scott_Oriole, Caspian_Tern, and Downy_Woodpecker) are selected. Then, we extract 1024-D features based on images with GoogLeNet, and 300-D vector features by Doc2Vec in Gensim.[7] **ADNI**[8] consists of 360 samples with magnetic resonance (MR) and positron emission tomography (PET) images, where 93 region of interest (ROI)-based neuroimaging features are extracted for each neuroimage.

2) Compared Methods: We compare our AE^2-Nets with the following methods.

1) *FeatConcate:* This method concatenates different types of features from multiple views, which usually acts as a baseline.
2) *CCA:* It maps multiple types of features into one common space by finding linear combinations of variables with maximum correlation and then concatenates these projected low-dimensional features together [15].
3) *DCCA:* It extends CCA using deep neural networks by maximizing the correlation between the learned representations of two views and concatenates the projected low-dimensional features of multiple views [2].
4) *Deep Canonically Correlated Autoencoder (DCCAE):* It consists of two autoencoders, maximizes the canonical correlation between the representations from these autoencoders, and then combines these projected low-dimensional features together [41].
5) *MDcR:* It applies kernel matching to regularize the dependence across multiple views and projects each view into a low-dimensional space. Then, these projected low-dimensional features are concatenated together [53].
6) *Deep Semi-NMF for MVC (DMF-MVC):* It utilizes a deep structure through semi-NMF to seek a common feature representation with consistent knowledge for multiview data [57].
7) *MVAE:* It models the joint posterior as a PoE over the marginal posteriors to conduct multimodal learning [45].

3) Evaluation Metrics: To comprehensively compare AE^2-Nets with other methods, we adopt four different metrics to evaluate the clustering quality, i.e., accuracy, normalized mutual information (NMI), F-score, and Rand index (RI), where each metric favors a different clustering property. There are different definitions for accuracy when evaluating clustering. In our experiments, accuracy is defined as follows: given a sample $\mathbf{x}_i$, its cluster label and class label (ground truth) are denoted by r_i and s_i, respectively. Then, we have

$$\mathrm{ACC} = \frac{\sum_{i=1}^{n} \delta(s_i, \mathrm{map}(r_i))}{n} \tag{12}$$

where $\delta(x, y) = 1$ when $x = y$; otherwise, $\delta(x, y) = 0$. $\mathrm{map}(r_i)$ is the permutation map function, which maps the cluster labels into class labels. The best map is obtained using the Kuhn–Munkres algorithm. We employ the standard classification accuracy and conduct experiments with four kinds of partitions of gallery and probe sets. For all these metrics, the higher the value, the superior the performance.

After learning the complete representation based on multiple views, we conduct clustering and classification tasks for evaluation. K-means and k-nearest neighbors (kNN) are employed for clustering and classification, respectively. We choose k-means and kNN because they are both simple and can be used based on the Euclidean distance to reflect the quality of representation.

In practice, the fully connected layer with the $\tanh(\cdot)$ activation function is employed for the inner-AE-networks and outer-AE-networks, where the numbers of layers are empirically set as 5 and 3, respectively. To ensure the nonnegativity, the sigmoid activation function is used in nonnegative multiview representation learning. We use the ℓ_2-norm as a parameter regularizer on all networks and the weight decay is empirically set to 0.0001. We select the dimensionality of latent representation $\mathbf{h}$ from $\{50, 100, 150, 200, 250, 300\}$ and tune the tradeoff parameter λ from $\{0.1, 0.2, \ldots, 1.0\}$. For all these compared methods, we use the codes released by authors and tune parameters for best performance. Due to the randomness involved, we test all algorithms 30 times and report the mean performances and standard deviations for each metric.

B. Results on Clustering

Detailed clustering results for each method are shown in Table I. As can be seen, our algorithm outperforms all the other methods, on all datasets, in terms of ACC. Taking the results on handwritten and Caltech101 for example, our improvements over the second best performers are about 5% and 4%, respectively. The original CCA method only seeks linear projections, which generally performs poorly. As expected, benefitting from nonlinearity, DCCA and DCCAE achieve much better performances than CCA on five out of six datasets, which also demonstrates the rationality behind our algorithm's modeling at complex correlations based on neural networks instead of in a linear way. Our performance is also better than that of DCCAE, which also employs autoencoders to learn view-specific representations. This could be due to the difficulty inherent in balancing complementarity and consistency among different views by simply maximizing the canonical correlation between the view-specific representations. Moreover, although DCCAE and MDcR perform favorably on Caltech101 and handwritten, respectively, they do not show

[4]https://www.cl.cam.ac.U.K./research/dtg/attarchive/facedatabase.html
[5]http://www.cs.columbia.edu/CAVE/software/softlib/
[6]http://www.vision.caltech.edu/visipedia/CUB-200.html
[7]https://radimrehurek.com/gensim/models/doc2vec.html
[8]http://adni.loni.usc.edu/

TABLE I
PERFORMANCE COMPARISON ON CLUSTERING TASK

dataset	method	ACC	NMI	F_score	RI
handwritten	FeatConcate	76.04 ± 2.28	75.70 ± 1.44	70.96 ± 2.05	93.93 ± 0.42
	CCA [15]	66.43 ± 7.62	69.62 ± 6.06	62.05 ± 7.70	91.83 ± 1.79
	DCCA [2]	66.26 ± 0.16	66.01 ± 0.45	59.05 ± 0.39	91.39 ± 0.06
	DCCAE [41]	69.17 ± 1.02	66.96 ± 0.91	60.50 ± 1.10	91.77 ± 0.21
	MDcR [53]	76.72 ± 2.77	76.68 ± 0.93	71.93 ± 2.22	94.11 ± 0.48
	DMF-MVC [57]	71.86 ± 4.25	73.09 ± 3.23	66.66 ± 4.69	92.85 ± 1.13
	AE^2-Nets	**88.33 ± 4.94**	**81.92 ± 2.11**	**81.85 ± 3.01**	**96.15 ± 0.78**
Caltech101	FeatConcate	47.23 ± 0.22	57.19 ± 0.61	52.15 ± 0.28	73.45 ± 0.16
	CCA [15]	45.37 ± 0.09	50.53 ± 0.03	52.15 ± 0.19	73.27 ± 0.09
	DCCA [2]	56.71±10.50	57.61 ± 6.78	62.32±12.75	76.34 ± 6.86
	DCCAE [41]	62.11 ± 2.78	**64.38 ± 4.11**	65.43 ± 4.24	79.31 ± 2.06
	MDcR [53]	46.51 ± 0.67	56.43 ± 0.56	51.55 ± 0.56	73.27 ± 0.30
	DMF-MVC [57]	55.75 ± 5.67	45.52 ± 2.28	55.67 ± 5.50	73.43 ± 2.33
	AE^2-Nets	**66.46 ± 4.55**	60.60 ± 1.93	**73.42 ± 4.91**	**83.14 ± 2.33**
ORL	FeatConcate	61.10 ± 1.51	79.28 ± 0.70	47.03 ± 2.21	97.10 ± 0.25
	CCA [15]	56.98 ± 2.06	76.03 ± 0.79	45.13 ± 1.83	97.32 ± 0.09
	DCCA [2]	59.68 ± 2.04	77.84 ± 0.83	47.72 ± 2.05	97.42 ± 0.13
	DCCAE [41]	59.40 ± 2.20	77.52 ± 0.86	46.71 ± 2.22	97.39 ± 0.14
	MDcR [53]	61.70 ± 2.19	79.45 ± 1.20	48.48 ± 2.59	97.28 ± 0.22
	DMF-MVC [57]	65.38 ± 2.86	82.87 ± 1.26	52.01 ± 3.43	97.29 ± 0.30
	AE^2-Nets	**68.85 ± 2.11**	**85.73 ± 0.78**	**59.93 ± 1.31**	**97.94 ± 0.11**
COIL20	FeatConcate	67.13 ± 4.09	79.94 ± 1.69	64.81 ± 4.05	96.24 ± 0.60
	CCA [15]	58.68 ± 1.34	70.64 ± 0.47	53.13 ± 0.90	95.18 ± 0.10
	DCCA [2]	63.73 ± 0.78	76.02 ± 0.50	58.76 ± 0.53	95.60 ± 0.06
	DCCAE [41]	62.72 ± 1.40	76.32 ± 0.66	57.56 ± 1.15	95.27 ± 0.30
	MDcR [53]	64.25 ± 2.98	79.44 ± 1.37	63.60 ± 2.57	96.11 ± 0.29
	DMF-MVC [57]	53.92 ± 5.89	72.36 ± 2.11	46.39 ± 4.97	92.56 ± 1.46
	AE^2-Nets	**73.42 ± 1.90**	**82.55 ± 1.03**	**69.38 ± 1.92**	**96.86 ± 0.22**
CUB	FeatConcate	73.80 ± 0.11	71.49 ± 0.24	61.07 ± 0.18	91.98 ± 0.04
	CCA [15]	45.82 ± 1.58	46.59 ± 0.98	39.93 ± 1.27	87.44 ± 0.31
	DCCA [2]	54.50 ± 0.29	52.53 ± 0.19	45.84 ± 0.31	88.61 ± 0.06
	DCCAE [41]	66.70 ± 1.52	65.76 ± 1.36	58.22 ± 1.18	91.27 ± 0.24
	MDcR [53]	73.68 ± 3.32	74.49 ± 0.75	65.72 ± 1.37	92.75 ± 0.44
	DMF-MVC [57]	37.50 ± 2.45	37.82 ± 2.04	28.95 ± 1.54	85.52 ± 0.26
	AE^2-Nets	**77.75 ± 1.63**	**78.61 ± 1.62**	**70.96 ± 2.63**	**93.92 ± 0.58**
ADNI	FeatConcate	42.86 ± 0.52	6.56 ± 0.22	41.79 ± 0.49	55.09 ± 0.07
	CCA [15]	45.78 ± 2.53	5.53 ± 2.98	43.57 ± 3.01	52.80 ± 2.92
	DCCA [2]	46.22 ± 1.88	3.93 ± 3.02	43.27 ± 2.27	51.60 ± 4.02
	DCCAE [41]	48.28 ± 1.73	2.85 ± 1.40	**46.08 ± 2.52**	49.03 ± 1.88
	MDcR [53]	43.11 ± 0.39	1.50 ± 0.11	37.54 ± 0.14	53.95 ± 0.09
	DMF-MVC [57]	41.44 ± 3.15	2.44 ± 1.62	39.26 ± 2.66	54.14 ± 0.61
	AE^2-Nets	**51.67 ± 0.81**	**10.18 ± 2.14**	45.16 ± 1.41	**55.27 ± 0.66**

superiority on other datasets. In contrast, the good performance of ours is relatively stable.

C. Results on Classification

For classification, we divide each dataset into different proportions of training and test sets, denoted as $G_{\text{train\_ratio}}/P_{\text{test\_ratio}}$, where G and P indicate "gallery set" and "probe set," respectively. Table II shows the comparison results for each $G_{\text{train\_ratio}}/P_{\text{test\_ratio}}$ partition. According to Table II, the proposed AE^2-Nets show superior performance than those of the compared methods. Specifically, AE^2-Nets consistently outperform the comparisons on all datasets with different partitions. When the number of training samples decreases, the decline in performance can be observed. Fortunately, the decline in our performance is less pronounced than some comparisons. Taking the results on Caltech101 for example, the decline of our method is about 2.4% from $G_{80\%}/P_{20\%}$ to $G_{20\%}/P_{80\%}$, compared with 11.3% for DMF-MVC. We observe that FeatConcate can be superior to the CCA-based method in some cases. One possible reason is that overemphasizing the correlation (consistency) may harm the complementarity across different views. The proposed AE^2-Nets flexibly encodes both consistent and complementary information among multiple views, where the superior performance further validates its advantages.

To further verify the effectiveness of AE^2-Net, we apply t-SNE [25] to visualize the feature distribution of the original single view and our learned compact representation. As shown in Fig. 2, compared with the two original views, the learned latent representation is able to better reflect the clustering

TABLE II
PERFORMANCE COMPARISON ON CLASSIFICATION TASK

dataset	method	$G_{80\%}/P_{20\%}$	$G_{70\%}/P_{30\%}$	$G_{50\%}/P_{50\%}$	$G_{20\%}/P_{80\%}$
handwritten	FeatConcate	89.60 ± 1.40	88.97 ± 0.73	88.87 ± 0.44	85.68 ± 0.53
	CCA [15]	93.78 ± 0.82	93.47 ± 0.93	93.28 ± 0.66	91.12 ± 0.74
	DCCA [2]	95.18 ± 0.55	94.62 ± 0.64	94.35 ± 0.46	92.79 ± 0.51
	DCCAE [41]	95.78 ± 0.46	95.10 ± 0.64	94.79 ± 0.58	92.63 ± 0.54
	MDcR [53]	92.33 ± 0.73	91.55 ± 0.39	91.41 ± 0.68	88.11 ± 0.61
	DMF-MVC [57]	94.68 ± 0.71	93.72 ± 0.60	93.33 ± 0.46	88.23 ± 0.57
	AE^2-Nets	**96.93 ± 0.71**	**96.55 ± 0.66**	**95.88 ± 0.71**	**93.38 ± 0.49**
Caltech101	FeatConcate	87.88 ± 0.67	87.47 ± 0.56	87.17 ± 0.49	87.10 ± 0.45
	CCA [15]	91.10 ± 0.96	90.07 ± 1.03	89.82 ± 0.49	89.08 ± 0.71
	DCCA [2]	92.12 ± 0.58	91.46 ± 0.70	91.30 ± 0.48	90.73 ± 0.38
	DCCAE [41]	91.58 ± 1.02	90.91 ± 0.75	90.54 ± 0.44	89.44 ± 0.43
	MDcR [53]	90.14 ± 0.74	89.45 ± 0.76	88.95 ± 0.41	88.46 ± 0.35
	DMF-MVC [57]	85.51 ± 1.05	84.67 ± 0.82	81.88 ± 0.73	74.19 ± 0.99
	AE^2-Nets	**93.77 ± 1.35**	**92.98 ± 1.37**	**92.49 ± 0.72**	**91.36 ± 0.69**
ORL	FeatConcate	79.13 ± 2.36	74.58 ± 1.32	68.00 ± 2.23	48.28 ± 2.27
	CCA [15]	77.13 ± 3.96	73.83 ± 4.89	67.95 ± 2.77	49.00 ± 1.84
	DCCA [2]	83.25 ± 2.71	78.92 ± 1.93	71.15 ± 1.86	51.69 ± 1.75
	DCCAE [41]	81.62 ± 2.95	80.00 ± 1.47	72.80 ± 2.04	51.25 ± 1.90
	MDcR [53]	92.00 ± 1.58	90.83 ± 2.08	83.35 ± 1.08	57.38 ± 2.08
	DMF-MVC [57]	93.13 ± 1.21	91.75 ± 1.64	85.45 ± 1.85	56.44 ± 2.50
	AE^2-Nets	**97.88 ± 1.19**	**96.00 ± 2.18**	**92.20 ± 1.18**	**70.16 ± 2.54**
COIL20	FeatConcate	78.50 ± 2.30	76.42 ± 2.33	67.05 ± 2.33	48.69 ± 2.08
	CCA [15]	90.50 ± 1.46	88.64 ± 0.95	86.86 ± 0.76	78.94 ± 0.87
	DCCA [2]	90.96 ± 1.24	90.48 ± 1.56	88.65 ± 0.84	83.35 ± 0.60
	DCCAE [41]	92.54 ± 0.70	91.88 ± 1.44	90.35 ± 0.58	84.11 ± 1.10
	MDcR [53]	91.11 ± 0.80	90.29 ± 1.05	87.63 ± 1.12	79.46 ± 1.39
	DMF-MVC [57]	95.25 ± 1.06	94.76 ± 0.77	92.07 ± 0.61	82.96 ± 1.03
	AE^2-Nets	**96.11 ± 1.10**	**95.55 ± 0.87**	**93.25 ± 0.73**	**88.85 ± 0.72**
CUB	FeatConcate	82.50 ± 3.04	81.50 ± 3.13	80.80 ± 1.41	78.33 ± 0.99
	CCA [15]	63.92 ± 3.14	61.39 ± 2.56	59.07 ± 2.32	53.06 ± 2.12
	DCCA [2]	65.67 ± 2.85	64.83 ± 1.83	62.37 ± 1.58	58.44 ± 2.92
	DCCAE [41]	77.00 ± 2.94	74.56 ± 2.74	72.60 ± 2.52	67.35 ± 3.84
	MDcR [53]	83.08 ± 3.43	82.44 ± 3.08	81.53 ± 1.67	78.58 ± 1.65
	DMF-MVC [57]	60.08 ± 2.79	58.56 ± 2.84	55.30 ± 1.90	49.60 ± 1.38
	AE^2-Nets	**85.83 ± 2.94**	**84.00 ± 1.41**	**82.67 ± 1.41**	**80.17 ± 1.83**
ADNI	FeatConcate	48.33 ± 0.62	47.22 ± 3.98	45.56 ± 3.64	44.24 ± 2.93
	CCA [15]	46.67 ± 7.19	45.93 ± 3.74	44.11 ± 4.54	42.71 ± 5.55
	DCCA [2]	47.50 ± 3.01	45.19 ± 2.21	43.33 ± 4.25	41.60 ± 5.74
	DCCAE [41]	46.94 ± 1.16	44.26 ± 4.87	43.11 ± 2.24	40.49 ± 5.38
	MDcR [53]	**53.89 ± 4.33**	**51.30 ± 1.24**	48.67 ± 1.87	46.63 ± 4.00
	DMF-MVC [57]	42.08 ± 5.60	41.11 ± 5.97	39.11 ± 6.66	38.75 ± 4.47
	AE^2-Nets	51.67 ± 2.67	50.37 ± 3.91	**48.89 ± 3.07**	**46.94 ± 3.25**

structure. Taking the results on handwritten for example, ten clusters (corresponding to digits "0"'–'9") are more easily identified by our model than either single view.

D. Experiments on (Noisy) Large-Scale Data

We conduct experiments on two larger datasets, i.e., MNIST[9] and Fashion-MNIST.[10] MNIST contains a training set of 60 000 samples (handwritten digits), and a test set of 10 000 samples. Each digit has been normalized and centered in a fixed-size image. Fashion-MNIST consists of a training set of 60 000 samples and a test set of 10 000 samples. Each sample is a 28×28 grayscale image, associated with a label from one of ten classes. For both sets, each image is equally split into left and right parts as two views. For classification, we randomly select 80% samples within each class as a training set and the left acts as a test set. We repeat 30 times to report the mean and standard deviation. As shown in Table III, our proposed algorithm achieves generally better performances for both clustering and classification. Specifically, on MNIST, the improvements of AE^2-Nets over the second best performers are 8% and 12% on clustering and classification, respectively. It is worth noting that the latent representation $\mathbf{H}$ can be updated on a mini-batch instead of on all data. This enables the proposed AE^2-Nets to handle large data under limited memory.

Furthermore, to test the robustness of our algorithm, we impose different degrees of noise on the two datasets. Specifically, in our experiment, a randomly generated noise matrix is produced by independently sampling elements from a uniform distribution within the range [0, 1]. Then, we multiply

[9]http://yann.lecun.com/exdb/mnist/
[10]https://github.com/zalandoresearch/fashion-mnist

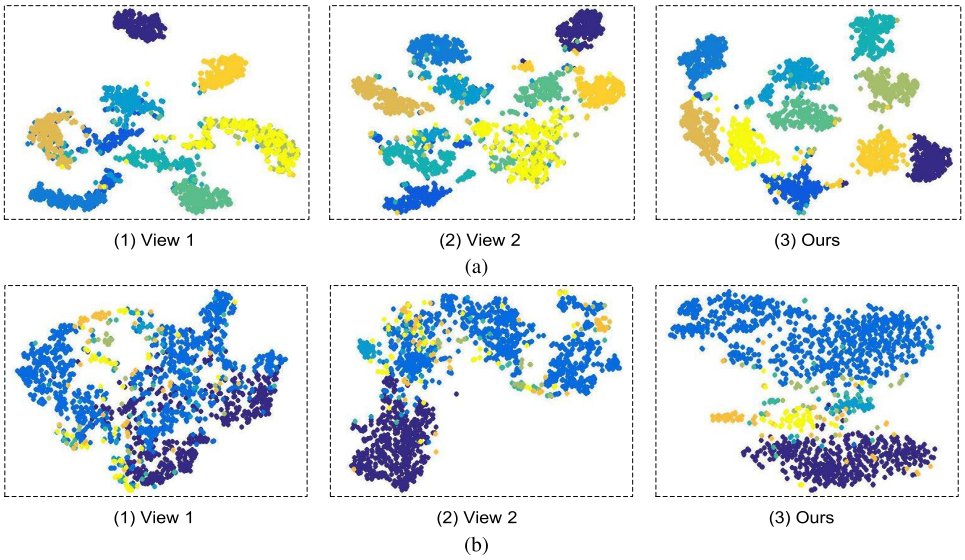

Fig. 2. Visualization of original features for each single view and the latent representation with t-SNE [25]. (a) Handwritten. (b) Caltech101.

TABLE III
PERFORMANCE COMPARISON ON LARGE DATASETS WITH NOISE (OM: OUT OF MEMORY)

dataset	noise (η)	metric	CCA	DCCA	DCCAE	MDcR	DMF-MVC	Ours
MNIST	0	ACC (clustering)	37.55±0.03	43.30±2.61	49.43±0.01	OM	OM	**57.44±2.11**
		ACC (classification)	75.82±0.02	75.58±0.04	75.19±0.01	OM	OM	**87.96±0.02**
	0.1	ACC (clustering)	37.35±0.01	43.59±0.07	40.84±0.38	OM	OM	**51.01±1.49**
		ACC (classification)	72.83±0.04	63.66±0.02	54.08±0.02	OM	OM	**76.28±0.03**
	0.2	ACC (clustering)	35.28±0.02	22.62±0.17	10.46±0.07	OM	OM	**51.08±0.87**
		ACC (classification)	56.39±0.02	23.50±0.02	10.00±0.02	OM	OM	**68.73±0.02**
Fashion-MNIST	0	ACC (clustering)	47.46±0.53	46.48±2.99	47.28±0.87	OM	OM	**53.19±0.50**
		ACC (classification)	64.34±0.00	67.00±0.00	65.81±0.00	OM	OM	**69.93±0.02**
	0.1	ACC (clustering)	47.58±0.94	46.76±1.52	48.96±1.86	OM	OM	**51.79±0.36**
		ACC (classification)	63.31±0.01	65.22±0.01	65.65±0.01	OM	OM	**67.77±0.00**
	0.2	ACC (clustering)	47.29±1.18	44.86±2.01	45.43±2.27	OM	OM	**50.78±0.90**
		ACC (classification)	59.54±0.01	55.40±0.00	56.75±0.01	OM	OM	**65.45±0.01**

the generated noise matrix with a scalar $0 < \eta < 1$ to tune the noise degree. According to the results in Table III, our algorithm performs more robustly with increasing noise. Taking MNIST for example, the declines in the performance of DCCA and DCCAE are quite sharp in terms of clustering accuracy, while the performance of AE^2-Nets is relatively stable. For visualization, we show the original and noisy (noise factor $\eta = 0.2$) images and corresponding reconstruction outputs that can be considered as degraded features from the latent common space. As shown in Fig. 3, the reconstructed images are generally consistent with the original ones, validating our claim that the view-specific representation encoded into the latent representation can well preserve the intrinsic information of the original data. Moreover, even with noise involved, the intrinsic information is still encoded into the learned latent representation, which is able to degrade into the (approximate) view-specific inputs.

The computational times for the different methods are reported in Table IV. All the methods are tested on a computer with 1 GeForce GTX TITAN Xs, except for CCA, which is not implemented on a GPU. For CCA, DCCA, and DCCAE, we use the published codes [41] in MATLAB. Our model is built on TensorFlow in Python. According to the results,

TABLE IV
TIME COST (IN SECONDS) COMPARISON ON LARGE DATASETS

method / dataset	CCA	DCCA	DCCAE	Ours
MNIST	3.45	1317.86	3907.64	165.01
Fashion-MNIST	4.82	1927.76	3357.99	322.07

CCA is the fastest due to it being a linear model without neural networks. The time cost of the proposed AE^2-Nets is lower than the compared network-based methods, DCCA and DCCAE. This is possible because maximizing the correlation is more computationally expensive due to the complex matrix operation [e.g., calculation of the covariance matrix and singular value decomposition (SVD)] in each iteration. Therefore, the proposed AE^2-Nets can manipulate large-scale multiview data with both efficient computation and much less memory requirement.

E. Ablation Study

We conduct extensive and detailed ablation studies to investigate the proposed model, which can clarify the effectiveness of each component or strategy.

TABLE V
RESULTS OF ABLATION STUDY WITH CLUSTERING TASK

dataset	method	ACC	NMI	F	RI
handwritten	MVAE [45]	73.30 ± 5.08	72.67 ± 3.15	68.85 ± 4.64	93.14 ± 1.01
	AE+CAT	78.29 ± 2.16	78.14 ± 0.67	76.20 ± 1.41	94.53 ± 0.34
	AE+DG	80.30 ± 4.79	76.50 ± 2.06	74.51 ± 2.75	94.49 ± 0.75
	AE^2-Nets	88.33 ± 4.94	81.92 ± 2.11	81.85 ± 3.01	96.15 ± 0.78
	AE^2+SNMF	89.47 ± 3.54	81.69 ± 1.80	82.05 ± 2.58	96.25 ± 0.61
	with Gaussian	67.19 ± 3.06	53.41 ± 0.70	51.29 ± 1.24	90.03 ± 0.32
	with MoG	**92.15 ± 0.37**	**85.55 ± 0.48**	**85.65 ± 0.60**	**97.06 ± 0.13**
CUB	MVAE [45]	72.06 ± 4.58	70.96 ± 2.25	65.23 ± 2.94	92.39 ± 0.66
	AE+CAT	70.15 ± 2.34	72.78 ± 1.03	66.03 ± 1.38	92.48 ± 0.21
	AE+DG	64.47 ± 3.34	67.16 ± 1.18	59.08 ± 1.67	91.21 ± 0.41
	AE^2-Nets	77.75 ± 1.63	78.61 ± 1.62	70.96 ± 2.63	93.92 ± 0.58
	AE^2+SNMF	79.72 ± 3.88	77.43 ± 1.42	72.65 ± 2.22	93.95 ± 0.58
	with Gaussian	54.55 ± 2.95	49.81 ± 1.21	43.48 ± 1.49	88.17 ± 0.42
	with MoG	**80.40 ± 4.07**	**79.38 ± 1.31**	**74.88 ± 1.63**	**94.32 ± 0.53**

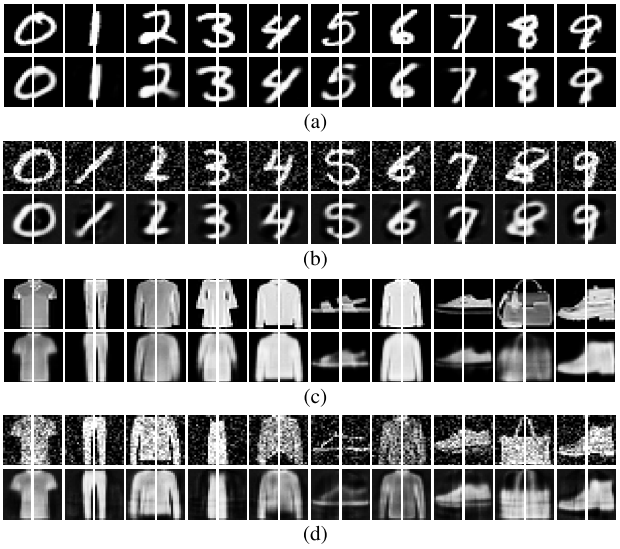

Fig. 3. Visualization of input images (top row) and corresponding reconstruction outputs (bottom row). (a) MNIST. (b) Noisy MNIST. (c) Fashion-MNIST. (d) Noisy Fashion-MNIST.

1) Is the Collaborative Learning More Effective Than AE-Based Methods?: To verify that the proposed integration method is superior to feature concatenation, we extract view-specific compact features with autoencoders and then concatenate them directly. As shown in Table V, our degradation networks comprehensively integrate the information of each view and as expected obtain a better performance than the baseline, i.e., AE + CAT.

2) Is the Proposed Collaborative Learning More Effective Than Two-Step Learning Strategy?: Our model jointly learns view-specific embedding and multiview latent representation. Here, we conduct experiments with a two-step learning strategy, i.e., first using inner-AE-networks to obtain view-specific embeddings and then integrating them with degradation networks. The results in Table V demonstrate that separately learning view-specific and multiview latent representations performs inferior to AE^2-Nets. This validates the effectiveness of joint learning strategy and hierarchical architecture. Beyond elegant probabilistic explanation, intuitively, the collaborative manner not only ensures the view-specific embeddings encode intrinsic information from observations but also ensures them to be sufficiently encoded into the final multiview latent representation.

3) Is Nonnegative Representation More Effective?: The nonnegative representation could be obtained with minor modification (i.e., adding one additional layer with ReLU), and then, we conduct experiments to investigate the effectiveness of nonnegative constraint. According to Table V, the nonnegative multiview representation outperforms the one without nonnegativity constraint in terms of accuracy. Moreover, the proposed nonnegative AE^2-Nets with multiple views generally perform better or competitively compared with that of using single view. This further supports the effectiveness of our integration strategy for multiple views.

4) Is MoG More Effective Than Gaussian?: We have shown that AE^2-Nets are essentially a hierarchical probability model in Section III-C. It is usually helpful to impose proper prior on the latent variable $\mathbf{h}$. We show visualization results under prior of Gaussian distribution and MoG distribution as in Fig. 4. The Gaussian distribution compels different samples to be similar to endow the model with ability of generation, which harms the discriminant property. As shown in Fig. 4, the learned representations are forced to be close to each other under the Gaussian distribution. Therefore, we instead impose MoG prior to address the anticlustering effect and the experimental results are shown in Table V. It is obvious that MoG prior is beneficial to representation learning and obtains the best clustering performance.

5) Comparison Between Different Number of Views: For comparative fairness, we use datasets of two views to adapt to other competitors. Our model is applicable to multiview scenario as well. Here, we further conduct a clustering task on three multiview datasets and compare them with the two-view counterpart, as shown in Table VI. Basically, more views provide more information so that the learned representation can have better performance. However, there may be noisy or redundant information for some views or combinations.

TABLE VI
RESULTS OF MORE VIEWS WITH CLUSTERING TASK

dataset	views	ACC	NMI	F	RI
handwritten	6	90.08 ± 4.75	85.08 ± 2.51	84.54 ± 3.61	96.71 ± 0.89
	2	88.33 ± 4.94	81.92 ± 2.11	81.85 ± 3.01	96.15 ± 0.78
ORL	3	72.73 ± 2.75	86.94 ± 1.32	73.00 ± 2.41	98.10 ± 0.21
	2	70.05 ± 1.90	84.79 ± 0.86	60.41 ± 2.27	98.08 ± 0.13
COIL20	3	72.23 ± 3.38	81.30 ± 1.33	70.43 ± 2.33	96.75 ± 0.36
	2	73.42 ± 1.90	82.55 ± 1.03	69.38 ± 1.92	96.86 ± 0.22

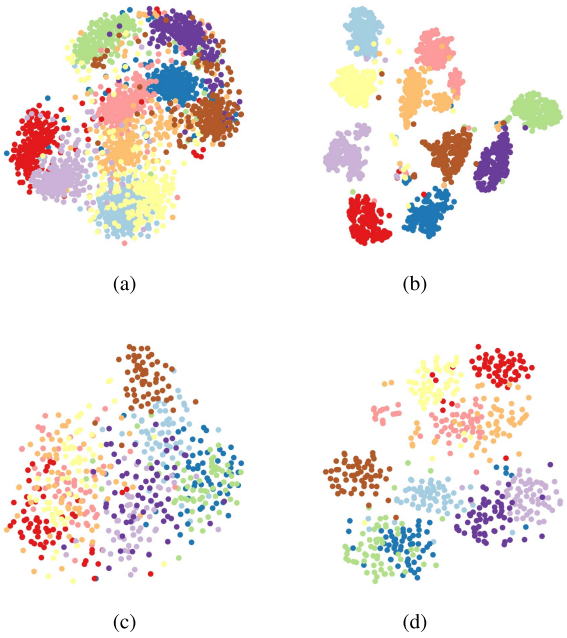

Fig. 4. Comparison between Gaussian and MoG on latent variable **h** on handwritten (top) and CUB (bottom). (a) Gaussian prior with clustering accuracy: 67.19%. (b) MoG prior with clustering accuracy: 92.15%. (c) Gaussian prior with clustering accuracy: 54.55%. (d) MoG prior with clustering accuracy: 80.40%.

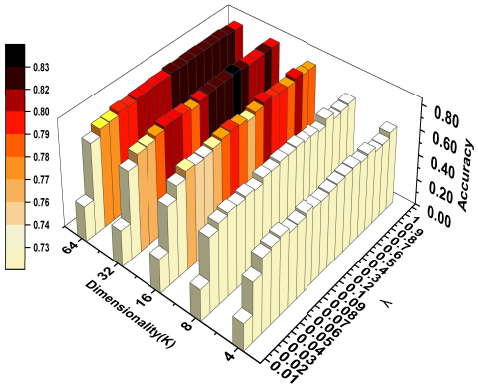

Fig. 5. Parameter tuning on handwritten dataset. We evaluate the representation on clustering task.

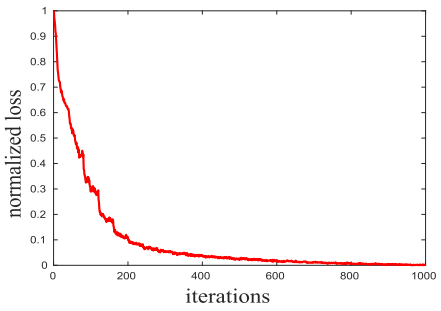

Fig. 6. Convergence curve. The objective value decreases quickly in a few iterations and converges within 1000 iterations.

For example, we note that the two-view version slightly outperforms the three-view counterpart on COIL20.

6) Parameter Analysis and Convergence: There is one essential hyperparameter (λ) in our objective function that balances the view-specific reconstruction and multiview fusion strength. In addition, the dimensionality (K) of the latent representation is another hyperparameter that must be specified in advance. As shown in Fig. 5, we conduct the parameter tuning experiment on the handwritten dataset to show the clustering performance of AE^2-Nets with various values for hyperparameters. When λ is set to zero, the accuracy is relatively low since the latent representation will not be updated at all. While when we increase λ, the performance is improved significantly. As can be observed, promising performance is expected when the value of λ falls within a wide range (e.g., setting $\lambda > 0.5$). For the dimensionality (K), promising performance is expected for small values since a compact latent representation can effectively encode the intrinsic information from different views. To demonstrate the convergence of our optimized algorithm through mini-batch gradient descent, a convergence experiment is conducted, as shown in Fig. 6. Typically, the objective value decreases quickly during the first few iterations and our optimized algorithm converges within 1000 iterations.

V. CONCLUSION AND FUTURE WORK

In this article, we propose a novel UMRL model. In contrast to existing multiview representation learning algorithms, which directly map different views into a common space to maximize linear/nonlinear correlations, the proposed AE^2-Nets architecture jointly performs view-specific embedding and multiview intact representation learning. In this way, the proposed model can flexibly encode intrinsic information from each view. We further provide a principled explanation for the AE^2-Nets from the hierarchical probabilistic perspective and demonstrate its flexibility in learning deep nonnegative

multiview representations. Extensive and detailed experiments validate the effectiveness and clarify the benefits of the proposed model compared with state-of-the-art algorithms. Note that the proposed method is not conducted in a strictly end-to-end manner, where feature vectors are used instead of raw data. Furthermore, our method is not designed for specific modalities. Therefore, suitable network architectures are needed for better applications. In the future work, we aim to adapt the specific network architectures for specific modalities and perform more automatic multiview integration.

REFERENCES

[1] S. Akaho, "A kernel method for canonical correlation analysis," 2006, *arXiv:cs/0609071*.

[2] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *Proc. ICML*, 2013, pp. 1247–1255.

[3] Y. Bengio, P. Lamblin, P. Dan, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. NIPS*, vol. 19, 2007, pp. 153–160.

[4] B. Bhanu and X. Zou, "Moving humans detection based on multi-modal sensor fusion," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshop*, 2004, p. 136.

[5] M. M. Bronstein, A. M. Bronstein, F. Michel, and N. Paragios, "Data fusion through cross-modality metric learning using similarity-sensitive hashing," in *Proc. CVPR*, Jun. 2010, pp. 3594–3601.

[6] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang, "Diversity-induced multi-view subspace clustering," in *Proc. CVPR*, Jun. 2015, pp. 586–594.

[7] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan, "Multi-view clustering via canonical correlation analysis," in *Proc. ICML*, 2009, pp. 129–136.

[8] P. Dhillon, D. P. Foster, and L. H. Ungar, "Multi-view learning of word embeddings via CCA," in *Proc. NIPS*, 2011, pp. 199–207.

[9] J. S. Duncan and N. Ayache, "Medical image analysis: Progress over two decades and the challenges ahead," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 85–106, Jan. 2000.

[10] D. P. Foster, S. M. Kakade, and T. Zhang, "Multi-view dimensionality reduction via canonical correlation analysis," Rutgers Univ., New Brunswick, NJ, USA, Tech. Rep., 2010.

[11] H. Gao, F. Nie, X. Li, and H. Huang, "Multi-view subspace clustering," in *Proc. ICCV*, Dec. 2015, pp. 4238–4246.

[12] Y. Geng, Z. Han, C. Zhang, and Q. Hu, "Uncertainty-aware multi-view representation learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 7545–7553.

[13] K. R. Gray *et al.*, "Random forest-based similarity measures for multi-modal classification of Alzheimer's disease," *NeuroImage*, vol. 65, pp. 167–175, Jan. 2013.

[14] Y. Guo, "Convex subspace representation learning from multi-view data," in *Proc. AAAI*, 2013, pp. 387–393.

[15] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, nos. 3–4, pp. 321–377, 1936.

[16] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang, "Jointly learning heterogeneous features for RGB-D activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5344–5352.

[17] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern Recognit.*, vol. 38, no. 12, pp. 2270–2285, Dec. 2005.

[18] 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.

[19] 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.

[20] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

[21] A. Kumar, P. Rai, and H. Daume, "Co-regularized multi-view spectral clustering," in *Proc. NIPS*, 2011, pp. 1413–1421.

[22] L. Le, A. Patterson, and M. White, "Supervised autoencoders: Improving generalization performance with unsupervised regularizers," in *Proc. NIPS*, 2018, pp. 107–117.

[23] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, p. 788, 1999.

[24] J. Lv, Z. Kang, B. Wang, L. Ji, and Z. Xu, "Multi-view subspace clustering via partition fusion," *Inf. Sci.*, vol. 560, pp. 410–423, Jun. 2021.

[25] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[26] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," 2015, *arXiv:1511.05644*.

[27] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proc. ICML*, 2011, pp. 689–696.

[28] F. Nie, S. Shi, J. Li, and X. Li, "Implicit weight learning for multi-view clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Nov. 1, 2021, doi: 10.1109/TNNLS.2021.3121246.

[29] N. C. Oza and K. Tumer, "Classifier ensembles: Select real-world applications," *Inf. Fusion*, vol. 9, no. 1, pp. 4–20, Jan. 2008.

[30] Y. Peng, X. Zhou, D. Z. Wang, I. Patwa, D. Gong, and C. Fang, "Multimodal ensemble fusion for disambiguation and retrieval," *IEEE Multimedia Mag.*, vol. 23, no. 2, pp. 42–52, Apr./Jun. 2016.

[31] M. P. Perrone and L. N. Cooper, "When networks disagree: Ensemble methods for hybrid neural networks," Inst. Brain Neural Syst., Brown Univ., Providence, RI, USA, Tech. Rep., 1992.

[32] S. Reed, Z. Akata, H. Lee, and B. Schiele, "Learning deep representations of fine-grained visual descriptions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 49–58.

[33] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. 31st Int. Conf. Mach. Learn.*, vol. 32, 2014, p. 1278.

[34] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," Inst. Cogn. Sci., California Univ., San Diego, CA, USA, Tech. Rep., 1985.

[35] A. Sharma and D. W. Jacobs, "Bypassing synthesis: PLS for face recognition with pose, low-resolution and sketch," in *Proc. CVPR*, Jun. 2011, pp. 593–600.

[36] Y. Shi, N. Siddharth, B. Paige, and P. Torr, "Variational mixture-of-experts autoencoders for multi-modal deep generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 15692–15703.

[37] N. Srivastava and R. R. Salakhutdinov, "Multimodal learning with deep Boltzmann machines," in *Proc. NIPS*, 2012, pp. 2222–2230.

[38] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. W. Schuller, "A deep matrix factorization method for learning attribute representations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 3, pp. 417–429, Mar. 2017.

[39] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.

[40] H. Wang, Y. Yang, and B. Liu, "GMC: Graph-based multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1116–1129, Jun. 2019.

[41] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1083–1092.

[42] X. Wang, D. Jin, X. Cao, L. Yang, and W. Zhang, "Semantic community identification in large attribute networks," in *Proc. AAAI*, 2016, pp. 265–271.

[43] Y. Wang, L. Wu, X. Lin, and J. Gao, "Multiview spectral clustering via structured low-rank matrix factorization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4833–4843, Oct. 2018.

[44] M. White, X. Zhang, D. Schuurmans, and Y.-L. Yu, "Convex multi-view subspace learning," in *Proc. NIPS*, 2012, pp. 1673–1681.

[45] M. Wu and N. Goodman, "Multimodal generative models for scalable weakly-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5575–5585.

[46] P. Xie and E. P. Xing, "Multi-modal distance metric learning," in *Proc. IJCAI*. Princeton, NJ, USA: Citeseer, 2013, pp. 1806–1812.

[47] J. Yang, J. McAuley, and J. Leskovec, "Community detection in networks with node attributes," in *Proc. ICDM*, Dec. 2013, pp. 1151–1156.

[48] G. Ye, D. Liu, I.-H. Jhuo, and S.-F. Chang, "Robust late fusion with rank minimization," in *Proc. CVPR*, Jun. 2012, pp. 3021–3028.

[49] M. Yin, J. Gao, S. Xie, and Y. Guo, "Multiview subspace clustering via tensorial t-product representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 851–864, Mar. 2018.

[50] X. Yu, H. Liu, Y. Wu, and H. Ruan, "Kernel-based low-rank tensorized multiview spectral clustering," *Int. J. Intell. Syst.*, vol. 36, no. 2, pp. 757–777, Feb. 2021.

[51] X. Yu, H. Liu, Y. Wu, and C. Zhang, "Fine-grained similarity fusion for multi-view spectral clustering," *Inf. Sci.*, vol. 568, pp. 350–368, Aug. 2021.

[52] 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.
[53] 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.
[54] C. Zhang, Y. Liu, and H. Fu, "AE2-Nets: Autoencoder in autoencoder networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2577–2585.
[55] C. Zhang, Y. Liu, Y. Liu, Q. Hu, X. Liu, and P. Zhu, "FISH-MML: Fisher-HSIC multi-view metric learning," in *Proc. IJCAI*, Jul. 2018, pp. 3054–3060.
[56] H. Zhang, V. M. Patel, and R. Chellappa, "Hierarchical multimodal metric learning for multimodal classification," in *Proc. CVPR*, Jul. 2017, pp. 3057–3065.
[57] H. Zhao, Z. Ding, and Y. Fu, "Multi-view clustering via deep matrix factorization," in *Proc. AAAI*, 2017, pp. 2921–2927.
[58] S. Zhao, J. Song, and S. Ermon, "InfoVAE: Information maximizing variational autoencoders," 2017, *arXiv:1706.02262*.

Authorized licensed use limited to: ASTAR. Downloaded on March 28,2023 at 05:02:43 UTC from IEEE Xplore. Restrictions apply.