A Multimodal Clustering Framework With Cross Reconstruction Autoencoders

QIANGLE ZHAO, LINLIN ZONG, XIANCHAO ZHANG, YUANGANG LI, AND XIAORUI TANG

1. INTRODUCTION

Clustering is a vital research area in unsupervised machine learning. Multimodal clustering, an important field in clustering, partitions the multimodal data into disjoint groups in an unsupervised manner. Multimodal dataset describe objects in multi aspects which are called modals. In each modal, an object has a group of properties which represent the object. Deep neural networks (DNN) [2] breed effective unsupervised tools such as autoencoders [3] and generative adversarial networks (GAN) [4]. Deep clustering methods [5]–[8] burst following the development of the deep unsupervised networks on account of the amazing performance for feature extraction [9] and dimensionality reduction. Deep multimodal clustering [10] utilizes multiple deep neural networks to extract common latent features. Then the final cluster assignment is retrieved by running a cluster method on fused latent features. But due to the high computational costs in high-dimensional data space [13], the DBM based methods have not been widely studied now. Jiquan et al. [14] use an autoencoder to extract one common feature between two modals. However, autoencoders do not really discover the similarity of the common feature distributions. Weiran et al. [15] learn most correlated features in different modalities by deep canonical correlation analysis (DCCA) which merged a deep autoencoder with canonically correlation analysis. Though it maxes the correlation between each pair modals’ latent features, different numerical characteristics of different modalities can end up with indistinct correlation. Moreover, the above methods failed to separate distinctive latent features from common latent features.

In this paper, we propose the cross reconstruction based on the following fact. Each modal has distinctive features that reflects the modal’s characteristics. Meanwhile, all modals share common features in order that they have similar cluster assignment. We highlight the importance of distinctive latent features and common latent features when reconstructing the data object by autoencoders in multimodal datasets. Furthermore, we build a two-step deep multimodal clustering framework with cross reconstruction loss. Specifically, we develop multimodal clustering, but it has not been comprehensively examined. Nitish et al. [10] propose an algorithm which learns a joint representation of different modalities by DBM. But due to the high computational costs in high-dimensional data space [13], the DBM based methods have not been widely studied now. Jiquan et al. [14] use an autoencoder to extract one common feature between two modals. However, autoencoders do not really discover the similarity of the common feature distributions. Weiran et al. [15] learn most correlated features in different modalities by deep canonical correlation analysis (DCCA) which merged a deep autoencoder with canonically correlation analysis. Though it maxes the correlation between each pair modals’ latent features, different numerical characteristics of different modalities can end up with indistinct correlation. Moreover, the above methods failed to separate distinctive latent features from common latent features.

In this paper, We propose the cross reconstruction based on the following fact. Each modal has distinctive features that reflects the modal’s characteristics. Meanwhile, all modals share common features in order that they have similar cluster assignment. We highlight the importance of distinctive latent features and common latent features when reconstructing the data object by autoencoders in multimodal datasets. Furthermore, we build a two-step deep multimodal clustering framework with cross reconstruction loss. Specifically, we develop multimodal clustering, but it has not been comprehensively examined. Nitish et al. [10] propose an algorithm which learns a joint representation of different modalities by DBM. But due to the high computational costs in high-dimensional data space [13], the DBM based methods have not been widely studied now. Jiquan et al. [14] use an autoencoder to extract one common feature between two modals. However, autoencoders do not really discover the similarity of the common feature distributions. Weiran et al. [15] learn most correlated features in different modalities by deep canonical correlation analysis (DCCA) which merged a deep autoencoder with canonically correlation analysis. Though it maxes the correlation between each pair modals’ latent features, different numerical characteristics of different modalities can end up with indistinct correlation. Moreover, the above methods failed to separate distinctive latent features from common latent features.

In this paper, We propose the cross reconstruction based on the following fact. Each modal has distinctive features that reflects the modal’s characteristics. Meanwhile, all modals share common features in order that they have similar cluster assignment. We highlight the importance of distinctive latent features and common latent features when reconstructing the data object by autoencoders in multimodal datasets. Furthermore, we build a two-step deep multimodal clustering framework with cross reconstruction loss. Specifically, we develop multimodal clustering, but it has not been comprehensively examined. Nitish et al. [10] propose an algorithm which learns a joint representation of different modalities by DBM. But due to the high computational costs in high-dimensional data space [13], the DBM based methods have not been widely studied now. Jiquan et al. [14] use an autoencoder to extract one common feature between two modals. However, autoencoders do not really discover the similarity of the common feature distributions. Weiran et al. [15] learn most correlated features in different modalities by deep canonical correlation analysis (DCCA) which merged a deep autoencoder with canonically correlation analysis. Though it maxes the correlation between each pair modals’ latent features, different numerical characteristics of different modalities can end up with indistinct correlation. Moreover, the above methods failed to separate distinctive latent features from common latent features.
a global cross reconstruction (GCR) autoencoder and a local cross reconstruction (LCR) autoencoder as two typical neural networks. The cross reconstruction networks early fuse multimodal features to accurately retrieve latent features.

In the GCR autoencoder, we extract latent features from different modalities with Variational Information Bottleneck (VIB) [16]. VIB controls the amount of information through the autoencoder during the feature extraction by minimizing mutual information between raw data and the extracted features. Meanwhile, we integrate cross reconstruction into the features extraction process, which promotes the distribution similarity of different modalities in latent feature space. We also provide theoretical analysis to prove the similarity promotion of multimodal distributions. Finally, we fuse the extracted features to common features using fusion layers and run a clustering algorithm on the common features. The above process is deep multimodal clustering with global cross reconstruction (DMGCR).

In the LCR autoencoder, we explicitly extract distinctive latent feature and common latent feature for each modal. The reconstruction of a data object in current modal uses the mixture of the distinctive latent feature and the common latent feature. The cross reconstruction utilizes the common latent features of other modalities and the distinctive latent feature of current modal. After the common latent features is trained, we perform clustering algorithm on the fused common latent feature by the same fusion layers as used in DMGCR. Finally, a clustering algorithms is performed on the fused latent features. We call the above LCR based process deep multimodal clustering with local cross reconstruction (DMLCR).

We summarize the contribution of this article as follows:

1. We propose two novel cross reconstruction networks for multimodal data feature extraction, which implement early fusion on all modalities.

2. We provide a theoretical analysis to prove that the proposed cross reconstruction method effectively reduces the distribution difference of different modalities in feature space.

3. We develop a deep clustering framework and two deep multimodal clustering algorithms, DMGCR and DMLCR, which are capable of performing multimodal clustering on comprehensive datasets.

4. Experiments show improvement over state-of-the-art methods on six benchmark datasets. We also explore the parameter selection by grid searching and provide the empirical values suggestion.

II. RELATED WORK

A. DEEP CLUSTERING

Deep clustering receive more attention in the past few years owing to the good performance of deep neural networks on unsupervised feature extraction. Existing methods share two common concepts, feature extraction and clustering assignment. The existing deep clustering methods are roughly divided into two categories: two-stage methods and end-to-end methods [15].

The two-stage methods first extract latent features from the original feature space, then conduct clustering on latent features. Tian et al. [8] uses an autoencoder to extract the features of graph and finally uses k-means to cluster. Chen [17] applies Deep Belief Network (DBN) to extract features and finally uses non-parametric maximum-margin clustering to cluster the features.

The end-to-end methods realize the feature extraction and clustering with unified deep neural networks. The joint unsupervised learning (JULE) algorithm [18] uses a recurrent framework for joint unsupervised learning of deep representations and image clustering, which are optimized jointly in training process. The deep embedding clustering (DEC) algorithm [5] clusters a set of data points in a jointly optimized feature space. Based on DEC, the improved deep embedding clustering (IDEC) algorithm [6] jointly optimizes and preserves local structure of data distribution. These algorithms are limited within single modal clustering tasks. DEPICT [19] develops a convolutional autoencoder for extracting latent features. It uses KL divergence as objective function to guide the training process. VaDE [20] proposes a generative deep clustering algorithm based on the variational autoencoder. It extracts the latent Gaussian distribution parameters instead of latent features. ADEC [21] utilizes adversarial reconstruction to retrieve better latent features for clustering.

B. MULTIMODAL CLUSTERING

We divide the existing multimodal clustering algorithms into two categories according to whether they integrate deep neural networks or not: classical multimodal clustering methods [22]–[26] and deep multimodal clustering methods [10], [14], [15], [27].

Classical multimodal clustering methods either learn a consensus matrix or minimize the divergence of multiple modalities simultaneously. For example, the multi-view spectral clustering (MMSC) algorithm [22] learns a commonly shared graph Laplacian matrix by unifying different modalities. Gao et al. [23] proposes a novel NMF-based multi-view clustering algorithm by searching for a factorization that gives compatible clustering solutions across multimodal. The diversity-induced multi-view subspace clustering (DIMSC) algorithm [24] extends the existing subspace clustering into the multimodal domain. The low-rank tensor constrained multi-view subspace clustering (LT-MSC) algorithm [25] introduces a low-rank tensor constraint to explore the complementary information from multimodal data. The exclusivity-consistency regularized multi-view subspace clustering (ECMSC) algorithm [26] attempts to harness the complementary information between different representations by introducing a novel position-aware exclusivity term.

Deep multimodal clustering methods firstly learn common low-dimensional features from multimodal data, then cluster the features into groups. Ngiam et al. [14] proposes a series of frameworks for deep multimodal learning based on autoencoders. Srivastava and Salakhutdinov [10] proposes a deep multimodal representation learning framework, which learns
a joint representation of different modalities by DBM. The DCCA [27] learns complex nonlinear transformations of two modalities of data such that the resulting representations are highly linearly correlated. The DCCAE [15] add an autoencoder regularization term to DCCA.

III. CROSS RECONSTRUCTION

Multimodal learning tasks mine the consensus information among different modalities. Traditional arts firstly extract latent features within single modality, then fuse the features of each modal into a shared latent feature. We call it Late Fusion. We now introduce the Early Fusion by fusing information among modalities with cross reconstruction when extracting latent features within single modality. Cross reconstruction is biologically inspired by cross-modal systems [28] within the visual and auditory systems of the human brain [29].

A. MULTIMODAL FEATURE EXTRACTION

Consider the problem of clustering a dataset of $n$ points $X = \{x_1, x_2, \ldots, x_n\}$ into $k$ clusters $\{c_1, c_2, \ldots, c_k\}$, each data point $x_i$ contains $M$ modalities $x_i = \{x_i^1, x_i^2, \ldots, x_i^M\}$. These modalities have different dimensions, i.e. $x_i^1 \in \mathbb{R}^{d_1}$, $x_i^2 \in \mathbb{R}^{d_2}$, $\ldots$, $x_i^M \in \mathbb{R}^{d_M}$. Data in the same modality have the same dimensions. Multimodal feature extraction tends to extract latent feature $Z$ from the multimodal data $X$. One of the most well-known tools for deep learning is the autoencoder. Apart from normal autoencoders, variational autoencoders (VAE) are capable of capturing the distribution feature of inputs.

One VAE contains two basic components, an encoder $E$ and a decoder $D$. Both the encoder and the decoder are composed of a series of nonlinear transforms. The encoder $E$ extracts the approximate latent feature distribution $q(z|x_i) \sim N(\mu_i, \sigma_i)$ of the input $p(x_i)$ where $p(\cdot)$ means ground truth distribution and $q(\cdot)$ approximates approximate distribution. Next, a $z_i$ was sampled by $z_i = \mu_i + \varepsilon \cdot \sigma_i$ using the reparameterization skill, where $\varepsilon \sim \mathcal{N}(0, 1)$. The decoder $D$ restores the reconstruction $\hat{x}_i$ of input $x_i$ according to the latent feature $z_i$. The optimization target of VAE is to minimize the difference between the ground truth distribution $p(x, z)$ and the approximate distribution $q(\hat{x}, z)$. That is

$$KL(p(x)p(z|x)||q(z)q(\hat{x}|z)) = \int \int p(x)p(z|x) \log \frac{p(x)p(z|x)}{q(\hat{x}|z)q(z)} dz dx.$$  \hfill (1)

The Eq.(1) can be deduced to the following form

$$\min_{\theta, \phi} E_{x \sim p(x)} \left[ E_{z \sim p(z|x)} \left[ -\log q(\hat{x}|z) + KL(p(z|x)||q(z)) \right] \right].$$  \hfill (2)

The first part in Eq.(2) represents the reconstruction loss which is implemented by the reparameterization trick. The second part can be treated as the Variational Information Bottleneck (VIB) [16] which is proved to be contributive to the feature extraction. Without the VIB form, the VAE will degrade to a normal autoencoder.

It is a trivial solution to extend the single modal VAE to the multimodal scenario, because a multimodal dataset contains distinctive latent features and common latent features. It is impracticable to extract high quality common latent features directly from all modalities with traditional encoders. We use the Variational Information Bottleneck (VIB) [16] to extract the common latent features among different modalities. The reconstruction loss of multimodal common latent feature is

$$\min \sum_{i=1}^{M} E_{z \sim p(z|x_i)} \left[ E_{z' \sim p(z'|x_i)} \left[ -\log q(\hat{x}'|z') + KL(p(z'|x_i)||q(z')) \right] \right].$$  \hfill (3)

It is widely accepted that common latent features of different modalities share similar distributions. Based on the above fact, we propose two cross reconstruction, global cross reconstruction and local cross reconstruction, to promote the feature extraction process. The cross reconstruction methods guide the common latent features of each modality shares similar distribution.

B. GLOBAL CROSS RECONSTRUCTION

The global cross reconstruction process assign each modal with a VAE which is described in the last section to extract latent features. The encoders in the network accept input $\{x^i| i = 1, \ldots, M\}$ of multimodal simultaneously to produce latent features $\{z^i| i = 1, \ldots, M\}$. The decoder $D^j$ of modal $i$ reconstructs the input $M$ times within each iteration, since it takes all modal latent feature $\{z^j| j = 1, \ldots, M\}$ as inputs, and results in multiple reconstructions $\{\hat{x}^j| j = 1, \ldots, M\}$. Providing data $x^i$ in modal $i$ and $x^j$ in modal $j$, the global cross reconstruction process of the $i$-th modal are described as follows.

$$\mu^i, \sigma^i = E^i(x^i; \theta^i),$$

$$\hat{z}^i = \mu^i + \varepsilon \cdot \sigma^i,$$

$$\hat{x}^j = D^j(\hat{z}^i; \phi^j),$$

where $\theta^i$ and $\phi^j$ are parameters of the encoder and the decoder of the $i$-th modality.

Therefore, we introduce the cross reconstruction loss besides the normal reconstruction loss described as Eq.(3). The global cross reconstruction loss of reconstructing $x^i$ with $z^j$ ($i \neq j$) is:

$$\min_{\theta^i, \phi^j} E_{z \sim p(z|x^i)} \left[ -E_{z' \sim p(z'|x^i; \theta^i)} \left[ \log q(\hat{x}'^j|z^j; \phi^j) \right] \right].$$  \hfill (4)

where $q(\hat{x}'^j|z^j; \phi^j)$ denotes the decoder used for reconstructing $x^i$ with $z^j$. Note that, it is also the same decoder for reconstructing $x^j$ with $z^i$.

The reconstruction loss of the $i$-th modal is

$$L_i^r = \min_{\theta^i, \phi^j} E_{z \sim p(z|x^i)} \left[ E_{z' \sim p(z'|x^i; \theta^i)} \| x^i - \hat{x}^j \|_2^2 \right].$$  \hfill (5)

The VIB loss of the $i$-th modal is

$$L_i^v = KL(p(z'|x^i)||q(z^j)) = \frac{1}{2} \left( \mu^i \mu^j + (\sigma^i \sigma^j - \ln(\sigma^j \sigma^i) - 1) \right)$$  \hfill (6)
where the $\mu_i(x)$ and $\sigma_i(x)$ are the output of the encoder of the $i$-th modal. The cross reconstruction loss of the $i$-th modal is

$$L_{ci} = \min \sum_{j=1, j \neq i}^{M} E_{x_i \sim p(x_i) \cdot x_j \sim p(x_j)}[E_{\hat{z}_j \sim p(\hat{z}_j | x_i, \theta)}[\|x_i - \hat{x}_j\|^2]]$$

(7)

Therefore, combining Eq.(5), Eq.(6) and Eq.(7), we get the final loss function of a GCR network

$$L_{gcr} = \sum_{i=1}^{M} (L_{ci}^i + \beta L_{vi}^i + \gamma L_{vi}^i).$$

(8)

where $\beta$ and $\gamma$ are super parameters to control the weight of VIB and the weight of cross construction respectively.

Figure 1(a) illustrates a two-modal global cross reconstruction network. The two encoders $E^1, E^2$ accept data $\{x^1, x^2\}$ from two modals simultaneously. In the first modal, the encoder $E^1$ encodes the $x^1$ into common latent feature $z^1$. The decoder generates a reconstruction $\hat{x}^1$ for the input $x^1$. Besides, the latent feature of the second modal $z^2$ is conveyed to the decoder of the first modal $D^1$ to produce the cross reconstruction $\hat{x}^{21}$. Consequently, the two losses, reconstruction loss of the first modal and cross reconstruction loss between the first modal and the second modal, are used to guide the network training. The same process is done for the second modal.

Global cross reconstruction builds implicit connection among different modalities by exploring the probability theory of connection among different modalities built by cross reconstruction. Here we assume that $q(\hat{x}_i^1 | z^i; \varphi^i)$ is a Gaussian distribution in the $i$-th modality with mean $\hat{\mu}_i(z^i)$ and variance $\hat{\sigma}_i^2(z^i)$, and $q(\hat{x}_j^2 | z^j; \varphi^j)$ is also a Gaussian distribution in the $i$-th modality with mean $\hat{\mu}_j(z^j)$ and variance $\hat{\sigma}_j^2(z^j)$. $\hat{\mu}_i(.)$ and $\hat{\sigma}_i^2(.)$ are the distribution parameters of reconstruction distribution of the $i$-th modal.

Then we further derive the loss function of cross reconstruction:

$$L^i = E_{x_i \sim p(x_i)}[-E_{x_j \sim p(x_j) | x_i, \theta}][\log q(\hat{x}_i^1 | z^i; \varphi^i)]$$

$$= E_{x_i \sim p(x_i)}[E_{\hat{z}_j \sim p(\hat{z}_j | x_i, \theta)}\frac{1}{2} \log 2\pi$$

$$+ \frac{1}{2} \log \hat{\sigma}_j^2(z^j) + \frac{(x_i - \hat{\mu}_j(z^j))^2}{2\hat{\sigma}_j^2(z^j)}]$$

$$L^j = E_{x_j \sim p(x_j)}[-E_{x_i \sim p(x_i) | x_j, \theta}][\log q(x_j^2 | z^j; \varphi^j)]$$

$$= E_{x_j \sim p(x_j)}[E_{\hat{z}_i \sim p(\hat{z}_i | x_j, \theta)}\frac{1}{2} \log 2\pi$$

$$+ \frac{1}{2} \log \hat{\sigma}_i^2(z^i) + \frac{(x_j - \hat{\mu}_i(z^i))^2}{2\hat{\sigma}_i^2(z^i)}]$$

where $L^i$ represents the loss of reconstructing $x^i$ with $\hat{z}^i$; $L^j$ represents the loss of reconstructing $x^j$ with $\hat{z}^j$.

Combining $L^i$ and $L^j$, both $\hat{\mu}_i(z^i)$ and $\hat{\mu}_j(z^j)$ reduce the difference from $x^i$, which means that the difference between $\hat{\mu}_i(z^i)$ and $\hat{\mu}_j(z^j)$ also decreases. Note that both $\hat{\mu}_i(z^i)$ and $\hat{\mu}_j(z^j)$ are outputs of the same decoder $D^j$, so the input of the decoder $D^j$, i.e. $z^i$ and $z^j$ are similar. $z^i$ and $z^j$ are sampled from Gaussian distribution $N_i(\mu(x^i), \sigma(x^i))$ and $N_j(\mu(x^j), \sigma(x^j))$ respectively, and are generated with reparameterization trick:

$$z^i = \text{Reparameterization}(\mu(x^i), \sigma(x^i))$$

$$z^j = \text{Reparameterization}(\mu(x^j), \sigma(x^j))$$

where $\mu(x^i)$ and $\sigma(x^i)$ are the latent distribution parameters which are outputs of the encoder $E^i$, $\mu(x^j)$ and $\sigma(x^j)$ are the outputs of the encoder $E^j$; $z^i$ and $z^j$ are randomly sampled from Gaussian distributions $N_i(\mu(x^i), \sigma(x^i))$ and $N_j(\mu(x^j), \sigma(x^j))$ respectively. Afterward, we calculate the Wasserstein distance [30] between $N_i$ and $N_j$.

$$W(N_i, N_j) = \inf_{\epsilon \in \Pi(N_i, N_j)} E_{ (z^i, z^j) \sim \epsilon}[\|z^i - z^j\|]$$

(9)

where $\Pi(N_i, N_j)$ denotes the set of all joint distributions where the marginals of $\epsilon(z^i, z^j)$ are $N_i$ and $N_j$ respectively. Since the $z^i$ and $z^j$ are close, we conclude that the two distribution of latent feature are also similar base on Eq. 9.

Therefore, under the constraints of cross reconstruction, these encoders reduce the distribution differences of multimodal features, which shows that the cross reconstruction constrains the latent features to share similar distributions in feature spaces of different modalities.

C. LOCAL CROSS RECONSTRUCTION

Furthermore, we consider a more effective latent common feature extraction method and name it Local Cross Reconstruction(LCR). We build the LCR based on the following fact. Each modal has distinctive latent features that
reflects the modal’s characteristics. Meanwhile, all modals share common latent features in order that they have similar cluster assignment. In the GCR network, we only extract common latent features from each modality using VIB. The GCR mixes the distinctive latent features with common latent features in one $z_i$, which leads to indistinct clusters. Therefore, apart from the GCR, we propose the Local Cross Reconstruction (LCR) network.

An LCR network consists of $M$ deep variational autoencoders. There are two variational encoders $E^i = (\tilde{E}^i, \tilde{E}^j)$ for each modality which extract a pair of latent features $(\tilde{z}^i, \tilde{z}^j)$ for each modal. $\tilde{z}^i$ means the distinctive latent features of modal $i$, and $\tilde{z}^j$ represents the common latent features of modal $i$. Specifically, the two encoders share the same parameters in each modal except the last layer and the reparameterization layer. At the last layer, the encoders generate distribution parameters ($\mu^i, \sigma^i_2$) and ($\mu^j, \sigma^j_2$) respectively. At the reparameterization layer, the two latent features, $\tilde{z}^i$ and $\tilde{z}^j$, are built using the reparameterization skill.

The decoder $D^j$ accepts multiple inputs. First, it takes a pair of latent features $(\tilde{z}^i, \tilde{z}^j)$ composed of the distinctive latent feature of current modal and common latent features from other modal.

Providing data $x^i$ in modal $i$ and $x^j$ in modal $j$, the local cross reconstruction process of the $i$-th modal are described as follows.

$$\hat{\mu}^i, \hat{\sigma}^i = \tilde{E}(x^i; \theta^i)$$

$$\hat{\mu}^j, \hat{\sigma}^j = \tilde{E}(x^j; \theta^j)$$

$$\tilde{z}^i = \hat{\mu}^i + \epsilon \cdot \hat{\sigma}^i$$

$$\tilde{z}^j = \hat{\mu}^j + \epsilon \cdot \hat{\sigma}^j$$

$$\hat{x}^i = D^j(\tilde{z}^i, \tilde{z}^j; \phi^i)$$

$$\hat{x}^j = D^j(\tilde{z}^i, \tilde{z}^j; \phi^j)$$

where $\tilde{z}^i$ indicates the distinctive latent feature of the $i$-th modal.

Naturally, the local cross reconstruction network loss also contains two parts, the reconstruction loss of current modal and the cross reconstruction losses between current modal and every other modal. The reconstruction loss of modal $i$ is

$$L^i_r = \min \left\{ E_{x^i \sim p(x)}[E_{z^i \sim p(z_i|x^i; \theta^i)} \|x^i - \hat{x}^i\|^2] \right\}$$

(10)

The cross reconstruction loss of modal $i$ is

$$L^j_c = \min \sum_{j = 1, j \neq i}^{M} E_{x^j \sim p(x^j)} [E_{z^j \sim p(z_j|x^j; \theta^j)} [\|x^j - \hat{x}^j\|^2]]$$

(11)

Therefore, Combining Eq.(10) and Eq.(11), we get the final loss function of a LCR network

$$L_{lcr} = \sum_{i=1}^{M} (L^i_r + \gamma L^j_c).$$

(12)

where $\gamma$ is super parameter to control the weight of cross construction.

For instance, Fig. 1(b) shows a two-modal local cross reconstruction network. The two encoders $[E^1, E^2]$ accept data $[x^1, x^2]$ from two modals simultaneously. The encoder in the first modal $E^2$ encodes the $x^1$ into distinctive latent feature $\tilde{z}^1$ and common latent feature $\tilde{z}^2$. After that, we retrieve a pair of latent features, $(\tilde{z}^1, \tilde{z}^2)$, as the input of the decoder of the current modal. The decoder generates a reconstruction $\hat{x}^1$ for the input $x^1$. Besides, the common latent feature $\tilde{z}^1$ is conveyed to the decoder of the second modal to produce the cross reconstruction $\hat{x}^{12}$. For the second modal, the same process is executed.

We now explain the LCR theoretically in detail. Given $\hat{\mu}^i$ and $\hat{\sigma}^j$ are the distribution parameters of the common latent feature $\tilde{z}^j$, $E^i$ and $D^j$ are the encoder and the decoder of modal $i$ with parameters $\theta^i$ and $\phi^j$ respectively. $\hat{x}^j$ means the reconstruction of the input $x^j$ using the distinctive latent feature $\tilde{z}^i$ and the common latent feature $\tilde{z}^j$.

In this scenario, the reconstruction loss and cross reconstruction loss of modal $i$ are

$$L^i_r = E_{x^i \sim p(x)}[E_{z^i \sim p(z_i|x^i; \theta^i)} \|x^i - \hat{x}^i\|^2]$$

$$L^j_c = E_{x^j \sim p(x^j)} [E_{z^j \sim p(z_j|x^j; \theta^j)} [\|x^j - \hat{x}^j\|^2]]$$

where $L^i_r$ demonstrates the reconstruction loss using the output of the decoder $\hat{x}^i = D^j(\tilde{z}^j, \tilde{z}^i; \phi^j)$, $L^j_c$ indicates the cross reconstruction loss using the output of the decoder $\hat{x}^j = D^j(\tilde{z}^i, \tilde{z}^j; \phi^j)$. The optimization of Eq.(13) and Eq.(14) updates the parameters $\theta^i$ and $\phi^j$ of the encoder $E^i$ and decoder $D^j$, which means the distinctive latent feature of the $j$-th modal $\tilde{z}^j$ keeps invariant when the $i$-th modal is being trained.

The two loss functions guide the training process to reduce the mean square error between the two reconstructions and the original data. Therefore, for the $i$-th modal, the two reconstructions approaches nearly equality with the input $x^i$ after the network is trained. Consequently, the distance between the cross reconstruction $\hat{x}^j$ and the reconstruction $\hat{x}^i$ are minimized. As they are the outputs of the same decoder $D^i$, the inputs of the decoder is also close. The both inputs contains two parts, in which the distinctive latent feature $\tilde{z}^i$ is not changed. Therefore, the common latent features from the $i$-th modal $\tilde{z}^i$ and the $j$-th modal $\tilde{z}^j$ are close to each other. We draw the conclusion that the common latent features of the two modal, $\tilde{z}^i$ and $\tilde{z}^j$, are close through removing the same distinctive latent feature $\tilde{z}^i$.

Given the common latent feature of the two modal are close, the distributions where they are sampled from are also close by calculating the Wasserstein distance between the two distributions. Providing $\tilde{z}$ is sampled from $\mathcal{N}(\tilde{\mu}^i, (\tilde{\sigma}^i)^2)$, $\tilde{z}^2$ is sampled from $\mathcal{N}(\tilde{\mu}^j, (\tilde{\sigma}^j)^2)$, the Wasserstein distance between $\mathcal{N}(\tilde{\mu}^i, (\tilde{\sigma}^i)^2)$ and $\mathcal{N}(\tilde{\mu}^j, (\tilde{\sigma}^j)^2)$ is

$$W(\mathcal{N}(\tilde{\mu}^i, (\tilde{\sigma}^i)^2), \mathcal{N}(\tilde{\mu}^j, (\tilde{\sigma}^j)^2)) = \inf_{\epsilon \in \Pi(\mathcal{N}(\tilde{\mu}^i, (\tilde{\sigma}^i)^2), \mathcal{N}(\tilde{\mu}^j, (\tilde{\sigma}^j)^2))} \int_{\tilde{z}} \|\tilde{z}^j - \tilde{z}\|.$$
where $\varepsilon \in \Pi(N^I, N^J)$ represents a sample collection from the joint distribution of $N^I, N^J$.

In summary, the Wasserstein distances between $p(\tilde{\mu}_i, (\tilde{\sigma}_i)^2)$ and $p(\tilde{\mu}_j, (\tilde{\sigma}_j)^2)$ is minimized after training. It is concluded that the LCR network extracts close common latent features from multi-modal data.

IV. THE PROPOSED FRAMEWORK

Generally, multimodal clustering partitions data $X$ in $M$ modalities into $k$ clusters. We propose two novel multimodal clustering algorithms called the Deep Multimodal Clustering with Global Cross Reconstruction(DMGCR) and the Deep Multimodal Clustering with Local Cross Reconstruction(DMLCR) respectively. The two algorithms both contain two stages, the feature extraction stage and the clustering stage. We introduce our algorithms in detail in this section.

A. FEATURE EXTRACTION

1) FEATURE EXTRACTION WITH GLOBAL CROSS RECONSTRUCTION

Multimodal data contain modality-unique features and modality-common features. It is unreachable to extract common latent features directly from different modalities with traditional encoders. We use the Variational Information Bottleneck (VIB) [16] to extract the common latent features among different modalities. And then we apply the global cross reconstruction method to ensure that the common latent features are deployed in the next feature fusion stage because they grasp the shared information among different modalities.

2) FEATURE EXTRACTION WITH LOCAL CROSS RECONSTRUCTION

Furthermore, we utilize the local cross reconstruction autoencoder to extract more precise common latent feature. As described in the previous section, the LCR autoencoder explicitly extracts common latent features from distinctive latent features. The encoder $E^j$ in LCR autoencoders retrieves the latent feature pair $(\tilde{\mu}_j, \tilde{\sigma}_j)^2$. The decoder $\{D^i\}_{j=1, j \neq i}$ reconstructs the input multiple times with every pair in $\{(\tilde{\mu}_i, \tilde{\sigma}_i)^2\}$. Training process take advantage of both distinctive latent feature $\tilde{\mu}_i$ and common latent feature $\tilde{\mu}_j$. However, only the common latent features are deployed in the next feature fusion stage.

B. FEATURE FUSION AND CLUSTERING

After extracting latent features $z^1, z^2, \ldots, z^M$ from all modalities, we fuse those features to merged latent features $z^*$, and cluster on the shared features $z^*$. All the latent features share the same dimensionality $d_z$, the merged features also have $d_z$ dimensions. As shown in FIGURE 2, we use fusion layers to merge extracted features. The fusion layers consist of fully connected layers. First, each latent feature $z^i$ uses a $d_z \times d_z$ dense layer to decide the weighted latent feature $w^i_z$. Next, we concatenate all weighted latent feature to $z^*$ and downsample the concatenated feature to the merged features $z^*$. The output $z^*$ is the merged features:

$$z^* = Fusion(z^1, \ldots, z^M; \eta),$$

where $\eta$ is the parameter of the fusion layers. For the GCR network, the inputs of the fusion layers are the latent features $z^1, z^2, \ldots, z^M$. For the LCR network, the inputs of the fusion layers are the common latent features $z^1, z^2, \ldots, z^M$.

We use mean squared error loss between the fused feature $z^*$ and each extracted latent features $z^i$ of the $i$-th modalities to train the fusion layers:

$$\min \eta \sum_{i=1}^{M} \|z^* - z^i\|^2.$$

After training the fusion layers, the merged latent features $z^*$ preferably grasps the shared information among modalities. Finally, we perform clustering on the merged features. We choose $k$-means as the final clustering algorithm in this article.
C. THE DMGCR ALGORITHM

We first implement the deep multimodal clustering with global cross reconstruction (DMGCR). The algorithm is composed of two stages.

In the first stage, we extract latent features from different modalities using global cross reconstruction to training autoencoders, which ensures the extracted features of different modalities sharing similar distributions. In this stage, we train the GCR autoencoders with the loss function Eq.(8).

In the second stage, we fuse the features to merged features by fusion layers, then we cluster on the merged features. This stage is guided by the fusion loss demonstrated as Eq.(17).

We adopt ADAM [31] as the optimizer with learning rate \( \eta = 10^{-3} \) in both stages. The whole process of DMGCR is summarized in Algorithm 1.

Algorithm 1 The DMGCR Algorithm

Require:
- The number of modality \( M \);
- The dataset of \( N \) data points for each modality \( i: x^i = \{x^i_1, x^i_2, \ldots, x^i_n\} \), \( i \in [1, M] \).
1: for number of training iterations do
2: Sampling a minibatch of samples from each modality;
3: for each \( i \in [1, M] \) do
4: \( \mu(x^i), \sigma(x^i) = E^i(x^i; \theta^i) \);
5: \( z^i = \text{Reparameterization}(\mu(x^i), \sigma(x^i)) \);
6: \( \hat{x}^i = D^i(z^i, \phi^i) \);
7: for each \( j \in [1, m] \backslash i \) do
8: \( \hat{x}^j = D^j(z^j, \hat{\phi}^j) \);
9: end for
10: Training \( E^i \) and \( D^i \) with Eq.(8);
11: end for
12: end for
13: for number of training iterations do
14: Sampling a minibatch of samples from each modality;
15: for each \( i \in [1, M] \) do
16: \( \mu(x^i), \sigma(x^i) = E^i(x^i; \theta^i) \);
17: \( z^i = \text{Reparameterization}(\mu(x^i), \sigma(x^i)) \);
18: end for
19: \( z^* = \text{Fusion}(z^1, z^2, \ldots, z^m; \eta) \);
20: Training the fusion layer with Eq.(17);
21: end for
22: for each \( i \in [1, M] \) do
23: \( \mu(x^i), \sigma(x^i) = E^i(x^i; \theta^i) \);
24: \( z^i = \text{Reparameterization}(\mu(x^i), \sigma(x^i)) \);
25: end for
26: \( z^* = \text{Fusion}(z^1, z^2, \ldots, z^m; \eta) \);
27: \( S = k\text{-means}(z^*, k) \).
28: end for

D. THE DMLCR ALGORITHM

We further implement the deep multimodal clustering with local cross reconstruction (DMLCR). The algorithm is composed of two stages too.

In the first stage, we extract latent features from different modalities using local cross reconstruction to training autoencoders, which explicitly separates common latent feature from distinctive latent feature. In this stage, we train the LCR autoencoders with the loss function Eq.(12).

In the second stage, we fuse the features to merged features by the same fusion layers as the former algorithm, then we cluster on the merged features. This stage is guided by the fusion loss demonstrated as Eq.(17).

We adopt ADAM [31] as the optimizer with learning rate \( \eta = 10^{-3} \) in both stages. The whole process of DMLCR is summarized in Algorithm 2.

Algorithm 2 The DMLCR Algorithm

Require:
- The number of modality \( M \);
- The dataset of \( N \) data points for each modality \( i: x^i = \{x^i_1, x^i_2, \ldots, x^i_n\} \), \( i \in [1, M] \).
1: for number of training iterations do
2: Sampling a minibatch of samples from each modality;
3: for each \( i \in [1, M] \) do
4: \( \tilde{\mu}(x^i), \tilde{\sigma}(x^i) = \tilde{E}^i(x^i; \theta^i) \);
5: \( \tilde{\mu}(x^i), \tilde{\sigma}(x^i) = \tilde{E}^i(x^i; \theta^i) \);
6: \( z^i = \text{Reparameterization}(\tilde{\mu}(x^i), \tilde{\sigma}(x^i)) \);
7: \( \tilde{z}^j = \text{Reparameterization}(\tilde{\mu}(x^i), \tilde{\sigma}(x^i)) \);
8: \( \hat{x}^i = D^i(\tilde{z}^j, \tilde{\hat{\phi}}^i) \);
9: for each \( j \in [1, m] \backslash i \) do
10: \( \hat{x}^j = D^j(\tilde{z}^j, \tilde{\hat{\phi}}^j) \);
11: end for
12: Training \( E^i \) and \( D^i \) with Eq.(12);
13: end for
14: end for
15: for number of training iterations do
16: Sampling a minibatch of samples from each modality;
17: for each \( i \in [1, M] \) do
18: \( \tilde{\mu}(x^i), \tilde{\sigma}(x^i) = \tilde{E}^i(x^i; \theta^i) \);
19: \( z^i = \text{Reparameterization}(\tilde{\mu}(x^i), \tilde{\sigma}(x^i)) \);
20: end for
21: \( z^* = \text{Fusion}(z^1, z^2, \ldots, z^m; \eta) \);
22: Training the fusion layer with Eq.(17);
23: end for
24: for each \( i \in [1, M] \) do
25: \( \tilde{\mu}(x^i), \tilde{\sigma}(x^i) = \tilde{E}^i(x^i; \theta^i) \);
26: \( \tilde{z}^i = \text{Reparameterization}(\tilde{\mu}(x^i), \tilde{\sigma}(x^i)) \);
27: end for
28: \( z^* = \text{Fusion}(z^1, z^2, \ldots, z^m; \eta) \);
29: \( S = k\text{-means}(z^*, k) \).
30: end for

V. EXPERIMENTS

We compare our DMGCR and DMLCR with nine state-of-art baseline algorithms on six multimodal datasets. We evaluate the clustering results on three metrics. The detail and results are described in this section.
A. DESCRIPTION OF DATASETS
We evaluate our framework on six benchmark multimodal datasets: AwA,\(^1\) CNN,\(^2\) Digits,\(^3\) Caltech101,\(^4\) LUse-21\(^5\) and Scene-15 [32].

AwA comprises images of animals, which has 5814 samples belonging to 10 clusters. We select three features as three modalities, local self-similarity features, SIFT features and SURF features, which are all pretrained features of the images provided by the original source. CNN is a news dataset that contains two modalities of 2107 samples belonging to 7 clusters. The first modality consists of text contents, the second modality contains the images of articles. Digits consists of features of handwritten numerals (‘0’–‘9’) extracted from a collection of Dutch utility maps. It contains three modalities of 2000 samples belonging to 10 clusters. The three modalities respectively have 76, 216 and 240 dimensions. Caltech101, LUse-21 and Scene-15 all consist of pictures of objects: we extract LBP, GIST and CENTRIST descriptors from these datasets as three modalities. Caltech101 is an object recognition data set containing 8677 images, belonging to 101 categories. We choose 712 samples and these samples belong to 10 clusters. LUse-21 contains 2100 samples belonging to 21 clusters. Scene-15 contains 3000 samples assigned to 15 clusters. Table 1 shows the statistics about the datasets, in which \(M\) represents for number of modalities, \(d^i\) is the dimension of the \(i\)-th modality, \(N\) indicate the number of objects, \(K\) means the number of clusters.

|     | \(M\) | \(d^1\) | \(d^2\) | \(d^3\) | \(N\) | \(K\) |
|-----|-------|-------|-------|-------|------|------|
| AwA | 3     | 2000  | 2000  | 2000  | 5814 | 10   |
| CNN | 2     | 35719 | 996   | -     | 2107 | 7    |
| Digits | 3 | 76    | 216   | 240   | 2000 | 10   |
| Caltech101 | 3 | 254   | 256   | 256   | 712  | 10   |
| LUse-21 | 3 | 254   | 256   | 256   | 3000 | 15   |
| Scene-15 | 3 | 254   | 256   | 256   | 2100 | 21   |

B. BASELINE ALGORITHMS
We compare the proposed framework with the following baseline algorithms: (1) Single modality clustering methods: DEC [5], IDEC [6] and JULE [18]. We test these methods on each modality and take the best result as the final results. (2) Multimodal clustering methods: MMSC [22], RMKMC [23], DIMSC [24], LT-MSC [25], ECMSC [26] and DCCA [15]. Among these multimodal clustering methods, DCCA and DCCA and DCCA are two-modal methods. In order to extend DCCA to multimodal clustering task, we combine every two of the multiple modalities, and take the average results as the final result. (3) The simplified version of DMGCR and DMLCR: the proposed DMGCR algorithm without cross reconstruction regularization term, called DMCV. The DMCV without the VIB regularization term, called DMC.

C. EVALUATION METRICS
We evaluate our approaches on three common metrics, Accuracy (ACC), Normalized Mutual Information (NMI), and Purity.

ACC measures the best mapping between cluster assignments and true labels, which is defined by

\[
\text{ACC} = \max_m \frac{\sum_{i=1}^N 1(l_i = m(c_i))}{N}
\]

where \(l_i\) and \(c_i\) are the true label and predicted cluster of data point \(x_i\), and \(m(c_i)\) is the permutation mapping function that maps each cluster label \(c_i\) to the equivalent label from data.

NMI calculates the normalized measure of similarity between two labels of the same data, which is defined as

\[
\text{NMI} = \frac{I(l; c)}{\max_l \{H(l), H(c)\}}
\]

where \(I(l, c)\) denotes the mutual information between true label \(l\) and predicted cluster \(c\), and \(H\) represents their entropy. Results of NMI do not change by permutations of clusters, and they are normalized to the range of [0, 1], with 0 meaning no correlation and 1 exhibiting perfect correlation.

Purity reflects the percentage of each cluster containing the correctly grouped samples, which is defined as

\[
\text{Purity} = \frac{1}{k} \sum_{i=1}^k \frac{s_i}{n_i}
\]

where \(k\) is the number of classes, \(s_i\) and \(n_i\) are the number of samples of the \(i\)-th cluster and total samples respectively. \(P_i\) denotes the distribution of correctly classified samples of all clusters.

D. MODEL AND PARAMETER SETTINGS
The model parameter settings of our experiments are as follows:

(1) For baseline algorithms, we set their parameters with values provided in the original papers. During training, we fine-tune the parameters of baseline methods to get the best performance.

(2) We use three autoencoders to handle datasets with three modalities and two autoencoders for datasets with two modalities. The encoders and decoders are composed of fully connected layers. We use sigmoid as the activation function in the last layer of decoders, and use ReLU activation in the other layers of encoders and decoders. The weights of neural networks in our framework are randomly initialized. We set the learning rate of ADAM to 0.001.

(3) We select \(k\)-means as the final clustering method considering the interpretation of the Euclidean distance in the feature space as diffusion distance in the input space [33]–[35]. It can be replaced by other clustering methods.
E. PERFORMANCE EVALUATION

1) CLUSTERING RESULTS

The experiment results of clustering ACC, NMI and Purity are summarized in Table 2, Table 3 and Table 4. The best results are marked in bold.

| TABLE 2. Clustering results ACC (%) |  |
|-------------------------------------|--|
| Digit               | CNN | AwA | Caltech101 | LuSe-21 | Scene-15 |
| DEC                  | 44.07 | 36.96 | 16.10 | 35.74 | 12.34 | 16.97 |
| IDEC                 | 46.70 | 31.69 | 20.08 | 43.54 | 25.14 | 25.03 |
| JULE                 | 76.36 | 40.00 | 23.05 | 63.20 | 27.14 | 38.10 |
| DCCAE                | 66.19 | 34.27 | 24.31 | 62.85 | 23.72 | 34.95 |
| MMSM                 | 54.00 | 32.02 | 20.28 | 49.02 | 20.38 | 18.20 |
| RMKMC                | 73.30 | 34.21 | 23.17 | 51.97 | 27.24 | 40.03 |
| LT-MSC               | 73.35 | 21.26 | 24.17 | 61.35 | 30.86 | 43.93 |
| DIMSC                | 52.10 | 21.64 | 20.79 | 39.47 | 22.81 | 23.87 |
| ECMSC                | 77.40 | 25.34 | 22.19 | 53.90 | 25.29 | 42.70 |
| DMCV                 | 87.36 | 54.64 | 26.17 | 64.50 | 28.29 | 45.20 |
| DMC                  | 85.55 | 55.52 | 26.23 | 62.50 | 30.14 | 44.60 |
| DMLGR                | 90.75 | 66.68 | 28.67 | 66.57 | 31.62 | 54.00 |
| DMLCR                | 91.55 | 67.75 | 26.56 | 68.75 | 34.86 | 55.97 |

| TABLE 3. Clustering results NMI (%) |  |
|-------------------------------------|--|
| Digit               | CNN | AwA | Caltech101 | LuSe-21 | Scene-15 |
| DEC                  | 35.11 | 20.79 | 4.10 | 22.63 | 15.18 | 16.10 |
| IDEC                 | 44.46 | 13.28 | 6.73 | 31.00 | 28.02 | 19.92 |
| JULE                 | 70.25 | 18.44 | 8.33 | 54.12 | 34.99 | 35.84 |
| DCCAE                | 67.59 | 14.27 | 10.13 | 57.96 | 32.05 | 37.82 |
| MMSM                 | 57.21 | 11.31 | 1.81 | 45.55 | 26.56 | 14.44 |
| RMKMC                | 70.74 | 13.07 | 9.23 | 50.49 | 30.55 | 39.03 |
| LT-MSC               | 70.40 | 13.84 | 9.24 | 53.41 | 35.62 | 41.36 |
| DIMSC                | 39.25 | 12.67 | 7.61 | 33.74 | 21.69 | 19.39 |
| ECMSC                | 71.21 | 18.87 | 7.33 | 51.53 | 25.93 | 41.16 |
| DMCV                 | 80.55 | 45.56 | 10.05 | 60.54 | 42.56 | 46.58 |
| DMC                  | 74.98 | 37.74 | 10.20 | 57.51 | 39.17 | 40.58 |
| DMLGR                | 83.17 | 50.93 | 12.72 | 64.31 | 40.39 | 55.55 |
| DMLCR                | 85.25 | 52.84 | 11.25 | 66.73 | 43.02 | 57.62 |

| TABLE 4. Clustering results Purity (%) |  |
|---------------------------------------|--|
| Digit                   | CNN | AwA | Caltech101 | LuSe-21 | Scene-15 |
| DEC                    | 47.97 | 40.84 | 17.96 | 37.88 | 14.29 | 17.65 |
| IDEC                   | 49.00 | 37.66 | 21.87 | 47.19 | 27.52 | 26.63 |
| JULE                   | 77.44 | 40.80 | 23.58 | 68.68 | 30.57 | 38.97 |
| DCCAE                  | 69.92 | 35.19 | 25.68 | 66.29 | 27.49 | 38.69 |
| MMSM                   | 58.25 | 32.21 | 20.64 | 54.21 | 21.81 | 18.60 |
| RMKMC                  | 76.40 | 34.87 | 24.70 | 58.85 | 29.00 | 41.40 |
| LT-MSC                 | 77.35 | 25.63 | 27.35 | 64.58 | 32.24 | 44.70 |
| DIMSC                  | 53.50 | 24.54 | 23.89 | 47.33 | 24.43 | 25.87 |
| ECMSC                  | 78.13 | 25.49 | 23.24 | 61.94 | 28.33 | 44.80 |
| DMCV                   | 84.64 | 62.35 | 25.64 | 64.58 | 36.50 | 54.55 |
| DMC                   | 85.55 | 56.95 | 27.95 | 62.50 | 34.43 | 46.57 |
| DMLGR                 | 90.75 | 61.70 | 31.84 | 64.46 | 35.05 | 56.60 |
| DMLCR                 | 93.85 | 65.52 | 28.35 | 66.25 | 36.80 | 57.62 |

Firstly, we compare DMLGR and DMLCR with the single modal algorithms DEC, IDEC and JULE. Generally, it can be seen that DMLCR and DMLGR outperforms the single modal algorithms when clustering multimodal data, which indicates the effectiveness of multimodal fusion. DMLGR and DMLCR perform better than DEC and IDEC on all datasets in terms of ACC, NMI and Purity, and outperform JULE in most cases in terms of ACC, NMI and Purity. The JULE algorithm performs the best on only one metric, the purity. However, on the other two metrics our algorithms overpass JULE obviously. Note that DEC and IDEC just use an autoencoder without VIB, and do not perform well on multimodal datasets. These series of results demonstrate that our algorithms are delicately designed for multimodal clustering.

Secondly, we compare DMLGR and DMLCR with multi-modal clustering algorithms MMSC, RMKMC, DIMSC, LT-MSC, ECMSC and DCCAE. Generally, DMLCR outperforms the baseline multimodal methods. Besides, DMLGR also performs better than most multimodal state-of-art works. For instance, compared with baseline methods, the NMI of DMLCR raises 12% on the Digits dataset, 3% on the AwA dataset, 13% on the Cal101 dataset, 14% on the LuSe-21 dataset. DMLCR outperforms MMSC, RMKMC, DIMSC and ECMSC on each dataset of all metrics. LT-MSC and DCCAE seldom perform little better on purity than the proposed algorithms. Nevertheless, the DMLCR is comparable with them in the other metrics. From the above results, we determine that the early fusion and the cross construction improve clustering performance on multimodal data.

Thirdly, we simplify our algorithms by dropping the cross reconstruction item from the loss function, and call them DMC and DMLC respectively. The results in the tables indicate that DMLCR outperforms DMC, DMLCR performs better than DMLC. That proves that the cross reconstruction regularization is effective on extracting common features of different modalities.

Finally, DMLCR surpasses the DMLGR on most cases of all metrics. This phenomenon indicates that the local cross construction is beyond the global cross construction by explicitly retrieve the common latent features.

In summary, it can be concluded that our methods performs the best on the multimodal datasets. The proposed cross reconstruction regularization improves the performance of multimodal clustering, which further indicates that it is beneficial to establish early fusion among different modalities.

2) PARAMETER SETTING RESULTS

The loss of the DMLGR algorithm has two super parameters which control the weight of VIB and the cross reconstruction respectively. In Table 5, we discuss the effect of the parameters $\beta$ and $\gamma$ to clustering performance of DMLGR on each dataset. Due to space limitation, we only present the experimental results on the Digits dataset and Cal101 dataset in this paper. Both $\beta$ and $\gamma$ vary in the set $[0, 0.5, 1]$ and the best results are marked in bold.

The value of $\beta$ and $\gamma$ separately reflects how much we want to enforce the cross reconstruction regularization and the VIB
regularization. The $\beta = 1$, $\gamma = 0$ stands for DMGCR without cross reconstruction regularization, the $\beta = 0$, $\gamma = 1$ stands for DMGCR without VIB regularization, and the $\beta = 0$, $\gamma = 0$ stands for DMCR without both cross reconstruction regularization and VIB regularization. The performance of DMGCR without one regularization is better than that of DMGCR without both regularization, but is worse than that of DMGCR with both regularization, which proves the validity of the VIB regularization and cross reconstruction regularization. Furthermore, as shown in the tables, we get the best results when $\beta = 1$ and $\gamma = 0.5$.

The loss function of DMLCR has one super parameter $\beta$ which represents the weight of cross reconstruction. Table 6 shows the clustering performance on difference values of the parameter $\beta$. Similarly, we set the $\beta$ to 0.6 to achieve the best clustering performance.

### Table 5. Parameter setting evaluation for DMGCR (%).

| $\beta$, $\gamma$ | Digits ACC | Caltech101 ACC | Digits NMI | Caltech101 NMI | Digits Purity | Caltech101 Purity |
|-------------------|-------------|----------------|-----------|----------------|---------------|------------------|
| $\beta = 0$, $\gamma = 0$ | 69.05 | 68.05 | 73.55 | 54.07 | 47.98 | 55.76 |
| $\beta = 0.5$, $\gamma = 0$ | 81.20 | 72.84 | 81.20 | 57.16 | 50.39 | 59.55 |
| $\beta = 1$, $\gamma = 0$ | 85.55 | 74.98 | 85.55 | 62.50 | 57.51 | 62.50 |
| $\beta = 0$, $\gamma = 0.5$ | 81.85 | 80.53 | 82.25 | 60.67 | 58.51 | 63.62 |
| $\beta = 0.5$, $\gamma = 0.5$ | 90.34 | 82.64 | 90.35 | 63.76 | 63.68 | 63.34 |
| $\beta = 1$, $\gamma = 0.5$ | 90.75 | 83.17 | 90.75 | 67.67 | 64.31 | 64.46 |
| $\beta = 0$, $\gamma = 1$ | 80.50 | 81.85 | 84.05 | 58.00 | 59.88 | 63.34 |
| $\beta = 0.5$, $\gamma = 1$ | 90.25 | 82.60 | 90.25 | 62.92 | 61.08 | 59.26 |
| $\beta = 1$, $\gamma = 1$ | 90.55 | 82.02 | 90.25 | 58.14 | 58.57 | 61.65 |

### Table 6. Parameter setting evaluation for DMLCR (%).

| $\beta$, $\gamma$ | Digits ACC | Caltech101 ACC | Digits NMI | Caltech101 NMI | Digits Purity | Caltech101 Purity |
|-------------------|-------------|----------------|-----------|----------------|---------------|------------------|
| $\beta = 0$ | 87.36 | 80.55 | 88.64 | 64.50 | 60.54 | 64.58 |
| $\beta = 0.2$ | 88.65 | 82.63 | 91.85 | 66.72 | 64.85 | 63.72 |
| $\beta = 0.4$ | 92.37 | 84.00 | 92.34 | 65.54 | 65.27 | 65.65 |
| $\beta = 0.6$ | 94.72 | 84.72 | 92.55 | 68.59 | 65.85 | 65.72 |
| $\beta = 0.8$ | 92.37 | 83.95 | 92.00 | 67.45 | 64.37 | 65.46 |
| $\beta = 1.0$ | 90.65 | 81.47 | 90.65 | 66.37 | 62.95 | 64.95 |

### VI. CONCLUSION

In this paper, we propose a novel deep multimodal clustering framework and two algorithms with cross reconstruction. Firstly, we analyzed the mechanism how the global cross reconstruction and the local cross reconstruction effectively reduce the distribution differences of multimodal latent features. Secondly, we build a deep multimodal clustering networks framework and two specific implementation, DMGCR and DMLCR. Finally, We compare our DMGCR and DMLCR algorithms with the state-of-the-art multimodal methods on representative multimodal datasets. Experimental results show that our algorithms achieve obviously improvement on multimodal clustering tasks. A further study could assess the end-to-end training instead of two-stage training to simplify the whole framework. Another possible area of future research would be to apply cross reconstruction to more deep clustering methods.
[24] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang, “Diversity-induced multiview subspace clustering,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 586–594.

[25] 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.

[26] X. Wang, X. Guo, Z. Lei, C. Zhang, and S. Z. Li, “Exclusivity-consistency regularized multi-view subspace clustering,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 1–9.

[27] G. Andrew, R. Arora, J. A. Bilmes, and K. Livescu, “Deep canonical correlation analysis,” in Proc. ICML, 2013, pp. 1247–1255.

[28] M. A. Eckert, N. V. Kamdar, C. E. Chang, C. F. Beckmann, M. D. Greicius, and V. Menon, “A cross-modal system linking primary auditory and visual cortices: Evidence from intrinsic fMRI connectivity analysis,” Hum. Brain Mapping, vol. 29, no. 7, pp. 848–857, 2008.

[29] A. L. Beer, T. Plank, and M. W. Greenlee, “Diffusion tensor imaging shows white matter tracts between human auditory and visual cortex,” Exp. Brain Res., vol. 213, nos. 2–3, p. 299, 2011.

[30] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in Proc. Int. Conf. Mach. Learn., 2017, pp. 214–223.

[31] O. Konur, D. Kingma, and J. Ba, “Adam: A method for stochastic optimization,” in Proc. Int. Conf. Learn. Represent., 2015, pp. 1–15.

[32] F.-F. Li and P. Perona, “A Bayesian hierarchical model for learning natural scene categories,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), vol. 2, Jun. 2005, pp. 524–531.

[33] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis, “Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators,” in Proc. NIPS, 2005, pp. 955–962.

[34] R. R. Coifman and S. Lafon, “Diffusion maps,” Appl. Comput. Harmon. Anal., vol. 21, no. 1, pp. 5–30, 2006.

[35] 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: http://arxiv.org/abs/1801.01587

QIANLI ZHAO received the bachelor’s degree in software engineering from the Dalian University of Technology, Dalian, China, in 2011, where he is currently pursuing the Ph.D. degree in computer science. His current research interests include deep learning and data mining.

LINLIN ZONG received the bachelor’s and Ph.D. degrees in software engineering from the Dalian University of Technology, Dalian, China, in 2011 and 2016, respectively. She is currently an Associate Professor with the Dalian University of Technology. Her current research interests include deep learning and data mining.

XIANCHAO ZHANG received the bachelor’s and master’s degrees in mathematics from the National University of Defense Technology, Changsha, China, in 1994 and 1998, respectively, and the Ph.D. degree in computer science from the University of Science and Technology of China, Hefei, China, in 2000. He was a Research and Development Manager in some international companies, from 2000 to 2003. He joined the Dalian University of Technology, Dalian, China, in 2003, where he is currently a Full Professor. His current research interests include design and analysis of algorithms, machine learning, data mining, and information retrieval.

YUANGANG LI received the bachelor’s degree in computer science from the Dalian University of Technology, China, in 1997, the M.B.A. degree from the Dongbei University of Finance and Economics, China, in 2006, and the Ph.D. degree in management science and engineering from the Shanghai University of Finance and Economics, China, in 2018. He is currently a Lecturer with the Business School, Dalian University of Foreign Languages, China. His research interests include complex system modeling and data mining.

XIAORUI TANG received the bachelor’s and master’s degrees in software engineering from the Dalian University of Technology, Dalian, China, in 2016 and 2020, respectively. He is currently an Engineer with Meituan Inc.