# CCA-Flow: Deep Multi-view Subspace Learning with Inverse Autoregressive Flow

**Jia He** HEJIA0149@GMAIL.COM
*Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China*
*Huawei EI Innovation Lab, China*

**Feiyang Pan** PANFEIYANG@ICT.AC.CN
**Fuzhen Zhuang*** ZHUANGFUZHEN@ICT.AC.CN
**Qing He** HEQING@ICT.AC.CN
*Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China*
*University of Chinese Academy of Sciences, Beijing 100049, China*
∗ *Corresponding Author*

**Editors:** Sinno Jialin Pan and Masashi Sugiyama

## Abstract

Multi-view subspace learning aims to learn a shared representation from multiple sources or views of an entity. The learned representation enables reconstruction of common patterns of multi-view data, which helps dimensional reduction, exploratory data analysis, missing view completion, and various downstream tasks. However, existing methods often use simple structured approximations of the posterior of shared latent variables for the sake of computational efficiency. Such oversimplified models have a huge impact on the inference quality and can hurt the representation power. To this end, we propose a new method for multi-view subspace learning that achieves efficient Bayesian inference with strong representation power. Our method, coined CCA-Flow, bases on variational Canonical Correlation Analysis and models the inference network as an Inverse Autoregressive Flow (IAF). With the flow-based variational inference imposed on the latent variables, the posterior approximations can be arbitrarily complex and flexible, and the model can still be efficiently trained with stochastic gradient descent. Experiments on three benchmark multi-view datasets show that our model gives improved representations of shared latent variables and has superior performance against previous works.

**Keywords:** Multi-view Learning, Inverse Autoregressive Flow, Variational Inference

## 1. Introduction

Multi-view machine learning focuses on exploring the consistency and complementary properties of different views to improve the learning performance. Specifically, subspace multi-view learning learns the shared representations of multiple views by considering the correlation between different views, which is one of the basic problem for modern machine learning because of its high efficiency and good representation ability. The learned latent representations should maintain information from multi-view data, which can be helpful for exploratory data analysis, missing view completion, dimension reduction, or for downstream tasks.

Canonical correlation analysis (CCA) is a well-established method for multi-view subspace learning. The basic idea is to project different views into a shared latent space and maximize the correlation among the projections Hotelling (1936). Bach and Jordan (2005) formulates CCA as a probabilistic graphical model (as shown in Figure 1), where the multi-view observations are assumed generated from a shared latent variable. Thus the subspace learning problem can be cast as an inference problem to infer the posterior distribution of the latent variable from multi-view observations.

Following the line of CCA, there has been much effort to design inference models with strong representation power. For example, kernel-based CCA methods transform multiple views into the Hilbert space and learn a linear CCA upon it Lai and Fyfe (2000); Akaho (2001); Melzer et al. (2001); Williams and Seeger (2001); Lopez-Paz et al. (2014); He et al. (2017). Although proven effective on tabular cases, the scalability of kernel-based methods are limited by the choice of kernels and the heavy computational cost of the Gram matrices.

With the rapid growth of deep learning, a more efficient and effective alternative is studied, namely the Deep CCA, which introduces deep neural networks to model the mappings from observations to the latent space Andrew et al. (2013). In the past few years, the use of multi-view auto-encoders (MVAE) Ngiam et al. (2011) has been proven effective across a number of tasks including acoustics processing Wang et al. (2015b), word embedding Lu et al. (2015), and image-text matching Yan and Mikolajczyk (2015). MVAE trains an auto-encoder to model both the inference model (the encoder) and the generative model (the decoder). Instead of learning a point estimation on the latent space as in Ngiam et al. (2011); Andrew et al. (2013), variational CCA Wang et al. (2017) optimizes over a parameterized class of density functions to approximate the posterior of the latent variable. In this way, the inference process tends to be more robust, and the method achieves state-of-the-art performances over a number of multi-view learning tasks.

However, for computational efficiency, variational CCA often uses oversimplified structure for the posterior approximation. I.e., it assumes that the latent variable obey a diagonal Gaussian distribution and the variables are independently distributed, which is obviously unrealistic. Such oversimplification has a huge impact on the representation power.

We are interested to build a multi-view subspace learning methods with strong representation power as well as scalability to large and complex datasets. To this end, we propose a novel model, coined *CCA-Flow*, which follows the structure of variational *CCA* and models the inference network using *Flow*-based variational inference.

We model the generative and inference process of variational canonical correlation analysis nonlinearly by deep neural networks. In particular, we incorporate inverse autoregressive flow (IAF) Kingma et al. (2016) into the deep inference network, as it is known that richer and more faithful posterior approximations do result in better performance Rezende and Mohamed (2015). In this way, we can model the correlations among latent variables, and obtain a sufficiently flexible and arbitrarily complex approximated posterior density. Hence, our model can achieve strong representation power, especially for complex multi-view data.

Even though we model the latent variables with complex posterior density, we would like to emphasis that our model is still able to achieve efficient training and fast test-time inference. Specifically, our flow-based inference network consists of a chain of invertible

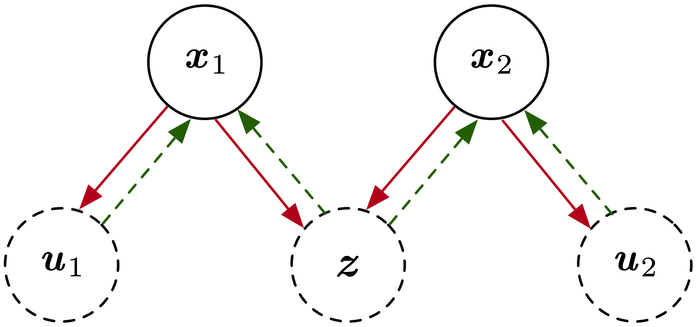

Figure 1: Probabilistic CCA with shared latent $\boldsymbol{z}$ and private view-specific latents $\boldsymbol{u}_1, \boldsymbol{u}_2$. Green dashed arrows: the generative process of the probabilistic graphical model. Red arrows: the inference process to learn the latent representations from observations.

transformations based on autoregressive neural networks, so the inference is as fast as any feed-forward neural networks. Meanwhile, since each network block of the inference network is sophistically designed, the variational posterior density has a simple closed form, so the whole model can be efficiently trained with back-propagation in an end-to-end manner. Therefore, our method is scalable and efficient to use on large high-dimensional datasets.

To summarize, our contributions are as follows.

- We propose a generative variational multi-view subspace model with strong representation power. We use deep neural networks to learn the latent representation, and build a flow-based inference network which is able to fit arbitrarily complex and flexible posterior density of the latent variables.

- We enable efficient learning by using IAF as the backbone of the inference network Kingma et al. (2016) and use Monte Carlo Variational Inference (MCVI) to train the model in an end-to-end manner.

- We conducted experiments on three benchmark multi-view datasets. The results showed that our model has good representation ability of shared latent variables, and can reach superior performance compared with a number of previous works.

## 2. Preliminary: Probabilistic CCA

In this section, we briefly introduce the probabilisitic formulation of canonical correlation analysis (Probabilistic CCA) for multi-view subspace learning and the method of variational CCA (VCCA). Note that, throughout this paper, we will formulate the multi-view learning problem with two views for simplicity of notations. All the methods and statements can be extended to multiple views without effort.

Let $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ be the input variables from two views with dimension $d_1$ and $d_2$, respectively. Following the probabilistic formulation of CCA Bach and Jordan (2005), it is assumed that there exists a shared latent variable $\boldsymbol{z}$ from which $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ can be generated. In this way, the two views $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are naturally connected. Therefore, the task of subspace learning is to inference $\boldsymbol{z}$ from both $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. The probabilistic graphical model in shown in Figure 1. The generative process is as follows:

$$\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{W}_i\boldsymbol{z} + \boldsymbol{u}_i, \boldsymbol{\Sigma}_i), \quad i = 1, 2, \tag{1}$$

where $\boldsymbol{z} \in \mathbb{R}^d$ denotes the shared latent representation that is considered come from a Guassian prior $\mathcal{N}(\boldsymbol{0},\ \boldsymbol{I})$. $\boldsymbol{u}_i \in \mathbb{R}^{d_i}$ denotes the *private* or view-specific representation for the $i^{\text{th}}$ view, and $\boldsymbol{W}_i$ and $\boldsymbol{\Sigma}_i$ denotes for the other parameters in the generative process.

Based on this probabilistic formulation, the problem of subspace learning turns into the inference of $\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z}$ given observations $\boldsymbol{x}_1, \boldsymbol{x}_2$. However, due to the fact that an exact inference of the posterior is intractable, variational CCA Wang et al. (2015a) adopts variational Bayesian inference to approximate the posterior densities of $\boldsymbol{u}_1, \boldsymbol{u}_2$, and $\boldsymbol{z}$. The basic structure is shown in Figure 2. In the inference network, it uses stochastic feed-forward with Gaussian noises to perform Monte Carlo Variational Inference (MCVI). Note that the shared vector $\boldsymbol{z}$ can be dependent on either $\boldsymbol{x}_1$ or $\boldsymbol{x}_2$, so it can deal with the samples with missing views.

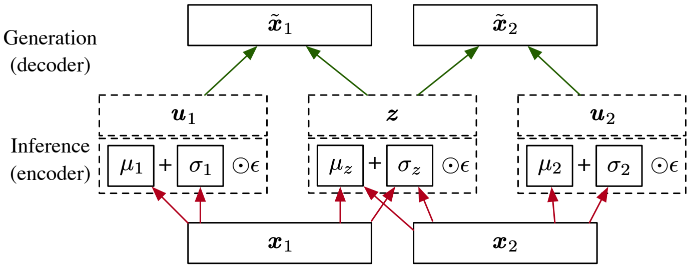
Generation
(decoder)
$\tilde{\boldsymbol{x}}_1$
$\tilde{\boldsymbol{x}}_2$
$\boldsymbol{u}_1$
$\boldsymbol{z}$
$\boldsymbol{u}_2$
Inference
(encoder)
$\mu_1$ + $\sigma_1$ $\odot\epsilon$
$\mu_z$ + $\sigma_z$ $\odot\epsilon$
$\mu_2$ + $\sigma_2$ $\odot\epsilon$
$\boldsymbol{x}_1$
$\boldsymbol{x}_2$

Figure 2: Variational CCA Wang et al. (2015a), which trains a Multi-View AutoEncoder (MVAE) with variational inference. The posterior distributions of latent variables $\boldsymbol{u}_1, \boldsymbol{u}_2$ and $\boldsymbol{z}$ are approximated with diagonal Gaussian distributions, which limits the quality of inferences.

However, there is a key drawback of VCCA that the latent variable $\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z}$'s variational posterior are assumed to be diagonal Gaussian, although the inference and generative processes (also known as the encoder and decoder) are both non-linear models. It does not consider the correlations among dimensions in the latent representations. Consequently, such an oversimplified model limits the fitting power, and makes it difficult to apply to multi-view data with complex structures and dependencies.

## 3. CCA-Flow

In order to enable efficient training and inference as well as better posterior approximation, we propose a new method for multi-view subspace learning, namely Canonical Correlation Analysis with Inverse Autoregressive Flow (CCA-Flow for short).

Firstly, instead of using linear mappings as in original CCA and VCCA, our CCA-Flow adopts deep neural networks as the encoder and decoder to model the generative process $p(\boldsymbol{x}_1, \boldsymbol{x}_2|\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z})$ and the inference process $q(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)$, respectively. Secondly, CCA-Flow incorporates IAFs at each layer of the inference network, so as to reach arbitrarily complex variational posteriors and also enable efficient learning.

### 3.1. Generation and inference with neural networks

By directly applying the Bayesian rule, the probability density of multi-view observations $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is formulated as follows:

$$p(\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{p(\boldsymbol{x}_1, \boldsymbol{x}_2|\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z})p(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z})}{p(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)}, \tag{2}$$

where $p(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z})$ is the prior density of latent variables, $p(\boldsymbol{x}_1, \boldsymbol{x}_2|\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z})$ is the likelihood of observations given the latents, and the denominator $p(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)$ is the true posterior density of latent variables.

However, as the true distribution of data is unknown, exact posterior inference is intractable. Therefore, Variational Inference (VI) uses optimization to approximate the true posterior with a parameterized variational posterior, denoted as $q(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)$. The goal of the optimization problem is to minimize the Kullback–Leibler (KL) divergence between the true posterior and the variational posterior, i.e.,

$$\min_{q \in \mathcal{Q}} \mathbb{D}_{\mathrm{KL}}\big(q(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)||p(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)\big) \tag{3}$$

where $\mathcal{Q}$ denotes for a pre-specified class of parametrized probability densities.

By replacing the true posterior with the variational one in (2), we can have

$$\begin{aligned} \log p(\boldsymbol{x}_1, \boldsymbol{x}_2) = {} & \mathbb{D}_{\mathrm{KL}}(q(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)||p(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)) \\ & + \mathbb{E}_q[\log p(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z}) + \log p(\boldsymbol{x}_1, \boldsymbol{x}_2|\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z})] \\ & - \mathbb{E}_q[\log q(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)], \end{aligned}$$

where $\mathbb{E}_q[\cdot]$ is short for $\mathbb{E}_{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z} \sim q(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)}[\cdot]$.

Since the KL divergence is always non-negative by its definition, minimizing the KL term is equivalent to maximizing the evidence lower bound (ELBO), also known as the variational lower bound,

$$\mathrm{ELBO}(q) = \mathbb{E}_q[\log p(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z}) + \log p(\boldsymbol{x}_1, \boldsymbol{x}_2|\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z}) - \log q(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)]. \tag{4}$$

In the rest of the paper, we use $\boldsymbol{\theta}$ to denote the parameters in the generative model $p(\boldsymbol{x}_1, \boldsymbol{x}_2|\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z})$, and $\boldsymbol{\phi}$ to parameterize the inference network $q(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2)$.

Specifically, in order to disentangle the relationships between multi-views, we further make a natural assumption that $\boldsymbol{u}_1, \boldsymbol{u}_2$ and $\boldsymbol{z}$ has independent prior distributions, and are independent given observations $(\boldsymbol{x}_1, \boldsymbol{x}_2)$, i.e.,

$$\begin{aligned} p(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z}) &= p_0(\boldsymbol{u}_1)p_0(\boldsymbol{u}_2)p_0(\boldsymbol{z}), \\ q_{\boldsymbol{\phi}}(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2) &= q_{\boldsymbol{\phi}_1}(\boldsymbol{u}_1|\boldsymbol{x}_1)q_{\boldsymbol{\phi}_2}(\boldsymbol{u}_2|\boldsymbol{x}_2)q_{\boldsymbol{\phi}_z}(\boldsymbol{z}|\boldsymbol{x}_1, \boldsymbol{x}_2), \end{aligned}$$

where the prior distributions are standard Gaussian: $p_0(\boldsymbol{u}_1)$, $p_0(\boldsymbol{u}_2)$, $p_0(\boldsymbol{z}) \sim \mathcal{N}(\boldsymbol{0},\ \boldsymbol{I})$. Also, we assume that each view $\boldsymbol{x}_i$ is only dependent on the shared variable $\boldsymbol{z}$ and its own private $\boldsymbol{u}_i$.

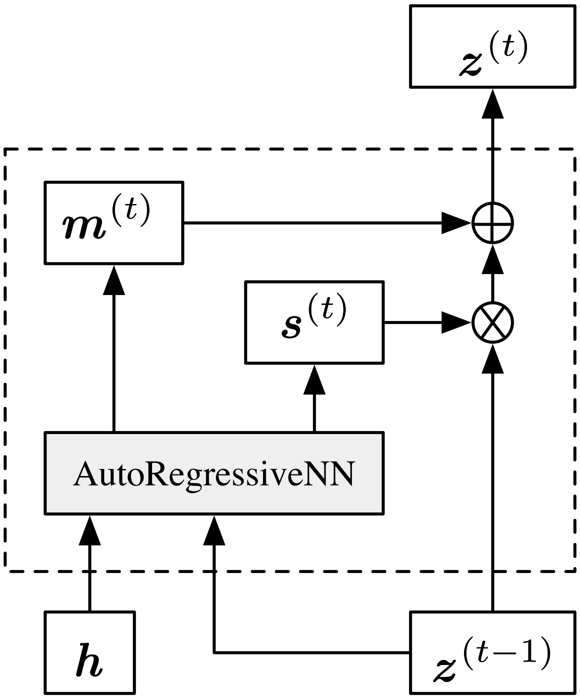

Figure 3: Inverse Autoregressive Transformation

With such parametrization, the variational lower bound is formulated as follows,

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}_1, \boldsymbol{x}_2) = \mathbb{E}_q[&\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_1, \boldsymbol{x}_2 | \boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z}) \\ &+ \log p_0(\boldsymbol{u}_1) + \log p_0(\boldsymbol{u}_2) + \log p_0(\boldsymbol{z}) \\ &- \log q_{\boldsymbol{\phi}}(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z} | \boldsymbol{x}_1, \boldsymbol{x}_2)]. \end{aligned} \tag{5}$$

### 3.2. Improving the representation power with IAF

As mentioned, the latent variables $\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{z}$'s variational posterior are usually oversimplified as diagonal Gaussian in existing works such as VCCA. However, the restriction is too strong and has a huge impact on the inference quality. Inverse Autoregressive Flow (IAF) Kingma et al. (2016), as an extension of Normalizing Flows Rezende and Mohamed (2015), provides a flexible strategy for variance inference of the latent variable's posterior distribution. It contains a chain of inverse autoregressive transformations and scales well to high-dimensional latent spaces, where each transformation is based on an autoregressive neural network. IAF can learn strong variational posterior approximations to closely fit to the true posterior distribution.

In this work, we incorporate IAFs to model the variational posterior for latent variables $\boldsymbol{u}_1$, $\boldsymbol{u}_2$ and $\boldsymbol{z}$. We take the shared latent variable $\boldsymbol{z}$ for an example to describe how to construct the flexible variational posterior by IAF. Similar processes are applied to $\boldsymbol{u}_1$, $\boldsymbol{u}_2$ as well.

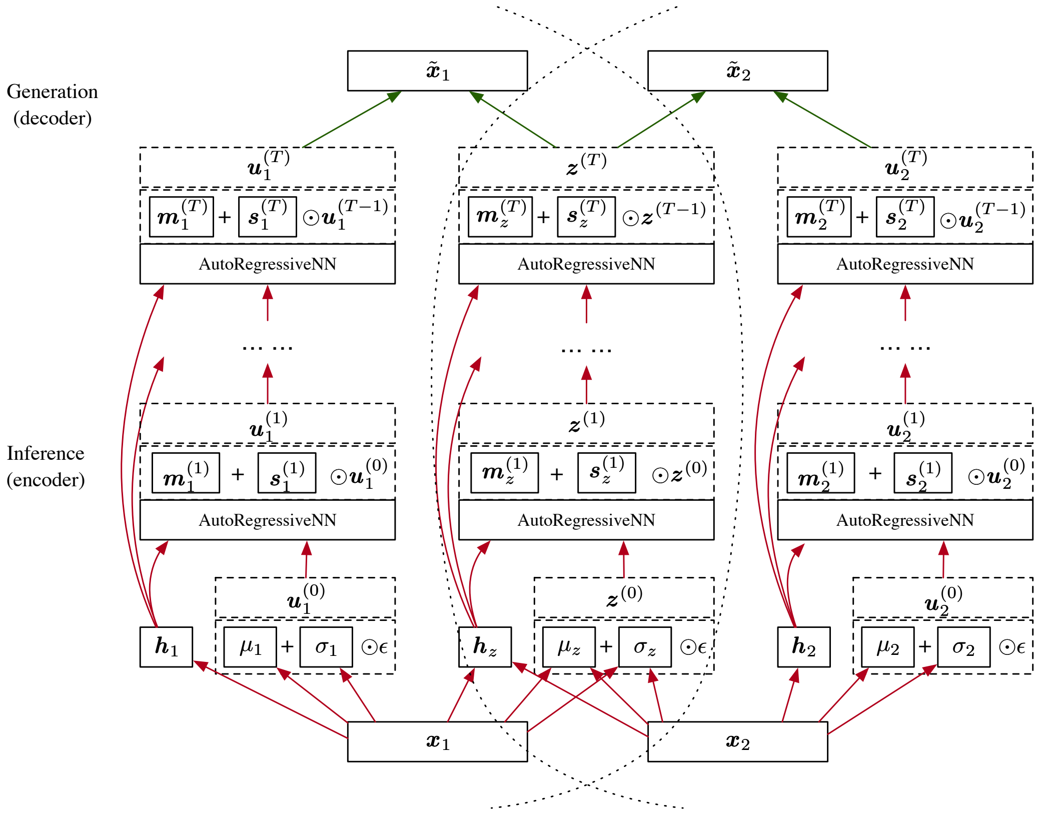
Generation
(decoder)
Inference
(encoder)
$\tilde{x}_1$
$\tilde{x}_2$
$u_1^{(T)}$
$z^{(T)}$
$u_2^{(T)}$
$m_1^{(T)} + s_1^{(T)} \odot u_1^{(T-1)}$
$m_z^{(T)} + s_z^{(T)} \odot z^{(T-1)}$
$m_2^{(T)} + s_2^{(T)} \odot u_2^{(T-1)}$
AutoRegressiveNN
AutoRegressiveNN
AutoRegressiveNN
... ...
... ...
... ...
$u_1^{(1)}$
$z^{(1)}$
$u_2^{(1)}$
$m_1^{(1)} + s_1^{(1)} \odot u_1^{(0)}$
$m_z^{(1)} + s_z^{(1)} \odot z^{(0)}$
$m_2^{(1)} + s_2^{(1)} \odot u_2^{(0)}$
AutoRegressiveNN
AutoRegressiveNN
AutoRegressiveNN
$u_1^{(0)}$
$z^{(0)}$
$u_2^{(0)}$
$h_1$
$\mu_1 + \sigma_1 \odot \epsilon$
$h_z$
$\mu_z + \sigma_z \odot \epsilon$
$h_2$
$\mu_2 + \sigma_2 \odot \epsilon$
$x_1$
$x_2$

Figure 4: Multi-view subspace learning with CCA-Flow. Similar to Figure 2, we use red arrows to denote the inference process and the green arrows to denote the generation process. The inference network for the shared latent $z$ is shown in the middle. When there is view-dropout, one side of the model (separated by dashed curve) is trained.

Firstly, given multi-view observations $x_1$ and $x_2$, we suppose the first latent variable $z^{(0)}$ obeys a diagonal Gaussian distribution, which can be re-parameterized as follows,

$$z^{(0)} = \mu_z + \sigma_z \odot \epsilon,$$

where $\mu_z$ and $\sigma_z$ are functions of the input observations, e.g., a neural network, $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ is a white noise, and $\odot$ is the element-wise product operator.

Additionally, we encode the inputs into a hidden state $\boldsymbol{h}$, also known as the context variable, as a deterministic representation of the inputs.

Next, we use a sequence of inverse autoregressive transformations (IATs), as shown in Figure 3, to build a hierarchy to transform the simple-structured latent variable $z^{(0)}$ into a flexible and complex one. We formulate the transformation from $z^{(t-1)}$ to $z^{(t)}$ as follows:

$$z^{(t)} = \boldsymbol{m}^{(t)} + \boldsymbol{s}^{(t)} \odot z^{(t-1)}. \tag{6}$$

where $\boldsymbol{m}_t, \boldsymbol{s}_t$ are given by an autoregressive neural networks

$$[\boldsymbol{m}_t, \boldsymbol{s}_t] = \text{AutoRegressiveNN}[t](\boldsymbol{h}, \boldsymbol{z}^{(t-1)}).$$

Inspired by the LSTM-type structure, $\boldsymbol{z}^{(t)}$ can be constructed as follows:

$$\boldsymbol{z}^{(t)} = (1 - \text{sigmoid}(\boldsymbol{s}^{(t)})) \odot \boldsymbol{m}^{(t)} + \text{sigmoid}(\boldsymbol{s}^{(t)}) \odot \boldsymbol{z}^{(t-1)}.$$

The key observation is that the Jacobian determinant of each IAT step can be easily computed. So we have the density of the last layer

$$\log q(\boldsymbol{z}^{(T)}|\mathbf{x}) = \log q(\boldsymbol{z}^{(0)}|\mathbf{x}) - \sum_{t=0}^{T} \log \left|\det(\frac{d\boldsymbol{z}^{(t)}}{d\boldsymbol{z}^{(t-1)}})\right| \tag{7}$$

$$= -\sum_{i=1}^{d} \big[\frac{1}{2}\epsilon_i^2 + \frac{1}{2}\log(2\pi) + \sum_{t=0}^{T} \log \boldsymbol{s}_i^{(t)}\big]. \tag{8}$$

### 3.3. Training CCA-Flow by view-dropout

The framework of CCA-Flow is shown in Figure 4. The model includes two parts: the inference network which consists of a first encoder layer and a set of IAF layers, and the generation network to reconstruct the data from the final latent variables.

Now we describe the detailed training method for CCA-Flow. The essential idea of our method is to minimizing the reconstruction error and the distance between projections from different views in the latent space, simultaneously.

To begin with, if ignoring the objective of maximizing correlation between projections, we can minimizing the reconstruction error through our model using all views as inputs. So we can compute the variational lower bound $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}_1, \boldsymbol{x}_2)$, the posterior densities of the latent variables according to Eq.(8). The log-likelihoods $\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i|\boldsymbol{u}_i^{(T)}, \boldsymbol{z}^{(T)})$ are transformed into distances between the original observations $\boldsymbol{x}_i$ to the reconstructed ones $\tilde{\boldsymbol{x}}_i$. For example, for continuous observations, we assume Gaussian distributions for the observations, i.e., $\boldsymbol{x}_i \sim \mathcal{N}(\tilde{\boldsymbol{x}}_i, \delta_i^2\boldsymbol{I})$ , so we can compute the mean-squared error between $\boldsymbol{x}_i$ and $\tilde{\boldsymbol{x}}_i$ as the log-likelihood functions. Similarly, for binary (or categorical) observations, we assume Bernoulli (or categorical) distribution over the observations. So we can add a sigmoid (or softmax) activation before the output layer to let the value of $\tilde{\boldsymbol{x}}_i$ in range $(0, 1)$ and then compute the cross-entropy as the log-likelihood functions.

Finally, with the mentioned re-parameterization tricks, the objective of variational inference

$$\min_{\boldsymbol{\theta},\boldsymbol{\phi}} \quad -\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}_1, \boldsymbol{x}_2), \tag{9}$$

can be minimized using stochastic gradient descent, which is easy to implement using modern deep learning toolboxes.

Next, we perform CCA by randomly dropout some of the views (if there are multiple views). In this way, it pushes the shared inference network on $\boldsymbol{z}$ to be robust when facing different input views. For example, when there are two views, we can formulate a loss function by bypassing one of the views in the input layer, i.e., by minimizing $-\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}_1, \varnothing)$, we can learn to infer the latent variable $\boldsymbol{z}$ by $\boldsymbol{x}_1$, where $\varnothing$ denotes for a masked view-2, e.g., an all-zero vector, or an average vector of $\boldsymbol{x}_2$ in the dataset.

By specifying a dropout rate $\gamma$ for each view, the overall objective function becomes

$$\min_{\boldsymbol{\theta},\boldsymbol{\phi}} \quad -(1-2\gamma)\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\boldsymbol{x}_1,\boldsymbol{x}_2) - \gamma\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\boldsymbol{x}_1,\varnothing) - \gamma\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\varnothing,\boldsymbol{x}_2). \tag{10}$$

Therefore, the model can naturally be trained when there are some sample with missing views.

Since we need to make the model consistent and stable when using view-dropout, the bottom layer $\boldsymbol{z}^{(0)}$ is explicitly modeled as an average of the available views,

$$\boldsymbol{z}^{(0)} = \text{Average}\big(\text{encoder}(\boldsymbol{x}_i)\big). \tag{11}$$

Note that, according to the original CCA, we can add a term to explicitly maximize the correlation of $\mathbb{E}[\boldsymbol{z}|\boldsymbol{x}_1]$ and $\mathbb{E}[\boldsymbol{z}|\boldsymbol{x}_2]$. However, because it adds extra computational complexity to the model and does not bring much gain on the performance, we do not include it in our final model.

## 4. Related work

There are two lines of related work for multi-view learning.

**Deep-learning-based multi-view subspace learning** adopts deep neural networks to learn the shared latent representation of the data. MVAE Ngiam et al. (2011) utilizes auto-encoders to extract shared representations by reconstructing both views from one of them. However, it did not consider the correlation between views. DCCA Andrew et al. (2013), as the deep version of CCA, uses two nets to extract nonlinear features for two views and then maximizes the canonical correlation between the extracted features. Inspired by both CCA and MVAE, DCCAE Wang et al. (2015a) minimizes reconstruction error with auto-encoders and maximizes the canonical correlation between the learned bottleneck representations simultaneously. Some subspace multi-view methods based on the distance like Contrastive Hermann and Blunsom (2013) and DistAE Wang et al. (2015a) are also proposed, but without a generative model to generate missing views.

**Generative multi-view learning** aims to learn a generative model on multi-view data, which is a relative but different field to subspace learning. Multi-view BiGAN (MV-BiGAN), based on Bidirectional Generative Adversarial Networks (BiGAN), Chen and Denoyer (2017) performs density estimation from multi-view inputs. DVCCA Wang et al. (2017) constructs the variational posterior approximation of the shared latent variable from one of the views, which is convenient for inference on the samples with missing views. However, it seeks suboptimal solutions as it can not fully exploit the data. As a modified version of DVCCA, bi-DVCCA Wang et al. (2017) combines two loss functions which are constructed from the two views respectively. Du et al. (2018) proposes a semi-supervised deep generative multi-view model (semiMVAE) which assumes a mixture of Gaussians structure of the variational posterior, utilizing information from multiple views. Among these methods, DVCCA and bi-DVCCA followed the idea of variational CCA to extract the shared latent variables as well as the private latent variables within each view simultaneously.

## 5. Experiments

In this section, we compare different multi-view subspace learning methods which are closely related to our model CCA-Flow on three different tasks.

### 5.1. Datasets

- **Noisy MNIST**[1]: as used in the VCCA paper Wang et al. (2015a), is generated from the MNIST dataset with ten classes. Every sample contains two views: view-1 is an original 28 × 28 image, and view-2 is a noisy image with the same size which is generated by adding noise to a randomly selected image which has the same label with view-1. We use the same data splitting with Wang et al. (2015a), i.e., 50,000 for training, 10,000 for validation, and 10,000 for testing.

- **XRMB dataset**[2]: a speech-articulation data for speech recognition with 39 phone labels, which is also a benchmark dataset for multi-view learning Wang et al. (2015b). Each sample includes two views: 39-dimensional acoustic view and 16-dimensional articulatory view. The data for latent representation learning (35 speakers) is fixed, the rest of the data are used in a 6-fold experiment, i.e., training / tuning / testing with 8 / 2 / 2 speakers, respectively.

- **MIR-Flickr dataset**[3]: A multimodal dataset containing Flickr images, as used in Wang et al. (2017). Each sample consists of two views: 3857-dimensional real-valued handcrafted features and 2000-dimensional textual features (frequent tags). The data has 38 topics, where each sample may be categorized to multiple topics. We use the unlabelled data (Feature Learning 975K) to learn latent representations for labeled data, then use projected features of the labeled data. We use a standard dataset splitting with 10,000 for training, 5,000 for validation, and 10,000 for testing. We evaluate the performance by learning a linear classifier that predicts the labels.

### 5.2. Compared baselines

We compare our method with 9 multi-view subspace learning methods, including one vanilla CCA with linear models, two approximated kernel methods, and six state-of-the-art deep-learning-based methods.

- CCA Hotelling (1992): the classic multi-view subspace learning model with linear models.

- FKCCA Lopez-Paz et al. (2014): the kernel CCA with random Fourier features.

- NKCCA Williams and Seeger (2001): the kernel CCA with Nyström based approximated kernels.

- MAVE Ngiam et al. (2011): the basic deep learning model for multi-view subspace learning. It uses auto-encoders to extract shared representations by reconstructing the observations.

[1] Noisy MNIST: https://ttic.uchicago.edu/∼wwang5/dccae.html

[2] XRMB: https://ttic.uchicago.edu/∼klivescu/XRMB_data

[3] MIR-Flickr: http://www.cs.toronto.edu/∼nitish/multimodal/

Table 1: Performance on the test set. Results marked with $^*$ are from Wang et al. (2017) and results marked with $^+$ are from Wang et al. (2015a).

| Algorithm | noisy MNIST | XRMB | MIR-Flickr |
|---|---|---|---|
| | Error(%) | PER(%) | mAP |
| Original Inputs | $13.1^*$ | $37.6^*$ | $0.48^+$ |
| CCA | $19.6^*$ | $26.7^*$ | $0.529^+$ |
| FKCCA | $5.1^*$ | $26.0^*$ | - |
| NKCCA | $4.5^*$ | $26.6^*$ | - |
| DistAE | $16.0^*$ | $33.2^*$ | - |
| DCCA | $2.9^*$ | $29.0^*$ | $0.573^+$ |
| DCCAE | $2.2^*$ | $24.8^*$ | $0.573^+$ |
| Contrastive | $2.7^+$ | $24.6^+$ | $0.565^+$ |
| MAVE | $11.7^+$ | $29.4^+$ | $0.477^+$ |
| DVCCA | $2.4^+$ | $25.2^+$ | $0.615^+$ |
| bi-DVCCA | - | - | $\mathbf{0.626}^+$ |
| CCA-Flow | **0.89** | **23.3** | 0.619 |

- Contrastive Hermann and Blunsom (2013): the method with a constraint that the distance between the nonlinear latent representations of paired view samples should be smaller than the distance between the nonlinear latent representations of unmatched view samples.

- DCCA Andrew et al. (2013): the deep neural network extension of CCA. It uses two networks to extract nonlinear feature for two views and then maximum the canonical correlation between the extracted features.

- DistAE Wang et al. (2015a): an extension of MVAE that minimizes the distance between the learned projections of the two views and the deep generative multi-view learning reconstruction error between two mappings.

- DCCAE Wang et al. (2015a): an extension of MVAE with two auto-encoders. It maximizes the canonical correlation between the learned bottleneck representations and minimizes the reconstruction errors.

- DVCCA and bi-DVCCA Wang et al. (2017): DVCCA constructs the variational posterior approximation of the shared latent variable from just one view and ignores the rest one; bi-DVCCA combines two lower bounds, for every lower bound, the approximated posterior distribution is constructed from every view.

### 5.3. Experimental settings

We use the experimental setting similar to Wang et al. (2015a, 2017). For each method and dataset, we first conduct unsupervised subspace learning to learn the inference networks. To evaluate the quality the latent representation inferred by different methods on the datasets,

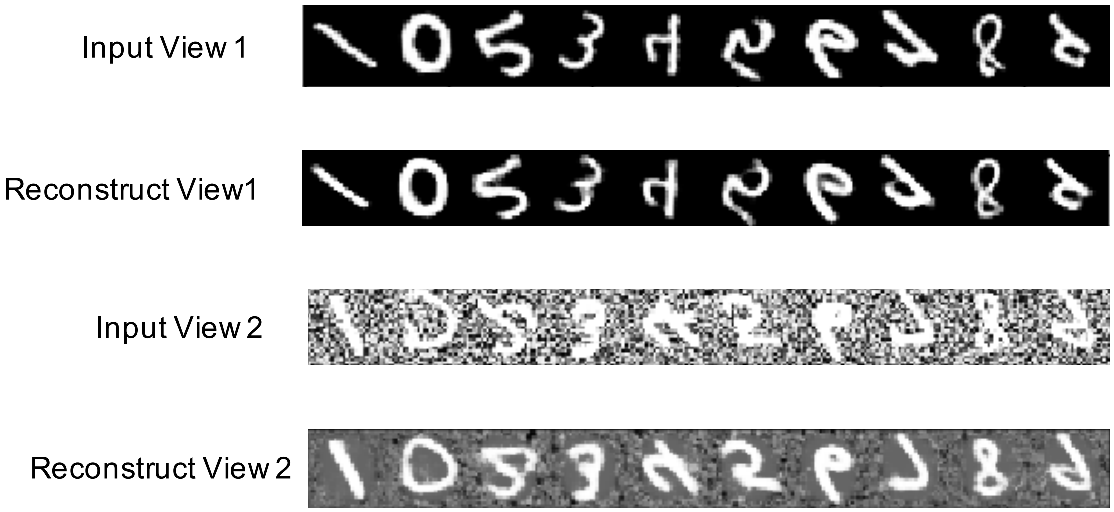

Figure 5: Reconstruction of the test data on noisy MNIST.

we train a linear SVM on the latent representations of train data and use it to classify the latent representations of test data. We adopt the three evaluation metrics: Classification error rate on noisy MNIST, mean phone error rate (PER) over 6 folds on XRMB and mean average precision (mAP) for multi-label classification task on MIR-Flick.

### 5.4. Implementation details

As shown in Figure 4, CCA-Flow consists of an inference network and a generation network. For the first encoding layer and the decoder, we adopt two-layered fully-connected ReLU networks with 128 hidden units. For each IAF step (IAT) as shown in Figure 3, the autoregressive neural networks are two-layered dense networks with 10 units.

For noisy MNIST, the number of IAF steps ($N_{\text{IAF}}$) is set to 2, $d$ is set to 64 and Bernoulli distribution is selected to be the likelihood. For MIR-Flickr, $N_{\text{IAF}} = 2$, $d = 64$, Bernoulli distribution is set for binary view and Gaussian distribution is set for continuous data view. For XRMB, $N_{\text{IAF}} = 3$, $d = 128$ and the Gaussian distribution is selected to be likelihood. We use the Adam optimizer Kingma and Ba (2014) with a learning rate 0.001 to perform stochastic gradient descent. The dropout rate $\gamma$ for each view is set to $\gamma = 0.1$.

### 5.5. Experimental results

All experiment results are shown in Table 1. We use the same data splitting and evaluation metrics with Wang et al. (2015a) and Wang et al. (2017). So in Table 1, results from Wang et al. (2017) are marked with $^*$ and results from Wang et al. (2015a) are marked with $^+$.

From Table 1, we have the following observations.

Firstly, to show the effectiveness of using a generative model, we find that DCCAE performs better than DCCA on these three tasks. It is because DCCAE considers the reconstruction error. Therefore, it shows that training a generative model helps to improve the quality of latent representations.

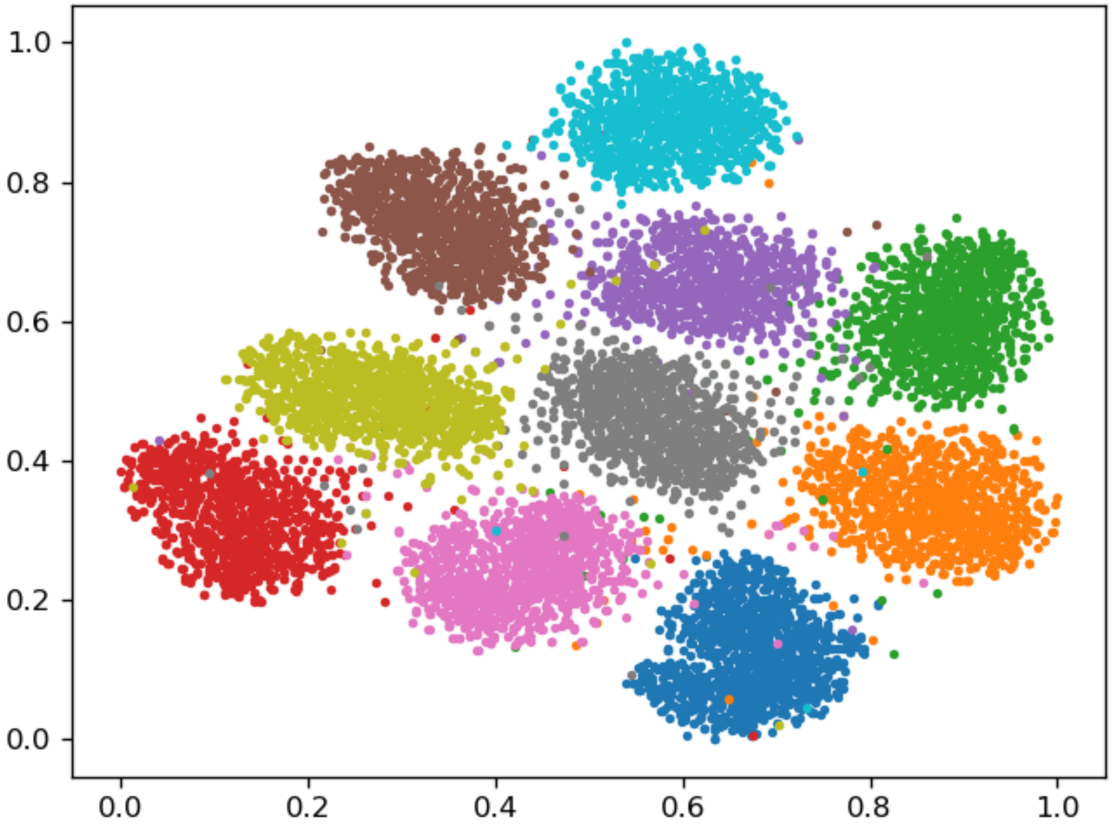

Figure 6: 2-D T-SNE of the shared latent variable for noisy MNIST, on the test data. Each color represents a class. It shows that our model separates the classes even without looking at the labels.

Also we find that DistAE performs poorly on the first two datasets. Although DistAE introduces the reconstruction error in the objective function, it does not model the correlation between the views, which might be the reason for its performance.

DVCCA and our CCF-Flow models both the reconstruction errors and the correlations between views, so they outperforms the previous methods. Moreover, our CCF-Flow uses flow-based inference to extract more flexible representations, which makes it performs the best comparing to all the compared baselines.

To better demonstrate the representation power, Figure 6 shows 2D t-SNE visualizations of the shared latent variables $\boldsymbol{z}^{(T)}$ learned by CCA-Flow. Each color represents one class. We did not use any label information when learning the latent representations. Therefore, in general, CCA-Flow separates the classes well and has a good representation ability of shared latent variables.

Also, Figure 5 shows sample reconstructions by CCA-Flow for the same pair views. We can see the reconstructions are pretty good. Specifically, for the noisy view-2, the results show that it not only achieves a denoising effect on the noisy background, but also make

Table 2: Effect of $N_{\text{IAF}}$ in CCA-Flow

| $N_{\text{IAF}}$ | noisy MNIST<br>Error(%) | XRMB<br>PER(%) | MIR-Flickr<br>mAP |
|---|---|---|---|
| 0 | 2.64 | 27.8 | 0.584 |
| 1 | 1.15 | 25.4 | 0.592 |
| 2 | 0.89 | 24.4 | 0.619 |
| 4 | 1.35 | 24.2 | 0.609 |
| 6 | 1.37 | 24.8 | 0.573 |

Table 3: Effect of $d$ in CCA-Flow

| $d$ | noisy MNIST<br>Error(%) | XRMB<br>PER(%) | MIR-Flickr<br>mAP |
|---|---|---|---|
| 8 | 1.68 | 34.2 | 0.530 |
| 16 | 1.24 | 28.9 | 0.586 |
| 32 | 1.12 | 26.9 | 0.591 |
| 64 | 0.89 | 24.2 | 0.619 |
| 128 | 0.97 | 23.3 | 0.598 |

the digits easier to identify. For example, the “0” in the second input image of view-2 is not a complete circle, but the reconstructed “0” is completed. Therefore, it shows that the model can successfully transfer the knowledge from one view to another.

### 5.6. Parameter study

We study the performance change of CCA-Flow with two important parameters, the number of IAF layers $N_{\text{IAF}}$, and the size of the latents $d$. Performances change when $N_{\text{IAF}}$ varies on all datasets. From the results in Table 2, we can see that when we do not use any IAF layer, the model performs the worst (in this case, it is actually the original VCCA model). It verified the effectiveness of using IAF to improve the inference quality. Also, we find that different datasets prefer different number of IAF steps. The best value of $N_{\text{IAF}}$ for MNIST and MIR-Flickr is two while for XRMB is four. It would be better to choose $N_{\text{IAF}}$ from range $[2, 4]$. The dimension of the shared latent representations $d$ also influences the performance of CCA-Flow. From the results shown in Table 3, we find different dataset prefers different choices of $d$. A robust choice of $d$ ranges from 64 to 128.

## 6. Conclusion

We propose CCA-Flow, a novel method for multi-view subspace learning that achieves strong representation power and efficient learning. Our model is based on a basic autoencoder structure and a variational CCA loss function. To obtain a sufficiently flexible and arbitrarily complex approximated posterior on the latent variables, we introduce a hierarchy of inverse autoregressive flow as the inference network. We use Monte Carlo variational inference to efficiently train the model with stochastic gradient descent. Furthermore, we use view-dropout to implicitly maximize the correlation of projections from different views in the shared latent space. In this way, the model can make inference at the samples with missing views. Experiments showed that our model achieved state-of-the-art performance on three benchmark multi-view datasets.

## Acknowledgments

The research work was supported by the National Key Research and Development Program of China under Grant No. 2018YFB1004300, the National Natural Science Foundation of China under Grant NOs. U1836206, U1811461, 61773361, the Project of Youth Innovation Promotion Association CAS under Grant No. 2017146.

## References

Shotaro Akaho. A kernel method for canonical correlation analysis. *IMPS*, 40(2):263–269, 2001.

Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *International conference on machine learning*, pages 1247–1255. ACM, 2013.

Francis R Bach and Michael I Jordan. A probabilistic interpretation of canonical correlation analysis. 2005.

Mickaël Chen and Ludovic Denoyer. Multi-view generative adversarial networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 175–188. Springer, 2017.

Changde Du, Changying Du, Hao Wang, Jinpeng Li, Wei-Long Zheng, Bao-Liang Lu, and Huiguang He. Semi-supervised deep generative modelling of incomplete multi-modality emotional data. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 108–116. ACM, 2018.

Jia He, Changying Du, Changde Du, Fuzhen Zhuang, Qing He, and Guoping Long. Nonlinear maximum margin multi-view learning with adaptive kernel. In *IJCAI*, pages 1830–1836. Morgan Kaufmann, 2017.

Karl Moritz Hermann and Phil Blunsom. Multilingual distributed representations without word alignment. *arXiv preprint arXiv:1312.6173*, 2013.

Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

Harold Hotelling. Relations between two sets of variates. In *Breakthroughs in statistics*, pages 162–190. Springer, 1992.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems*, pages 4743–4751. MIT press, 2016.

Pei Ling Lai and Colin Fyfe. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10(05):365–377, 2000.

David Lopez-Paz, Suvrit Sra, Alex Smola, Zoubin Ghahramani, and Bernhard Schölkopf. Randomized nonlinear component analysis. In *International Conference on Machine Learning*, pages 1359–1367. ACM, 2014.

Ang Lu, Weiran Wang, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Deep multilingual correlation for improved word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 250–256. ACL, 2015.

Thomas Melzer, Michael Reiter, and Horst Bischof. Nonlinear feature extraction using generalized canonical correlation analysis. In *International Conference on Artificial Neural Networks*, pages 353–360. Springer, 2001.

Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng. Multimodal deep learning. In *ICML*, pages 689–696. ACM, 2011.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *The 32nd International Conference on Machine Learning*, pages 1530–1538. ACM, 2015.

Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *International Conference on Machine Learning*, pages 1083–1092. ACM, 2015a.

Weiran Wang, Raman Arora, Karen Livescu, and Jeff A Bilmes. Unsupervised learning of acoustic features via deep canonical correlation analysis. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4590–4594. IEEE, 2015b.

Weiran Wang, Xinchen Yan, Honglak Lee, and Karen Livescu. Deep variational canonical correlation analysis. *arXiv preprint arXiv:1610.03454*, 2017.

Christopher KI Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Advances in neural information processing systems*, pages 682–688. MIT press, 2001.

Fei Yan and Krystian Mikolajczyk. Deep correlation for matching images and text. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3441–3450. IEEE, 2015.