Deep Co-Attention Network for Multi-View Subspace Learning

Lecheng Zheng[1], Yu Cheng[2], Hongxia Yang[3], Nan Cao[4] and Jingrui He[1]
[1]University of Illinois at Urbana-Champaign, {lecheng4, jingrui}@illinois.edu;
[2]Microsoft AI, chengyu05@gmail.com;
[3]Alibaba Group, firewater1984@gmail.com;
[4] Tongji University, nan.cao@gmail.com

ABSTRACT

Many real-world applications involve data from multiple modalities and thus exhibit the view heterogeneity. For example, user modeling on social media might leverage both the topology of the underlying social network and the content of the users' posts; in the medical domain, multiple views could be X-ray images taken at different poses. To date, various techniques have been proposed to achieve promising results, such as canonical correlation analysis based methods, etc. In the meanwhile, it is critical for decision-makers to be able to understand the prediction results from these methods. For example, given the diagnostic result that a model provided based on the X-ray images of a patient at different poses, the doctor needs to know why the model made such a prediction. However, state-of-the-art techniques usually suffer from the inability to utilize the complementary information of each view and to explain the predictions in an interpretable manner.

To address these issues, in this paper, we propose a deep co-attention network for multi-view subspace learning, which aims to extract both the common information and the complementary information in an adversarial setting and provide robust interpretations behind the prediction to the end-users via the co-attention mechanism. In particular, it uses a novel cross reconstruction loss and leverages the label information to guide the construction of the latent representation by incorporating the classifier into our model. This improves the quality of latent representation and accelerates the convergence speed. Finally, we develop an efficient iterative algorithm to find the optimal encoders and discriminator, which are evaluated extensively on synthetic and real-world data sets. We also conduct a case study to demonstrate how the proposed method robustly interprets the predictions on an image data set.

KEYWORDS

Multi-view Learning, Attention Mechanism, Interpretable Machine Learning

1 INTRODUCTION

In many real-world applications, data are usually collected from multiple sources or modalities, exhibiting the view heterogeneity. For example, many images posted on Facebook or Twitter are usually surrounded by the text descriptions, both of which can be considered as two distinct views; in the face attribute classification problem, the data consist of different poses of the same person, and each pose can be considered as a single view with complementary information to each other; in stock price forecasting, the related factors include not only historical stock prices and financial statements from companies, but also news, weather, etc.

Up to now, many researchers have proposed various techniques to model the view heterogeneity based on different assumptions. Some works assume that there exists a latent lower-dimensional subspace shared by multiple views, and these views can be reconstructed from this subspace. However, these state-of-the-art methods usually suffer from the inability to utilize the complementary information of each view and the label information to enhance the quality of the representation. For example, canonical correlation analysis (CCA) [13], kernel canonical correlation analysis (KCCA) [29], deep canonical correlation auto-encoder (DCCAE)[35] aim to explore the linear/non-linear transformation of multi-view features by maximizing the correlation coefficient of the two views. However, in addition to not utilizing the label information, the major drawback of [13, 29, 33, 35] is that they ignore the complementary information in the data since the CCA based methods target to extract the common information that is mostly correlated between views, which might result in a sub-optimal solution.

Recent years have witnessed the tremendous efforts devoted to developing interpretable learning algorithms [9, 15, 22, 25, 27, 47]. Understanding the reasons behind the prediction is of key importance for those who plan to take action based on the prediction. For example, in the medical domain, a doctor expects to know why the model made a prediction of a potential disease given the X-ray images of a patient taken at different poses; in the financial domain, when investors utilize a model to predict the trend of the stock price, they expect to see the reasons behind the prediction so that they could analyze a large number of assets to form a diversified portfolio and mitigate risks in a reasonable way. However, in multi-view learning, most existing techniques ignore the interpretability of the predictive model. Although we could simply apply the existing explanation methods on the concatenated multi-view features to interpret the result, it may suffer from the noisy features by accidentally including such features for interpretation. On the other hand, utilizing the consensus information of multi-view data could potentially help us build a more robust interpretable model against the noisy features.

Motivated by these limitations, we propose a deep adversarial co-attention model for multi-view subspace learning named *ANTS*, extracting both the shared information and the view-specific information in an adversarial manner and providing the robust interpretation behind the prediction to the end-users. Within this model, the co-attention encoder module is designed to extract the common information shared by multiple views with the help of a view discriminator, and interpret the predictive results by weighing the importance of the input features; the decoder module is designed to project the common information back to the original input space; the view discriminator is included to regularize the generation quality of the shared representation. After reconstructing the multi-view features based on the shared representation, we use the residual between the original input features and the reconstructed features as the view-specific information or the complementary information. We integrate both the common information and complementary information to yield richer and more discriminative feature representation. Our main contributions are summarized below:

- A novel deep model for multi-view subspace learning, which extracts both the common information and the complementary information, and provides a robust model explanation for each view via co-attention module.
- Novel cross reconstruction loss and the use of label information to guide the construction of the latent representation.
- A case study that show how the proposed method interprets the predictions on an image data set.
- Experimental results on synthetic and real-world data sets, which demonstrate the effectiveness of the proposed model.

The rest of this paper is organized as follows. After a brief review of the related work in Section 2, we introduce our proposed model for deep multi-view subspace learning in Section 3. The systematic evaluation of the proposed method on synthetic and real-world data sets is presented in Section 4. In Section 5, we conduct a case study to show how the proposed method interprets the prediction on an image data set before we conclude the paper in Section 6.

2 RELATED WORK

In this section, we briefly review the related work on multi-view learning, interpretable Learning, adversarial learning, and cycle-consistency.

2.1 Multi-view Learning and Interpretable Learning

Learning multi-view data has been studied for decades. In multi-view learning [38], researchers aim to model the similarity and difference among multiple views. The existing algorithms can be classified into three categories: 1) co-training based methods, 2) multiple kernel learning, and 3) subspace learning. Co-training [2] is one of the earliest methods proposed for multi-view learning, which aims to find the maximal consistency of several independent views given only a few labeled examples and many unlabeled ones. Since then, many variants of co-training have been proposed to find the consistency among views. For example, [39] proposed an undirected graphical model for co-training to minimize the disagreement among multi-view classifiers. In multiple kernel learning, [5] proposed a two-view Support Vector Machine method (SVM-2K) to find multiple kernels to maximize the correlation of the two views; [28] proposed a co-regularization method to jointly regularize two Reproducing Kernel Hilbert Spaces associated with the two views; [1] proposed Deep Canonical Correlation Analysis to find two deep networks such that the output layers of the two networks are maximally correlated. As for the subspace learning, the authors of [14] proposed a deep multi-view robust representation learning algorithm based on auto-encoder to learn a shared representation from multi-view observations;[11] proposed online Bayesian subspace multi-view learning by modeling the variational approximate posterior inferred from the past samples; [51] proposed M2VW for multi-view multi-worker learning problem by leveraging the structural information between multiple views and multiple workers; [30] proposed CR-GAN method to learn a complete representation for multi-view generations in the adversarial setting by the collaboration of two learning pathways in a parameter-sharing manner. Different from [30, 41–44, 50–53], in this paper, we focus on multi-view classification problem and aim to extract both the shared information and the view-specific information in the adversarial setting, and the view consistency constraint with label information is utilized to further regularize the generated representation in order to improve the predictive performance.

Recently, more and more studies on model explanation [9, 15, 22, 25, 27, 47] reveal a surge of research interest in the model interpretation. [25] is one of the earliest works in the model interpretation, which proposes the LIME algorithm to explain the predictions of any model in an interpretable manner. In [15], the authors propose a black-box explanation algorithm to interpret how a training example influences the parameters of a model; in [47], the authors propose a domain adaptive attention network to explore the relatedness of multiple tasks and leverage consistency of multi-modality financial data to predict stock price. In this paper, we leverage the co-attention mechanism to interpret the prediction by weighting the importance of the input features.

2.2 Adversarial Learning and Cycle Consistency

Adversarial learning is a technique attempting to fool models through malicious input, which is a promising way to train robust deep networks, and can generate complex samples across diverse domains [8, 17, 23, 45, 46, 54]. The generative adversarial networks (GANs) [8] is an effective approach to estimate intractable probabilities, learned by playing a min-max game between generator and discriminator. [23] applied both reconstruction error and adversarial training criteria to a traditional auto-encoder. The MMD-GAN work [17] introduced the adversarial kernel learning technique for the discriminator loss. Recently, some adversarial learning approaches [4, 6, 18, 31, 40] have been proposed to minimize the distance between feature distributions. The domain-adaption works in [6, 31, 40] tried to learn a domain-invariant representation in adversarial settings. In [4], the adversarially learned inference model aimed to find the shared latent representations of both views. [18] proposed an adversarial multi-task learning framework to separate the shared and private features of multiple tasks. Different from these methods solely focusing on aligning the global marginal distribution by fooling a domain discriminator, we explore to further align the learned representation by considering the label information. On the other hand, the cycle-consistency or the idea of using transitivity to regularize structured data has been applied in many applications, including image matching [49], co-segmentation [32], style transfer [55], etc. In [7, 48], the cycle consistency constraint is utilized as a regularizer to push the mappings to be as consistent with each other as possible in the supervised convolution neural network training. In [55], the authors proposed the Cycle-Consistent generative adversarial network (Cycle-GAN) to learn two mappings or generators and two discriminators by using transitivity to supervise CNN training. Different from Cycle-GAN [55], our paper mainly focuses on finding a latent representation shared by multiple views by leveraging the label information and extending the cycle consistency idea, rather than transforming data from one domain to another domain.

3 PROPOSED *ANTS* FRAMEWORK

In this section, we present our proposed *ANTS*, a deep co-attention multi-view subspace learning framework. We start by introducing the major notation used in this paper and then discuss the extraction of both the shared latent information and the view-specific information, as well as how to leverage the label information and incorporate the classification model. Notice that in the subsequent discussion, for the sake of exposition, we focus on multi-view learning with two views. The extension of the proposed techniques to the more generic settings with multiple views is discussed in Subsection 3.4.

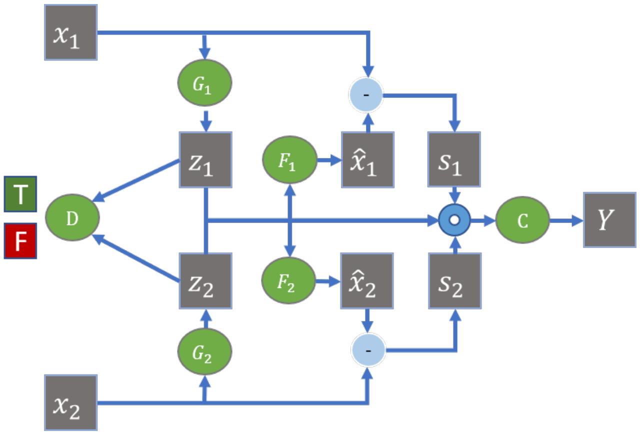


Figure 1: Given an example drawn from a data set with two views x_1 and x_2, we use two encoders $G_1(\cdot)$ and $G_2(\cdot)$ to generate the shared latent representation z. The goal of the discriminator $D(\cdot)$ is to determine whether the shared latent representation z is generated by the first view (True) or the second view (False). To constrain the generated representation, two decoders $F_1(\cdot)$ and $F_2(\cdot)$ are utilized to transform the shared representation (either z_1 or z_2) back to two recovered views $\hat{x}_1$ and $\hat{x}_2$, and we want the recovered views to be as similar to the real views as possible. Meanwhile, we extract the view-specific information s_1 and s_2 by simply subtracting $\hat{x}_1$ from x_1 and subtracting $\hat{x}_2$ from x_2, respectively. After extracting both the shared information and the view-specific information, we concatenate them in a hidden layer and feed them into a classifier $C(\cdot)$. Finally, label information Y is utilized to enforce the encoders to create better representations in order to improve the predictive performance of the classifier.

3.1 Notation

Throughout this paper, we use lower-case letters for scalars (e.g., γ), a bold lower-case letter for a sample (e.g., $\boldsymbol{x^j}$), and a bold upper-case letter for a matrix (e.g., $\boldsymbol{X}$). We use $\mathcal{D} = (\boldsymbol{X_1}, \ldots, \boldsymbol{X_v}, Y)$ to denote a data set, where $\boldsymbol{X_i} \in \mathbb{R}^{n \times l_i}$ is the feature for the i^{th} view, $Y \in \mathbb{R}^{n \times c}$ is the binary label matrix, n is the number of samples, l_i is the dimension of the input feature, v is the number of views which is set to be 2 if not specified, and c is the number of classes. For a single sample $\boldsymbol{X_i(j,:)} \in \mathbb{R}^{l_i}$, if it is image data, we could apply image partition algorithm, e.g., Mask R-CNN [12], to split this image into multiple segments; if it is text data, we could use Word2vec [24] to extract the input feature. Following this idea, we could reshape the dimension of a sample, e.g., $\boldsymbol{x_i^j} \in \mathbb{R}^{d_i \times k_i}$, where $k_i \times d_i = l_i$, $\boldsymbol{x_i^j} = \{\boldsymbol{x_{i,1}^j}; \boldsymbol{x_{i,2}^j}; ...; \boldsymbol{x_{i,k_i}^j}\}$ and $\boldsymbol{x_{i,k_i}^j} \in \mathbb{R}^{d_i}$. For image data, k_i is the number of image segments and d_i is the dimension of the representation of each segment (for text data, k_i is the number of words and d_i is the dimension of word embedding). We denote $G_i(\cdot)$ and $F_i(\cdot)$ as an encoder-decoder pair for the i^{th} view, one for projecting the data point to the shared subspace and another for transferring the projection back to the original data point. We also denote $D(\cdot)$ as the discriminator, $C(\cdot)$ as the classifier and $P_{g_i}(\boldsymbol{X_i})$ as a prior on the i^{th} view. Given a sample x^j, we denote $\boldsymbol{z} \in \mathbb{R}^h$ as the shared representation generated by encoders, $\boldsymbol{\hat{x}_i^j} \in \mathbb{R}^{d_i \times k_i}$ as the i^{th} the recovered view generated by $F_i(\cdot)$, $\boldsymbol{z^j} \in \mathbb{R}^h$ as the representation generated from a sample x_i^j by $G_i(\cdot)$, where h is the dimension of the hidden representation. Note that $\hat{Y} \in \mathbb{R}^{n \times c}$ is the prediction made by the classifier $C(\cdot)$ for labeled data, and $\hat{Y}_i \in \mathbb{R}^{n \times c}$ is the prediction of the i^{th} view for unlabeled data.

3.2 Objective Function

Now, we are ready to introduce the overall objective function:

$$\min_{G_1,G_2,F_1,F_2,C} \max_{D} L = L_0(G_1, G_2, D) + \alpha L_v(G_1, G_2, F_1, F_2) \\ + \beta L_c(C, G_1, G_2, F_1, F_2) \tag{1}$$

where α and β are two positive constants that balance the three terms in the objective function: L_0 is the objective function of the min-max game, L_v is the cross reconstruction loss to regularize the generation of the representation and L_c is the cross-entropy loss by leveraging the label information. Figure 1 provides an overview of the proposed framework. Next, we elaborate on each module respectively.

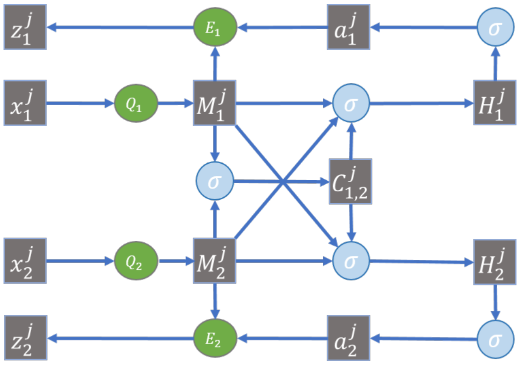


Figure 2: Co-attention encoder module, where Q_1, Q_2, E_1 and E_2 are neural networks, and σ is the non-linear transformation function, e.g., tanh or softmax.

3.2.1 *Co-attention Encoder Module.*

The main idea of co-attention encoder is to explore the correlation of each pair of features in two views in order to select the most important features by weighing the importance of the input features via co-attention mechanism [19, 21]. Please refer to Figure 2 for the network structure of the co-attention encoder module. Suppose we are given a sample $\boldsymbol{x^j}$ with two views, the affinity matrix $\boldsymbol{C}_{1,2}^{j} \in \mathbb{R}^{k_1 \times k_2}$ of these two views is computed by:

$$\begin{aligned} \boldsymbol{M}_1^j &= Q_1(\boldsymbol{x}_1^j), \quad \boldsymbol{M}_2^j = Q_2(\boldsymbol{x}_2^j) \\ \boldsymbol{C}_{1,2}^j &= \tanh\left((\boldsymbol{M}_1^j)^T \boldsymbol{W}_{1,2} \boldsymbol{M}_2^j\right) \end{aligned} \tag{2}$$

where $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, $\boldsymbol{M_1} \in \mathbb{R}^{d_1 \times k_1}$, $\boldsymbol{M_2} \in \mathbb{R}^{d_2 \times k_2}$ are two hidden representations mapped by two neural networks (Q_1 and Q_2) and $\boldsymbol{W}_{1,2} \in \mathbb{R}^{d_1 \times d_2}$ is a weight matrix shared by all samples. Basically, this affinity matrix measures the similarities between the segments among two views. By leveraging such an affinity matrix of two views, we aim to encode the information from both views in the hidden representation H_1 and H_2 that could be formulated as follows:

$$\begin{aligned} H_1^j &= \tanh\left(\boldsymbol{W_1}\boldsymbol{M}_1^j + (\boldsymbol{W_2}\boldsymbol{M}_2^j)(\boldsymbol{C}_{1,2}^j)^{\boldsymbol{T}}\right) \\ H_2^j &= \tanh\left(\boldsymbol{W_2}\boldsymbol{M}_2^j + (\boldsymbol{W_1}\boldsymbol{M}_1^j)\boldsymbol{C}_{1,2}^j\right) \end{aligned} \tag{3}$$

where $\boldsymbol{W_1} \in \mathbb{R}^{d_3 \times d_1}$ and $\boldsymbol{W_2} \in \mathbb{R}^{d_3 \times d_2}$ are weight matrices and d_3 is the output dimension. By encoding the information from both views into two hidden representations H_1^j and H_2^j, we aim to capture the consensus information of two views so that we could build a more robust interpretable model to mitigate the negative impact of noisy features. The importance of the input features is measured as follows:

$$\boldsymbol{a}_1^j = \text{softmax}\,(\boldsymbol{w_{h1}}\boldsymbol{H}_1^j), \quad \boldsymbol{a}_2^j = \text{softmax}\,(\boldsymbol{w_{h2}}\boldsymbol{H}_2^j) \tag{4}$$

where $\boldsymbol{w_{h1}} \in \mathbb{R}^{d_3}$ and $\boldsymbol{w_{h2}} \in \mathbb{R}^{d_3}$ are weight vectors. Based on the above attention weight vectors $\boldsymbol{a}_1^j \in \mathbb{R}^{k_1}$ and $\boldsymbol{a}_2^j \in \mathbb{R}^{k_2}$, the extracted representation for the first view and the second view can be calculated by:

$$\boldsymbol{z}_1^j = E_1\Big(\sum_{l=1}^{k_1} \boldsymbol{a}_{1,l}^j \boldsymbol{M}_{1,l}^j\Big), \quad \boldsymbol{z}_2^j = E_2\Big(\sum_{l=1}^{k_2} \boldsymbol{a}_{2,l}^j \boldsymbol{M}_{2,l}^j\Big) \tag{5}$$

where $E_1(\cdot)$ and $E_2(\cdot)$ are two neural networks that map two views to a latent space denoted as $\boldsymbol{z}$. Here, we denote the entire structure of the co-attention module as $G_1(\cdot)$ and $G_2(\cdot)$ for the first and second view, respectively (e.g., $z_i = G_i(x_i)$). In the next subsection, we introduce the discriminator module to help encoders extract the common representation shared by two views.

3.2.2 Discriminator Module for Extraction of Shared Representation. The discriminator module aims to help the encoder module find the latent representation shared by two views, which will be utilized for training a high-quality classifier. To learn the distribution of the two encoders P_{g_1} and P_{g_2} over the shared latent space $\boldsymbol{z}$, we define $P_{g_1}(\boldsymbol{X_1})$ to be a prior on the first view $\boldsymbol{X_1}$ and $P_{g_2}(\boldsymbol{X_2})$ to be a prior on the second view $\boldsymbol{X_2}$. Thus, $\boldsymbol{z} \sim P_{g_1}(\boldsymbol{X_1})$ is produced by the first data encoder $G_1(\cdot)$ fed with view $\boldsymbol{X_1}$; at the same time, $\boldsymbol{z} \sim P_{g_2}(\boldsymbol{X_2})$ is produced by the second data encoder $G_2(\cdot)$ fed with view $\boldsymbol{X_2}$. Let $D(\cdot)$ be the discriminator that distinguishes whether the shared latent representation $\boldsymbol{z}$ is generated by the first encoder $G_1(\boldsymbol{x_1})$ or by the second encoder $G_2(\boldsymbol{x_2})$. The goal of the discriminator $D(\cdot)$ is to maximize the probability of assigning the correct labels to the representation generated by both encoders, e.g., assigning true to the representation generated by the first view and false to the representation generated by the second view. Two encoders aim to minimize the probability that the discriminator $D(\cdot)$ successfully distinguishes the representation generated from the first view or the second view. Thus, the min-max objective function can be formulated by:

$$\begin{aligned} \min_{G_1,G_2} \max_{D} L_0 &= \mathbb{E}_{\boldsymbol{x_1} \sim p(X_1)}[\log(D(G_1(\boldsymbol{x_1})))] \\ &+ \mathbb{E}_{\boldsymbol{x_2} \sim p(X_2)}[\log(1 - D(G_2(\boldsymbol{x_2})))] \\ &= \mathbb{E}_{\boldsymbol{z} \sim P_{g_1}(X_1)}[\log(D(\boldsymbol{z}))] \\ &+ \mathbb{E}_{\boldsymbol{z} \sim P_{g_2}(X_2)}[\log(1 - D(\boldsymbol{z}))] \end{aligned} \tag{6}$$

In this min-max game, two encoders $G_1(\cdot)$ and $G_2(\cdot)$ aim to generate the same latent representation $\boldsymbol{z}$ shared by both views so that the discriminator $D(\cdot)$ cannot reliably decide which encoder creates such a latent representation.

PROPOSITION 1. *For the fixed encoders G_1 and G_2, the optimal discriminator $D^*(\boldsymbol{z})$ of Equation 1 is given by $D^*(\boldsymbol{z}) = \frac{P_{g_1}}{P_{g_1}+P_{g_2}}$.*

PROOF. Given the fixed encoder G_1 and G_2, the training criterion for the discriminators is to maximize
$L_0 = \int_z P_{g_1} \log(D(\boldsymbol{z}))dz + \int_z P_{g_2} \log(1 - D(\boldsymbol{z}))dz$
$L_0 = \int_z [P_{g_1} \log(D(\boldsymbol{z})) + P_{g_2} \log(1 - D(\boldsymbol{z}))]dz$
For any (a,b) $\in \mathbb{R}^2 \setminus (0,0)$, the function $f(c) = a\log(c) + b(\log(1-c))$ achieves its maximum at $c = \frac{a}{a+b}$. Thus, $D^*(\boldsymbol{z}) = \frac{P_{g_1}}{P_{g_1}+P_{g_2}}$. □

3.2.3 Decoder Module for Extraction of View-specific Representation. In this subsection, we propose to reconstruct the views based on the extracted shared representation to enforce the shared representation to be robust and only contain the common information. Besides, subtracting the reconstructed data from the original input features leaves us with the information specific to each view, which could be utilized to further boost the predictive performance. To reconstruct the views, we exploit the idea of transitivity to regularize structured data used in [55] in order to restrict the distribution of the representation. Our main intuition is that we could find the latent representation shared by the two views in a low-dimensional space and reconstruct these two views from this shared representation generated by either view. In other words, since the representation $\boldsymbol{z}$ is shared by two views, if we use the encoder $G_2(\boldsymbol{x_2})$ to generate the shared representation $\boldsymbol{z}$, the decoder $F_1(\boldsymbol{z})$ should be able to transform the representation $\boldsymbol{z}$ back to the first view $\boldsymbol{x_1}$. Similarly, the decoder $F_2(\boldsymbol{z})$ should also be able to transform the representation $\boldsymbol{z}$ generated by the encoder $G_1(\boldsymbol{x_1})$ back to the second view $\boldsymbol{x_2}$. Therefore, we let $\hat{\boldsymbol{x}}_i = \gamma_1 \hat{\boldsymbol{x}}_{(i,1)} + \gamma_2 \hat{\boldsymbol{x}}_{(i,2)}$, where $\gamma_1 + \gamma_2 = 1$, $\hat{\boldsymbol{x}}_{(i,j)}$ means the reconstruction of the i^{th} view from the j^{th} view, and $\gamma_1, \gamma_2 \in [0,1]$ are positive learnable parameters that balance between the two reconstructed views. In this way, the loss function can be updated as follows:

$$\begin{aligned} L_v &= \mathbb{E}_{\boldsymbol{x_1} \sim p(X_1)}\, \gamma_1 \|\boldsymbol{x_1} - \hat{\boldsymbol{x}}_1\|_2^2 + \mathbb{E}_{\boldsymbol{x_2} \sim p(X_2)}\, \gamma_2 \|\boldsymbol{x_2} - \hat{\boldsymbol{x}}_2\|_2^2 \\ &= \mathbb{E}_{\boldsymbol{x_1} \sim p(X_1)}\, \gamma_1 \|\gamma_1 \boldsymbol{x_1} + \gamma_2 \boldsymbol{x_1} - \gamma_1 \hat{\boldsymbol{x}}_{(1,1)} - \gamma_2 \hat{\boldsymbol{x}}_{(1,2)}\|_2^2 \\ &+ \mathbb{E}_{\boldsymbol{x_2} \sim p(X_2)}\, \gamma_2 \|\gamma_1 \boldsymbol{x_2} + \gamma_2 \boldsymbol{x_2} - \gamma_2 \hat{\boldsymbol{x}}_{(2,2)} - \gamma_1 \hat{\boldsymbol{x}}_{(2,1)}\|_2^2 \\ &= \mathbb{E}_{(\boldsymbol{x}_1,\boldsymbol{x}_2) \sim p(X_1,X_2)}\, \gamma_1 \|\gamma_1(\boldsymbol{x_1} - \hat{\boldsymbol{x}}_{(1,1)}) + \gamma_2(\boldsymbol{x_1} - \hat{\boldsymbol{x}}_{(1,2)})\|_2^2 \\ &+ \mathbb{E}_{(\boldsymbol{x}_1,\boldsymbol{x}_2) \sim p(X_1,X_2)}\, \gamma_2 \|\gamma_2(\boldsymbol{x_2} - \hat{\boldsymbol{x}}_{(2,2)}) + \gamma_1(\boldsymbol{x_2} - \hat{\boldsymbol{x}}_{(2,1)})\|_2^2 \\ &\leq 2\,\mathbb{E}_{(\boldsymbol{x}_1,\boldsymbol{x}_2) \sim p(X_1,X_2)}[\gamma_1^3 \|\boldsymbol{x_1} - \hat{\boldsymbol{x}}_{(1,1)}\|_2^2 + \gamma_2^3 \|\boldsymbol{x_2} - \hat{\boldsymbol{x}}_{(2,2)}\|_2^2 \\ &+ \gamma_1 \gamma_2^2 \|\boldsymbol{x_1} - \hat{\boldsymbol{x}}_{(1,2)}\|_2^2 + \gamma_2 \gamma_1^2 \|\boldsymbol{x_2} - \hat{\boldsymbol{x}}_{(2,1)}\|_2^2] \end{aligned}$$

where the last inequality is based on $||a+b||_2^2 \leq 2||a||_2^2+2||b||_2^2, \forall a, b \in \mathbb{R}$. Based on the above analysis, we define the view reconstruction loss L_v in the overall objective function as follows:

$$L_v = \mathbb{E}_{(\boldsymbol{x_1},\boldsymbol{x_2})\sim p(X_1,X_2)} [\gamma_1^3 ||\boldsymbol{x_1} - \hat{\boldsymbol{x}}_{(1,1)}||_2^2 + \gamma_2^3 ||\boldsymbol{x_2} - \hat{\boldsymbol{x}}_{(2,2)}||_2^2 + \gamma_1\gamma_2^2 ||\boldsymbol{x_1} - \hat{\boldsymbol{x}}_{(1,2)}||_2^2 + \gamma_2\gamma_1^2 ||\boldsymbol{x_2} - \hat{\boldsymbol{x}}_{(2,1)}||_2^2] \tag{7}$$

One advantage of this constraint is that it enforces the representations $\mathbf{z_1}$ and $\mathbf{z_2}$ generated by two encoders $G_1(\cdot)$ and $G_2(\cdot)$ to be as similar as possible in order to minimize the reconstruction error, which helps the encoders to find the shared representation at a faster pace. After extracting the common information shared by multiple views, we also want to extract the view-specific representation to further boost the predictive performance because the view-specific information contains complementary features for the sake of classification. For each sample x_i, the view-specific information $\boldsymbol{s_i}$ for the i^{th} view could be easily extracted by subtracting the constructed view from the original view, written as follows:

$$\boldsymbol{s_i} = \boldsymbol{x_i} - \hat{\boldsymbol{x}}_i = \boldsymbol{x_i} - \gamma_1\hat{\boldsymbol{x}}_{(i,1)} - \gamma_2\hat{\boldsymbol{x}}_{(i,2)} \tag{8}$$

To leverage both the shared and complementary information from all the views for the downstream classification task, we propose to concatenate both common information and complementary information from all views to make final prediction.

3.2.4 Classification Module for Incorporating Label Information. With both the shared information and the view-specific information, we propose to incorporate the classifier in our model to enhance the quality of the concatenated representation. As mentioned earlier, one drawback of CCA based methods is that they only capture the information learned from the features but ignore the useful information from the labels. Nevertheless, the label information can be used to significantly improve the model performance, because the examples with different labels tend to have different features, and utilizing the label information may result in better representation for classification purposes. In our setting, we denote $C(\cdot)$ as a multi-layer neural network or a classifier that takes both the shared representation and the view-specific information as the input and outputs the prediction results. We denote Y as the ground truth label and $\hat{Y} = C(z_s \oplus s_1 \oplus s_2)$ as the label predicted by the classifier, where $\oplus$ is concatenation operator and $z_s = (z_1 + z_2)/2$. Here we aim to minimize the following loss function:

$$L_c = \mathbb{E}_{(\boldsymbol{x_1},\boldsymbol{x_2})\sim p(X_1,X_2)} \mathbb{H}(Y, \hat{Y}) \tag{9}$$

where $\mathbb{H}(Y, \hat{Y})$ is the cross-entropy loss. One goal of the incorporated classifier is to help the two encoders $G_1(\cdot)$ and $G_2(\cdot)$ find a better representation by leveraging the label information. Notice that according to Eq. 5, with the help of the incorporated classifier, *ANTS* tends to assign a small weight to noisy or irrelevant features but a large weight to important features to minimize the prediction loss. This could further help us get a robust interpretable model and alleviate the negative influence of the noisy features. Meanwhile, when the final shared representation is determined, the classifier is also well-trained and ready for the prediction, which saves the time for training the classifier after finding the shared representation.

In addition, it is straightforward to extend the model to include unlabeled data. Similar to the idea of co-training [2], these two predicted labels for unlabeled data should be consistent with each other. Thus, the loss function can be revised as follows:

$$L_c = \mathbb{E}_{(\boldsymbol{x_1},\boldsymbol{x_2})\sim p(X_1,X_2)} \mathbb{H}(\hat{Y}_1, \hat{Y}_2) + \mathbb{H}(Y, \hat{Y}_1) + \mathbb{H}(Y, \hat{Y}_2) \tag{10}$$

where $\hat{Y}_1 = C_1(z_1 \oplus s_1)$, $\hat{Y}_2 = C_2(z_2 \oplus s_2)$ are the predictions of the first view and the second view for unlabeled data, respectively and $C_1(\cdot)$ and $C_2(\cdot)$ are multi-layer neural network that take the concatenation of both view-specific information and shared representation as the input, and output the prediction results. The first term aims to minimize the inconsistency between two predicted labels for unlabeled data, and the remaining terms are the cross-entropy loss on the labeled data for the two views respectively.

Algorithm 1 *ANTS* Algorithm

```
Input: The total number of iterations $T$, two views $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$, and
  the label $Y$, parameters $t_1, t_2, t_3, \alpha, \beta$.
Output: The well-trained classifier $C$.
  Randomly initialize the weights of the neural network.
  for $t = 1$ to $T$ do
    Step 1: Update the discriminator $D$ for $t_1$ times via stochastic
    gradient ascent by fixing the rest variables.
    $D \sim \nabla \mathbb{E}_{\boldsymbol{x_1},\boldsymbol{x_2}\sim p(X_1,X_2)} [\log(D(G_1(\boldsymbol{x_1})))+\log(1-D(G_2(\boldsymbol{x_2})))]$
    Step 2: Update $G_1$, $G_2$, $F_1$ and $F_2$ for $t_2$ times via stochastic
    gradient descent by fixing the rest variables.
    $G_1 \sim \nabla \mathbb{E}_{\boldsymbol{x_1},\boldsymbol{x_2}\sim p(X_1,X_2)} [\log D(G_1(\boldsymbol{x_1})) + \beta||Y - \hat{Y}||_2^2$
        $+\alpha\gamma_2\gamma_1^2||\boldsymbol{x_2} - F_2(G_1(\boldsymbol{x_1}))||_2^2 + \alpha\gamma_1^3||\boldsymbol{x_1} - F_1(G_1(\boldsymbol{x_1}))||_2^2]$
    $G_2 \sim \nabla \mathbb{E}_{\boldsymbol{x_1},\boldsymbol{x_2}\sim p(X_1,X_2)} [\log(1 - D(G_2(\boldsymbol{x_2}))) + \beta||Y - \hat{Y}||_2^2$
        $+\alpha\gamma_1\gamma_2^2||\boldsymbol{x_1} - F_1(G_2(\boldsymbol{x_2}))||_2^2 + \alpha\gamma_2^3||\boldsymbol{x_2} - F_2(G_2(\boldsymbol{x_2}))||_2^2]$
    $F_1 \sim \nabla \mathbb{E}_{\boldsymbol{x_1},\boldsymbol{x_2}\sim p(X_1,X_2)} [\alpha\gamma_1\gamma_2^2||\boldsymbol{x_1} - F_1(G_2(\boldsymbol{x_2}))||_2^2 + \beta||Y - \hat{Y}||_2^2$
        $+\alpha\gamma_1^3||\boldsymbol{x_1} - F_1(G_1(\boldsymbol{x_1}))||_2^2]$
    $F_2 \sim \nabla \mathbb{E}_{\boldsymbol{x_1},\boldsymbol{x_2}\sim p(X_1,X_2)} [\alpha\gamma_2^3||\boldsymbol{x_2} - F_2(G_2(\boldsymbol{x_2}))||_2^2 + \beta||Y - \hat{Y}||_2^2$
        $+\alpha\gamma_2\gamma_1^2||\boldsymbol{x_2} - F_2(G_1(\boldsymbol{x_1}))||_2^2]$
    Step 3: Update the classifier $C$ for $t_3$ times via stochastic
    gradient descent by fixing the rest variables.
    $C \sim \nabla \mathbb{E}_{\boldsymbol{x_1},\boldsymbol{x_2}\sim p(X_1,X_2)} ||Y - C(z_s \oplus S_1 \oplus S_2)||_2^2$
  end for
```

3.3 Proposed Algorithm

We propose to solve the overall objective function via the stochastic block coordinate gradient descent method. Our algorithm is presented in Algorithm 1. It takes as input a data set with two views and the label information, the total number of iterations, as well as several parameters, and outputs the well-trained model. The algorithm works as follows. In Step 1, we first update the discriminator for t_1 times by fixing the other variables and performing a stochastic gradient ascent. Then in Step 2, we only update the two encoders and the two decoders for t_2 times via stochastic gradient descent, as discussed in the previous subsection regarding the optimal encoders and decoders. Finally, in Step 3, we update the classifier for t_3 times via stochastic gradient descent.

3.4 Extension to Multiple Views

In this subsection, we extend our proposed techniques to more generic settings with multiple views. For each pair of views x_i^j

and x_k^j, we could compute its affinity matrix $C_{i,k}^j$ and the hidden representation H_i as follows:

$$\begin{aligned} \boldsymbol{C_{i,k}^j} &= \tanh\left((\boldsymbol{M_i^j})^T \boldsymbol{W_{i,k}} \boldsymbol{M_k^j}\right) \\ H_i^j &= \tanh\left(\boldsymbol{W_i} \boldsymbol{M_i^j} + \sum_{k \neq i}^{v} (\boldsymbol{W_k} \boldsymbol{M_k^j})(\boldsymbol{C_{i,k}^j})^{\boldsymbol{T}}\right) \end{aligned} \tag{11}$$

where $\boldsymbol{W_{i,k}} \in \mathbb{R}^{d_i \times d_j}$, $\boldsymbol{W_i} \in \mathbb{R}^{d \times d_i}$ and $\boldsymbol{W_k} \in \mathbb{R}^{d \times d_k}$ are weight matrices, and d is the output dimension. Using the one vs. all strategy [26], we propose *Centroid-ANTS* framework for the scenario with multiple views, where we consider the first view as the centroid view (assuming we have the prior knowledge that the first view is the most important view). With this setting, we want the discriminator $D(\cdot)$ to distinguish whether the representation is generated by the first view $\boldsymbol{X_1}$ or not, instead of determining whether the representation is generated by the first view or the second view. Thus, the min-max objective function could be adjusted as follows:

$$\begin{aligned} L_0^{(v)} &= \mathbb{E}_{\boldsymbol{x} \sim p(X_1)} \log(D(G_1(\boldsymbol{x_1}))) \\ &+ \frac{1}{v-1} \sum_{i=2}^{v} \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{X_i})} \log(1 - D(G_i(\boldsymbol{x_i}))) \end{aligned} \tag{12}$$

The goal of the discriminator $D(\cdot)$ is to maximize the probability of labeling true to the representation generated by the first view and labeling false to the representation generated by the other views. This enforces the representation generated by the other views to be as similar to the representation generated by the first view as possible. Similar to the cross reconstruction loss introduced in the previous subsection, we expect that for each view other than the first view, the decoder $F_i(\cdot)$ can transform the representation generated by the i^{th} view back to the first view. In addition, we also expect that $F_i(\cdot)$ be able to transform the representation back to the i^{th} view. Thus, the loss function could be modified as follows:

$$\begin{aligned} L_v^{(v)} &= \mathbb{E}_{\boldsymbol{x}_1 \sim p(X_1)} \|\boldsymbol{x_1} - \hat{\boldsymbol{x}}_1\|_2 + \mathbb{E}_{\boldsymbol{x}_2 \sim p(X_2)} \|\boldsymbol{x_2} - \hat{\boldsymbol{x}}_2\|_2 \\ &+ \ldots \mathbb{E}_{\boldsymbol{x_v} \sim p(X_v)} \|x_v - \hat{\boldsymbol{x}}_v\|_2 \\ &= \frac{1}{v} \sum_{i=1}^{v} \mathbb{E}_{(\boldsymbol{x}_1, \boldsymbol{x_i}) \sim p(X_1, \boldsymbol{X_i})} \|\boldsymbol{x_1} - F_1(G_i(\boldsymbol{x_i}))\|_2 \\ &+ \frac{1}{v} \sum_{i=1}^{v} \mathbb{E}_{\boldsymbol{x_i} \sim p(\boldsymbol{x_i})} \|\boldsymbol{x_i} - F_i(G_i(\boldsymbol{x_i}))\|_2 \end{aligned} \tag{13}$$

where the first summation is the loss of transforming the representation generated from the other views back to the first view, and the second term is the reconstruction error of transforming the representation back to its original view. To leverage the label information, we propose to update the cross-entropy loss as follows:

$$L_c^{(v)} = E_{(\boldsymbol{x}_1 \ldots \boldsymbol{x}_v) \sim p(X_1 \ldots X_v)} \mathbb{H}(Y, \hat{Y}) \tag{14}$$

where $\hat{Y} = C(z_s \oplus s_1 \oplus s_2 \oplus \ldots \oplus s_v)$ and $z_s = \frac{1}{v} \sum_{i=1}^{v} z_i$.

4 EXPERIMENTAL RESULTS

In this section, we demonstrate the performance of our proposed algorithm *ANTS* in terms of the effectiveness by comparing it with state-of-the-art methods.

4.1 Experiment Setup

Data sets and experiment setting: We mainly evaluate our proposed algorithm on the following data sets: one semi-synthetic data set based on WebKB*, one synthetic data set, four real world data sets, including Noisy MNIST [34]; XRMB [37], CelebA [20] and Caltech-UCSD Birds [36]. Table 1 shows the statistics of these data sets. In the experiments, we set $t_1 = 2$, $t_2 = 2$, $t_3 = 3$, $\alpha = 1$, $\beta = 1$, the initial learning rate to be 0.03 with decay rate 0.96 if not specified, and the optimizer is momentum stochastic gradient descent. To yield richer and more discriminative feature representation, we adjust the dimensionality of the representation denoted as h based on the dimensionality of the input data, and the dimensionality h is specified for each data set. The number of layers for the encoders, the decoders, and the discriminator can be adjusted to specific application scenarios. Since all the deep model baselines (e.g., Deep IB) are using fully connected layer networks, in the experiments, we set the encoders, the decoders, and the discriminator to be three-layer fully connected layer networks, and the classifier to be a two-layer fully connected layer network.

Reproducibility: All of the real-world data sets are publicly available. The code of our algorithms could be found via the link †. The experiments are performed on a Windows machine with 8GB GTX 1080 GPU.

Comparison methods: In our experiments, we compare with the following methods: FCL, a four-layer fully-connected neural network trained with two concatenated views; Linear CCA [3], linear transformations of two views; DCCA [1], nonlinear transformations of two views by canonical correlation analysis; DCCAE [34], deep canonical correlated auto-encoders for two views; Deep IB [33], deep information bottleneck for multi-view learning. Following [34], we also use SVM to make the final prediction for CCA based methods, after finding the latent representation.

4.2 Semi-synthetic and Synthetic Data Set

The goal of these two semi-synthetic/synthetic data sets is to show that CCA based methods cannot handle the noisy data as they ignore the label information, thus leading to a decrease in prediction performance. In our experiments, we evaluate the performance of our proposed method on one semi-synthetic and one synthetic data set.

The semi-synthetic data set is based on an online textual data set named WebKB, which consists of over 8000 web pages from multiple universities, and these web pages are manually classified into seven categories (labels), e.g., student, faculty, staff, course, project, etc. We extract TF-IDF representation by converting the content of the web pages to word vectors. The dimensionality of this representation is 10,111 after we remove the stop words. Then, we use two different non-linear mapping functions (sigmoid$(\cdot)$ and tanh$(\cdot)$) after normalization to construct the two views. Besides, we add random normal distribution noise ($\mathcal{N}(0, 0.2)$) to the constructed views and shuffle the order of the features. The final dimensionalities of the two views are both 10,611. For our proposed algorithm, we set the dimensionality of the representation to be 300, $k_1 = k_2 = 50, d_1 = d_2 = 233$ (we use the zero-padding to fill the last segment)

*http://www.cs.cmu.edu/~webkb/

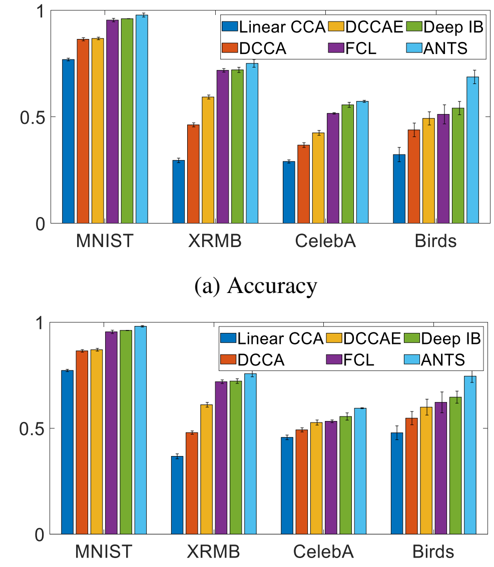
†https://github.com/Leo02016/ANTS

Data Sets	# of Training Samples	# of Test Samples	# of views	# of labels
WebKB	6,626	1,656	2	7
Noisy MNIST	32,000	10,000	2	10
XRMB	75,000	15,000	2	15
CalebA	45,000	12,000	2	40
Caltech-UCSD Bird	950	238	2	40

Table 1: Statistics about five real-world data sets

WebKB		
Model	**Accuracy**	**F1 Score**
linear CCA	0.575 ± 0.012	0.522 ± 0.013
DCCA	0.617 ± 0.009	0.604 ± 0.015
DCCAE	0.652 ± 0.010	0.629 ± 0.010
FCL	0.680 ± 0.002	0.656 ± 0.003
Deep IB	0.650 ± 0.009	0.646 ± 0.009
ANTS	**0.708 ± 0.004**	**0.686 ± 0.004**
Synthetic Data		
Model	**Accuracy**	**F1 Score**
linear CCA	0.674 ± 0.009	0.672 ± 0.006
DCCA	0.709 ± 0.007	0.711 ± 0.009
DCCAE	0.778 ± 0.007	0.773 ± 0.012
FCL	0.897 ± 0.004	0.897 ± 0.004
Deep IB	0.854 ± 0.007	0.854 ± 0.007
ANTS	**0.953 ± 0.005**	**0.953 ± 0.005**

Table 2: Results on WebKB and synthetic data sets



(a) Accuracy

(b) F1 Score

Figure 3: Performance comparison on Noisy MNIST, XRMB, CelebA and Caltech-UCSD Birds (Best viewed in color)

and we use 80% data as the training set and the remaining 20% as the test set.

The synthetic data set is generated in the following way. First, we sample 10,000 data points with the feature dimensionality of 500 categorized into two balanced classes based on the method used in [10]. Then, we use two different non-linear mapping functions ($\text{sigmoid}(\cdot)$ and $\tanh(\cdot)$) after normalization to construct two views. Besides, we add random noise from two normal distributions ($\mathcal{N}(0, 0.5)$ and $\mathcal{N}(1, 0.7)$) to the constructed views and shuffle the order of the features. The noise with the dimensionality of 50 is added to both views and the final dimensionality of both views are 550. For our proposed algorithm, we set the dimensionality of the representation to be 100, $k_1 = k_2 = 11, d_1 = d_2 = 50$ and we use 80% data as the training set and the remaining 20% as the test set.

The comparison results in terms of the accuracy and the F1 score are shown in Table 2. This table shows that our proposed model is better than the others in terms of both evaluation metrics. By observation, we find that for the synthetic data set, the accuracy of *ANTS* reaches 0.953 compared with 0.897 for the second-best baseline. This shows that our proposed method handles noisy data very well because the co-attention module helps users filter out the noisy features by putting a small weight on these noisy features, while the CCA based methods achieve suboptimal performance as they do not utilize the label information to derive the latent representation. For the semi-synthetic data set WebKB, our proposed method achieves the best performance compared with all other baselines.

4.3 Real World Data Sets

Next, we test the performance of our proposed method on four real-world data sets, including Noisy MNIST, XRMB, CelebA, and Caltech-UCSD Birds. The first data set is generated from MNIST data set [16]. We rotate the original MNIST data set at angles randomly chosen from $[-\frac{\pi}{4}, \frac{\pi}{4}]$ to generate the first view and blur the images to create the second view. Both views have 784 features and 10 digits are considered as labels. Because some baselines are time-consuming, we repeat the experiments by randomly sampling 32,000 data points for training and another 10,000 data points for testing. For our proposed method, we set $k_1 = k_2 = 7, d_1 = d_2 = 112$. The second data set is from the Wisconsin X-ray Microbeam Database, which consists of two views. The first view is acoustic data with 273 features and the second view is articulatory data with 112 features. We repeat the experiments by randomly sampling 75,000 data points for training from the first 15 classes and another 15,000 data points for testing. For our proposed method, we set the dimensionality of the representation to be 50, $\alpha = 2$, $\beta = 1$, $k_1 = k_2 = 5, d_1 = 55$, and $d_2 = 23$ (we use the zero-padding to fill the last segment). The third data set is composed of 202,599 images of celebrities, and 40 labeled facial attributes. We repeat the experiments by randomly sampling 45,000 data points for training and over 12000 images for testing. In this experiment, we choose 5 facial attributes related to hair as the labels, e.g., bald, black hair, straight hair, wavy hair, and wearing a hat. To get two views for this data set, we adopt the idea of creating two views in the Noisy MNIST data set by randomly rotating an image to create the first view and blurring the image to generate the second view. To reduce the run-time for the baselines, we first convert the RGB images to gray-scale images and then resize them to 100×100 to be fair to all competitors. Therefore, the final dimensionalities of the two views are both 10,000. For our proposed method, we set $k_1 = k_2 = 50, d_1 = d_2 = 200$.

For the last data set, we sample 1188 images from the first 40 classes. To get the two views, we follow the similar procedure of generating Noisy MNIST [34]. We rotate each image at angles randomly chosen from $[-\frac{\pi}{4}, \frac{\pi}{4}]$ to generate the first view. The procedure for generating the second view consists of two steps. In the first step, we compute the mean μ and the standard deviation σ of each image x_2^j, generate the random noise $\mathcal{N}(\mu, \sigma)$. In the second step, we sample 5% index denoted as *sampled_index* of the image x_2^j and add the noise to the image based on the following function, $x_2^j(sampled\_index) = x_2^j(sampled\_index) + 0.1 * \mathcal{N}(\mu, \sigma)$. Considering the high time complexity of some baselines (e.g., CCA, DCCAE, Deep IB) for high dimensional data like images, we fine-tune a pre-training model (e.g., VGG-19 [‡]) on our dataset to extract the hidden representation of each image and then use the representation as the input for our experiments. The dimension of the hidden representation for both views is 1024. For our proposed method, we first apply the Mask-RCNN [12] to split each image into three segments, and then we also use the same pre-training model to get the representation for each segment of each image (if the image could be split into more than three segments, we pick the top two segments with the most pixels and keep the rest in the third segment; if the image could only be split into two segments, we randomly pick a segment as the third one). For this dataset, we set the learning rate to be 0.08 with decay rate 0.96, the batch size to be 3 (as we find a large batch size leads to the worse performance) and the number of epoch to be 150, $\alpha = 2$, $\beta = 1$ and we use 80% data as the training set and the remaining 20% as the test set.

For these four real-world data sets, the y-axis is accuracy in Figure 3 (a) and F1 score in Figure 3 (b), respectively. These figures show that our proposed model, *ANTS*, outperforms the others with both evaluation metrics in all three data sets. Notice that on the Noisy MNIST data set, most CCA based methods achieve an accuracy above 85%, while *ANTS* even reaches the accuracy of 98.05%, compared with the accuracy of 96.12% for the second-best method (Deep IB). As we mentioned, the two views of the Noisy MNIST data set are either contaminated by the Gaussian noise or distorted by the random rotation but the experimental results on this data set demonstrate that our proposed method could handle the noise very well. In the XRMB data set, *ANTS* boosts the prediction performance by more than 3.5% compared with the second-best baseline in terms of both accuracy and F1 score. In Caltech-UCSD Birds data set, *ANTS* outperforms the second-best algorithm by more than 8%, which demonstrates the effectiveness of handling the noisy features for our proposed method. To show how the proposed method interprets the predictions, we conduct a case study in Section 5.1, which further shows why our proposed method outperforms state-of-the-art algorithms in this data set.

4.4 Parameters Analysis

In this section, we analyze the parameter sensitivity of our proposed *ANTS* algorithm on Noisy MNIST data set, including t_1, t_2, t_3, the dimensionality of the representation h and the hyper-parameters α and β. In all experiments, we use 10,000 data point as the training data and 10,000 data point as the test data; we set the batch size to be 50, and the total iterations number to be 10,000 and the learning rate to be 0.03. In this parameters analysis, if not specified, we let $h = 30$, $t_1 = 2$, $t_2 = 2$, $t_3 = 3$, $\alpha = 1$ and $\beta = 1$.

To evaluate t_1, t_2, and t_3, we increase the value of one parameter and fix the other three parameters to be 1, parameter h to be 30. The x-axis in the Figure 4 (a) is the value of each parameter, e.g. $t_1 = 1, \ldots, 5$ and the y-axis is the classification error. Based on the observations, we find that the classification error increases as t_1 increases, and a large value of t_1 leads to a bad representation and high classification error. The classification error decreases as t_2 and t_3 increases. The model reaches its optimal state when $t_2 = 2$ and $t_3 = 3$ and does not change much when t_2 and t_3 increases. In conclusion, t_1 and t_2 should be set to an identical number, e.g. $t_1 = 2$ and $t_2 = 2$, since we need to balance the loss among the discriminator and the generators. The suitable value for t_3 could be 3 based on the classification error shown in the Figure 4 (a). Similar to the Figure 4 (a), the x-axis of the Figure 4 (b) is the dimensionality of the latent representation and the y-axis is the classification error. Based on the classification error in Figure 4 (b), we observe that the dimensionality of representation does not influence the classification error too much, although the model achieves the slightly better performance when $h = 30$ than other value of h. However, when the dimensionality of representation is less than 10, the classification error increases rapidly. One explanation for this is that the lower-dimensional representation (h<10) fails to capture all useful information from original data, thus leading to an increase in classification error. For the hyper-parameters α and β that balance the cross reconstruction loss and the classification loss, we fix one parameter and increase the value of another one. In Figure 4 (c), the y-axis is the classification error and the x-axis is the value of these hyper-parameters. By observation, we find that when we increase the value of alpha or beta, the classification error also increases slightly. One possible explanation for this observation is that the latent representation is slightly influenced by the imbalance between the cross reconstruction loss and the classification loss.

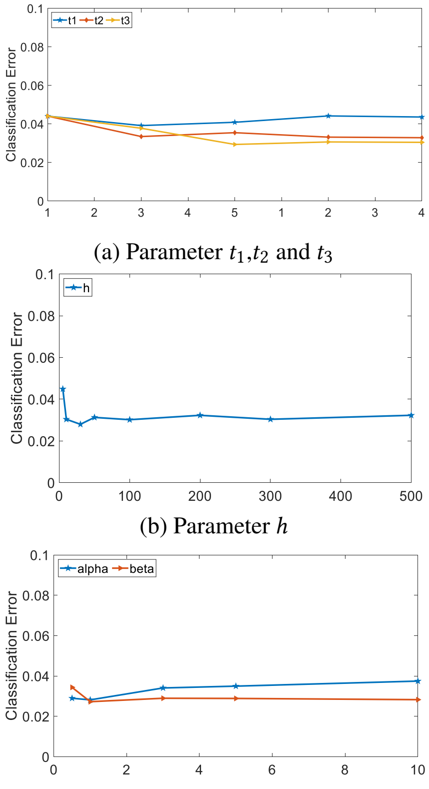
5 CASE STUDIES

In this section, we present a case study on Caltech-UCSD Birds to show the interpretability of the proposed method in Subsection 5.1, and case studies on Noisy MNIST to demonstrate the effectiveness of adding complementary information in Subsection 5.2. Finally, We compare several different regularizers in Subsection 5.3.

5.1 Interpretation via Co-attention

In many real-world applications, the end-user must understand the prediction results from predictive models. Here we conduct a case study to interpret the prediction results and provide relevant clues

[‡] https://pytorch.org/docs/stable/torchvision/models.html



(a) Parameter t_1,t_2 and t_3

(b) Parameter h

(c) Parameter α and β

Figure 4: Parameter analysis (Best viewed in color)

for the end-users to investigate the vulnerability of the proposed algorithm in the noisy environment. We visualize the experimental results on the Caltech-UCSD Birds in the Figure 5, where the Figure 5 (a) and (b) are selected from the training set and the Figure 5 (c) and (d) are selected from the test set. The experimental setting is stated in Section 4.3. In each sub-figure, the first row is the first view and the second row is the second view; seg_1, seg_2 and seg_3 are three image segments for each view. The attention wights of each image segment for two views, (e.g., a_1 and a_2 in Equation 4) are visualized in the bar charts on the right-hand side. The x-axis of the bar chart is the index of the segment, the y-axis is the attention weight, and the bars in blue (orange) are the attention weights for the first (second) view.

Based on these four sub-figures, we have the following observations. First, we find that in Figure 5 (a) and (b), *ANTS* assigns larger weights to the important segments, compared with the bar chart in Figure 5 (a) and (b). This happens because (a) and (b) are selected from the training set, while (c) and (d) are from the test set. Second, in Figure 5 (a), there are two different types of birds, and the label of this image is "Laysan albatross", a large white seabird. *ANTS* assigns 0.80 to the seg_1 of the first view and 0.72 to the seg_2 of the second view, which suggests that *ANTS* successfully captures the most important features of this bird to make an correct prediction. Third, in Figure 5 (c), *ANTS* makes an incorrect prediction because the model puts a large weight on the wrong features (the background of the image) instead of the features of the birds. With these observations, we could draw the following conclusions. First, our proposed model interprets the prediction results and shows the end-users why the model makes an correct or incorrect prediction by assigning the weights to different segments for each view. Second, our proposed methods is capable of handling both noises very well, as two views are generated in noisy environment.

5.2 Ablation Study

In the ablation study, we aim to demonstrate that the boost in the predictive performance is due to the view-specific information rather than the increased dimensionality of the hidden representation. Thus, we examine our hypothesis in the experiment on the Noisy MNIST data set based on the metric of the classification error. The methods for comparison are listed as follows:

- *Method 1*: the proposed method utilizing both the shared information and the view-specific information, namely, *ANTS*.
- *Method 2*: the proposed method utilizing the shared information only but the shared information is copied three times.

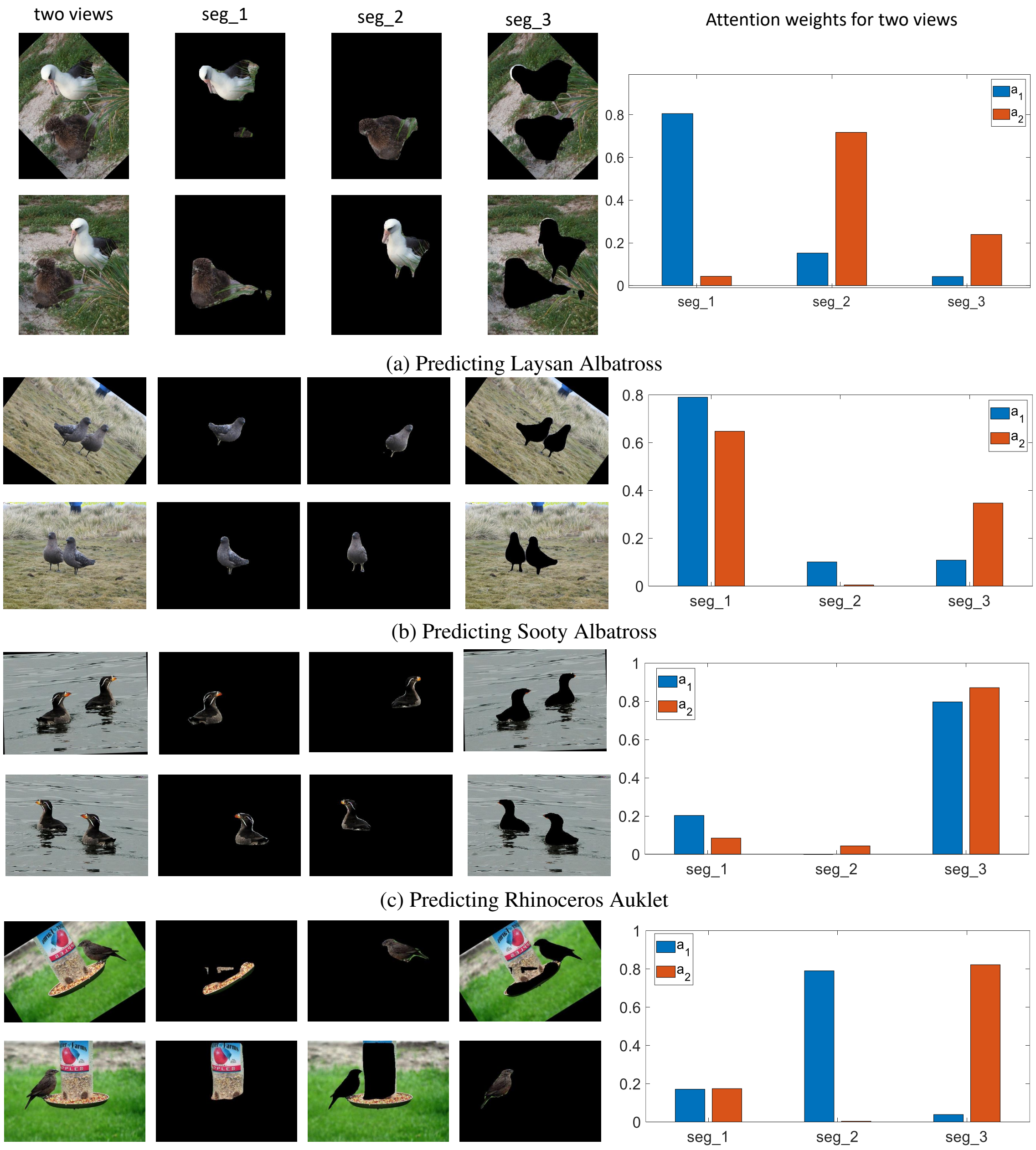
Since *ANTS* utilizes not only the shared information but also view-specific information, for a fair comparison, in *Method 2*, the shared information is copied three times such that the dimensionality of the hidden layer in *Method 2* is the same as that of *Method 1*. In Figure 6 (a), we observe that at the iteration of 500, the classification error of both methods decreases to 0.05. Notice that the classification error of *Method 1* is less than 2.3%, while *Method 2* drops to 4% in the end, which demonstrates that leveraging complementary information could further boost the predictive performance.

5.3 Comparison of View Reconstruction Regularizers

In this case study, we demonstrate the effectiveness of the *cross reconstruction loss* and the effectiveness of utilizing the label information by evaluating the quality of the generated representation with different regularizers on the Noisy MNIST data set. The regularizers for comparison are listed as follows:

- *Reg 1*: Using $F_1(G_1(x_1))$ to reconstruct x_1 and $F_2(G_2(x_2))$ to reconstruct x_2 without utilizing the label information.
- *Reg 2*: Using $F_1(G_2(x_2))$ to reconstruct x_1 and $F_2(G_1(x_1))$ to reconstruct x_2 without utilizing the label information.
- *Reg 3*: Cross reconstruction loss without utilizing the label information (Equation 7).
- *Reg 4*: The L_1 norm of the difference between two representations generated from two views $\|G_1(x_1) - G_2(x_2)\|_1$ without utilizing the label information.
- *Reg 5*: Cross reconstruction loss utilizing the label information.

Because we only aim to find out how different regularizers imposed on the encoders influence the generation of the representation, we exclude the classifier during the training stage in Algorithm 1 for the first four regularizers except for *Reg 5*. In other words, we do not utilize the label information to constrain the generation of the representation except for *Reg 5*. For all the experiments, we randomly sample 32,000 examples from the Noisy MNIST data set as our training data, and 10,000 examples as the test data. For a



(a) Predicting Laysan Albatross

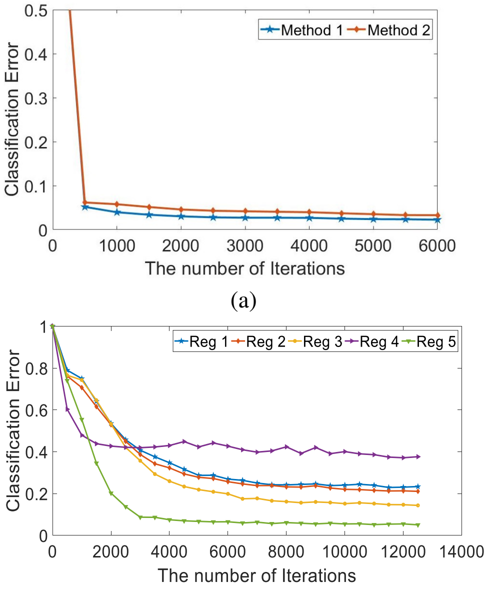
(b) Predicting Sooty Albatross

(c) Predicting Rhinoceros Auklet

(d) Predicting Rusty Blackbird

Figure 5: A case study on the Caltech-UCSD Birds (Best viewed in color). In each sub-figure, the first row is the first view and the second row is the second view; seg_1, seg_2 and seg_3 are three image segments for each view. The x-axis of the bar chart is the index of the segment, the y-axis is the attention weight, and the bars in blue (orange) are the attention weights for the first (second) view.

fair comparison, we set t_1 to be 2, t_2 to be 2 for all five regularizers, t_3 to be 3 for the last regularizers, the batch size to be 64, the total iteration number T to be 12,500, and the initial learning rate to be 0.05 with decay rate 0.96. Furthermore, we evaluate the quality of the representation based on the classification error at every 500 iterations. In Figure 6 (b), the y-axis is the classification error evaluated on 10,000 test examples, and the x-axis is the number of iterations. It is easy to see that: when the label information is not utilized during the training stage, *Reg 2* converges slightly faster than *Reg 1*; compared with *Reg 1*, *Reg 2* and *Reg 4*, the model with cross reconstruction loss (*Reg 3*) converges at the faster rate and the quality of the representation generated by *Reg 3* is also better than



(a)

(b)

Figure 6: (a) Classification error vs the number of iterations on Noisy MNIST data set. (b) Analysis of the quality of the representation generated by different regularizers. (Best viewed in color)

the representation generated by *Reg 1* and *Reg 2*; when the label information is utilized, the quality of the generated representation is further enhanced, and the classification error rate drops rapidly (see the comparison between *Reg 3* and *Reg 5* in Figure 6 (b)). We also observe that the classification error for the model with L_1 norm regularizer (*Reg 4*) decreases to a local minimal error rate of 0.4195 after 3000 iterations and starts to vibrate. The main reason is that there are no constraints imposed to restrict the distribution of P_{g_1} and P_{g_2} and thus, the model with L_1 norm regularizer yields the trivial representation. Based on the results shown in the Figure 6 (b), we can conclude that our model with cross reconstruction loss has a faster convergence rate and tend to find better representation than other regularizers. This demonstrates that cross reconstruction loss indeed enforces the two encoders $G_1(\cdot)$ and $G_2(\cdot)$ to find the shared representation at a faster pace, as mentioned in Subsection 3.2. When the label information is leveraged during the training stage, the convergence rate and the quality of the representation are greatly improved.

6 CONCLUSION

In this paper, we propose *ANTS* - a deep adversarial co-attention model for multi-view subspace learning. We extract and integrate both the shared information and complementary information to obtain a more comprehensive representation. By imposing the cross reconstruction loss and incorporating a classifier into the proposed framework, we further enhance the quality of this representation. We also extend our method to accommodate the more generic scenario with multiple views. We compare our proposed method with state-of-the-art techniques on synthetic, semi-synthetic, and real-world data sets to demonstrate that our method leads to significant improvements in the performance. Finally, case studies show how the proposed method interprets the predictive results on an image data set, the importance of leveraging complementary information, and the effectiveness of the regularizer imposed on view reconstruction.

7 ACKNOWLEDGEMENT

This work is supported by National Science Foundation under Award No. IIS-1947203 and IIS-2002540, and IBM-ILLINOIS Center for Cognitive Computing Systems Research (C3SR) - a research collaboration as part of the IBM AI Horizons Network. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agencies or the government.

REFERENCES

[1] G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. In *Proceedings of International Conference on Machine Learning*, pages 1247–1255, 2013.

[2] A. Blum and T. M. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT 1998, Madison, Wisconsin, USA, July 24-26, 1998.*, pages 92–100, 1998.

[3] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan. Multi-view clustering via canonical correlation analysis. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, pages 129–136, 2009.

[4] C. Du, C. Du, X. Xie, C. Zhang, and H. Wang. Multi-view adversarially learned inference for cross-domain joint distribution matching. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 1348–1357, 2018.

[5] J. D. R. Farquhar, D. R. Hardoon, H. Meng, J. Shawe-Taylor, and S. Szedmák. Two view learning: Svm-2k, theory and practice. In *Advances in Neural Information Processing Systems 18 [Neural Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada]*, pages 355–362, 2005.

[6] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[7] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6602–6611, 2017.

[8] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680, 2014.

[9] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

[10] I. Guyon. Design of experiments of the nips 2003 variable selection benchmark. In *NIPS 2003 workshop on feature extraction and feature selection*, 2003.

[11] J. He, C. Du, F. Zhuang, X. Yin, Q. He, and G. Long. Online bayesian max-margin subspace multi-view learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 1555–1561, 2016.

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

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

[14] Z. Jiao and C. Xu. Deep multi-view robust representation learning. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*, pages 2851–2855, 2017.

[15] P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. *ICML (2017)*, 2017.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[17] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang, and B. Poczos. Mmd gan: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems 30*, pages 2203–2213. 2017.

[18] P. Liu, X. Qiu, and X. Huang. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1–10, 2017.

[19] Y. Liu, Y. Gu, Z. Ding, J. Gao, Z. Guo, Y. Bao, and W. Yan. Decoupled graph convolution network for inferring substitutable and complementary items. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2621–2628, 2020.

[20] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *2015 IEEE ICCV, Santiago, Chile, 2015*, pages 3730–3738, 2015.

[21] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances in neural information processing systems*, pages 289–297, 2016.

[22] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774, 2017.

[23] A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow. Adversarial autoencoders. In *International Conference on Learning Representations*, 2016.

[24] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[25] M. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *SIGKDD (2016)*, 2016.

[26] R. M. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

[27] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[28] V. Sindhwani, P. Niyogi, and M. Belkin. A co-regularization approach to semi-supervised learning with multiple views. In *Proceedings of International Conference on Machine Learning workshop on learning with multiple views*, pages 74–79, 2005.

[29] S. Su, H. Ge, and Y. Yuan. A label embedding kernel method for multi-view canonical correlation analysis. *Multimedia Tools Appl.*, 76(12):13785–13803, 2017.

[30] Y. Tian, X. Peng, L. Zhao, S. Zhang, and D. N. Metaxas. CR-GAN: learning complete representations for multi-view generation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 942–948, 2018.

[31] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2962–2971, 2017.

[32] F. Wang, Q. Huang, M. Ovsjanikov, and L. J. Guibas. Unsupervised multi-class joint image segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 3142–3149, 2014.

[33] Q. Wang, C. Boudreau, Q. Luo, P.-N. Tan, and J. Zhou. Deep multi-view information bottleneck. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 37–45. SIAM, 2019.

[34] W. Wang, R. Arora, K. Livescu, and J. A. Bilmes. On deep multi-view representation learning. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 1083–1092, 2015.

[35] W. Wang, H. Lee, and K. Livescu. Deep variational canonical correlation analysis. *CoRR*, abs/1610.03454, 2016.

[36] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[37] J. Westbury. X-ray microbeam speech production database user's handbook: Madison. *WI: Waisman Center, University of Wisconsin*, 1994.

[38] C. Xu, D. Tao, and C. Xu. A survey on multi-view learning. *CoRR*, abs/1304.5634, 2013.

[39] S. Yu, B. Krishnapuram, R. Rosales, H. Steck, and R. B. Rao. Bayesian co-training. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, 2007*, pages 1665–1672, 2007.

[40] Y. Zhang, R. Barzilay, and T. S. Jaakkola. Aspect-augmented adversarial networks for domain adaptation. *TACL*, 5:515–528, 2017.

[41] L. Zheng, Y. Cheng, and J. He. Deep multimodality model for multi-task multi-view learning. In T. Y. Berger-Wolf and N. V. Chawla, editors, *Proceedings of the 2019 SIAM International Conference on Data Mining, SDM 2019, Calgary, Alberta, Canada, May 2-4, 2019*, pages 10–18. SIAM, 2019.

[42] D. Zhou, J. He, K. S. Candan, and H. Davulcu. MUVIR: multi-view rare category detection. In Q. Yang and M. J. Wooldridge, editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 4098–4104. AAAI Press, 2015.

[43] D. Zhou, J. He, Y. Cao, and J. Seo. Bi-level rare temporal pattern detection. In F. Bonchi, J. Domingo-Ferrer, R. Baeza-Yates, Z. Zhou, and X. Wu, editors, *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain*, pages 719–728. IEEE Computer Society, 2016.

[44] D. Zhou, J. Luo, V. M. B. Silenzio, Y. Zhou, J. Hu, G. Currier, and H. A. Kautz. Tackling mental health by integrating unobtrusive multimodal sensing. In B. Bonet and S. Koenig, editors, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 1401–1409. AAAI Press, 2015.

[45] D. Zhou, L. Zheng, J. Han, and J. He. A data-driven graph generative model for temporal interaction networks. In R. Gupta, Y. Liu, J. Tang, and B. A. Prakash, editors, *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 401–411. ACM, 2020.

[46] D. Zhou, L. Zheng, J. Xu, and J. He. Misc-gan: A multi-scale generative model for graphs. *Frontiers in Big Data*, 2:3, 2019.

[47] D. Zhou, L. Zheng, Y. Zhu, J. Li, and J. He. Domain adaptive multi-modality neural attention network for financial forecasting. In *Proceedings of The Web Conference 2020*, pages 2230–2240, 2020.

[48] T. Zhou, P. Krähenbühl, M. Aubry, Q. Huang, and A. A. Efros. Learning dense correspondence via 3d-guided cycle consistency. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 117–126, 2016.

[49] T. Zhou, Y. J. Lee, S. X. Yu, and A. A. Efros. Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1191–1200, 2015.

[50] Y. Zhou and J. He. Crowdsourcing via tensor augmentation and completion. In S. Kambhampati, editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2435–2441. IJCAI/AAAI Press, 2016.

[51] Y. Zhou and J. He. A randomized approach for crowdsourcing in the presence of multiple views. In V. Raghavan, S. Aluru, G. Karypis, L. Miele, and X. Wu, editors, *2017 IEEE International Conference on Data Mining, ICDM 2017, New Orleans, LA, USA, November 18-21, 2017*, pages 685–694. IEEE Computer Society, 2017.

[52] Y. Zhou, L. Ying, and J. He. Multic^2: an optimization framework for learning from task and worker dual heterogeneity. In N. V. Chawla and W. Wang, editors, *Proceedings of the 2017 SIAM International Conference on Data Mining, Houston, Texas, USA, April 27-29, 2017*, pages 579–587. SIAM, 2017.

[53] Y. Zhou, L. Ying, and J. He. Multi-task crowdsourcing via an optimization framework. *ACM Trans. Knowl. Discov. Data*, 13(3):27:1–27:26, 2019.

[54] C. Zhu, Y. Cheng, Z. Gan, S. Sun, T. Goldstein, and J. Liu. Free{lb}: Enhanced adversarial training for language understanding. In *International Conference on Learning Representations*, 2020.

[55] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2242–2251, 2017.