Embedded Deep Bilinear Interactive Information and Selective Fusion for Multi-view Learning

Jinglin Xu, Wenbin Li, Jiantao Shen, Xinwang Liu, Peicheng Zhou, Xiangsen Zhang, Xiwen Yao, and Junwei Han

Abstract—As a concrete application of multi-view learning, multi-view classification improves the traditional classification methods significantly by integrating various views optimally. Although most of the previous efforts have been demonstrated the superiority of multi-view learning, it can be further improved by comprehensively embedding more powerful cross-view interactive information and a more reliable multi-view fusion strategy in intensive studies. To fulfill this goal, we propose a novel multi-view learning framework to make the multi-view classification better aimed at the above-mentioned two aspects. That is, we seamlessly embed various intra-view information, cross-view multi-dimension bilinear interactive information, and a new view ensemble mechanism into a unified framework to make a decision via the optimization. In particular, we train different deep neural networks to learn various intra-view representations, and then dynamically learn multi-dimension bilinear interactive information from different bilinear similarities via the bilinear function between views. After that, we adaptively fuse the representations of multiple views by flexibly tuning the parameters of the view-weight, which not only avoids the trivial solution of weight but also provides a new way to select a few discriminative views that are beneficial to make a decision for the multi-view classification. Extensive experiments on six publicly available datasets demonstrate the effectiveness of the proposed method.

Index Terms—Multi-view learning, Bilinear function, Intra-view information, Cross-view interactive information, Selective strategy.

I. INTRODUCTION

Classification has been a fundamental technique of machine learning and broadly applied in image classification [1]–[3], image reconstruction [4], cross-modal retrieval [5], gaze tracking [6], visual tracking [7], [8], multi-view boosting [9], multi-view biological data [10] and so on. The typical classification methods mainly include Logistic Regress, K-Nearest Neighbor, Decision Tree, and Support Vector Machines (SVM), where the kernel trick is an important innovation of SVM since it can not only utilize the convex optimization method with convergence to learn the nonlinear model but also make the implementation of kernel function (such as Radial Basis Function (RBF)) high-efficiency. Unfortunately, the computation cost of the kernel methods is too large for the large dataset. The deep learning method, therefore, is proposed to overcome the limitations of the kernel methods. For example, [11] has been demonstrated that the performance of Neural Networks is better than that of SVM with RBF kernel on the MNIST dataset.

Recently, with the rapid development of data mining techniques, in many scientific data analysis tasks, data are often collected through different ways, such as various sensors, as usually a single way cannot comprehensively describe the entire information of a data instance. In this case, different kinds of features of each data instance can be considered as different views of this instance, where each view may capture a specific facet of this instance. Importantly, the combination of multiple views can provide a more comprehensive view of this instance. Obviously, the multi-view features can be utilized to sufficiently represent an instance compared to the single view features. As an important branch of multi-view learning [12], the multi-view classification [13]–[15] uses multiple distinct representations of data and models a multi-view learning framework to perform the classification task.

Canonical Correlation Analysis (CCA) [16] and Kernel CCA (KCCA) [17]–[19] show their abilities of effectively modeling the relationship between two or more views. However, they still have some limitations on capturing high-level associations between different views. To be specific, CCA ignores the nonlinearities in multi-view data and KCCA may suffer from the effect of small data when the data acquisition in one or more modalities is expensive or otherwise limited. Therefore, in the early efforts, [20] investigates a neural network implementation of CCA and maximizes the correlation between the output of the networks for each view. [21] formulates a nonlinear CCA method using three feedforward neural networks, where the first network maximizes the correlation between two canonical variates, while the remaining two networks map from the canonical variates back to the original two sets of variables. However, the above CCA-based methods have been proposed for many years and are needed to be improved.

Inspired by the recent successes of deep neural networks [22], [23], correlation can be naturally applied to multi-view neural network learning to learn deep and abstract multi-view interactive information. The deep neural networks extension of CCA (Deep CCA) [24] learns representations of two views by using multiple stacked layers and maximizes the correlation of two representations. Later, [25] proposes a deep canonically correlated auto-encoder (DCCAE) by combining the advantages of both Deep CCA and auto-encoder. Specifically, DCCAE consists of two auto-encoders and optimizes the
combination of canonical correlation between two learned bottleneck representations and the reconstruction errors of the auto-encoders. Similar to the principle of DCCA, [26] proposes a correspondence auto-encoder (Corr-AE) via constructing correlations between the hidden representations of two unimodal deep auto-encoders. [27] suggests exploiting the cross weights between the representations of views for gradually learning interactions of the modalities (views) in a multi-modal deep auto-encoder network. Theoretical analysis of [27] shows that considering these interactions from a high level provides more intra-modality (intra-view) information.

In addition, a number of multi-view analysis methods [28], [29] have been proposed. Based on Fisher Discriminant Analysis (FDA) [30], both regularized two-view FDA and its kernel extension can be cast as equivalent disciplined convex optimization problems. Then, [28] introduces Multi-view Fisher Discriminant Analysis (MFDA) that learns classifiers in multiple views, by minimizing the variance of the data along with the projection while maximizing the distance between the average outputs for classes over all the views. However, MFDA can only be used for binary classification problems. [29] proposes a Multi-view Discriminant Analysis (MvDA), which seeks for a discriminant common space by maximizing the between-class and minimizing the within-class variations, across all the views. Later, based on bilinear models [31] and general graph embedding framework [32], [33] introduces Generalized Multi-view Analysis (GMA). As an example of GMA, Generalized Multi-view Linear Discriminant Analysis (GMLDA) finds a set of projection directions in each view that tries to separate different contents’ class means and unify different views of the same class in the common subspace.

Based on the above analyses, although CCA-based deep neural networks can learn the multi-view interactive information and DA-based methods can consider the discriminative information, there is not a unified framework that simultaneously embeds the intra-view information, cross-view interactive information, and a reliable multi-view fusion strategy.

To tackle this issue, we propose a novel multi-view learning framework named MvNNBiIn, which integrates both the intra-view information and cross-view multi-dimension bilinear interactive information for each view and designs a multi-view selective loss fusion. Specifically, (a) multiple intra-view information coming from different faceted representations, ensures the diversity and complementarity among different views to enhance multi-view learning. (b) The cross-view multi-dimension bilinear interactive information dynamically learns multi-dimension bilinear interactive information from different bilinear similarities. Each bilinear similarity is calculated by the similarity between intra-view information via the bilinear function. Therefore, the multi-dimension bilinear interactive information comprehensively models the relationships between views by learning different metric matrices. (c) The multi-view selective loss fusion is to calculate multiple losses for multiple views and fuses them in an adaptive weighting way with the selective strategy. This selective strategy can choose several discriminative views that are beneficial to make a decision for the multi-view classification, by tuning the sparseness of the weight vector.

It is worth mentioning that we have developed a preliminary work [34] named deep embedded complementary and interactive information for multi-view classification (denoted as MvNNcor). In this paper, our proposed MvNNBiIn method extends and improves the MvNNcor method significantly. Their major differences are summarized as follows. On one hand, MvNNcor utilizes the cross-correlations between attributes of multiple representations to generate interactive information. In contrast, our MvNNBiIn models a novel cross-view multi-dimension bilinear interactive information which consists of different bilinear similarities for each view with respect to another view. Each bilinear similarity is generated by the similarity between intra-view information passed by the bilinear function. On the other hand, MvNNcor uses the multi-view fusion strategy to integrate multiple views. By contrast, our MvNNBiIn designs a novel view ensemble mechanism to select more discriminative views that are beneficial to the multi-view classification.

We summarize the main contributions of the proposed MvNNBiIn method as follows.

• We propose a unified framework, which seamlessly embeds intra-view information, cross-view multi-dimension bilinear interactive information, and a novel view ensemble mechanism, to make a decision during the optimization and improve the classification performance.
• We model the cross-view interactive information by capturing the multi-dimension bilinear interactive information which is calculated by simultaneously learning multiple metric matrices via the bilinear function between views. It comprehensively models the relationships between different views.
• We develop a new view ensemble mechanism which not only selects some discriminative views but also fuses them via an adaptive weighting method. The selective strategy can ensure that the selected views are beneficial to the multi-view classification.
• We perform extensive experiments on several publicly available datasets to demonstrate the effectiveness of our model.

The rest of this paper is organized as follows. The Preliminary Knowledge is introduced in Section II. We formulate the framework of MvNNBiIn in Section III. Section IV briefly provides a tractable and skillful optimization method of MvNNBiIn. Section V evaluates MvNNBiIn on several public datasets, followed by some theoretical and empirical analyses of experiments. The conclusion and further works are shown in Section VI.

II. PRELIMINARY KNOWLEDGE

In this section, we briefly review several multi-view learning methods from the perspectives of CCA-based methods and MvDA method.

A. CCA-based Methods

CCA [16] is popular for its capability of modeling the relationship between two or more sets of variables. CCA computes a shared embedding of both or more sets of variables through
maximizing the correlations among the variables among these sets. CCA has been widely used in the multi-view learning tasks to generate low-dimensional representations [35], [36]. Improved generalization performance has been witnessed in areas including dimensionality reduction [37], clustering [38], [39], regression [40], [41], word embeddings [42]–[44], and discriminant learning [45], [46].

Supposing that \( X^v | v = 1, 2 \in \mathbb{R}^{d_x \times N} \) is a data matrix for the \( v \)-th view, CCA tends to find the linear projections \( w^v | v = 1, 2 \in \mathbb{R}^{d_x} \) which make the instances from two data matrices \( X^v | v = 1, 2 \) maximally correlated in the projected space. Therefore, CCA is modeled as the following constrained optimization problem,

\[
\begin{aligned}
\max_{w^v | v = 1, 2} & \quad w^v \mathbf{T} X^1 X^2 \mathbf{T} w^v \\
\text{s.t.} & \quad w^v \mathbf{T} X^v w^v = 1, v = 1, 2
\end{aligned}
\]  

(1)

where \( X^v | v = 1, 2 \) are centralized.

KCCA is a kernel extension of CCA for pursuing maximally correlated nonlinear projections. Let \( K_v \) denote the kernel matrix such that \( [K_v]_{ij} = k_v(x_i^v, x_j^v) \), where \( k_v : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R} \), \( \mathcal{H} \) is a centering matrix and \( 1 \in \mathbb{R}^n \) denotes a vector of all ones. Therefore, KCCA is formulated as the following optimization problem,

\[
\begin{aligned}
\max_{\alpha, \beta} & \quad \alpha^\mathbf{T} K_1 K_2 \beta \\
\text{s.t.} & \quad \alpha^\mathbf{T} K_1^2 \alpha = 1, \beta^\mathbf{T} K_2^2 \beta = 1
\end{aligned}
\]  

(2)

where \( \alpha \) and \( \beta \) are the coefficient vectors optimized by KCCA.

As a DNN extension of CCA, DCCA utilizes two DNNs \( f_1 \) and \( f_2 \) to extract nonlinear features for each view, and then maximizes the canonical correlation between \( f_1(X^1) \) and \( f_2(X^2) \). That is,

\[
\begin{aligned}
\min_{w_{j1}, w_{j2}, U, V} & \quad \frac{1}{N} \text{Tr} \left( U \mathbf{T} f_1(X^1) f_2(X^2) \mathbf{T} V \right) \\
\text{s.t.} & \quad U \mathbf{T} \left( \frac{1}{N} f_1(X^1) f_1(X^1)^\mathbf{T} + r_1 I \right) U = I \\
& \quad V \mathbf{T} \left( \frac{1}{N} f_2(X^2) f_2(X^2)^\mathbf{T} + r_2 I \right) V = I \\
& \quad u_i^\mathbf{T} f_1(X^1) f_2(X^2)^\mathbf{T} v_j = 0, \quad \text{for} \ i \neq j
\end{aligned}
\]  

(3)

where \( U = [u_1, \cdots, u_d] \in \mathbb{R}^{d_x \times d} \) and \( V = [v_1, \cdots, v_d] \in \mathbb{R}^{d_y \times d} \) are the CCA directions that project the DNN outputs, and \( (r_1, r_2) > 0 \) are the regularization parameters for the sample covariance estimation.

Inspired by both CCA and reconstruction-based objectives, DCCAE constructs a model by consisting of two auto-encoders and optimizing the combination of canonical correlation between the learned representations and the reconstruction errors of the auto-encoders. That is,

\[
\begin{aligned}
\min_{w_{j1}, w_{j2}, w_{i1}, w_{i2}} & \quad \frac{1}{N} \text{Tr} \left( U \mathbf{T} f_1(X^1) f_2(X^2) \mathbf{T} V \right) \\
& \quad + \frac{r}{N} \sum_{i=1}^{d_x} (\|x_i^1 - g_1(f_1(x_i^1))\|^2 + \|x_i^2 - g_2(f_2(x_i^2))\|^2) \\
\text{s.t.} & \quad U \mathbf{T} \left( \frac{1}{N} f_1(X^1) f_1(X^1)^\mathbf{T} + r_1 I \right) U = I \\
& \quad V \mathbf{T} \left( \frac{1}{N} f_2(X^2) f_2(X^2)^\mathbf{T} + r_2 I \right) V = I \\
& \quad u_i^\mathbf{T} f_1(X^1) f_2(X^2)^\mathbf{T} v_j = 0, \quad \text{for} \ i \neq j
\end{aligned}
\]  

(4)

where \( g_1 \) and \( g_2 \) are the reconstruction networks for each view, and \( r \) is the trade-off parameter.

B. MvDA Method

MvDA attempts to find \( M \) linear transforms \( w^v | v = 1 \) that project the samples from \( M \) views to one discriminant common space, respectively, where the between-class variation is maximized while the within-class variation is minimized. Defined \( X^v = \{x_{ik}^v\} \) as the samples from the \( v \)-th view, where \( x_{ik}^v \in \mathbb{R}^{d_v} \) is the \( i \)-th sample from the \( v \)-th view of the \( k \)-th class of \( d_v \) dimension and \( C \) denotes the number of classes, and \( n_k^v \) is the number of samples from the \( v \)-th view of the \( k \)-th class.

The samples from \( M \) views are then projected to the same common space by using \( M \) view-specific linear transforms. The projected results are denoted as \( Y = \{y_{ik}^v = w^v \mathbf{T} x_{ik}^v\} \). In the common space, according to our goal, the between-class variation \( S_B^v \) from all views should be maximized while the within-class variation \( S_W^v \) from all views should be minimized. Therefore, the objective is formulated as a generalized Rayleigh quotient,

\[
(w^1, \cdots, w^M) = \arg \max_{w^1, \cdots, w^M} \frac{\text{Tr}(S_B^v)}{\text{Tr}(S_W^v)}
\]  

(5)

where the within-class scatter matrix \( S_W^v \) and the between-class scatter matrix \( S_B^v \) are computed as below:

\[
S_W^v = \sum_{k=1}^{C} \sum_{v=1}^{M} \sum_{i=1}^{n_k^v} (y_{ik}^v - \mu_k)(y_{ik}^v - \mu_k)^\mathbf{T}
\]  

(6)

\[
S_B^v = \sum_{k=1}^{C} n_k (\mu_k - \mu)(\mu_k - \mu)^\mathbf{T}
\]  

(7)

where \( n_k = \sum_{v=1}^{M} n_k^v \) denotes the number of samples of the \( k \)-th class in all views, \( N = \sum_{k=1}^{C} n_k \) is the number of samples from all the classes and all the views, \( \mu_k \) is the mean of all the samples of the \( k \)-th class over all the views in the common space, \( \mu \) is the mean of all the samples over all views in the common space. That is,
Multi-view data vectors for each sample

III. THE PROPOSED METHOD

In this section, we propose the architecture of our model MnNNBiIn, depicted in Figure 1, which consists of four parts: various intra-view extracted networks, multi-dimension bilinear interactive modules between views, combination of intra-view and cross-view information, and the multi-view selective loss fusion strategy.

A. Various Intra-view Information Extraction

Given an instance \( x \), we utilize \( M \) views \( \{ x^v \}_{v=1}^M \) to denote its \( M \) kinds of visual features, which ensures the diverse and complementary information during the multi-view learning. Defining a set of neural networks \( \{ f_v \}_{v=1}^M \), each \( f_v \) projects \( x^v \) from \( \mathbb{R}^{d_v} \) into \( \mathbb{R}^d \), which captures the high-level intra-view information for the \( v \)-th view, that is,

\[
x_f^v = f_v(x^v),
\]

where \( x_f^v \in \mathbb{R}^d \) and \( f_v \) is a neural network with \( L_f \) layers,

\[
h_f^v = \sigma(W_f^l h_f^{v,l-1} + b_f^v),
\]

where \( W_f^l \in \mathbb{R}^{m_l \times m_{l-1}} \) denotes the weight matrix and \( b_f^v \in \mathbb{R}^{m_l} \) denotes the bias vector, and \( h_f^v \in \mathbb{R}^{m_l} \) is the output of the \( l \)-th layer, and \( \sigma \) is the activation function applied component-wise. Specifically, \( l = 1, \cdots, L_f \), \( m_0 = d_v \), \( m_{L_f} = d \), \( h_f^0 = x^v \), and \( h_f^L = x_f^v \). It is worth mentioning that \( \{ f_v \}_{v=1}^M \) are trained coordinatively by solving the optimization problem shown in subsection III-D, where the parameters of \( f_v \) are not shared with those of \( \{ f_l \}_{l=1}^L \). Although we utilize a multi-layer perceptron as \( f_v \), it also can be replaced with any deterministic neural networks with the input and output layers.

Fig. 2. The construction of the multi-dimension bilinear interactive information of each pair of views. Taking the first view \( x_f^1 \) as an example, we calculate its multi-dimension bilinear interactive information relative to the second view \( x_f^2 \) by learning multiple metric matrices, i.e., \( (v, \bar{v}) = (1, 2) \), and then obtain \( x_f^{1,2} \in \mathbb{R}^{d_B} \).

B. Multi-dimension Bilinear Interactive Information

Inspired by the metric learning, the multi-dimension bilinear interactive information is proposed to construct the second-order feature interactions on various intra-view information. To be specific, given the \( v \)-th view \( x_f^v \) and the \( \bar{v} \)-th view \( x_f^{\bar{v}} \), the multi-dimension bilinear interactive information between different views can be formulated via the bilinear function \( B^{v,\bar{v}} \), that is,

\[
x_f^{v,\bar{v}} = B^{v,\bar{v}}(x_f^v, x_f^{\bar{v}}) = \begin{bmatrix}
x_f^v \top B_{1}^{v,\bar{v}} x_f^{\bar{v}} + b_1^{v,\bar{v}} \\
x_f^v \top B_{2}^{v,\bar{v}} x_f^{\bar{v}} + b_2^{v,\bar{v}} \\
\vdots \\
x_f^v \top B_{d_B}^{v,\bar{v}} x_f^{\bar{v}} + b_{d_B}^{v,\bar{v}}
\end{bmatrix}
\]

where \( x_f^{v,\bar{v}} \in \mathbb{R}^{d_B} \), \( B_{p}^{v,\bar{v}} \in \mathbb{R}^{d \times d}, \) \( p = 1, \cdots, d_B \), and \( b_{p}^{v,\bar{v}} \in \mathbb{R} \) is the bias. There are multiple metric matrices (i.e., \( B_{1}^{v,\bar{v}}, \cdots, B_{d_B}^{v,\bar{v}} \)) to be learned simultaneously. Intuitively, Figure 2 shows the construction of the multi-dimension bilinear interactive information of each pair of views.

For the \( v \)-th view, we define \( \Gamma_v \) as a set contained \( M - 1 \) view pairs relative to the \( v \)-th view, that is,

\[
\Gamma_v = \{(v, \bar{v})|\bar{v} \in \{1, \cdots, M\} \setminus v\},
\]

\( \alpha \geq 0 \) and \( \|w\|_0 = s \).
where \( v \in \{1, \cdots, M\} \) and each view pair \((v, \bar{v})\) is undirected. For the \( v \)-th view, the bilinear interactive information of the \( v \)-th view relative to other \( M - 1 \) views is collected into a set \( x_{v}^{Γ} \),

\[
x_{v}^{Γ} = \{ x_{v}^{Γ} \mid (v, \bar{v}) = Γ_{v}\}
\]

where the \( v \)-th set \( Γ_{v} \) contains \( M - 1 \) view pairs for the \( v \)-th view and there are \( M \) sets \( Γ_{v} = 1 \) for \( M \) views.

### C. Combination and Prediction of Each View

According to the above two subsections, we combine the intra-view information \( x_{v}^{Γ} \) and multi-dimension bilinear interactive information \( x_{v}^{Γ} \) of the \( v \)-th view as follows,

\[
x^{Γ}_{v} = \text{Con}[x_{f}, x_{v}^{Γ}]
\]

where Con denotes the concatenation operation.

After that, \( x^{Γ}_{v} \in \mathbb{R}^{d_{v}+d_{o}} \) is passed through \( φ \) to obtain the prediction \( z^{v} \) of the \( v \)-th view, that is,

\[
z^{v} = φ(x^{Γ}_{v})
\]

where \( C \) is the number of classes and \( z^{v} \in \mathbb{R}^{C} \) produces a distribution over the possible classes for each view. \( φ \) is a neural network with \( L_{φ} \) layers, that is,

\[
h_{φ}^{l} = σ(W_{φ}^{l}h_{φ}^{l-1} + b_{φ}^{l})
\]

where \( W_{φ}^{l} \in \mathbb{R}^{n_{φ} \times n_{φ} - 1} \) denotes the weight matrix and \( b_{φ}^{l} \in \mathbb{R}^{n_{φ}} \) denotes the bias vector, and \( h_{φ}^{0} \in \mathbb{R}^{n_{φ}} \) is the output of the \( l \)-th layer, and \( σ \) is the activation function applied component-wise. In particular, \( l = 1, \cdots, L_{φ}, n_{0} = d_{v} + d_{o}, n_{L_{φ}} = C, h_{φ}^{0} = x^{Γ}_{v}, \) and \( h_{φ}^{L_{φ}} = z^{v} \). Here, the parameters of \( φ \) are shared among \( M \) views.

### D. Multi-view Selective Loss Fusion Strategy

In this subsection, we design a novel view ensemble mechanism, which is to calculate multiple losses for multiple views and fuses them in an adaptive weighting way with the selective strategy. This selective strategy can choose several discriminative views that are beneficial to the multi-view classification, by tuning the sparseness of the weight vector. That is described as,

\[
\min_{α} \sum_{v=1}^{M} α_{v}L^{v}(z^{v}, y) ~ \text{s.t.} ~ \sum_{v=1}^{M} α_{v} = 1, α_{v} ≥ 0, \|α\|_{0} = s
\]

where \( L^{v}(z^{v}, y) = -\log \left( \frac{\exp(z^{v}_{y})}{\sum_{c=1}^{C} \exp(z^{v}_{c})} \right) \),

\[
\|α\|_{0} = s \quad \text{is used to constrain the}
\]

sparseness of the weight vector \( α \), where \( s \in \mathbb{N}_{+} \) denotes the number of nonzero elements in \( α \). Crucially, the \( L_{0}\)-norm constraint is able to capture the global relationship among different views and to achieve the view-wise sparsity, which realizes selecting a few discriminative views can improve the performance during the multi-view classification. Intuitively, the architecture of our proposed method MvNNBiIn is shown in Figure 1.

### IV. OPTIMIZATION

We utilize the alternate optimization method to update the parameters \( \{f_{v}\}_{v=1}^{M}, B, φ \) and the view-weight \( α \), respectively. For convenience, \( B \) denotes the bilinear function set \( \{B^{v,\bar{v}}(v, \bar{v}) = Γ_{v}\}_{v=1}^{M} \) and \( ψ \) is used to denote the concatenation operation in Figure 3.

#### A. Update \( \{f_{v}\}_{v=1}^{M}, B, \) and \( φ \)

We fix \( α \) and update \( \{f_{v}\}_{v=1}^{M}, B, \) and \( φ \), which utilizes Adam with batch normalization and the autograd package in PyTorch to train. Figure 3 briefly shows the gradient computations of our proposed MvNNBiIn method.

#### B. Update \( α \)

We learn the non-negative normalized weight \( α_{v} \) for each view and assign the higher weight to more discriminative view. Therefore, we fix the parameters \( \{f_{v}\}_{v=1}^{M}, B, φ \) and update \( α \) by solving the optimization problem (16).

To efficiently minimize the problem (16), we define a function \( ϕ \) on \( L(z, y) = [L^{1}(z^{1}, y), \cdots, L^{M}(z^{M}, y)] \in \mathbb{R}^{M} \), that is,

\[
ϕ(L(z, y)) = L(z, y)P
\]

where \( P \) is a permutation matrix which results in permuting elements of \( L(z, y) \) along the ascending order, i.e., \( L^{0}(z^{(1)}, y) ≤ \cdots ≤ L^{M}(z^{(M)}, y) \). Based on equation (18), we select the first \( s \) smallest elements and optimize their corresponding weights \( α_{p(v)} \), meanwhile, setting the rest
In this paper, we evaluate the performance of our proposed MvNNBiIn method on six publicly available datasets.

### A. Datasets

We follow many state-of-the-art multi-view methods, which treat different kinds of pre-extracted feature vectors for an image as different views to make the experimental comparisons fair. Different kinds of pre-extracted features present different facets of the images, such as color, texture, shape, and geographic information, yet the above information is not intuitive. We describe all the publicly available datasets in Table I. Specifically, Caltech101 [47], [48] and Caltech20 [48] datasets consist of images of objects belonging to 102 (101 classes plus one background clutter class) and 20 classes, respectively, and each dataset is described as 6 views. AWA (Animals with Attributes) [49] dataset provides 6 kinds of features (6 views) for attribute base classification. NUSOBJ dataset is a subset of NUS-WIDE [50], which describes each object image using 5 types of low-level features (5 views). Reuters [51] dataset is used for document categorization and written in 5 different languages (5 views). SUN is a subset of SUN397 (Scene Categorization Benchmark) [52], [53] and utilizes 3 kinds of public features matrices (3 views) to represent each image.

Referring to the previous works [24], [54], we split each dataset into three parts, that is, 70% samples for training, 20% samples for validating, and 10% samples for testing. We utilize the classification accuracy (e.g., Top@1 accuracy and Top@5 accuracy) to evaluate the performance of all the methods and report the final results in TABLEs II, III, and IV.

### B. Experimental Settings

1) **Comparison Methods:** We first compare our MvNNBiIn method with several state-of-the-art multi-view methods, including SVMcon, DCCA [24], DCCAE [54], DeepLDA [55], MvDA [56], MvDN [57], and MvNNcor [34], in Table III. In particular, SVMcon is a baseline that concatenates all the features of an image as different views to make the experimental comparisons fair. Different kinds of pre-extracted features present different facets of the images, such as color, texture, shape, and geographic information, yet the above information is not intuitive. We describe all the publicly available datasets in Table I.

Specifically, Caltech101 [47], [48] and Caltech20 [48] datasets consist of images of objects belonging to 102 (101 classes plus one background clutter class) and 20 classes, respectively, and each dataset is described as 6 views. AWA (Animals with Attributes) [49] dataset provides 6 kinds of features (6 views) for attribute base classification. NUSOBJ dataset is a subset of NUS-WIDE [50], which describes each object image using 5 types of low-level features (5 views). Reuters [51] dataset is used for document categorization and written in 5 different languages (5 views). SUN is a subset of SUN397 (Scene Categorization Benchmark) [52], [53] and utilizes 3 kinds of public features matrices (3 views) to represent each image.

Referring to the previous works [24], [54], we split each dataset into three parts, that is, 70% samples for training, 20% samples for validating, and 10% samples for testing. We utilize the classification accuracy (e.g., Top@1 accuracy and Top@5 accuracy) to evaluate the performance of all the methods and report the final results in TABLEs II, III, and IV.

### V. Experiments

In this section, we evaluate the performance of our proposed MvNNBiIn method on six publicly available datasets.
views and feeds into the SVM classifier. DCCA and DC-CAE belong to the CCA-based method inputting two views. DeepLDA, MvDA, and MvDN are the Discriminant Analysis based method, where DeepLDA inputs the concatenation of all the views and feeds into a deep neural network followed by LDA; MvDA inputs multiple views with the same dimensionality; MvDN is the non-linear version of LDA, which uses deep neural networks to replace the linear transformations.

What’s more, Table IV demonstrates the effectiveness of three important parts of our proposed MvNNBiIn, i.e., intra-view information, cross-view bilinear interactive information, and a novel view ensemble mechanism, respectively. The highest performance is obtained when all the parts are available while the performance is lower when any part is absent.

2) Parameters Setup: Deep LDA is a fully connected neural network consisted of three hidden layers, i.e., 400, 200, and 300 units equipped with ReLU activation function. In DCCA, there is a feature extraction network and a reconstruction network for each view, where each network is a fully connected network consisted of three hidden layers, i.e., 400, 200, and 300 units equipped with ReLU activation function, followed by a linear output of $C$ (for feature extraction network) $/\bar{v}$, (for reconstruction network) units. The capacities of the above networks are the same as those of their counterparts in DCCA and MvDN.

In our MvNNBiIn, two kinds of networks $\{\{f_\nu\}_{\nu=1}^M, \phi\}$ and the bilinear function set $B = \{B_\nu,\bar{v}\} = \Gamma_\nu \{f_\nu\}_{\nu=1}^M$ are needed to learn. Each $f_\nu$ is a fully connected network consisted of two hidden layers (i.e., 400 and 200 units equipped with ReLU activation function). $\phi$ consists of 200-$M$ input units and 300 hidden units equipped with ReLU activation function, followed by a linear output of $C$ units. Each $B_\nu,\bar{v}$ is a bilinear function which outputs $d_B = 200$ linear interactive units from each pair $(x_\nu^{(i)}, x_\nu^{(j)})$. That is, the inputs of $B_\nu,\bar{v}$ are the outputs of $f_\nu$ and $f_\bar{v}$, and the input of $\phi$ is concatenated by the outputs of $f_\nu$ and $B_\nu,\bar{v}$. The capacities of the above networks are the same as those of their counterparts in the ablation studies shown in Table IV. In MvNNcor, three kinds of networks $\{\{f_\nu\}_{\nu=1}^M, \psi, \phi\}$ are needed to learn, where $\{f_\nu\}_{\nu=1}^M$ are the same as the counterparts of MvNNBiIn, and $\psi$ consists of 200$^2$ input units and 200 hidden units with ReLU activation function.

In this paper, all the networks are optimized by Adam with batch normalization, where the learning rate is $10^{-3}$, $\beta_1 = 0.5$, $\beta_2 = 0.9$, and batch size is 64. In addition, we vary $\gamma$ and $s$, respectively, to explore the influence of different values of $\gamma$ and $s$ on the classification accuracy. Based on the optimal $\gamma$ and $s$, we can learn the optimal model to achieve the highest
TABLE II
THE INVESTIGATION OF THE DIMENSIONALITY $d_B$ OF THE BILINEAR FUNCTION $B^{v, \bar{v}}$ ON DIFFERENT DATASETS. $d_B$ IS SET AS 50, 100, 200, AND 400, RESPECTIVELY, WHICH REFLECTS THE COMBINATION PROPORTION OF INTRA-VIEW INFORMATION AND CROSS-VIEW BILINEAR INTERACTIVE INFORMATION FOR EACH VIEW. DUE TO CUDA OUT OF MEMORY, THE RESULTS OF $d_B = 400$ CANNOT BE OBTAINED EXCEPT FOR NUSOBJ DATASET.

| $d_B$ | Caltech101 | Caltech20 | AWA | NUSOBJ | Reuters | SUN |
|-------|------------|-----------|-----|--------|---------|-----|
|       | Top@1 | Top@5 | Top@1 | Top@5 | Top@1 | Top@5 | Top@1 | Top@5 | Top@1 | Top@5 | Top@1 | Top@5 | Top@1 | Top@5 | Top@1 | Top@5 | Top@1 | Top@5 | Top@1 | Top@5 |
| 50    | 76.228 | 87.277 | 97.397 | 98.958 | 48.210 | 74.142 | 41.116 | 99.838 | 33.770 | 71.069 |
| 100   | 75.781 | 88.058 | 97.397 | 98.958 | 47.428 | 74.740 | 41.116 | 99.838 | 33.770 | 71.069 |
| 200   | 76.562 | 89.063 | 98.958 | 99.473 | 49.316 | 75.521 | 41.116 | 99.838 | 33.770 | 71.069 |
| 400   | /     | /     | /     | /     | /     | /     | /     | /     | /     | /     |

TABLE III
COMPARISON RESULTS OF MVNNBIIK AND SEVERAL STATE-OF-THE-ART METHODS ON ALL THE DATASETS.

| Method | Caltech101 | Caltech20 | AWA | NUSOBJ | Reuters | SUN |
|--------|------------|-----------|-----|--------|---------|-----|
| SVMcon | 47.901 | 83.827 | 31.044 | 42.719 | 88.180 | / |
| DeepLDA| 45.649 | 76.508 | 25.598 | 20.320 | 84.907 | / |
| MVDA   | 45.200 | 76.276 | 9.788 | 11.457 | 78.831 | / |
| DCCA   | 66.159 | 86.504 | 20.677 | 28.753 | 64.917 | 16.116 |
| DCCAE  | 26.894 | 50.267 | 13.484 | 27.477 | 56.530 | / |
| MvDN   | 70.214 | 94.833 | 42.356 | 47.241 | 88.339 | 40.538 |
| MvNNcor| 76.002 | 97.924 | 47.687 | 52.049 | 89.276 | 45.632 |
| MvNNBII | 76.562 | 98.958 | 49.316 | 51.529 | 90.194 | 47.036 |

TABLE IV
ABLATION EXPERIMENTS OF OUR PROPOSED MVNNBIIK ON ALL THE DATASETS.

| φ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ | ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ |
| $\{ f_v \}_{v=1}^d$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| B | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| $\{ a_v \}_{v=1}^d$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| $\{ a_v \}_{v=1}^d, \| \alpha \|_0 = s$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

C. Experimental Results

Tables III and IV show the classification performance of all the methods, where Table III reports the experimental results of several recent methods and Table IV provides the results of ablation experiments.

Firstly, compared with the single-view methods, i.e., SVMcon and DeepLDA, our MvNNBIIk consistently outperforms them on all the datasets. For example, MvNNBIIk achieves 28.661% and 30.913% improvements, respectively, on the Caltech101 dataset. That is because the concatenation of all the views may reduce the interpretability of different views and ignore the cross-view interactive information during the multi-view classification. We compare our MvNNBIIk with MVDA and the classification accuracy of our MvNNBIIk is better than that of MVDA on all the datasets. For instance, our MvNNBIIk obtains 31.362% improvements on the Caltech101 dataset, since the linear transformations of MVDA cannot deal well with some subtle but important structures in some challenging scenarios. Compared to MvDN, our MvNNBIIk achieves 6.348% improvements on the Caltech101 dataset due to embedding the multi-dimension bilinear interactive classification accuracy. The results are shown in Figure 4 where Caltech101, Caltech20, AWA, NUSOBJ, Reuters, and SUN datasets can achieve the best performance when $(\gamma, s)$ are set as $(5, 5), (4, 4), (2, 6), (10, 4), (6, 5),$ and $(11, 3)$, respectively.

Besides, to explore the proper combination proportion of the intra-view information and multi-dimension bilinear interactive information for each view, we investigate the dimensionality $d_B$ of the output of the bilinear function $B^{v, \bar{v}}$, i.e., setting $d_B$ as 50, 100, 200, and 400, respectively, and the results are reported in Table II. It can be seen that the classification accuracy is the highest when $d_B = 200$, which shows that the combination proportion also has an impact on the classification performance.
information between different views.

Secondly, we compare our MvNNBiIn with the CCA-based methods, i.e., DCCA and DCCAE, and the results are reported in Table III. It can be seen that our MvNNBiIn performs better than DCCA and DCCAE, since these two methods are limited to the double-view input and unable to capture more diverse and complementary information from more views. For example, compared to DCCA and DCCAE, our MvNNBiIn achieves 28.63% and 35.83% improvements on the AWA dataset, respectively.

Thirdly, as the extension work of MvNNcor, our MvNNBiIn almost achieves better performance on all the datasets. For example, compared with MvNNcor on the AWA dataset, MvNNBiIn achieves 1.629% improvements. For one reason, the cross-view multi-dimension bilinear interactive information of our MvNNBiIn is more able to capture the interactive information between different views. For another reason, our MvNNBiIn designs a novel view ensemble mechanism which can select more discriminative views and is more beneficial to the multi-view classification.

Besides, in the ablation studies of our MvNNBiIn shown in Table IV, taking the NUSOBJ dataset as an example, we can achieve 2.682%, 1.163%, 15.060%, 3.790%, 3.474%, 0.856%, and 0.399% improvements compared with our framework respectively containing some of the following modules, i.e., $\phi$, $\phi + \{f_v\}_{v=1}^M + B$, $\phi + \{f_v\}_{v=1}^M + \{\alpha_v\}_{v=1}^M$, $\phi + \{f_v\}_{v=1}^M | \{\alpha_v\}_{v=1}^M | \{x_v\}_{v=1}^M$, $\phi + \{f_v\}_{v=1}^M + B + \{\alpha_v\}_{v=1}^M$. These results successively demonstrate the effectiveness of integrating the intra-view information and multi-dimension bilinear interactive information between views as well as the adaptive weighting multi-view loss fusion with the selective strategy.

Moreover, Figure 5 shows the view-weights of each dataset learned by our MvNNBiIn, where the $x$-axis means the indices of different views and the $y$-axis denotes the weight of each view. The higher weight indicates that the view provides more valuable information and makes more contribution.

D. Discussion

Actually, our MvNNBiIn is a generic framework that can improve the multi-view classification using not only the hand-crafted features (such as HOG, LBP, or SURF) but also the deep model-learned features.

We apply four popular CNNs (including AlexNet [58], GoogLeNet [59], VGGNet-16 [60], and ResNet-101 [61]) on three publicly available image datasets (i.e., Caltech101, NUSOBJ, and SUN397), respectively, to generate the CNN feature representations including four transferred CNN features and four fine-tuned CNN features. Then, we compare our proposed MvNNBiIn method with the single-view CNN feature-based methods to demonstrate the superiority of our multi-view learning framework.

To be specific, for the transferred CNN feature-based methods, four off-the-shelf CNN models including VGGNet-16, ResNet-101, AlexNet, and GoogLeNet are first adopted as general feature extractors to extract CNN features and then linear one-versus-all SVMs ($C=0.001$) are used for classification. For the fine-tuned CNN feature-based methods, we fine-tune the aforementioned four CNN models on the training datasets to extract better CNN features and then adopt linear

---

Fig. 5. The learned weights for different views through MvNNBiIn for Caltech101/20, AWA, NUSOBJ, Reuters, and SUN datasets. The blue dashed boxes respectively in Caltech101, Caltech20, and NUSOBJ datasets denote that the view weights are zeros.
In this paper, we propose a novel multi-view learning framework denoted as MvNNBiIn which seamlessly embeds various intra-view information and cross-view multi-dimension bilinear interactive information as well as introducing a new view ensemble mechanism to jointly make decisions during the multi-view classification. Extensive experiments on several publicly available datasets demonstrate the effectiveness of our proposed MvNNBiIn method. Furthermore, we demonstrate the superiority of multi-view learning using the CNN feature representations, which provides a novel idea of fusing outputs of different deterministic neural networks in further work.

VI. CONCLUSION

In this paper, we propose a novel multi-view learning framework denoted as MvNNBiIn which seamlessly embeds various intra-view information and cross-view multi-dimension bilinear interactive information as well as introducing a new view ensemble mechanism to jointly make decisions during the multi-view classification. Extensive experiments on several publicly available datasets demonstrate the effectiveness of our proposed MvNNBiIn method. Furthermore, we demonstrate the superiority of multi-view learning using the CNN feature representations, which provides a novel idea of fusing outputs of different deterministic neural networks in further work.

REFERENCES

[1] G. Cheng, J. Han, and X. Lu, “Remote sensing image scene classification: Benchmark and state of the art,” Proceedings of the IEEE, vol. 10, no. 105, pp. 1865–1883, 2017.
[2] J. Han, X. Yao, G. Cheng, X. Feng, and D. Xu, “P-cnn: Part-based convolutional neural networks for fine-grained visual categorization,” IEEE Transactions on Pattern Analysis and Machine Intelligence, DOI: 10.1109/TPAMI.2019.2935110.
[3] J. Xu, F. Nie, and J. Han, “Feature selection via scaling factor integrated multi-class support vector machines,” in International Joint Conference on Artificial Intelligence, 2017.
[4] C. Du, C. Du, L. Huang, and H. He, “Reconstructing perceived images from human brain activities with bayesian deep multi-view learning,” IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 8, pp. 2310–2323, 2018.
[5] Y. Yu, S. Tang, K. Aizawa, and A. Aizawa, “Category-based deep cca for fine-grained venue discovery from multimodal data,” IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 4, pp. 1250–1258, 2018.
[6] D. Lian, L. Hu, W. Luo, Y. Xu, L. Duan, J. Yu, and S. Gao, “Multiview multitask gaze estimation with deep convolutional neural networks,” IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 10, pp. 3010–3023, 2018.
[7] S. Zhang, X. Yu, Y. Sui, S. Zhao, and L. Zhang, “Object tracking with multi-view support vector machines,” IEEE Transactions on Multimedia, vol. 17, no. 3, pp. 265–278, 2015.
[8] G. Cheng, J. Han, P. Zhou, and D. Xu, “Learning rotation-invariant and fisher discriminative convolutional neural networks for object detection,” IEEE Transactions on Image Processing, vol. 28, no. 1, pp. 265–278, 2019.
[9] J. Peng, A. J. Aved, G. Seetharaman, and K. Palaniappan, “Multiview boosting with information propagation for classification,” IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 3, pp. 657–669, 2017.
[10] C.-M. Feng, Y. Xu, J.-X. Liu, Y.-L. Gao, and C.-H. Zheng, “Supervised discriminative sparse pca for com-characteristic gene selection and tumor classification on multiview biological data,” IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 10, pp. 2926–2937, 2019.
