Multiview Orthonormalized Partial Least Squares: Regularizations and Deep Extensions

Li Wang, Ren-Cang Li, and Wen-Wei Lin

Abstract— **In this article, we establish a family of subspace-based learning methods for multiview learning using least squares as the fundamental basis. Specifically, we propose a novel unified multiview learning framework called multiview orthonormalized partial least squares (MvOPLSs) to learn a classifier over a common latent space shared by all views. The regularization technique is further leveraged to unleash the power of the proposed framework by providing three types of regularizers on its basic ingredients, including model parameters, decision values, and latent projected points. With a set of regularizers derived from various priors, we not only recast most existing multiview learning methods into the proposed framework with properly chosen regularizers but also propose two novel models. To further improve the performance of the proposed framework, we propose to learn nonlinear transformations parameterized by deep networks. Extensive experiments are conducted on multiview datasets in terms of both feature extraction and cross-modal retrieval. Results show that the subspace-based learning for a common latent space is effective and its nonlinear extension can further boost performance, and more importantly, one of two proposed methods with nonlinear extension can achieve better results than all compared methods.**

Index Terms— **Deep learning, multiview learning, orthonormalized partial least squares (OPLSs), regularization, subspace learning.**

I. Introduction

DATA sets are increasingly collected from different views of the underlying object in many real-world applications [1]. They are capable of depicting, more comprehensively, the object from multiple views than solely relying on a single view. Each view is composed of its own set of features. As the data for each view within the multiview dataset contain complementary information, it is expected that learning algorithms should make good use of these views for the best learning outcome [2].

Multiview learning [1], [3] is a learning mechanism seeking to leverage the complementary information of multiple views to boost learning performance. Many multiview learning algorithms have been proposed in the literature. Among them, subspace-based learning approaches have attracted much attention. They aim to obtain a common latent subspace shared by all views under the assumption that these views are generated from the common subspace. The subspace-based learning algorithms have demonstrated a great deal of success in learning tasks, such as cross-modal retrieval [4], [5] and feature extraction [6], [7].

In this article, we will focus on the study of a family of subspace-based multiview learning algorithms in terms of the least squares formulation from three different perspectives: 1) two or more views; 2) linear/nonlinear representation; and 3) unsupervised/supervised learning.

The most representative model in multiview subspace learning is canonical correlation analysis (CCA), which was originally proposed to learn two linear projection matrices by maximizing the correlation between two views in a common space [8]. It has since been extended for more than two views [9], [10], nonlinear projections via either kernel representation [11] or deep representation [12], and supervised learning [5]. Moreover, different least squares reformulations of CCA have been proposed for supervised multilabel classification [6] and unsupervised learning of more than two views [13]. They have demonstrated great advantages in yielding effective models and efficient learning algorithms. However, the least squares reformulation in [6] is essentially a single-view classification method since it treats data points as one view and class labels as another. In addition to CCA, other forms of least squares have been studied for two views, such as coupled spectral regression [14] and partial least squares (PLS) [15], [16].

The least squares formulation has been previously studied for single-view supervised learning, but it is seldom explored for subspace-based multiview learning. As to single-view learning, linear discriminant analysis (LDA) can be formulated as least squares for both binary classification [17] and multiclass classification [18]. CCA for supervised classification is equivalent to LDA for multiclass classification [19], and therefore, CCA shares the same least squares characteristics as that of LDA. For example, LDA has been generalized

Manuscript received 22 December 2020; revised 20 May 2021 and 5 August 2021; accepted 26 September 2021. Date of publication 12 October 2021; date of current version 4 August 2023. The work of Li Wang was supported by NSF under Grant DMS-2009689. The work of Ren-Cang Li was supported by NSF under Grant DMS-1719620 and Grant DMS-2009689. The work of Wen-Wei Lin was supported in part by the Ministry of Technology and Science (MoST) under Grant 106-2628-M-009-004, in part by the National Center for Theoretical Science (NCTS), and in part by the ST Yau Centre in Taiwan. *(Corresponding author: Li Wang.)*

Li Wang is with the Department of Mathematics, The University of Texas at Arlington, Arlington, TX 76019 USA, and also with the Department of Computer Science and Engineering, The University of Texas at Arlington, Arlington, TX 76019 USA (e-mail: li.wang@uta.edu).

Ren-Cang Li is with the Department of Mathematics, The University of Texas at Arlington, Arlington, TX 76019 USA (e-mail: rcli@uta.edu).

Wen-Wei Lin is with the Department of Applied Mathematics, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: wwlin@am.nctu.edu.tw).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TNNLS.2021.3116784.

Digital Object Identifier 10.1109/TNNLS.2021.3116784

to learn projection matrices for binary classification of two views [20]. Although CCA and LDA have their own least squares formulations for two views, their extensions to multiple views of more than two views are not that straightforward. Nonetheless, multiview discriminant analysis (MvDA) extends LDA for multiclass classification of more than two views [4]. Various combinations of CCA and LDA are also proposed in [5], [7], and [21]. Instead of least squares, these methods are originally modeled as trace ratio problems, but it is the relaxed ratio trace problems that get finally solved numerically because of existing highly efficient eigendecomposition packages.

In this article, we will investigate orthonormalized PLSs (OPLSs) [22], which was proposed to perform dimensionality reduction only in the input space and that makes it different from and also less popular than CCA and PLS. Precisely this property of reduction in the input space becomes its advantage for multivariate analysis in the setting of supervised learning because prediction primarily relies on reliable extraction of good features in the input space. The equivalence between supervised CCA and OPLS was established in [23]. Kernel OPLS was proposed in [24] for learning nonlinear transformations. We emphasize that OPLS admits a least squares formulation which also produces an optimal regression classifier in the latent space at the same time [24], [25]. That is an important advantage. However, these methods only work for single-view learning. Hence, they are not designed for supervised multiview learning.

Inspired by the least squares characteristics of OPLS, we propose a novel unified multiview learning framework for subspace-based learning. The framework can learn a classifier over the latent space shared by all views. Several regularizations are presented to enrich the proposed framework, which not only can recast many existing methods but also generate new models by the practitioner if needed. The proposed framework can deal with any number of views with or without class labels and can learn either linear or nonlinear projections.

The main contributions of this article are summarized as follows.

1) We propose multiview OPLS (MvOPLS) as an extension of OPLS to multiview subspace learning. MvOPLS can simultaneously learn a latent common space, mappings, and a decision function shared by all views. We further study its advantages including the inherited ones from OPLS and ones that are unique to multiview learning.

2) We propose a generalized formulation of regularized MvOPLS in order to facilitate the inclusion of various prior knowledge and still retain the advantages of MvOPLS. Three general purposed regularizations are studied, with examples, on model parameters, decision values, and latent projected points. Two new models are showcased as a demonstration.

3) We propose to extend regularized MvOPLS to learn nonlinear transformations parameterized by deep networks. All methods instantiated from the proposed regularized MvOPLS can take advantage of the proposed nonlinear extension with little additional effort. This creates a large family of deep supervised subspace-based multiview learning methods.

4) We recast several existing methods under the proposed regularized MvOPLS framework, putting firm theoretical foundations underneath them as most of them were conceived heuristically. To deepen the understanding of existing methods, we highlight their differences in terms of the choices of regularizations. This provides guidelines to not only choose proper methods based on regularizers but also develop new methods with new regularizers that take advantage of domain priors.

5) We conduct extensive experiments to compare nine methods and their two variants of each, a total of 27 methods, instantiated from the proposed framework on nine multiview datasets with various numbers of views. These methods are evaluated and compared on two different tasks: feature extraction and cross-modal retrieval. Results show that subspace-based learning for a common latent space is effective and its nonlinear extension can further boost performance, and more importantly, one of two proposed methods with nonlinear extension achieves better results than all compared methods.

In the rest of this article, we first briefly review existing methods related to this work in Section II. In Section III, we propose MvOPLS for multiview subspace learning with proper regularizations for incorporating priors in Section IV and nonlinear transformation via deep neural networks in Section V. Extensive experiments are conducted in Section VI. Finally, we draw our conclusion in Section VII.

Notation: $\mathbb{R}^{m\times n}$ is the set of $m \times n$ real matrices and $\mathbb{R}^n = \mathbb{R}^{n\times 1}$. $I_n \in \mathbb{R}^{n\times n}$ is the identity matrix of size $n \times n$, $\mathbf{1}_n \in \mathbb{R}^n$ is the vector of all ones, and $H_n = I_n - (1/n)\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}}$ is the centering matrix. For a matrix B, $\|B\|_{\mathrm{F}}$, and $\mathrm{tr}(B)$ are its Frobenius norm and trace (assuming it is square then), respectively. B^{T} is the transpose of a matrix or vector.

II. Related Work

We briefly review existing methods relevant to this work. Specially, we first discuss methods in two broad categories: unsupervised and supervised learning in the case of two views, and then their extensions to more than two views and nonlinear transformations.

In the setting of unsupervised learning, CCA has been the workhorse for learning a common latent space of two views [8]. To deal with more than two views, multiset CCA (MCCA) [9], [10] based on pairwise correlations and generalized CCA (GCCA) [26] based on aligning all views via a common representation have been proposed. MCCA with the least squares formulation [13] is widely used due to its simplicity. In addition, the multiview uncorrelated locality preserving projection [27] maximizes the sum of all the pairwise correlations and the high-order correlation. Kernel CCA (KCCA) [11] and deep CCA (DCCA) [12] are two representative approaches to explore nonlinear projections to model complex real-world datasets via the kernel trick and deep learning, respectively. DMCCA [28] extends MCCA to nonlinear transformations via deep networks, but it can only deal with the very special case where all views reside in the same input space. Deep GCCA (DGCCA) [29] extends GCCA to nonlinear transformations but it does not reduce to CCA for two views, suggesting that the extension is not a natural one. Other linear models closely related to CCA [30] have also

been explored especially for two views, including PLS [15] and OPLS [23], [24]. In addition, spectral regression is used to learn a common space between two views in two separate steps [14]. These methods do not explicitly take into account supervised information, such as the class labels for multiclass classification.

Several supervised subspace-based learning approaches have been proposed to integrate supervised information in order to improve multiview learning. LDA [31] is the main tool for subspace learning with supervised information. The combination of LDA and CCA has been successfully used to find discriminatory subspaces. Generalized multiview analysis (GMA) [5] obtains a discriminatory common space by incorporating intraview discriminatory information and cross-view correlation and is readily extensible via the kernel trick to learn nonlinear transformations for more than two views. Different from GMA, MLDA [7] replaces the within-class scatter matrices with the covariance matrices. MvDA [4] considers both interview and intraview variations leading to a more discriminative common space. Its nonlinear extension through deep networks has been studied in [32], resulting in a discriminatory and view-invariant representation shared among all multiple views. Multiview modular discriminant analysis (MvMDA) [21] is proposed to maximize the distances between different class centers across different views and minimize the within-class scatter of each view. Some of these methods are originally formulated as trace ratio problems but solved as relaxed ratio trace problems because of available numerical linear algebra packages. However, the two types of optimization problems are not equivalent [33], and, as a result, their solutions are not optimal ones to their original models, leaving lingering questions, such as how these solutions should be interpreted and how effective they may be.

In the following, we will propose a unified multiview learning framework, which can recast most of the above-mentioned subspace-based learning approaches into it. Hence, the framework offers a natural and accurate interpretation of the relaxed problems of existing models. The proposed framework combined with judiciously chosen regularizations can inspire novel and powerful models for different learning tasks, and, without much additional effort, their nonlinear extensions.

III. Multiview OPLS

We will establish a simple and yet natural multiview extension of the classical OPLS model and conduct a detailed analysis to uncover the advantages of the proposed model for better understanding and application.

A. Motivation

Most existing subspace learning methods seek a latent space by optimizing application-agnostic criteria. Examples are PCA (covariance maximization or reconstruction error), LDA (class separability), CCA (correlation maximization), and PLS (cross-covariance maximization). Often, the projected data by these methods are later used for other learning purposes, such as classification, clustering, and retrieval, as preprocessed input, and therefore, it is highly likely that the latent space obtained from one criterion may not work well for another learning task that is more aligned with a different criterion. Although a prediction method can learn a specified mapping function directly from original raw input data without any preprocessing with a properly chosen criterion, it may still suffer from poor generalizations because raw data are usually noisy and with intrinsic latent structure concealed and, as a result, the learned mapping function directly from raw data does not reflect the concealed structure information and is sensitive to noise.

OPLS [22], [24] is a subspace learning model with a built-in multivariate regression system for predicting the output of any given input. The built-in prediction system benefits from its least squares reformulation [24]. OPLS generalizes many other models, such as CCA [23] and LDA (see Section I-B of the Supplementary Material) and has been successfully applied in many applications. Unfortunately, the success has so far been limited to single-view subspace learning. Our goal in this article is to explore OPLS for multiview subspace learning.

For multiview learning, the most fundamental challenge is how multiview data can be truthfully represented and summarized in such a way that heterogeneity gaps [2] among different views can be satisfactorily overcome and comprehensive information concealed in the data can be properly exploited by multiview learning models. As each view in multiple views intends to represent the same object but characterize it by heterogeneous features, the ultimate goal of multiview subspace learning is to find a common k-dimensional latent space $\mathbb{R}^k$ such that transformed data points of all views for the same underlying object in the latent space are "similar" to each other in order to reduce the heterogeneity gap. However, it is generally difficult to give a uniform "similarity" quantification among projected points. Instead, we strike to provide a platform, in which different prior knowledge can be used to shape the "similarity" among projected points in various contexts.

Our platform is built upon OPLS. We first propose our plain multiview OPLS (MvOPLS) as an extension of OPLS via a tied built-in classifier. We will then demonstrate that MvOPLS not only provides a powerful vehicle to incorporate many types of prior knowledge but also facilitates the learning of deep representation.

B. Proposed MvOPLS

Multiview subspace learning strives to learn from data consisting of more than one views, namely, multiview data. Consider a multiview dataset of v views consisting of n labeled data instances: $\{(\boldsymbol{x}_i^{(1)}, \ldots, \boldsymbol{x}_i^{(v)}, \boldsymbol{y}_i)\}_{i=1}^n$, where $\boldsymbol{x}_i^{(s)} \in \mathbb{R}^{d_s}$ is the data point of view s of the ith instance, and $\boldsymbol{y}_i \in \mathbb{R}^c$ is the class label of the ith instance, and c is the number of classes. In the case when the class labels $\boldsymbol{y}_i$ are not available, it will just be $\{(\boldsymbol{x}_i^{(1)}, \ldots, \boldsymbol{x}_i^{(v)})\}_{i=1}^n$. Let

$$X_s = \left[\boldsymbol{x}_1^{(s)}, \ldots, \boldsymbol{x}_n^{(s)}\right] \in \mathbb{R}^{d_s \times n}, \quad Y = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n] \in \mathbb{R}^{c \times n}$$

called the data matrix and label matrix of view s, respectively. It is worth noting that there are many different ways to define

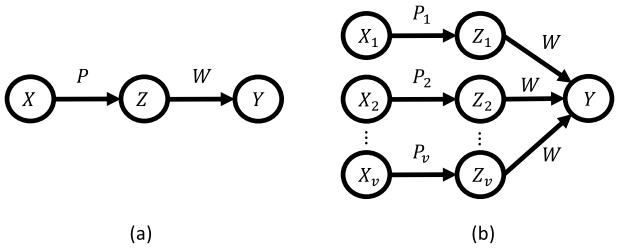

Fig. 1. (a) Illustration of single-view OPLS and (b) multiview OPLS.

$\boldsymbol{y}_i$ to best suit specific learning problems, such as multitarget regression [34], multilabel classification [6], [35], cross-modal retrieval [4], [5], and feature extraction [6], [7]. Our later development works equally well regardless of how $\boldsymbol{y}_i$ are defined.

In what follows, we start with linear transformations and leave the study of nonlinear transformations to Section V.

For view s, we look for projection matrix $P_s \in \mathbb{R}^{d_s \times k}$ to transform $\boldsymbol{x}_i^{(s)}$ from $\mathbb{R}^{d_s}$ to $\boldsymbol{z}_i^{(s)} = P_s^{\mathrm{T}} \boldsymbol{x}_i^{(s)}$ in the common space $\mathbb{R}^k$ under suitable criteria. Let

$$Z_s = \left[\boldsymbol{z}_1^{(s)}, \ldots, \boldsymbol{z}_n^{(s)}\right] = P_s^{\mathrm{T}} X_s \in \mathbb{R}^{k \times n}$$

be the projected data matrix of view s in the common latent space.

Ideally, we would like to have the same projected data matrix of all v views, i.e., $Z_s = Z_{s'}, \forall s, s' = 1, \ldots, v$, but it is too restrictive to be feasible due to the heterogeneous features among views and the generalization of unseen data later on. We instead require that there is a shared classifier for all views in the common latent space, i.e., all views share the same coefficient matrix $W \in \mathbb{R}^{k \times c}$. Heuristically, the same shared classifier should be expected to correctly classify sets of "similar" projected data points of each view. This shared classifier can be regarded as some kind of similarity quantification indirectly implied upon projected points of all views.

For ease of presentation and flexibility to include slightly preprocessing the inputs as needed such as centering X_s to $X_s H_n$, let ϕ and ψ be two simple transformations on the inputs for preprocessing, and let

$$\widetilde{X}_s \equiv \left[\widetilde{\boldsymbol{x}}_1^{(s)}, \ldots, \widetilde{\boldsymbol{x}}_n^{(s)}\right] = \phi(X_s), \quad \widetilde{Y} \equiv \left[\widetilde{\boldsymbol{y}}_1^{(s)}, \ldots, \widetilde{\boldsymbol{y}}_n^{(s)}\right] = \psi(Y).$$

Later at the times of presenting particular learning methods, ϕ and ψ will be specified.

Our proposed MvOPLS model in its most plain form is formulated as

$$\min_{\{P_s\}, W} \sum_{s=1}^{v} \|\widetilde{Y} - W^{\mathrm{T}} P_s^{\mathrm{T}} \widetilde{X}_s\|_{\mathrm{F}}^2. \tag{1}$$

Later it will be empowered with judiciously chosen regularizers as situations call for. Accordingly, we will denote the projected data points of $\widetilde{\boldsymbol{x}}_i^{(s)}$ by $\widetilde{\boldsymbol{z}}_i^{(s)} = P_s^{\mathrm{T}} \widetilde{\boldsymbol{x}}_i^{(s)}$ and the projected data matrix by $\widetilde{Z}_s \equiv [\widetilde{\boldsymbol{z}}_1^{(s)}, \ldots, \widetilde{\boldsymbol{z}}_n^{(s)}] = P_s^{\mathrm{T}} \widetilde{X}_s$.

Fig. 1 shows the data transformation from input to output in OPLS on single-view data and MvOPLS on multiview data. In (1), each view has its own projection matrix P_s, but all views share the same classifier as determined by W that characterizes the similarity among projected points of all views.

For single-view data, i.e., $v = 1$, (1) reduces to OPLS [30]. For the convenience of the reader, in Section I of the Supplementary Material, we review OPLS for multivariate regression analysis and multiclass classification in detail and discuss its many nice properties. There could be other ways to extend OPLS to multiview learning, but we choose to impose a shared classifier in our MvOPLS (1) while let each view has its own projection matrix. The choice leads to some of the desirable properties to be explained in Section III-D later.

C. Optimization via GEP

For convenience of analysis, denote by $d = \sum_{s=1}^{v} d_s$, the total number of features from all v views, and by

$$P = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_v \end{bmatrix} \in \mathbb{R}^{d \times k}, \quad \widetilde{X} = \begin{bmatrix} \widetilde{X}_1 \\ \widetilde{X}_2 \\ \vdots \\ \widetilde{X}_v \end{bmatrix} \in \mathbb{R}^{d \times n}, \quad X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_v \end{bmatrix} \in \mathbb{R}^{d \times n} \tag{2}$$

the concatenations of $\{P_s\}$, $\{\widetilde{X}_s\}$, and $\{X_s\}$ introduced in Section III-B. Define

$$\widetilde{C} = \widetilde{X}\widetilde{X}^{\mathrm{T}} = \begin{bmatrix} \widetilde{C}_{1,1} & \widetilde{C}_{1,2} & \ldots & \widetilde{C}_{1,v} \\ \widetilde{C}_{2,1} & \widetilde{C}_{2,2} & \ldots & \widetilde{C}_{2,v} \\ \vdots & \vdots & \ddots & \vdots \\ \widetilde{C}_{v,1} & \widetilde{C}_{v,2} & \ldots & \widetilde{C}_{v,v} \end{bmatrix} \in \mathbb{R}^{d \times d} \tag{3}$$

and its block diagonal part

$$\widetilde{C}_{\mathrm{diag}} = \begin{bmatrix} \widetilde{C}_{1,1} & & & \\ & \widetilde{C}_{2,2} & & \\ & & \ddots & \\ & & & \widetilde{C}_{v,v} \end{bmatrix} \in \mathbb{R}^{d \times d} \tag{4}$$

where $\widetilde{C}_{s,t} = \widetilde{X}_s \widetilde{X}_t^{\mathrm{T}}, \forall s, t = 1, \ldots, v$.

Most commonly, $\widetilde{X} = X H_n$, but it may not be as we will see in Sections IV and V. When $\widetilde{X} = X H_n$, $\widetilde{C}_{s,t}$ coincides with the ordinary (cross-)covariance matrix between view s and view t, denoted conventionally by $C_{s,t} = X_s H_n X_t^{\mathrm{T}}$, which is the convention we will stick to in the rest of this article. We will also introduce $\widehat{C} = X X^{\mathrm{T}}$, and, accordingly, $\widehat{C}_{s,t}$ and $\widehat{C}_{\mathrm{diag}}$ for data that have not been preprocessed yet.

Problem (1) can be solved as a generalized eigenvalue problem (GEP). Specifically, we first use the first-order optimality condition of (1) with respect to W

$$\sum_{s=1}^{v} -2\ P_s^{\mathrm{T}} \widetilde{X}_s \left(\widetilde{Y} - W^{\mathrm{T}} P_s^{\mathrm{T}} \widetilde{X}_s\right)^{\mathrm{T}} = 0$$

to get

$$\begin{aligned} W &= \left(\sum_{s=1}^{v} P_s^{\mathrm{T}} \widetilde{C}_{s,s} P_s\right)^{-1} \sum_{s=1}^{v} P_s^{\mathrm{T}} \widetilde{X}_s \widetilde{Y}^{\mathrm{T}} \\ &= \left(P^{\mathrm{T}} \widetilde{C}_{\mathrm{diag}} P\right)^{-1} P^{\mathrm{T}} \widetilde{X} \widetilde{Y}^{\mathrm{T}}. \end{aligned} \tag{5}$$

Substituting this W in (5) back into (1), we obtain a reformulated problem of (1) as

$$\max_P \operatorname{tr}\Big(\big(P^{\mathrm{T}}\widetilde{C}_{\mathrm{diag}}P\big)^{-1}P^{\mathrm{T}}\widetilde{X}\widetilde{Y}^{\mathrm{T}}\widetilde{Y}\widetilde{X}^{\mathrm{T}}P\Big). \tag{6}$$

which is a ratio trace maximization problem.

In the following, we show that problem (6) is equivalent to a GEP. The proof is provided in Section II of the Supplementary Material.

Proposition 1: Let A, $B \in \mathbb{R}^{m\times m}$ be symmetric and A is positive definite. The problem

$$\max_P \operatorname{tr}\Big(\big(P^{\mathrm{T}}AP\big)^{-1}P^{\mathrm{T}}BP\Big) \tag{7}$$

is equivalent to

$$\max_P \operatorname{tr}\big(P^{\mathrm{T}}BP\big):\ \text{s.t.}\ \ P^{\mathrm{T}}AP = I_k \tag{8}$$

whose optimizer P can be constructed by packing the normalized eigenvectors of matrix pencil $B-\lambda A$ associated with its k top (largest) eigenvalues.

According to Proposition 1, (6) is equivalent to

$$\max_P\ \operatorname{tr}\big(P^{\mathrm{T}}\widetilde{X}\widetilde{Y}^{\mathrm{T}}\widetilde{Y}\widetilde{X}^{\mathrm{T}}P\big):\ \text{s.t.}\ \ P^{\mathrm{T}}\widetilde{C}_{\mathrm{diag}}P = I_k \tag{9}$$

whose optimizer P can be made of the eigenvectors associated with the top k eigenvalues of the following GEP:

$$\widetilde{X}\widetilde{Y}^{\mathrm{T}}\widetilde{Y}\widetilde{X}^{\mathrm{T}}P = \widetilde{C}_{\mathrm{diag}}P\Lambda \tag{10}$$

where Λ is the diagonal matrix of the k largest eigenvalues of matrix pencil $\widetilde{X}\widetilde{Y}^{\mathrm{T}}\widetilde{Y}\widetilde{X}^{\mathrm{T}}-\lambda\widetilde{C}_{\mathrm{diag}}$ [36], [37]. After P is obtained, the optimal W is recovered by (5).

D. Desirable Properties

Although MvOPLS (1) seems like a simple extension of OPLS to multiview subspace learning in appearance, it possesses many nice properties, some of which are inherited from OPLS, while others are unique to multiview subspace learning.

1) MvOPLS is in a least squares formulation as OPLS, and it can naturally take advantage of many types of known outputs to guide the learning of the latent common space, for example, multitarget regression with $Y \in \mathbb{R}^{c\times n}$ and multilabel classification with $Y \in \{0,1\}^{c\times n}$ and multiclass classification with $Y \in \{0,1\}^{c\times n}$ and $\mathbf{1}_c^{\mathrm{T}}Y = \mathbf{1}_n$.

2) MvOPLS admits analytical optimal solution. In Section III-C, we show that MvOPLS is equivalent to GEP, which can be solved efficiently by existing linear algebra algorithms and packages [36], [38]–[40].

3) MvOPLS fulfills three learning objectives simultaneously: compact representations in the latent common subspace, mappings from original data spaces to output space, and a shared classifier as determined by W. All these make it an ideal candidate for further learning nonlinear multiview representation in deep neural networks as we will show later in Section V.

4) MvOPLS possesses the ability of learning a latent common space with contributions from all views. According to (5), the coefficients of the built-in classifier is $W = \sum_{s=1}^{v} P_s^{\mathrm{T}}\widetilde{X}_s\widetilde{Y}^{\mathrm{T}}$ since $P^{\mathrm{T}}\widetilde{C}_{\mathrm{diag}}P = I_k$ at the optimum of (9). To make a prediction for a new input $\boldsymbol{x}^{(s)}$ from view s, MvOPLS relies on the decision values

$$W^{\mathrm{T}}P_s^{\mathrm{T}}\widetilde{\boldsymbol{x}}^{(s)} = \widetilde{Y}\sum_{t=1}^{v}\widetilde{X}_t^{\mathrm{T}}P_tP_s^{\mathrm{T}}\widetilde{\boldsymbol{x}}^{(s)} = \widetilde{Y}\sum_{t=1}^{v}\widetilde{Z}_t^{\mathrm{T}}\widetilde{\boldsymbol{z}}^{(s)}. \tag{11}$$

$\widetilde{Z}_t^{\mathrm{T}}\widetilde{\boldsymbol{z}}^{(s)}$ can be interpreted as the learned similarities in the latent common space between $\boldsymbol{x}^{(s)}$ and $\boldsymbol{x}_i^{(t)}, \forall i = 1,\ldots,n$. We emphasize that (11) is distinctively different from the single-view OPLS, applied to just view s, in that in (11) all views, not just view s, are involved in the decision making, and hence, complementary and corroborative information from other views comes into play; (11) is distinctively different from the single-view OPLS, applied to the concatenated view of all views, in that MvOPLS learns a shared classifier for all views, and thus, the classifier can make viewwise prediction independently, whereas OPLS on the concatenated view can make prediction only when data from all views are put together.

5) MvOPLS, though in a supervised formulation, can include MCCA as a special case. Recall that MCCA [10], [13] is an unsupervised subspace learning method because of no class label information, but each instance of all views may be regarded as in its own class, and hence, there are n classes, i.e., $c = n$. This will transform originally unlabeled data into artificially pseudolabeled data with $Y = I_n$ because we assign a unique class label to each instance. Now, apply MvOPLS (1) with $\widetilde{X} = XH_n$ and $\widetilde{Y} = I_n$ to get

$$\min_{\{P_s\},W}\sum_{s=1}^{v}\|I_n - W^{\mathrm{T}}P_s^{\mathrm{T}}X_sH_n\|_{\mathrm{F}}^2. \tag{12}$$

It is equivalent to the following MCCA formulation [13]:

$$\max_P \sum_{s=1}^{v}\sum_{t=1}^{v}\operatorname{tr}\big(P^{\mathrm{T}}C_{s,t}P\big):\ \text{s.t.}\ \ \sum_{s=1}^{v}P_s^{\mathrm{T}}C_{s,s}P_s = I_k. \tag{13}$$

6) MvOPLS is a versatile framework that can incorporate different types of prior knowledge across multiple views from three perspectives: model parameters, decision values of built-in classifiers, and projected points in the latent common space. This point will be elaborated in detail in Section IV.

E. Computational Complexity

The computational complexity of MvOPLS consists of three parts: $O(n\sum_{s=1}^{v}d_s^2)$ for $\widetilde{C}_{\mathrm{diag}}$, $O(nd^2)$ for $\widetilde{X}\widetilde{Y}^{\mathrm{T}}\widetilde{Y}\widetilde{X}^{\mathrm{T}}$, and $O(d^3)$ for the generalized eigenvalue decomposition problem (9) by a dense method or $O(d^2k)$ by iterative methods [36], [40]. Hence, the computational complexity is $O(d^3+nd^2)$ or $O(nd^2)$ depending on the choice of methods for GEP.

IV. Regularized Multiview OPLS

As MvOPLS (1) is formulated as least squares, regularization techniques can be leveraged to regulate model parameters and integrate prior knowledge to shape the similarity among projected points for the purpose of narrowing or even eliminating heterogeneity gaps among different views. In considering the special structure of MvOPLS, we will explore three types of regularizations with respect to different types of priors in

terms of: 1) model parameters; 2) decision values; and 3) latent projected points. Some examples of the three types will be demonstrated.

Before discussing these regularizers, we first propose a general framework for our regularized MvOPLS, based on the plain MvOPLS (1), that retains all of the appealing properties as enumerated in Section III-D.

A. A Generalized Formulation

Our treatment to integrate priors into the plain MvOPLS (1) for best learning is largely motivated by the past successful marriage of least squares and the popular regularization techniques widely used for regulating solutions of inverse problems in computational sciences [41]. In this section, we focus on a generalized regularized MvOPLS (GMvOPLS) framework given by

$$\min_{\{P_s\},W} \sum_{s=1}^{v} \|\widetilde{Y} - W^{\mathrm{T}} P_s^{\mathrm{T}} \widetilde{X}_s\|_{\mathrm{F}}^2 + \mathrm{tr}\big(W^{\mathrm{T}} P^{\mathrm{T}} A P W\big) + \mathrm{tr}\big(\Omega^{-1} P^{\mathrm{T}} B P\big) \quad (14)$$

where $A \in \mathbb{R}^{d\times d}$ and $B \in \mathbb{R}^{d\times d}$ are matrices derived from priors, and $\Omega \in \mathbb{R}^{k\times k}$ is a flexibility matrix purposely designed for ease of its numerical treatment and, more importantly, for retaining the properties of MvOPLS as enumerated in Section III-D.

For example, letting

$$\Omega = P^{\mathrm{T}}\big(\widetilde{C}_{\mathrm{diag}} + A\big)P \quad (15)$$

makes (14) equivalent to a GEP. First, given P, (14) has the analytic solution for W

$$W = \big[P^{\mathrm{T}}\big(\widetilde{C}_{\mathrm{diag}} + A\big)P\big]^{-1} P^{\mathrm{T}} \widetilde{X}\widetilde{Y}^{\mathrm{T}}. \quad (16)$$

Substituting (16) back into (14), we get

$$\min_{P} -\mathrm{tr}\Big(\big[P^{\mathrm{T}}\big(\widetilde{C}_{\mathrm{diag}} + A\big)P\big]^{-1} P^{\mathrm{T}} \widetilde{X}\widetilde{Y}^{\mathrm{T}} \widetilde{Y}\widetilde{X}^{\mathrm{T}} P\Big) + \mathrm{tr}\big(\Omega^{-1} P^{\mathrm{T}} B P\big).$$

According to Proposition 1 and with (15), we conclude that (14) is equivalent to GEP

$$\max_{P} \ \mathrm{tr}\big(P^{\mathrm{T}}\big(\widetilde{X}\widetilde{Y}^{\mathrm{T}}\widetilde{Y}\widetilde{X}^{\mathrm{T}} - B\big)P\big) \quad \text{s.t.} \ \ P^{\mathrm{T}}\big(\widetilde{C}_{\mathrm{diag}} + A\big)P = I_k. \quad (17)$$

We caution the reader that at its generality Ω may not be given as in (15) and then GMvOPLS (14) may not be equivalent to a GEP. In that case, how to solve (14) numerically becomes an issue that warrants further investigation. However, in the rest of this article, all instantiated models from (14), including the existing ones, are with (15).

We emphasize that the introduction of the additional terms in (14) to MvOPLS (1) does not destroy any of the enumerated properties, but only empowers it with the capability of incorporating priors for best learning, some of which will be demonstrated later.

It is worth noting that regularizations for MvOPLS (1) are not limited to the form of (14). In fact, one may use other types, for example, $\ell_{2,1}$ norm on P for sparse subspace learning intended for feature selection [42] and orthogonal projection for distance preservation [35], but they will change the solution structure and the resulting models require very different numerical techniques from those for solving a GEP, and therefore, we will not delve in them further in this article.

B. Regularizations

We present three types of regularizations with concrete examples designed to go with the proposed GMvOPLS (14) to achieve different objectives. Specifically, three categories are: 1) regularization on model parameter; 2) regularization on decision values; and 3) regularization on projected points. The three categories are inspired by the properties of MvOPLS presented in Section III-D. Category 1) regularizes model parameters to prevent the model from overfitting the input data, especially when the input data are small to avoid the ill-posed singularity issue in the generalized eigenvalue decomposition and to capture structure priors, such as sparsity or group sparsity if desired. Category 2) regulates view-specific prediction such that classification results are consistent among different views. Category 3) regulates the relationships among projected points from different views. In the following, we will elaborate on each category with examples.

1) Regularization on Model Parameters: The Tikhonov regularization is widely used to mitigate the problem of multicollinearity in linear regression in OPLS for single-view learning. Model parameters in (1) include W and $\{P_s\}_{s=1}^{v}$, but they appear in products $P_s W$, and thus, each can be treated as a single-matrix variable as far as mitigating multicollinearity is concerned. Therefore, we formulate our weighted Tikhonov regularizer as

$$\mathcal{R}_{\mathrm{tikh}}(W, \{P_s\}) = \sum_{s=1}^{v} \gamma_s \|P_s W\|_{\mathrm{F}}^2 = \mathrm{tr}\big(W^{\mathrm{T}} P^{\mathrm{T}} A_{\mathrm{tikh}} P W\big) \quad (18a)$$

where $\gamma_s \geq 0$ is the weight for view s and A_{tikh} is the block diagonal matrix

$$A_{\mathrm{tikh}} = \mathrm{diag}\big(\gamma_1 \ I_{d_1}, \gamma_2 \ I_{d_2}, \ldots, \gamma_v I_{d_v}\big) \in \mathbb{R}^{d\times d}. \quad (18b)$$

A_{tikh} takes the place of A in (14). It can be seen that $\widetilde{C}_{\mathrm{diag}} + A_{\mathrm{tikh}}$ is guaranteed positive definite.

The Tikhonov regularizer $\mathcal{R}_{\mathrm{tikh}}(W, \{P_s\})$ also prevents the magnitude of optimal $P_s W$ from being too large at an optimum.

2) Regularization on Decision Values: The decision function, also known as classifier, of MvOPLS for view s is given by (11)

$$g_s\big(\boldsymbol{x}^{(s)}\big) = W^{\mathrm{T}} P_s^{\mathrm{T}} \boldsymbol{x}^{(s)} \quad (19)$$

where $\boldsymbol{x}^{(s)}$ is any new data point of view s that needs a decision on.

Since there are s view-specific decision values, one from each view-specific data point of the same data instance, and at the same time, only one class label should be assumed for each data instance that consists of v data points, one for each view, it makes sense to regulate these v decisions across v

views to compel similarity among projected points in the latent common space. In what follows, we propose three ways for that purpose.

a) Mean discrepancy minimization: For view s, the mean of projected points and the decision value on the mean are given by

$$\mu_s = \frac{1}{n} P_s^{\mathrm{T}} X_s \mathbf{1}_n, \quad \overline{g}_s = W^{\mathrm{T}} \mu_s \tag{20}$$

respectively. It is reasonable to expect that these decision values $\overline{g}_s$ for all views be close to each other, assuming certain similarity among the views. The closeness among $\{\overline{g}_s\}$ can be enforced by the regularizer

$$\begin{aligned}\mathcal{R}_{\mathrm{mean}}(W, \{P_s\}; \{X_s\}) &= \frac{n}{2v} \sum_{s=1}^{v} \sum_{t=1}^{v} \|\overline{g}_s - \overline{g}_t\|^2 \\ &= \mathrm{tr}\big(W^{\mathrm{T}} P^{\mathrm{T}} A_{\mathrm{mean}} P W\big) \end{aligned} \tag{21a}$$

where

$$A_{\mathrm{mean}} = \mathrm{diag}\left(\left[\frac{1}{n} X_s \mathbf{1}_n \mathbf{1}_n^{\mathrm{T}} X_s^{\mathrm{T}}\right]_{s=1}^{v}\right) - \frac{1}{nv} X_s \mathbf{1}_n \mathbf{1}_n^{\mathrm{T}} X_s^{\mathrm{T}}. \tag{21b}$$

b) View consistency: The decision values can be parameterized by the representer theorem [43] for least squares, that is, the decision values of view s can be represented by a weighted combination of the input data points, given by

$$P_s W = X_s \boldsymbol{\beta}_s \tag{22a}$$

$$g_s(X_s) = W^{\mathrm{T}} P_s^{\mathrm{T}} X_s = \boldsymbol{\beta}_s^{\mathrm{T}} X_s^{\mathrm{T}} X_s \tag{22b}$$

where $\boldsymbol{\beta}_s \in \mathbb{R}^{n \times c}$ is the weight matrix. Note that $X_s^{\mathrm{T}} X_s$ is the linear kernel of view s. According to (22), we have $\boldsymbol{\beta}_s = X_s^{\dagger} P_s W$, where $X_s^{\dagger} = (X_s^{\mathrm{T}} X_s)^{-1} X_s^{\mathrm{T}}$ is the Moore–Penrose pseudoinverse of X_s [37, p.102]. Any consistency among views leads to similar kernels, and in turn, closeness among the weight matrices $\{\boldsymbol{\beta}_s\}_{s=1}^{v}$, which leads to the following regularizer:

$$\begin{aligned}\mathcal{R}_{\mathrm{vc}}(W, \{P_s\}; \{X_s\}) &= \frac{1}{2} \sum_{s=1}^{v} \sum_{t=1}^{v} \|\boldsymbol{\beta}_s - \boldsymbol{\beta}_t\|_{\mathrm{F}}^2 \\ &= \mathrm{tr}\big(W^{\mathrm{T}} P^{\mathrm{T}} A_{\mathrm{vc}} P W\big) \end{aligned} \tag{23a}$$

where A_{vc} is a v-by-v block matrix with its (s, t)th block given by

$$A_{\mathrm{vc}}(s, t) = \begin{cases} (v-1)\big(X_s^{\dagger}\big)^{\mathrm{T}} X_s^{\dagger}, & s = t \\ -\big(X_s^{\dagger}\big)^{\mathrm{T}} X_t^{\dagger}, & s \neq t. \end{cases} \tag{23b}$$

c) Maximum alignment: Alignment between the kernel matrix of class labels and that of the predicted values for each view is a proven useful criterion for learning projections [44], [45]. Denote by $g_s(X_s) = W^{\mathrm{T}} P_s^{\mathrm{T}} X_s$ the predicted soft labels and by K_Y the kernel matrix[1] of the class labels. The HSIC criterion [45] for multiview data is a natural regularizer for enforcing the alignment

$$\begin{aligned}\mathcal{R}_{\mathrm{hsic}}(W, \{P_s\}; \{X_s\}, Y) &= -\sum_{s=1}^{v} \mathrm{tr}\big(g_s(X_s)^{\mathrm{T}} g_s(X_s) H_n K_Y H_n\big) \\ &= \mathrm{tr}\big(W^{\mathrm{T}} P^{\mathrm{T}} A_{\mathrm{hsic}} P W\big) \end{aligned} \tag{24a}$$

[1]For example, in the multiclass classification, $K_Y = Y^{\mathrm{T}} \Sigma^{-1} Y$, where $\Sigma = Y Y^{\mathrm{T}}$.

where

$$A_{\mathrm{hsic}} = -\mathrm{diag}\big(\big[X_s H_n K_Y H_n X_s^{\mathrm{T}}\big]_{s=1}^{v}\big). \tag{24b}$$

We point out that (21) captures the first-order statistics of the decision values, while (23) and (24) characterize the second-order statistics.

3) Regularization on Projected Points: According to (1), the projected data in the common space is

$$\widetilde{Z}_s = P_s^{\mathrm{T}} \widetilde{X}_s \quad \forall s = 1, \ldots, v. \tag{25}$$

Regularizers can be designed directly or indirectly on these projected points to achieve certain objectives. In the following, we will showcase two commonly used priors.

a) Embedding consistency: We may expect that the projected points of the same instance from different views are close to each other. This expectation can be maintained by keeping the following regularizer:

$$\begin{aligned}\mathcal{R}_{\mathrm{ec}}\big(\{P_s\}; \big\{\widetilde{X}_s\big\}\big) &= \frac{1}{2} \sum_{s=1}^{v} \sum_{t=1}^{v} \|\widetilde{Z}_s - \widetilde{Z}_t\|_{\mathrm{F}}^2 \\ &= \mathrm{tr}\big(P^{\mathrm{T}} B_{\mathrm{ec}} P\big) \end{aligned} \tag{26a}$$

in check, where

$$B_{\mathrm{ec}} = v \widetilde{C}_{\mathrm{diag}} - \widetilde{C}. \tag{26b}$$

In fact, (26) is analogous to MCCA discussed in Section III-D for unlabeled data. For supervised classification, this regularizer can be used to explore the cross-view correlation from unlabeled data.

b) Class separability: The class separability criterion has been popularly used to learn discriminatory features in a low-dimensional space by LDA for multiclass classification. For view s, the within-class scatter matrix $S_w^{(s)}$ and the between-class scatter matrix $S_b^{(s)}$ can be written as

$$S_b^{(s)} = X_s \left(Q - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^{\mathrm{T}}\right) X_s^{\mathrm{T}}, \quad S_w^{(s)} = X_s H_n X_s^{\mathrm{T}} - S_b^{(s)} \tag{27}$$

where

$$\Sigma = Y Y^{\mathrm{T}}, \quad Q = Y^{\mathrm{T}} \Sigma^{-1} Y. \tag{28}$$

LDA can also be used to regulate projection matrix P_s of each view. The LDA regularizer is

$$\begin{aligned}\mathcal{R}_{\mathrm{lda}}(\{P_s\}; \{X_s\}) &= \sum_{s=1}^{v} \mathrm{tr}\left(P_s^{\mathrm{T}} \left(S_w^{(s)} - (\lambda - 1) S_b^{(s)}\right) P_s\right) \\ &= \mathrm{tr}\big(P^{\mathrm{T}} B_{\mathrm{lda}} P\big) \end{aligned} \tag{29a}$$

where

$$B_{\mathrm{lda}} = \mathrm{diag}\left(\left[C_{s,s} - \lambda X_s \left(Q - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^{\mathrm{T}}\right) X_s^{\mathrm{T}}\right]_{s=1}^{v}\right). \tag{29b}$$

Each diagonal block of B_{lda} stands for the difference between covariance matrix $C_{s,s}$ and the scaled between-class scatter matrix $\lambda S_b^{(s)}$, and the scaling parameter λ is the tradeoff parameter between the two scatter matrices. Regularizer (29) essentially plays the role in analogy to the fractional formulation of LDA for each view.

TABLE I
MODELS UNDER THE PROPOSED REGULARIZED MVOPLS FRAMEWORK (14) WITH DIFFERENT REGULARIZATION

model	$\widetilde{X}$	$\widetilde{Y}$	Regularization	A	B
MCCA [13], [10]	XH_n	I_n	N/A	0	0
MvLDA [32]	XH_n	$\Sigma^{-1/2}Y$	$\mathcal{R}_{\text{tikh}}(W,\{P_s\})$	A_{tikh}	0
MvDA [4]	XH_n	$\Sigma^{-1/2}Y$	$\mathcal{R}_{\text{mean}}(W,\{P_s\};\{X_s\})$	A_{mean}	0
MvDA-VC [4]	XH_n	$\Sigma^{-1/2}Y$	$\mathcal{R}_{\text{mean}}(W,\{P_s\};\{X_s\})+\lambda\mathcal{R}_{\text{vc}}(W,\{P_s\};\{X_s\})$	$A_{\text{mean}}+\lambda A_{\text{vc}}$	0
MvMDA [21]	XH_n	$H_c\Sigma^{-1/2}Y$	$\mathcal{R}_{\text{hsic}}(W,\{P_s\};\{X_s\},Y)$	A_{hsic}	0
MLDA [7]	XH_n	I_n	$\mathcal{R}_{\text{lda}}(\{P_s\};\{X_s\},Y)$	0	B_{lda}
GMA [5]	XH_n	I_n	$\mathcal{R}_{\text{lda}}(\{P_s\};\{X_s\},Y,\Omega)+\mathcal{R}_{\text{hsic}}(W,\{P_s\};\{X_s\},Y)$	A_{hsic}	B_{lda}
MvOPLS (proposed model)	XH_n	Y	$\mathcal{R}_{\text{tikh}}(W,\{P_s\})$	A_{tikh}	0
MvLDA-CCA (proposed model)	XH_n	$\Sigma^{-1/2}Y$	$\lambda\mathcal{R}_{\text{ec}}(\{P_s\};\{\widetilde{X}_s\})$	0	λB_{ec}

For regularizations over projected points, it is proper to add certain weighting constraints for ease of optimization. Instead of (26) and (29), more generally, we may use

$$\mathcal{R}_{\text{ec}}\big(\{P_s\};\{\widetilde{X}_s\},\Omega\big)=\text{tr}\big(\Omega^{-1}P^{\text{T}}B_{\text{ec}}P\big) \tag{30}$$

$$\mathcal{R}_{\text{lda}}(\{P_s\};\{X_s\},Y,\Omega)=\text{tr}\big(\Omega^{-1}P^{\text{T}}B_{\text{lda}}P\big) \tag{31}$$

where Ω is as in (15).

C. Connections to Existing Models

In Section III-D-5), we explained that MCCA [10], [13] can be regarded as a special case of MvOPLS. Next, we will show that many existing multiview subspace learning models can be reformulated to fit in the proposed regularized MvOPLS framework (14) with different choices of regularizers, including MvLDA [32], MvDA [4], MvDA-VC [4], MvMDA [21], MLDA [7], and GMA [5], as shown in Table I. Details can be found in Section III of the Supplementary Material. These methods more or less spawn from the GEP formulation (10) of plain MvOPLS (1), mostly based on heuristic and plausible arguments at best. Fitting them into MvOPLS (1) combined with suitable regularizers really puts firm foundations underneath them and deepens our understanding as to how they work the way they work and, as a result, can help the practitioner tremendously when it comes to pick up a proper model for the underlying learning task that comes with certain prior information.

1) Existing Models Explained Under Framework (14): The six existing methods discussed in Table I together with MCCA can be partitioned into two groups based on whether the involved MvOPLS least squares parts take class labels into account or not. The first group includes MCCA, MLDA, and GMA that do not use labeled data to construct their least squares parts, while the second group consists of MvLDA, MvDA, MvDA-VC, and MvMDA that do. However, MLDA and GMA in the first group do incorporate labeled data via regularizers, such as $\mathcal{R}_{\text{lda}}$ and $\mathcal{R}_{\text{hsic}}$. The methods in the second group may or may not use regularizers with prior knowledge. The choices of both input–output transformations (from X and Y to $\widetilde{X}$ and $\widetilde{Y}$, respectively) and regularizers are the key factors that differentiate one from another. For example, GMA differs from MLDA in that GMA takes an additional regularizer $\mathcal{R}_{\text{hsic}}$ to minimize the within-class scatter of each view. MvMDA takes the centered normalized label matrix and supervised MvOPLS with $\mathcal{R}_{\text{hsic}}$, while MvDA takes the normalized label matrix and unsupervised MvOPLS with $\mathcal{R}_{\text{mean}}$. As these regularizers are the consequences of some level of understanding of data, our proposed generalized formulation (14) provides a helpful guideline to the practitioner in either choosing a proper existing method with compatible regularizers or designing an entirely new model.

D. Proposed Novel Models

The proposed unified framework (14) offers much flexibility and easiness to instantiate new models through incorporating different inputs/outputs and regularizations in consideration of underlying learning objectives. Evidently, there are many possible ways of combinations to integrate one or more regularizers discussed in Section IV-B into MvOPLS (14), and it makes no sense for us to exhaust them all here. The practitioner should choose proper regularizers or customize one for the learning task in question. To demonstrate this point, we showcase two novel models.

1) MvOPLS With Tikhonov Regularizer (Namely MvOPLS): Tikhonov regularization is the default when it comes to regularize ill-posed least squares problems [41]. It effectively boosts small singular values to damp high-frequency noises. Here, for MvOPLS, the Tikhonov regularizer $\mathcal{R}_{\text{tikh}}$ in (18) can be used to stabilize the resulting GEP. MvOPLS with the Tikhonov regularizer is formulated as

$$\min_{\{P_s\},W}\sum_{s=1}^{v}\|\widetilde{Y}-W^{\text{T}}P_s^{\text{T}}\widetilde{X}_s\|_{\text{F}}^2+\mathcal{R}_{\text{tikh}}(W,\{P_s\}). \tag{32}$$

Or equivalently

$$\begin{aligned}&\max_{P}\ \text{tr}\big(P^{\text{T}}\widetilde{X}\widetilde{Y}^{\text{T}}\widetilde{Y}\widetilde{X}^{\text{T}}P\big)\\&\text{s.t.}\ \ P^{\text{T}}\big(\widetilde{C}_{\text{diag}}+A_{\text{tikh}}\big)P=I_k.\end{aligned} \tag{33}$$

This differs from (9) derived from the plain MvOPLS (1) in replacing $\widetilde{C}_{\text{diag}}$ by $\widetilde{C}_{\text{diag}}+A_{\text{tikh}}$ that makes latter better conditioned and the associated GEP less sensitive to noise and rounding errors [37], [46]. While this is a good thing to have, the Tikhonov regularizer does not bring much informative prior into the model.

2) MvOPLS With Embedding Consistency (Namely, MvLDA-CCA): We can require some closeness of the projected points of corresponding data instances among all views through maintaining embedding consistency by

incorporating regularizer $\mathcal{R}_{\text{ec}}$ into the plain MvOPLS. The new model is with

$$\widetilde{X} = XH_n, \quad \Sigma = YY^{\mathrm{T}}, \ \widetilde{Y} = \Sigma^{-1/2}Y \tag{34}$$

and formulated as

$$\min_{\{P_s\},W} \sum_{s=1}^{\upsilon} \|\widetilde{Y} - W^{\mathrm{T}}P_s^{\mathrm{T}}\widetilde{X}_s\|_{\mathrm{F}}^2 + \lambda\mathcal{R}_{\text{ec}}\big(\{P_s\}; \{\widetilde{X}_s\}, \Omega\big) \tag{35}$$

where $\Omega = P^{\mathrm{T}}\widetilde{C}_{\text{diag}}P$. Or equivalently

$$\begin{aligned} &\max_{P} \ \operatorname{tr}\left(P^{\mathrm{T}}\left[X\left(Q - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}}\right)X + \lambda\big(\widetilde{C} - \upsilon\widetilde{C}_{\text{diag}}\big)\right]P\right) \\ &\text{s.t. } P^{\mathrm{T}}\widetilde{C}_{\text{diag}}P = I_k \end{aligned} \tag{36}$$

where Q is as defined in (28).

V. Deep Regularized Multiview OPLS

Regularized MvOPLS aims to learn a set of linear projections. Lately, extending a linear projection method to a nonlinear one via the kernel trick becomes almost mechanical and immediate because the extension is often very much straightforward. The case for MvOPLS methods so far is no different. However, as pointed out in [12], kernel-based nonlinear extensions encounter several drawbacks.

1) Nonlinear representations are limited by the fixed kernel function.
2) Inner products between any two of input data points are required, and therefore, the training set has to be stored and repeatedly called for during the entire testing phase.
3) The time required to train a subspace learning model or compute the representations of new data points scales poorly with the size of the training set.

Those drawbacks are intrinsic to any kernel-based nonlinear extension and there is no way to get around them. Deep learning methods seem to be potent alternatives that use a completely different technique. They have been used rather successfully to learn a set of nonlinear parametric functions for subspace-based multiview learning [12], [21], [32], such as MvLDA [32] and MvMDA [21]. MvLDA takes a ratio trace formulation of LDA as its objective function by concatenating all views into one, while MvMDA takes a trace ratio as its original objective function but minimizes it approximately as a ratio trace via GEP because of the availability of high-quality numerical linear algebra packages. In that sense, numerically MvMDA also takes a ratio trace formulation nonetheless. As discussed in [33], ratio trace and trace ratio actually yield different projections.

Our regularized MvOPLS (14) takes a ratio trace formulation and is equivalent to GEP (17). We propose a nonlinear extension via deep networks in the general form

$$\max_{P} \ \operatorname{tr}\big(P^{\mathrm{T}}\mathscr{B}P\big) \ \text{ s.t. } \ P^{\mathrm{T}}\mathscr{A}P = I_k \tag{37}$$

where

$$\mathscr{B} = f\big(\{\boldsymbol{h}_s(X_s)\}_{s=1}^{\upsilon}, Y\big), \quad \mathscr{A} = g\big(\{\boldsymbol{h}_s(X_s)\}_{s=1}^{\upsilon}, Y\big)$$

are some matrix-valued functions of $\{\boldsymbol{h}_s(X_s)\}_{s=1}^{\upsilon}$ parameterized by υ independent deep networks and label matrix Y, and $\{\boldsymbol{h}_s\}_{s=1}^{\upsilon}$ is a set of nonlinear functions of the deep networks. According to (17), all the regularized models listed in Table I can be formulated into (37) as

$$\begin{aligned} g\big(\{\boldsymbol{h}_s(X_s)\}_{s=1}^{\upsilon}, Y\big) &= \widetilde{C}_{\text{diag}} + A \\ f\big(\{\boldsymbol{h}_s(X_s)\}_{s=1}^{\upsilon}, Y\big) &= \widetilde{X}\widetilde{Y}^{\mathrm{T}}\widetilde{Y}\widetilde{X}^{\mathrm{T}} - B \end{aligned}$$

where the matrices in the right-hand sides are defined similarly as before except with X_s replaced by $\boldsymbol{h}_s(X_s), \forall s$.

Following [12], we will use multiple stacked layers with nonlinear activation functions as the deep network architecture. The ith layer in the network for view s has $m_s^{(i)}$ units, and the output layer has k units. The output of the first layer for input $\boldsymbol{x}^{(s)}$ from view s is

$$\boldsymbol{h}_s^{(1)} = \sigma\big(V_s^{(1)}\boldsymbol{x}^{(s)} + \boldsymbol{b}_s^{(1)}\big) \in \mathbb{R}^{m_s^{(1)}}$$

where $V_s^{(1)} \in \mathbb{R}^{m_s^{(1)} \times d_s}$ is the weight matrix, $\boldsymbol{b}_s^{(1)} \in \mathbb{R}^{m_s^{(1)}}$ is the vector of biases, and $\sigma : \mathbb{R} \to \mathbb{R}$ is a nonlinear activation function. The output $\boldsymbol{h}_s^{(1)}$ can then be used as the input to the next layer whose output $\boldsymbol{h}_s^{(2)} = \sigma(V_s^{(2)}\boldsymbol{h}_s^{(1)} + \boldsymbol{b}_s^{(2)}) \in \mathbb{R}^{m_s^{(2)}}$, and the construction repeats ℓ times until the final output

$$\boldsymbol{h}_s(\boldsymbol{x}_s) \equiv \boldsymbol{h}_s^{(\ell)} = \sigma\big(V_s^{(\ell)}\boldsymbol{h}_s^{(\ell-1)} + \boldsymbol{b}_s^{(\ell)}\big) \in \mathbb{R}^k$$

is reached. The same construction process can be used for each of the υ views. As a result, we have a set of nonlinear functions $\{\boldsymbol{h}_s\}_{s=1}^{\upsilon}$ with ℓ layers and their associated parameters $\{V_s^{(i)}, \boldsymbol{b}_s^{(i)}\}$ for $s = 1, \ldots, \upsilon, i = 1, \ldots, \ell$. To simplify the notation, we have suppressed the dependency of the nonlinear transformed matrix $\boldsymbol{h}_s(X_s) \in \mathbb{R}^{k \times n}$ on the network parameters.

We further rewrite (37) as a standard eigenvalue problem so that the gradient of the transformed objective with respect to network parameters can be computed by automatic differentiation tools. Specifically, let the Cholesky decomposition of $\mathscr{A}$ be

$$\mathscr{A} = \Psi^{\mathrm{T}}\Psi. \tag{38}$$

To ensure that $\mathscr{A}$ is positive definite, the regularizer $\mathcal{R}_{\text{tikh}}$ is applied to all methods studied in this article. Let

$$U = \Psi P \Rightarrow P = \Psi^{-1}U. \tag{39}$$

Problem (37) is equivalent to

$$\max_{U} \ \operatorname{tr}\big(U^{\mathrm{T}}\Psi^{-\mathrm{T}}\mathscr{B}\Psi^{-1}U\big): \ \text{s.t. } \ U^{\mathrm{T}}U = I_k \tag{40}$$

which is equivalent to calculating the partial eigendecomposition

$$\Psi^{-\mathrm{T}}\mathscr{B}\Psi^{-1}U = U\Lambda \tag{41}$$

where $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_k)$ consists of the largest k eigenvalues $\lambda_1 \geq \lambda_2 \geq, \ldots, \geq \lambda_k$ of $\Psi^{-\mathrm{T}}\mathscr{B}\Psi^{-1}$ and U of the associated eigenvectors. After the optimal U is obtained, we recover optimal P using (39). The optimal objective function value is then given by

$$\operatorname{tr}\big(P^{\mathrm{T}}\mathscr{B}P\big) = \operatorname{tr}\big(U^{\mathrm{T}}\Psi^{-\mathrm{T}}\mathscr{B}\Psi^{-1}U\big) = \sum_{i=1}^{k}\lambda_i. \tag{42}$$

Treating the negative of the objective of (40) as loss, we actually minimize the loss over network parameters

TABLE II
MULTIVIEW DATASETS USED IN THE EXPERIMENTS, WHERE THE NUMBER OF FEATURES FOR EACH VIEW IS SHOWN INSIDE THE BRACKET

Data set	n	c	view 1	view 2	view 3	view 4	view 5	view 6
Mfeat	2000	10	fac (216)	fou (76)	kar (64)	mor (6)	pix (240)	zer (47)
Ads	3279	2	url+alt+caption (588)	origurl (495)	ancurl (472)	-	-	-
Caltech101-7	1474	7	CENTRIST (254)	GIST (512)	LBP (1180)	HOG (1008)	CH (64)	SIFT-SPM (1000)
Caltech101-20	2386	20	CENTRIST (254)	GIST (512)	LBP (1180)	HOG (1008)	CH (64)	SIFT-SPM (1000)
Scene15	4310	15	CENTRIST (254)	GIST (512)	LBP (531)	HOG (360)	SIFT-SPM (1000)	-
NUS-wide-object	23953	31	BOW (500)	CH (64)	CM55 (255)	CORR (144)	EDH (73)	WT (128)
Pascal	1000	20	Text (100)	Image (1024)	-	-	-	-
TVGraz	2058	10	Text (100)	Image (1024)	-	-	-	-
Wikipedia	2866	10	Text (100)	Image (1024)	-	-	-	-

$\{V_s^{(i)}, \boldsymbol{b}_s^{(i)}\}, s = 1, \ldots, v, i = 1, \ldots, \ell$, and the projection matrices $\{P_s\}_{s=1}^v$ simultaneously via the gradient descent method.

VI. EXPERIMENTS

A. Datasets

The statistics of the nine datasets with their corresponding descriptions are summarized in Table II.

The first six multiview datasets are used for multiview feature extraction evaluated through multiclass classification. Multiple features (Mfeat)[2] and Internet advertisements (Ads)[3] are downloaded from UCI's machine learning repository, where the descriptions of each view can be found in their documentations. Image datasets Caltech101[4] [47] and Scene15[5] [48] are created by applying the following descriptors to each image: CENTRIST [49], GIST [50], LBP [51], histogram of oriented gradient (HOG), color histogram (CH), and scale invariant feature transform (SIFT)-spatial pyramid matching (SPM) [48]. Note that we drop CH from Scene15 due to the gray-level images, and Caltech101 with two datasets consisting of seven and 20 categories are used by following [52]. NUS-wide-object is a Web image dataset consisting of six precomputed low-level features.[6]

The last three datasets, TVGraz [53], Wikipedia [54], and Pascal [55] are employed for cross-modal retrieval, where the image query is used to retrieve text articles and vice-versa. As pointed out in [56], these three datasets demonstrate different characteristics. Both image and text classifications are low in accuracy for Pascal. On Wikipedia, image classification has low accuracy, but its text classification accuracy is high. TVGraz has good accuracies for both text and image. These datasets are also used in [56], where the training/testing data splits are at: 1558/500 for TVGraz, 2173/693 for Wikipedia, and 700/300 for Pascal.

B. Compared Methods

The methods to be compared are as follows.

1) The six existing supervised methods discussed in Section IV-C: MvLDA [32], MvDA [4], MvDA-VC [4], MvMDA [21], MLDA [7], and GMA [5].
2) Unsupervised method MCCA [13].
3) The two new variants: the plain MvOPLS combined with the Tikhonov regularizer (33), denoted simply by MvOPLS, and MvLDA-CCA (36) in Section IV-D.
4) Their nonlinear extensions via deep networks proposed in Section V.

There are nine methods in items 1)–3), along with two variants of each that will be evaluated. The first variants, distinguished by attaching a prefix "D" to each of them, are their nonlinear extensions via deep networks in item 4). The second variants, distinguished by attaching a suffix "p" to each of them, are the results of the input data reduction by first applying PCA to each view in order to reduce the dimension of each view while retaining 95% energy but otherwise the same methods. For example associated with MvDA, MvDAp stands for MvDA applied to the PCA reduced data, and DMvDA is the nonlinear extension of MvDA via deep networks.

To prevent $\widetilde{C}_{\text{diag}} + A$ in the unified form (17) from being singular or nearly singular, the Tikhonov regularization (18) is applied to all methods.

C. Experimental Settings

We first evaluate the baseline methods for multiview feature extraction in terms of classification. All methods aim to learn a set of linear/nonlinear projections, which transform the data points of each view to points in the common space. Classification is then conducted in the common space. As observed in [2], the concatenation of the projected points from all views as the new representation of the input instance is proper for use by a regression algorithm, and the main finding of CCA there is that there is little loss of predictive power by using the reduced data in the lower dimensional space and yet the regression problem gains a lower sample complexity due to that the reduced multiview data matrix resides in $\mathbb{R}^{vk \times n}$. It is worth noting that the new representation of multiview data is consistent with our proposed framework based on least squares.

The proposed regularized MvOPLS has its built-in classifier, but some variants, such as MCCA, MLDA, and GMA, do not because their least squares losses are independent of the class labels. To make fair comparisons of all baseline methods in terms of classification performance, we seek an independent classifier for performance evaluation. Among them, linear support vector machines (SVMs) and 1-nearest neighbor classifier have been popularly used in the literature [4], [5], [12], [57].

[2]https://archive.ics.uci.edu/ml/datasets/Multiple+Features
[3]https://archive.ics.uci.edu/ml/datasets/internet+advertisements
[4]http://www.vision.caltech.edu/Image_Datasets/Caltech101/
[5]https://figshare.com/articles/15-Scene_Image_Dataset/7007177
[6]https://lms.comp.nus.edu.sg/wp-content/uploads/2019/research/nuswide/NUS-WIDE.html

TABLE III

MEAN ACCURACY AND STANDARD DEVIATION BY NINE METHODS AND THEIR TWO VARIANTS ON THE FIRST SIX MULTIVIEW DATASETS IN TABLE II OVER 10 RANDOM SPLITS OF 10% TRAINING AND 90% TESTING. THE NUMBER IN THE BRACKET IS THE RANK OF EACH METHOD FOR GIVEN DATA. THE LAST COLUMN IS THE AVERAGE RANK OF EACH METHOD OVER SIX DATASETS. A SMALLER RANK NUMBER STANDS FOR A BETTER METHOD

Method	Mfeat	Ads	Caltech101-7	Caltech101-20	Scene15	NUS-wide-object	rank
MCCA	79.11 ± 1.36 (21)	91.05 ± 1.51 (26)	88.53 ± 2.03 (25)	63.54 ± 2.58 (25)	43.41 ± 1.93 (27)	34.53 ± 0.79 (24)	24.7
MvOPLS	74.29 ± 1.94 (23)	92.04 ± 1.42 (23)	93.80 ± 1.04 (21)	84.75 ± 1.21 (22)	55.87 ± 1.26 (23)	34.97 ± 0.37 (20)	22.0
MvDA	74.05 ± 2.23 (25)	91.62 ± 1.65 (25)	92.77 ± 3.63 (23)	84.09 ± 1.29 (23)	55.78 ± 1.52 (24)	34.96 ± 0.37 (22)	23.7
MvDA-VC	89.06 ± 1.74 (18)	93.04 ± 2.35 (19)	94.21 ± 0.82 (20)	88.76 ± 1.04 (18)	75.75 ± 2.18 (19)	34.81 ± 0.52 (23)	19.5
MvLDA	95.46 ± 0.80 (8)	92.90 ± 1.62 (20)	93.13 ± 1.15 (22)	90.53 ± 0.69 (10)	92.24 ± 0.60 (12)	27.22 ± 0.70 (27)	16.5
MvMDA	41.13 ± 9.31 (27)	91.67 ± 1.66 (24)	77.47 ± 3.51 (26)	52.60 ± 1.40 (26)	77.15 ± 1.19 (18)	31.28 ± 0.37 (26)	24.5
MLDA	78.96 ± 1.32 (22)	90.92 ± 1.52 (27)	88.56 ± 2.12 (24)	63.81 ± 2.55 (24)	44.96 ± 1.62 (26)	34.48 ± 0.78 (25)	24.7
GMA	41.14 ± 9.30 (26)	93.07 ± 1.81 (17)	76.21 ± 3.63 (27)	52.18 ± 1.38 (27)	77.16 ± 1.19 (17)	35.66 ± 0.57 (18)	22.0
MvLDA-CCA	74.19 ± 2.40 (24)	93.20 ± 1.80 (16)	94.59 ± 0.93 (19)	84.91 ± 1.18 (21)	57.34 ± 1.05 (22)	34.97 ± 0.37 (21)	20.5
MCCAp	88.41 ± 1.70 (19)	92.53 ± 1.01 (21)	95.25 ± 0.45 (16)	88.82 ± 0.77 (16)	72.24 ± 1.08 (20)	37.54 ± 0.41 (16)	18.0
MvOPLSp	95.23 ± 0.86 (11)	95.21 ± 0.76 (13)	95.87 ± 0.32 (14)	91.54 ± 0.61 (9)	92.92 ± 0.52 (7)	37.63 ± 0.54 (12)	11.0
MvDAp	95.23 ± 0.86 (12)	95.29 ± 0.71 (12)	95.99 ± 0.49 (12)	91.58 ± 0.67 (7)	92.95 ± 0.49 (6)	37.62 ± 0.53 (14)	10.5
MvDA-VCp	95.21 ± 0.88 (14)	95.32 ± 0.77 (11)	95.95 ± 0.48 (13)	91.55 ± 0.65 (8)	92.84 ± 0.54 (8)	37.63 ± 0.55 (13)	11.2
MvLDAp	93.53 ± 1.17 (17)	93.05 ± 0.94 (18)	95.17 ± 0.97 (18)	90.01 ± 0.76 (12)	45.83 ± 5.16 (25)	35.53 ± 0.67 (19)	18.2
MvMDAp	95.32 ± 0.69 (10)	95.34 ± 0.61 (10)	96.31 ± 1.04 (10)	87.41 ± 1.29 (19)	92.54 ± 0.54 (10)	36.28 ± 0.31 (17)	12.7
MLDAp	88.41 ± 1.69 (20)	92.49 ± 1.02 (22)	95.25 ± 0.45 (17)	88.82 ± 0.78 (17)	72.24 ± 1.09 (21)	37.55 ± 0.41 (15)	18.7
GMAp	95.44 ± 0.75 (9)	95.44 ± 0.59 (8)	96.73 ± 0.77 (7)	87.02 ± 1.20 (20)	92.38 ± 0.57 (11)	38.60 ± 0.38 (10)	10.8
MvLDA-CCAp	95.22 ± 0.87 (13)	95.44 ± 0.76 (9)	95.85 ± 0.35 (15)	91.60 ± 0.71 (6)	92.79 ± 0.49 (9)	37.64 ± 0.54 (11)	10.5
DMCCA	95.17 ± 0.64 (15)	93.95 ± 0.49 (15)	96.61 ± 0.33 (8)	90.28 ± 0.66 (11)	81.68 ± 1.31 (15)	40.11 ± 0.35 (7)	11.8
DMvOPLS	95.85 ± 0.42 (3)	95.76 ± 0.42 (7)	96.83 ± 0.37 (6)	92.58 ± 0.35 (3)	93.31 ± 0.56 (5)	41.12 ± 0.29 (2)	4.3
DMvDA	95.61 ± 0.50 (6)	95.78 ± 0.47 (5)	96.84 ± 0.39 (4)	92.57 ± 0.42 (4)	93.36 ± 0.50 (3)	41.01 ± 0.38 (3)	4.2
DMvDA-VC	95.61 ± 0.50 (7)	95.78 ± 0.47 (6)	96.84 ± 0.39 (5)	92.57 ± 0.42 (5)	93.36 ± 0.50 (4)	41.01 ± 0.38 (4)	5.2
DMvLDA	96.35 ± 0.73 (2)	95.82 ± 0.36 (4)	97.60 ± 0.56 (2)	92.72 ± 0.63 (2)	94.25 ± 0.24 (2)	40.81 ± 0.36 (5)	2.8
DMvMDA	95.76 ± 0.59 (4)	95.91 ± 0.35 (2)	97.25 ± 0.54 (3)	89.71 ± 1.12 (14)	92.24 ± 0.35 (13)	39.27 ± 0.36 (9)	7.5
DMLDA	95.09 ± 0.72 (16)	94.45 ± 0.34 (14)	96.45 ± 0.33 (9)	89.89 ± 0.73 (13)	80.64 ± 0.81 (16)	39.83 ± 0.42 (8)	12.7
DGMA	95.74 ± 0.78 (5)	95.84 ± 0.43 (3)	96.18 ± 0.56 (11)	88.90 ± 0.94 (15)	90.70 ± 0.45 (14)	40.55 ± 0.48 (6)	9.0
DMvLDA-CCA	**96.67 ± 0.41 (1)**	**96.05 ± 0.30 (1)**	**98.01 ± 0.37 (1)**	**93.69 ± 0.40 (1)**	**94.31 ± 0.45 (1)**	**41.60 ± 0.36 (1)**	**1.0**

We will evaluate baselines in terms of SVMs since it is consistent with the built-in classifier of the proposed framework. Specifically, each dataset is split into training and testing sets. Each baseline method takes in a training set and outputs the learned projections and the new representation of the training set. The classifier is trained on the new representation of the training set. In the testing step, the testing set is first transformed to the common space via the given projections, and then the classifier is applied to make predictions of the testing data. We repeat the experiment for each baseline method over ten randomly drawn training and testing sets, and the mean accuracy with standard deviation on testing sets is reported.

Regularized MvOPLS and its variants share some common parameters, including the regularization parameters for $\mathcal{R}_{\text{tikh}}$ and the dimension k of the common space. In addition, MvDA-VC, MLDA, GMA, MvLDA-CCA, and their nonlinear versions have another regulating parameter λ for an additional regularization term. For simplicity, we set $\gamma_s = \gamma = 10^{-4}, \forall s$ in (18), and the second regularization parameter is set to $\lambda = 10^{-2}$ for all experiments. The dimension k is an important parameter for all subspace learning methods. Following the convention, we will evaluate all baseline methods over a set of ks. For Mfeat data, $k \in \{2, 3, 4, 5, 6\}$ is used since there are only six morphological features. For other datasets, $k \in \{2, 3, 5{:}5{:}50\}$ is used. The architecture of deep networks used for all nonlinear methods follows [12], where the widths of the hidden layers are 500 and 500, and there are three layers, including the output layer. During the training process, we take the full-batch optimization approach, as suggested in [12]. All nonlinear extensions are implemented in Pytorch [58] for tensor operations and eigenvalue decompositions. The Adam optimizer is used with the learning rate set to 10^{-3}, and others are set by default. It is worth noting that our work in this article mainly focuses on the generalized framework and demonstrates its versatility to recast existing methods and inspire new models as needed, and thus fine-tuning all involved parameters to achieve the best possible performance by each method is not our main concern.

Cross-modal retrieval is different from multiview feature extraction. After the common space is learned on the training data, the testing data are used to query each other through the common space. We take the retrieval method proposed in [54] with L_2 metric to evaluate the performance of each method. Following [5], we set the latent dimension to 20 for the last three datasets in Table II for cross-modal multimedia retrieval.

D. Performance Evaluation via Multiview Feature Extraction

The classification performance of all methods on the first six datasets in Table II is compared from three different perspectives: the best overall accuracy of each method, the accuracy by varying the dimension of the common space, and the best accuracy by varying the training ratios.

1) Overall Classification Performance: We first evaluate the nine methods and their two variants by comparing their best accuracies over all ks with 10% training and 90% testing split of data, and the results are shown in Table III. We have the following observations: 1) supervised methods significantly outperform unsupervised MCCA; 2) methods (with suffix "p")

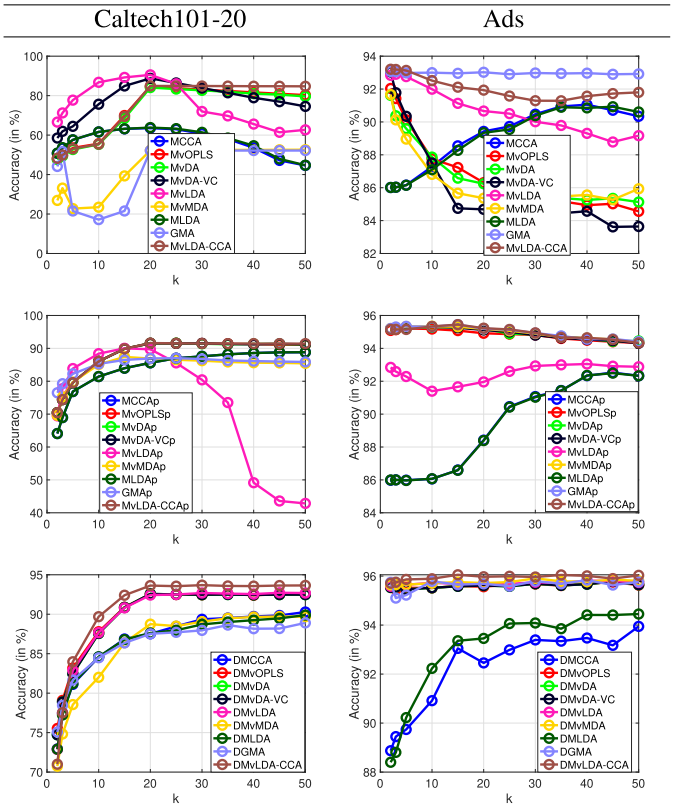

Fig. 2. Accuracies of nine methods and their variants on Caltech101-20 and Ads with 10% training and 90% testing data split as k varies.

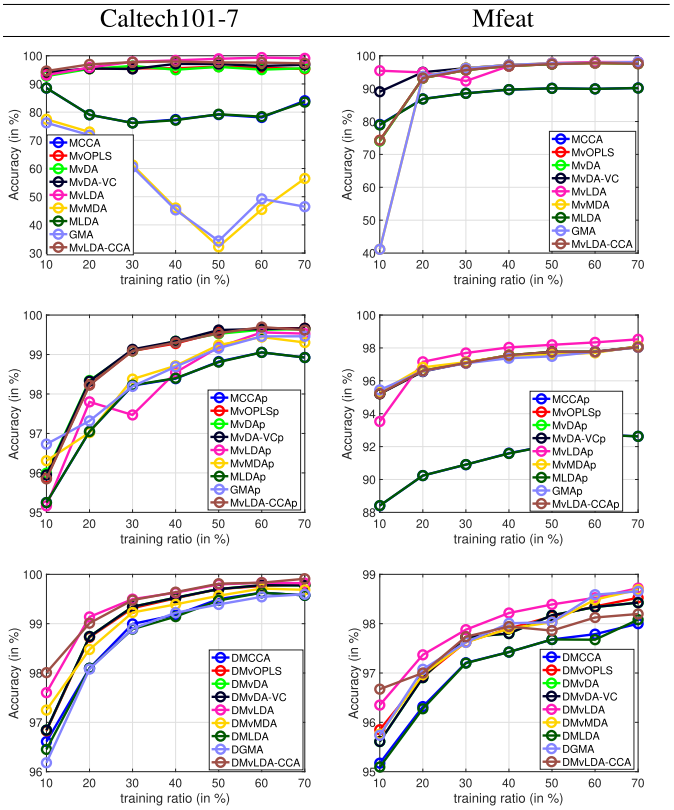

Fig. 3. Accuracies of nine methods and their two variants on Caltech101-7 and Mfeat with varying training data ratio from 10% to 70%.

trained on data of reduced dimensions have better accuracies than the same methods on original features (without suffix "p" and prefix "D"), and nonlinear transformations via deep networks (with suffix "D") perform better than the same linear methods (with or without suffix "p"); 3) among the supervised MvOPLS models, GMA and MLDA demonstrate relatively worse results than others (that is possibly caused by the model assumption that their least squares in MvOPLS are built on the unlabeled data); and 4) our proposed method DMvLDA-CCA consistently outperforms others over all six datasets.

To get an overall picture of the nine methods and their two variants over the six datasets, we rank them in terms of their accuracies on each dataset, from 1 to 27 with 1 being the best and 27 being the worst. The average ranking over the six datasets is reported to measure the overall performance of each method. The top four methods are DMvLDA, DMvOPLS, DMvDA, and DMvLDA-CCA, where the two newly proposed models, DMvLDA-CCA and DMvOPLS, are in the top 1 and 4, respectively.

2) Impact of Common Subspace: The impact of common subspace on the performance of classification is evaluated by varying dimension k of the subspace in a given range. Results of the nine methods and their two variants on Caltech101-20 and Ads are shown in Fig. 2. It can be observed that deep variants consistently achieve better accuracy as k increases. However, those models on original data can behave somehow unpredictably. Among them, MvLDAp behaves very differently from its other counterparts. In any case, the deep variants produce consistently better results than their two counterparts. Moreover, our proposed methods DMvLDA-CCA and MvLDA-CCAp show consistently better accuracy than their counterparts over most of the tested ks.

3) Impact of the Size of Training Data: We further evaluate the impact of the training ratio on the performance of classification. The results are shown in Fig. 3 where the training ratio is varied from 10% to 70% for the nine methods and their two variants on Mfeat and Caltech101-7. We observe that deep variants perform consistently better as the training ratio increases. Their two counterparts show similar trends but achieve lower accuracies compared with corresponding deep variants. Another observation is that MvOPLS and its variants can achieve relatively better accuracy for large training ratios on Mfeat, while DMvLDA-CCA achieves best results on small training ratios on both datasets. This is possibly because more labeled data degrades the importance of $\mathcal{R}_{\text{ec}}$.

4) Impact of the Layers of Deep Networks: We also evaluate the sensitivity of deep network variants with respect to the depth of networks from 2 to 8 layers. Each layer consists of a linear layer with a nonlinear activation function. Here, two functions: sigmoid and tanh are used. The widths of all layers are set to 500. The results on two datasets Caltech101-7 and Ads are shown in Fig. 4. With both activation functions, DGMA and DMLDA show a bit worse accuracy, while the proposed DMvLDA-CCA performs well in general. For sigmoid, it is shown that the best performance occurs near 2–4 layers, but the accuracy drops significantly when the

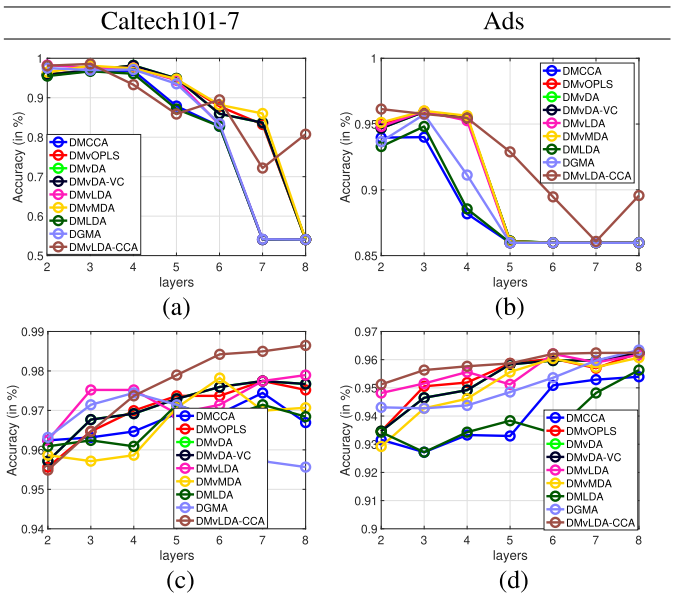

Fig. 4. Accuracies of nine deep network variants on Caltech101-7 and Ads as the number of layers varies from 2 to 8. (a) Sigmoid. (b) Sigmoid. (c) Tanh. (d) Tanh.

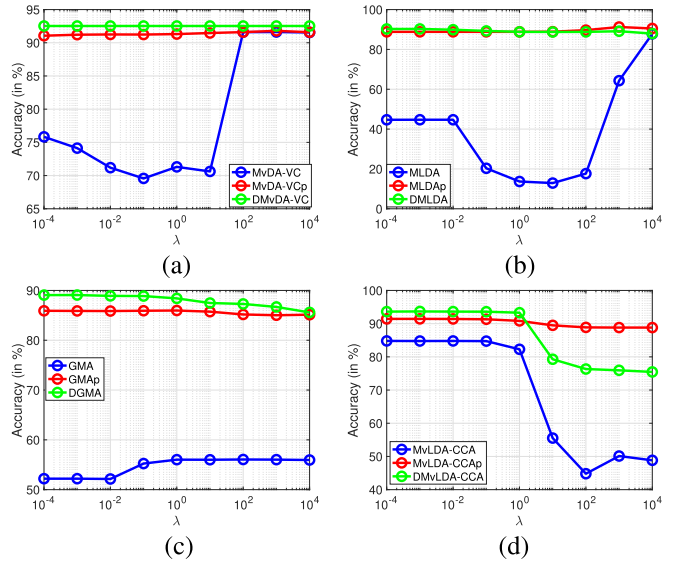

Fig. 5. Parameter sensitivity analysis of four methods on Caltech101-20 as parameter λ varies in a wide range from 10^{-4} to 10^{4}. (a) MvDA-VC. (b) MLDA. (c) GMA. (d) MvLDA-CCA.

number of layers becomes large. This has been observed in the literature. However, tanh generally requires more layers to reach similar performance and continues to improve as the number of layers increases.

5) Parameter Sensitivity Analysis: MvDA-VC, MLDA, GMA, MvLDA-CCA, and their nonlinear versions have another parameter λ (see Section IV-C). To investigate the impact of λ on the four models, we repeat the experiments in Section VI-D1 on Caltech101-20 by fixing $k = 50$ and varying $\lambda \in [10^{-4}, 10^{4}]$. Experimental results are shown in Fig. 5. The four methods show large variations on the original input data, but they behave less sensitively to λ for reduced data using either PCA or deep networks.

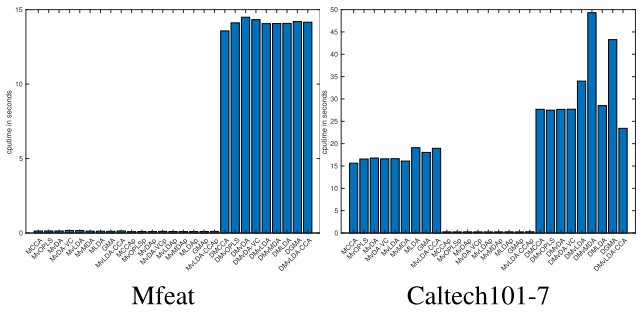

Fig. 6. CPU time in seconds of all compared methods on Mfeat and Caltech101-7 with 10% training.

TABLE IV

MAP SCORES OF THE NINE METHODS AND THEIR TWO VARIANTS ON THREE DATASETS, WHERE THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

Method	Pascal			TVGraz			Wikipedia		
	Image	Text	Ave	Image	Text	Ave	Image	Text	Ave
MCCA	0.102	0.085	0.093	0.191	0.282	0.237	0.144	0.174	0.159
MvOPLS	0.132	0.069	0.101	0.201	0.354	0.278	0.142	0.219	0.181
MvDA	0.131	0.070	0.101	0.203	0.351	0.277	0.146	0.219	0.183
MvDA-VC	0.134	0.069	0.102	0.141	0.348	0.244	0.131	0.196	0.163
MvLDA	0.070	0.068	0.069	0.148	0.225	0.186	0.130	0.138	0.134
MvMDA	0.159	0.067	0.113	0.163	0.359	0.261	0.140	0.224	0.182
MLDA	0.101	0.085	0.093	0.231	0.270	0.250	0.163	0.164	0.164
GMA	0.159	0.067	0.113	0.163	0.346	0.255	0.135	0.183	0.159
MvLDA-CCA	0.129	0.069	0.099	0.271	0.452	0.361	0.153	0.235	0.194
MCCAp	0.115	0.108	0.111	0.191	0.287	0.239	0.144	0.175	0.159
MvOPLSp	0.136	0.154	0.145	0.171	0.309	0.240	0.130	0.167	0.149
MvDAp	0.136	0.153	0.145	0.178	0.327	0.253	0.132	0.180	0.156
MvDA-VCp	0.137	0.154	0.145	0.138	0.340	0.239	0.125	0.193	0.159
MvLDAp	0.073	0.079	0.076	0.121	0.154	0.138	0.121	0.118	0.120
MvMDAp	0.129	0.173	0.151	0.159	0.355	0.257	0.131	0.220	0.175
MLDAp	0.115	0.108	0.111	0.234	0.276	0.255	0.163	0.164	0.163
GMAp	0.130	0.168	0.149	0.168	0.337	0.253	0.134	0.180	0.157
MvLDA-CCAp	0.136	0.149	0.143	0.269	0.400	0.335	0.151	0.217	0.184
DMCCA	0.132	0.118	0.125	0.115	0.220	0.168	0.121	0.156	0.138
DMvOPLS	0.153	0.161	0.157	0.406	0.411	0.409	0.198	0.236	0.217
DMvDA	0.155	0.162	0.158	0.420	0.399	0.410	0.210	0.223	0.217
DMvDA-VC	0.155	0.162	0.158	0.420	0.399	0.410	0.210	0.223	0.217
DMvLDA	0.133	0.120	0.126	0.296	0.290	0.293	0.129	0.161	0.145
DMvMDA	0.112	0.173	0.142	0.376	0.340	0.358	0.141	0.226	0.184
DMLDA	0.131	0.117	0.124	0.229	0.259	0.244	0.175	0.179	0.177
DGMA	0.122	0.179	0.151	0.261	0.276	0.268	0.146	0.179	0.162
DMvLDA-CCA	**0.188**	**0.198**	**0.193**	**0.422**	**0.453**	**0.438**	**0.219**	**0.241**	**0.230**

6) Empirical Time Complexity Analysis: We present the empirical time spent by each compared method on Mfeat ($k = 6$) and Caltech101-7 ($k = 50$) in Fig. 6. On Mfeat with 649 features in total, deep methods need much more time than others since many iterations are required even though each iteration takes less time. On Caltech101-7 with about 4018 features in total, deep methods take comparable time to counterpart methods with the original data as input, while counterparts with PCA become faster variants. This is because GEP on 4018 features dominates, while deep methods and counterparts with PCA can avoid this issue by working on reduced dimensions.

E. Performance Evaluation via Text-Image Retrieval

Text-image retrieval is used to evaluate MvOPLS and its variants on datasets whose two views are text and image, respectively. This task aims to retrieve an image (text) from a database for a given text (image) query. A correct retrieval is

the one with the same class as the query. The mean average precision (mAP) score is the performance measurement for text-image retrieval, and it has been popularly used [5], [56]. The parameters of all methods are the same as those used in Section VI-D.

Table IV shows the mAP scores by the nine methods and their two variants on three datasets. We observe that: 1) PCA is helpful on Pascal, but not on TVGraz and Wikipedia; 2) deep variants outperform their counterparts, and DMvLDA-CCA produces the best performance in terms of the mAP score over all three datasets; and 3) DMvLDA does not show good performance for cross-modal retrieval as in classification (Section VI-D). These results demonstrate that deep network variants of MvOPLS for learning nonlinear transformations are effective for certain models, but not always.

VII. Conclusion

Previously, several multiview subspace learning methods are formulated equivalently as GEPs of diverse matrix pencils, mostly based on heuristic understanding and plausible fixes to incorporate information that comes to light. There is not much coherent foundation laid out for them, not to mention a unified one upon which they can be built. In this article, we have proposed a unified multiview learning framework for that purpose. The framework not only provides a deep understanding of many existing methods from the viewpoint of regularized least squares but also guides the development of new methods. Furthermore, the framework affords a nonlinear variant via deep networks with little additional effort. Extensive experiments in terms of two multiview learning tasks validate the proposed framework, the two newly instantiated models, and the new deep variants.

The proposed framework can provide appealing flexibility to design effective models for a wide range of learning objectives, beyond what we have discussed in this article. For example, the sparse CCA [42], [59] can be reformulated under the proposed framework with sparsity regularization over projection matrices, and more importantly, this reformulation lends itself to immediate extensions for multiview data of more than two views and of nonlinear representations via a deep network, following our developments in this article. Our framework can also be extended for another learning paradigm, such as semisupervised multiview learning, missing view learning, and other label matrix learning approaches [60], which we will investigate elsewhere.

References

[1] Y. Li, M. Yang, and Z. Zhang, "A survey of multi-view representation learning," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 10, pp. 1863–1883, Oct. 2018.

[2] D. P. Foster, S. M. Kakade, and T. Zhang, "Multi-view dimensionality reduction via canonical correlation analysis," Toyota Technol. Inst. Chicago, Chicago, IL, USA, Tech. Rep. TTI-TR-2008-4, 2008. [Online]. Available: http://www.ttic.edu/technical_reports/ttic-tr-2008-4.pdf

[3] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," 2013, *arXiv:1304.5634*. [Online]. Available: http://arxiv.org/abs/1304.5634

[4] M. Kan, S. Shan, H. Zhang, S. Lao, and X. Chen, "Multi-view discriminant analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 188–194, Jan. 2015.

[5] A. Sharma, A. Kumar, H. Daume, and D. W. Jacobs, "Generalized multiview analysis: A discriminative latent space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2160–2167.

[6] L. Sun, S. Ji, and J. Ye, "Canonical correlation analysis for multilabel classification: A least-squares formulation, extensions, and analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 194–200, Jan. 2011.

[7] S. Sun, X. Xie, and M. Yang, "Multiview uncorrelated discriminant analysis," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 3272–3284, Dec. 2015.

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

[9] J. R. Kettenring, "Canonical analysis of several sets of variables," *Biometrika*, vol. 58, no. 3, pp. 433–451, 1971.

[10] A. A. Nielsen, "Multiset canonical correlations analysis and multispectral, truly multitemporal remote sensing data," *IEEE Trans. Image Process.*, vol. 11, no. 3, pp. 293–305, Mar. 2002.

[11] P. L. Lai and C. Fyfe, "Kernel and nonlinear canonical correlation analysis," *Int. J. Neural Syst.*, vol. 10, pp. 365–377, Oct. 2000.

[12] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1247–1255.

[13] J. Vía, I. Santamaría, and J. Pérez, "A learning algorithm for adaptive canonical correlation analysis of several data sets," *Neural Netw.*, vol. 20, no. 1, pp. 139–152, Jan. 2007.

[14] Z. Lei and S. Z. Li, "Coupled spectral regression for matching heterogeneous faces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1123–1128.

[15] D. Li, N. Dimitrova, M. Li, and I. K. Sethi, "Multimedia content processing through cross-modal association," in *Proc. 11th ACM Int. Conf. Multimedia (MULTIMEDIA)*, 2003, pp. 604–611.

[16] S. Wold *et al.*, "Multivariate data analysis in chemistry," in *Chemometrics*. Dordrecht, The Netherlands: Springer, 1984, pp. 17–95. [Online]. Available: https://link.springer.com/chapter/10.1007/978-94-017-1026-8_2#citeas

[17] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006. [Online]. Available: https://www.springer.com/gp/book/9780387310732

[18] J. Ye, "Least squares linear discriminant analysis," in *Proc. 24th Int. Conf. Mach. Learn. (ICML)*, 2007, pp. 1087–1093.

[19] T. Hastie, A. Buja, and R. Tibshirani, "Penalized discriminant analysis," *Ann. Statist.*, vol. 23, no. 1, pp. 73–102, Feb. 1995.

[20] T. Diethe, D. R. Hardoon, and J. Shawe-Taylor, "Multiview Fisher discriminant analysis," in *Proc. NIPS Workshop Learn. Multiple Sour.*, 2008.

[21] G. Cao, A. Iosifidis, K. Chen, and M. Gabbouj, "Generalized multiview embedding for visual recognition and cross-modal retrieval," *IEEE Trans. Cybern.*, vol. 48, no. 9, pp. 2542–2555, Sep. 2017.

[22] K. J. Worsley, J.-B. Poline, K. J. Friston, and A. C. Evans, "Characterizing the response of PET and fMRI data using multivariate linear models," *NeuroImage*, vol. 6, no. 4, pp. 305–319, Nov. 1997.

[23] L. Sun, S. Ji, S. Yu, and J. Ye, "On the equivalence between canonical correlation analysis and orthonormalized partial least squares," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, 2009, pp. 1230–1235. [Online]. Available: https://www.ijcai.org/Proceedings/09/Papers/207.pdf

[24] J. Arenas-García and G. Camps-Valls, "Efficient kernel orthonormalized PLS for remote sensing applications," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 10, pp. 2872–2881, Oct. 2008.

[25] S. Roweis and C. Brody, "Linear heteroencoders," in *Gatsby Computational Neuroscience Unit*. London, U.K.: Alexandra House, 1999.

[26] P. Horst, "Generalized canonical correlations and their applications to experimental data," *J. Clin. Psychol.*, vol. 17, no. 4, pp. 331–347, Oct. 1961.

[27] J. Yin and S. Sun, "Multiview uncorrelated locality preserving projection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3442–3455, Sep. 2019.

[28] K. Somandepalli, N. Kumar, R. Travadi, and S. Narayanan, "Multimodal representation learning using deep multiset canonical correlation," 2019, *arXiv:1904.01775*. [Online]. Available: http://arxiv.org/abs/1904.01775

[29] A. Benton, H. Khayrallah, B. Gujral, D. A. Reisinger, S. Zhang, and R. Arora, "Deep generalized canonical correlation analysis," in *Proc. 4th Workshop Represent. Learn. NLP (RepL NLP)*, 2019, pp. 1–6.

[30] J. Arenas-Garcıa and K. B. Petersen, "Kernel multivariate analysis in remote sensing feature extraction," in *Kernel Methods for Remote Sensing Data Analysis*, Sep. 2009, p. 329, ch. 14. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/9780470748992.ch14

[31] S. Yan, D. Xu, B. Zhang, and H.-J. Zhang, "Graph embedding: A general framework for dimensionality reduction," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2005, pp. 830–837.

[32] M. Kan, S. Shan, and X. Chen, "Multi-view deep network for cross-view classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4847–4855.
[33] H. Wang, S. Yan, D. Xu, X. Tang, and T. Huang, "Trace ratio vs. ratio trace for dimensionality reduction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.
[34] L. Wang, L.-H. Zhang, Z. Bai, and R.-C. Li, "Orthogonal canonical correlation analysis and applications," *Optim. Methods Softw.*, vol. 35, no. 4, pp. 1–21, 2020.
[35] L. Zhang, L. Wang, Z. Bai, and R.-C. Li, "A self-consistent-field iteration for orthogonal CCA," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jul. 28, 2020, doi: 10.1109/TPAMI.2020.3012541.
[36] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed. Baltimore, MD, USA: Johns Hopkins Univ. Press, 2013.
[37] G. W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*. Boston, MA, USA: Academic, 1990.
[38] E. Anderson *et al.*, *LAPACK Users' Guide*, 3rd ed. Philadelphia, PA, USA: SIAM, 1999.
[39] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, Eds., *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. Philadelphia, PA, USA: SIAM, 2000.
[40] R.-C. Li, "Rayleigh quotient based optimization methods for eigenvalue problems," in *Matrix Functions and Matrix Equations* (Series in Contemporary Applied Mathematics), vol. 19, Z. Bai, W. Gao, and Y. Su, Eds. Singapore: World Scientific, 2015, pp. 76–108.
[41] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. Philadephia, PA, USA: SIAM, 1998.
[42] M. Xu, Z. Zhu, X. Zhang, Y. Zhao, and X. Li, "Canonical correlation analysis with $L_{2,1}$-norm for multiview data representation," *IEEE Trans. Cybern.*, vol. 50, no. 11, pp. 4772–4782, Nov. 2020.
[43] B. Schölkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2002.
[44] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola, "On kernel-target alignment," *Adv. neural Inf. Process. Syst.*, vol. 14, 2001, pp. 367–373.
[45] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with Hilbert-Schmidt norms," in *Proc. Int. Conf. Algorithmic Learn. Theory*. Berlin, Germany: Springer, 2005, pp. 63–77. [Online]. Available: https://link.springer.com/chapter/10.1007/11564089_7
[46] R.-C. Li, "On perturbations of matrix pencils with real spectra, a revisit," *Math. Comput.*, vol. 72, no. 242, pp. 715–728, Apr. 2003.
[47] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," *Comput. Vis. Image Understand.*, vol. 106, no. 1, pp. 59–70, Jan. 2007.
[48] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2006, pp. 2169–2178.
[49] J. Wu and J. M. Rehg, "Where am I: Place instance and category recognition using spatial PACT," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[50] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.
[51] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.
[52] C. Deng, Z. Lv, W. Liu, J. Huang, D. Tao, and X. Gao, "Multi-view matrix decomposition: A new scheme for exploring discriminative information," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 2438–3444. [Online]. Available: https://www.ijcai.org/Proceedings/15/Papers/484.pdf
[53] I. Khan, A. Saffari, and H. Bischof, "TVGraz: Multi-modal learning of object categories by combining textual and visual features," in *Proc. AAPR Workshop*, 2009, pp. 213–224.
[54] N. Rasiwasia *et al.*, "A new approach to cross-modal multimedia retrieval," in *Proc. Int. Conf. Multimedia (MM)*, 2010, pp. 251–260.
[55] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier, "Collecting image annotations using Amazon's mechanical Turk," in *Proc. Workshop Creating Speech Lang. Data Amazon's Mech. Turk (NAACL HLT)*, 2010, pp. 139–147.
[56] J. C. Pereira and N. Vasconcelos, "On the regularization of image semantics by modal expansion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3093–3099.
[57] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1083–1092.
[58] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2019, pp. 8024–8035.
[59] D. M. Witten and R. J. Tibshirani, "Extensions of sparse canonical correlation analysis with applications to genomic data," *Stat. Appl. Genet. Mol. Biol.*, vol. 8, no. 1, pp. 1–27, Jan. 2009.
[60] X. Zhang, L. Wang, S. Xiang, and C. Liu, "Retargeted least squares regression algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 9, pp. 2206–2213, Sep. 2014.

Li Wang received the B.S. degree in information and computing science from China University of Mining and Technology, Jiangsu, China, in 2006, the M.S. degree from Xi'an Jiaotong University, Xi'an, Shaanxi, China, in 2009, and the Ph.D. degree from the Department of Mathematics, University of California at San Diego, San Diego, CA, USA, in 2014.

She was a Research Assistant Professor with the Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, Chicago, IL, USA, from 2015 to 2017, and a Post-Doctoral Fellow with the University of Victoria, Victoria, BC, Canada, in 2015, and Brown University, Providence, RI, USA, in 2014. She is currently an Assistant Professor with the Department of Mathematics, The University of Texas at Arlington, Texas, USA, and also with the Department of Computer Science and Engineering. Her research interests include large-scale optimization, polynomial optimization, and machine learning.

Ren-Cang Li received the B.S. degree from Xiamen University, Xiamen, China, in 1985, the M.S. degree from the Chinese Academy of Science, Beijing, China, in 1988, and the Ph.D. degree from the University of California at Berkeley, Berkeley, CA, USA, in 1995.

He is currently a Professor with the Department of Mathematics, The University of Texas at Arlington, Arlington, TX, USA. His research interests include floating-point support for scientific computing, large and sparse linear systems, eigenvalue problems, model reduction, machine learning, and unconventional schemes for differential equations.

Dr. Li received the 1995 Householder Fellowship in scientific computing from the Oak Ridge National Laboratory, a Friedman Memorial Prize in applied mathematics from the University of California at Berkeley in 1996, and the CAREER Award from NSF in 1999.

Wen-Wei Lin received the Ph.D. degree in applied mathematics/numerical analysis from Bielefeld University, Bielefeld, Germany.

He is currently the Life-Time Chair Professor of the Department of Applied Mathematics, National Chiao Tung University, Hsinchu, Taiwan. His current research interests include numerical solutions for large-scale eigenvalue problems and applications, fast solvers for 3-D Maxwell's equations, efficient numerical methods of 3-D surface morphing based on deformable models, conformal mapping, and data science.

Dr. Lin serves/served on several editorial boards of international journals, including the Editors-in-Chief for the *Annals of Mathematical Sciences and Applications* and the *Taiwanese Journal of Mathematics*, an Advisory Editor for *Numerical Linear Algebra With Applications*, and an Associate Editor for the *SIAM Journal on Matrix Analysis and Applications*, *Communications in Information and Systems*, the *Tamkang Journal of Mathematics*, and the *Electronic Transactions on Numerical Analysis*. He was a recipient of the Chern Prize at the International Consortium of Chinese Mathematicians in 2019.