Scalable and Flexible Multiview MAX-VAR Canonical Correlation Analysis

Xiao Fu, Kejun Huang, Mingyi Hong, Nicholas D. Sidiropoulos, and Anthony M.-C. So

***Abstract*— This paper considers generalized (multiview) canonical correlation analysis (GCCA) for large-scale datasets. GCCA is gaining renewed interest in various applications such as speech processing and natural language processing. The classic MAX-VAR GCCA problem can be solved optimally via eigen-decomposition of a matrix that compounds the (whitened) correlation matrices of the views. However, this route can easily lead to memory explosion and a heavy computational burden when the size of the views becomes large. In addition, it was unclear how to promote pre-specified structure (e.g. sparsity) on the canonical components sought, while structured component analysis is often desired in data analytics. In this work, we propose an alternating optimization (AO)-based algorithm to handle large-scale MAX-VAR GCCA as well as its variants that impose structure on the canonical components. The algorithm avoids instantiating the correlation matrices of the views and thus can achieve substantial saving in memory. It also maintains data sparsity, which can be exploited to alleviate the computational burden. Consequently, the proposed algorithm is highly scalable and flexible in handling a variety of regularizations. Convergence properties of the proposed algorithm are carefully studied: It is shown to converge to a critical point at a sublinear convergence rate using a certain class of (possibly non-smooth) structure-promoting regularizations; the algorithm also approaches a global optimum at a linear rate if the original MAX-VAR problem is considered. Simulations and large-scale word embedding tasks are employed to showcase the effectiveness of the proposed algorithm.**

***Index Terms*— Canonical correlation analysis, multiview, word embedding, optimization, scalability, feature selection**

I. Introduction

Canonical Correlation Analysis (CCA) [1] produces low dimensional representations by finding common structure of two or more views corresponding to the same entities. A view contains high-dimensional representations of the entities in some domain – e.g., the text and audio representations corresponding to a given word can be considered as different views of this word. CCA is able to deal with views that have different dimensions, and this flexibility is very useful in data fusion, where one is interested in integrating information gathered from different domains. Multiview analysis finds numerous applications in signal processing and machine learning, such as blind source separation [2], [3], direction-of-arrival estimation [4], wireless channel equalization [5], regression [6], clustering [7], speech modeling and recognition [8], [9], and word embedding [10], to name a few. Classical CCA was derived for the two-view case, but Generalized Canonical Correlation Analysis (GCCA) that aims at handling more than two views has a long history as well [11]. A typical application of GCCA, namely, multi-language word embedding, is shown in Fig. 1.

X. Fu, K. Huang and N.D. Sidiropoulos are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN55455, e-mail (xfu,huang663,nikos)@umn.edu. M. Hong is with Department of Industrial and Manufacturing Systems Engineering, Iowa State University, Ames, Iowa 50011, (515) 294-4111, Email: mingyi@iastate.edu. Anthony M.-C. So is with the Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong, Email: manchoso@se.cuhk.edu.hk

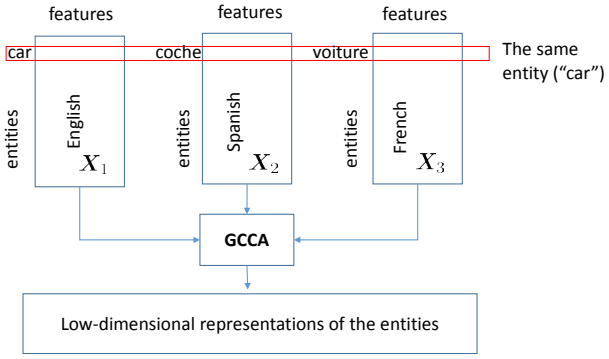

Fig. 1. Word embedding seeks low-dimensional representations of the entities that are well-aligned with human judgment. Different language data (i.e., $\boldsymbol{X}_1$-$\boldsymbol{X}_3$) can be considered as different views / feature spaces of the same entities. Applying GCCA to integrate multiple languages was shown to yield better embedding results relative to single-view analyses such as principle component analysis (PCA) [10].

Computationally, GCCA poses interesting and challenging optimization problems. Unlike the two-view case that admits an algebraically simple solution (via eigen-decomposition), GCCA is in general not easily solvable. Many prior works considered the GCCA problem with different cost functions [11], [12], while the proposed algorithms often can only extract a single canonical component and then find others through a deflation process, which is known to suffer from error propagation. CCA and GCCA can also pose serious scalability challenges, since they involve auto- and cross-correlations of different views and a whitening stage [13]. These procedures can easily lead to memory explosion and require a large number of flops for computation. They also destroy the sparsity of the data, which is usually what one relies upon to deal with large-scale problems. In recent years, effort has been spent on solving these scalability issues, but the focus is mostly on the two-view case [13]–[15].

Among all different formulations of GCCA, there is a particular one that admits a conceptually simple solution, the so-called MAX-VAR GCCA [11], [12], [16]. MAX-VAR GCCA was first proposed in [11], and its solution amounts to finding the 'directions' aligned to those exhibiting maximum

variance for a matrix aggregated from the (whitened) auto-correlations of the views. It can also be viewed as a problem of enforcing *identical* latent representations of different views as opposed to highly correlated ones, which is the more general goal of (G)CCA. The merit of MAX-VAR GCCA is that it can be solved via eigen-decomposition and finds all the canonical components simultaneously (i.e., no deflation is involved). In practice, MAX-VAR GCCA also demonstrates promising performance in various applications such as word embedding [10] and speech recognition [8]. On the other hand, MAX-VAR GCCA has the same scalability problem as the other GCCA formulations: It involves correlation matrices of different views and their inverses, which is prohibitive to even instantiate when the data dimension is large. The work in [10] provided a pragmatic way to circumvent this difficulty: PCA was first applied to each view to reduce the rank of the views, and then MAX-VAR GCCA was applied to the rank-truncated views. Such a procedure significantly reduces the number of parameters for characterizing the views and is feasible in terms of memory. However, truncating the rank of the views is prone to information loss, and thus leads to performance degradation.

Besides the basic (G)CCA formulations, *structured* (G)CCA [17] that seeks canonical components with pre-specified structure is often considered in applications. For example, sparse CCA is desired in data analytics for the purpose of discarding outlying or irrelevant features when performing CCA [18]–[20]. In multi-lingual word embedding [10], [14], [21], for example, it is known that there are many outlying features (i.e., outlying columns in $\boldsymbol{X}_i$ in Fig. 1), known as "stop words". In brain activation analysis, it is also believed that many voxels are irrelevant and should be somehow discarded [22], [23]. Gene analysis is another example [18]–[20]. Ideally, CCA seeks a few highly correlated latent components, and so it should naturally be able to identify and down-weight irrelevant features automatically. In practice, however, this ability is often impaired when correlations cannot be reliably estimated, when one only has access to relatively few and/or very noisy samples, or when there is model mismatch due to bad preprocessing (e.g., registration). In those cases, performing feature selection jointly with (G)CCA is well-motivated. However, introducing structure regularizations on the sought canonical components makes the optimization problem even harder, since many regularizers are non-differentiable.

Contributions In this work, our goal is to provide a scalable and flexible algorithmic framework for dealing with the MAX-VAR GCCA problem and its variants with structure-promoting regularizers. Instead of truncating the rank of the views as in [10], we keep the data *intact* and devise a scalable algorithmic framework to handle the formulated problem. Specifically, our idea is to deal with problem using a two-block alternating optimization (AO) algorithm. Under the AO framework, the proposed algorithm alternates between a regularized least squares subproblem and an orthogonality-constrained subproblem. The merit of this framework is that correlation matrices of the views never need to be explicitly instantiated, and the inversion procedure is avoided. Consequently, the algorithm consumes significantly less memory compared to that required by the original solution using eigen-decomposition. In addition, the proposed algorithmi can easily handle different structure-promoting regularizers without increasing memory and computational costs, including the feature-selective regularizers that we are mainly interested in.

Convergence properties of the algorithm are also carefully studied: We first show that the proposed algorithm *globally* converges to a Karush-Kuhn-Tucker (KKT) point of the formulated problem when a variety of regularizers are employed, even when the subproblems are solved in a grossly inexact manner. We also show that the optimality gap shrinks to at most $\mathcal{O}(1/r)$ after r iterations – i.e., at least a sublinear convergence rate can be guaranteed. In addition, we show that when there is no regularization, the proposed algorithm solves the classic MAX-VAR problem to *global optimality* and enjoys a *linear* convergence rate. The proposed algorithm is applied to synthetic data and a real large-scale word embedding problem, and promising results are observed.

Notation We use $\boldsymbol{X}$ and $\boldsymbol{x}$ to denote a matrix and a vector, respectively. $\boldsymbol{X}(m,:)$ and $\boldsymbol{X}(:,n)$ denote the mth row and the nth column of $\boldsymbol{X}$, respectively; in particular, $\boldsymbol{X}(:,n_1:n_2)$ ($\boldsymbol{X}(n_1:n_2,:)$) denotes a submatrix of $\boldsymbol{X}$ consisting of the n_1-n_2th columns (rows) of $\boldsymbol{X}$ (MATLAB notation). $\|\boldsymbol{X}\|_F$ and $\|\boldsymbol{X}\|_p$ for $p \geq 1$ denote the Frobenius norm and the matrix-induced p-norm, respectively. $\|\boldsymbol{X}\|_{p,1} = \sum_{i=1}^{m} \|\boldsymbol{X}(i,:)\|_p$ for $p \geq 1$ denotes the ℓ_p/ℓ_1-mixed norm of $\boldsymbol{X} \in \mathbb{R}^{m\times n}$. The superscripts "$T$", "$\dagger$", and "$-1$" denote the matrix operators of transpose, pseudo-inverse and inverse, respectively. The operator $<\boldsymbol{X},\boldsymbol{Y}>$ denotes the inner product of $\boldsymbol{X}$ and $\boldsymbol{Y}$.

II. Background

Let $\boldsymbol{X}_i \in \mathbb{R}^{L\times M_i}$ denote the ith view and its ℓth row $\boldsymbol{X}_i(\ell,:)$ be a feature vector that defines the ℓth data point (entity) in the ith view (cf. Fig. 1). The classic two-view CCA aims at finding common structure of the views via linear transformation. Specifically, the corresponding problem can be expressed as the following form [1]:

$$\min_{\boldsymbol{Q}_1,\boldsymbol{Q}_2} \ \|\boldsymbol{X}_1\boldsymbol{Q}_1 - \boldsymbol{X}_2\boldsymbol{Q}_2\|_F^2 \tag{1a}$$

$$\text{s.t. } \boldsymbol{Q}_i^T\left(\boldsymbol{X}_i^T\boldsymbol{X}_i\right)\boldsymbol{Q}_i = \boldsymbol{I}, \quad i=1,2, \tag{1b}$$

where the columns of $\boldsymbol{Q}_i \in \mathbb{R}^{M_i\times K}$ correspond to the K canonical components of view $\boldsymbol{X}_i$. Note that we are essentially maximizing the trace of the estimated cross-correlations between the dimension-reduced views, i.e., $\mathrm{Tr}(\boldsymbol{Q}_2^T\boldsymbol{X}_2^T\boldsymbol{X}_1\boldsymbol{Q}_1)$ subject to a normalization term in (1b). Problem (1) can be solved via the generalized eigen-decomposition, but this simple solution only applies to the two-view case. To analyze the case with more than two views, one natural thought is to extend the formulation in (1) to a pairwise matching cirterion, i.e., $\sum_{i=1}^{I-1}\sum_{j=i+1}^{I}\|\boldsymbol{X}_i\boldsymbol{Q}_i - \boldsymbol{X}_j\boldsymbol{Q}_j\|_F^2$ with orthogonality constraints on $\boldsymbol{X}_i\boldsymbol{Q}_i$ for all i, where I is the number of views. Such an extension leads to the so-called sum-of-correlations (SUMCOR) generalized or multiview CCA (GCCA) [11]. Unfortunately, the SUMCOR formulation has been shown to be NP-hard [24]. Another formulation of GCCA is more tractable: Instead of forcing pairwise similarity of

the reduced-dimension views, one can seek a common latent representation of different views, i.e., [8], [10]–[12], [16]

$$\min_{\{\boldsymbol{Q}_i\}_{i=1}^I, \boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}} \sum_{i=1}^{I} (1/2) \left\| \boldsymbol{X}_i\boldsymbol{Q}_i - \boldsymbol{G} \right\|_F^2, \tag{2}$$

where $\boldsymbol{G} \in \mathbb{R}^{L\times K}$ is a common latent representation of the different views. Conceptually, Problems (2) also finds highly correlated reduced-dimension views as SUMCOR does. The upshot of Problem (2) is that it "transfers" the difficult constraints $\boldsymbol{Q}_i^T\boldsymbol{X}_i^T\boldsymbol{X}_i\boldsymbol{Q}_i = \boldsymbol{I}$ to a single constraint $\boldsymbol{G}^T\boldsymbol{G} = \boldsymbol{I}$, and thus admits a *conceptually* simple algebraic solution, which, as we will show, has the potential to be scaled up to deal with very large problems. In this work, we will focus on Problem (2) and its variants.

Problem (2) is referred to as the MAX-VAR formulation of GCCA since the optimal solution amounts to taking principal eigenvectors of a matrix aggregated from the correlation matrices of the views. To explain, let us first assume that $\boldsymbol{X}_i$ has full column rank and solve (2) with respect to (w.r.t.) $\boldsymbol{Q}_i$, i.e., $\boldsymbol{Q}_i = \boldsymbol{X}_i^\dagger\boldsymbol{G}$, where $\boldsymbol{X}_i^\dagger = (\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T$. By substituting it back to (2), we see that an optimal solution $\boldsymbol{G}_{\rm opt}$ can be obtained via solving the following:

$$\boldsymbol{G}_{\rm opt} = \arg\max_{\boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}} \ {\rm Tr}\left(\boldsymbol{G}^T\left(\sum_{i=1}^{I}\boldsymbol{X}_i\boldsymbol{X}_i^\dagger\right)\boldsymbol{G}\right). \tag{3}$$

Let $\boldsymbol{M} = \sum_{i=1}^I \boldsymbol{X}_i\boldsymbol{X}_i^\dagger$. Then, an optimal solution is $\boldsymbol{G}_{\rm opt} = \boldsymbol{U}_M(:, 1:K)$, the first K principal eigenvectors of $\boldsymbol{M}$ [25]. Although Problem (2) admits a seemingly easy solution, implementing it in practice has two major challenges:

1) **Scalability Issues**: Implementing the eigen-decomposition based solution for large-scale data is prohibitive. As mentioned, instantiating $\boldsymbol{M} = \sum_{i=1}^I \boldsymbol{X}_i(\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T$ is not doable when L and M_i's are large. The matrix $\boldsymbol{M}$ is an $L\times L$ matrix. In applications like word embedding, L and M_i are the vocabulary size of a language and the number of features defining the terms, respectively, which can both easily exceed $100,000$. This means that the memory for simply instantiating $\boldsymbol{M}$ or $(\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}$ can reach 75GB. In addition, even if the views $\boldsymbol{X}_i$ are sparse, computing $(\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}$ will create large dense matrices and make it difficult to exploit sparsity in the subsequent processing. To circumvent these difficulties, Rastogi *et al.* [10] proposed to first apply the singular value decomposition (SVD) to the views, i.e., ${\rm svd}(\boldsymbol{X}_i) = \boldsymbol{U}_i\boldsymbol{\Sigma}_i\boldsymbol{V}_i^T$, and then let

$$\hat{\boldsymbol{X}}_i = \boldsymbol{U}_i(:, 1:P)\boldsymbol{\Sigma}_i(1:P, 1:P)(\boldsymbol{V}_i(:, 1:P))^T \approx \boldsymbol{X}_i,$$

where P is much smaller than M_i and L. This procedure enables one to represent the views with significantly fewer parameters, i.e., $(L+M_i+1)P$ compared to LM_i, and allows the original eigen-decomposition based solution to MAX-VAR GCCA to be applied; see more details in [10]. The drawback, however, is also evident: The procedure essentially truncates the rank of the views significantly (since in practice the views almost always have full column-rank, i.e., ${\rm rank}(\boldsymbol{X}_i) = M_i$), and rank-truncation may lose information. Therefore, it is much more appealing to deal with the original views directly.

2) **Structure-Promoting**: Aside from scalability, another aspect that is under-addressed by existing approaches is how to incorporate regularizations on $\boldsymbol{Q}_i$ to multiview large-scale CCA. Note that finding structured $\boldsymbol{Q}_i$ is well-motivated in practice. Taking multilingual word embedding as an example, $\boldsymbol{X}_i(m, n)$ represents the nth feature of the mth word in language i, which is usually defined by the co-occurrence frequency of term m and feature n (also a word in language i). However, many features of $\boldsymbol{X}_i$ may not be informative (e.g., "the" and "of") or not correlated to data in $\boldsymbol{X}_j$. These *irrelevant* or *outlying features* could result in unsatisfactory performance of GCCA if not taken into account. Under such scenarios, a more appealing formulation may include a row-sparse promoting regularization on $\boldsymbol{Q}_i$ so that some columns corresponding to the irrelevant features in $\boldsymbol{X}_i$ can be downweighted when seeking $\boldsymbol{Q}_i$. Structured (G)CCA is also desired in a variety of applications such as gene analytics and fMRI prediction [18]–[20], [22], [23].

III. Proposed Algorithm

In this work, we consider a scalable and flexible algorithmic framework for handling MAX-VAR GCCA and its variants with structure-promoting regularizers on $\boldsymbol{Q}_i$. We aim at offering simple solutions that are memory-efficient, admit light per-iteration complexity, and feature good convergence properties under certain mild conditions. Specifically, we consider the following formulation

$$\min_{\{\boldsymbol{Q}_i\}, \boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}} \sum_{i=1}^{I} (1/2) \left\| \boldsymbol{X}_i\boldsymbol{Q}_i - \boldsymbol{G} \right\|_F^2 + \sum_{i=1}^{I} g_i\left(\boldsymbol{Q}_i\right), \tag{4}$$

where $g_i(\cdot)$ is a regularizer that imposes a certain structure on $\boldsymbol{Q}_i$. Popular regularizers are $g_i(\boldsymbol{Q}_i) = \mu_i \cdot \|\boldsymbol{Q}_i\|_F^2$, $g_i(\boldsymbol{Q}_i) = \mu_i \cdot \|\boldsymbol{Q}_i\|_{2,1}$ and $g_i(\boldsymbol{Q}_i) = \mu_i \cdot \|\boldsymbol{Q}_i\|_{1,1}$, where $\mu_i \geq 0$ is a regularization parameter for balancing the least squares fitting term and the regularization term. The first regularizer is commonly used for controlling the energy of the dimension-reducing matrix $\boldsymbol{Q}_i$, which also has an effect of improving the conditioning of the $\boldsymbol{Q}$-subproblem. The latter two regularizers are used to select features automatically. To be specific, $g_i(\boldsymbol{Q}_i) = \mu_i\|\boldsymbol{Q}_i\|_{2,1}$ that we are mainly interested in has the ability of promoting rows of $\boldsymbol{Q}_i$ to be zeros (or approximately zeros), and thus can suppress the impact of the corresponding columns (features) in $\boldsymbol{X}_i$ – which is effectively feature selection. Combined regularizations such as $g_i(\cdot) = \mu_i\|\cdot\|_F^2 + \upsilon_i\|\cdot\|_{2,1}$ are also of interest, which could improve conditioning of the $\boldsymbol{Q}$-subproblem and perform feature selection simultaneously. The function $g_i(\boldsymbol{Q}_i) = \mu_i\|\boldsymbol{Q}_i\|_{1,1}$ also does feature selection, but different canonical components may use different features. Many other $g_i(\cdot)$'s can also be considered, such as non-negativity and smoothness regularizers. In this section, we propose an algorithm that can deal with the regularized and the original version of MAX-VAR GCCA under a unified framework.

A. Alternating Optimization

To deal with Problem (4), our idea is alternating optimization; i.e., we solve two subproblems w.r.t. $\{\boldsymbol{Q}_i\}$ and $\boldsymbol{G}$,

respectively. As will be seen, such a simple strategy will leads to highly scalable algorithms in terms of both memory and computational cost.

To begin with, let us assume that after r iterations the current iterate is $(\boldsymbol{Q}^{(r)}, \boldsymbol{G}^{(r)})$ where $\boldsymbol{Q} = [\boldsymbol{Q}_1^T, \ldots, \boldsymbol{Q}_I^T]^T$ and consider the subproblem

$$\min_{\boldsymbol{Q}_i} \ (1/2) \left\| \boldsymbol{X}_i \boldsymbol{Q}_i - \boldsymbol{G}^{(r)} \right\|_F^2 + g_i(\boldsymbol{Q}_i), \ \forall i. \tag{5}$$

The above problem is a regularized least squares problem. When $\boldsymbol{X}_i$ is large and sparse, many efficient algorithms can be considered to solve it. For example, the alternating direction method of multipliers (ADMM) [26] is frequently employed to handle Problem (5) in a scalable manner. However, ADMM is a primal-dual method that does not guarantee monotonic decrease of the objective value, which will prove useful in later convergence analysis. Hence, we propose to employ a simple proximal gradient (PG) method for handling Problem (5). By proximal gradient, we update $\boldsymbol{Q}_i$ by the following update rule:

$$\begin{aligned} \boldsymbol{Q}_i^+ &\leftarrow \mathtt{prox}_{\alpha_i g_i} \left(\hat{\boldsymbol{Q}}_i - \alpha_i \nabla_{\boldsymbol{Q}_i} f\left(\hat{\boldsymbol{Q}}, \boldsymbol{G}_i^{(r)} \right) \right) \qquad (6) \\ &= \arg\min_{\boldsymbol{Q}} \ \frac{1}{2} \left\| \hat{\boldsymbol{Q}}_i - \left(\hat{\boldsymbol{Q}}_i - \alpha_i \nabla_{\boldsymbol{Q}_i} f\left(\hat{\boldsymbol{Q}}, \boldsymbol{G}_i^{(r)} \right) \right) \right\|_F^2 + g_i(\boldsymbol{Q}_i) \end{aligned}$$

where $\boldsymbol{Q}_i^+$ and $\hat{\boldsymbol{Q}}_i$ denote the next and the current iterates of $\boldsymbol{Q}_i$, respectively, and we have used the notation $f(\boldsymbol{Q}, \boldsymbol{G}^{(r)}) = \sum_{i=1}^I \frac{1}{2} \left\| \boldsymbol{X}_i \boldsymbol{Q}_i - \boldsymbol{G}^{(r)} \right\|_F^2$, and

$$\nabla_{\boldsymbol{Q}_i} f\left(\boldsymbol{Q}, \boldsymbol{G}_i^{(r)} \right) = \boldsymbol{X}_i^T \boldsymbol{X}_i \boldsymbol{Q}_i - \boldsymbol{X}_i^T \boldsymbol{G}^{(r)}. \tag{7}$$

For many functions $g_i(\cdot)$, the proximity operator in (6) has closed-form or lightweight solutions [27]. For example, for the regularization of interest such as $g_i(\boldsymbol{Q}_i) = \mu_i \|\boldsymbol{Q}_i\|_F^2$, the solution to Problem (6) is simply

$$\boldsymbol{Q}_i^+ \leftarrow (\boldsymbol{X}_i^T \boldsymbol{X}_i + \mu_i \boldsymbol{I}) \hat{\boldsymbol{Q}}_i - \boldsymbol{X}_i^T \boldsymbol{G}^{(r)}, \tag{8}$$

and in this case PG boils down to gradient descent (GD). When $g_i(\boldsymbol{Q}_i) = \mu_i \|\boldsymbol{Q}_i\|_{2,1}$, the update rule becomes

$$\boldsymbol{Q}_i^+(m,:) \leftarrow \begin{cases} \boldsymbol{0}, & \|\boldsymbol{H}_i(m,:)\|_2 < \mu_i, \\ \left(1 - \frac{\mu_i}{\|\boldsymbol{H}_i(m,:)\|_2}\right) \|\boldsymbol{H}_i(m,:)\|_2, & \text{o.w.}, \end{cases} \tag{9}$$

where $\boldsymbol{H}_i = \hat{\boldsymbol{Q}}_i - \alpha_i \nabla_{\boldsymbol{Q}_i} f\left(\hat{\boldsymbol{Q}}, \boldsymbol{G}_i^{(r)} \right)$. For $g_i(\boldsymbol{Q}_i) = \mu_i \|\boldsymbol{Q}_i\|_{1,1}$, the update rule is similar to that in (9), which is known as the *soft-thresholding operator*.

By updating $\boldsymbol{Q}_i$ using the rule in (6) for T times where $T \geq 1$, we obtain $\boldsymbol{Q}_i^{(r+1)}$. Next, we consider solving the subproblem w.r.t. $\boldsymbol{G}$ when fixing $\{\boldsymbol{Q}_i\}_{i=1}^I$. Now we can drop the regularization term since it does not affect the cost value when $\boldsymbol{Q}_i$ is fixed. Then, the $\boldsymbol{G}$-subproblem amounts to the following:

$$\max_{\boldsymbol{G}^T \boldsymbol{G} = \boldsymbol{I}} \ \mathrm{Tr}\left(\boldsymbol{G}^T \sum_{i=1}^I \boldsymbol{X}_i \boldsymbol{Q}_i^{(r+1)} / I \right).$$

Therefore, an optimal solution of $\boldsymbol{G}$ is as follows: Let $\boldsymbol{P} = \sum_{i=1}^I \boldsymbol{X}_i \boldsymbol{Q}_i^{(r+1)}$. Then, we have $\boldsymbol{G}^{(r+1)} \leftarrow \boldsymbol{U}_P \boldsymbol{V}_P^T$, where $\boldsymbol{U}_P \boldsymbol{\Sigma}_P \boldsymbol{V}_P^T = \mathrm{svd}(\boldsymbol{P}, 'econ')$, and $\mathrm{svd}(\cdot, 'econ')$ denotes the economy-size SVD that produces $\boldsymbol{U}_P \in \mathbb{R}^{L \times K}$, $\boldsymbol{\Sigma}_P \in \mathbb{R}^{K \times K}$ and $\boldsymbol{V}^T \in \mathbb{R}^{K \times K}$. The above update is optimal in terms of solving the subproblem. In practice, one may also combine the knowledge of the previous iterate $\boldsymbol{G}^{(r)}$ and let

$$\boldsymbol{P} = \gamma \sum_{i=1}^I \boldsymbol{X}_i \boldsymbol{Q}_i^{(r+1)} / I + (1 - \gamma) \boldsymbol{G}^{(r)}, \tag{10}$$

where $\gamma \in (0, 1]$. Such a slight modification of forming $\boldsymbol{P}$ does not increase the complexity and is very easy to implement. More importantly, such a simple combination helps establish nice convergence properties of the algorithm, as we will see in the next section.

The algorithm is summarized in Algorithm 1, which we call the alternating optimization-based MAX-VAR GCCA (`AltMaxVar`). As one can see, the algorithm does not instantiate any large dense matrix during the procedure and thus is highly efficient in terms of memory. Also, the procedure does not destroy sparsity of the data, and thus the computational burden is light when the data is sparse – which is often the case in large-scale learning applications. Detailed complexity analysis will be presented in the next subsection.

Algorithm 1: `AltMaxVar`

input : $\{\boldsymbol{X}_i, \mu_i, \}_{i=1}^I$; $\gamma \in (0, 1]$; K; T; $(\{\boldsymbol{Q}_i^{(0)}\}_{i=1}^I, \boldsymbol{G}^{(0)})$.

$r \leftarrow 0$;
repeat
  $t \leftarrow 0$;
  $\boldsymbol{E}_i^{(t)} \leftarrow \boldsymbol{Q}_i^{(r)}$ for $i = 1, \ldots, I$;
  while $t < T$ *and convergence not reached* **do**
    for all i, update
    $$\boldsymbol{E}_i^{(t+1)} \leftarrow \mathtt{prox}_{\alpha_i g_i} \left(\boldsymbol{E}_i^{(t)} - \alpha_i \nabla f_{\boldsymbol{Q}_i} \left(\boldsymbol{E}_i^{(t)}; \boldsymbol{G}_i^{(r)} \right) \right)$$
    where $\nabla f_{\boldsymbol{Q}_i}(\boldsymbol{E}_i^{(t)}; \boldsymbol{G}_i^{(r)}) = -\boldsymbol{X}_i^T \boldsymbol{G}^{(r)} + \boldsymbol{X}_i^T \boldsymbol{X}_i \boldsymbol{E}_i^{(t)}$;
    $t \leftarrow t + 1$;
  end
  $\boldsymbol{Q}_i^{(r+1)} \leftarrow \boldsymbol{E}_i^{(t)}$;
  $\boldsymbol{P} \leftarrow \gamma \sum_{i=1}^I \boldsymbol{X}_i \boldsymbol{Q}_i^{(r+1)} / I + (1 - \gamma) \boldsymbol{G}^{(r)}$;
  $\boldsymbol{U}_P \boldsymbol{D}_P \boldsymbol{V}_P^T \leftarrow \mathrm{svd}(\boldsymbol{P}, 'econ')$;
  $\boldsymbol{G}^{(r+1)} \leftarrow \boldsymbol{U}_P \boldsymbol{V}_P^T$;
  $r \leftarrow r + 1$;
until *Some stopping criterion is reached*;

output: $\left\{\boldsymbol{Q}_i^{(r)}\right\}_{i=1}^I$, $\boldsymbol{G}^{(r)}$

B. Computational and Memory Complexities

The update rule in (6) inherits the good features from the proximal gradient (PG) method. First, there is no "heavy computation" if the views $\boldsymbol{X}_i$ for $i = 1, \ldots, I$ are sparse. Specifically, the major computation in the update rule of (6) is computing the partial gradient of the smooth part of the cost function, i.e., $\nabla_{\boldsymbol{Q}_i} f(\boldsymbol{Q}_i, \boldsymbol{G}_i^{(r)})$. To this end, $\boldsymbol{X}_i \boldsymbol{Q}_i$ should be calculated first, since if $\boldsymbol{X}_i$ is sparse, this matrix multiplication step has a complexity order of $\mathcal{O}(\mathrm{nnz}(\boldsymbol{X}_i) \cdot K)$ flops, where $\mathrm{nnz}(\cdot)$ counts the number of non-zeros. The next multiplication, i.e., $\boldsymbol{X}_i^T (\boldsymbol{X}_i \boldsymbol{Q}_i)$, has the same complexity order. Similarly, the operation of $\boldsymbol{X}_i^T \boldsymbol{G}$ has the same complexity. For solving the $\boldsymbol{G}$-subproblem, the major operation is the SVD of $\boldsymbol{P}$. This step is also not computationally heavy - what we

ask for is an economy-size SVD of a very thin matrix (of size $L \times K$, $L \gg K$). This has a complexity order of $\mathcal{O}(LK^2)$ flops [25], which is light.

In terms of memory, all the terms involved (i.e., $\boldsymbol{Q}_i$, $\boldsymbol{G}_i$, $\boldsymbol{X}_i\boldsymbol{Q}_i$, $\boldsymbol{X}_i^T\boldsymbol{X}_i\boldsymbol{Q}_i$ and $\boldsymbol{X}_i^T\boldsymbol{G}_i$) only require $\mathcal{O}(LK)$ memory or less, but the eigen-decomposition-based solution needs $\mathcal{O}(M_i^2)$ and $\mathcal{O}(L^2)$ memory to store $(\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}$ and $\boldsymbol{M}$, respectively. Note that K is usually very small compared to L and M_i and can be controlled by the designer.

IV. Convergence Properties

In this section, we study convergence properties of `AltMaxVar`. Note that the algorithm alternates between a (possibly) non-smooth subproblem and a manifold-constrained subproblem, and the subproblems may or may not be solved to optimality. Existing convergence analysis for exact and inexact block coordinate descent such as those in [28]–[31] can not be directly applied to analyze `AltMaxVar`, and thus the convergence properties are not obvious. For the purpose of discussion, we first define a critical point, or, a KKT point, of Problem (4). A KKT point $(\boldsymbol{G}^*, \boldsymbol{Q}^*)$ satisfies the following first-order optimality conditions:

$$\begin{cases} \boldsymbol{0} \in \nabla_{\boldsymbol{Q}}\ f(\boldsymbol{Q}^*, \boldsymbol{G}^*) + \partial_{\boldsymbol{Q}} g(\boldsymbol{Q}^*) \\ \boldsymbol{0} = \nabla_{\boldsymbol{G}}\ f(\boldsymbol{Q}^*, \boldsymbol{G}^*) + \boldsymbol{G}\boldsymbol{\Lambda}^*, \quad (\boldsymbol{G}^*)^T\boldsymbol{G}^* = \boldsymbol{I}, \end{cases} \tag{11}$$

where $\boldsymbol{\Lambda}$ is a Lagrangian multiplier associated with the constraint $\boldsymbol{G}^T\boldsymbol{G} = \boldsymbol{I}$, and

$$\nabla_{\boldsymbol{Q}}\ f(\boldsymbol{Q}, \boldsymbol{G}) = [(\nabla_{\boldsymbol{Q}_1} f(\boldsymbol{Q}, \boldsymbol{G}))^T, \ldots, (\nabla_{\boldsymbol{Q}_I} f(\boldsymbol{Q}, \boldsymbol{G}))^T]^T,$$
$$\partial_{\boldsymbol{Q}} g(\boldsymbol{Q}) = [(\partial_{\boldsymbol{Q}_1} g_1(\boldsymbol{Q}_1))^T, \ldots, (\partial_{\boldsymbol{Q}_I} g_I(\boldsymbol{Q}_I))^T]^T,$$

in which $\partial_{\boldsymbol{Q}_i} g_i(\boldsymbol{Q}_i)$ denotes a subgradient of the (possibly) non-smooth function $g_i(\boldsymbol{Q}_i)$. Using the above, we first show that

Proposition 1 *Assume that $\alpha_i \leq 1/L_i$ for all i, where $L_i = \lambda_{\max}(\boldsymbol{X}_i^T\boldsymbol{X}_i)$ is the largest eigenvalue of $\boldsymbol{X}_i^T\boldsymbol{X}_i$. Also assume that $g_i(\cdot)$ is a closed convex function, $T \geq 1$, and $\gamma \in (0, 1]$. Then, the following holds:*

(a) *The objective value of Problem* (2) *is non-increasing. In addition, every limit point of the solution sequence $\{\boldsymbol{G}^{(r)}, \boldsymbol{Q}^{(r)}\}_{r=0,1,\ldots}$ is a KKT point of Problem* (2).

(b) *If $\boldsymbol{X}_i$ and $\boldsymbol{Q}_i^{(0)}$ for $i = 1, \ldots, I$ are bounded and $\mathrm{rank}(\boldsymbol{X}_i) = M_i$, then, the whole solution sequence converges to the set $\mathcal{K}$ that consists of all the KKT points, i.e., $\lim_{r\to\infty}\ d^{(r)}(\mathcal{K}) \to 0$, where $d^{(r)}(\mathcal{K}) = \min_{\boldsymbol{Y}\in\mathcal{K}} \|(\boldsymbol{G}^{(r)}, \boldsymbol{Q}^{(r)}) - \boldsymbol{Y}\|_F$.*

Proposition 1 (a) characterizes the limit points of the solution sequence. According to Theorem 1, even only one proximal gradient step is performed in each iteration r, every limit point of the algorithm is a KKT point of Problem (4). As we demonstrated in the proof, `AltMaxVar` can be viewed as an algorithm that successively deals with local upper bounds of two subproblems, which has a similar flavor of *block successive upper bound minimization* (BSUM) [29] and the *block prox-linear* (BPL) framework [30], [31]. However, BSUM does not cover nonconvex constraints such as $\boldsymbol{G}^T\boldsymbol{G} = \boldsymbol{I}$, and BPL does not cover the $\gamma = 1$ case where the $\boldsymbol{G}$-subproblem is optimally solved. Hence, the convergence properties of BSUM and BPL do not imply Proposition 1. The (b) part of Proposition 1 establishes the convergence of the whole solution sequence – which is a much stronger result. The assumptions, on the other hand, are also more restrictive, where the views all have full column rank, which was not assumed in the (a) part.

It is also meaningful to estimate the number of iterations that is needed for the algorithm reaching a neighborhood of a KKT point. To this end, let us define the following potential function:

$$Z^{(r,r+1)} = \sum_{t=0}^{T-1} \left\| \tilde{\nabla}_{\boldsymbol{Q}} F\left(\boldsymbol{Q}^{(r,t)}, \boldsymbol{G}^{(r)}\right) \right\|_F^2 + \left\| \nabla_{\boldsymbol{G}}\ f\left(\boldsymbol{Q}^{(r+1)}, \boldsymbol{G}^{(r)}\right) + \boldsymbol{\Lambda}^{(r+1)}\boldsymbol{G}^{(r+1)} \right\|_F^2,$$

where $\tilde{\nabla}_{\boldsymbol{Q}} F(\boldsymbol{Q}^{(r,t)}, \boldsymbol{G}^{(r)}) = [(\tilde{\nabla}_{\boldsymbol{Q}_1} F(\boldsymbol{Q}^{(r,t)}, \boldsymbol{G}^{(r)}))^T, \ldots, (\tilde{\nabla}_{\boldsymbol{Q}_I} F(\boldsymbol{Q}^{(r,t)}, \boldsymbol{G}^{(r)}))^T]^T$ is the proximal gradient w.r.t. $\boldsymbol{Q}$ at $(\boldsymbol{Q}^{(r,t)}, \boldsymbol{G}^{(r)})$, in which

$$\tilde{\nabla}_{\boldsymbol{Q}_i} F(\boldsymbol{Q}^{(r,t)}, \boldsymbol{G}^{(r)}) = \frac{1}{\alpha_i}\left(\boldsymbol{Q}_i^{(r+1,t)} - \mathtt{prox}_{\alpha_i g_i}(\boldsymbol{Q}_i^{(r,t)} - \alpha_i \nabla_{\boldsymbol{Q}_i} f(\boldsymbol{Q}_i^{(r,t)}, \boldsymbol{G}^{(r)}))\right).$$

Note that the proximal gradient update w.r.t. $\boldsymbol{Q}_i$ can be written as $\boldsymbol{Q}_i^+ = \hat{\boldsymbol{Q}}_i - \alpha_i \tilde{\nabla}_{\boldsymbol{Q}_i} F(\hat{\boldsymbol{Q}}_i, \boldsymbol{G}_i)$ [27]. One can see that $Z^{(r,r+1)}$ is a value that is determined by two consecutive outer iterates of the algorithm. $Z^{(r,r+1)}$ has the following property:

Lemma 1 *$Z^{(r,r+1)} \to 0$ implies that $(\boldsymbol{Q}^{(r)}, \boldsymbol{G}^{(r)})$ approaches a KKT point.*

The proof of Lemma 1 is in Appendix B. As a result, we can use the value of $Z^{(r,r+1)}$ to measure how close is the current iterate to a KKT point, thereby estimating the iteration complexity. Following this rationale, we show that

Theorem 1 *Assume that $\alpha_i < 1/L_i$, $0 < \gamma < 1$ and $T \geq 1$. Let $\delta > 0$ and J be the number of iterations needed for that $Z^{(r,r+1)} \leq \delta$ holds for the first time. Then, there exists a constant v such that $\delta \leq v/J-1$; that is, the algorithm converges to a KKT point at least sublinearly.*

The proof of Theorem 1 is relegated to Appendix C. By Theorem 1, `AltMaxVar` reduces the optimality gap (measured by the Z-function) between the current iterate and a KKT point to δ to $\mathcal{O}(1/r)$ after r iterations. One subtle point that is worth mentioning is that the analysis in Theorem 1 holds when $\gamma < 1$ – it corresponds to the case where the $\boldsymbol{G}$-subproblem is *not* optimally solved. This reflects some interesting facts in alternating optimization – when the subproblems are handled in a more conservative way and using a controlled step size, convergence may be guaranteed under milder conditions. On the other hand, more conservative step sizes may result in slower convergence. Hence, choosing an optimization strategy usually poses a trade-off between practical considerations such as speed and theoretical guarantees.

Proposition 1 and Theorem 1 characterize convergence properties of `AltMaxVar` with a general regularization term $g_i(\cdot)$. It is also interesting to consider the special cases where $g_i(\cdot) = 0$ and $g_i(\cdot) = \mu_i\|\cdot\|_F^2$ – which correspond to the original MAX-VAR formulation and its "diagonal loaded" version [10]. These two cases are *optimally solvable* via eigen-decomposition [1]. It is natural for us to think if these two special cases are also "easier" to solve using `AltMaxVar`. The answer is affirmative – for these two cases, the convergence rate can be shown to be *linear* instead on sublinear as in the general case, and solution sequence converges to a *global optimal solution* under some conditions. To explain, let us denote $\boldsymbol{U}_1 = \boldsymbol{U}_M(:, 1:K)$ and $\boldsymbol{U}_2 = \boldsymbol{U}_M(:, K+1:L)$ as the K principal eigenvectors of $\boldsymbol{M}$ and the eigenvectors spanning its orthogonal complement, respectively. Recall that our ultimate goal is to find $\boldsymbol{G}$ that is a basis of the range space of $\boldsymbol{U}_1$, denoted by $\mathcal{R}(\boldsymbol{U}_1) = \mathcal{R}_K(\boldsymbol{M})$. Therefore, the speed of convergence can be measured by the speed of the distance to $\mathcal{R}_K(\boldsymbol{M})$ approaching zero, where we adopt the definition of subspace distance in [25], i.e., $\text{dist}\left(\mathcal{R}(\boldsymbol{G}^{(r)}), \mathcal{R}_K(\boldsymbol{M})\right) = \|\boldsymbol{U}_2^T\boldsymbol{G}^{(r)}\|_2$. We show that

Theorem 2 *Denote the eigenvalues of* $\boldsymbol{M} \in \mathbb{R}^{L\times L}$ *by* $\lambda_1, \ldots, \lambda_L$ *in descending order. Consider* $g_i(\cdot) = 0$ *and* $g_i(\cdot) = \mu_i\|\cdot\|_F^2$ *and let* $\gamma = 1$. *Assume that* $\text{rank}(\boldsymbol{X}_i) = M_i$, $\lambda_K > \lambda_{K+1}$, *and* $\mathcal{R}(\boldsymbol{G}^{(0)})$ *is not orthogonal to any component in* $\mathcal{R}_K(\boldsymbol{M})$, *i.e.,* $\cos(\theta) = \min_{\boldsymbol{u}\in\mathcal{R}_K(\boldsymbol{M}), \boldsymbol{v}\in\mathcal{R}(\boldsymbol{G}^{(0)})} \frac{|\boldsymbol{u}^T\boldsymbol{v}|}{(\|\boldsymbol{u}\|_2\|\boldsymbol{v}\|_2)} > 0$. *In addition, assume that each subproblem in* (5) *is solved to accuracy* ϵ, *i.e.,* $\|\boldsymbol{Q}_i^{(r+1)} - \tilde{\boldsymbol{Q}}_i^{(r+1)}\|_2 \le \epsilon$, *where* $\tilde{\boldsymbol{Q}}_i^{(r+1)} = (\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i\boldsymbol{G}^{(r)}$. *Then, after* r *iterations, we have*

$$\text{dist}\left(\mathcal{R}(\boldsymbol{G}^{(r)}), \mathcal{R}_K(\boldsymbol{M})\right) \le \tan(\theta)\left(\lambda_{K+1}/\lambda_K\right)^r + C,$$

where $C = \mathcal{O}\left(\sum_{i=1}^I \lambda_{\max}(\boldsymbol{X}_i)\epsilon\right)$ *is a constant .*

Theorem 2 makes much sense for inexact alternating optimization: in practice, solving the subproblem in (5) is not an easy task, and one may want to stop early (e.g., using a small T). Theorem 2 ensures that if a T suffices for the $\boldsymbol{Q}$-subproblem to obtain a good enough approximation of the solution of Problem (5), the algorithm converges *linearly* to the desired solution up to some accuracy loss. In our simulations, we observe that even using $T = 1$ already gives very satisfactory results (as will be shown in the next section), which leads to computationally very cheap updates.

V. Numerical Results

In this section, we use synthetic data and real experiments to showcase the effectiveness of the proposed algorithm. The experiments are carried out using MATLAB on a Linux server with 128GB RAM and 2.0GHz CPU cores.

A. Sanity Check: Small-Size Problems

In this subsection, we first use small-size problem instances to verify the convergence properties that were discussed in the last section and also showcase the effectiveness of the algorithm.

[1]For the diagonal loading form, the solution is to change that in (3) to the leading eigenvectors of $\tilde{\boldsymbol{M}} = \sum_{i=1}^I \boldsymbol{X}_i(\boldsymbol{X}_i^T\boldsymbol{X}_i + \mu_i\boldsymbol{I})^{-1}\boldsymbol{X}_i^T$ [10].

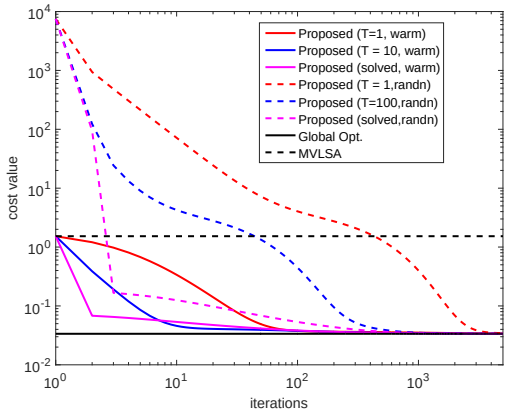

Fig. 2. Convergence curves. Small size

1) Original Max-Var GCCA: We generate the synthetic data in the following way: First, we let $\boldsymbol{Z} \in \mathbb{R}^{L\times N}$ be a common latent factor of different views, where the entries of $\boldsymbol{Z}$ are drawn from the zero-mean i.i.d. Gaussian distribution and $L \ge N$. Then, a 'mixing matrix' $\boldsymbol{A}_i \in \mathbb{R}^{N\times M_i}$ is multiplied to $\boldsymbol{Z}$, resulting in $\boldsymbol{Y}_i = \boldsymbol{Z}\boldsymbol{A}_i$. We let $M_1 = \ldots = M_I = M$ in this section. Finally, we add noise so that $\boldsymbol{X}_i = \boldsymbol{Y}_i + \sigma\boldsymbol{N}_i$. Here, $\boldsymbol{A}_i$ and $\boldsymbol{N}_i$ are generated in the same way as $\boldsymbol{Z}$. We first apply the algorithm with the regularization term $g_i(\cdot) = \mu_i\|\cdot\|_F^2$ and let $\mu_i = 0.1$. Since L and M are small in this subsection, we employ the optimal solution that is based on eigen-decomposition as a baseline, since this case is solvable. The multiview latent semantic analysis (`MVLSA`) algorithm that was proposed in [10] as baselines. Recall that `MVLSA` approximates the views using several leading singular values and vectors.

In Fig. 2, we let $(L, M, N, I) = (500, 25, 20, 3)$. We set $\sigma = 0.1$ in this case, let $P = 8$ and $\gamma = 1$ for `MVLSA` and `AltMaxVar`, respectively, and ask for $K = 5$ canonical components. The results are averaged over 50 random trials, where $\boldsymbol{Z}$, $\{\boldsymbol{A}_i\}$, $\{\boldsymbol{N}_i\}$ are randomly generated in each trial. We test the proposed algorithm under different settings: We let $T = 1$, $T = 10$, and the gradient descent run until the inner loop converges (denoted as 'solved' in the figures). We also initialize the algorithm with random initializations (denoted as 'randn') and warm starts (denoted as 'warm') – i.e., using the solutions of `MVLSA` as starting points. Some observations from Fig. 2 are in order. First, the proposed algorithm using various T's including $T = 1$ and random initialization can reach the global optimum, which supports the analysis in Theorem 2. Second, by increasing T, the overall cost value decreases faster in terms of number of outer iterations – using $T = 10$ already gives very good speed of decreasing the cost value. Third, `MVLSA` cannot attain the global optimum, as expected. However, it provides good initialization: Using the warm start, the cost value comes close to the optimal value within 100 iterations in this case, even when $T = 1$ is employed. In fact, the combination of `MVLSA`-based initialization and using $T = 1$ offers the most computationally efficient way of

implementating the proposed algorithm – especially for the large-scale case. In the remaining part of this section, we will employ MVLSA as the initialization of AltMaxVar and perform the $\boldsymbol{Q}$-subproblem with $T = 1$.

2) Feature-Selective Max-Var GCCA: To test the proposed algorithm with non-smooth regularizers, we generate cases where outlying features are present in all views. Specifically, we let $\boldsymbol{X}_i = [\boldsymbol{Z}\boldsymbol{A}_i, \boldsymbol{O}_i] + \sigma\boldsymbol{N}_i$, where $\boldsymbol{O}_i \in \mathbb{R}^{L\times N_o}$ denotes the irrelevant outlying features and the elements of $\boldsymbol{O}_i$ follows the i.i.d. zero-mean unit-variance Gaussian distribution. We wish the algorithm to perform MAX-VAR GCCA of the views while discounting $\boldsymbol{O}_i$ at the same time. To deal with outlying features, we employ the regularizer $g_i(\cdot) = \mu_i\|\cdot\|_{2,1}$. Under such cases, the optimal solution to Problem (4) is unknown. Nevertheless, we evaluate the performance by observing

$$\text{metric}_1 = {}^1\!/_I \sum_{i=1}^{I} \|\boldsymbol{X}_i(:,\mathcal{S}_i^c)\hat{\boldsymbol{Q}}_i(\mathcal{S}_i^c,:) - \hat{\boldsymbol{G}}\|_F^2,$$

$$\text{metric}_2 = {}^1\!/_I \sum_{i=1}^{I} \|\boldsymbol{X}_i(:,\mathcal{S}_i)\hat{\boldsymbol{Q}}_i(\mathcal{S}_i,:)\|_F^2,$$

where $\mathcal{S}_i^c$ and $\mathcal{S}_i$ denote the index sets of "clean" and outlying features of view i, respectively – i.e., $\boldsymbol{X}_i(:,\mathcal{S}_i^c) = \boldsymbol{Z}_i\boldsymbol{A}_i$ and $\boldsymbol{X}_i(:,\mathcal{S}_i) = \boldsymbol{O}_i$ if noise is absent. metric_1 measures the performance of matching $\hat{\boldsymbol{G}}$ with the relevant part of the views, while metric_2 measures the performance of suppressing the irrelevant part. We wish an algorithm to give low values of metric_1 and metric_2 simultaneously.

Table I presents the results of a small-size case which are averaged from 50 random trials, where $(L, M, N, I) = (150, 60, 60, 3)$ and $|\mathcal{S}| = \{61, \ldots, 120\}$; i.e., 60 out of 120 features of $\boldsymbol{X}_i \in \mathbb{R}^{150\times 120}$ are outlying features. The average power of the outlying features is set to be the same as that of the clean features, i.e., $\|\boldsymbol{Q}_i\|_F^2/\text{nnz}(\boldsymbol{O}_i) = \|\boldsymbol{Z}\boldsymbol{A}_i\|_F^2/\text{nnz}(\boldsymbol{Z}\boldsymbol{A}_i)$ so that the outlying features would have an impact on the CCA result if not discounted. We ask for $K = 10$ canonical components. For MVLSA, we let the rank-truncation parameter to be $P = 50$; for AltMaxVar, we set $\gamma = 0.99$. One can see that the eigen-decomposition based algorithm gives similar high values of both the evaluation metrics since it treats $\boldsymbol{X}_i(:,\mathcal{S}_i^c)$ and $\boldsymbol{X}_i(:,\mathcal{S}_i)$ equally. It is interesting to see that MVLSA suppresses the irrelevant features to some extent – although it does not explicitly consider outlying features, our understanding is that the PCA pre-processing on the views can somewhat suppress the outliers. Nevertheless, MVLSA does not fit the relevant part of the views well. The proposed algorithm gives the lowest values of both metrics. In particular, when $\mu = 1$, the irrelevant part is almost suppressed completely. Another observation is that using $\mu = 0.5$, the obtained score of metric_1 is slightly lower than that under $\mu = 1$, which makes sense since the algorithm pays more attention to feature selection using a larger μ. An illustrative example using a random trial can be seen in Fig. 3. From there, one can see that the proposed algorithm gives $\boldsymbol{Q}_i$'s with almost zero rows over $\mathcal{S}$, thereby performing feature selection.

TABLE I
PERFORMANCE OF THE ALGORITHMS WHEN IRRELEVANT FEATURES ARE PRESENT. $(L, M, N) = (150, 60, 60)$; $|\mathcal{S}| = 60$; $\boldsymbol{X}_i \in \mathbb{R}^{150\times 120}$; $\sigma = 1$.

Algorithm	metric_1	metric_2
eigen-decomp	9.547	9.547
MVLSA	15.506	1.456
proposed ($\mu = .5$)	**0.486**	9.689×10^{-3}
proposed ($\mu = 1$)	1.074	$\mathbf{8.395 \times 10^{-4}}$

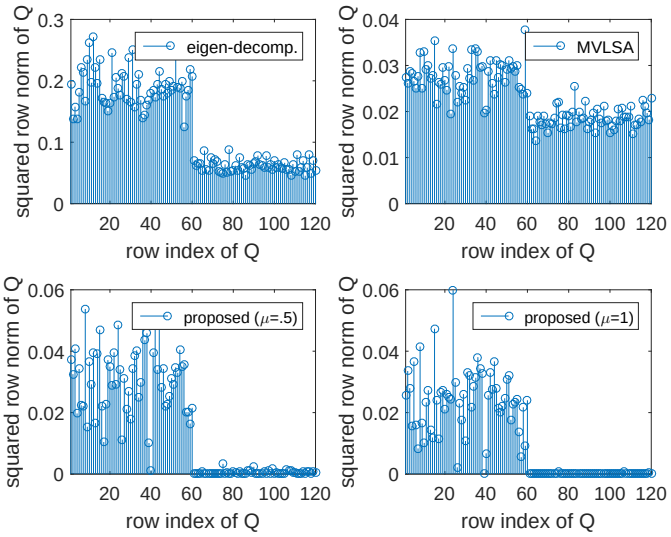

Fig. 3. Average row-norms of $\boldsymbol{Q}_i$ (i.e., $(1/I)\sum_{i=1}^{I}\|\boldsymbol{Q}_i(m,:)\|_2^2$) for all m given by the algorithms.

B. Scalability Test: Large-Size Problems

1) Original Max-Var GCCA: We first test the case where no outlying features are involved and the regularizer $g_i(\cdot) = \mu_i\|\cdot\|_F^2$ is employed. The views $\boldsymbol{X}_i = \boldsymbol{Z}\boldsymbol{A}_i + \sigma\boldsymbol{N}_i$ are generated following a similar way as in the last subsection, but $\boldsymbol{Z}$, $\boldsymbol{A}_i$ and $\boldsymbol{N}_i$ are sparse so that $\boldsymbol{X}_i$ are sparse with a density level ρ_i that is definied as

$$\rho_i = \frac{\text{nnz}(\boldsymbol{X}_i)}{LM}.$$

In the simulations, we let $\rho = \rho_1 = \ldots = \rho_I$. and the results are obtained via averaging 10 random trials.

In Fig. 4, we show the runtime performance of the algorithms under various sizes of the views. The density of the views are controlled so that $\rho \leq 10^{-3}$. We let $M = L \times 0.8$, $M = N$ and change M from $5,000$ to $50,000$. The eigen-decomposition based global optimal solution and MVLSA (using $P = 100$) are used baselines, and the proposed AltMaxVar is initialized by MVLSA and we let $T = 1$. The diagonal loading parameter is set to be $\mu_i = 0.1$ for all algorithms. One can see that the eigen-decomposition based algorithm does not scale well since the matrix $(\boldsymbol{X}_i^T\boldsymbol{X}_i + \mu_i\boldsymbol{I})^{-1}$ is dense. In particular, the algorithm exhausts the memory quota (32GB RAM) when $M = 30,000$. MVLSA and the proposed algorithm both scale very well from $M = 5,000$ to $M = 50,000$: When $M = 20,000$, the global optimal solution uses almost 80 minutes to finish the computations, while MVLSA and AltMaxVar both use less than 2 minutes. Note

that the runtime of the proposed algorithm already includes the runtime of the initialization time by MVLSA, and thus the runtime curve of AltMaxVar is slightly higher than that of MVLSA in Fig. 4. The cost values given by the algorithms can be seen in Table II. The eigen-decomposition based method gives the lowest cost values when applicable as it is an optimal solution. In terms of accuracy, the proposed algorithm gives favorable cost values that are close to the optimal value – note that only one iteration of the $\boldsymbol{Q}$-subproblem is implemented. However, MVLSA is not so promising in terms of cost value.

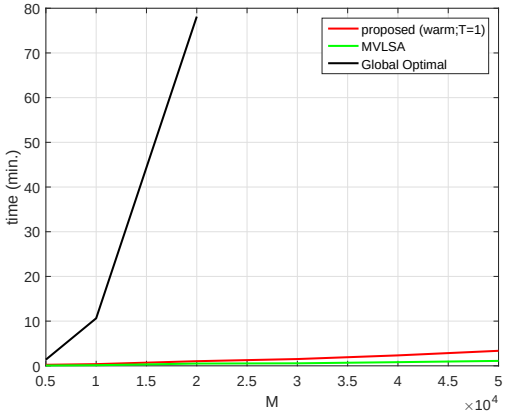

Fig. 4. Runtime of the algorithms under various problem sizes. $L = M/0.8$, $\rho \leq 10^{-3}$, $\sigma = 0.1$.

TABLE II
COST VALUES OF THE ALGORITHMS UNDER DIFFERENT PROBLEM SIZES. $L = M/0.8$, $\rho = 10^{-3}$, $\sigma = 0.1$. † MEANS "OUT OF MEMORY".

Algorithm	M					
	5,000	10,000	20,000	30,000	40,000	50,000
Global Opt.	**0.053**	**0.034**	**0.020**	†	†	†
MVLSA	2.103	3.749	5.135	6.018	6.579	7.044
proposed	0.088	0.062	0.047	0.043	0.041	0.039

2) Feature-Selective Max-Var GCCA: Table III presents the simulation results of a large-scale case in the presence of outlying features. Here, we fix $L = 100,000$ and $M = 80,000$ and change the density level ρ. We add $|\mathcal{S}_i| = 30,000$ outlying features to each view and every outlying feature is a random sparse vector whose non-zero elements follow the zero-mean i.i.d. unit-variance Gaussian distribution. We also scale the outlying features as before so that the energy of the clean and outlying features are comparable. The other settings follow those in the last simulation. One can see from Table III that the proposed algorithm with $\mu_i = 0.5$ gives the most balanced result – both evaluation metrics are with fairly low levels. Using $\mu_i = 1$ suppresses the $\boldsymbol{Q}_i(\mathcal{S},:)$ quite well, but using a larger μ_i also brings some sacrifice to the fitting metric. In terms of runtime, one can see that the proposed algorithm operates within the same order of magnitude of time as MVLSA does. Note that the proposed algorithm works with the intact views with the size $L \times M$ but MVLSA works with heavily truncated data. Therefore, such runtime performance of AltMaxVar is very satisfactory.

Similar results can be seen in Table IV, where we let $\rho = 10^{-4}$ and change I from 3 to 8. One can see that increasing the number of views does not increase the runtime of the proposed algorithm. The reason is that the updates of different $\boldsymbol{Q}_i$'s can be easily parallelized. One can easily implement the parallel computations using the parfor function of Matlab – which is what we do in this simulation.

TABLE III
EVALUATION OF THE ALGORITHM UNDER DIFFERENT DATA DENSITIES IN THE PRESENCE OF OUTLYING FEATURES. $L = 100,000$, $M = 80,000$, $|\mathcal{S}| = 30,000$, $\sigma = 1$, $I = 3$.

Algorithm	measure	ρ (density of views)			
		10^{-5}	5×10^{-4}	10^{-4}	10^{-3}
MVLSA	metric1	16.843	13.877	17.159	16.912
	metric2	**0.003**	**0.010**	0.009	0.003
	time (min)	**0.913**	**1.019**	**1.252**	**3.983**
proposed ($\mu = .05$)	metric1	**0.478**	**0.610**	**0.565**	**0.775**
	metric2	0.018	0.134	0.034	0.003
	time (min)	3.798	5.425	5.765	24.182
proposed ($\mu = .1$)	metric1	0.942	1.054	0.941	1.265
	metric2	0.006	0.054	**0.004**	**0.000**
	time (min)	2.182	3.791	4.510	16.378
proposed ($\mu = .5$)	metric1	1.592	1.497	1.306	1.538
	metric2	**0.003**	0.021	**0.000**	**0.000**
	time (min)	1.735	2.714	3.723	13.447

TABLE IV
EVALUATION OF THE ALGORITHM UNDER DIFFERENT DATA DENSITIES IN THE PRESENCE OF OUTLYING FEATURES. $L = 100,000$, $M = 80,000$, $|\mathcal{S}| = 30,000$, $\sigma = 1$, $\rho = 5 \times 10^{-5}$.

Algorithm	measure	I (no. of views)					
		3	4	5	6	7	8
MVLSA	metric1	15.813	15.715	14.667	16.904	17.838	17.691
	metric2	**0.008**	0.009	0.009	0.009	0.007	0.009
	time (min)	**1.087**	**0.975**	**0.960**	**0.958**	**0.989**	**1.026**
proposed ($\mu = .05$)	metric1	**0.731**	**0.590**	**0.670**	**0.611**	**0.517**	**0.628**
	metric2	0.172	0.078	0.100	0.101	0.065	0.098
	time (min)	5.870	6.064	5.762	5.070	5.895	5.776
proposed ($\mu = .1$)	metric1	1.070	1.057	1.110	1.026	1.042	1.112
	metric2	0.055	0.019	0.018	0.024	0.023	0.023
	time (min)	3.240	2.974	3.313	3.210	3.083	3.529
proposed ($\mu = .5$)	metric1	1.461	1.482	1.578	1.443	1.472	1.561
	metric2	0.018	**0.002**	**0.003**	**0.001**	**0.006**	**0.007**
	time (min)	2.700	2.441	2.528	2.569	2.431	2.567

C. Real-Data Validation

We test the algorithms on a large-scale multilingual dataset. The views are extracted from a large word co-occurrence matrix, which is available at https://sites.google.com/a/umn.edu/huang663/research. The original data contains words of three languages, namely, English, Spanish, and French, and all the words are defined by the co-occurences pointwise mutual information (PMI) with other words. We use the English words to form our first view, $\boldsymbol{X}_1$, which contains $L = 183,034$ words and each word is defined by $M_i = 100,000$ features (co-occurrences). Note that $\boldsymbol{X}_1$ is sparse – only 1.21% of its entries are non-zeros. Using a dictionary, we pick out the translations of the English words contained in $\boldsymbol{X}_1$ in Spanish

and French to form $\boldsymbol{X}_2$ and $\boldsymbol{X}_3$, respectively. Note that many English words do not have a corresponding word in Spanish (or French). In such cases, we simply let $\boldsymbol{X}_i(\ell,:)=\boldsymbol{0}$ for $i=2$ (or $i=3$), resulting in sparser $\boldsymbol{X}_2$ and $\boldsymbol{X}_3$. Our objective is to find $\boldsymbol{G}$ whose rows are the low-dimensional embeddings of the English words (cf. the motivating example in Fig. 1).

To evaluate the output, we use the evaluation tool provided at wordvectors.org [32], which runs several word embedding tasks to evaluate a set of given embeddings. Simply speaking, the tasks compare the algorithm-learned embeddings with the judgment of humans and yield high scores if the embeddings are consistent with the humans. The scores are between zero and one, and a score equal to one means a perfect alignment between the learned result and human judgment. We use the result of MVLSA with $P=640$ as benchmark. The result of applying SVD to $\boldsymbol{X}_1$ without considering different languages is also presented. We apply the proposed algorithm warm started by MVLSA and set $T=1$. We run two versions of our algorithm. The first one uses $g_i(\cdot)=\|\cdot\|_F^2$ for $i=1,2,3$. The second one uses $g_i(\cdot)=0.05\|\cdot\|_{2,1}$ for $i=2,3$. The reason for adding ℓ_2/ℓ_1 mixed norm regularization to the French and Spanish views is towfold: First, the ℓ_2/ℓ_1 norm promotes row sparsity of $\boldsymbol{Q}_i$ and thus performs feature selection on $\boldsymbol{X}_2$ and $\boldsymbol{X}_3$ – this physically means that we aim at selecting the most useful features from the other languages to help enhance English word embeddings. Second, the languages are effectively 'fat matrices' and thus need to use a column-selective regularizer can help improve the conditioning.

Tables V and VI show the word embedding results using $K=50$ and $K=100$, respectively. We see that using the information from multiviews does help in improving the word embeddings: For $K=50$ and $K=100$, the multiview approaches perform better relative to SVD in 11 and 8 tasks out of 12 tasks. In addition, the proposed algorithm with the regularizer $g_i(\cdot)=\|\cdot\|_F^2$ gives similar or slightly better in average on both experiments compared to MVLSA. The proposed algorithm with the feature-selective regularizer ($g_i(\cdot)=\mu_i\|\cdot\|_{2,1}$) gives the best evaluation results on both experiments – this suggests that for large-scale multiview analysis, feature selection is much meaningful.

TABLE V
EVALUATION ON 12 WORD EMBEDDING TASKS; $K=50$.

Task	Algorithm ($K=50$)			
	SVD	MVLSA	Proposed ($\|\cdot\|_F^2$)	Proposed ($\|\cdot\|_{2,1}$)
EN-WS-353-SIM	0.63	**0.69**	0.67	0.68
EN-MC-30	0.56	0.63	0.63	**0.64**
EN-MTurk-771	0.54	0.58	0.59	**0.60**
EN-MEN-TR-3k	0.67	0.66	0.67	**0.68**
EN-RG-65	0.51	0.53	0.55	**0.58**
EN-MTurk-287	**0.65**	0.64	**0.65**	0.64
EN-WS-353-REL	0.50	0.51	0.53	**0.55**
EN-VERB-143	0.21	**0.22**	0.21	0.21
EN-YP-130	0.36	0.39	0.38	**0.41**
EN-SIMLEX-999	0.31	**0.42**	0.41	0.39
EN-RW-STANFORD	0.39	**0.43**	**0.43**	**0.43**
EN-WS-353-ALL	0.56	0.59	0.59	**0.60**
Average	0.49	0.52	0.53	**0.54**
Median	0.53	0.56	0.57	**0.59**

TABLE VI
EVALUATION ON 12 WORD EMBEDDING TASKS; $K=100$.

Task	Algorithm ($K=100$)			
	SVD	MVLSA	Proposed ($\|\cdot\|_F^2$)	Proposed ($\|\cdot\|_{2,1}$)
EN-WS-353-SIM	0.68	**0.72**	0.71	**0.72**
EN-MC-30	0.73	0.68	0.72	**0.74**
EN-MTurk-771	0.59	0.60	0.60	**0.61**
EN-MEN-TR-3k	**0.72**	0.70	0.70	0.71
EN-RG-65	**0.68**	0.63	0.64	**0.68**
EN-MTurk-287	0.61	**0.66**	0.65	0.64
EN-WS-353-REL	**0.57**	0.54	0.55	0.56
EN-VERB-143	0.19	0.28	0.27	**0.29**
EN-YP-130	0.42	0.41	0.41	**0.45**
EN-SIMLEX-999	0.34	**0.42**	0.41	0.41
EN-RW-STANFORD	0.44	**0.46**	0.45	**0.46**
EN-WS-353-ALL	**0.62**	**0.62**	**0.62**	**0.62**
Average	0.55	0.56	0.56	**0.58**
Median	0.60	0.61	0.61	**0.62**

VI. CONCLUSION

In this work, we revisited the MAX-VAR GCCA problem with an eye towards scenarios involving large-scale and sparse data. The proposed approach is memory-efficient and has light per-iteration computational complexity if the views are sparse, and is thus suitable for dealing with big data. The algorithm is also flexible for incorporating different structure-promoting regularizers on the canonical components such as the feature-selective regularizations. A thorough convergence analysis was presented, showing that the proposed algorithmic framework guarantees a KKT point to be obtained with a sublinear rate in general cases with a variety of structure-promoting regularizers. We also showed that the algorithm guarantees approaching a global optimal solution with a linear convergence rate if the original MAX-VAR problem is considered. Simulations and careful experiments with large-scale multi-lingual data showed that the performance of the proposed algorithm is promising in dealing with large and sparse multiview data.

APPENDIX A
PROOF OF PROPOSITION 1

Recall that we have defined $\boldsymbol{Q}=[\boldsymbol{Q}_1^T,\ldots,\boldsymbol{Q}_I^T]^T$ as a collection of $\boldsymbol{Q}_i$'s and we define $F(\boldsymbol{G},\boldsymbol{Q})=\sum_{i=1}^I\frac{1}{2}\|\boldsymbol{X}_i\boldsymbol{Q}_i-\boldsymbol{G}\|_F^2+\sum_{i=1}^I g_i(\boldsymbol{Q}_i)$. Since the algorithm is essentially a two-block alternating optimization (i.e., $\boldsymbol{Q}_i$ for all i are updated simultaneously), the above notation suffices to describe the updates. We also define

$$u_Q\left(\boldsymbol{Q};\hat{\boldsymbol{G}},\hat{\boldsymbol{Q}}\right)=f(\hat{\boldsymbol{G}},\hat{\boldsymbol{Q}})+\left\langle\nabla_{\boldsymbol{Q}}f(\hat{\boldsymbol{Q}},\hat{\boldsymbol{G}}),\boldsymbol{Q}-\hat{\boldsymbol{Q}}\right\rangle+\sum_{i=1}^I\frac{1}{2\alpha_i}\|\boldsymbol{Q}_i-\hat{\boldsymbol{Q}}\|_F^2+\sum_{i=1}^I g_i(\boldsymbol{Q}_i);$$

i.e., $u_Q\left(\boldsymbol{Q};\hat{\boldsymbol{G}},\hat{\boldsymbol{Q}}\right)$ is an approximation of $F(\boldsymbol{G},\boldsymbol{Q})$ locally at the point $(\hat{\boldsymbol{G}},\hat{\boldsymbol{Q}})$. We further define $\tilde{u}_Q\left(\boldsymbol{Q};\hat{\boldsymbol{G}},\hat{\boldsymbol{Q}}\right)=u_Q\left(\boldsymbol{Q};\hat{\boldsymbol{G}},\hat{\boldsymbol{Q}}\right)-\sum_{i=1}^I g_i(\boldsymbol{Q}_i)$; i.e., $\tilde{u}_Q\left(\boldsymbol{Q};\hat{\boldsymbol{G}},\hat{\boldsymbol{Q}}\right)$ is an approximation of the continuously differentiable part $f(\boldsymbol{G},\boldsymbol{Q})$

locally at the point $(\hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}})$. One can see that,

$$\nabla_{\boldsymbol{Q}_i} f\left(\hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right) = \nabla_{\boldsymbol{Q}} \tilde{u}\left(\hat{\boldsymbol{Q}}; \hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right), \tag{12a}$$

$$\nabla_{\boldsymbol{Q}} f\left(\hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right) + \partial_{\boldsymbol{Q}} g(\hat{\boldsymbol{Q}}) = \nabla_{\boldsymbol{Q}} \tilde{u}\left(\hat{\boldsymbol{Q}}; \hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right) + \partial_{\boldsymbol{Q}} g(\hat{\boldsymbol{Q}}). \tag{12b}$$

Since $\nabla_{\boldsymbol{Q}_i} f(\boldsymbol{G}, \boldsymbol{Q})$ is L_i-Lipschitz continuous w.r.t. $\boldsymbol{Q}_i$ and $\alpha_i \leq 1/L_i$ for all i, we have the following holds:

$$u_Q\left(\boldsymbol{Q}; \hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right) \geq F\left(\hat{\boldsymbol{G}}, \boldsymbol{Q}\right), \ \forall \ \boldsymbol{Q}, \tag{13}$$

where the equality holds if and only if $\boldsymbol{Q}_i = \hat{\boldsymbol{Q}}_i$ for all i, i.e.,

$$u_Q\left(\hat{\boldsymbol{Q}}; \hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right) = F\left(\hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right). \tag{14}$$

Now, let us denote by $\boldsymbol{Q}^{(r,t)}$ (where $0 \leq t \leq T-1$) the solution of $\boldsymbol{Q}$ after t PG updates when $\boldsymbol{G}^{(r)}$ is fixed, where r is the iteration index of the outer loop. With the above notation, we have $\boldsymbol{Q}^{(r,0)} = \boldsymbol{Q}^{(r)}$ and $\boldsymbol{Q}^{(r,T)} = \boldsymbol{Q}^{(r+1)}$. Also, it can be seen that the update of $\boldsymbol{Q}_i$ can be written as [33]:

$$\boldsymbol{Q}_i^{(r,t+1)} = \arg\min_{\boldsymbol{Q}_i} \ u_Q\left(\boldsymbol{Q}; \boldsymbol{G}^{(r)}, \boldsymbol{Q}^{(r,t)}\right). \tag{15}$$

Similarly, we define

$$\begin{aligned} u_G\left(\boldsymbol{G}; \hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right) =& f(\hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}) + \left\langle \nabla_{\boldsymbol{G}} f(\hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}), \boldsymbol{G} - \hat{\boldsymbol{G}} \right\rangle \\ &+ \frac{I}{2\gamma} \left\| \boldsymbol{G} - \hat{\boldsymbol{G}} \right\|_F^2 + \sum_{i=1}^{I} g_i(\hat{\boldsymbol{Q}}_i), \end{aligned}$$

where the last term is a constant if $\boldsymbol{Q}$ is fixed. We also have the following holds:

$$\nabla_{\boldsymbol{G}} f\left(\hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right) = \nabla_{\boldsymbol{G}} u\left(\hat{\boldsymbol{G}}; \hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right) = I \cdot \hat{\boldsymbol{G}} - \sum_{i=1}^{I} \boldsymbol{X}_i \boldsymbol{Q}_i. \tag{16}$$

The update rule of $\boldsymbol{G}$ in Algorithm 1 can be re-expressed as follows:

$$\begin{aligned} &\boldsymbol{G} \in \arg\min_{\boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}} \ \left\| \boldsymbol{G} - \left((1-\gamma)\hat{\boldsymbol{G}} + \gamma \sum_{i=1}^{I} \boldsymbol{X}_i \boldsymbol{Q}_i / I \right) \right\|_F^2 \\ \Leftrightarrow\ &\boldsymbol{G} \in \arg\min_{\boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}} \ \left\| \boldsymbol{G} - \left(\hat{\boldsymbol{G}} - (\gamma/I) \nabla_{\boldsymbol{G}} f\left(\hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right) \right) \right\|_F^2 \\ \Leftrightarrow\ &\boldsymbol{G} \in \arg\min_{\boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}} \ u_G\left(\boldsymbol{G}; \hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right) \end{aligned}$$

Since $\nabla_{\boldsymbol{G}} f(\boldsymbol{G}, \boldsymbol{Q})$ is I-Lipschitz continuous w.r.t. $\boldsymbol{G}$ and $\gamma \leq 1$, we have

$$u_G\left(\boldsymbol{G}; \hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right) \geq F\left(\boldsymbol{G}, \hat{\boldsymbol{Q}}\right), \ u_G\left(\hat{\boldsymbol{G}}; \hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right) = F\left(\hat{\boldsymbol{G}}, \hat{\boldsymbol{Q}}\right). \tag{17}$$

Hence, Algorithm 1 boils down to

$$\boldsymbol{Q}_i^{(r,t+1)} = \arg\min_{\boldsymbol{Q}_i} \ u_Q\left(\boldsymbol{Q}; \boldsymbol{G}^{(r)}, \boldsymbol{Q}^{(r,t)}\right), \quad \forall t \tag{18a}$$

$$\boldsymbol{G}^{(r+1)} \in \arg\min_{\boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}} \ u_G\left(\boldsymbol{G}; \boldsymbol{G}^{(r)}, \boldsymbol{Q}^{(r+1)}\right). \tag{18b}$$

When $\gamma = 1$, (18b) amounts to SVD of $\sum_{i=1}^{I} \boldsymbol{X}_i \boldsymbol{Q}_i / I$ and the $\boldsymbol{G}$-subproblem $\min_{\boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}} F(\boldsymbol{Q}^{(r+1)}, \boldsymbol{G})$ is optimally solved; otherwise, both (18a) and (18b) are local upper bound minimizations.

Note that the following holds:

$$\begin{aligned} F\left(\boldsymbol{G}^{(r)}, \boldsymbol{Q}^{(r)}\right) &= u_Q(\boldsymbol{Q}^{(r)}; \boldsymbol{G}^{(r)}, \boldsymbol{Q}^{(r)}) && \text{(19a)} \\ &\geq u_Q(\boldsymbol{Q}^{(r+1)}; \boldsymbol{G}^{(r)}, \boldsymbol{Q}^{(r,T-1)}) && \text{(19b)} \\ &\geq F\left(\boldsymbol{G}^{(r)}, \boldsymbol{Q}^{(r+1)}\right) && \text{(19c)} \\ &= u_G\left(\boldsymbol{G}^{(r)}; \boldsymbol{G}^{(r)}, \boldsymbol{Q}^{(r+1)}\right) && \text{(19d)} \\ &\geq u_G\left(\boldsymbol{G}^{(r+1)}; \boldsymbol{G}^{(r)}, \boldsymbol{Q}^{(r+1)}\right) && \text{(19e)} \\ &\geq F\left(\boldsymbol{G}^{(r+1)}, \boldsymbol{Q}^{(r+1)}\right), && \text{(19f)} \end{aligned}$$

where (19a) holds because of (14), (19b) holds since PG is a descending method when $\alpha_i \leq 1/L_i$ [33], (19c) holds by the property in (14), (19d) holds due to (17), (19e) is due to the fact that (18b) is optimally solved, and (19f) holds also because of the first equation in (17).

Next, we show that every limit point is a KKT point. Assume that there exists a convergent subsequence of $\{\boldsymbol{G}^{(r)}, \boldsymbol{Q}^{(r)}\}_{r=0,1,\ldots}$, whose limit point is $(\boldsymbol{G}^*, \boldsymbol{Q}^*)$ and the subsequence is indexed by $\{r_j\}_{j=1,\ldots,\infty}$. We have the following chain of inequalities:

$$\begin{aligned} u_Q\left(\boldsymbol{Q}; \boldsymbol{G}^{(r_j)}, \boldsymbol{Q}^{(r_j)}\right) &\geq u_Q\left(\boldsymbol{Q}^{(r_j,1)}; \boldsymbol{G}^{(r_j)}, \boldsymbol{Q}^{(r_j)}\right) && \text{(20a)} \\ &\geq u_Q\left(\boldsymbol{Q}^{(r_j,T)}; \boldsymbol{G}^{(r_j)}, \boldsymbol{Q}^{(r_j,T-1)}\right) && \text{(20b)} \\ &\geq F(\boldsymbol{G}^{(r_j)}, \boldsymbol{Q}^{(r_j+1)}) && \text{(20c)} \\ &\geq F\left(\boldsymbol{G}^{(r_j+1)}, \boldsymbol{Q}^{(r_j+1)}\right) && \text{(20d)} \\ &\geq F\left(\boldsymbol{G}^{(r_{j+1})}, \boldsymbol{Q}^{(r_{j+1})}\right) && \text{(20e)} \\ &= u_Q\left(\boldsymbol{Q}^{(r_{j+1})}; \boldsymbol{G}^{(r_{j+1})}, \boldsymbol{Q}^{(r_{j+1})}\right), && \text{(20f)} \end{aligned}$$

where (20a) holds because of the update rule in (18a), (20b) holds, again, by the descending property of PG, (20d) follows (19f), and (20f) is again because of the way that we construct $u_Q(\boldsymbol{Q}; \boldsymbol{G}^{(r_{j+1})}, \boldsymbol{Q}^{(r_{j+1})})$. Taking $j \to \infty$, and by continuity of $u_Q(\cdot)$, we have

$$u_Q(\boldsymbol{Q}; \boldsymbol{G}^*, \boldsymbol{Q}^*) \geq u_Q(\boldsymbol{Q}^*; \boldsymbol{G}^*, \boldsymbol{Q}^*), \tag{21}$$

i.e., $\boldsymbol{Q}^*$ is a minimum of $u_Q(\boldsymbol{Q}; \boldsymbol{G}^*, \boldsymbol{Q}^*)$. Consequently, $\boldsymbol{Q}^*$ satisfies the conditional KKT conditions, i.e., $\boldsymbol{0} \in \nabla_{\boldsymbol{Q}} \tilde{u}_Q(\boldsymbol{Q}^*; \boldsymbol{G}^*, \boldsymbol{Q}^*) + \partial_{\boldsymbol{Q}} g(\boldsymbol{Q}^*)$, which also means that the following holds:

$$\boldsymbol{0} \in \nabla_{\boldsymbol{Q}_i} f(\boldsymbol{G}^*, \boldsymbol{Q}^*) + \partial_{\boldsymbol{Q}} g(\boldsymbol{Q}^*), \tag{22}$$

following (12).

We now show that $\boldsymbol{Q}^{(r_j,t)}$ for $t = 1, \ldots, T$ also converges to $\boldsymbol{Q}^*$. Indeed, we have

$$\begin{aligned} u_Q(\boldsymbol{Q}^{(r_{j+1})}; \boldsymbol{G}^{(r_{j+1})}, \boldsymbol{Q}^{(r_{j+1})}) &\leq u_Q(\boldsymbol{Q}^{(r_j,1)}; \boldsymbol{G}^{(r_j)}, \boldsymbol{Q}^{(r_j)}) \\ &\leq u_Q(\boldsymbol{Q}^{(r_j)}; \boldsymbol{G}^{(r_j)}, \boldsymbol{Q}^{(r_j)}), \end{aligned}$$

where the first inequality was derived from (20). Taking $j \to \infty$, we see that $u_Q(\boldsymbol{Q}^*; \boldsymbol{G}^*, \boldsymbol{Q}^*) \leq u_Q(\boldsymbol{Q}^{(r_j,1)}; \boldsymbol{G}^*, \boldsymbol{Q}^*) \leq u_Q(\boldsymbol{Q}^*; \boldsymbol{G}^*, \boldsymbol{Q}^*)$, which implies that $u_Q(\boldsymbol{Q}^{(r_j,1)}; \boldsymbol{G}^*, \boldsymbol{Q}^*) = u_Q(\boldsymbol{Q}^*; \boldsymbol{G}^*, \boldsymbol{Q}^*) \leq u_Q(\boldsymbol{Q}; \boldsymbol{G}^*, \boldsymbol{Q}^*)$. On the other hand, the

problem in (18a) has a unique minimizer when $g_i(\cdot)$ is a convex closed function [27], which means that $\boldsymbol{Q}^{(r_j,1)} \to \boldsymbol{Q}^*$. By the same argument, we can show that $\boldsymbol{Q}^{(r_j,t)}$ for $t = 1,\ldots,T-1$ also converges to $\boldsymbol{Q}^*$ using the same argument. Consequently, we have $\boldsymbol{Q}^{(r_j,T)} = \boldsymbol{Q}^{(r_j+1)} \to \boldsymbol{Q}^*$. Now, we repeat the proof in (20) to $\boldsymbol{G}$:

$$\begin{aligned} u_G\left(\boldsymbol{G};\boldsymbol{G}^{(r_j)},\boldsymbol{Q}^{(r_j+1)}\right) &\geq u_G\left(\boldsymbol{G}^{(r_j+1)};\boldsymbol{G}^{(r_j)},\boldsymbol{Q}^{(r_j+1)}\right) \\ &\geq F(\boldsymbol{G}^{(r_j+1)},\boldsymbol{Q}^{(r_j+1)}) \\ &\geq F\left(\boldsymbol{G}^{(r_j+1)},\boldsymbol{Q}^{(r_j+1)}\right) \\ &= u_G\left(\boldsymbol{G}^{(r_{j+1})};\boldsymbol{G}^{(r_{j+1})},\boldsymbol{Q}^{(r_{j+1})}\right), \end{aligned}$$

Taking $j \to \infty$ and invoking (**??**), we have

$$u_G\left(\boldsymbol{G};\boldsymbol{G}^*,\boldsymbol{Q}^*\right) \geq u_G\left(\boldsymbol{G}^*;\boldsymbol{G}^*,\boldsymbol{Q}^*\right), \quad \forall \boldsymbol{G}^T\boldsymbol{G} = \boldsymbol{I}.$$

The above means that $\boldsymbol{G}^*$ satisfies the partial conditional KKT conditions w.r.t. $\boldsymbol{G}$. Combining with (22), we see that $(\boldsymbol{G}^*,\boldsymbol{Q}^*)$ is a KKT point of the original problem.

Now, we show the b) part. First, we show that $\boldsymbol{Q}_i$ remains in a bounded set (the variable $\boldsymbol{G}$ is always bounded since we keep it feasible in each iteration). Since the objective value is non-increasing (cf. Proposition 1), if we denote the initial objective value as V, then $F(\boldsymbol{G}^{(r)},\boldsymbol{Q}^{(r)}) \leq V$ holds in all subsequent iterations. Note that when $\boldsymbol{X}_i^{(0)}$ and $\boldsymbol{Q}_i^{(0)}$ are bounded, V is also finite. In particular, we have $\|\boldsymbol{X}_i\boldsymbol{Q}_i - \boldsymbol{G}\|_F^2 + 2\sum_{i=1}^I g_i(\boldsymbol{Q}_i) \leq 2V$ holds, which implies $\|\boldsymbol{X}_i\boldsymbol{Q}_i\|_F \leq \|\boldsymbol{G}\|_F + \sqrt{2V}$ by the triangle inequality. The right-hand side is finite since both terms are bounded. Denote $(\|\boldsymbol{G}\|_F+\sqrt{2V})$ by V'. Then, we have $\|\boldsymbol{Q}_i\|_F = \|(\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T\boldsymbol{X}_i\boldsymbol{Q}_i\|_F \leq \|(\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T\|_F \cdot \|\boldsymbol{X}_i\boldsymbol{Q}_i\|_F \leq V' \cdot \|(\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T\|_F$. Now, by the assumption that $\mathrm{rank}(\boldsymbol{X}_i) = M_i$, the term $\|(\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T\|_F$ is bounded. This shows that $\|\boldsymbol{Q}_i\|_F$ is bounded. Hence, starting from a bounded $\boldsymbol{Q}_i^{(0)}$, the solution sequence $\{\boldsymbol{Q}(r),\boldsymbol{G}^{(r)}\}$ remains in a bounded set. Since the constraints of $\boldsymbol{Q}_i$, i.e., $\mathbb{R}^{M_i\times K}$ and $\boldsymbol{G}$ are also closed sets, $\{\boldsymbol{Q}^{(r)},\boldsymbol{G}^{(r)}\}$ remains in a compact set.

Now, let us denote $\mathcal{K}$ as the set containing all the KKT points. Suppose the whole sequence does not converge to $\mathcal{K}$. Then, there exists a convergent subsequence indexed by $\{r_j\}$ such that $\lim_{j\to\infty} d^{(r)}(\mathcal{K}) \geq \gamma$ for some positive γ, where $d^{(r)}(\mathcal{K}) = \min_{\boldsymbol{Y}\in\mathcal{K}} \|(\boldsymbol{G}^{(r)},\boldsymbol{Q}^{(r)}) - \boldsymbol{Y}\|$. Since the subsequence indexed by $\{r_j\}$ lies in a closed and bounded set as we have shown, this subsequence has a limit point. However, as we have shown in Theorem 1, every limit point of the solution sequence is a KKT point. This is a contradiction. Therefore, the whole sequence converges to a KKT point.

Appendix B
Proof of Lemma 1

First, we have the update rule $\boldsymbol{Q}_i^{(r,t+1)} = \boldsymbol{Q}_i^{(r,t)} - \alpha_i\tilde{\nabla}_{\boldsymbol{Q}_i}F(\boldsymbol{Q}_i^{(r,t)},\boldsymbol{G}^{(r)})$, which yields the following relationship:

$$\frac{1}{\alpha_i}(\boldsymbol{Q}_i^{(r,t+1)} - \boldsymbol{Q}_i^{(r,t)}) = -\tilde{\nabla}_{\boldsymbol{Q}_i}F(\boldsymbol{Q}_i^{(r,t)},\boldsymbol{G}^{(r)}). \tag{24}$$

Meanwhile, the updating rule can also be expressed as

$$\begin{aligned} \boldsymbol{Q}_i^{(r+1,t)} = \arg\min_{\boldsymbol{Q}_i} &\left\langle \nabla_{\boldsymbol{Q}_i} f(\boldsymbol{Q}_i^{(r,t)},\boldsymbol{G}^{(r)}), \boldsymbol{Q}_i - \boldsymbol{Q}_i^{(r,t)}\right\rangle \\ &+ g_i(\boldsymbol{Q}_i) + \frac{1}{2\alpha_i}\|\boldsymbol{Q}_i - \boldsymbol{Q}_i^{(r,t)}\|_F^2. \end{aligned} \tag{25}$$

Therefore, there exists a $\partial_{\boldsymbol{Q}_i} g_i(\boldsymbol{Q}^{(r+1,t)})$ and a $\boldsymbol{Q}^{(r+1,t)}$ satisfy the following optimality conditions:

$$\boldsymbol{0} = \nabla_{\boldsymbol{Q}} f(\boldsymbol{Q}^{(r,t)},\boldsymbol{G}^{(r)}) + \partial_{\boldsymbol{Q}} g(\boldsymbol{Q}^{(r,t+1)}) + \frac{1}{\alpha_i}(\boldsymbol{Q}_i^{(r,t+1)} - \boldsymbol{Q}_i^{(r,t)}).$$

Consequently, we see that

$$\begin{aligned} &\sum_{i=1}^I\sum_{t=0}^T \left\|\tilde{\nabla}_{\boldsymbol{Q}_i}F(\boldsymbol{Q}_i^{(r,t)},\boldsymbol{G}^{(r)})\right\|_F^2 \to 0 \\ &\Rightarrow \boldsymbol{Q}_i^{(r,t)} - \boldsymbol{Q}_i^{(r,t+1)} \to \boldsymbol{0},\ \forall\ t = 0,\ldots,T-1 \\ &\Rightarrow \boldsymbol{Q}_i^{(r)} - \boldsymbol{Q}_i^{(r+1)} \to 0,\ \forall i \\ &\Rightarrow \nabla_{\boldsymbol{Q}}\ f\left(\boldsymbol{Q}^{(r)},\boldsymbol{G}^{(r)}\right) + \partial_{\boldsymbol{Q}} g\left(\boldsymbol{Q}^{(r)}\right) \to 0 \end{aligned}$$

which holds since T is finite. The above means that $\boldsymbol{0} \in \nabla_{\boldsymbol{Q}} f(\boldsymbol{Q}^{(r)},\boldsymbol{G}^{(r)}) + \partial_{\boldsymbol{Q}} g(\boldsymbol{Q}^{(r)})$ is satisfied when $Z^{r,r+1} \to 0$.

Recall that the update rule of $\boldsymbol{G}$ is equivalent to solving

$$\min_{\boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}} \quad \left\|\boldsymbol{G} - \left(\boldsymbol{G}^{(r)} - \gamma_I\left(\nabla_{\boldsymbol{G}}\ f(\boldsymbol{G}^{(r)},\boldsymbol{Q}^{(r+1)})\right)\right)\right\|_F^2. \tag{26}$$

Therefore, following the argument in (25), we have

$$\begin{aligned} \boldsymbol{G}^{(r+1)} \in \arg\min_{\boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}} &\left\langle \nabla_{\boldsymbol{G}} f(\boldsymbol{Q}^{(r+1)},\boldsymbol{G}^{(r)}), \boldsymbol{G} - \boldsymbol{G}^{(r)}\right\rangle \\ &+ \frac{1}{2\tilde{\gamma}}\|\boldsymbol{G} - \boldsymbol{G}^{(r)}\|_F^2, \end{aligned} \tag{27}$$

where $\tilde{\gamma} = \gamma/I$ and hence we have the following optimality condition holds

$$\begin{aligned} &\nabla_{\boldsymbol{G}} f(\boldsymbol{Q}^{(r+1)},\boldsymbol{G}^{(r)}) + \frac{1}{\tilde{\gamma}}\left(\boldsymbol{G}^{(r+1)} - \boldsymbol{G}^{(r)}\right) \\ &\quad + \boldsymbol{G}^{(r+1)}\boldsymbol{\Lambda}^{(r+1)} = \boldsymbol{0} \end{aligned} \tag{28}$$

Combining (28) and (24), we have

$$Z^{(r,r+1)} = \frac{1}{\tilde{\gamma}^2}\left\|\boldsymbol{G}^{(r+1)} - \boldsymbol{G}^{(r)}\right\|_F^2 + \sum_{i=1}^I \frac{1}{\alpha_i^2}\left\|\boldsymbol{Q}_i^{(r+1)} - \boldsymbol{Q}_i^{(r)}\right\|_F^2. \tag{29}$$

We see that $Z^{(r,r+1)} \to 0$ implies that $(\boldsymbol{G}^{(r+1)},\boldsymbol{Q}^{(r+1)}) \to (\boldsymbol{G}^{(r)},\boldsymbol{Q}^{(r)})$ and that a KKT point is reached and this completes the proof of Lemma 1.

Appendix C
Proof of Theorem 1

We show that every iterate of $\boldsymbol{Q}$ and $\boldsymbol{G}$ gives sufficiently large decrease of the overall objective function. Since $\nabla_{\boldsymbol{Q}_i} f(\boldsymbol{Q},\boldsymbol{G})$ is L_i-Lipschitz continuous for all i, we have the following:

$$\begin{aligned} F(\boldsymbol{Q}^{(r,t+1)},\boldsymbol{G}^{(r)}) \leq\ & f(\boldsymbol{Q}^{(r,t)},\boldsymbol{G}^{(r)}) \\ &+ \left\langle \nabla_{\boldsymbol{Q}} f(\boldsymbol{Q}^{(r,t)},\boldsymbol{G}^{(r)}), \boldsymbol{Q} - \boldsymbol{Q}^{(r)}\right\rangle \\ &+ \sum_{i=1}^I g_i(\boldsymbol{Q}_i) + \sum_{i=1}^I \frac{L_i}{2}\left\|\boldsymbol{Q}_i - \boldsymbol{Q}_i^{(r,t)}\right\|_F^2. \end{aligned} \tag{30}$$

Since $\boldsymbol{Q}^{(r+1,t)}$ is a minimizer of Problem (25), we also have

$$\left\langle \nabla_{\boldsymbol{Q}} f(\boldsymbol{Q}^{(r,t)}, \boldsymbol{G}^{(r)}), \boldsymbol{Q}^{(r,t+1)} - \boldsymbol{Q}^{(r,t)} \right\rangle + \sum_{i=1}^{I} g_i(\boldsymbol{Q}_i^{(r+1)}) + \sum_{i=1}^{I} \frac{1}{2\alpha_i} \left\| \boldsymbol{Q}_i^{(r,t+1)} - \boldsymbol{Q}_i^{(r,t)} \right\|_F^2 \leq \sum_{i=1}^{I} g_i(\boldsymbol{Q}_i^{(r,t+1)}), \quad (31)$$

which is obtained by letting $\boldsymbol{Q}_i = \boldsymbol{Q}_i^{(r,t)}$. Combining (30) and (31), we have

$$F(\boldsymbol{Q}^{(r,t+1)}, \boldsymbol{G}^{(r)}) - F(\boldsymbol{Q}^{(r,t)}, \boldsymbol{G}^{(r)}) \leq -\sum_{i=1}^{I} \left(\frac{1}{2\alpha_i} - \frac{L_i}{2} \right) \left\| \boldsymbol{Q}_i^{(r,t+1)} - \boldsymbol{Q}_i^{(r,t)} \right\|_F^2. \quad (32)$$

Summing up the above over $t = 0, \ldots, T-1$, we have

$$F(\boldsymbol{Q}^{(r)}, \boldsymbol{G}^{(r)}) - F(\boldsymbol{Q}^{(r+1)}, \boldsymbol{G}^{(r)}) \geq \sum_{t=0}^{T-1} \sum_{i=1}^{I} \left(\frac{1}{2\alpha_i} - \frac{L_i}{2} \right) \left\| \boldsymbol{Q}_i^{(r,t+1)} - \boldsymbol{Q}_i^{(r,t)} \right\|_F^2. \quad (33)$$

By the same derivation, we have

$$F(\boldsymbol{Q}^{(r+1)}, \boldsymbol{G}^{(r+1)}) - F(\boldsymbol{Q}^{(r+1)}, \boldsymbol{G}^{(r)}) \leq -\left(\frac{1}{2\tilde{\gamma}} - \frac{1}{2} \right) \left\| \boldsymbol{G}^{(r+1)} - \boldsymbol{G}^{(r)} \right\|_F^2, \quad \forall \boldsymbol{G}^T \boldsymbol{G} = \boldsymbol{I}. \quad (34)$$

Combining (33) and (34), we have

$$\begin{aligned} &F(\boldsymbol{Q}^{(r)}, \boldsymbol{G}^{(r)}) - F(\boldsymbol{Q}^{(r+1)}, \boldsymbol{G}^{(r+1)}) \\ &\geq \left(\frac{1}{2\tilde{\gamma}} - \frac{1}{2} \right) \left\| \boldsymbol{G}^{(r+1)} - \boldsymbol{G}^{(r)} \right\|_F^2 \\ &\quad + \sum_{t=0}^{T-1} \sum_{i=1}^{I} \left(\frac{1}{2\alpha_i} - \frac{L_i}{2} \right) \left\| \boldsymbol{Q}_i^{(r,t+1)} - \boldsymbol{Q}_i^{(r,t)} \right\|_F^2. \end{aligned} \quad (35)$$

Summing up $F(\boldsymbol{Q}^{(r)}, \boldsymbol{G}^{(r)})$ over $r = 0, 1, \ldots, J-1$, we have the following:

$$\begin{aligned} &F(\boldsymbol{Q}^{(r)}, \boldsymbol{G}^{(r)}) - F(\boldsymbol{Q}^{(r+1)}, \boldsymbol{G}^{(r+1)}) \\ &\geq \sum_{r=0}^{J-1} \left(\frac{1}{2\tilde{\gamma}} - \frac{1}{2} \right) \left\| \boldsymbol{G}^{(r+1)} - \boldsymbol{G}^{(r)} \right\|_F^2 \\ &\quad + \sum_{r=0}^{J-1} \sum_{t=0}^{T-1} \sum_{i=1}^{I} \left(\frac{1}{2\alpha_i} - \frac{L_i}{2} \right) \left\| \boldsymbol{Q}_i^{(r,t+1)} - \boldsymbol{Q}_i^{(r,t)} \right\|_F^2. \\ &= \sum_{r=0}^{J-1} \left(\frac{1}{2\tilde{\gamma}} - \frac{1}{2} \right) \tilde{\gamma}^2 \left\| \nabla_{\boldsymbol{G}} f(\boldsymbol{Q}^{(r+1)}, \boldsymbol{G}^{(r)}) + \boldsymbol{G}^{(r+1)} \boldsymbol{\Lambda}^{(r+1)} \right\|_F^2 \\ &\quad + \sum_{r=0}^{J-1} \sum_{i=1}^{I} \sum_{t=0}^{T-1} \left(\frac{1}{2\alpha_i} - \frac{L_i}{2} \right) \alpha_i^2 \left\| \tilde{\nabla}_{\boldsymbol{Q}_i} F(\boldsymbol{Q}^{(r,t)}, \boldsymbol{G}^{(r)}) \right\|_F^2 \\ &\geq \sum_{r=0}^{J-1} c Z^{(r,r+1)}, \end{aligned} \quad (36)$$

where $c = \min\{(\frac{1}{2\tilde{\gamma}} - \frac{1}{2})\tilde{\gamma}^2, \{(\frac{1}{2\alpha_i} - \frac{L_i}{2})\alpha_i^2\}_{i=1,\ldots,I}\}$. By the definition of J, we have

$$\frac{F(\boldsymbol{Q}^{(0)}, \boldsymbol{G}^{(0)}) - F(\boldsymbol{Q}^{(J)}, \boldsymbol{G}^{(J)})}{J-1} \geq \frac{\sum_{r=0}^{J-1} c Z^{(r,r+1)}}{J-1} \geq c \cdot \epsilon$$
$$\Rightarrow \epsilon \leq \frac{1}{c} \frac{F(\boldsymbol{Q}^{(0)}, \boldsymbol{G}^{(0)}) - \bar{F}}{J-1} \Rightarrow \epsilon \leq \frac{v}{J-1},$$

where $\bar{F}$ is the lower bound of the cost function and $v = {}^{(F(\boldsymbol{Q}^{(0)}, \boldsymbol{G}^{(0)}) - \bar{F})}/_{c}$. This completes the proof.

APPENDIX D
PROOF OF THEOREM 2

Let us first consider an easier case where $\epsilon = 0$, i.e., the subproblem w.r.t. $\boldsymbol{Q}_i$ is solved at every outer iteration. Then, by the first-order optimality condition and the assumption that $\boldsymbol{X}_i$ has full column rank, we have $\boldsymbol{Q}_i^{(r+1)} = (\boldsymbol{X}_i^T \boldsymbol{X}_i)^{-1} \boldsymbol{X}_i^T \boldsymbol{G}^{(r+1)}$. Therefore, the update w.r.t. $\boldsymbol{G}$ is simply to apply SVD on $\sum_{i=1}^{I} \boldsymbol{X}_i \boldsymbol{Q}_i / I = \boldsymbol{M}\boldsymbol{G}^{(r)}/I$. In other words, there exists an invertible $\boldsymbol{\Theta}^{(r+1)}$ such that

$$\boldsymbol{G}^{(r+1)} \boldsymbol{\Theta}^{(r+1)} = \boldsymbol{M}\boldsymbol{G}^{(r)}, \quad (37)$$

since the SVD procedure is nothing but a change of bases. The update rule in (37), is essentially the orthogonal iteration algorithm in [25]. Invoking [25, Theorem 8.2.2], one can show that $\|\boldsymbol{U}_2^T \boldsymbol{G}^{(r)}\|_2$ approaches zero linearly.

The proof of the case where $\epsilon > 0$ can be considered as an extension of round-off error analysis of orthogonal iterations and follows the insight of the proof in [15]; proper modifications are made to accommodate the problem structure of MAX-VAR GCCA. At the rth iteration, ideally, we have $\tilde{\boldsymbol{Q}}_i^{(r+1)} = (\boldsymbol{X}_i^T \boldsymbol{X}_i)^{-1} \boldsymbol{X}_i^T \boldsymbol{G}^{(r)}$ if the $\boldsymbol{Q}$-subproblem is solved to optimality. In practice, what we have is an inexact solution, i.e., $\boldsymbol{Q}_i^{(r+1)} = (\boldsymbol{X}_i^T \boldsymbol{X}_i)^{-1} \boldsymbol{X}_i^T \boldsymbol{G}^{(r)} + \boldsymbol{W}_i^{(r)}$, where we assume that the largest singular value of $\boldsymbol{W}_i^{(r)}$ is bounded by ϵ, i.e., $\|\boldsymbol{W}_i^{(r)}\|_2 \leq \epsilon$. Hence, we see that $\sum_{i=1}^{I} \boldsymbol{X}_i^T \boldsymbol{Q}_i^{(r+1)} = \boldsymbol{M}\boldsymbol{G}^{(r)} + \sum_{i=1}^{I} \boldsymbol{X}_i \boldsymbol{W}_i^{(r)}$. Therefore, following the same reason of obtaining (37), we have

$$\boldsymbol{G}^{(r+1)} = \left(\boldsymbol{M}\boldsymbol{G}^{(r)} + \sum_{i=1}^{I} \boldsymbol{X}_i \boldsymbol{W}_i^{(r)} \right) \boldsymbol{\Theta}^{(r)},$$

where $\boldsymbol{\Theta}^{(r)} \in \mathbb{R}^{K \times K}$ is a full-rank matrix since the solution via SVD is a change of bases. Consequently, we have

$$\begin{bmatrix} \boldsymbol{U}_1^T \boldsymbol{G}^{(r+1)} \\ \boldsymbol{U}_2^T \boldsymbol{G}^{(r+1)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Lambda}_1 \boldsymbol{U}_1^T \boldsymbol{G}^{(r)} + \boldsymbol{U}_1^T \sum_{i=1}^{I} \boldsymbol{X}_i \boldsymbol{W}_i^{(r)} \\ \boldsymbol{\Lambda}_2 \boldsymbol{U}_2^T \boldsymbol{G}^{(r)} + \boldsymbol{U}_2^T \sum_{i=1}^{I} \boldsymbol{X}_i \boldsymbol{W}_i^{(r)} \end{bmatrix} \boldsymbol{\Theta}^{(r)}.$$

Now, denote $\boldsymbol{F}^{(r)} = \sum_{i=1}^{I} \boldsymbol{X}_i \boldsymbol{W}_i^{(r)}$. Then, we have

$$\begin{aligned} &\left\| \boldsymbol{U}_2^T \boldsymbol{G}^{(r+1)} \left(\boldsymbol{U}_1^T \boldsymbol{G}^{(r+1)} \right)^{-1} \right\|_2 \\ &= \left\| \left(\boldsymbol{\Lambda}_2 \boldsymbol{U}_2^T \boldsymbol{G}^{(r)} + \boldsymbol{U}_2^T \boldsymbol{F}^{(r)} \right) \left(\boldsymbol{\Lambda}_2 \boldsymbol{U}_2^T \boldsymbol{G}^{(r)} + \boldsymbol{U}_2^T \boldsymbol{F}^{(r)} \right)^{-1} \right\|_2. \end{aligned} \quad (38)$$

Note that we can normalize the matrix $\boldsymbol{U}_1^T \boldsymbol{F}^{(r)}$ as follows $\boldsymbol{U}_2^T \boldsymbol{F}^{(r)} = \tau \cdot (\boldsymbol{U}_2^T \boldsymbol{F}^{(r)} / \|\boldsymbol{U}_2^T \boldsymbol{F}^{(r)}\|_2) = \tau \tilde{\boldsymbol{W}}^{(r)}$, where τ can be shown to be bounded by $\tau \leq \sum_{i=1}^{I} \lambda_{\max}(\boldsymbol{X}_i)\epsilon$. Consider the Taylor expansion $(\boldsymbol{\Lambda}_2 \boldsymbol{U}_2^T \boldsymbol{G}^{(r)} + \boldsymbol{U}_2^T \boldsymbol{F}^{(r)})^{-1} = \left(\boldsymbol{\Lambda}_1 \boldsymbol{U}_1^T \boldsymbol{G}^{(r)} \right)^{-1} + \tau \left(\boldsymbol{\Lambda}_1 \boldsymbol{U}_1^T \boldsymbol{G}^{(r)} \right)^{-1} \tilde{\boldsymbol{W}}^{(r)} \left(\boldsymbol{\Lambda}_1 \boldsymbol{U}_1^T \boldsymbol{G}^{(r)} \right)^{-1} + \mathcal{O}(\tau^2)$. Now, let us drop the second- and higher-oder terms of τ which are sufficiently small and 'absorb' them in $\mathcal{O}(\|(\boldsymbol{U}_1^T \boldsymbol{G}^{(r)})^{-1}\|_2^2)$. Consequently, we can upper bound the

left hand side of (38) by the following:

$$\begin{aligned}&\left\|\boldsymbol{U}_2^T\boldsymbol{G}^{(r+1)}\left(\boldsymbol{U}_1^T\boldsymbol{G}^{(r+1)}\right)^{-1}\right\|_2\\ \leq&\frac{\lambda_{K+1}}{\lambda_K}\left\|(\boldsymbol{U}_2^T\boldsymbol{G}^{(r)})\left(\boldsymbol{U}_1^T\boldsymbol{G}^{(r)}\right)^{-1}\right\|_2+C_1\end{aligned}\tag{39}$$

where $C_1=(\sum_{i=1}^I\lambda_{\max}(\boldsymbol{X}_i)\epsilon)\mathcal{O}(\|(\boldsymbol{U}_1^T\boldsymbol{G}^{(r)})^{-1}\|_2^2)$. Now, we show that $\|(\boldsymbol{U}_1^T\boldsymbol{G}^{(r)})^{-1}\|_2^2$ is bounded for all r. This can be seen by induction. For $r=1$, we see that $\|\boldsymbol{U}_2^T\boldsymbol{G}^{(1)}t(\boldsymbol{U}_1^T\boldsymbol{G}^{(1)})^{-1}\|_2$ has to be bounded since we assumed that $\mathrm{rank}(\boldsymbol{U}_1^T\boldsymbol{G}^{(0)})=K$ and since (39) holds. Using the same argument, we see that for all finite $r\geq 1$, $(\boldsymbol{U}_1^T\boldsymbol{G}^{(1)})^{-1}$ is bounded. Let us denote an upper bound as β, i.e., $\|(\boldsymbol{U}_1^T\boldsymbol{G}^{(r)})^{-1}\|_2\leq\beta,\quad\forall r$. Then, we have $C_1\leq C_2=(\sum_{i=1}^I\lambda_{\max}(\boldsymbol{X}_i)\epsilon)\mathcal{O}(\beta^2)$. The above leads to

$$\begin{aligned}\left\|\boldsymbol{U}_2^T\boldsymbol{G}^{(r+1)}\right\|_2\leq&\left(\frac{\lambda_{K+1}}{\lambda_K}\right)^r\left\|(\boldsymbol{U}_2^T\boldsymbol{G}^{(0)})\left(\boldsymbol{U}_1^T\boldsymbol{G}^{(0)}\right)^{-1}\right\|_2\\&+\sum_{t=0}^{r-1}\left(\frac{\lambda_{K+1}}{\lambda_K}\right)^t\left(\sum_{i=1}^I\lambda_{\max}(\boldsymbol{X}_i)\epsilon\right)\mathcal{O}\left(\beta^2\right),\end{aligned}$$

where the first inequality because of $\|\boldsymbol{U}_1^T\boldsymbol{G}^{(r+1)}\|_2\leq 1$. Note that we have $\|\boldsymbol{U}_2^T\boldsymbol{G}^{(0)}\|_2=\sin(\theta)$ and $\|(\boldsymbol{U}_1^T\boldsymbol{G}^{(0)})^{-1}\|_2=1/\cos(\theta)$ [25, Theorem 8.2.2]. Therefore, the proof is completed.

REFERENCES

[1] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural computation*, vol. 16, no. 12, pp. 2639–2664, 2004.

[2] Y.-O. Li, T. Adali, W. Wang, and V. D. Calhoun, "Joint blind source separation by multiset canonical correlation analysis," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 3918–3929, 2009.

[3] A. Bertrand and M. Moonen, "Distributed canonical correlation analysis in wireless sensor networks with application to distributed blind source separation," *IEEE Trans. Signal Process.*, vol. 63, no. 18, pp. 4800–4813, 2015.

[4] Q. Wu and K. M. Wong, "Un-music and un-cle: An application of generalized correlation analysis to the estimation of the direction of arrival of signals in unknown correlated noise," *IEEE Trans. Signal Process.*, vol. 42, no. 9, pp. 2331–2343, 1994.

[5] A. Dogandzic and A. Nehorai, "Finite-length mimo equalization using canonical correlation analysis," *IEEE Trans. Signal Process.*, vol. 50, no. 4, pp. 984–989, 2002.

[6] S. M. Kakade and D. P. Foster, "Multi-view regression via canonical correlation analysis," in *Learning Theory*. Springer, 2007, pp. 82–96.

[7] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan, "Multi-view clustering via canonical correlation analysis," in *Proc. ICML*. ACM, 2009, pp. 129–136.

[8] R. Arora and K. Livescu, "Multi-view learning with supervision for transformed bottleneck features," in *Proc. ICASSP*. IEEE, 2014, pp. 2499–2503.

[9] H. Wang, T. Lee, C.-C. Leung, B. Ma, and H. Li, "Acoustic segment modeling with spectral clustering methods," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 2, pp. 264–277, 2015.

[10] P. Rastogi, B. Van Durme, and R. Arora, "Multiview LSA: Representation learning via generalized cca," in *Proc. NAACL*, 2015.

[11] J. D. Carroll, "Generalization of canonical correlation analysis to three or more sets of variables," in *Proc. annual convention of the American Psychological Association*, vol. 3, 1968, pp. 227–228.

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

[13] Z. Ma, Y. Lu, and D. Foster, "Finding linear structure in large datasets with scalable canonical correlation analysis," *arXiv preprint arXiv:1506.08170*, 2015.

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

[15] Y. Lu and D. P. Foster, "Large scale canonical correlation analysis with iterative least squares," in *Proc. NIPS*, 2014, pp. 91–99.

[16] M. Van De Velden and T. H. A. Bijmolt, "Generalized canonical correlation analysis of matrices with missing rows: a simulation study," *Psychometrika*, vol. 71, no. 2, pp. 323–331, 2006.

[17] D. R. Hardoon and J. Shawe-Taylor, "Sparse canonical correlation analysis," *Machine Learning*, vol. 83, no. 3, pp. 331–353, 2011.

[18] D. M. Witten, R. Tibshirani, and T. Hastie, "A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis," *Biostatistics*, p. kxp008, 2009.

[19] X. Chen, H. Liu, and J. G. Carbonell, "Structured sparse canonical correlation analysis," in *International Conference on Artificial Intelligence and Statistics*, 2012, pp. 199–207.

[20] D. M. Witten and R. J. Tibshirani, "Extensions of sparse canonical correlation analysis with applications to genomic data," *Statistical applications in genetics and molecular biology*, vol. 8, no. 1, pp. 1–27, 2009.

[21] M. Faruqui and C. Dyer, "Improving vector space word representations using multilingual correlation." Association for Computational Linguistics, 2014.

[22] I. Rustandi, M. A. Just, and T. Mitchell, "Integrating multiple-study multiple-subject fmri datasets using canonical correlation analysis," in *Proceedings of the MICCAI 2009 Workshop: Statistical modeling and detection issues in intra-and inter-subject functional MRI data analysis*, 2009.

[23] T. M. Mitchell, S. V. Shinkareva, A. Carlson, K.-M. Chang, V. L. Malave, R. A. Mason, and M. A. Just, "Predicting human brain activity associated with the meanings of nouns," *science*, vol. 320, no. 5880, pp. 1191–1195, 2008.

[24] J. Rupnik, P. Skraba, J. Shawe-Taylor, and S. Guettes, "A comparison of relaxations of multiset cannonical correlation analysis and applications," *arXiv preprint arXiv:1302.0974*, 2013.

[25] G. H. Golub and C. F. V. Loan., *Matrix Computations*. The Johns Hopkins University Press, 1996.

[26] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, pp. 1–122, 2011.

[27] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in optimization*, vol. 1, no. 3, pp. 123–231, 2013.

[28] D. P. Bertsekas, *Nonlinear programming*. Athena Scientific, 1999.

[29] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1126–1153, 2013.

[30] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM Journal on imaging sciences*, vol. 6, no. 3, pp. 1758–1789, 2013.

[31] ——, "A globally convergent algorithm for nonconvex optimization based on block coordinate update," *arXiv preprint arXiv:1410.1386*, 2014.

[32] M. Faruqui and C. Dyer, "Community evaluation and exchange of word vectors at wordvectors.org," in *Proc. 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Baltimore, USA: Association for Computational Linguistics, June 2014.

[33] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.