
Article:
Zhou, Y., Lu, H. orcid.org/0000-0002-0349-2181 and Cheung, Y.-M. (2019) Probabilistic rank-one tensor analysis with concurrent regularizations. IEEE Transactions on Cybernetics. ISSN 2168-2267

https://doi.org/10.1109/TCYB.2019.2914316

Probabilistic Rank-One Tensor Analysis with Concurrent Regularizations

Yang Zhou, Haiping Lu, *Member, IEEE,* and Yiu-ming Cheung, *Fellow, IEEE*

***Abstract*—Subspace learning for tensors attracts increasing interest in recent years, leading to the development of multilinear extensions of Principal Component Analysis (PCA) and Probabilistic PCA (PPCA). Existing multilinear PPCAs are based on the Tucker or CANDECOMP/PARAFAC (CP) models. Although both kinds of multilinear PPCAs have shown their effectiveness in dealing with tensors, they also have their own limitations. Tucker-based multilinear PPCAs have a restrictive subspace representation and suffer from rotational ambiguity, while CP-based ones are more prone to overfitting. To address these problems, we propose *Probabilistic Rank-One Tensor Analysis* (PROTA), a CP-based multilinear PPCA. PROTA has a more flexible subspace representation than Tucker-based PPCAs, and avoids rotational ambiguity. To alleviate overfitting for CP-based PPCAs, we propose two simple and effective regularization strategies, named as *concurrent regularizations*. By adjusting the noise variance or the moments of latent features, our strategies concurrently and coherently penalize the whole subspace. This relaxes unnecessary scale restrictions and gains more flexibility in regularizing CP-based PPCAs. To take full advantage of the probabilistic framework, we further propose a Bayesian treatment of PROTA, which achieves both automatic feature determination and robustness against overfitting. Experiments on synthetic and real-world datasets demonstrate the superiority of PROTA in subspace estimation and classification, as well as the effectiveness of concurrent regularizations in alleviating overfitting.**

I. Introduction

Multiway or multidimensional arrays, a.k.a. tensors, are abundant in real-world applications, such as signal processing, computer vision, social network analysis, etc. [1]–[3]. The order of a tensor is the number of dimensions of the array, and a mode is one dimension of it. For example, a gray-level image can be represented by a second-order tensor (matrix) with the dimensions of *height* $\times$ *width*, and a gait silhouette sequence can be organized as a third-order tensor of *height* $\times$ *width* $\times$ *time*. By preserving the structural information in each mode, tensors can naturally characterize data from multiple aspects, providing compact and meaningful representations. Tensorial data are typically high-dimensional, and difficult to be directly handled in their original space. In addition, interesting latent information or interactions among multiple modes often lie in a low-dimensional subspace [4]. Therefore, subspace learning, as a useful technique for dimensionality reduction, is frequently used to represent high-dimensional tensors in a low-dimensional subspace without losing much useful underlying information or structures.

Principal Component Analysis (PCA) [5] is one of the most popular subspace learning techniques. It aims to find a subspace that preserves maximum data variance. In the past few decades, many PCA extensions have been proposed. Among them, one important and fundamental representative is *Probabilistic PCA* (PPCA) [6]. PPCA reformulates PCA under the probabilistic framework by learning a generative model that relates low-dimensional latent features with high-dimensional observations. In this way, PPCA obtains two main advantages over PCA: 1) It can capture data uncertainty and handle missing values; 2) It enables automatic model selection or incorporation of certain desirable properties such as robustness [7], sparsity [8], and large-margin separability [9].

Although PCA and PPCA have wide applications, they have limitations in dealing with *tensors*. Since PCA and PPCA can only take *vectors* as inputs, they have to vectorize or reshape tensors into vectors first. This breaks the meaningful tensor structures, and leads to larger parameter sizes and higher memory demands [10]. To address these problems, two kinds of multilinear PCA extensions have been proposed, which learn subspaces directly from tensorial inputs for preserving structural information. One is based on the Tucker model [11] that projects high-dimensional *tensors* into low-dimensional *tensors* [12]–[16]. The other is based on the CANDECOMP/PARAFAC (CP) model [17], [18] that projects high-dimensional *tensors* into low-dimensional *vectors* [19]–[21].

Along this line, several multilinear *PPCA* extensions have been proposed to take advantages of both probabilistic models and tensor representations. Most of them are based on the Tucker model. For example, Matrix-Variate Factor Analysis (MVFA) [22] attempts to extend PPCA for matrix inputs. It constructs a bilinear Tucker model to relate each matrix observation to a low-dimensional latent matrix via column and row factor matrices. Probabilistic Second-Order PCA (PSOPCA) [23] provides a probabilistic interpretation of bilinear PCAs by employing *matrix-variate normal* distributions [24] and variational approximation techniques. Bilinear Probabilistic PCA (BPPCA) [25] further adds two extra noise terms into the PSOPCA model. This leads to tractable probability density functions and closed-form updates for maximum likelihood estimation (MLE).

Compared with Tucker-based approaches, CP-based PPCAs are relatively under-developed. To the best of our knowledge, Tensor Bayesian Vectorial Dimension Reduction (TBVDR)

Yang Zhou and Yiu-ming Cheung are with the Department of Computer Science, Hong Kong Baptist University, Hong Kong (e-mail: yangzhou@comp.hkbu.edu.hk, ymc@comp.hkbu.edu.hk). Yiu-ming Cheung is the corresponding author.

Haiping Lu is with the Department of Computer Science, University of Sheffield, UK (e-mail: h.lu@sheffield.ac.uk).

[26] is the only existing CP-based multilinear PPCA. It introduces an additional linear projection into the CP model, so that the model complexity and the number of extracted features can be controlled separately. There are also several related works on probabilistic/Bayesian CP decomposition (CPD), which were developed for tensor completion but can be applied to subspace learning. Bayesian Probabilistic Tensor Factorization (BPTF) [27] formalizes the collaborative filtering problem as a CPD with time factors and smooth constraints for capturing temporal correlations. It is further extended to a parameter-free Bayesian version to automatically control the model complexity. Bayesian CP Factorization (BCPF) [28] applies automatic relevance determination (ARD) [29], [30] for CPD, so that the CP rank can be determined automatically. Variational Bayesian Tensor CP decomposition (VBTCP) [31] extends BCPF to deal with noisy complex-valued tensors, and imposes orthogonal constraints on one or more dimensions.

Although both Tucker- and CP-based multilinear PPCAs have shown their effectiveness in dealing with tensors, they have their own limitations. Tucker-based approaches suffer from *rotational ambiguity* [6], [32], in the sense that their solutions with and without rotation transformations are equally good, and have a *compact yet restrictive* subspace representation. On the other hand, CP-based ones are more flexible in representing subspaces without rotational ambiguity, whereas they are more *prone to overfitting*, leading to poor generalization abilities. A few regularization strategies have been studied in Bayesian CPD methods for alleviating overfitting. However, they are designed for tensor completion, taking no prior knowledge of subspace learning into account and introducing strong restrictions into the CP model.

To address the above problems, we propose **P**robabilistic **R**ank-**O**ne **T**ensor **A**nalysis (PROTA) with *concurrent regularizations*. Our contributions are three-fold:

- We propose PROTA, a new CP-based multilinear PPCA, which represents each observation as a linear combination of rank-one tensors. Compared with Tucker-based PPCAs, PROTA is more flexible in capturing data characteristics, and avoids rotational ambiguity. Its advantages over existing CP-based PPCAs are described in the next contribution.
- To alleviate overfitting for CP-based PPCAs, we propose two simple and effective regularization strategies in PROTA, named as *concurrent regularizations*, where we control the model complexity by adjusting the *noise variance* or the *moments* of latent features. Different from existing Bayesian CPDs that penalize each factor *independently*, we make use of the group-wise scale invariance of the CP model to *concurrently and coherently* regularize the *whole subspace*, while keeping the latent features *unconstrained*. As a result, our new regularizations avoid imposing unnecessary restrictions, leading to a more flexible and effective way of regularizing CP-based PPCAs.
- To fully utilize the probabilistic framework, we recast the idea of *whole subspace regularization* as prior distributions, and further propose a Bayesian treatment of PROTA, along with model estimation schemes via variational inference. It inherits both the ability of Bayesian CPD methods in automatically pruning irrelevant features and the robustness of concurrent regularizations against overfitting.

TABLE I
CONVENTION OF NOTATIONS.

Notation	Description
$\mathbf{z}_m$	the mth latent vector
$\mathcal{X}_m$	the mth observed tensor
I_n	the mode-n dimension of observed tensors
$\mathbf{X}_{m(n)}$	the mode-n unfolding of $\mathcal{X}_m$
$\mathbf{U}^{(n)}$	the mode-n factor matrix
$\mathbf{U}^{(n^-)}$	the mode-n complement factor matrix with $\mathbf{U}^{(n^-)} = \mathbf{U}^{(N)} \odot \ldots \odot \mathbf{U}^{(n+1)} \odot \mathbf{U}^{(n-1)} \odot \ldots \odot \mathbf{U}^{(1)}$
$\text{vec}(\mathcal{X}_m)$	the vector stacked by the columns of $\mathcal{X}_m$
$\text{diag}(\mathcal{X}_m)$	the vector formed by the diagonal elements of $\mathcal{X}_m$
$\text{diag}^N(\mathbf{z}_m)$	the Nth order diagonal tensor formed by $\mathbf{z}_m$
$\circ$	the outer product
$\otimes$	the Kronecker product
$\circledast$	the Hadamard (entrywise) product
$\odot$	the Khatri-Rao (column-wise Kronecker) product

We presented a preliminary work called Probabilistic Rank-One Matrix Analysis (PROMA) only for second-order tensors in [33]. This paper differs from [33] in three aspects:

1) *Generalized model*: We generalize PROMA to PROTA for dealing with higher-order tensors.
2) *New regularization strategy*: We propose a new concurrent regularization strategy, which is more effective in alleviating overfitting than the one proposed in [33].
3) *Bayesian extension*: We recast the new regularization into a prior distribution, and further propose a Bayesian extension of PROTA for both robustness against overfitting and automatic feature determination.
4) *Additional experiments*: We conduct additional experiments on both 2D and 3D real-world datasets.

II. PRELIMINARIES

This section introduces basic multilinear notations and operations used in this paper, and provides a brief review on PPCA and its multilinear extensions.

A. Notations and Multilinear Operations

Vectors are denoted by bold lowercase letters ($\mathbf{x}$). Matrices are denoted by bold uppercase letters ($\mathbf{X}$). Tensors are denoted by calligraphic letters ($\mathcal{X}$). The transpose of a vector or matrix is denoted by $\cdot^\top$. Symbols $\circ$, $\otimes$, $\circledast$, and $\odot$ denote the outer, Kronecker, Hadamard (entrywise), and Khatri-Rao (column-wise Kronecker) products, respectively[1]. $\langle\cdot\rangle$ denotes the expectation w.r.t. a certain distribution. $\text{vec}(\cdot)$ is the vectorization operator that turns a tensor into a column vector. For a vector $\mathbf{x}$, $\text{diag}^N(\mathbf{x})$ is the Nth order diagonal tensor formed by $\mathbf{x}$. For a matrix $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$, $\text{tr}(\mathbf{X})$ is its matrix trace. $\text{Ga}(x|a, b)$

[1]Please refer to Sec. 12.3 in [34] and Sec. 2.6 in [35] for the formal definitions and their relationships.

denotes the *Gamma distribution* with the hyper-parameters a and b. Table I summarizes the notations used in this paper.

Matrix-Variate Normal Distribution [24]: A random matrix $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$ that follows the matrix-variate normal distribution $\mathcal{N}_{I_1,I_2}(\mathbf{X}|\mathbf{\Xi}, \mathbf{\Sigma}_1, \mathbf{\Sigma}_2)$ with the mean matrix $\mathbf{\Xi}$, column covariance matrix $\mathbf{\Sigma}_1 \in \mathbb{R}^{I_1 \times I_1}$, and row covariance matrix $\mathbf{\Sigma}_2 \in \mathbb{R}^{I_2 \times I_2}$, has the following probability density function:

$$p(\mathbf{X}) = (2\pi)^{-\frac{1}{2}I_1 I_2} |\mathbf{\Sigma}_1|^{-\frac{1}{2}I_2} |\mathbf{\Sigma}_2|^{-\frac{1}{2}I_1} \exp\left\{ -\frac{1}{2} \text{tr}\left(\mathbf{\Sigma}_1^{-1} (\mathbf{X} - \mathbf{\Xi}) \mathbf{\Sigma}_2^{-1} (\mathbf{X} - \mathbf{\Xi})^\top \right) \right\}.$$

The matrix-variate normal distribution is related to the multivariate normal distribution in the following way: $p(\mathbf{X}) = \mathcal{N}_{I_1,I_2}(\mathbf{X}|\mathbf{\Xi}, \mathbf{\Sigma}_1, \mathbf{\Sigma}_2)$ if and only if $p(\text{vec}(\mathbf{X})) = \mathcal{N}(\text{vec}(\mathbf{X})|\text{vec}(\mathbf{\Xi}), \mathbf{\Sigma}_2 \otimes \mathbf{\Sigma}_1)$. $\mathcal{N}(\text{vec}(\mathbf{X})|\text{vec}(\mathbf{\Xi}), \mathbf{\Sigma}_2 \otimes \mathbf{\Sigma}_1)$ denotes a multivariate normal distribution, whose mean and covariance matrix are given by $\text{vec}(\mathbf{\Xi})$ and $\mathbf{\Sigma}_2 \otimes \mathbf{\Sigma}_1$, respectively.

For an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$, it is addressed by N indices $\{i_n\}_{n=1}^N$. Each i_n addresses the mode-n of $\mathcal{X}$.

Mode-n unfolding: $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times (I_1 \times \ldots \times I_{n-1} \times I_{n+1} \times \ldots \times I_N)}$ denotes the mode-n unfolding matrix of $\mathcal{X}$, where each column of $\mathbf{X}_{(n)}$ is a I_n-dimensional mode-n vector of $\mathcal{X}$.

Mode-n product: $\mathcal{Y} = \mathcal{X} \times_n \mathbf{U}^{(n)} \in \mathbb{R}^{I_1 \times \ldots \times P_n \times \ldots \times I_n}$ denotes the mode-n product of $\mathcal{X}$ by a matrix $\mathbf{U}^{(n)} \in \mathbb{R}^{J_n \times I_n}$, whose entries are given by:

$$\mathcal{Y}(i_1, \ldots, j_n, \ldots, i_N) = \sum_{i_n=1}^{I_n} \mathcal{X}(i_1, \ldots, i_N) \cdot \mathbf{U}^{(n)}(j_n, i_n).$$

Multilinear product: The multilinear product of $\mathcal{X}$ by N matrices $\{\mathbf{U}^{(n)} \in \mathbb{R}^{J_n \times I_n}\}_{n=1}^N$ is denoted by

$$\mathcal{Y} = \mathcal{X} \times_1 \mathbf{U}^{(1)} \times \ldots \times_N \mathbf{U}^{(N)} = \mathcal{X} \times_{n=1}^N \mathbf{U}^{(n)}.$$

B. Probabilistic PCA

Classical PPCA method is designed only for vector inputs. It learns a subspace from high-dimensional observed vectors by estimating the following latent variable model:

$$\mathbf{x} = \mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^I$ is the observation, $\mathbf{z} \in \mathbb{R}^P$ with $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$ is the latent variable that serves as the low-dimensional representation of $\mathbf{x}$, $\mathbf{I}$ is the identity matrix with an appropriate size, $\mathbf{W} \in \mathbb{R}^{I \times P}$ is the factor loading matrix that spans the P-dimensional latent subspace, $\boldsymbol{\epsilon} \in \mathbb{R}^P$ with $p(\boldsymbol{\epsilon}) = \mathcal{N}(\boldsymbol{\epsilon}|\mathbf{0}, \sigma^2\mathbf{I})$ is the random noise with the variance σ^2, and $\boldsymbol{\mu}$ is the mean vector.

With the above model, PPCA generalizes PCA to take advantage of the probabilistic framework. It also lays the foundations of probabilistic interpretations for other subspace learning techniques such as Linear Discriminant Analysis and Canonical Component Analysis [36]. Despite its success, PPCA still has some limitations. When the observations are *tensors*, PPCA has to first *reshape* them into vectors, which breaks the tensor structures and discards some useful data information.

C. Tucker-Based Multilinear PPCAs

To overcome the above limitation, several Tucker-based multilinear PPCAs [22], [23], [25] have been proposed. These methods directly formulate tensorial observations in the Tucker model without vectorization, so that the tensor structures can be preserved. Typically, they represent each Nth-order observed tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ as follows:

$$\mathcal{X} = \mathcal{Z} \times_{n=1}^N \mathbf{V}^{(n)^\top} + \Xi + \mathcal{E}, \tag{2}$$

where $\mathcal{Z} \in \mathbb{R}^{P_1 \times \ldots \times P_N}$ is the Nth-order low-dimensional latent tensor with $P_n \leq I_n$, $\mathbf{V}^{(n)} \in \mathbb{R}^{I_n \times P_n} = (\mathbf{v}_1^{(n)}, \ldots, \mathbf{v}_{P_n}^{(n)})$ is the mode-n factor matrix, Ξ is the mean tensor, and $\mathcal{E}$ is the random noise following $p(\text{vec}(\mathcal{E})) = \mathcal{N}(\text{vec}(\mathcal{E})|\mathbf{0}, \sigma^2\mathbf{I})$ with the noise variance σ^2.

Compared with PPCA, Tucker-based multilinear PPCAs have lower model complexity and a smaller parameter size. Specifically, to learn a $P = \prod_{n=1}^N P_n$-dimensional subspace from N-th order tensors $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$, they only need to estimate $\sum_{n=1}^N I_n P_n$ parameters for $\{\mathbf{V}^{(n)}\}_{n=1}^N$ rather than $P \cdot \prod_{n=1}^N I_n$ ones for $\mathbf{W}$ as in PPCA. However, as will be shown in the next section, such compact subspace representation is relatively restrictive and may limit the flexibility of Tucker-based PPCAs in capturing data characteristics.

D. CP-Based Multilinear PPCAs

CP-based multilinear PPCAs such as TBVDR [26] use the CP model for preserving the tensor structures. They have a more flexible subspace representation, whereas are more prone to overfitting than Tucker-based PPCAs. To alleviate overfitting, existing Bayesian CPD methods have studied several regularization strategies. However, these strategies are designed in the context of tensor completion. They bring strong restrictions into the CP model and can exclude good solutions for CP-based PPCAs. These issues (points) will be analyzed in detail when presenting PROTA in Sections III-B and III-E.

III. Probabilistic Rank-One Tensor Analysis

This section proposes PROTA with concurrent regularizations to address the problems of existing multilinear PPCAs. PROTA has both the flexible CP-based subspace representation and robustness against overfitting.

A. The PROTA Model

PROTA is based on the CP model. It relates each Nth-order observed tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ to a latent vector $\mathbf{z} \in \mathbb{R}^P$ by representing $\mathcal{X}$ as a linear combination of P rank-one tensors as follows [34], [35]:

$$\begin{aligned} \mathcal{X} &= \sum_{p=1}^P z_p \mathbf{u}_p^{(1)} \circ \mathbf{u}_p^{(2)} \circ \ldots \circ \mathbf{u}_p^{(N)} + \mathcal{E} \\ &= \text{diag}^N(\mathbf{z}) \times_{n=1}^N \mathbf{U}^{(n)^\top} + \mathcal{E}, \end{aligned} \tag{3}$$

where we have assumed that data are centered with zero mean, $\text{diag}^N(\mathbf{z}) \in \mathbb{R}^{P \times \ldots \times P}$ is the Nth-order diagonal tensor whose super-diagonal elements are given by $\mathbf{z}$ with $p(\mathbf{z}) =$

$\mathcal{N}(\mathbf{z}|\mathbf{0},\mathbf{I})$, $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times P} = (\mathbf{u}_1^{(n)}, \ldots, \mathbf{u}_P^{(n)})$ is the mode-n factor matrix, and $\mathcal{E}$ is the Nth-order noise tensor following $p(\text{vec}(\mathcal{E})) = \mathcal{N}(\text{vec}(\mathcal{E})|\mathbf{0}, \sigma^2\mathbf{I})$ with the variance σ^2.

Conditional distributions: Let $I = \prod_{n=1}^{N} I_n$ be the number of features in $\mathcal{X}$. By vectorizing the both sides of (3) with $\text{vec}(\mathbf{u}_p^{(1)} \circ \mathbf{u}_p^{(2)} \circ \ldots \circ \mathbf{u}_p^{(N)}) = \mathbf{u}_p^{(N)} \otimes \mathbf{u}_p^{(N-1)} \otimes \ldots \otimes \mathbf{u}_p^{(1)}$, we have $\text{vec}(\mathcal{X}) = \sum_{p=1}^{P} z_p \mathbf{u}_p^{(N)} \otimes \mathbf{u}_p^{(N-1)} \otimes \ldots \otimes \mathbf{u}_p^{(1)} + \text{vec}(\mathcal{E})$, and obtain the conditional distribution $p(\mathcal{X}|\mathbf{z})$ in a vectorized form as follows:

$$p(\text{vec}(\mathcal{X})|\mathbf{z}) = \mathcal{N}(\text{vec}(\mathcal{X})|\mathbf{W}\mathbf{z}, \sigma^2\mathbf{I}), \tag{4}$$

where $\mathbf{W} \in \mathbb{R}^{I \times P} = (\mathbf{w}_1, \ldots, \mathbf{w}_P) = \mathbf{U}^{(N)} \odot \ldots \odot \mathbf{U}^{(1)}$ is the joint factor matrix, and $\mathbf{w}_p \in \mathbb{R}^I$ with $\mathbf{w}_p = \mathbf{u}_p^{(N)} \otimes \mathbf{u}_p^{(N-1)} \otimes \ldots \otimes \mathbf{u}_p^{(1)}$ is the pth column of $\mathbf{W}$.

Let $\mathbf{X}_{(n)}$ be the mode-n unfolding of $\mathcal{X}$ and $I^{(n^-)} = \prod_{k \neq n} I_k$. The CP model (3) can also be expanded along the nth mode (see Sec. 12.5.4 in [34] for more details). This leads to $p(\mathcal{X}|\mathbf{z})$ in a unfolded form as follows:

$$p(\mathbf{X}_{(n)}|\mathbf{z}) = \mathcal{N}_{I_n, I^{(n^-)}}(\mathbf{X}_{(n)}|\mathbf{U}^{(n)}\text{diag}(\mathbf{z})\mathbf{U}^{(n^-)\top}, \sigma\mathbf{I}, \sigma\mathbf{I}), \tag{5}$$

where $\mathbf{U}^{(n^-)} \in \mathbb{R}^{I^{(n^-)} \times P} = (\mathbf{u}_1^{(n^-)}, \ldots, \mathbf{u}_P^{(n^-)}) = \mathbf{U}^{(N)} \odot \ldots \odot \mathbf{U}^{(n+1)} \odot \mathbf{U}^{(n-1)} \odot \ldots \odot \mathbf{U}^{(1)}$ is the mode-n *complement* factor matrix.

Log-likelihood function: Combining (3) with the above probabilistic model specifications, we complete the PROTA model. Given the dataset of M tensorial examples $\{\mathcal{X}_m\}_{m=1}^{M}$, we can obtain the "complete-data" log-likelihood $\mathcal{L} = \sum_{m=1}^{M} \ln p(\mathbf{X}_{m(n)}, \mathbf{z}_m) = \sum_{m=1}^{M} (\ln p(\mathbf{X}_{m(n)}|\mathbf{z}_m) + \ln p(\mathbf{z}_m))$ from (5), where $\mathbf{X}_{m(n)}$ is the mode-n unfolding of $\mathcal{X}_m$, and $\mathbf{z}_m$ with $p(\mathbf{z}_m) = \mathcal{N}(\mathbf{z}_m|\mathbf{0}, \mathbf{I})$ is an example of the latent variable $\mathbf{z}$. Then, the MLE of the PROTA parameters $\boldsymbol{\theta} = \{\{\mathbf{U}^{(n)}\}_{n=1}^{N}, \sigma^2\}$ can be obtained by maximizing the posterior expectation of $\mathcal{L}$ (see the supplementary materials for detailed derivations):

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= \sum_{m=1}^{M} \langle \ln p(\mathbf{X}_{m(n)}|\mathbf{z}_m) + \ln p(\mathbf{z}_m) \rangle \\ &= -\sum_{m=1}^{M} \Big[\frac{I}{2} \ln \sigma^2 + \frac{1}{2} \langle \mathbf{z}_m^\top \mathbf{z}_m \rangle \\ &+ \frac{1}{2\sigma^2} \langle ||\mathbf{X}_{m(n)} - \mathbf{U}^{(n)}\text{diag}(\mathbf{z}_m)\mathbf{U}^{(n^-)\top}||_F^2 \rangle \Big] + \text{const.}\end{aligned} \tag{6}$$

B. Connections with Existing PPCAs

After formally presenting the PROTA model for general tensors, this section studies the connections between PROTA with other PPCAs. In what follows, different PPCA models are compared in a *typical scenario of subspace learning*, where the subspace dimensionality P is *predetermined*.

Connections with PPCA: Firstly, we explore the connections between PPCA and its multilinear extensions.

Proposition 1. *Given $P = \prod_{n=1}^{N} P_n$, the Tucker and CP models, (2) and (3), are equivalent to the PPCA model (1) with the factor matrices $\mathbf{W}^{Tucker} = \mathbf{V}^{(N)} \otimes \ldots \otimes \mathbf{V}^{(1)}$ and $\mathbf{W}^{CP} = \mathbf{U}^{(N)} \odot \ldots \odot \mathbf{U}^{(1)}$, respectively.*

Proof. The above conclusion can be drawn by vectorizing the Tucker and CP models, (2) and (3), and applying $\text{vec}(\mathcal{Z} \times_{n=1}^{N} \mathbf{V}^{(n)\top}) = (\mathbf{V}^{(N)} \otimes \ldots \otimes \mathbf{V}^{(1)})\mathbf{z}$ and $\text{vec}(\text{diag}^N(\mathbf{z}) \times_{n=1}^{N} \mathbf{U}^{(n)\top}) = (\mathbf{U}^{(N)} \odot \ldots \odot \mathbf{U}^{(1)})\mathbf{z}$, respectively. □

Proposition 1 implies that the PPCA model can be viewed as the Tucker and CP ones with specific parameterizations of the factor matrix $\mathbf{W}$. It also indicates that the subspaces learned by Tucker and CP-based multilinear PPCAs are spanned by the columns of $\mathbf{W}^{\text{Tucker}}$ and $\mathbf{W}^{\text{CP}}$, respectively.

Connections with Tucker-based PPCAs: The CP model is commonly considered as a special case of the Tucker one, where the core tensor $\mathcal{Z}$ in (2) is super-diagonal with $P = P_1 = \ldots = P_N$. However, we can view their relationships from an opposite perspective, when the CP and Tucker models are used to extract the same number of features with $P = \prod_{n=1}^{N} P_n$.

Theorem 1. *Given $P = \prod_{n=1}^{N} P_n$, the Tucker model (2) can be written as a special case of the CP model (3).*

Proof. By expanding the tensor multiplication, the Tucker model (2) can be rewritten in the following summation form:

$$\begin{aligned}\mathcal{X} &= \sum_{n=1}^{N} \left(\sum_{i_n=1}^{P_n} \mathcal{Z}(i_1, \ldots, i_N) \mathbf{v}_{i_1}^{(1)} \circ \ldots \circ \mathbf{v}_{i_N}^{(N)} \right) + \mathcal{E} \\ &= \text{diag}^N(\mathbf{z}) \times_{n=1}^{N} \hat{\mathbf{V}}^{(n)\top} + \mathcal{E},\end{aligned}$$

where $\hat{\mathbf{V}}^{(n)} \in \mathbb{R}^{I_n \times P}$ is constructed by $\frac{P}{P_n}$ *repeated* factors $\mathbf{v}_{i_n}^{(n)}$ $(i_n = 1, \ldots, P_n)$. Therefore, the Tucker model can be written as a CP model with the parameterized factor matrices $\{\hat{\mathbf{V}}^{(n)}\}_{n=1}^{N}$. □

Generalized subspace representation: Theorem 1 implies that the CP model is in fact more general than the Tucker one in the scenario of subspace learning. To make this clear, we discuss the Tucker and CP models with $N = 2$ in detail, while similar conclusions can be drawn for higher-order cases. Given $N = 2$ and $P = P_1 + P_2$, the Tucker model (2) becomes

$$\mathbf{X} = \sum_{i_1, i_2 = 1}^{P_1, P_2} Z_{i_1 i_2} \mathbf{v}_{i_1}^{(1)} \mathbf{v}_{i_2}^{(2)\top} + \mathbf{E} = \hat{\mathbf{V}}^{(1)}\ \text{diag}(\mathbf{z}) \hat{\mathbf{V}}^{(2)\top} + \mathbf{E}, \tag{7}$$

where $\hat{\mathbf{V}}^{(1)} = (\overbrace{\mathbf{v}_1^{(1)}, \ldots, \mathbf{v}_1^{(1)}}^{P_2}, \ldots, \overbrace{\mathbf{v}_{P_1}^{(1)}, \ldots, \mathbf{v}_{P_1}^{(1)}}^{P_2})$ and $\hat{\mathbf{V}}^{(2)} = (\overbrace{\mathbf{U}^r, \ldots, \mathbf{U}^r}^{P_1})$.

We can view (7) as a specific CP model (3) whose factor matrices $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ are given by P_2 and P_1 *repeated* $\mathbf{v}_{i_1}^{(1)}$ $(i_1 = 1, \ldots, P_1)$ and $\mathbf{v}_{i_2}^{(2)}$ $(i_2 = 1, \ldots, P_2)$, respectively. Combining (7) with Proposition 1, we have $\mathbf{W}^{\text{Tucker}} = \mathbf{V}^{(2)} \otimes \mathbf{V}^{(1)} = \hat{\mathbf{V}}^{(2)} \odot \hat{\mathbf{V}}^{(1)} = (\mathbf{v}_1^{(2)} \otimes \mathbf{v}_1^{(1)}, \mathbf{v}_2^{(2)} \otimes \mathbf{v}_1^{(1)}, \ldots, \mathbf{v}_{P_2}^{(2)} \otimes \mathbf{v}_1^{(1)}, \mathbf{v}_1^{(2)} \otimes \mathbf{v}_2^{(1)}, \ldots, \mathbf{v}_{P_2}^{(2)} \otimes \mathbf{v}_{P_1}^{(1)})$. This is a relatively restrictive subspace representation, since each column of $\mathbf{V}^{(n)}$ is *reused* to construct *multiple* subspace bases. For example, the first P_2 columns of $\mathbf{W}^{\text{Tucker}}$ can only capture some common information, since they are constructed by the same factor $\mathbf{v}_1^{(1)}$ and different $\mathbf{v}_{i_2}^{(2)}$s.

In contrast, the CP model (3) represents the latent subspace by $\mathbf{W}^{\text{CP}} = \mathbf{U}^{(2)} \odot \mathbf{U}^{(1)} = (\mathbf{u}_1^{(2)} \otimes \mathbf{u}_1^{(1)}, \cdots, \mathbf{u}_P^{(2)} \otimes \mathbf{u}_P^{(1)})$. Such subspace representation is much more flexible than its Tucker-based counterpart, since each subspace basis $\mathbf{u}_p^{(2)} \otimes \mathbf{u}_p^{(1)}$ ($p = 1, \cdots, P$) is allowed to be constructed by *distinct* pair of factors. Therefore, PROTA generalizes Tucker-based PPCAs and has more flexibility in capturing data characteristics. However, the generalized subspace representation also makes the CP model more prone to overfitting than the Tucker one, since it has more parameters to be estimated.

Avoided rotational ambiguity: Apart form the more flexible subspace representation, PROTA also puts an edge over Tucker-based PPCAs in learning subspaces *without rotational ambiguity*. It is well known that the Tucker model suffers from rotational ambiguity, whose solutions with and without rotation transformations are equally good in the sense of yielding the maximum likelihood [25]. This implies that Tucker-based PPCAs can only find *arbitrary bases* of the latent subspace. In contrast, PROTA is based on the CP model, whose solutions are unique up to rotation transformations. Formally, let $\hat{\mathbf{U}}^{(n)} \in \mathbb{R}^{I_n \times P}$ be the maximum likelihood solution in terms of $\mathcal{L}(\boldsymbol{\theta})$ (6). For an arbitrary orthogonal matrix $\mathbf{R} \in \mathbb{R}^{P \times P}$, the rotation transformation $\hat{\mathbf{U}}^{(n)}\mathbf{R}$ yields $\mathcal{L}(\hat{\mathbf{U}}^{(n)}\mathbf{R}) < \mathcal{L}(\hat{\mathbf{U}}^{(n)})$[2], and thus is not the maximum likelihood solution anymore. This means that PROTA can find the *exact coordinate axes* rather than just the subspace bases, which facilitates certain applications such as data interpretation and visualization.

Connections with CP-based PPCAs: To the best of our knowledge, TBVDR [26] is the only existing CP-based PPCA. It introduces an additional linear projection $\mathbf{W}_h \in \mathbb{R}^{P \times Q}$ into the CP model (3) and defines $\mathbf{z} = \mathbf{W}_h\mathbf{h}$, where $\mathbf{h} \in \mathbb{R}^Q \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ serves as the latent features. In this way, TBVDR can control the complexity of the CP model (reflected by P) and the number of the latent features Q *separately*. Such modification can be viewed as specifying $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{W}_h\mathbf{W}_h^\top)$, which is restrictive in capturing general data characteristics. Different from TBVDR, we simply model the latent features $\mathbf{z}$ as i.i.d. Gaussian without additional constraints. Instead, we impose proper regularizations on the factor matrices $\mathbf{U}^{(n)}$ to alleviate overfitting (see Section III-D). In addition, we further propose a Bayesian treatment of PROTA in Section III-E to achieve both automatic feature determination and robustness against overfitting.

C. ECM Algorithm for PROTA

This section develops an EM-type algorithm for estimating the PROTA parameters. Although it is intractable to maximize (6) w.r.t. all the factor matrices $\{\mathbf{U}^{(n)}\}_{n=1}^N$ simultaneously, it is easy to solve $\mathbf{U}^{(n)}$ of each mode sequentially provided that the others are fixed. We achieve this by using the expectation-conditional maximization (ECM) approach [37], which leads to both closed-form solutions and good convergence properties. The ECM algorithm consists of the **E**xpectation (E-step) and the **C**onditional **M**aximization (CM-step).

[2]For clarity, we omit the parameters other than $\mathbf{U}^{(n)}$, i.e., $\{\mathbf{U}^{(k)}\}_{k \neq n}$ and σ^2, in $\boldsymbol{\theta}$.

E-step: In this step, we calculate the expectations $\langle \mathbf{z}_m \rangle$ and $\langle \mathbf{z}_m\mathbf{z}_m^\top \rangle$ w.r.t. the posterior distribution $p(\mathbf{z}_m|\text{vec}(\mathcal{X}_m))$. Using Bayes's rule for Gaussian variables (see Sec. 2.3.3 of [38] for more details), we can derive $p(\mathbf{z}_m|\text{vec}(\mathcal{X}_m))$ from (4) as follows:

$$p(\mathbf{z}_m|\text{vec}(\mathcal{X}_m)) = \mathcal{N}(\mathbf{z}_m|\mathbf{M}^{-1}\mathbf{W}^\top\text{vec}(\mathcal{X}_m), \sigma^2\mathbf{M}^{-1}), \quad (8)$$

where $\mathbf{M} = \mathbf{W}^\top\mathbf{W} + \sigma^2\mathbf{I}$ is a $P \times P$ matrix. Then given the model parameters at the kth iteration $\boldsymbol{\theta}^{(k)}$, the expectations $\langle \mathbf{z}_m \rangle$ and $\langle \mathbf{z}_m\mathbf{z}_m^\top \rangle$ can be computed by:

$$\langle \mathbf{z}_m \rangle = \mathbf{M}^{-1}\mathbf{W}^\top\text{vec}(\mathcal{X}_m), \quad (9)$$

$$\langle \mathbf{z}_m\mathbf{z}_m^\top \rangle = \sigma^2\mathbf{M}^{-1} + \langle \mathbf{z}_m \rangle \langle \mathbf{z}_m \rangle^\top. \quad (10)$$

CM-step: In this step, we partition the model parameters $\boldsymbol{\theta}$ into three groups: $\mathbf{U}^{(n)}$, $\mathbf{U}^{(n^-)}$, and σ^2. Then we alternately maximize $\mathcal{L}(\boldsymbol{\theta})$ (6) w.r.t. each group of the parameters with the others fixed. With fixed $\mathbf{U}^{(n^-)}$ and σ^2, we can estimate $\mathbf{U}^{(n)}$ by solving $\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \mathbf{U}^{(n)}} = 0$ and obtain

$$\tilde{\mathbf{U}}^{(n)} = \left[\sum_{m=1}^M \mathbf{X}_{m(n)}\mathbf{U}^{(n^-)}\text{diag}(\langle \mathbf{z}_m \rangle)\right] \left[\sum_{m=1}^M \langle \mathbf{z}_m\mathbf{z}_m^\top \rangle \circledast {\mathbf{U}^{(n^-)}}^\top\mathbf{U}^{(n^-)}\right]^{-1}. \quad (11)$$

After estimating all the factor matrices ($n = 1, \ldots, N$), the noise variance σ^2 can be estimated by solving $\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \sigma^2} = 0$ with $\{\tilde{\mathbf{U}}^{(n)}\}_{n=1}^N$ fixed, leading to

$$\tilde{\sigma}^2 = \frac{1}{MI}\sum_{m=1}^M \left\{\text{tr}\left(\mathbf{X}_{m(n)}^\top\mathbf{X}_{m(n)}\right) - \text{tr}\left(\mathbf{X}_{m(n)}\mathbf{U}^{(n^-)}\,\text{diag}(\langle \mathbf{z}_m \rangle)\tilde{\mathbf{U}}^{(n)\top}\right)\right\}. \quad (12)$$

By alternating between the E-step and CM-step, we can find the MLE solutions for $\{\mathbf{U}^{(n)}\}_{n=1}^N$ and σ^2. Besides the closed-form updates, the ECM algorithm monotonically increases the log-likelihood (6) at each iteration, and achieves a provable convergence guarantee [37]. The detailed derivations for (11) and (12) can be found in the supplementary materials.

D. Concurrent Regularizations for CP-Based PPCAs

Next, we develop regularization strategies for PROTA to achieve robustness against overfitting.

1) L_2 regularization: A conventional way of regularizations is introducing certain regularization terms into the log-likelihood function (6). This leads to a *regularized CM-step* that gives preference to solutions with desirable properties. The most popular representative of this approach is L_2 regularization, which penalizes larger norms and enforces smoothness on the factor matrices. Specifically, it regularizes the log-likelihood function (6) as follows:

$$\mathcal{L}^{L_2}(\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}) - \gamma\sum_{n=1}^N \text{tr}(\mathbf{U}^{(n)}{\mathbf{U}^{(n)}}^\top) = \mathcal{L}(\boldsymbol{\theta}) - \gamma\sum_{p=1}^P\sum_{n=1}^N \|\mathbf{u}_p^{(n)}\|^2, \quad (13)$$

Algorithm 1 PROTA with variance-based CR

Input: Dataset $\{\mathcal{X}_m\}_{m=1}^M$, the number of extracted features P, and the regularization parameter γ.
Initialize $\{\mathbf{U}^{(n)}\}_{n=1}^N$ and σ^2 randomly, and normalize each column of $\mathbf{U}^{(n)}$ to have unit norm.
Set the noise variance $\sigma^2 = \gamma$.
repeat
  Compute $\langle \mathbf{z}_m \rangle$ and $\langle \mathbf{z}_m \mathbf{z}_m^\top \rangle$ via (9) and (10), respectively.
  for $n = 1$ to N **do**
    Update the mode-n factor matrices $\mathbf{U}^{(n)}$ via (11).
  end for
until convergence.
Output: The factor matrices $\{\mathbf{U}^{(n)}\}_{n=1}^N$.

where γ is the regularization parameter. By maximizing (13) w.r.t. $\mathbf{U}^{(n)}$, we can obtain the following regularized CM-step for each factor matrix:

$$\tilde{\mathbf{U}}^{(n)} = \left[\sum_{m=1}^{M} \mathbf{X}_{m(n)} \mathbf{U}^{(n^-)} \text{diag}(\langle \mathbf{z}_m \rangle)\right] \left[\sum_{m=1}^{M} \langle \mathbf{z}_m \mathbf{z}_m^\top \rangle \circledast {\mathbf{U}^{(n^-)}}^\top \mathbf{U}^{(n^-)} + \gamma \mathbf{I}\right]^{-1}, \tag{14}$$

where the L_2 regularization term $\gamma\mathbf{I}$ improves the conditioning of the inverse, and leads to more stable and robust solutions against overfitting.

2) Scale restriction: Although L_2 regularization has been widely used, it introduces strong scale restrictions into the CP model and is not flexible enough for regularizing PROTA. Recall that the subspace learned by PROTA is spanned by the columns of $\mathbf{W} = \mathbf{U}^{(N)} \odot \ldots \odot \mathbf{U}^{(1)}$. For better generalization, we eventually pursuit robust/smoothed estimations for the *whole subspace* $\mathbf{W}$ rather than the *individual factor matrices* $\mathbf{U}^{(n)}$. L_2 regularization gives preference to a smoothed $\mathbf{W}$ by *independently* restricting the norms of *all* the factors to be small. However, we could still obtain a smoothed $\mathbf{W}$ for the CP model even if certain factors $\mathbf{u}_p^{(n)}$ have large norms, since the log-likelihood (6) is invariant to the scale transformations $\mathbf{u}_p^{(n)} \mapsto s\mathbf{u}_p^{(n)}$, $\mathbf{u}_p^{(n^-)} \mapsto s^{-1}\mathbf{u}_p^{(n^-)}$ ($s \neq 0$). Therefore, L_2 regularization introduces strong scale restrictions into the CP model, and may exclude some good solutions in terms of (6). Can we relax such scale restrictions in regularizing PROTA?

3) Concurrent regularizations: To address the above problem, we propose two strategies, named as variance-based and moment-based *concurrent regularizations* (CRs), respectively. Our aim is to regularize the whole subspace in a concurrent and coherent way, so that the strong scale restrictions of L_2 regularization can be avoided.

Variance-based CR: PROTA can be implicitly regularized by adjusting the noise level of the CP model (3). Specifically, we replace the noise variance σ^2 by a *fixed* regularization parameter γ *without further updating*. Adjusting σ^2 to an appropriate level makes the bias-variance tradeoff for the CP model, and thus improves the generalization ability of PROTA. In more detail, variance-based CR regularizes the E-step for more robust expectation estimations. It solves the ill-conditioned problems of $\mathbf{M}^{-1}$ involved in computing $\langle \mathbf{z}_m \rangle$ via (9), and $\langle \mathbf{z}_m \mathbf{z}_m^\top \rangle$ via (10), as follows:

$$\mathbf{M} = \mathbf{W}^\top \mathbf{W} + \gamma \mathbf{I}. \tag{15}$$

In this way, we avoid directly restricting the scale of each factor $\mathbf{u}_p^{(n)}$, and regularize the whole subspace and the CP model concurrently. Algorithm 1 gives the pseudocode of PROTA with variance-based CR.

Moment-based CR: Besides variance-based CR that introduces *implicit* regularization via adjusting the noise variance σ^2, we propose moment-based CR to *explicitly* regularize the second-order moment $\langle \mathbf{z}_m \mathbf{z}_m^\top \rangle$ (10) as follows:

$$\langle \mathbf{z}_m \mathbf{z}_m^\top \rangle^{\text{MCR}} = \sigma^2 \mathbf{M}^{-1} + \langle \mathbf{z}_m \rangle \langle \mathbf{z}_m \rangle^\top + \frac{\gamma}{M} \mathbf{I}, \tag{16}$$

where the noise variance σ^2 still serves a model parameter to be estimated rather than the regularization parameter as in variance-based CR. Moment-based CR improves the conditioning of $\langle \mathbf{z}_m \mathbf{z}_m^\top \rangle$, and solves the possibly ill-posed inverse in the $\mathbf{U}^{(n)}$ update (11). To make this clear, substituting (16) into (11) leads to:

$$\tilde{\mathbf{U}}^{(n)} = \left[\sum_{m=1}^{M} \mathbf{X}_{m(n)} \mathbf{U}^{(n^-)} \text{diag}(\langle \mathbf{z}_m \rangle)\right] \left[\sum_{m=1}^{M} \langle \mathbf{z}_m \mathbf{z}_m^\top \rangle \circledast {\mathbf{U}^{(n^-)}}^\top \mathbf{U}^{(n^-)} + \gamma \mathbf{\Lambda}^{(n^-)}\right]^{-1}, \tag{17}$$

where $\mathbf{\Lambda}^{(n^-)} = \mathbf{I} \circledast ({\mathbf{U}^{(n^-)}}^\top \mathbf{U}^{(n^-)})$ is a $P \times P$ diagonal matrix whose pth diagonal element is the norm of the pth complement factor $||\mathbf{u}_p^{(n^-)}||^2$.

Similar to L_2 regularization, moment-based CR regularizes the log-likelihood function as follows:

$$\begin{aligned} \mathcal{L}^{\text{MCR}}(\boldsymbol{\theta}) &= \mathcal{L}(\boldsymbol{\theta}) - \gamma \sum_{n=1}^{N} \text{tr}(\mathbf{U}^{(n)} \mathbf{\Lambda}^{(n^-)} {\mathbf{U}^{(n)}}^\top) \\ &= \mathcal{L}(\boldsymbol{\theta}) - \gamma N \text{tr}(\mathbf{W}\mathbf{W}^\top) = \mathcal{L}(\boldsymbol{\theta}) - \gamma N \sum_{p=1}^{P} \prod_{n=1}^{N} \|\mathbf{u}_p^{(n)}\|^2. \end{aligned} \tag{18}$$

Compared (18) with (13), moment-based CR essentially penalizes the *whole subspace* $\mathbf{W}$ rather than each factor matrix $\mathbf{U}^{(n)}$. It also generalizes L_2 regularization by adopting $\mathbf{\Lambda}^{(n^-)}$ instead of an identity matrix to penalize each mode-n factor in a *weighted* manner. Moment-based CR not only favors *individual factors* $\mathbf{u}_p^{(n)}$ with smaller norms, but also those leading to smaller norms $||\mathbf{w}_p||^2 = \prod_{n=1}^N ||\mathbf{u}_p^{(n)}||^2 = ||\mathbf{u}_p^{(n)}||^2 ||\mathbf{u}_p^{(n^-)}||^2$ for each *subspace basis* $\mathbf{w}_p$. In this way, a mode-n factor $\mathbf{u}_p^{(n)}$ is allowed to have a relatively large norm as long as the norm of the corresponding subspace basis $\mathbf{w}_p$ is small.

In this way, moment-based CR relaxes the scale restrictions of L_2 regularization, allows PROTA to search larger solution space, and thus has potential to learn better subspaces. It is also worth noting that with the update of each factor matrix, the elements of $\mathbf{\Lambda}^{(n^-)}$ in (18) are also updated accordingly. This indicates that MCR adaptively adjusts its regularization strength to coherently regularize all the factor matrices in the sense of penalizing large $||\mathbf{w}_p||^2$. Because of the above

Algorithm 2 PROTA with moment-based CR

Input: Dataset $\{\mathcal{X}_m\}_{m=1}^M$, the number of extracted features P, and the regularization parameter γ.
Initialize $\{\mathbf{U}^{(n)}\}_{n=1}^N$ and σ^2 randomly, and normalize each column of $\mathbf{U}^{(n)}$ to have unit norm.
repeat
  Compute $\langle \mathbf{z}_m \rangle$ and $\langle \mathbf{z}_m \mathbf{z}_m^\top \rangle$ via (9) and (10), respectively.
  for $n = 1$ to N **do**
    Update the mode-n factor matrices $\mathbf{U}^{(n)}$ via (17).
  end for
  Update the noise variance σ^2 via (12).
until convergence.
Output: The factor matrices $\{\mathbf{U}^{(n)}\}_{n=1}^N$ and the noise variance σ^2.

mentioned benefits, MCR has an edge over L_2 regularization in alleviating overfitting for CP-based PPCAs.

Remarks: Different from variance-based CR that can be applicable for both Tucker-based and CP-based PPCAs, moment-based CR can only be applied to PROTA or other CP-based PPCAs, because its capability of whole subspace regularization relies on the *group-wise* scale invariance of the CP model. We provide the detailed derivations of (14) and (17) in the supplementary materials. Algorithm 2 gives the pseudocode of PROTA with moment-based CR.

E. PROTA with Bayesian CR

To fully utilize the probabilistic framework, we further propose a Bayesian treatment of PROTA, along with the model estimation schemes via variational inference. It is based on a probabilistic implementation of moment-based CR, and achieves automatic feature determination and robustness against overfitting.

1) Model Specification: **Prior distributions:** To regularize the whole subspace $\mathbf{W}$ in a Bayesian treatment, we recast *moment-based CR* as prior distributions, and specify them over each factor matrix $\mathbf{U}^{(n)}$ as follows:

$$\mathbf{U}^{(n)} \sim \prod_{p=1}^{P} \mathcal{N}(\mathbf{u}_p^{(n)} | \mathbf{0}, (\gamma \langle \tau \rangle \langle ||\mathbf{u}_p^{(n^-)}||^2 \rangle)^{-1} \mathbf{I}), \tag{19}$$

where γ is the regularization parameter, $\tau \equiv 1/\sigma^2$ is the precision (inverse of the noise variance), and $\langle \tau \rangle$ is the expectation obtained from the variational posterior $q(\tau)$ shown in (26).

The above prior distribution provides a probabilistic implementation of *moment-based CR*, which essentially leads to a similar likelihood function as (18). If $\langle ||\mathbf{u}_p^{(n^-)}||^2 \rangle$ becomes large, $\mathbf{u}_p^{(n)}$ tends to be small. When the inverse variance $\gamma \langle \tau \rangle \langle ||\mathbf{u}_p^{(n^-)}||^2 \rangle$ concentrates at large values, $\mathbf{u}_p^{(n)}$ is constrained to be zero. In this case, $\mathbf{u}_p^{(n)}$ and the corresponding latent feature have no effect on explaining the training data, and thus can be pruned from the PROTA model.

Recall that we have specified the latent feature $\mathbf{z}_m \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ without further constraints. To complete the Bayesian specification of the PROTA model, we introduce a conjugate (Gamma) prior over τ. Thus,

$$\tau \sim \text{Ga}(\tau | a_0, b_0), \tag{20}$$

where we follow the convention and set $a_0 = b_0 = 10^{-6}$ to obtain a broad and non-informative prior for τ.

Remarks: As in the ARD framework [30], a conjugate prior can also be specified over the regularization parameter γ so that γ can be optimized like other random variables. However, we find such optimization leads to overfitting in our empirical studies, as it only reflects which factors are relevant to fitting the *training set*. Therefore, we still leave γ as a hyper-parameter for improving the generalization ability.

Joint distribution: Let the dataset be $\mathcal{D} = \{\mathcal{X}_m\}_{m=1}^M$, and the variable set be $\mathbf{\Theta} = \big\{\{\mathbf{z}_m\}_{m=1}^M, \{\mathbf{U}^{(n)}\}_{n=1}^N, \tau\big\}$. Combining the conditional distribution (4) and the above priors, the complete PROTA model can be obtained by:

$$p(\mathcal{D}, \mathbf{\Theta}) = \prod_m \{p(\mathcal{X}_m | \mathbf{z}_m, \{\mathbf{U}^{(n)}\}, \tau) p(\mathbf{z}_m)\} \prod_n p(\mathbf{U}^{(n)}) p(\tau). \tag{21}$$

2) Variational Inference: Armed with the above results, the PROTA model can be learned by estimating the posterior distribution $p(\mathbf{\Theta}|\mathcal{D}) = \frac{p(\mathcal{D},\mathbf{\Theta})}{\int p(\mathcal{D},\mathbf{\Theta}) d\mathbf{\Theta}}$. Since $p(\mathbf{\Theta}|\mathcal{D})$ is generally intractable, we apply Variational Bayesian (VB) methods [39] for the model estimation. VB methods seek a variational distribution $q(\mathbf{\Theta})$ to approximate the true posterior by minimizing the KL divergence $\text{KL}(q(\mathbf{\Theta})||p(\mathbf{\Theta}|\mathcal{D})) = \ln p(\mathcal{D}) - \mathcal{L}(q)$ or equivalently maximizing the *variational lower bound* $\mathcal{L}(q) = \int q(\mathbf{\Theta}) \ln\{\frac{p(\mathcal{D},\mathbf{\Theta})}{q(\mathbf{\Theta})}\} d\mathbf{\Theta}$.

To achieve this, we assume that $q(\mathbf{\Theta})$ is factorized as:

$$q(\mathbf{\Theta}) = \prod_m q(\mathbf{z}_m) \prod_n q(\mathbf{U}^{(n)}) q(\tau). \tag{22}$$

Then, the optimal distribution of the jth parameter set in terms of $\max_{q_j(\mathbf{\Theta}_j)} \mathcal{L}(q)$ takes the following form:

$$\ln q_j(\mathbf{\Theta}_j) \propto \langle \ln p(\mathcal{D}, \mathbf{\Theta}) \rangle_{\mathbf{\Theta} \setminus \mathbf{\Theta}_j}, \tag{23}$$

where $\langle \cdot \rangle_{\mathbf{\Theta} \setminus \mathbf{\Theta}_j}$ denotes the expectation w.r.t. the variational distributions of all random variables in $\mathbf{\Theta}$ except $\mathbf{\Theta}_j$.

Variational posterior distributions: Substituting the joint distribution (21) into the explicit forms (23), we can obtain the desirable variational posterior distributions for each set of random variables in $\mathbf{\Theta}$ as follows:

$$q(\mathbf{z}_m) = \mathcal{N}(\mathbf{z}_m | \bar{\mathbf{z}}_m, \mathbf{\Sigma}_\mathbf{z}), \tag{24}$$
$$q(\mathbf{U}^{(n)}) = \mathcal{N}_{I_n, P_n}(\mathbf{U}^{(n)} | \bar{\mathbf{U}}^{(n)}, \mathbf{I}, \mathbf{\Sigma}^{(n)}), \tag{25}$$
$$q(\tau) = \text{Ga}(\tau | a_\tau, b_\tau), \tag{26}$$

where the posterior parameters can be updated by

$$\bar{\mathbf{z}}_m = \langle \tau \rangle \mathbf{\Sigma}_\mathbf{z} \langle \mathbf{W} \rangle^\top \text{vec}(\mathcal{X}_m), \tag{27}$$
$$\mathbf{\Sigma}_\mathbf{z} = \big(\langle \tau \rangle \langle \mathbf{W}^\top \mathbf{W} \rangle + \mathbf{I}\big)^{-1}, \tag{28}$$

Algorithm 3 PROTA with Bayesian CR

Input: Dataset $\{\mathcal{X}_m\}_{m=1}^M$, and the regularization parameter γ.
Initialize $\{\mathbf{U}^{(n)}\}_{n=1}^N$ and σ^2 randomly.
repeat
  Update the latent features $\mathbf{z}_m$ via (24).
  for $n = 1$ to N **do**
    Update the mode-n factor matrices $\mathbf{U}^{(n)}$ via (25).
  end for
  Update the precision τ via (26).
until convergence.
Output: The variational distributions (24), (25), (26).

$$\bar{\mathbf{U}}^{(n)} = \sum_{m=1}^{M} \mathbf{X}_{m(n)} \langle \mathbf{U}^{(n^-)} \rangle \text{diag}(\langle \mathbf{z}_m \rangle) \boldsymbol{\Sigma}^{(n)}, \tag{29}$$

$$\boldsymbol{\Sigma}^{(n)} = \{\langle \tau \rangle (\sum_{m=1}^{M} \langle \mathbf{z}_m \mathbf{z}_m^\top \rangle + \gamma \mathbf{I}) \circledast \langle {\mathbf{U}^{(n^-)}}^\top \mathbf{U}^{(n^-)} \rangle\}^{-1}, \tag{30}$$

$$a_\tau = a_0 + \frac{1}{2} M \prod_{n=1}^{N} I_n, \tag{31}$$

$$b_\tau = b_0 + \frac{1}{2} \sum_{m=1}^{M} \langle ||\text{vec}(\mathcal{X}_m) - \mathbf{W}\mathbf{z}_m||^2 \rangle. \tag{32}$$

The derivations of the joint distribution (21) and the expectations involved in the above variational updates can be found in the supplementary materials. Algorithm 3 shows the pseudo-code for PROTA with Bayesian CR.

Connections with Bayesian CPDs: PROTA also has close connections with Bayesian CPD methods [27], [28], [31], [40]. They are all based on the CP model and incorporate regularizations. However, PROTA tailors the CP model for multilinear subspace learning, and utilizes very distinct regularization strategies. Bayesian CPD methods adapt the CP model for tensor completion. They commonly assume that the latent features $\mathbf{z}$ and the factor matrices $\mathbf{U}^{(n)}$ play the same role in explaining tensor inputs, and regularize them *equally* and *independently*. Such assumption is reasonable for tensor completion, whereas could be too restrictive for other applications. For instance, many Bayesian CPD methods employ ARD for automatic CP rank determination. This in fact can be viewed as imposing L_2 regularization on both the factors and latent features with data-dependent regularization parameters. As discussed in Section III-E, such L_2 regularization brings strong scale restrictions into the CP model. In contrast, PROTA advocates that $\mathbf{U}^{(n)}$ needs proper regularizations while $\mathbf{z}$ should remain unconstrained. This motivates our concurrent regularizations to *concurrently* and *coherently* regularize the whole subspace, leading to a more flexible and effective way of regularizing CP-based PPCAs.

F. Algorithmic Issues

Initialization: For PROTA with variance- and moment-based CRs, the factor matrices $\{\mathbf{U}^{(n)}\}_{n=1}^N$ are randomly initialized by sampling from the standard uniform distribution. Then they are normalized to have unit column norms, which leads to good performance empirically. For PROTA with Bayesian CR, we randomly initialize $\mathbf{U}^{(n)}$ by sampling from $\mathcal{N}(0, 1)$. The noise variance σ^2 $(1/\tau)$ is initialized to be data variance for all the regularized PROTAs.

Prediction: With the learned PROTA model, we can project a high-dimensional tensor $\mathcal{X}$ into the low-dimensional latent subspace. This is achieved by computing the expectation of $\mathbf{z}$ w.r.t. $p(\mathbf{z}|\mathcal{X})$ (8) and (27) for the ECM-based and Bayesian PROTA, respectively.

Time complexity: Suppose the input dataset consists of M tensors $\{\mathcal{X}_m \in \mathbb{R}^{I_1 \times \ldots \times I_N}\}_{m=1}^M$. Let $I = \prod_{n=1}^N I_n$ be the number of input features, and P be the number of extracted features. ECM-based and Bayesian PROTA have comparable time complexity. At each iteration, they take $O(MIP^2)$ for expectation computations, $O(MIP)$ for (variational) parameter updates, and $O(P^3)$ for matrix inverse. Therefore, the overall time complexity of PROTA at each iteration is dominated by $O(MIP^2 + P^3)$, which is comparable with that of existing EM-based and Bayesian PPCAs.

IV. Experiments

This section evaluates the performance of PROTA in subspace estimation and classification on synthetic and real-world datasets.

A. Subspace Estimation on Synthetic Data

We first validate the capability of the PROTA model in subspace estimation *without regularization* on synthetic datasets. The synthetic tensors are generated from the CP model (3) as follows: M latent vectors $\{\mathbf{z}_m^* \in \mathbb{R}^{P^*}\}_{m=1}^M$ are drawn from a standard Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_{P^*})$, and N factor matrices $\{\mathbf{U}^{(n)*} \in \mathbb{R}^{I_n \times P^*}\}_{n=1}^N$ are constructed by drawing each row from $\mathcal{N}(\mathbf{0}, \mathbf{I}_{P^*})$. Then the observed tensors are generated by $\mathcal{X}_m = \text{diag}^N(\mathbf{z}_m) \times_{n=1}^N {\mathbf{U}^{(n)*}}^\top + \mathcal{E}$ for $m = 1, \ldots, M$, where $\mathcal{E}(i_1, \ldots, i_N) \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ is the i.i.d. random noise with the variance σ_ε^2.

In this experiment, we generate multiple 3D synthetic datasets under varying noise levels. Each dataset consists of $M = 1000$ examples of third-order ($N = 3$) tensors with the size of $10 \times 10 \times 10$ and the true dimensionality $P^* = 8$. Based on Proposition 1, such synthetic tensors lie in the subspace spanned by the columns of $\mathbf{W}^* = \mathbf{U}^{(N)*} \odot \ldots \odot \mathbf{U}^{(1)*}$. We use the *arc length distance* $||\boldsymbol{\beta}||_2$ between the estimated subspace $\mathbf{W}$ and the ground truth $\mathbf{W}^*$ as the criterion to measure the accuracy of subspace estimation. The pth element of $\boldsymbol{\beta}$ is given by $\arccos(\lambda_p)$, where λ_p is the pth largest singular value of $\mathbf{W}^\top \mathbf{W}^*$ [25].

Given the true dimensionality P^*, PROTA is compared with the competing multilinear PCAs and PPCAs: MPCA, TRDO, and TBVDR, as well as Bayesian CPDs: BCPF and VBTCP. Results of all the methods are averaged over 10 repetitions of the above data generations. To estimate the P^*-dimensional latent subspace, the *reduced* dimensions of each mode are set to $(P^*)^{\frac{1}{N}}$ for MPCA, and P^* for TROD, BCPF, TBTCP, TBVDR, and PROTA. In addition, to reduce the variability caused by random initializations, BCPF and PROTA are randomly initialized 10 times, and the subspace yielding the largest log-likelihood (or variational lower bound) is used for test.

TABLE II
AVERAGE ARC LENGTH DISTANCES AND RUNNING TIME ON 3D SYNTHETIC DATASETS UNDER VARYING NOISE LEVELS (**BEST**; SECOND BEST).

SNR	0 dB	10 dB	20 dB	50 dB	100 dB	Time (s)
MPCA	3.57±0.10	3.58±0.10	3.58±0.10	3.58± 0.10	3.58±0.10	2.76
TROD	1.78±0.15	1.60±0.44	1.60±0.43	1.61±0.43	1.61±0.43	1.52
BCPF	**0.23±0.20**	0.13±0.16	0.11±0.16	0.06±0.12	0.06±0.12	2.84
VBTCP	0.77±1.04	0.52±1.04	0.87±1.16	1.14±1.12	1.14±1.12	9.83
TBVDR	0.89±0.86	1.92±0.41	1.10±0.76	1.34±0.76	1.38±0.80	**0.52**
PROTA	0.69±0.76	**0.04±0.01**	**1.17e-2±0.42e-2**	**3.58e-4±1.15e-4**	**1.16e-6±0.38e-6**	1.82

TABLE III
CLASSIFICATION ACCURACIES (MEAN±STD.%) ON THE CMU PIE DATASET (**BEST**; SECOND BEST; COMPARABLE* BASED ON t-TEST WITH $p = 0.05$).

L	2	3	4	5	6	8	10	20
PCA	26.41±3.35	37.25±1.50	43.04±2.51	49.50±2.14	52.08±2.58	60.68±1.74	66.26±0.87	82.40±0.64
PPCA	24.41±2.14	38.00±0.94	45.48±1.82	51.24±0.93	55.54±0.99	64.25±1.25	69.82±0.48	86.66±0.92
MPCA	35.27±2.97	46.25±2.56	51.74±1.79	56.61±1.63	59.60±0.58	66.75±0.66	71.48±0.78	84.35±0.88
UMPCA	29.08±3.06	38.11±2.11	42.52±3.42	48.34±3.03	51.04±3.05	58.12±3.31	61.61±3.24	76.38±2.39
TROD	34.52±1.84	42.92±2.75	47.90±2.52	52.92±1.87	56.33±1.52	63.30±0.93	67.70±1.21	81.07±1.54
PSOPCA	31.09±2.27	39.21±1.91	45.79±1.76	52.38±1.14	56.60±1.28	63.99±1.09	68.76±1.22	84.37±0.97
PSOPCA$^{\text{VCR}}$	35.15±1.23	44.92±1.23	50.61±2.05	56.02±1.16	60.32±1.02	67.77±0.81	71.71±1.16	85.72±0.65
BPPCA	36.07±1.88	47.41±1.93	53.23±2.39	59.25±2.27	63.84±1.81	71.14±1.13	74.83±2.00	88.06±0.51
BPPCA$^{\text{VCR}}$	37.23±2.71	47.67±1.91	54.03±2.37	60.21±1.70	63.91±1.88	71.02±1.97	75.09±0.83	87.78±0.94
BCPF	32.21±1.30	43.30±2.07	50.70±1.87	57.74±1.64	61.83±0.91	69.77±0.67	74.83±0.61	81.27±1.10
VBTCP	35.50±2.25	47.46±2.30	54.20±2.64	59.75±2.38	61.96±1.82	61.42±3.08	65.05±4.97	77.52±4.54
TBVDR	36.45±1.29	45.33±1.00	50.88±1.44	55.23±0.99	59.20±1.06	66.63±1.07	71.51±0.84	87.78±0.90
TBVDR$^{\text{MCR}}$	35.53±1.10	44.28±0.97	51.26±1.45	56.26±1.02	60.09±0.70	67.34±1.04	72.21±0.82	87.87±0.86
PROTA^{L_2}	35.15±1.89	47.17±1.15	56.40±2.16	62.13±1.74	65.77±1.43	73.62±1.42	77.97±0.76	89.72±0.51
PROTA$^{\text{VCR}}$	42.23±1.73	53.70±1.71*	59.99±1.68	65.72±1.65*	69.07±1.23*	75.30±1.27*	79.12±0.92	89.38±0.61
PROTA$^{\text{MCR}}$	**44.28±1.94***	**54.67±1.76***	**61.07±1.40***	**66.03±0.93***	**69.55±1.40***	**76.16±1.02***	**80.18±0.87***	**90.54±0.68***
PROTA$^{\text{BCR}}$	40.61±1.84	51.78±1.71	58.48±1.21	64.07±1.17	68.16±1.04	74.85±1.32	78.51±1.01	90.02±0.69

Table II shows the average arc length distances and running time on the 3D synthetic datasets under varying noise levels. As can be seen, PROTA is as efficient as other tensor-based PPCAs. Moreover, it can accurately estimate the ground truth subspace when the noise level is low, and outperforms other methods in the noisy cases except SNR $=$ 0dB. This confirms the ability of PROTA in fitting the ideal data. Since MPCA is based on the Tucker model, it fails to perform well in learning the subspace generated from the CP model. On the other hand, BCPF, VBTCP, and TBVDR have the CP-based subspace representation and thus obtain better results. However, they tend to be trapped into local optimums when SNR becomes larger, and thus fail to accurately recovery the true subspace.

B. Classification on 2D Images

This section evaluates the classification performance of PROTA on two image datasets. The first one is a subset from the CMU PIE database [41]. It consists of 9,987 face images from 68 subjects, with seven poses (C05, C07, C09, C27, C29, C37, C11) of at most 45 degrees of pose variations, and under 21 illumination conditions (02 to 22). The second one is the COIL20 dataset [42]. It includes 1,440 images of 20 objects taken from 72 views varying at every five degrees of rotations. All face images are normalized to 32×32 graylevel pixels.

Algorithms and their settings: PROTA is compared against ***linear baselines***: PCA, PPCA; ***Tucker-based PCA***: MPCA [16]; ***CP-based PCAs***: TROD [19], UMPCA [20]; ***Tucker-based PPCAs***: PSOPCA, BPPCA; **Bayesian CPDs**: BCPF [28] and VBTCP [31]; and ***CP-based PPCA***: TBVDR [26]. BPPCA has both MLE and MAP implementations. Here, we follow the settings in [25] that apply the MLE-based one for classification. We test PROTA equipped with four regularization strategies including L_2 regularization, variance-based CR, moment-based CR, and Bayesian CR, which are denoted by the superscripts ^{L_2}, $^{\text{VCR}}$, $^{\text{MCR}}$, and $^{\text{BCR}}$, respectively. PROTA$^{\text{VCR}}$ for 2D tensors is the PROMA algorithm in [33]. For fair comparisons, we also test PSOPCA and BPPCA with variance-based CR, and TBVDR with moment-based CR.

Extracted feature numbers: We set PCA and MPCA to preserve 97% energy, after verifying that preserving more energy just leads to similar results. Up to 1023, 32, 961, and 961 features are tested for PPCA, UMPCA, PSOPCA, and BPPCA, respectively. They are the maximum numbers of features that can be extracted by these methods. TROD, BCPF, VBTCP, TBVDR, and PROTA are tested up to $P = 600$ features, since their maximum numbers of extracted features are not bounded by the input dimensionality.

Regularization parameters: For all the regularized methods except PROTA$^{\text{VCR}}$, we select the regularization param-

TABLE IV
CLASSIFICATION ACCURACIES (MEAN±STD.%) ON THE COIL20 DATASET (**BEST**; SECOND BEST; COMPARABLE* BASED ON t-TEST WITH $p = 0.05$).

L	2	3	4	5	6	7	8	10
PCA	73.84±1.68	78.22±2.46	81.30±1.94	85.16±1.55	86.98±1.79	88.32±1.46	89.60±1.84	92.13±1.12
PPCA	40.41±21.01	57.45±23.51	78.96±2.33	83.34±2.98	85.27±2.52	87.65±1.91	88.85±0.99	91.03±1.67
MPCA	73.86±2.06	77.56±1.90	80.37±1.94	83.63±1.12	86.44±1.59	87.07±1.44	88.64±1.77	90.69±1.21
UMPCA	**77.22±2.44***	81.22±2.55*	83.91±3.12	86.05±2.09	87.74±1.40	88.73±1.52	90.11±1.72	91.56±1.65
TROD	76.69±4.23*	81.65±4.11*	85.03±2.39	88.90±2.60	90.88±1.67	92.06±1.56	92.63±1.45	94.31±1.46
PSOPCA	42.41±1.84	47.16±2.02	50.30±1.42	53.40±1.57	56.05±0.92	57.35±0.57	58.98±1.75	62.31±1.44
PSOPCA$^{\text{VCR}}$	50.06±3.19	56.96±3.49	58.58±3.58	62.45±2.33	65.57±2.74	66.53±1.90	69.05±1.90	72.99±1.72
BPPCA	72.36±6.40*	81.65±3.56*	85.32±3.44*	88.67±2.24	90.30±1.59	90.79±2.90	92.25±1.94	93.39±1.30
BPPCA$^{\text{VCR}}$	72.49±6.39*	81.25±3.39*	85.33±3.79*	88.67±2.23	90.30±1.58	90.82±1.58	92.28±1.92	93.37±1.32
BCPF	68.38±2.91	72.75±2.82	75.01±2.82	77.97±1.10	80.59±2.69	82.25±2.09	83.59±0.71	85.01±1.93
VBTCP	67.04±5.16	72.64±3.18	74.65±2.16	79.19±3.08	81.58±3.33	83.04±2.48	85.54±1.38	87.75±1.68
TBVDR	65.16±2.05	69.92±3.67	70.90±1.99	73.61±2.73	75.40±1.98	75.62±1.83	77.54±0.82	79.97±0.98
TBVDR$^{\text{MCR}}$	65.96±2.23	72.25±2.87	75.16±1.91	78.76±0.81	80.28±2.29	81.51±1.63	83.39±1.14	85.28±1.31
PROTA^{L_2}	73.87±4.04	80.43±2.22	85.12±3.50*	88.04±2.17	91.91±1.61*	92.94±1.86*	**95.07±1.59***	**95.62±1.59***
PROTA$^{\text{VCR}}$	76.64±3.70*	82.25±3.17*	86.60±2.10*	89.92±2.00*	91.70±1.57*	92.52±1.18	93.59±1.05	94.74±1.38*
PROTA$^{\text{MCR}}$	77.11±2.65*	**82.50±2.62***	86.52±2.40*	**90.66±1.34***	**92.42±1.91***	**93.71±1.39***	94.79±1.16*	95.61±1.53*
PROTA$^{\text{BCR}}$	76.54±2.79*	82.14±2.36*	**87.00±2.57***	90.07±1.60*	92.14±1.39*	92.67±1.19	93.97±1.18*	95.30±1.43*

eters from $\{10^{-5}, 10^{-4}, \ldots, 10^{5}\}$, and then report the best results. For PROTA$^{\text{VCR}}$, we select the best parameter from $\{0.1\tilde{\sigma}^2, 0.5\tilde{\sigma}^2, \tilde{\sigma}^2, 2\tilde{\sigma}^2, 10\tilde{\sigma}^2\}$, where $\tilde{\sigma}^2$ is the noise variance learned by PROTA with $P = 1$ [33].

Iteration number and convergence criterion: The maximum iteration numbers for MPCA, TROD, and UMPCA are set to their default settings with up to $1, 10$, and 10 iterations, respectively. For probabilistic methods such as PPCA, PSOPCA, BPPCA, BCPF, VBTCP, TBVDR, and PROTA, we iterate them until convergence or 500 iterations, where we define a method converges if the relative change of the log-likelihood or the variational lower bound is smaller than 10^{-5}.

Experimental setup: Each dataset is randomly split into training and test sets so that each class has L images for training, and the rest for test. After subspace learning, we sort the extracted features based on their corresponding Fisher scores [43] in descending order. Then, different numbers of the extracted features (up to the maximums) are fed into the *nearest neighbor classifier* to obtain classification results. For each method and L, we report the best averaged classification accuracies over ten such random splits. The best and the second best results are highlighted to be **bold** and underlined, respectively. The comparable results in terms of t-test with a p-value of 0.05 are marked by *.

Results and analysis: Table III shows the classification accuracies on the CMU PIE dataset. As can be seen, PROTA$^{\text{MCR}}$ consistently achieves the best performance with statistical significance in all the cases. PROTA$^{\text{VCR}}$ is the second best method, and PROTA$^{\text{BCR}}$ obtains the third best overall results. BPPCA with variance-based CR (BPPCA$^{\text{VCR}}$) also performs reasonably well, whereas it is much worse than PROTA$^{\text{MCR}}$ by 5.69% on average. This could be attributed to not only the CP model in capturing data characteristics with more flexibility but also moment-based CR in alleviating overfitting. Although BCPF and VBTCP are also based on the CP model and impose regularizations, they perform much worse than PROTA. A possible reason could be that Bayesian CPD methods are not aware of the prior knowledge of subspace learning and introduce unnecessary restrictions into the CP model.

Table IV shows the classification results on the COIL20 dataset. Again, regularized PROTAs perform much better than the competing methods in most cases, while only PROTA$^{\text{MCR}}$ consistently obtains the top two results except $L = 4$. Among the competing methods, TROD obtains better results except $L = 2, 4$, while it is still worse than PROTA$^{\text{MCR}}$ by 1.4% on average. In addition, the best Tucker-based PPCAs, BPPCA and BPPCA$^{\text{VCR}}$, perform worse than CP-based methods such as TROD and PROTA on the whole, especially when L is large. This indicates that the Tucker model may not be flexible enough in learning subspaces on the COIL20 dataset.

In summary, PROTA outperforms the competing methods in most cases by taking advantages of both the CP model and concurrent regularizations. Among all the regularization strategies, moment-based CR is the best one, which achieves the top two performance in most cases. PROTA$^{\text{VCR}}$ and PROTA$^{\text{BCR}}$ are generally better than or at least comparable with PROTA^{L_2}. Specifically, PROTA$^{\text{MCR}}$ outperforms PROTA^{L_2} and PROTA$^{\text{VCR}}$ by 6.47% and 2.53% on average for all the 2D datasets, respectively. This demonstrates that by penalizing the whole subspace in a concurrent and coherent way, the moment-based CR relaxes unnecessary scale restrictions for the CP model, and could further improve the performance of PROTA.

Although PROTA$^{\text{BCR}}$ is a Bayesian extension of PROTA$^{\text{MCR}}$, it has to employ variational inference to approximate the true posterior for analytical tractability. This may lead to the degenerated performance of PROTA$^{\text{BCR}}$ on the CMU PIE dataset. Nevertheless, PROTA$^{\text{BCR}}$ still achieves similar performance with PROTA$^{\text{MCR}}$ on the COIL20 dataset. More importantly, as will be shown in Section IV-D, it can

TABLE V
GAIT RECOGNITION RESULTS (%) ON THE USF GAIT DATASET (**BEST**; SECOND BEST).

Recognition Type	Individual gait examples					Gait sequences				
Probe	A	B	C	D	E	A	B	C	D	E
PCA	49.79	44.68	27.38	18.18	16.78	76.06	70.73	53.66	26.87	25.58
PPCA	55.85	49.41	30.48	18.91	16.78	80.28	80.49	53.66	29.85	27.91
MPCA	54.75	50.35	34.29	18.91	18.16	84.51	80.49	60.98	28.36	23.26
UMPCA	26.82	23.17	14.29	4.99	5.06	57.75	58.54	31.71	10.45	11.63
TROD	57.77	48.94	33.57	18.18	17.24	90.14	75.61	63.41	28.36	25.58
PSOPCA	15.27	12.06	9.29	8.21	6.67	28.17	21.95	17.07	19.40	11.63
PSOPCA$^{\text{VCR}}$	37.55	22.46	15.71	10.85	9.89	66.20	36.59	24.39	20.90	20.93
BPPCA	62.04	54.14	37.14	20.38	19.54	84.51	78.05	58.54	**35.82**	27.91
BPPCA$^{\text{VCR}}$	60.94	53.19	36.67	19.94	18.16	**91.55**	**80.49**	**68.29**	29.85	23.26
BCPF	60.11	49.65	36.19	19.94	16.78	90.14	78.05	60.98	34.33	25.58
VBTCP	53.37	44.44	32.38	19.35	17.01	81.69	75.61	53.66	28.36	25.58
TBVDR	40.99	39.48	19.52	13.93	11.49	61.97	58.54	34.15	20.90	16.28
TBVDR$^{\text{MCR}}$	56.95	52.01	30.71	20.53	19.54	78.87	78.05	51.22	32.84	27.91
PROTA^{L_2}	55.16	45.15	32.38	17.89	17.70	84.51	73.17	51.22	34.33	**32.56**
PROTA$^{\text{VCR}}$	63.14	52.96	39.05	**21.99**	18.62	90.14	75.61	63.41	**35.82**	27.91
PROTA$^{\text{MCR}}$	**64.37**	**56.26**	37.62	20.82	**21.61**	**91.55**	78.05	58.54	**35.82**	30.23
PROTA$^{\text{BCR}}$	62.59	55.56	**39.29**	21.70	19.54	87.32	78.05	63.41	34.33	30.23

automatically determine the number of extracted features P, which is more convenient to use in practice than other regularized PROTAs.

C. Classification on 3D Sequences

This section evaluates PROTA on two 3D Sequences (third-order tensors) datasets. The first one is a subset of the USF gait challenge dataset [44]. Following the standard settings of gait recognition, we use the same gallery set with 731 examples of 71 subjects (classes) for training as in [20], and select the probes A (727 examples), B (423 examples), C (420 examples), D (682 examples), and E (435 examples) for test. So there is no random partitioning of the training and test sets for this dataset. All the gait examples are $32 \times 22 \times 10$ (binary) silhouette sequences.

The second one is the Cambridge-Gesture database [45], which consists of 900 image sequences of 9 hand gestures (classes). Each gesture class includes 100 examples from two subjects, under five illumination conditions, and with 10 motions. Following the same preprocessing steps in [46], we select the middle 32 frames from each sequence, and resize each image frame to 20×20, resulting in $20 \times 20 \times 32$ tensorial examples. For each gesture class, we randomly select L examples for training, and the rest for test. We report the best averaged results over ten such training/test partitions.

We apply the similar algorithmic settings in Section IV-B for PROTA and the competing methods. Since PSOPCA and BPCCA are bilinear approaches and cannot be directly applied to higher-order tensors, the tensorial examples are first unfolded along the *third* mode into matrices, so that they can be fed into PSOPCA and BPCCA. In addition to the recognition results of individual gait examples, we also report those of gait sequences for the USF gait dataset, following [44].

Results and analysis: Table V shows the gait recognition results on the USF gait dataset. For classifying individual gait examples, CR-based PROTAs achieve good overall performance, which demonstrates again the effectiveness of PROTA and concurrent regularizations. In contrast, PROTA^{L_2} obtains much worse results than other regularized PROTAs. This indicates that L_2 regularization could be too restrictive, and may exclude good solutions for PROTA. For classifying gait sequences, PROTA$^{\text{MCR}}$ obtains good overall results except on Probe C, and PROTA$^{\text{BCR}}$ is the second best method except on Probe A. BPPCA$^{\text{VCR}}$ outperforms others on Probes B and C. PSOPCA$^{\text{VCR}}$ and TBVDR$^{\text{MCR}}$ perform significantly better than their plain versions. These indicate that besides PROTA, concurrent regularizations are also effective in alleviating overfitting for other multilinear PPCAs.

Table VI shows the classification results on the Cambridge-Gesture dataset. Similar to the experiments on other datasets, PROTA$^{\text{VCR}}$ and PROTA$^{\text{MCR}}$ obtain the top two results with statistical significance in most cases. In more detail, PROTA$^{\text{MCR}}$ outperforms PROTA$^{\text{VCR}}$ and the best competing method by 0.9% and 3.14% on average, respectively. Among the competing methods, PPCA and MPCA achieve better overall performance, while the best Tucker-based PPCA, BPPCA, obtains poor results. This can be attributed to the limited flexibility of the Tucker model in capturing data characteristics as well as the broken tensor structures due to unfolding.

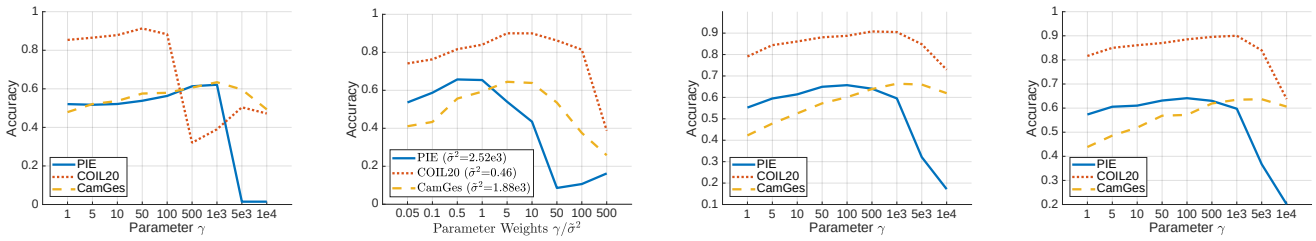
It is also worth noting that the performance of PSOPCA and BPPCA greatly depends on which mode is selected as the base dimension for unfolding. In our experiments, *the third mode*, the dimension of *time*, is the best choice for PSOPCA and BPPCA. However, if the input tensors are unfolded along other modes, PSOPCA and BPPCA can only obtain much worse results (about 10$\sim$20% lower than their best).

D. Parameter Sensitivity and Convergence Study

This section studies the parameter sensitivity and the convergence property of PROTA. We follow the same experimental settings in Section IV-B, and conduct experiments on both 2D

TABLE VI
CLASSIFICATION ACCURACIES (MEAN±STD.%) ON THE CAMBRIDGE-GESTURE DATASET (**BEST**; SECOND BEST; COMPARABLE* BASED ON t-TEST WITH $p = 0.05$).

L	5	10	15	20	25	30
PCA	29.53±2.31	39.75±3.62	46.60±2.45	51.36±3.00	56.58±2.99	58.38±3.31
PPCA	**43.86±2.75***	56.73±2.01	62.05±3.35	66.06±2.10	68.27±2.26	67.87±3.14
MPCA	41.38±6.14*	54.68±4.49	61.11±3.04	68.74±1.93	70.04±2.88	69.87±2.10
UMPCA	22.84±3.34	28.10±2.23	30.31±1.86	31.07±2.24	34.18±1.53	36.86±2.27
TROD	34.41±4.78	49.95±2.81	56.76±4.25	61.82±3.13	66.01±3.72	68.35±1.12
PSOPCA	29.08±3.15	40.16±2.41	44.63±3.41	50.04±3.42	55.56±2.05	55.81±3.21
PSOPCA$^{\text{VCR}}$	33.82±5.37	43.42±7.96	46.90±1.47	50.76±2.04	55.97±2.31	57.62±1.67
BPPCA	33.80±5.32	46.44±3.62	52.43±2.87	59.35±2.53	62.77±1.68	61.84±3.10
BPPCA$^{\text{VCR}}$	35.53±4.17	46.79±2.31	54.43±1.21	58.85±1.83	61.11±2.58	60.79±2.42
BCPF	31.35±3.55	40.60±2.75	46.63±2.28	52.13±2.60	55.51±3.05	58.68±1.44
VBTCP	31.27±2.98	42.15±4.67	35.92±5.63	40.85±5.20	37.11±13.53	38.44±5.84
TBVDR	32.83±3.02	46.28±3.53	52.93±2.71	58.29±3.29	62.50±1.80	63.19±2.24
TBVDR$^{\text{MCR}}$	37.31±2.29	49.49±2.76	55.24±3.35	60.22±1.67	63.85±1.84	64.21±2.22
PROTA^{L_2}	39.71±5.13	54.93±3.51	62.76±3.31	69.67±2.39*	70.40±1.58	72.90±2.10
PROTA$^{\text{VCR}}$	42.64±4.86*	59.07±3.37*	65.10±2.95*	69.74±3.13*	72.83±3.16*	75.35±2.38*
PROTA$^{\text{MCR}}$	43.77±5.47*	**59.85±3.82***	**65.32±2.54***	**71.32±1.82***	**73.63±1.40***	**76.24±1.92***
PROTA$^{\text{BCR}}$	39.85±4.78	56.80±2.39	62.97±3.09	69.38±2.07	73.48±1.53*	75.17±1.52*



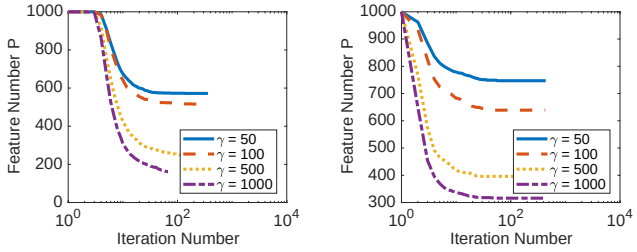
(a) PROTA^{L_2} (b) PROTA$^{\text{VCR}}$ (c) PROTA$^{\text{MCR}}$ (d) PROTA$^{\text{BCR}}$

Fig. 1. Classification results of regularized PROTAs with different parameter settings on 2D and 3D datasets.

(CMU PIE, COIL20) and 3D (Cambridge-Gesture) datasets. Since the USF gait dataset is constructed by fixed training and test sets without repeated random partitions, it is not included in this study for fair comparisons, while we have verified that the behavior of PROTA on the USF gait dataset is not much different from that on the other datasets. We report experimental results with moderate training sizes by setting $L = 5$ and $L = 15$ for the 2D and 3D datasets, respectively.

Parameter sensitivity: Firstly, we study how different values of the regularization parameters affect the performance of regularized PROTAs. Figure 1 illustrates the classification accuracies obtained by regularized PROTAs. At the beginning, the performance of PROTA consistently improves as the regularization parameters increase for all the datasets. This demonstrates that imposing regularization on PROTA is effective in alleviating overfitting.

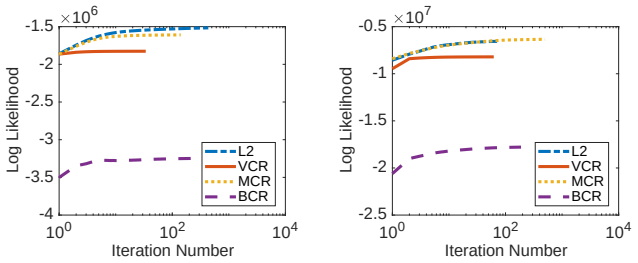
Among the four regularized PROTAs, PROTA$^{\text{MCR}}$ and PROTA$^{\text{BCR}}$ consistently achieve good performance on all the datasets when γ is around $100 \sim 1000$, and thus are less sensitive in terms of different parameter configurations and datasets. On the other hand, PROTA^{L_2} and PROTA$^{\text{VCR}}$ are more sensitive to the regularization parameters. Although the best value of γ^{VCR} varies a lot on different datasets, it is often close to $\tilde{\sigma}^2$, the noise variance learned by performing PROTA with $P = 1$. This suggests that plain PROTA (*without regularization*) could be used to roughly determine the regularization parameter for variance-based CR.



(a) CMU PIE (b) Cambridge-Gesture

Fig. 2. The number of features extracted by PROTA$^{\text{BCR}}$ at each iteration with different parameter settings on the CMU PIE and Cambridge-Gesture datasets.

Number of extracted features: We investigate the behavior of PROTA$^{\text{BCR}}$ in pruning irrelevant features. Figure 2 shows how the feature number P of PROTA$^{\text{BCR}}$ varies at each



(a) CMU PIE (b) Cambridge-Gesture

Fig. 3. Log-likelihood of regularized PROTAs at each iteration on the CMU PIE and Cambridge-Gesture datasets.

iteration given different values of γ^{BCR} on the CMU PIE and Cambridge-Gesture datasets. As can be seen, PROTA$^{\text{BCR}}$ prunes a large number of features after several iterations, indicating its ability of automatic feature determination. Since γ^{BCR} controls the range of variation that each subspace basis $\mathbf{w}_p$ can take, a larger γ^{BCR} will eliminate more features. Considering PROTA$^{\text{BCR}}$ is not sensitive to γ^{BCR} as shown in , it is relatively easy for PROTA$^{\text{BCR}}$ to determine an appropriate feature number with good performance.

Convergence: Finally, we study the convergence properties of regularized PROTAs by fixing $\gamma^{L_2} = 100$, $\gamma^{\text{VCR}}/\tilde{\sigma}^2 = 1$, $\gamma^{\text{MCR}} = 100$, and $\gamma^{\text{BCR}} = 100$ respectively. From Figure 1, such parameter settings yield reasonably good performance for all the datasets. Figure 3 shows the log-likelihood (or variational lower bound) of regularized PROTAs at each iteration on the CMU PIE and Cambridge-Gesture datasets. As can be seen, all PROTAs monotonically increase their objective functions and converge properly.

In addition, the behavior of PROTA is affected by the imposed regularization strategies. Moment-based CR leads to higher log-likelihood than the variance-based one, which suggests that PROTA$^{\text{MCR}}$ fits the PROTA model better and is less restrictive than PROTA$^{\text{VCR}}$. On the other hand, PROTA$^{\text{VCR}}$ converges faster than PROTA$^{\text{MCR}}$. This is because PROTA$^{\text{VCR}}$ has no need to estimate the noise variance σ^2 while fixing it to a relatively large value instead. By making the bias-variance tradeoff, a larger σ^2 improves the convergence speed of PROTA though at the expense of goodness-of-fit. For PROTA$^{\text{BCR}}$, the values of its objective function are smaller than those of other regularized PROTAs. This is expected because PROTA$^{\text{BCR}}$ aims at maximizing the *variational lower bound* rather than the log-likelihood.

V. Concluding Remarks

We have proposed PROTA, a new CP-based multilinear PPCA. Compared with Tucker-based PPCAs, PROTA has a more flexible subspace representation, and does not suffer from rotational ambiguity. Compared with existing CP-based PPCAs, our new concurrent regularizations penalize the whole subspace and avoid introducing unnecessary restrictions into the CP model, making PROTA more robust against overfitting. To fully utilize the probabilistic framework, we have further proposed a Bayesian treatment of PROTA, which achieves both automatic feature determination and robustness against overfitting. Experiments on both synthetic and real-world data have demonstrated the superiority of PROTA in subspace estimation and classification, as well as the effectiveness of concurrent regularizations in alleviating overfitting for PROTA and other multilinear PPCAs.

Besides the classical Tucker and CP models, recently some t-product based tensor decomposition models have been proposed [47]–[50], providing a new way of tensor analysis. By utilizing the new tensor multiplication, i.e., t-product, along with a newly defined tensor rank, they have obtained the state-of-the-art performance in many computer vision applications such as image denoising and background modeling. Despite of their success in image and video processing, we did not find any work for incorporating t-product based PCA models into the probabilistic framework yet, which could be an interesting future work.

References

[1] W. K. Wong, Z. Lai, Y. Xu, J. Wen, and C. P. Ho, "Joint tensor feature analysis for visual object recognition," *IEEE Trans. on Cybernetics*, vol. 45, no. 11, pp. 2425–2436, 2015.

[2] B. Jiang, C. Ding, J. Tang, and B. Luo, "Image representation and learning with graph-laplacian tucker tensor decomposition," *IEEE Trans. on Cybernetics*, vol. PP, no. 99, pp. 1–10, 2018.

[3] M. Pang, Y. ming Cheung, B. Wang, and R. Liu, "Robust heterogeneous discriminative analysis for face recognition with single sample per person," *Pattern Recognition*, vol. 89, pp. 91–107, 2019.

[4] X. Li, S. Lin, S. Yan, and D. Xu, "Discriminant locally linear embedding with high-order tensor data," *IEEE Trans. on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 2, pp. 342–352, 2008.

[5] I. T. Jolliffe, *Principal Component Analysis*. Springer Series in Statistics, second edition, 2002.

[6] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[7] T. Chen, E. Martin, and G. Montague, "Robust probabilistic PCA with missing data and contribution analysis for outlier detection," *Computational Statistics & Data Analysis*, vol. 53, no. 10, pp. 3706–3716, 2009.

[8] R. Khanna, J. Ghosh, R. Poldrack, and O. Koyejo, "Sparse submodular probabilistic PCA," in *Proc. of the 18th Int. Conf. on Artificial Intelligence and Statistics*, 2015, pp. 453–461.

[9] C. Du, S. Zhe, F. Zhuang, Y. Qi, Q. He, and Z. Shi, "Bayesian maximum margin principal component analysis," in *Proc. of 29th AAAI Conf. on Artificial Intelligence*, 2015, pp. 2582–2588.

[10] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, *Multilinear Subspace Learning: Dimensionality Reduction of Multidimensional Data*. CRC Press, 2013.

[11] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[12] J. Yang, D. Zhang, A. F. Frangi, and J. Yang, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131–137, 2004.

[13] J. Ye, "Generalized low rank approximations of matrices," *Machine Learning*, vol. 61, no. 1-3, pp. 167–191, 2005.

[14] J. Ye, R. Janardan, and Q. Li, "GPCA: An efficient dimension reduction scheme for image compression and retrieval," in *Proc. of ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2004, pp. 354–363.

[15] D. Xu, S. Yan, L. Zhang, S. Lin, H.-J. Zhang, and T. S. Huang, "Reconstruction and recognition of tensor-based objects with concurrent subspaces analysis," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 18, no. 1, pp. 36–47, 2008.

[16] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "MPCA: Multilinear principal component analysis of tensor objects," *IEEE Trans. on Neural Networks*, vol. 19, no. 1, pp. 18–39, 2008.

[17] J. D. Carroll and J. Chang, "Analysis of individual differences in multidimensional scaling via an N-way generalization of Eckart-Young decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[18] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, 1970.
[19] A. Shashua and A. Levin, "Linear image coding for regression and classification using the tensor-rank principle," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, vol. I, 2001, pp. 42–49.
[20] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Uncorrelated multilinear principal component analysis for unsupervised multilinear subspace learning," *IEEE Trans. on Neural Networks*, vol. 20, no. 11, pp. 1820–1836, 2009.
[21] M. Che and Y. Wei, "Randomized algorithms for the approximations of tucker and the tensor train decompositions," *Advances in Computational Mathematics*, vol. 45, no. 1, pp. 395–428, 2019.
[22] X. Xie, S. Yan, J. T. Kwok, and T. S. Huang, "Matrix-variate factor analysis and its applications," *IEEE Trans. on Neural Networks*, vol. 19, no. 10, pp. 1821–1826, 2008.
[23] S. Yu, J. Bi, and J. Ye, "Matrix-variate and higher-order probabilistic projections," *Data Mining and Knowledge Discovery*, vol. 22, no. 3, pp. 372–392, 2011.
[24] A. K. Gupta and D. K. Nagar, *Matrix Variate Distributions*. CRC Press, 1999, vol. 104.
[25] J. Zhao, P. L. H. Yu, and J. T. Kwok, "Bilinear probabilistic principal component analysis," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 23, no. 3, pp. 492–503, 2012.
[26] F. Ju, Y. Sun, J. Gao, Y. Hu, and B. Yin, "Vectorial dimension reduction for tensors based on bayesian inference," *IEEE Trans. on Neural Networks and Learning Systems*, 2017.
[27] L. Xiong, X. Chen, T. Huang, J. G. Schneider, and J. G. Carbonell, "Temporal collaborative filtering with Bayesian probabilistic tensor factorization," in *Proc. of SIAM Int. Conf. on Data Mining*, vol. 10. SIAM, 2010, pp. 211–222.
[28] Q. Zhao, L. Zhang, and A. Cichocki, "Bayesian CP factorization of incomplete tensors with automatic rank determination," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1751–1763, 2015.
[29] D. J. MacKay, "Bayesian interpolation," *Neural computation*, vol. 4, no. 3, pp. 415–447, 1992.
[30] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.
[31] L. Cheng, Y.-C. Wu, and H. V. Poor, "Probabilistic tensor canonical polyadic decomposition with orthogonal factors," *IEEE Trans. on Signal Processing*, vol. 65, no. 3, pp. 663–676, 2017.
[32] J. Ahn and J. Oh, "A constrained EM algorithm for principal component analysis," *Neural Computation*, vol. 15, no. 1, pp. 57–65, 2003.
[33] Y. Zhou and H. Lu, "Probabilistic rank-one matrix analysis with concurrent regularization." in *Proc. of the 25th Int. Joint Conf. on Artificial Intelligence*, 2016, pp. 2428–2434.
[34] G. H. Golub and C. F. van Loan, *Matrix Computations*, 4th ed. JHU Press, 2013.
[35] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.
[36] F. R. Bach and M. I. Jordan, "A probabilistic interpretation of canonical correlation analysis," University of California, Berkeley, Tech. Rep. TR 688, 2005.
[37] X. Meng and D. B. Rubin, "Maximum likelihood estimation via the ECM algorithm: A general framework," *Biometrika*, vol. 80, no. 2, pp. 267–278, 1993.
[38] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
[39] J. M. Winn and C. M. Bishop, "Variational message passing," *Journal of Machine Learning Research*, vol. 6, pp. 661–694, 2005.
[40] H. Shan, A. Banerjee, and R. Natarajan, "Probabilistic tensor factorization for tensor completion," Department of Computer Science and Engineering, University of Minnesota, Tech. Rep. TR 11-026, 2011.
[41] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression database," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1615–1618, 2003.
[42] S. A. Nene, S. K. Nayar, H. Murase *et al.*, "Columbia object image library (COIL-20)," Columbia University, Tech. Rep. CUCS-005-96, 1996.
[43] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. John Wiley & Sons, 2012.
[44] S. Sarkar, P. Phillips, Z. Liu, I. R. Vega, P. Grother, and K. W.Bowyer, "The human ID gait challenge problem: Data sets, performance, and analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 162–177, 2005.
[45] T.-K. Kim, S.-F. Wong, and R. Cipolla, "Tensor canonical correlation analysis for action classification," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.
[46] Y. M. Lui, J. R. Beveridge, and M. Kirby, "Action classification on product manifolds," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2010, pp. 833–839.
[47] N. Hao, M. E. Kilmer, K. Braman, and R. C. Hoover, "Facial recognition using tensor-tensor decompositions," *SIAM Journal on Imaging Sciences*, vol. 6, no. 1, pp. 437–463, 2013.
[48] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2016, pp. 5249–5257.
[49] C. Lu, J. Feng, y. chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis with a new tensor nuclear norm," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 1–14, 2019.
[50] C. Lu, J. Feng, Z. Lin, and S. Yan, "Exact low tubal rank tensor recovery from gaussian measurements," in *Proc. of the 27th Int. Joint Conf. on Artificial Intelligence*, 2018.