Causal Capsules and Tensor Autoencoders

M. Alex O. Vasilescu, maov@cs.ucla.edu
IPAM, University of California, Los Angeles
Tensor Vision Technologies, Los Angeles

ABSTRACT

This paper introduces a set of neural network architectures for forward and inverse causal inference that are consistent with capsule theory and implement multilinear (tensor) factor analysis methods. Forward causal inference is addressed with a causal autoencoder-decoder architecture composed of a set of causal capsules that estimate the latent variables representing the constituent factor of data formation, and a tensor-autoencoder that governs the latent variable interaction. A recurrent non-linear causal capsule chain that employs kernel activations computes the optimal linearized subspace for every causal factor, and implements the kernel multilinear principal component analysis or the kernel multilinear independent component analysis. For distributed computation, we break the chain links and each causal representation is computed separately, shuttling causal information between capsules. The causal factor representations may be computed efficiently by restructuring the input into a hierarchy of parts with a set of part-based causal capsules that are "glommed" together to create a part-based hierarchy of causal capsules. Inverse causal inference, the estimation of causes of effects, is addressed with a multilinear projection architecture that inverts the estimated forward causal model and employs a set of observations to constrain the solution set rendering the problem well-posed.

1 INTRODUCTION

Neural networks are being employed increasingly in high-stakes application areas, such as face recognition [Taigman et al. (2014); Huang (2012); Sun et al. (2013); Chen et al. (2015); Xiong et al. (2016)], and medical technologies [Kermany et al. (2018); Madani et al. (2018); Topol (2019)]. Developing a set of neural network architectures that are causally explainable is important in developing a trustworthy machine learning, where "A causes B" means "the effect of A is B", a measurable and experimentally repeatable quantity [Holland (1986)].

Forward causal inference models the mechanism of data formation, and estimates the effects of interventions [Pearl (2000); Imbens & Rubin (2015); Spirtes et al. (2000); Vasilescu et al. (2021); Vasilescu & Terzopoulos (2002a; 2005; 2004)]. Unlike, conventional statistics and conventional machine learning that model the observed data distribution, and make predictions about a variable that has been co-observed with another. Inverse causal inference estimates the causes of effects given an estimated forward causal model that is inverted subject to a set of observations that constrain the solution set [Vasilescu (2011); Vasilescu & Terzopoulos (2007)].

There are two conceptual frameworks for causal inference: DAGs or path analysis and potential-outcome. Donald Rubin and his collaborators have advocated the potential outcome approach which framed causal inference as a missing data problem [Imbens (2020)]. Judea Pearl has been advocating *do*-calculus – a directed acyclic graph approach as a mathematical language that he has unified with structural equations and counterfactuals [Bollen & Pearl (2013)]. Judea Pearl's causation ladder [Pearl (2000)] provides a way of thinking about causal discovery, causal reasoning, and decision making. Pearl & Bareinboim (2014); Bareinboim & Pearl (2016) have parameterized the differences between experimental and observational studies based on possible sources of error.

Tensor data analysis is a type of structural equation modeling that has been employed to perform dimensionality reduction, develop regression models, and to model cause-and-effect, Fig. 1. Tensor factor analysis has been employed in psychometrics [Tucker (1966); Harshman (1970); Carroll & Chang (1970); Bentler & Lee (1979); Kroonenberg & de Leeuw (1980)], econometrics [Kapteyn et al. (1986); Magnus & Neudecker (1988)], chemometrics [Bro (1997); Acar et al. (2014)], signal processing [de Lathauwer (1997; 2008); Cichocki et al. (2009)], computer vision [Vasilescu & Terzopoulos (2002b); Wang & Ahuja (2003)], computer graphics [Vasilescu (2002); Davis & Gao (2003); Vasilescu & Terzopoulos (2004); Vlasic et al. (2005); Hsu et al. (2005)], and machine learning [Vasilescu (2009); Vasilescu & Terzopoulos (2005)]. In machine learning, tensor methods have been

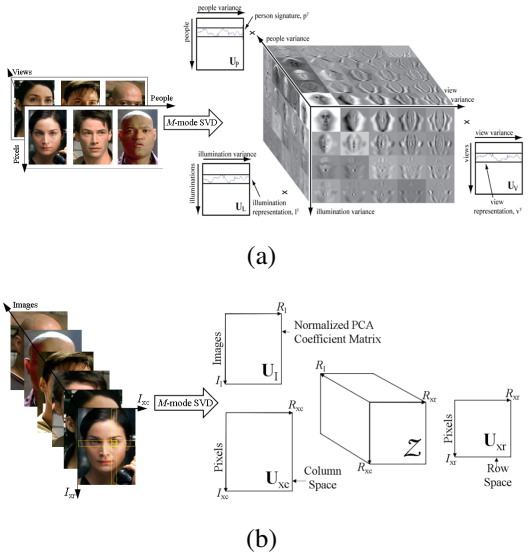

Figure 1: (a) M-mode SVD estimates the parameters of tensor factor model from a collection of vectorized images that have been acquired combinatorially. (b) M-mode SVD computes a regression model and computes the column and row space from a collection of images where each image is a grid of numbers, a "data matrix" or a 2-way array. (All images in this paper have been vectorized, except in this sub-figure.)

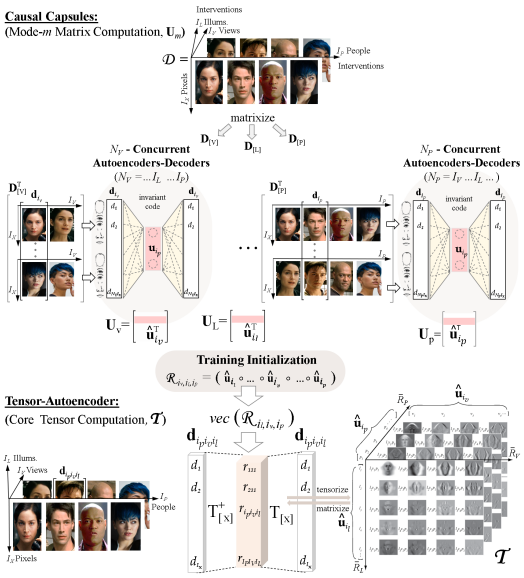

Figure 2: Naive neural network implementation of the M-mode SVD, alg. 1. Depiction of the TensorFaces model estimation, Fig.1(a). Computing each mode matrix, $\mathbf{U}_m$, naively with a single autoencoder-decoder. The core tensor, $\mathcal{T}$, is computed by an autoencoder that is initialized with the vectorized multilinear (tensor) codes formed from the product of factor representations.

effectively employed to reparameterize neural networks. Neural network weights have been organized into "data tensors" and dimensionally reduced in order to achieve greater computational efficiency [Lebedev et al. (2014); Novikov et al. (2015); Kim et al. (2015); Khrulkov (2020); Onu et al. (2020); Iwen et al. (2021). Tensor methods have also been applied in regression analysis [Kolda et al. (2005); Chu & Ghahramani (2009); Tang et al. (2013); Anandkumar et al. (2014); Kossaifi et al. (2017); Wang et al. (2017); Benesty et al. (2021); Vendrow et al. (2021).

This paper introduces a set of causal capsule architectures for forward and inverse causal inference that implement tensor factor analysis operations. These architectures are consistent with capsule theory proposed by Geoffrey Hinton and his collaborators [Hinton et al. (2011); Sabour et al. (2017); Hinton (2021)].

Forward causal inference, the estimation of effects of causes, is performed with a causal autoencoder architecture that consists of several causal capsules that compute the causal factor representations, and a tensor-autoencoder that governs the causal factor interaction. A *causal capsule* is formed from a set of constrained "cluster"-based autoencoders[1] that transform the basis vectors spanning the "cluster" subspace, such that a causal factor representation is invariant of the "cluster" membership, *i.e.*, invariant to all the other causal factors Vasilescu (2009). A tensor autoencoder is an autoencoder with a vectorized multilinear (tensor) product of factor representations, Fig 2.

The M-Mode SVD algorithm that computes an optimal subspace, Alg. 2, and the corresponding architecture serve as a basis for nonlinear causal capsule architectures that employ kernel activations and implements a kernel MPCA or a kernel MICA. Linearizing the manifold space reduces the subspace size, and results in straightforward solutions for any extrapolation tasks.

Causal representations for an object whole may be computed efficiently by parts. As Hinton (2021) has also indicated, a part-based causal capsule architectures may also be "glommed" together and analyze a hierarchy of data columns [Vasilescu et al. (2021); Vasilescu & Kim (2019); de Lathauwer (2008). The hierarchical neural network architecture implements the Incremental M-mode Block

[1] In the context of multifactor data analysis, a cluster is a set of observations for which all factors are fixed except one. Data belonging to the same cluster may not form a cluster in Euclidean space and not easily identifiable by an EM algorithm [Dempster et al. (1977)].

SVD algorithm. The Incremental M-mode Block SVD which is suitable for streaming data, and it is a a compositional hierarchical part-based tensor factorization.[2]

Inverse causal inference is performed with a multilinear projection architecture [Vasilescu & Terzopoulos (2007); Vasilescu (2009) that is performed by inverting an estimated forward model subject to data constraints. Fig. 5.

The architectures are derived based on two mathematical principles: (i)Linear autoencoders-decoders weights are the principal component analysis basis vectors, Sec. A.1, (ii) a part-based eigenspace hierarchy compute object-whole eigenspace, **??**, Sec 3.2.

After reviewing the mathematical foundations of our work in the next section, we discuss forward causal models and depict their neural network arcitectures in Section 3 and discuss inverse causal inference and depict the multilinear projection neural network architectures in Section 4. Section 5 concludes the paper.

Algorithm 1 M-mode SVD algorithm.

Input the data tensor $\mathcal{D} \in \mathbb{C}^{I_0 \times \cdots \times I_M}$.

1. For $m := 0, \ldots, M$,
   Let $\mathbf{U}_m$ be the left orthonormal matrix of $[\mathbf{U}_\mathrm{m} \mathbf{S}_\mathrm{m} \mathbf{V}_\mathrm{m}^\mathrm{T}] := \mathrm{svd}(\mathbf{D}_{[m]})$[a]
2. Set $\mathcal{Z} := \mathcal{D} \times_0 \mathbf{U}_0{}^\mathrm{T} \times_1 \mathbf{U}_1{}^\mathrm{T} \cdots \times_m \mathbf{U}_m{}^\mathrm{T} \cdots \times_M \mathbf{U}_M{}^\mathrm{T}$.

Output mode matrices $\mathbf{U}_0, \mathbf{U}_1 \ldots, \mathbf{U}_M$ and the core tensor $\mathcal{Z}$.

[a] The computation of $\mathbf{U}_m$ in the SVD $\mathbf{D}_{[m]} = \mathbf{U}_m \mathbf{\Sigma} \mathbf{V}_m{}^\mathrm{T}$ can be performed efficiently, depending on which dimension of $\mathbf{D}_{[m]}$ is smaller, by decomposing either $\mathbf{D}_{[m]} \mathbf{D}_{[m]}{}^\mathrm{T} = \mathbf{U}_m \mathbf{\Sigma}^2 \mathbf{U}_m{}^\mathrm{T}$ (note that $\mathbf{V}_m{}^\mathrm{T} = \mathbf{\Sigma}^+ \mathbf{U}_m{}^\mathrm{T} \mathbf{D}_{[m]}$) or by decomposing $\mathbf{D}_{[m]}{}^\mathrm{T} \mathbf{D}_{[m]} = \mathbf{V}_m \mathbf{\Sigma}^2 \mathbf{V}_m{}^\mathrm{T}$ and then computing $\mathbf{U}_m = \mathbf{D}_{[m]} \mathbf{V}_m \mathbf{\Sigma}^+$.

2 Linear Autoencoder and Linear PCA

An autoencoder-decoder that minimizes the reconstruction loss function,

$$l = \sum_{\mathrm{i}=1}^{I} \|\mathbf{d}_\mathrm{i} - \mathbf{B}\mathbf{c}_\mathrm{i}\| + \lambda \|\mathbf{B}^\mathrm{T}\mathbf{B} - \mathbf{I}\| \tag{1}$$

and has a linear decoder learns a set of weights, $\mathbf{b}_\mathrm{r}$ that are identical to the PCA basis vectors when the weights of each neuron, c_r are computed sequentially. An autoencoder is implemented with a cascade of Hebb neurons [Hebb (1949)]. The contribution of each neuron, $c_1, \ldots, c_\mathrm{r}$, are the PCA sequentially computed and subtracted from a centered training data set, and the difference is driven through the next Hebb neuron, $c_{\mathrm{r}+1}$ [Sejnowski et al. (1989); Sanger (1989); Rumelhart et al. (1986); Ackley et al. (1985); Oja (1982)]. The weights of a Hebb neuron, c_r, are updated by

$$\Delta \mathbf{b}_\mathrm{r}(t+1) = \eta \left(\mathbf{d} - \sum_{i_\mathrm{r}=1}^{r} \mathbf{b}_{i_\mathrm{r}}(t) c_{i_\mathrm{r}}(t) \right) c_\mathrm{r}(t) = \eta \left(\mathbf{d} - \sum_{i_\mathrm{r}=1}^{r} \mathbf{b}_{i_\mathrm{r}}(t) \mathbf{b}_{i_\mathrm{r}}^\mathrm{T}(t) \mathbf{d} \right) \mathbf{d}^\mathrm{T} \mathbf{b}_r(t),$$

$$\mathbf{b}_\mathrm{r}(t+1) = \frac{(\mathbf{b}_\mathrm{r}(t) + \Delta \mathbf{b}_\mathrm{r}(t+1))}{\|\mathbf{b}_\mathrm{r}(t) + \Delta \mathbf{b}_\mathrm{r}(t+1)\|}$$

where $\mathbf{d} \in \mathbb{C}^{I_0}$ is a vectorized centered observation with I_0 measurements, η is the learning rate, $\mathbf{b}_\mathrm{r}$ are the autoencoder weights of the r neuron, c_r is the activation, and t is the time iteration. Back-propagation [LeCun et al. (1988; 2012)] is equivalent to performing PCA gradient descent [Jolliffe (1986)].

3 Causal Inference

Throughout this article, we will denote scalars by lower case italic letters $(a, b, \ldots)$, vectors by bold lower case letters $(\mathbf{a}, \mathbf{b}, \ldots)$, matrices by bold uppercase letters $(\mathbf{A}, \mathbf{B}, \ldots)$, and higher-order tensors by bold uppercase calligraphic letters $(\mathcal{A}, \mathcal{B}, \ldots)$. Index upper bounds are denoted by italic uppercase letters (*i.e.*, $1 \leq a \leq A$ or $1 \leq i \leq I$). The zero matrix is denoted by $\mathbf{0}$, and the identity matrix is denoted by $\mathbf{I}$. The TensorFaces paper [Vasilescu & Terzopoulos (2002a) is a gentle introduction to tensor factor analysis, Kolda and Bader [Kolda & Bader (2009) is a nice survey of tensor methods and references [Vasilescu (2009); de Lathauwer (1997); Bro (1997) provide an in depth treatment of tensor factor analysis.

[2] By comparison, a hierarchical Tucker is a resource efficient hierarchical computational scheme that employs a hierarchical re-balancing of the modes trick in which one flattens a data tensor in multiple modes at the same time to avoid computing SVDs of skinny matrices [Hackbusch & Kühn (2009); Grasedyck (2010); Perros et al. (2015).

Algorithm 2 Kernel Multilinear PCA/ICA (K-MPCA/MICA) algorithm.

Input the data tensor $\mathcal{D} \in \mathbb{C}^{I_0 \times \cdots \times I_M}$, where mode $m = 0$ is the measurement mode, and the desired ranks $\tilde{R}_1, \ldots, \tilde{R}_M$.

1. For $m := 1, \ldots, M$,

   Compute the elements of the mode-m covariance matrix, for $j, k := 1, \ldots, I_m$, as follows:

   $$[\mathbf{D}_{[m]}\mathbf{D}_{[m]}{}^{\mathrm{T}}]_{\mathrm{jk}} := \sum_{i_1=1}^{I_1} \cdots \sum_{i_{\mathrm{m-1}}=1}^{I_{m-1}} \sum_{i_{\mathrm{m+1}}=1}^{I_{\mathrm{m+1}}} \cdots \sum_{i_{\mathrm{M=1}}}^{I_M} K(\mathbf{d}_{i_1 \ldots i_{\mathrm{m-1}}\, j\, i_{\mathrm{m+1}} \ldots i_{\mathrm{M}}}, \mathbf{d}_{i_1 \ldots i_{\mathrm{m-1}}\, k\, i_{\mathrm{m+1}} \ldots i_{\mathrm{M}}}).$$

   For K-MPCA: Set $\mathbf{U}_m$ to the left matrix of the SVD of $\mathbf{D}_{[m]}\mathbf{D}_{[m]}{}^{\mathrm{T}} = \mathbf{U}_m \mathbf{\Sigma}^2 \mathbf{U}_m{}^{\mathrm{T}}$
   Truncate to $\tilde{R}_m$ columns $\mathbf{U}_m \in \mathbb{C}^{I_m \times \tilde{R}_m}$.

   For K-MICA: Compute $\mathbf{U}_m := \mathbf{C}_m \in \mathbb{C}^{I_m \times \tilde{R}_m}$ based on
   [Vasilescu & Terzopoulos (2005). The initial SVD truncates to $\tilde{R}_m$.

2. Set $\mathcal{T} := \mathcal{D} \times_1 \mathbf{U}_1^+ \cdots \times_m \mathbf{U}_m^+ \cdots \times_M \mathbf{U}_M^+$.
3. *Local optimization via alternating least squares:*
   Iterate for $n := 1, \ldots, N$
   For $m := 1, \ldots, M$,
   Set $\mathcal{X}_{\mathrm{m}} := \mathcal{D} \times_1 \mathbf{U}_1^+ \cdots \times_{m-1} \mathbf{U}_{m-1}^+ \times_{m+1} \mathbf{U}_{m+1}^+ \cdots \times_M \mathbf{U}_M^+$.
   Set $\mathbf{U}_m$ to the $\tilde{R}_m$ leading left-singular vectors of the SVD of $\mathbf{X}_{\mathrm{m,[m]}}$[a].
   Set $\mathcal{T} := \mathcal{X}_{\mathrm{M}} \times_M \mathbf{U}_M^+$.
   until convergence.

Output the converged extended core tensor $\mathcal{T} \in \mathbb{C}^{I_0 \times \tilde{R}_1 \times \cdots \times \tilde{R}_M}$ and causal factor mode matrices $\mathbf{U}_1, \ldots, \mathbf{U}_M$.

[a] See Alg. 1, footnote *a*

Linear kernel:	$K(\mathbf{u}, \mathbf{v}) = \mathbf{u}^{\mathrm{T}}\mathbf{v} = \mathbf{u} \cdot \mathbf{v}$
Polynomial kernel of degree d:	$K(\mathbf{u}, \mathbf{v}) = (\mathbf{u}^{\mathrm{T}}\mathbf{v})^d$
Polynomial kernel up to degree d:	$K(\mathbf{u}, \mathbf{v}) = (\mathbf{u}^{\mathrm{T}}\mathbf{v} + 1)^d$
Sigmoidal kernel:	$K(\mathbf{u}, \mathbf{v}) = \tanh(\alpha\, \mathbf{u}^{\mathrm{T}}\mathbf{v} + \beta)$
Gaussian (radial basis function (RBF)) kernel:	$K(\mathbf{u}, \mathbf{v}) = \exp\left(-\frac{\lVert \mathbf{u}-\mathbf{v} \rVert^2}{2\sigma^2}\right)$

Table 1: Common kernel functions. Kernel functions are symmetric, positive semi-definite functions(corresponding to symmetric, positive semi-definite Gram matrices). The linear kernel does not modify or warp the feature space.

3.1 Forward Causal Inference

Forward causal inference frames questions in terms of interventions. What if the causal factor c were changed by one unit, how much would the observed measurements, $\mathbf{d}$, be expected to change?

For modeling individual level-effects rather than the average effects of causes, observations are acquired by systematically varying each causal factor while holding the rest of the causal factors fixed.

Within the tensor mathematical framework, a M-way array or "data-tensor", $\mathcal{D} \in \mathbb{C}^{I_0 \times I_1 \cdots \times I_{\mathrm{m}} \cdots \times I_{\mathrm{M}}}$ contains a collection of vectorized and centered observations,[3] $\mathbf{d}_{i_1 \ldots i_{\mathrm{m}} \ldots i_{\mathrm{M}}} \in \mathbb{R}^{I_0}$ that are the result of M causal factors. The m causal factor ($1 \leq m \leq M$) takes one of I_m values that are indexed by i_m, $1 \leq i_m \leq I_m$. An observation that is result of the confluence M causal factors is modeled by a multilinear tensor equation with multimode latent variables,

$$\mathbf{d}_{i_1, \ldots, i_{\mathrm{M}}} \;=\; \mathcal{T} \times_1 (\mathbf{r}_{i_1}^{\mathrm{T}} + \epsilon_{i_1}^{\mathrm{T}}) \cdots \times_{\mathrm{M}} (\mathbf{r}_{i_{\mathrm{M}}}^{\mathrm{T}} + \epsilon_{i_{\mathrm{M}}}^{\mathrm{T}}) + \epsilon_{i_1, \ldots, i_{\mathrm{M}}}$$

where $\mathcal{T} = \mathcal{Z} \times_0 \mathbf{U}_0$ is the extended core which modulates the interaction between the latent variables, $\mathbf{r}_{i_1} \ldots \mathbf{r}_{i_{\mathrm{m}}} \ldots \mathbf{r}_{i_{\mathrm{M}}}$, that represent the causal factors, and $\boldsymbol{\epsilon}_{i_{\mathrm{m}}} \in \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\mathrm{m}})$ is an IID Gaussian noise.

[3] Reference [(Vasilescu, 2009, Appendix A) evaluates some of the arguments found in highly cited publications in favor of treating an image as a matrix (tensor) rather than a vector. While technically speaking, it is not incorrect to treat an image as a matrix in linear/tensor algebra, most arguments do not stand up to analytical scrutiny, and it is preferable to vectorize an image and treat it as a single observation rather than a collection of independent column/row observations.

The M-mode SVD, Alg. **??** and Alg. 2, and their neural network architecture counterparts, Fig. 2, Fig. 3 may be employed to represent data in terms of their causal factors by minimizing the reconstruction loss function,

$$l = \|\mathcal{D} - \mathcal{T} \times_1 \mathbf{U}_1 \cdots \times_m \mathbf{U}_m \cdots \times_M \mathbf{U}_M\| + \sum_{m=1}^{M} \lambda_m \|\mathbf{U}_m \mathbf{U}_m^T - \mathbf{I}\|, \tag{2}$$

where $\mathcal{T}$ is the extended core and the mode matrices, $\mathbf{U}_m$, spans the m causal factor representation. Each mode matrix, $\mathbf{U}_m$ is computed using alternating least squares where a set of M least squares are computed by moving the mode matrices $\mathbf{U}_1, \ldots, \mathbf{U}_{m-1}, \mathbf{U}_{m+1}, \ldots, \mathbf{U}_M$ to the knowns side of the equation, setting $\mathcal{X}_m := \mathcal{D} \times_1 \cdots \times_{m-1} \mathbf{U}_{m-1}^T \times_{m+1} \mathbf{U}_{m+1}^T \cdots \times_M \mathbf{U}_M^T$ for distributed computation or setting $\mathcal{X}_m := (\mathcal{X}_{m-1} \times_{m-1} \mathbf{U}_{m-1}^T) \times_m \mathbf{U}_m)$ for sequential computation, and optimizing the loss function

$$\begin{aligned} l &= \|\mathcal{X}_m - \mathcal{T} \times_m \mathbf{U}_m\| + \lambda \|\mathbf{U}_m \mathbf{U}_m^T - \mathbf{I}\| = \|\mathbf{X}_{m[m]} - \mathbf{U}_m \mathbf{T}_{[m]}\| + \lambda \|\mathbf{U}_m \mathbf{U}_m^T - \mathbf{I}\|, && (3) \\ &\text{where } \mathcal{X}_m := \mathcal{D} \times_1 \cdots \times_{m-1} \mathbf{U}_{m-1}^T \times_{m+1} \mathbf{U}_{m+1}^T \cdots \times_M \mathbf{U}_M^T && \text{- distributed computation} \quad (4) \\ &= (\mathcal{X}_{m-1} \times_{m-1} \mathbf{U}_{m-1}^T) \times_m \mathbf{U}_m = \mathcal{T} \times_m \mathbf{U}_m && \text{- sequential computation} \quad (5) \end{aligned}$$

The mode matrix $\mathbf{U}_m$ is set to the subspace of the matrixized $\mathcal{X}_m$, $\mathbf{U}_m \mathbf{S}_m \mathbf{V}_m^T := \text{svd}(\mathbf{X}_{m[m]})$[4]. For a distributed computation, M different $\mathcal{X}_m$ are computed by shuttling mode matrices between the different threads, eq.(4). Figure 3(e) displays the sequential mode matrix computations by unrolling the for-loop from step 3, Alg 2 based on eq.(5).

3.2 Derivation: Invariance and Hierarchy of Causal Capsules

Computing the mode matrices, $\mathbf{U}_m$, may be viewed as equivalent to computing a set of mutually constrained, cluster-based PCAs. When dealing with data that can be separated into clusters, the standard machine learning approach is to compute a separate PCA. When data from different clusters are generated by the same underlying process (e.g., facial images of the same people under different viewing conditions), the underlying data can be concatenated in the measurement mode and the common causal factor can be modeled by one PCA.[5]

Thus, we define a *constrained, cluster-based PCA* as the computation of a set of PCA basis vectors that are rotated such that the latent representation is constrained to be the invariant of the cluster.

MPCA performs M constrained, cluster-based PCAs, since the computation of the mode-m matrix $\mathbf{U}_m$, which involves a mode-m data tensor flattening and subsequent SVD, is equivalent to performing a constrained, cluster-based PCA; i.e., data cluster concatenation followed by an SVD. This is self evident when employing our modified datum-centric flattening operator, Fig. 7.

In the context of our multifactor data analysis, we define a cluster as a set of observations for which all factors are fixed except one, the m factor. Note that there are $N_m = I_1 I_2 \ldots I_{m-1} I_{m+1} \ldots I_M$ possible clusters and the data in each cluster varies with the same causal mode.[6] Thus, the data across different clusters share one of the underlying causal factors. The constrained, cluster-based PCA concatenates the clusters in the measurement mode and analyzes the data with a linear model, such as PCA or ICA [Bartlett et al. (2002); Common (1994); Lathauwer et al. (1995); De Lathauwer et al. (1996); Anandkumar et al. (2014).

To see this, let $\mathcal{D}_{i_1 \ldots i_{m-1} i_{m+1} \ldots i_M} \in \mathbb{C}^{I_0 \times 1 \times 1 \cdots \times 1 \times I_m \times 1 \cdots \times 1}$ denote a subtensor of $\mathcal{D}$ that is obtained by fixing all modes except causal factor mode m and mode (the measurement mode). Matrixizing this subtensor in the measurement mode 0, we obtain $\mathbf{D}_{i_1 \ldots i_{m-1} i_{m+1} \ldots i_M [0]} \in \mathbb{C}^{I_0 \times I_m}$. This data matrix comprises a cluster of data obtained by varying the m causal factor, to which one can traditionally apply PCA. Since there are $N_m = I_1 I_2 \ldots I_{m-1} I_{m+1} \ldots I_M$ possible clusters that share the same underlying space associated with the $cmth$ factor, the data can be concatenated and PCA performed in order to extract the same representation for the m factor regardless of the cluster. Now, consider the MPCA computation of mode matrix $\mathbf{U}_m$, Fig. 3(a), which can be written in terms of matrixized

[4] See Alg. 1, footnote *a*

[5] The active appearance model concatenated two different measurements, facial feature locations and texture, to compute a person representation for a particular viewpoint invariant of measurement. More generally, one can concatenate the measurements of a person from different viewpoint clusters to compute a person representation that is invariant of the measurement and the viewpoint [Cootes et al. (2001); Si et al. (2013).

[6] Observations in the cluster may not be in Euclidean proximity in the measurement space. Consequently, a cluster may not be easily identified through a standard EM algorithm.

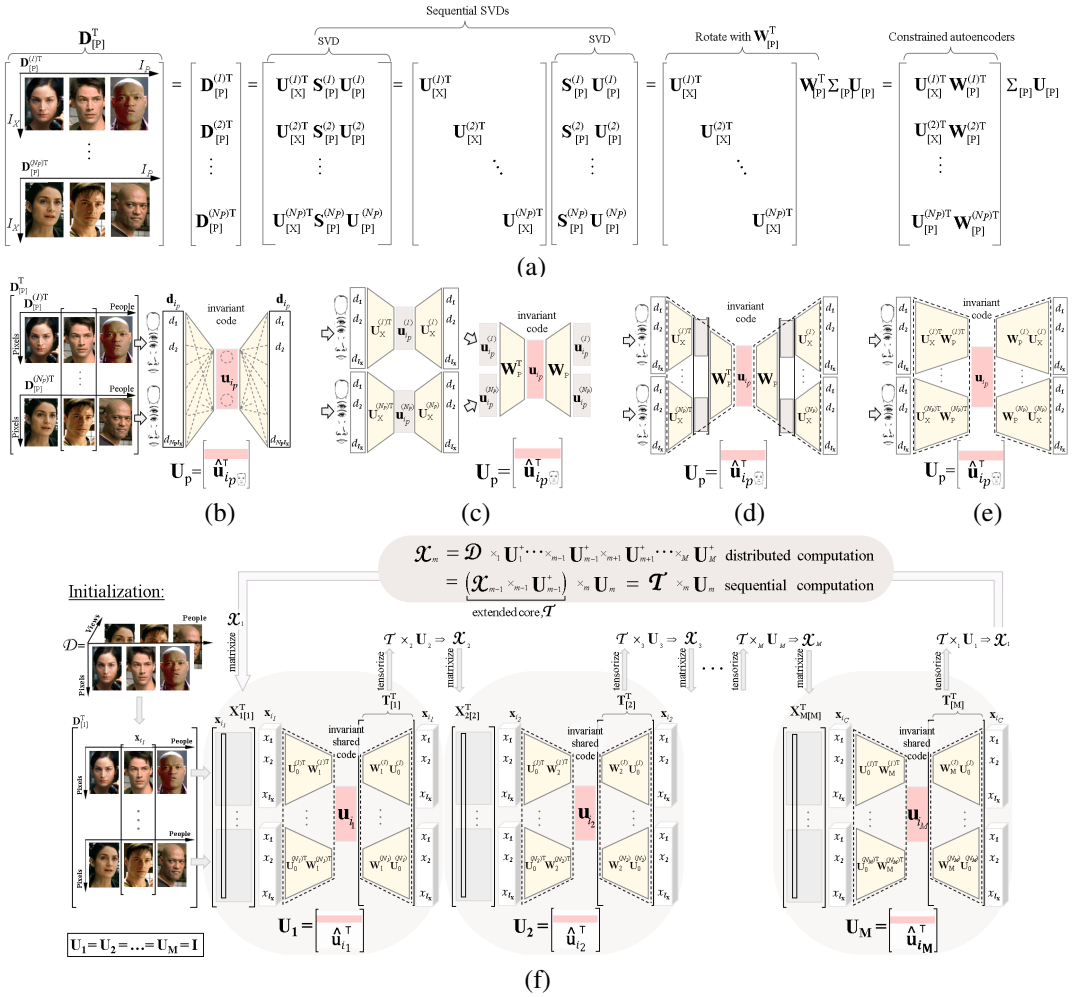

Figure 3: Face recognition example. (a) An ensemble of vectorized images is organized into $\mathcal{D} \in \mathbb{C}^{I_{\mathrm{X}} \times I_{\mathrm{P}} \times I_{\mathrm{V}} \times I_{\mathrm{L}} \times I_{\mathrm{E}}}$ is matrixized into a data matrix, $\mathbf{D}_{\mathrm{P}}$ from which one can compute the mode matrix, $\mathbf{U}_{\mathrm{P}}$, that spans the person representation. This depicts how a single SVD($\mathbf{D}_{[P]}$ can be written in terms of (i) constrained cluster-based autoencoder (PCA) and (ii) concurrent autoencoder. This is depicted as a neural network architecture in (b), (c) and (d), respectively. (b) Mode matrix computation using a single autoencoder-decoder. (c) Mode matrix computation using a constrained cluster-based autoencoder-decoder based on the derivation in part (a). (d) Concurrent-autoencoder. (e) The neural network architecture consists of a chain of constrained autoencoders-decoders where the weights of one constrained autoencoder-decoder are the inputs of the next one. This constrained recurrent causal chain is the unrolled for-loop that computes the mode matrices by employing alternating least squares. When the autoencoders employ kernels then the architecture implements K-MPCA/ K-MICA, Alg. 2.

subtensors as

$$\mathbf{D}_{\mathrm{m}}{}^{\mathrm{T}} = \begin{bmatrix} \mathbf{D}_{1\ldots 11\ldots 1\,[m]}{}^{\mathrm{T}} \\ \vdots \\ \mathbf{D}_{I_1\ldots 11\ldots 1\,[m]}{}^{\mathrm{T}} \\ \vdots \\ \mathbf{D}_{I_1\ldots I_{m-1} I_{m+1}\ldots I_M\,[m]}{}^{\mathrm{T}} \end{bmatrix} = \mathbf{V}_m \mathbf{\Sigma}_m \mathbf{U}_m{}^{\mathrm{T}}. \tag{6}$$

Clearly, this is equivalent to computing a set of $N_m = I_1 I_2 \ldots I_{m-1} I_{m+1} \ldots I_M$ cluster-based PCAs concurrently by combining them into a single statistical model and representing the underlying causal factor m common to the clusters. Thus, rather than computing a separate linear PCA model for each cluster, MPCA concatenates the clusters into a single statistical model and computes a representation (coefficient vector) for mode m that is invariant relative to the other causal factor modes $1, ..., (m-1), (m+1), ..., M$. Thus, MPCA is a multilinear, constrained, cluster-based PCA.

To clarify the relationship, let us number each of the matrices $\mathbf{D}_{i_1 \ldots i_{m-1} i_{m+1} \ldots i_{M\,[m]}} = \mathbf{D}_m^{(n)}$ with a parenthetical superscript $1 \leq n = 1 + \sum_{k=1,k\neq m}^{M} (i_n - 1) \prod_{l=1,l\neq m}^{k-1} I_l \leq N_m$.

Let each of the linear SVDs be

$$\mathbf{D}_m^{(n)} = \mathbf{U}_m^{(n)} \mathbf{\Sigma}_m^{(n)} \mathbf{U}_0^{(n)\mathrm{T}} \tag{7}$$

$$\mathbf{D}_{[m]} = \underbrace{\left[\mathbf{U}_m^{(1)} \mathbf{\Sigma}_m^{(1)} \ldots \mathbf{U}_m^{(N_M)} \mathbf{\Sigma}_m^{(N_m)}\right]}_{SVD} \mathrm{diag}([\mathbf{U}_0^{(1)} \ldots \mathbf{U}_0^{(N_m)}])^{\mathrm{T}}, \tag{8}$$

$$= \quad \mathbf{U}_m \mathbf{\Sigma}_m W_{\mathrm{m}}^{\mathrm{T}} \mathrm{diag}([\mathbf{U}_0^{(1)} \ldots \mathbf{U}_0^{(N_m)}])^{\mathrm{T}} \tag{9}$$

$$= \quad \mathbf{U}_m \mathbf{\Sigma}_m [\mathbf{W}_{\mathrm{m}}^{(1)\mathrm{T}} \mathbf{U}_0^{(1)} \ldots \mathbf{W}_{\mathrm{m}}^{(N_m)\mathrm{T}} \mathbf{U}_{\mathrm{m}}^{(N_m)}]^{\mathrm{T}} \tag{10}$$

where $\mathrm{diag}(\cdot)$ denotes a diagonal matrix whose elements are each of the elements of its vector argument. The mode matrix $\mathbf{U}_0^{(n_{\mathrm{m}})}$ is the measurement matrix that contains the eigenvectors that span the observed data in cluster n_{m}, $1 \leq n_{\mathrm{m}} \leq N_{\mathrm{m}}$. MPCA can be thought as computing a rotation matrix, $\mathbf{W}_{\mathrm{m}}$, that contains a set of blocks $\mathbf{W}_{\mathrm{m}}^{(n)}$ along the diagonal that transform the PCA cluster eigenvectors, $\mathbf{U}_0^{(n_{\mathrm{m}})}$, such that the mode matrix $\mathbf{U}_m$ is the same regardless of cluster membership, eqs.(**??**-10). The constrained "cluster"-based PCAs may also be implemented with a set of concurrent "cluster"-based PCAs.

Object wholes appear to have have a hierarchy of perceptual parts. Part-based causal capsule architectures may be "glommed" together to create a hierarchy of part-based causal capsules that analyze a hierarchy of data columns [Vasilescu et al. (2021); Vasilescu & Kim (2019); de Lathauwer (2008). This hierarchical architecture implements the incremental hierarchical multilinear (tensor) block decomposition algorithm.

Causal factors of object wholes may be computed efficiently from their parts, Fig **??**. The matrixized data tensor may be organized into part "clusters" by applying a permutation

$$\mathcal{D}\ {}^{\mathrm{T}}\!\times_{\mathrm{m}} \mathbf{P} \Leftrightarrow \mathbf{P}\mathbf{D}_{[m]}{}^{\mathrm{T}}$$

(11)

where $\mathbf{P} is a permutation matrix. The resulting hierarchical architecture implements the Incremental M-mode Block SVD algorithm. The Incremental M-mode Block SVD is a generalized hierarchical part-based decomposition that computes an exact global decomposition and is suitable for streaming data [Vasilescu et al. (2021$

3.3 Nonlinear Causal Capsules and Kernel MPCA/Kernel MICA

An autoencoder with a non-linear activation function represents an observation with

$$\mathbf{d}_{\mathrm{i}} = f_{\mathrm{d}}(\mathbf{B}_{\mathrm{d}} \underbrace{f_{\mathrm{e}}(\mathbf{B}_{\mathrm{e}} d_{\mathrm{i}} + \mathbf{a}_{\mathrm{e}})}_{\mathbf{c}_{\mathrm{i}}} + \mathbf{a}_{\mathrm{d}}), \tag{12}$$

where f_{e}, f_{d} are the encoder, decoder activation functions, and $\mathbf{B}_{\mathrm{e}}, \mathbf{B}_{\mathrm{d}}$ are the encoder, decoder weights respectively. Kernel PCA (KPCA) is often given as an example of a "true" nonlinear model. KPCA first applies a nonlinear transformation to the data and then it performs a linear decomposition. Thus, KPCA derives its nonlinearity from its preprocessing step. Other nonlinear methods include nonlinear PCA (NLPCA) [Kramer (1991), as well as kernel PCA (KPCA) [Schölkoph et al. (1998) and kernel LDA (KLDA) [Yang (2002) methods in which kernel functions that satisfy Mercer's theorem correspond to inner products in infinite-dimensional space. An alternative approach is to apply linear models to nonlinear problems through the "kernel trick", specifically the kernel PCA [Schölkoph et al. (1998) and kernel ICA [Yang et al. (2005) techniques.[7] Kernel PCA/ICA are nonlinear versions of their conventional linear counterparts, but of course they are not multimodal factor models.

[7] The so-called "kernel trick" maps the original non-linear measurements into a higher-dimensional space, where a linear classifier is subsequently used; this makes a linear classification in the new space equivalent to non-linear classification in the original space. This is done using Mercer's theorem, which states that any continuous, symmetric, positive semi-definite kernel $K(\mathbf{u}, \mathbf{v})$ can be expressed as an inner product in a high-dimensional space. Wherever an inner product between two vectors is used, it is replaced with a kernel of the vectors. Thus, a linear algorithm is easily transformed into a nonlinear algorithm. This trick has been applied to numerous algorithms in machine learning and statistics.

The kernel trick can also be applied to our multilinear, multifactor PCA/ICA models to further nonlinearize them, thus enabling them to deal with arbitrarily nonlinear data, Alg. 2.

To accomplish this, recall that the computation of the mode-m covariance matrix $\mathbf{D}_{[m]}\mathbf{D}_{[m]}{}^{\mathrm{T}}$ involves inner products $\mathbf{d}^{\mathrm{T}}_{i_1 \ldots i_{\mathrm{m-1}}\, j\, i_{\mathrm{m+1}} \ldots i_{\mathrm{M}}} \mathbf{d}_{i_2 \ldots i_{\mathrm{m-1}}\, k\, i_{\mathrm{m+1}} \ldots i_{\mathrm{M}}}$ between pairs of images in the image data tensor $\mathcal{D}$ associated with causal factor mode m, for $m = 1, \ldots, M$. We replace the inner products with a generalized distance measure between images, $K(\mathbf{d}_{i_1 \ldots i_{m-1}\, j\, i_{m+1} \ldots i_M}, \mathbf{d}_{i_2 \ldots i_{m-1}\, k\, i_{m+1} \ldots i_M})$, where $K(\cdot, \cdot)$ is a suitable kernel function (Table 1), which corresponds to an inner product in some expanded feature space. This generalization naturally leads us to a *Kernel Multilinear PCA (K-MPCA) Algorithm*, where the covariance step computation in in Algorithm 1 is replaced by

$$[\mathbf{D}_{[m]}\mathbf{D}_{[m]}{}^{\mathrm{T}}]_{jk} := \sum_{i_1=1}^{I_1} \cdots \sum_{i_{m-1}=1}^{I_{m-1}} \sum_{i_{m+1}=1}^{I_{m+1}} \cdots \sum_{i_M=1}^{I_M} K(\mathbf{d}_{i_1 \ldots i_{m-1}\, j\, i_{m+1} \ldots i_M}, \mathbf{d}_{i_1 \ldots i_{m-1}\, k\, i_{m+1} \ldots i_M}).$$

Similarly, a *Kernel Multilinear ICA (K-MICA) Algorithm* results from making the same generalization in the MICA algorithm [Vasilescu & Terzopoulos (2005). Algorithm 2 specifies both K-MPCA and K-MICA. Figure 3(d) unrolls the for-loop in step 3 of Alg. 2.

4 Inverse Causal Inference: Multilinear and Multiple Linear Projections

Inverse causal inference estimates the causes of effects, and addresses the why question. Inverse problems often violate one of the conditions of a well-posed problem, and Donald Rubin has referred to the "why" question as "cocktail party chatter" [Gelman & Imbens (2013). For a problem to be well-posed a solution must exit, it must be unique and the solution's behaviour ought to change continuously with the initial conditions [Hadamard (1952). Often, there are multiple combinations of same causal factors that have the same potential outcome. In imaging, these types of outcomes are known as visual illusions.

Therefore, inverse causal inference is the estimation of causes of effects given an estimated forward causal model that is inverted subject to a set of observations that constrain the solution set and render the problem well-posed [Vasilescu et al. (2021)].Similar to reverse causal inference [Gelman & Imbens (2013)], inverse causal inference may be employed as a model checking mechanism and motivation for forward inference question.

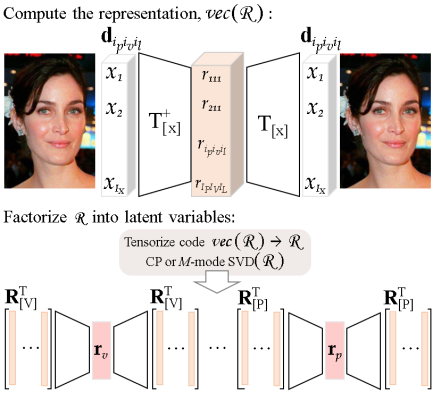

Figure 5: Neural network architecture of the multilinear projection algorithm [Vasilescu & Terzopoulos (2007) given an estimated interaction causal model, $\mathcal{T}$ (*i.e.*, $\mathbf{T}_{[\mathrm{x}]}$).

Multilinear projection simultaneously projects one or more unlabeled test images that are not part of the training data set into multiple constituent causal factor spaces associated with data formation, in order to infer the mode labels:

$$\text{CP or } M\text{-mode SVD}(\mathcal{T}^{+}_{\mathrm{x}} \times^{\mathrm{T}}_{\mathrm{x}} \mathbf{d}_{\mathrm{test}}) \approx \mathbf{r}_1 \ldots \circ \mathbf{r}_{\mathrm{m}} \ldots \circ \mathbf{r}_{\mathrm{M}} \circ \mathbf{r}_{\mathrm{E}}.$$

Topologically the multilinear projection architecture, Fig. 5, is an inverted M-mode SVD architecture. When the dimensionality of $vec(\mathcal{R})$ is larger than the number of measurements in $\mathbf{d}$, then the system of equations is under determined. There are three possible solutions – dimensionality reduction of the mode matrices, and modeling the mechanism of data formation by multiple linear or tensor models. Instead of performing a multilinear projection, [Vasilescu & Terzopoulos (2002b) perform a set of linear projections.

5 Conclusion

This paper introduces deep causal learning architectures that implement tensor factor analysis operations and model the mechanism of data formation. The tensor factor analysis methods, the M-mode SVD, the Kernel MPCA/MICA, and the associated causal capsules architectures transform the "cluster" eigenvectors such that the constituent causal representations are invariant of the cluster membership, *i.e.*, invariant of other causal factors of data formation. Causal representation may be computed efficiently by "glomming" together a hierarchy of part-based causal capsules. The

hierarchical part-based causal architecture implements the compositional hierarchical tensor factorization, the Incremental M-mode Block SVD. Each part-based capsule analyzes a data column from a hierarchy of data columns [Vasilescu et al. (2021). Inverse causal inference, the estimation of causes of effects, is accomplished with a multilinear projection algorithm. The neural architecture that implements the multilinear projection is an inverted M-mode SVD architecture. Tensor causal factor analysis and their associates neural networks have properties consistent with the capsule theory. Tensor causal factor analysis has been applied on real and synthetic data in many domains, including face recognition where the approach is known as TensorFaces and computer graphics where a set of TensorTextures are synthesized for arbitrary geometries.

REFERENCES

Evrim Acar, Evangelos E Papalexakis, Gözde Gürdeniz, Morten A Rasmussen, Anders J Lawaetz, Mathias Nilsson, and Rasmus Bro. Structure-revealing data fusion. *BMC bioinformatics*, 15(1):239, 2014. 1

David H. Ackley, Geoffrey A. Hinton, and Terrence J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985. ISSN 0364-0213. doi: https://doi.org/10.1016/S0364-0213(85)80012-4. URL https://www.sciencedirect.com/science/article/pii/S0364021385800124. 3, 14

Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M. Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. 15(1):27732832, 2014. 2, 5

Elias Bareinboim and Judea Pearl. Causal inference and the data-fusion problem. *Proc. of the National Academy of Sciences*, 113(27):7345–52, 2016. ISSN 0027-8424. doi: 10.1073/pnas.1510507113. URL https://www.pnas.org/content/113/27/7345. 1

M.S. Bartlett, J.R. Movellan, and T.J. Sejnowski. Face recognition by independent component analysis. *IEEE Transactions on Neural Networks*, 13(6):1450–64, 2002. 5

Jacob Benesty, Constantin Paleologu, Laura-Maria Dogariu, and Silviu Ciochin. Identification of linear and bilinear systems: A unified study. *Electronics*, 10(15), 2021. ISSN 2079-9292. doi: 10.3390/electronics10151790. URL https://www.mdpi.com/2079-9292/10/15/1790. 2

Peter M. Bentler and Sik-Yum Lee. A statistical development of three-mode factor analysis. *British J. of Math. and Stat. Psych.*, 32(1):87–104, 1979. doi: 10.1111/j.2044-8317.1979.tb00754.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.2044-8317.1979.tb00754.x. 1

V. Blanz and T. A. Vetter. Morphable model for the synthesis of 3D faces. In *Proc. ACM SIGGRAPH 99 Conf.*, pp. 187–194, 1999. 15

Kenneth A. Bollen and Judea Pearl. *Eight Myths About Causality and Structural Equation Models*, pp. 301–328. Springer Netherlands, Dordrecht, 2013. ISBN 978-94-007-6094-3. doi: 10.1007/978-94-007-6094-3_15. URL https://doi.org/10.1007/978-94-007-6094-3_15. 1

Rasmus Bro. Parafac: Tutorial and applications. In *Chemom. Intell. Lab Syst., Special Issue 2nd Internet Cont. in Chemometrics (INCINC'96)*, volume 38, pp. 149–171, 1997. 1, 3, 14

J. D. Carroll and J. J. Chang. Analysis of individual differences in multidimensional scaling via an N-way generalization of 'Eckart-Young' decomposition. *Psychometrika*, 35:283–319, 1970. 1

J. D. Carroll, S. Pruzansky, and J. B. Kruskal. CANDELINC: A general approach to multidimensional analysis of many-way arrays with linear constraints on parameters. *Psychometrika*, 45:3–24, 1980. 15

J. C. Chen, R. Ranjan, A. Kumar, C. H. Chen, V. M. Patel, and R. Chellappa. An end-to-end system for unconstrained face verification with deep convolutional neural networks. In *IEEE International Conf. on Computer Vision Workshop (ICCVW)*, pp. 360–368, Dec 2015. doi: 10.1109/ICCVW.2015.55. 1

Wei Chu and Zoubin Ghahramani. Probabilistic models for incomplete multi-dimensional arrays. volume 5 of *Proceedings of Machine Learning Research*, pp. 89–96, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR. URL http://proceedings.mlr.press/v5/chu09a.html. 2

Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009. 1

Pierre Common. Independent component analysis, a new concept? *Signal Processing*, 36:287–314, 1994. URL https://hal.archives-ouvertes.fr/hal-00417283. 5

T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001. 5

J. Davis and H. Gao. Recognizing human action efforts: An adaptive three-mode PCA framework. In *Proc. IEEE Inter. Conf. on Computer Vision, (ICCV)*, pp. 1463–69, Nice, France, Oct 13-16 2003. 1

L. de Lathauwer. Decompositions of a higher-order tensor in block termspart ii: Definitions and uniqueness. *SIAM J. on Matrix Analysis and Applications*, 30(3):1033–1066, 2008. doi: 10.1137/070690729. URL https://doi.org/10.1137/070690729. 1, 2, 7

L. De Lathauwer, B. De Moor, and J. Vandewalle. Independent component analysis based on higher-order statistics only. In *Proceedings of 8th Workshop on Statistical Signal and Array Processing*, pp. 356–359, 1996. doi: 10.1109/SSAP.1996.534890. 5

Lieven de Lathauwer. *Signal Processing Based on Multilinear Algebra*. PhD dissertation, Katholieke Univ. Leuven, Belgium, 1997. 1, 3, 14, 15

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977. doi: https://doi.org/10.1111/j.2517-6161.1977.tb01600.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1977.tb01600.x. 2

Andrew Gelman and Guido Imbens. Why ask why? forward causal inference and reverse causal questions. Tech.report, Nat.Bureau of Econ Research, 2013. 8

Lars Grasedyck. Hierarchical singular value decomposition of tensors. *SIAM J. on Matrix Analysis and Applications*, 31(4):2019–54, 2010. 3

Wolfgang Hackbusch and Stefan Kühn. A new scheme for the tensor representation. *Journal of Fourier Analysis and Applications*, 15(5):706–722, 2009. 3

J. Hadamard. *Lectures on Cauchy's Problem in Linear Partial Differential Equations*. Dover Publisher, 1952. 8

R. Harshman. Foundations of the PARAFAC procedure: Model and conditions for an explanatory factor analysis. Tech. Report Working Papers in Phonetics 16, UCLA, CA, Dec 1970. 1

Ali Hatamizadeh, Demetri Terzopoulos, and Andriy Myronenko. End-to-end boundary aware networks for medical image segmentation. In *Inter. Workshop on Machine Learning in Medical Imaging*, pp. 187–194. Springer, 2019. 15

D. O. Hebb. *The organization of behavior: A neuropsychological theory*. John Wiley And Sons, Inc., New York, 1949. 3, 14

Geoffrey Hinton. How to represent part-whole hierarchies in a neural network, 2021. 2

Geoffrey E. Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders, 2011. 2

Paul W. Holland. Statistics and causal inference: Rejoinder. *J. of the American Statistical Association*, 81(396): 968970, 1986. URL https://users.nber.org/~rdehejia/!@SAEM/Topic01Causality/holland_JASA_1986.pdf. 1

Eugene Hsu, Kari Pulli, and Jovan Popovic. Style translation for human motion. *ACM Transactions on Graphics*, 24(3):1082–89, 2005. URL http://people.csail.mit.edu/jovan/assets/papers/hsu-2005-stf.pdf. 1

G. B. Huang. Learning hierarchical representations for face verification with convolutional deep belief networks. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 2518–25, Jun 2012. doi: 10.1109/CVPR.2012.6247968. 1

Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, Oct 2007. 16, 17

Guido W. Imbens. Potential outcome and directed acyclic graph approaches to causality: Relevance for empirical practice in economics. *Journal of Economic Literature*, 58(4):1129–79, December 2020. doi: 10.1257/jel.20191597. URL https://www.aeaweb.org/articles?id=10.1257/jel.20191597. 1

Guido W. Imbens and Donald B. Rubin. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge Univ. Press, 2015. ISBN 0521885884. 1

Mark A Iwen, Deanna Needell, Elizaveta Rebrova, and Ali Zare. Lower memory oblivious (tensor) subspace embeddings with fewer random bits: modewise methods for least squares. *SIAM Journal on Matrix Analysis and Applications*, 42(1):376–416, 2021. 2

I.T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986. 3, 14

A. Kapteyn, H. Neudecker, and T. Wansbeek. An approach to n-mode component analysis. *Psychometrika*, 51 (2):269–275, Jun 1986. 1

Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, CVPR '14, pp. 1867–74, Washington, DC, USA, 2014. IEEE Computer Society. ISBN 978-1-4799-5118-5. doi: 10.1109/CVPR.2014.241. URL http://dx.doi.org/10.1109/CVPR.2014.241. 15

Daniel S. Kermany, Michael Goldbaum, Wenjia Cai, Carolina C.S. Valentim, Huiying Liang, Sally L. Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, Justin Dong, Made K. Prasadha, Jacqueline Pei, Magdalene Y.L. Ting, Jie Zhu, Christina Li, Sierra Hewett, Jason Dong, Ian Ziyar, Alexander Shi, Runze Zhang, Lianghong Zheng, Rui Hou, William Shi, Xin Fu, Yaou Duan, Viet A.N. Huu, Cindy Wen, Edward D. Zhang, Charlotte L. Zhang, Oulan Li, Xiaobo Wang, Michael A. Singer, Xiaodong Sun, Jie Xu, Ali Tafreshi, M. Anthony Lewis, Huimin Xia, and Kang Zhang. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5):1122–1131.e9, 2018. ISSN 0092-8674. doi: https://doi.org/10.1016/j.cell.2018.02.010. URL https://www.sciencedirect.com/science/article/pii/S0092867418301545. 1

Valentin Khrulkov. *Geometrical Methods in Machine Learning and Tensor Analysis*. PhD dissertation, Skolkovo Institute of Science and Technology, 2020. 2

Yong-Deok Kim, Eunhyeok Park, Sungjoo Yoo, Taelim Choi, Lu Yang, and Dongjun Shin. Compression of deep convolutional neural networks for fast and low power mobile applications. *CoRR*, abs/1511.06530, 2015. URL http://arxiv.org/abs/1511.06530. 2

Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009. 15

Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009. 3, 14

Tamara G. Kolda, Brett W. Bader, and Joseph P. Kenny. Higher-order web link analysis using multilinear algebra. In *ICDM 2005: Proceedings of the 5th IEEE International Conference on Data Mining*, pp. 242–249, 2005. doi: 10.1109/ICDM.2005.77. 2

Jean Kossaifi, Aran Khanna, Zachary C. Lipton, Tomasso Furlanello, and Anima Anandkumar. Tensor contraction layers for parsimonious deep nets. In *Computer Vision and Pattern Recognition (CVPR), Tensor Methods in Computer Vision Workshop*, pp. 1940–46, Jul 2017. 2

M. A. Kramer. Nonlinear principal components analysis using autoassociative neural networks. *AIChE J.*, 32(2): 233–243, Feb 1991. 7

P. M. Kroonenberg and J. de Leeuw. Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika*, 45:69–97, 1980. 1

Lieven De Lathauwer, P Comon, Bart De Moor, and Joos Vandewalle. Higher-order power method - application in independent component analysis. *Proc. of the International Symposium on Nonlinear Theory and its Applications (NOLTA'95)*, pp. 91–96, 1995. 5

Vadim Lebedev, Yaroslav Ganin, Maksim Rakhuba, Ivan V. Oseledets, and Victor S. Lempitsky. Speeding-up convolutional neural networks using fine-tuned cp-decomposition. *CoRR*, abs/1412.6553, 2014. URL http://arxiv.org/abs/1412.6553. 2

Yann LeCun, D Touresky, G Hinton, and T Sejnowski. A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*, volume 1, pp. 21–28, 1988. 3, 14

Yann A. LeCun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. *Efficient BackProp*, pp. 9–48. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-35289-8. doi: 10.1007/978-3-642-35289-8_3. URL https://doi.org/10.1007/978-3-642-35289-8_3. 3, 14

I. Macedo, E. V. Brazil, and L. Velho. Expression transfer between photographs through multilinear aam's. pp. 239–246, Oct 2006. doi: 10.1109/SIBGRAPI.2006.18. 15

Ali Madani, Mehdi Moradi, Alexandros Karargyris, and Tanveer Syeda-Mahmood. Semi-supervised learning with generative adversarial networks for chest x-ray classification with ability of data domain adaptation. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pp. 1038–1042, 2018. doi: 10.1109/ISBI.2018.8363749. 1

J. R. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. John Wiley & Sons, 1988. 1

Alexander Novikov, Dmitrii Podoprikhin, Anton Osokin, and Dmitry P Vetrov. Tensorizing neural networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 28*, pp. 442–450. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5787-tensorizing-neural-networks.pdf. 2

Erkki Oja. A simplified neuron model as a principal component analyzer. 15:267–2735, 1982. 3, 14

Charles C. Onu, Jacob E. Miller, and Doina Precup. A fully tensorized recurrent neural network. *CoRR*, abs/2010.04196, 2020. URL https://arxiv.org/abs/2010.04196. 2

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge Univ. Press, 2000. ISBN 9780521773621. URL https://books.google.com/books?id=wnGU_TsW3BQC. 1

Judea Pearl and Elias Bareinboim. External validity: From do-calculus to transportability across populations. *Statistical Science*, 29(4):57995, 2014. 1

I. Perros, R. Chen, R. Vuduc, and J. Sun. Sparse hierarchical tucker factorization and its application to healthcare. In *Proc. IEEE Inter. Conf. on Data Mining*, pp. 943–948, Nov 2015. doi: 10.1109/ICDM.2015.29. 3

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. *Learning internal representations by error propagation*. 1986. 3, 14

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/2cad8fa47bbef282badbb8de5374b894-Paper.pdf. 2

Terry Sanger. Optimal unsupervised learnig in a single layer linear feedforward neural network. 12:459–473, 1989. 3, 14

B. Schölkoph, A. Smola, and K.-R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998. 7

Terrence Sejnowski, Sumantra Chattarji, and Patric Sfanton. *Induction of Synaptic Plasticity by Hebbian Covariance in the Hippocampus*, pp. 105–124. Addison-Wesley, 1989. 3, 14

Weiguang Si, Kota Yamaguchi, and M. Alex O. Vasilescu. Face tracking with multilinear (tensor) active appearance models. http://pdfs.semanticscholar.org/6c64/59d7cadaa210e3310f3167dc181824fb1bff.pdf, Jun 2013. URL https://pdfs.semanticscholar.org/6c64/59d7cadaa210e3310f3167dc181824fb1bff.pdf. 5, 15

Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. *Causation, prediction, and search*. MIT press, 2000. 1

Yi Sun, Xiaogang Wang, and Xiaoou Tang. Hybrid deep learning for face verification. In *Proc. IEEE International Conf. on Computer Vision (ICCV)*, pp. 1489–96, Dec 2013. doi: 10.1109/ICCV.2013.188. 1

Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1701–08, 2014. doi: 10.1109/CVPR.2014.220. 1

Yichuan Tang, Ruslan Salakhutdinov, and Geoffrey Hinton. Tensor analyzers. volume 28 of *Proceedings of Machine Learning Research*, pp. 163–171, Atlanta, Georgia, USA, 17–19 Jun 2013. URL http://proceedings.mlr.press/v28/tang13.html. 2

Eric J Topol. High-performance medicine: the convergence of human and artificial intelligence. *Nature medicine*, 25(1):44–56, 2019. 1

L. R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31:279–311, 1966. 1

M. A. O. Vasilescu. Human motion signatures: Analysis, synthesis, recognition. In *Proc. Int. Conf. on Pattern Recognition*, volume 3, pp. 456–460, Quebec City, Aug 2002. 1

M. A. O. Vasilescu. Multilinear projection for face recognition via canonical decomposition. In *Proc. IEEE Inter. Conf. on Automatic Face Gesture Recognition (FG 2011)*, pp. 476–483, Mar 2011. doi: 10.1109/FG.2011.5771445. 1, 16

M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: TensorFaces. In *Proc. European Conf. on Computer Vision (ECCV 2002)*, pp. 447–460, Copenhagen, Denmark, May 2002a. 1, 3, 14, 15

M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis for facial image recognition. In *Proc. Int. Conf. on Pattern Recognition*, volume 2, pp. 511–514, Quebec City, Aug 2002b. 1, 8

M. A. O. Vasilescu and D. Terzopoulos. TensorTextures: Multilinear image-based rendering. *ACM Transactions on Graphics*, 23(3):336–342, Aug 2004. Proc. ACM SIGGRAPH 2004 Conf., Los Angeles, CA. 1

M. A. O. Vasilescu and D. Terzopoulos. Multilinear independent components analysis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume I, pp. 547–553, San Diego, CA, 2005. 1, 4, 8

M. A. O. Vasilescu and D. Terzopoulos. Multilinear projection for appearance-based recognition in the tensor framework. In *Proc. 11th IEEE Inter. Conf. on Computer Vision (ICCV'07)*, pp. 1–8, 2007. 1, 3, 8, 16

M. Alex O. Vasilescu. *A Multilinear (Tensor) Algebraic Framework for Computer Graphics, Computer Vision, and Machine Learning*. PhD dissertation, University of Toronto, 2009. 1, 2, 3, 4, 14

M. Alex O. Vasilescu and Eric Kim. Compositional hierarchical tensor factorization: Representing hierarchical intrinsic and extrinsic causal factors. In *The 25th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD19): Tensor Methods for Emerging Data Science Challenges Workshop*, Aug. 5 2019. 2, 7

M. Alex O. Vasilescu, Eric Kim, and Xiao S. Zeng. Causalx: Causal explanations and block multilinear factor analysis. In *2020 25th International Conference of Pattern Recognition (ICPR 2020)*, pp. 10736–10743, Jan 2021. 1, 2, 7, 8, 9

Joshua Vendrow, Jamie Haddock, and Deanna Needell. A generalized hierarchical nonnegative tensor decomposition, 2021. 2

D. Vlasic, M. Brand, H. Pfister, and J. Popovic. Face transfer with multilinear models. *ACM Transactions on Graphics (TOG)*, 24(3):426–433, Jul 2005. URL http://people.csail.mit.edu/jovan/assets/papers/vlasic-2005-ftm.pdf. 1

Hongcheng Wang and Narendra Ahuja. Facial expression decomposition. In *Proc, 9th IEEE Inter. Conf. on Computer Vision (ICCV)*, pp. 958–65,v.2, 2003. doi: 10.1109/ICCV.2003.1238452. 1

M. Wang, Y. Panagakis, P. Snape, and S. Zafeiriou. Learning the multilinear structure of visual data. In *2017 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 6053–6061, Jul 2017. doi: 10.1109/CVPR.2017.641. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.641. 2

C. Xiong, L. Liu, X. Zhao, S. Yan, and T. K. Kim. Convolutional fusion network for face verification in the wild. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(3):517–528, Mar 2016. ISSN 1051-8215. doi: 10.1109/TCSVT.2015.2406191. 1

J. Yang, X. Gao, D. Zhang, and J. Yang. Kernel ICA: An alternative formulation and its application to face recognition. *Pattern Recognition*, 38(10):1784–87, 2005. 7

M. Yang. Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. In *Proc. 5th IEEE Inter. Conf. on Automatic Face Gesture Recognition*, pp. 215–220, Washington, DC, May 2002. doi: 10.1109/AFGR.2002.4527207. 7

A Mathematical Background

Throughout this article, we will denote scalars by lower case italic letters $(a, b, ...)$, vectors by bold lower case letters $(\mathbf{a}, \mathbf{b}, ...)$, matrices by bold uppercase letters $(\mathbf{A}, \mathbf{B}, ...)$, and higher-order tensors by bold uppercase calligraphic letters $(\mathcal{A}, \mathcal{B}, ...)$. Index upper bounds are denoted by italic uppercase letters (*i.e.*, $1 \leq a \leq A$ or $1 \leq i \leq I$). The zero matrix is denoted by $\mathbf{0}$, and the identity matrix is denoted by $\mathbf{I}$. The TensorFaces paper [Vasilescu & Terzopoulos (2002a) is a gentle introduction to tensor factor analysis, [Kolda & Bader (2009) is a great survey of tensor methods and references [Vasilescu (2009); de Lathauwer (1997); Bro (1997) provide an in depth treatment of tensor factor analysis.

A.1 PCA computation with Linear Autoencoder

An autoencoder-decoder that minimizes the reconstruction loss function for a set of observations, $\mathbf{d}_{\mathrm{i}} \in \mathbb{C}^{I_0}$,

$$l \quad = \quad \sum_{\mathrm{i}=1}^{I} \|\mathbf{d}_{\mathrm{i}} - \mathbf{B}\mathbf{c}_{\mathrm{i}}\| + \lambda \|\mathbf{B}^{\mathrm{T}}\mathbf{B} - \mathbf{I}\|, \tag{13}$$

and has a linear decoder learns a set of weights, $b_{i_0,r}$, that are identical to the elements of the PCA basis matrix, $\mathbf{B} \in \mathbb{C}^{I_0 \times R}$, when the weights of each neuron are computed sequentially, Fig. 6. An autoencoder is implemented with a cascade of Hebb neurons [Hebb (1949). The contribution of each neuron, $c_1, \ldots, c_{\mathrm{r}}$, is sequentially computed and subtracted from a centered training data set, and the difference is driven through the next Hebb neuron, $c_{\mathrm{r}+1}$ [Sejnowski et al. (1989); Sanger (1989); Rumelhart et al. (1986); Ackley et al. (1985); Oja (1982).

The weights of a Hebb neuron, c_{r}, are updated by

$$\begin{aligned} \Delta \mathbf{b}_{\mathrm{r}}(t+1) &= \eta \left(\mathbf{d} - \sum_{i_{\mathrm{r}}=1}^{r} \mathbf{b}_{i_{\mathrm{r}}}(t) c_{i_{\mathrm{r}}}(t) \right) c_{\mathrm{r}}(t) \qquad (14) \\ &= \eta \left(\mathbf{d} - \sum_{i_{\mathrm{r}}=1}^{r} \mathbf{b}_{i_{\mathrm{r}}}(t) \mathbf{b}_{i_{\mathrm{r}}}^{\mathrm{T}}(t) \mathbf{d} \right) \mathbf{d}^{\mathrm{T}} \mathbf{b}_r(t), \\ \mathbf{b}_{\mathrm{r}}(t+1) &= \frac{(\mathbf{b}_{\mathrm{r}}(t) + \Delta \mathbf{b}_{\mathrm{r}}(t+1))}{\|\mathbf{b}_{\mathrm{r}}(t) + \Delta \mathbf{b}_{\mathrm{r}}(t+1)\|} \end{aligned}$$

where $\mathbf{d} \in \mathbb{C}^{I_0}$ is a vectorized centered observation with I_0 measurements, η is the learning rate, $\mathbf{b}_{\mathrm{r}}$ are the autoencoder weights of the r neuron, c_{r} is the activation, and t is the time iteration. Back-propagation [LeCun et al. (1988; 2012) performs PCA gradient descent [Jolliffe (1986).

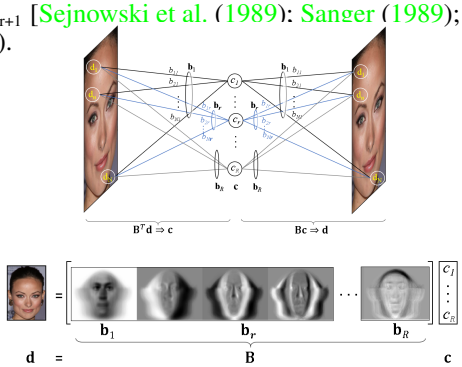

Figure 6: Autoencoder-decoder architecture and Principal Component Analysis. (All images have been vectorized, but they are displayed as a grid of numbers. The eigenvector $\mathbf{b}_1$ is the mean and activation c_1 is set to 1.)

A.2 Relevant Tensor Algebra

Briefly, the natural generalization of matrices (i.e., linear operators defined over a vector space), tensors define multilinear operators over a *set* of vector spaces. A *"data tensor"* denotes an M-way data array.

Definition 1 (Tensor) *Tensors are multilinear mappings over a set of vector spaces, $\mathbb{C}^{I_m}$, $1 \leq m \leq M$, to a range vector space $\mathbb{C}^{I_0}$:*

$$\mathcal{A} : \left\{ \mathbb{C}^{I_1} \times \mathbb{C}^{I_2} \times \cdots \times \mathbb{C}^{I_M} \right\} \mapsto \mathbb{C}^{I_0}. \tag{15}$$

The order of tensor $\mathcal{A} \in \mathbb{C}^{I_0 \times I_1 \times \cdots \times I_M}$ is $M+1$. An element of $\mathcal{A}$ is denoted as $\mathcal{A}_{i_0 i_1 \ldots i_m \ldots i_M}$ or $a_{i_0 i_1 \ldots i_m \ldots i_M}$, where $1 \leq i_m \leq I_m$.

The mode-m vectors of an M-order tensor $\mathcal{A} \in \mathbb{C}^{I_0 \times I_1 \times \cdots \times I_M}$ are the I_m-dimensional vectors obtained from $\mathcal{A}$ by varying index i_m while keeping the other indices fixed. In tensor terminology, column vectors are the mode-0 vectors and row vectors as mode-1 vectors. The mode-m vectors of a tensor are also known as *fibers*. The mode-m vectors are the column vectors of matrix $\mathbf{A}_{[m]}$ that results from *matrixizing* (a.k.a. *flattening*) the tensor $\mathcal{A}$.

Definition 2 (Mode-m Matrixizing) *The mode-m matrixizing of tensor $\mathcal{A} \in \mathbb{C}^{I_0 \times I_1 \times \ldots I_M}$ is defined as the matrix $\mathbf{A}_{[m]} \in \mathbb{C}^{I_m \times (I_0 \ldots I_{m-1} I_{m+1} \ldots I_M)}$. As the parenthetical ordering indicates, the*

Algorithm 3 M-mode SVD algorithm.

Input the data tensor $\mathcal{D} \in \mathbb{C}^{I_0 \times \cdots \times I_M}$.

1. For $m := 0, \ldots, M$,
   Let $\mathbf{U}_m$ be the left orthonormal matrix of $[\mathbf{U}_\mathrm{m}\mathbf{S}_\mathrm{m}\mathbf{V}_\mathrm{m}^\mathrm{T}] := \mathrm{svd}(\mathbf{D}_{[m]})$[a]
2. Set $\mathcal{Z} := \mathcal{D} \times_0 \mathbf{U}_0{}^\mathrm{T} \times_1 \mathbf{U}_1{}^\mathrm{T} \cdots \times_m \mathbf{U}_m{}^\mathrm{T} \ldots \times_M \mathbf{U}_M{}^\mathrm{T}$.

Output mode matrices $\mathbf{U}_0, \mathbf{U}_1 \ldots, \mathbf{U}_M$, and the core tensor $\mathcal{Z}$.

[a]The computation of $\mathbf{U}_m$ in the SVD $\mathbf{D}_{[m]} = \mathbf{U}_m \mathbf{\Sigma} \mathbf{V}_m{}^\mathrm{T}$ can be performed efficiently, depending on which dimension of $\mathbf{D}_{[m]}$ is smaller, by decomposing either $\mathbf{D}_{[m]}\mathbf{D}_{[m]}{}^\mathrm{T} = \mathbf{U}_m \mathbf{\Sigma}^2 \mathbf{U}_m{}^\mathrm{T}$ (note that $\mathbf{V}_m{}^\mathrm{T} = \mathbf{\Sigma}^+ \mathbf{U}_m{}^\mathrm{T} \mathbf{D}_{[m]}$) or by decomposing $\mathbf{D}_{[m]}{}^\mathrm{T}\mathbf{D}_{[m]} = \mathbf{V}_m \mathbf{\Sigma}^2 \mathbf{V}_m{}^\mathrm{T}$ and then computing $\mathbf{U}_m = \mathbf{D}_{[m]}\mathbf{V}_m\mathbf{\Sigma}^+$.

mode-m column vectors are arranged by sweeping all the other mode indices through their ranges, with smaller mode indexes varying more rapidly than larger ones; thus,

$$[\mathbf{A}_{[m]}]_{jk} = a_{i_1 \ldots i_m \ldots i_M}, \quad \textit{where} \tag{16}$$

$$j = i_m \quad \textit{and} \quad k = 1 + \sum_{\substack{n=0 \\ n \neq m}}^{M} (i_n - 1) \prod_{\substack{l=0 \\ l \neq m}}^{n-1} I_l.$$

A generalization of the product of two matrices is the product of a tensor and a matrix [de Lathauwer (1997); Carroll et al. (1980).

Definition 3 (Mode-m Product, $\times_\mathrm{m}$) *The mode-m product of a tensor $\mathcal{A} \in \mathbb{C}^{I_1 \times I_2 \times \cdots \times I_m \times \cdots \times I_M}$ and a matrix $\mathbf{B} \in \mathbb{C}^{J_m \times I_m}$, denoted by $\mathcal{A} \times_m \mathbf{B}$, is a tensor of dimensionality $\mathbb{C}^{I_1 \times \cdots \times I_{m-1} \times J_m \times I_{m+1} \times \cdots \times I_M}$ whose entries are computed by*

$$[\mathcal{A} \times_m \mathbf{B}]_{i_1 \ldots i_{m-1} j_m i_{m+1} \ldots i_M} = \sum_{i_m} a_{i_1 \ldots i_{m-1} i_m i_{m+1} \ldots i_M} b_{j_m i_m},$$

$$\mathcal{C} = \mathcal{A} \times_m \mathbf{B}. \quad \underset{\textit{tensorize}}{\overset{\textit{matrixize}}{\rightleftarrows}} \quad \mathbf{C}_{[m]} = \mathbf{B}\mathbf{A}_{[m]}.$$

The M-mode SVD, Alg. 1 [Vasilescu & Terzopoulos (2002a) is a "generalization" of the conventional matrix (i.e., 2-mode) SVD which may be written in tensor notation as

$$\mathbf{D} = \mathbf{U}_0 \mathbf{S} \mathbf{U}_1{}^\mathrm{T} \quad \Leftrightarrow \quad \mathbf{D} = \mathbf{S} \times_0 \mathbf{U}_0 \times_1 \mathbf{U}_1$$

The M-mode SVD orthogonalizes the M spaces and decomposes a tensor as the *mode-m product*, denoted $\times_m$, of M-orthonormal mode matrices, and a core tensor $\mathcal{Z}$

$$\mathcal{D} = \mathcal{Z} \times_0 \mathbf{U}_0 \cdots \times_m \mathbf{U}_m \cdots \times_M \mathbf{U}_M. \tag{17}$$

$$\mathbf{D}_{[m]} = \mathbf{U}_\mathrm{m} \mathbf{Z}_{[m]} \left(\mathbf{U}_M \cdots \otimes \mathbf{U}_{\mathrm{m+1}} \otimes_{\mathrm{m-1}} \mathbf{U} \cdots \otimes \mathbf{U}_0\right)^\mathrm{T}, \tag{18}$$

$$vec(\mathcal{D}) = (\mathbf{U}_M \cdots \otimes \mathbf{U}_{\mathrm{m+1}} \otimes \mathbf{U}_{\mathrm{m-1}} \cdots \otimes \mathbf{U}_0)\, vec(\mathcal{Z}). \tag{19}$$

The latter two equations express the decomposition in matrix form and in terms of vec operators.

A.3 Compositional Hierarchical Block TensorFaces

Training Data: In our experiments, we employed gray-level facial training images rendered from 3D scans of 100 subjects. The scans were recorded using a CyberwareTM 3030PS laser scanner and are part of the 3D morphable faces database created at the University of Freiburg [Blanz & Vetter (1999). Each subject was combinatoriall y imaged in Maya from 15 different viewpoints ($\theta = -60°$ to $+60°$ in $10°$ steps on the horizontal plane, $\phi = 0°$) with 15 different illuminations ($\theta = -35°$ to $+35°$ in $5°$ increments on a plane inclined at $\phi = 45°$).

Data Preprocessing: Facial images were warped to an average face template by a piecewise affine transformation given a set of facial landmarks obtained by employing Dlib software [King (2009); Kazemi & Sullivan (2014); Si et al. (2013); Macedo et al. (2006); Hatamizadeh et al. (2019). Illumination was normalized with an adaptive contrast histogram equalization algorithm, but rather than performing contrast correction on the entire image, subtiles of the image were contrast normalized, and tiling artifacts were eliminated through interpolation. Histogram clipping was employed to avoid over-saturated regions.

Experiments: Each image, $\mathbf{d} \in \mathbb{R}^{I_0 \times 1}$, was convolved with five filters banks $\{\mathbf{H}_s \| s = 1...S\}$. The filtered images, $\mathbf{d} \times_0 \mathbf{H}_s$, resulted in five facial part hierarchies composed of (i) independent pixel parts (ii) parts segmented from different layers of a Gaussian pyramid that were equally or (iii) unequally weighed, (iv) parts were segmented from a Laplacian pyramid that were equally or (v) unequally weighed. We ran five experiments with five facial part hierarchies from which a person representation was computed, Fig. 8. The composite person signature was computed for every test image by employing the multilinear projection algorithm [Vasilescu (2011); Vasilescu & Terzopoulos (2007), and signatures were compared with a nearest neighbor classifier.

To validate the effectiveness of our system on real-world images, we report results on "LFW" dataset (*LFW*) [Huang et al. (2007). This dataset contains 13,233 facial images of 5,749 people. The photos are unconstrained (*i.e.*, "in the wild"), and include variation due to pose, illumination, expression, and occlusion. The dataset consists of 10 train/test splits of the data. We report the mean accuracy and standard deviation across all splits in Table 2. Fig. 8(b-c) depicts the experimental ROC curves. We follow the supervised *"Unrestricted, labeled outside data"* paradigm.

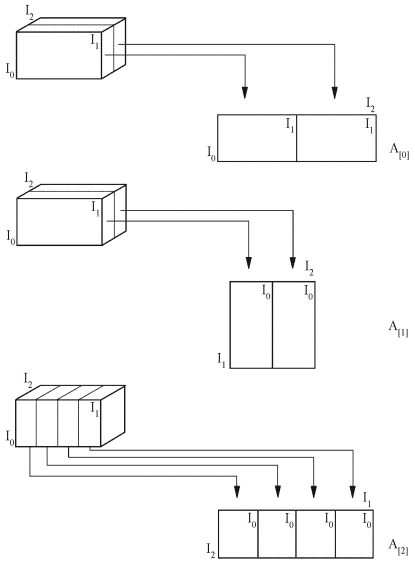

Figure 7: Matrixizing a 3rd order tensor, $\mathcal{A}$. The tensor can be matrixized in 3 ways.

Results: While we cannot celebrate closing the gap on human performance, our results are promising. DeepFace, a CNN model, improved the prior art verification rates on LFW from 70% to 97.35%, by training on $4.4M$ images of 200×200 pixels from $4,030$ people, the same order of magnitude as the number of people in the LFW database.

We trained on less than one percent (1%) of the 4.4M total images used to train DeepFace. Images were rendered from 3D scans of 100 subjects with an the intraocular distance of approximately 20 pixels and with a facial region captured by $10,414$ pixels (image size $\approx 100 \times 100$ pixels). We have currently achieved verification rates just shy of 80% on LFW. When data is limited, CNN models do not convergence or generalize.

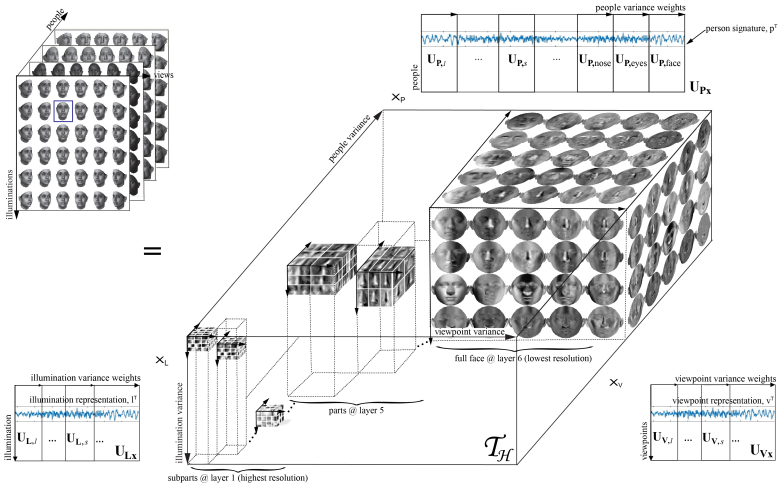

Figure 8: Compositional hierarchical Block TensorFaces learns a hierarchy of features, and reesents each person as a part-based compositional representation. Figure depicts the training data factorization, $\mathcal{D} = \mathcal{T}_{\mathcal{H}} \times_{\mathrm{L}} \mathbf{U}_{\mathrm{L}} \times_{\mathrm{V}} \mathbf{U}_{\mathrm{V}} \times_{\mathrm{P}} \mathbf{U}_{\mathrm{P}}$, where an observation is represented as $\mathbf{d}(\mathbf{p}, \mathbf{v}, \mathbf{l}) = \mathcal{T}_{\mathcal{H}} \times_{\mathrm{L}} \mathbf{l} \times_{\mathrm{V}} \mathbf{v} \times_{\mathrm{P}} \mathbf{p}$ and $\mathcal{T}_{\mathcal{H}}$ spans the *h*ierarchical causal factor variance.

Test Dataset	PCA	TensorFaces	compositional hierarchical Block TensorFaces				
			Pixels	Gaussian Pyramid	Weighted Gaussian Pyramid	Laplacian Pyramid	Weighted Laplacian Pyramid
Freiburg	65.23%	71.64%	90.50%	88.17%	94.17%	90.96%	93.98%
LFW	69.23% ±1.51	66.25% ±1.60	72.72% ±2.14	76.72% ±1.65	77.85% ±1.83	77.58% ±1.45	78.93% ±1.77

Table 2: Empirical results reported for LFW : PCA, TensorFaces and compositional hierarchical Block TensorFaces. *Pixels* denotes independent facial part analysis *Gaussian/Laplacian* use a multi resolution pyramid to analyze facial features at different scales. *Weighted* denotes a weighted composite signature.
Freiburg Experiment:
Train on Freiburg: 6 views (±60°,±30°,±5°); 6 illuminations (±60°,±30°,±5°), 45 people
Test on Freiburg: 9 views (±50°, ±40°, ±20°, ±10°, 0°), 9 illums (±50°, ±40°, ±20°, ±10°, 0°), 45 different people
LFW Experiment: Models were trained on approximately half of one percent ($0.5\% < 1\%$) of the 4.4M images used to train DeepFace.
Train on Freiburg:
15 views (±60°,±50°, ±40°,±30°, ±20°, ±10°,±5°, 0°), 15 illuminations (±60°,±50°, ±40°,±30°, ±20°, ±10°,±5°, 0°), 100 people
Test on LFW: We report the mean accuracy and standard deviation across standard literature partitions [Huang et al. (2007), following the
Unrestricted, labeled outside data supervised protocol.

Summary: This paper contributes to the tensor algebraic paradigm and models cause-and-effect as a hierarchical block tensor interaction between intrinsic and extrinsic hierarchical causal factors of data formation.

A data tensor expressed as a function of a hierarchical data tensor is a unified tensor model of wholes and parts from which a new compositional hierarchical block tensor factorization was derived. The resulting causal factor representations are interpretable, hierarchical, and statistically invariant to all other causal factors. Our approach was demonstrated in the context of facial images by training on a very small set of synthetic images. While we have not closed the gap on human performance, we report encouraging face verification results on two test data sets–the Freiburg, and the Labeled Faces in the Wild datasets. CNN verification rates improved the 70% prior art to 97.35% when they employed 4.4M images from $4,030$ people, the same order of magnitude as the number of people in the LFW database. We have currently achieved verification rates just shy of 80% on LFW by employing synthetic images from 100 people for a total of less than one percent (1%) of the total images employed by DeepFace. By comparison, when data is limited, CNN models do not convergence, or generalize.

REFERENCES

Evrim Acar, Evangelos E Papalexakis, Gözde Gürdeniz, Morten A Rasmussen, Anders J Lawaetz, Mathias Nilsson, and Rasmus Bro. Structure-revealing data fusion. *BMC bioinformatics*, 15(1):239, 2014. 1

David H. Ackley, Geoffrey A. Hinton, and Terrence J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985. ISSN 0364-0213. doi: https://doi.org/10.1016/S0364-0213(85)80012-4. URL https://www.sciencedirect.com/science/article/pii/S0364021385800124. 3, 14

Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M. Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. 15(1):27732832, 2014. 2, 5

Elias Bareinboim and Judea Pearl. Causal inference and the data-fusion problem. *Proc. of the National Academy of Sciences*, 113(27):7345–52, 2016. ISSN 0027-8424. doi: 10.1073/pnas.1510507113. URL https://www.pnas.org/content/113/27/7345. 1

M.S. Bartlett, J.R. Movellan, and T.J. Sejnowski. Face recognition by independent component analysis. *IEEE Transactions on Neural Networks*, 13(6):1450–64, 2002. 5

Jacob Benesty, Constantin Paleologu, Laura-Maria Dogariu, and Silviu Ciochin. Identification of linear and bilinear systems: A unified study. *Electronics*, 10(15), 2021. ISSN 2079-9292. doi: 10.3390/electronics10151790. URL https://www.mdpi.com/2079-9292/10/15/1790. 2

Peter M. Bentler and Sik-Yum Lee. A statistical development of three-mode factor analysis. *British J. of Math. and Stat. Psych.*, 32(1):87–104, 1979. doi: 10.1111/j.2044-8317.1979.tb00754.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.2044-8317.1979.tb00754.x. 1

V. Blanz and T. A. Vetter. Morphable model for the synthesis of 3D faces. In *Proc. ACM SIGGRAPH 99 Conf.*, pp. 187–194, 1999. 15

Kenneth A. Bollen and Judea Pearl. *Eight Myths About Causality and Structural Equation Models*, pp. 301–328. Springer Netherlands, Dordrecht, 2013. ISBN 978-94-007-6094-3. doi: 10.1007/978-94-007-6094-3_15. URL https://doi.org/10.1007/978-94-007-6094-3_15. 1

Rasmus Bro. Parafac: Tutorial and applications. In *Chemom. Intell. Lab Syst., Special Issue 2nd Internet Cont. in Chemometrics (INCINC'96)*, volume 38, pp. 149–171, 1997. 1, 3, 14

J. D. Carroll and J. J. Chang. Analysis of individual differences in multidimensional scaling via an N-way generalization of 'Eckart-Young' decomposition. *Psychometrika*, 35:283–319, 1970. 1

J. D. Carroll, S. Pruzansky, and J. B. Kruskal. CANDELINC: A general approach to multidimensional analysis of many-way arrays with linear constraints on parameters. *Psychometrika*, 45:3–24, 1980. 15

J. C. Chen, R. Ranjan, A. Kumar, C. H. Chen, V. M. Patel, and R. Chellappa. An end-to-end system for unconstrained face verification with deep convolutional neural networks. In *IEEE International Conf. on Computer Vision Workshop (ICCVW)*, pp. 360–368, Dec 2015. doi: 10.1109/ICCVW.2015.55. 1

Wei Chu and Zoubin Ghahramani. Probabilistic models for incomplete multi-dimensional arrays. volume 5 of *Proceedings of Machine Learning Research*, pp. 89–96, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR. URL http://proceedings.mlr.press/v5/chu09a.html. 2

Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009. 1

Pierre Common. Independent component analysis, a new concept? *Signal Processing*, 36:287–314, 1994. URL https://hal.archives-ouvertes.fr/hal-00417283. 5

T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001. 5

J. Davis and H. Gao. Recognizing human action efforts: An adaptive three-mode PCA framework. In *Proc. IEEE Inter. Conf. on Computer Vision, (ICCV)*, pp. 1463–69, Nice, France, Oct 13-16 2003. 1

L. de Lathauwer. Decompositions of a higher-order tensor in block termspart ii: Definitions and uniqueness. *SIAM J. on Matrix Analysis and Applications*, 30(3):1033–1066, 2008. doi: 10.1137/070690729. URL https://doi.org/10.1137/070690729. 1, 2, 7

L. De Lathauwer, B. De Moor, and J. Vandewalle. Independent component analysis based on higher-order statistics only. In *Proceedings of 8th Workshop on Statistical Signal and Array Processing*, pp. 356–359, 1996. doi: 10.1109/SSAP.1996.534890. 5

Lieven de Lathauwer. *Signal Processing Based on Multilinear Algebra*. PhD dissertation, Katholieke Univ. Leuven, Belgium, 1997. 1, 3, 14, 15

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977. doi: https://doi.org/10.1111/j.2517-6161.1977.tb01600.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1977.tb01600.x. 2

Andrew Gelman and Guido Imbens. Why ask why? forward causal inference and reverse causal questions. Tech.report, Nat.Bureau of Econ Research, 2013. 8

Lars Grasedyck. Hierarchical singular value decomposition of tensors. *SIAM J. on Matrix Analysis and Applications*, 31(4):2019–54, 2010. 3

Wolfgang Hackbusch and Stefan Kühn. A new scheme for the tensor representation. *Journal of Fourier Analysis and Applications*, 15(5):706–722, 2009. 3

J. Hadamard. *Lectures on Cauchy's Problem in Linear Partial Differential Equations*. Dover Publisher, 1952. 8

R. Harshman. Foundations of the PARAFAC procedure: Model and conditions for an explanatory factor analysis. Tech. Report Working Papers in Phonetics 16, UCLA, CA, Dec 1970. 1

Ali Hatamizadeh, Demetri Terzopoulos, and Andriy Myronenko. End-to-end boundary aware networks for medical image segmentation. In *Inter. Workshop on Machine Learning in Medical Imaging*, pp. 187–194. Springer, 2019. 15

D. O. Hebb. *The organization of behavior: A neuropsychological theory*. John Wiley And Sons, Inc., New York, 1949. 3, 14

Geoffrey Hinton. How to represent part-whole hierarchies in a neural network, 2021. 2

Geoffrey E. Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders, 2011. 2

Paul W. Holland. Statistics and causal inference: Rejoinder. *J. of the American Statistical Association*, 81(396): 968970, 1986. URL https://users.nber.org/~rdehejia/!@SAEM/Topic01Causality/holland_JASA_1986.pdf. 1

Eugene Hsu, Kari Pulli, and Jovan Popovic. Style translation for human motion. *ACM Transactions on Graphics*, 24(3):1082–89, 2005. URL http://people.csail.mit.edu/jovan/assets/papers/hsu-2005-stf.pdf. 1

G. B. Huang. Learning hierarchical representations for face verification with convolutional deep belief networks. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 2518–25, Jun 2012. doi: 10.1109/CVPR.2012.6247968. 1

Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, Oct 2007. 16, 17

Guido W. Imbens. Potential outcome and directed acyclic graph approaches to causality: Relevance for empirical practice in economics. *Journal of Economic Literature*, 58(4):1129–79, December 2020. doi: 10.1257/jel.20191597. URL https://www.aeaweb.org/articles?id=10.1257/jel.20191597. 1

Guido W. Imbens and Donald B. Rubin. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge Univ. Press, 2015. ISBN 0521885884. 1

Mark A Iwen, Deanna Needell, Elizaveta Rebrova, and Ali Zare. Lower memory oblivious (tensor) subspace embeddings with fewer random bits: modewise methods for least squares. *SIAM Journal on Matrix Analysis and Applications*, 42(1):376–416, 2021. 2

I.T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986. 3, 14

A. Kapteyn, H. Neudecker, and T. Wansbeek. An approach to n-mode component analysis. *Psychometrika*, 51 (2):269–275, Jun 1986. 1

Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, CVPR '14, pp. 1867–74, Washington, DC, USA, 2014. IEEE Computer Society. ISBN 978-1-4799-5118-5. doi: 10.1109/CVPR.2014.241. URL http://dx.doi.org/10.1109/CVPR.2014.241. 15

Daniel S. Kermany, Michael Goldbaum, Wenjia Cai, Carolina C.S. Valentim, Huiying Liang, Sally L. Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, Justin Dong, Made K. Prasadha, Jacqueline Pei, Magdalene Y.L. Ting, Jie Zhu, Christina Li, Sierra Hewett, Jason Dong, Ian Ziyar, Alexander Shi, Runze Zhang, Lianghong Zheng, Rui Hou, William Shi, Xin Fu, Yaou Duan, Viet A.N. Huu, Cindy Wen, Edward D. Zhang, Charlotte L. Zhang, Oulan Li, Xiaobo Wang, Michael A. Singer, Xiaodong Sun, Jie Xu, Ali Tafreshi, M. Anthony Lewis, Huimin Xia, and Kang Zhang. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5):1122–1131.e9, 2018. ISSN 0092-8674. doi: https://doi.org/10.1016/j.cell.2018.02.010. URL https://www.sciencedirect.com/science/article/pii/S0092867418301545. 1

Valentin Khrulkov. *Geometrical Methods in Machine Learning and Tensor Analysis*. PhD dissertation, Skolkovo Institute of Science and Technology, 2020. 2

Yong-Deok Kim, Eunhyeok Park, Sungjoo Yoo, Taelim Choi, Lu Yang, and Dongjun Shin. Compression of deep convolutional neural networks for fast and low power mobile applications. *CoRR*, abs/1511.06530, 2015. URL http://arxiv.org/abs/1511.06530. 2

Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009. 15

Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009. 3, 14

Tamara G. Kolda, Brett W. Bader, and Joseph P. Kenny. Higher-order web link analysis using multilinear algebra. In *ICDM 2005: Proceedings of the 5th IEEE International Conference on Data Mining*, pp. 242–249, 2005. doi: 10.1109/ICDM.2005.77. 2

Jean Kossaifi, Aran Khanna, Zachary C. Lipton, Tomasso Furlanello, and Anima Anandkumar. Tensor contraction layers for parsimonious deep nets. In *Computer Vision and Pattern Recognition (CVPR), Tensor Methods in Computer Vision Workshop*, pp. 1940–46, Jul 2017. 2

M. A. Kramer. Nonlinear principal components analysis using autoassociative neural networks. *AIChE J.*, 32(2): 233–243, Feb 1991. 7

P. M. Kroonenberg and J. de Leeuw. Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika*, 45:69–97, 1980. 1

Lieven De Lathauwer, P Comon, Bart De Moor, and Joos Vandewalle. Higher-order power method - application in independent component analysis. *Proc. of the International Symposium on Nonlinear Theory and its Applications (NOLTA'95)*, pp. 91–96, 1995. 5

Vadim Lebedev, Yaroslav Ganin, Maksim Rakhuba, Ivan V. Oseledets, and Victor S. Lempitsky. Speeding-up convolutional neural networks using fine-tuned cp-decomposition. *CoRR*, abs/1412.6553, 2014. URL http://arxiv.org/abs/1412.6553. 2

Yann LeCun, D Touresky, G Hinton, and T Sejnowski. A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*, volume 1, pp. 21–28, 1988. 3, 14

Yann A. LeCun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. *Efficient BackProp*, pp. 9–48. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-35289-8. doi: 10.1007/978-3-642-35289-8_3. URL https://doi.org/10.1007/978-3-642-35289-8_3. 3, 14

I. Macedo, E. V. Brazil, and L. Velho. Expression transfer between photographs through multilinear aam's. pp. 239–246, Oct 2006. doi: 10.1109/SIBGRAPI.2006.18. 15

Ali Madani, Mehdi Moradi, Alexandros Karargyris, and Tanveer Syeda-Mahmood. Semi-supervised learning with generative adversarial networks for chest x-ray classification with ability of data domain adaptation. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pp. 1038–1042, 2018. doi: 10.1109/ISBI.2018.8363749. 1

J. R. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. John Wiley & Sons, 1988. 1

Alexander Novikov, Dmitrii Podoprikhin, Anton Osokin, and Dmitry P Vetrov. Tensorizing neural networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 28*, pp. 442–450. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5787-tensorizing-neural-networks.pdf. 2

Erkki Oja. A simplified neuron model as a principal component analyzer. 15:267–2735, 1982. 3, 14

Charles C. Onu, Jacob E. Miller, and Doina Precup. A fully tensorized recurrent neural network. *CoRR*, abs/2010.04196, 2020. URL https://arxiv.org/abs/2010.04196. 2

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge Univ. Press, 2000. ISBN 9780521773621. URL https://books.google.com/books?id=wnGU_TsW3BQC. 1

Judea Pearl and Elias Bareinboim. External validity: From do-calculus to transportability across populations. *Statistical Science*, 29(4):57995, 2014. 1

I. Perros, R. Chen, R. Vuduc, and J. Sun. Sparse hierarchical tucker factorization and its application to healthcare. In *Proc. IEEE Inter. Conf. on Data Mining*, pp. 943–948, Nov 2015. doi: 10.1109/ICDM.2015.29. 3

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. *Learning internal representations by error propagation*. 1986. 3, 14

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/2cad8fa47bbef282badbb8de5374b894-Paper.pdf. 2

Terry Sanger. Optimal unsupervised learnig in a single layer linear feedforward neural network. 12:459–473, 1989. 3, 14

B. Schölkoph, A. Smola, and K.-R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998. 7

Terrence Sejnowski, Sumantra Chattarji, and Patric Sfanton. *Induction of Synaptic Plasticity by Hebbian Covariance in the Hippocampus*, pp. 105–124. Addison-Wesley, 1989. 3, 14

Weiguang Si, Kota Yamaguchi, and M. Alex O. Vasilescu. Face tracking with multilinear (tensor) active appearance models. http://pdfs.semanticscholar.org/6c64/59d7cadaa210e3310f3167dc181824fb1bff.pdf, Jun 2013. URL https://pdfs.semanticscholar.org/6c64/59d7cadaa210e3310f3167dc181824fb1bff.pdf. 5, 15

Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. *Causation, prediction, and search*. MIT press, 2000. 1

Yi Sun, Xiaogang Wang, and Xiaoou Tang. Hybrid deep learning for face verification. In *Proc. IEEE International Conf. on Computer Vision (ICCV)*, pp. 1489–96, Dec 2013. doi: 10.1109/ICCV.2013.188. 1

Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1701–08, 2014. doi: 10.1109/CVPR.2014.220. 1

Yichuan Tang, Ruslan Salakhutdinov, and Geoffrey Hinton. Tensor analyzers. volume 28 of *Proceedings of Machine Learning Research*, pp. 163–171, Atlanta, Georgia, USA, 17–19 Jun 2013. URL http://proceedings.mlr.press/v28/tang13.html. 2

Eric J Topol. High-performance medicine: the convergence of human and artificial intelligence. *Nature medicine*, 25(1):44–56, 2019. 1

L. R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31:279–311, 1966. 1

M. A. O. Vasilescu. Human motion signatures: Analysis, synthesis, recognition. In *Proc. Int. Conf. on Pattern Recognition*, volume 3, pp. 456–460, Quebec City, Aug 2002. 1

M. A. O. Vasilescu. Multilinear projection for face recognition via canonical decomposition. In *Proc. IEEE Inter. Conf. on Automatic Face Gesture Recognition (FG 2011)*, pp. 476–483, Mar 2011. doi: 10.1109/FG.2011.5771445. 1, 16

M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: TensorFaces. In *Proc. European Conf. on Computer Vision (ECCV 2002)*, pp. 447–460, Copenhagen, Denmark, May 2002a. 1, 3, 14, 15

M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis for facial image recognition. In *Proc. Int. Conf. on Pattern Recognition*, volume 2, pp. 511–514, Quebec City, Aug 2002b. 1, 8

M. A. O. Vasilescu and D. Terzopoulos. TensorTextures: Multilinear image-based rendering. *ACM Transactions on Graphics*, 23(3):336–342, Aug 2004. Proc. ACM SIGGRAPH 2004 Conf., Los Angeles, CA. 1

M. A. O. Vasilescu and D. Terzopoulos. Multilinear independent components analysis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume I, pp. 547–553, San Diego, CA, 2005. 1, 4, 8

M. A. O. Vasilescu and D. Terzopoulos. Multilinear projection for appearance-based recognition in the tensor framework. In *Proc. 11th IEEE Inter. Conf. on Computer Vision (ICCV'07)*, pp. 1–8, 2007. 1, 3, 8, 16

M. Alex O. Vasilescu. *A Multilinear (Tensor) Algebraic Framework for Computer Graphics, Computer Vision, and Machine Learning*. PhD dissertation, University of Toronto, 2009. 1, 2, 3, 4, 14

M. Alex O. Vasilescu and Eric Kim. Compositional hierarchical tensor factorization: Representing hierarchical intrinsic and extrinsic causal factors. In *The 25th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD19): Tensor Methods for Emerging Data Science Challenges Workshop*, Aug. 5 2019. 2, 7

M. Alex O. Vasilescu, Eric Kim, and Xiao S. Zeng. Causalx: Causal explanations and block multilinear factor analysis. In *2020 25th International Conference of Pattern Recognition (ICPR 2020)*, pp. 10736–10743, Jan 2021. 1, 2, 7, 8, 9

Joshua Vendrow, Jamie Haddock, and Deanna Needell. A generalized hierarchical nonnegative tensor decomposition, 2021. 2

D. Vlasic, M. Brand, H. Pfister, and J. Popovic. Face transfer with multilinear models. *ACM Transactions on Graphics (TOG)*, 24(3):426–433, Jul 2005. URL http://people.csail.mit.edu/jovan/assets/papers/vlasic-2005-ftm.pdf. 1

Hongcheng Wang and Narendra Ahuja. Facial expression decomposition. In *Proc, 9th IEEE Inter. Conf. on Computer Vision (ICCV)*, pp. 958–65,v.2, 2003. doi: 10.1109/ICCV.2003.1238452. 1

M. Wang, Y. Panagakis, P. Snape, and S. Zafeiriou. Learning the multilinear structure of visual data. In *2017 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 6053–6061, Jul 2017. doi: 10.1109/CVPR.2017.641. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.641. 2

C. Xiong, L. Liu, X. Zhao, S. Yan, and T. K. Kim. Convolutional fusion network for face verification in the wild. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(3):517–528, Mar 2016. ISSN 1051-8215. doi: 10.1109/TCSVT.2015.2406191. 1

J. Yang, X. Gao, D. Zhang, and J. Yang. Kernel ICA: An alternative formulation and its application to face recognition. *Pattern Recognition*, 38(10):1784–87, 2005. 7

M. Yang. Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. In *Proc. 5th IEEE Inter. Conf. on Automatic Face Gesture Recognition*, pp. 215–220, Washington, DC, May 2002. doi: 10.1109/AFGR.2002.4527207. 7