Auto-Encoding Topographic Factors

Antonio Moretti [*] [1] **Andrew Stirn** [*] [1] **Gabriel Marks** [1] **Itsik Pe'er** [1]

Abstract

Topographic factor models separate overlapping signals into latent spatial functions to identify correlation structure across observations. These methods require underlying structure to be held fixed and are not robust to deviations commonly found across images. We present Auto-Encoding Topographic Factors, a novel variational inference scheme to decompose irregular observations on a lattice into a superposition of low rank sources. By exploiting recent developments in variational autoencoders, we replace fixed sources with a non-linear mapping that parameterizes an unnormalized distribution on the lattice. In doing so, we permit sources to drift dynamically filtering residual differences in location across comparable areas of interest. This gives an implicit mapping to a unique latent representation while simultaneously forcing the posterior to model group variability in spatial structure. Simulation results and applications to functional imaging demonstrate the effectiveness of our method and its ability to outperform existing spatial factor models.

1. Introduction

The analysis of biomedical images has accelerated in recent years due to domain specific methodologies developed for multiple application areas. Calcium imaging in neurons (Pnevmatikakis et al., 2016), transcriptome profiling from single cells (Svensson et al., 2017) and functional imaging of various biomarkers (Gershman et al., 2014; Manning et al., 2014a) are exciting examples. Latent variable models are the predominant method for visualizing and extracting structure in spatial data. This data is characterized by a location vector $\mathbf{x}_i \in \Omega \subseteq \mathbb{R}^d$ parameterizing each observation $\mathbf{y}(\mathbf{x}_i)$. Given a tensor $\mathcal{Y} \equiv \{\mathbf{y}(\mathbf{x}_1), \cdots, \mathbf{y}(\mathbf{x}_m)\}_{n=1}^{N}$ of N realizations, each a sequence of m correlated random variables $Y(\mathbf{x}_1), \cdots, Y(\mathbf{x}_m)$, a fundamental challenge is to identify a subset of physical locations that define areas of interest. To this end, lattice based models formalize an encoding of a latent probability distribution over $Y(\mathbf{x}_1), \cdots, Y(\mathbf{x}_m)$ to quantify statistical dependencies based on distance. This representation is often used for Gaussian process regression or Kriging methods to predict covariance structure between hidden variables and observed features across physical location in an ensemble (Svensson et al., 2017). For example, extracting relevant voxels from a collection of functional images to discover a latent hemodynamic response enables comparing baseline vs pathological populations (Worsley et al., 1996).

Techniques such as robust principal component analysis (Candès et al., 2011), independent components analysis and dictionary learning are commonly applied to blind source separation problems; however they require an inherently linear demixing or deconvolution and may fail if there is no linear mixture that leads to independent outputs (Mukamel et al., 2009). Notably these methods do not learn a distribution on the lattice that can be used to quantify uncertainty or to generate new data. Topographic factor models (Gershman et al., 2011; 2014; Manning et al., 2014a) are a family of Bayesian variational techniques for images that require underlying structure on the set of random variables to be held constant to produce a matrix factorization with spatially interpretable sources.

Here we develop Auto-Encoding Topographic Factors (AETF), a novel Bayesian algorithm to infer spatial dependencies by decomposing observations on a lattice into a weighted set of low rank sources. We are particularly interested in a solution that generalizes to unseen data and that is robust to non-collocated regions of interest. The key insight of AETF is to leverage recent advances in variational inference (Gershman et al., 2014; Ranganath et al., 2014) and Stochastic Gradient Variational Bayes (Kingma & Welling, 2013; Rezende et al., 2014) to learn a latent probability model that preserves group variability in spatial structure. Our contributions are to combine two paradigms where

[*]Equal contribution [1]Department of Computer Science, Columbia University. Correspondence to: Antonio Moretti <amoretti@cs.columbia.edu>.

Joint ICML and IJCAI Workshop on Computational Biology, Stockholm, Sweden, 2018.

convolutional neural networks define the loading matrix and the factor matrix itself maps data to source functions that transform across observations. This is achieved without hard coding hyper parameters that control an a-priori generative model. In doing so, we remove the propensity on initialization of domain specific priors. Experiments on two simulated datasets and on functional imaging data show that our model returns a higher proportion of variance explained than existing Topographic factor models.

2. Related Work

Non-negative matrix factorization has been adapted to calcium imaging data to infer both location and spiking dynamics of neurons from fluorescence movies (Pnevmatikakis et al., 2016), (J. Friedrich, 2015). In order to facilitate inference, significant domain specific prepossessing must be done to constrain the region of interest to patches and model spatial background structure. A more general approach to factor analysis for space-time variation involves identifying a common set of shared factors whose time varying dynamics are modeled with autoregressive weights (Lopes et al., 2008). Our method differs in that we allow each observation to have a unique set of factors generated by a common non-linear mapping from the input space. In the case of Spatial Dynamic Factor Analysis, Bayesian inference is performed with a reversible jump MCMC algorithm whose convergence is difficult to asses and must be tailored for the problem at hand.

Topographic Factor Analysis (TFA) is a matrix decomposition method for functional imaging proposed (Manning et al., 2014b) and fit using Black Box VI (Ranganath et al., 2014). Hierarchical Topographic Factor Analysis (HTFA) (Manning et al., 2014a) extends TFA for hierarchical data by inferring a separate set of parameters for each subject. The parameter of the generative model are picked heuristically while the parameters of the variational posterior are often pre-fit using a preprocessing algorithm. Our approach uses similar spatial functions and a posterior which is mean field in time, however (Manning et al., 2014b;a) hard codes the structure of the factors which are shared across observations. AETF requires no heuristic choice of generative model parameters and uses only variational inference to fit the approximate posterior. Previous models are unable to capture the variation among observations while fitting individual specific factors. In this sense, we develop a robust and expressive posterior which does not require hand tuning hyper-parameters for the priors. Unlike TFA and HTFA, our model does not require knowledge of the hierarchical covariance structure a-priori. Additionally, the TFA and HTFA models do not generalize to unseen data suffering linear parameter growth with respect to the size of the dataset.

3. Auto-Encoding Topographic Factors

3.1. Standard Lattice Modeling

Following the convention of factor analysis, we assume that our data $\mathbf{Y} \in \mathbb{R}^{N\times V}$ can be decomposed into a set of unobserved weights and latent factors. We use N to denote the number of observations (images), K the number of sources and V the number of lattice positions (voxels). We will be discussing lattices in both 2D as well as 3D for our analysis. Each latent source is defined using a function that assigns a value to each point on the lattice (in voxel space) based on its location. For example, using the MVN:

$$K(\mathbf{x}_i|\mu, \mathbf{\Sigma}) = exp\big\{ -\frac{1}{2}(\mathbf{x}_i - \mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}_i - \mu)\big\} \quad (1)$$

We posit each observation $\mathbf{y}_n \in \mathbb{R}^{1\times V}$ has a low rank approximation that is a product of factor loadings $\mathbf{w}_n \in \mathbb{R}^{1\times K}$ and a factor matrix $\mathbf{F} \in \mathbb{R}^{K\times V}$. The generative distribution of our model factorizes using a Gaussian as follows:

$$P(\mathbf{Y}) = \prod_{n=1}^{N} P(\mathbf{y}_n) \quad (2)$$

$$P(\mathbf{y}_n) = \mathcal{N}(\mathbf{y}_n|\mathbf{w}_n\mathbf{F}, \sigma_y^2) \quad (3)$$

where σ_y^2 denotes the location or voxel noise. In Manning (Manning et al., 2014b), radial basis source functions $f_k \in \mathbb{R}^V$ are used to generate basis images and to define $\mathbf{F}$, the source image matrix. In general rows of $\mathbf{F}$ are computed by evaluating each of the K source functions at all V lattice points of the voxel space.

While it is common to focus on $\mathbf{\Sigma} = \sigma\mathbf{I}$ or the MVN case in which $\mathbf{\Sigma}$ is full, a larger class of kernels are supported through the Matérn family of covariance functions. Here $K_\nu(\cdot)$ is the modified Bessel function of the second-kind with order parameter ν, where ρ defines correlation length and $\lfloor\nu\rfloor$ describes the smoothness of the process. $\Gamma(\cdot)$ is the gamma function.

$$K(\mathbf{x}_i|\mu, \nu) = \frac{1}{\Gamma(\nu)2^{\nu-1}}\Big(\frac{\sqrt{2\nu}}{\rho}\cdot\|\mathbf{x}_i - \mu\|\Big)^{\nu} \times K_\nu\Big(\frac{\sqrt{2\nu}}{\rho}\|\mathbf{x}_i - \mu\|\Big) \quad (4)$$

The above simplifies for half-integer values of ν and reduces to the rational quadratic function with $\nu, \rho > 0$ to express a scale mixture of squared exponentials:

$$K(\mathbf{x}_i|\mu, \nu, \rho) = \left(1 + \frac{\|\mathbf{x}_i - \mu\|^2}{2\nu\rho^2}\right)^{-\nu} \quad (5)$$

Samples from the Gaussian process are $\lfloor\nu - 1\rfloor$ times differentiable producing the RBF case when $\nu \to \infty$. As with the above, the choice of distance metric can produce isotropy or anisotropy.

We are interested in the posterior distribution which involves integrating over the set of possible values for the latent variables:

$$P(\mathbf{W},\mathbf{F}|\mathbf{Y}) = \frac{P(\mathbf{Y},\mathbf{W},\mathbf{F})}{P(\mathbf{Y})} \tag{6}$$

$$P(\mathbf{Y}) = \int\int P(\mathbf{Y},\mathbf{W},\mathbf{F}) d\mathbf{W} d\mathbf{F} \tag{7}$$

The problem is in general intractable to compute. To perform variational inference, a mean field distribution is defined in which each variable is independent:

$$\begin{aligned} Q(\mathbf{W},\mathbf{M},\boldsymbol{\Lambda}) = \prod_{n=1}^{N}\prod_{k=1}^{K} \mathcal{N}(w_{n,k}|\mathbf{m}_{w_{n,k}}, \boldsymbol{\Lambda}_{w_{n,k}}) \\ \mathcal{N}(c_{n,k}|\mathbf{m}_{c_{n,k}}, \boldsymbol{\Lambda}_{c_{n,k}}) \mathcal{N}(s_{n,k}|\mathbf{m}_{s_{n,k}}, \boldsymbol{\Lambda}_{s_{n,k}}) \end{aligned} \tag{8}$$

We introduce notation for the set $\phi_k \in \phi$ to denote hyperparameters where c, s, w denote centers, width scales and weights respectively:

$$\phi_k = \{\mathbf{m}_{c,k}, \boldsymbol{\Lambda}_{c,k}, \mathbf{m}_{s,k}, \boldsymbol{\Lambda}_{s,k}, \mathbf{m}_{w,k}, \boldsymbol{\Lambda}_{w,k}\} \tag{9}$$

These allow drawing corresponding latent random variables for centers, width scales and weights for the kth latent source:

$$Z_k = \{z_{c,k}, z_{s,k}, z_{w,k}\} \tag{10}$$

where $z_{\xi,k} \sim \mathcal{N}(\mathbf{m}_{\xi,k}, \boldsymbol{\Lambda}^2_{\xi,k})$ for $\xi \in \{c, s, w\}$. Note that in the isotropic case $\phi \in \mathbb{R}^{K(D+5)}$ and $\mathbf{Z} \in \mathbb{R}^{K(D+2)}$ where D is the dimensionality of the lattice.

Across all ξ, k one can define $\mathbf{m}_\phi = (\mathbf{m}_{\xi,k})_{\forall \xi,k}$ and $\boldsymbol{\Sigma}_\phi = \boldsymbol{\Lambda}_\phi \boldsymbol{\Lambda}_\phi^T$ for $\boldsymbol{\Lambda}_\phi = (\boldsymbol{\Lambda}_{\xi,k})_{\forall \xi,k}$, thus the parameters $\mathbf{m}_\phi, \boldsymbol{\Sigma}_\phi$ denote the means and covariances which are used to draw $\mathbf{Z}$. $\mathbf{Z}$ then defines $\mathbf{F}$, by f_k being a Gaussian function with parameters $z_{c,k}$ and $z_{s,k}$.

3.2. Auto-Encoding Topographic Factors

The idea of AETF is to replace the fixed latent sources by defining a function that parameterizes $\mathbf{Z}$ using the output of a probabilistic encoder. The encoder creates an implicit mapping from each $\mathbf{y}_n \in \mathbf{Y}$ across the set of observations to a unique factor representation while requiring that ϕ encodes the group variability in spatial structure.

Formally, the variational inference framework states the ELBO for the marginal log likelihood $\mathcal{L}(\mathbf{Y}) \leq \log\, p(\mathbf{Y})$ with respect to the variational approximation $q_\phi(\mathbf{Z}|\mathbf{Y})$:

$$\begin{aligned} \mathcal{L}(\mathbf{Y}) &= \mathbb{E}_{q(\mathbf{z}|\mathbf{y})}[\log\, p_\theta(\mathbf{Y},\mathbf{Z})] - \mathbb{E}_{q(\mathbf{z}|\mathbf{y})}[\log\, q_\phi(\mathbf{Z}|\mathbf{Y})] \\ &= \mathbb{E}_{q(\mathbf{z}|\mathbf{y})}[\log\, p_\theta(\mathbf{Y}|\mathbf{Z})] - D_{KL}(q_\phi(\mathbf{Z}|\mathbf{Y})||p(\mathbf{Z}))] \end{aligned} \tag{11}$$

We wish to compute the expectation in (11) numerically and differentiate with respect to ϕ.

We now rewrite Equation (3) as

$$P(\mathbf{y}_n) = \mathcal{N}(\mathbf{y}_n|\mathbf{w}_n(\mathbf{y}_n)\mathbf{F}(\mathbf{y}_n), \sigma_y^2) \tag{12}$$

and decompose $\mathbf{F}$ as

$$\mathbf{F}(\mathbf{y}_n) = \begin{pmatrix} f_1(\mathbf{y}_n) \\ \vdots \\ f_K(\mathbf{y}_n) \end{pmatrix} \tag{13}$$

where $f_k(\mathbf{y}_n)$ is the lattice values of a Gaussian function parameterized by $z_{c,k}(\mathbf{y}_n)$ and $z_{s,k}(\mathbf{y}_n)$. $z_{\xi,k}(\mathbf{y}_n)$ itself is a latent variable drawn from a normal distribution $z_{\xi,k}(\mathbf{y}_n) \sim \mathcal{N}(\mathbf{m}_{k,\xi,\phi}(\mathbf{y}_n), \boldsymbol{\Lambda}_{k,\xi,\phi}(\mathbf{y}_n))$ whose parameters are the encoder output.

Employing the well known reparameterization trick (Kingma & Welling, 2013; Rezende et al., 2014), we sample from $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ to compute the following as the Monte Carlo estimates of gradients have high variance:

$$\mathbf{Z}_c = \mu_c + \epsilon \odot \sigma_c \tag{14}$$

$$\mathbf{Z}_s = \mu_s + \epsilon \odot \sigma_s \tag{15}$$

One is now free to choose the weights $\mathbf{Z}_w \in \phi$ as variational parameters of the recognition model or parameters with the generative model: $\mathbf{Z}_w \in \theta$. Including the weights in ϕ gives:

$$\mathbf{Z}_w = \mu_w + \epsilon \odot \sigma_w \tag{16}$$

When $\mathbf{Z}_w \notin \phi$, we learn the weights as point estimates using the update rule:

$$\mathbf{W}^{i+1} \leftarrow \mathbf{W}^i \odot \mathbf{Y}\mathbf{F}(\mathbf{y}_n)^T \oslash \mathbf{W}^i\mathbf{F}(\mathbf{y}_n)\mathbf{F}(\mathbf{y}_n)^T \tag{17}$$

Note that the problem is hard due to the non-convexity in the source image matrix. With the parameters ϕ of the recognition model in hand, we have the full model specification. In contrast to standard autoencoder formalization, where the generative model involves a decoder whose parameters need to be inferred, AETF specifies the generative model. We thus compute the approximation $\hat{\mathbf{y}}_n = \mathbf{W}(\mathbf{y}_n) \cdot \mathbf{F}(\mathbf{y}_n)$.

Standard autoencoders learn the respective encoder/decoder parameters θ, ϕ by maximizing the conditional log likelihood $\mathbb{E}_{q(\mathbf{z}|\mathbf{y}^i)}[log\;\; p_\theta(\mathbf{y}^i|\mathbf{z})]$ by differentiating through $g \leftarrow \nabla_{\theta,\phi}\mathcal{L}^M(\theta, \phi; \mathbf{Y}^M, \epsilon)$ (Kingma & Welling, 2013; Rezende et al., 2014). AETF only needs to learn the encoder parameters ϕ, which is achieved by analogous maximization of the conditional log likelihood $\mathbb{E}_{q(\mathbf{z}|\mathbf{y}^i)}[log\;\; p(\mathbf{y}^i|\mathbf{z})]$, differentiating through $g \leftarrow \nabla_\phi \mathcal{L}^M(\phi; \mathbf{Y}^M, \epsilon)$.

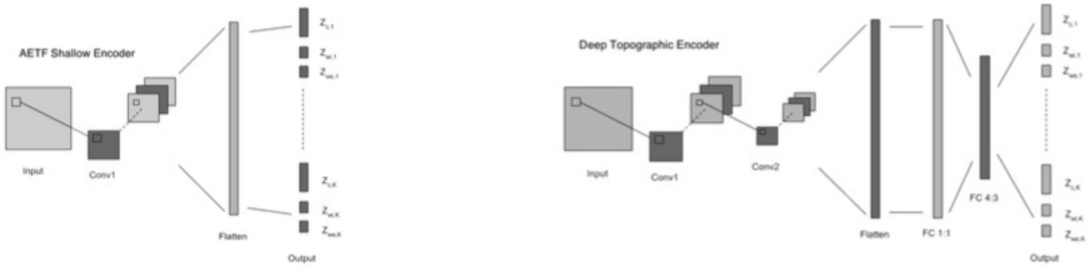

Figure 1. Schematic illustrating encoder network architectures for the AETF framework. Shallow (a) and deep (b) architectures shown.

3.3. Implementation

The encoder takes as input an observation and outputs the parameters of the distributions over latent variables. Two recognition models are implemented, one with isotropic and another with full covariance source functions. The isotropic decoder receives as input the sampled latent space vector $\mathbf{Z} \in \mathbb{R}^{k(d+2)}$ including $\mathbf{Z}_c$, $\mathbf{Z}_s$, and $\mathbf{Z}_w$. Note that in the second case of a full covariance matrix $\mathbf{Z}_{\mathbf{s}_{\mathbf{\Sigma}}k} = \mathbf{\Lambda}\mathbf{\Lambda}^T$, we learn parameters $\mathbf{Z}_{\mathbf{s}_{\mathbf{\Sigma}}} \in \mathbb{R}^{kd(d+1)/2}$. The spatial factorization constraints of our probability model are imposed within the decoder. Thus unlike traditional variational autoencoders where both the encoder and decoder are neural networks, AETF uses a neural network only for the encoder. The decoder uses the sampled latent space to reconstitute the input according to our imposed factorization and therefore is not parameterized by a neural network.

The encoder network can be comprised of any number of convolutional layers followed by any number of fully-connected layers before the output layer. The convolutional layer executes a $L^{(1)} \otimes \cdots \otimes L^{(D)}$ convolution along the number of lattice dimensions D (where L is specified for each layer) with k (the number of sources) output channels, a bias addition, a $tanh$ non-linearity, and max pooling with a $3^{(1)} \otimes \cdots \otimes 3^{(D)}$ kernel and a stride of 1. Our fully-connected layers begin operating on the flattened output of the last convolutional layer or the flattened image if a convolution layer is not employed. Their output dimensions are specified ratiometrically according their output-to-input dimensions. Like most autoencoders, our encoder seeks to compress information. Thus, we only consider output-to-input ratios for our fully-connected layers that are all less than or equal to 1. These fully-connected layers invoke an affine transformation followed by a $tanh$ non-linearity.

Our final output layer varies according to the latent space parameter class. Those parameters that are means (μ_c, μ_s, and μ_e) have no restrictions on their values except the last one, which must be positive. We handle this exception in the decoder. Therefore, we are free to use a vanilla affine transform as the output layers for these parameters. Conversely, those parameters that are standard deviations (σ_c, σ_s, and σ_w) must be greater than or equal to zero. Thus for those standard deviations that parameterize our latent space, we employ an affine transformation followed by a custom non-linearity we call PostReAct (Positive Real Activation in equation 18). This non-linearity is a piecewise combination of a shifted ReLU and a decaying exponential. In this manner, we benefit from ReLU's positive regime that avoids vanishing gradients that are common with double-saturating activations while avoiding the potential of neuron death associated with ReLU's negative regime.

$$\Psi(\lambda) = \begin{cases} \exp(\lambda) & , \lambda < 0 \\ \lambda + 1 & , \lambda \geq 0 \end{cases} \tag{18}$$

Our decoder has two responsibilities. First, it constructs the spatial factors using the $\mathbf{Z}_c$ and $\mathbf{Z}_s$ latent space. However and as aforementioned, $\mathbf{Z}_s$ arrives at the encoder on the incorrect support. The RBF function assumes this number is positive. We convert $\mathbf{Z}_s$ to the correct support in two ways. First, we pass it through a PostReAct non-linearity. Second, we square it in our isotropic implementation. Equation (19) captures this process that we use for each of our basis image calculations. Here, $f_k(v)$ represents the value of the kth RBF source at voxel position v. Unlike traditional RBF functions, we add a 1 to the denominator to clamp the source's width in a continuously differentiable fashion. Prior to this modification, sampled $\mathbf{Z}_s$ that resulted in small source widths produced exploding gradients for our optimizer. Once, the decoder constructs the k basis images it recombines them into a single image via a weighted summation that uses $\mathbf{Z}_w$.

$$f_k(v) = \exp\left(- \frac{||\mathbf{Z}_{c,k} - v||_2^2}{2 \cdot \Psi(\mathbf{Z}_{s,k})^2 + 1} \right) \tag{19}$$

We present two encoder network architectures. Our first, uses only a 7×7 convolutional layer followed by the output layer. Our second uses–in order of appearance–a 7×7 convolutional layer, a 5×5 convolutional layer, a $1 : 1$ output-to-input fully-connected layer, and a $4 : 3$ output-to-input fully-connected layer followed by the output layer. We then permute these two architectures for differing numbers of latent sources. We note that k modifies the size of

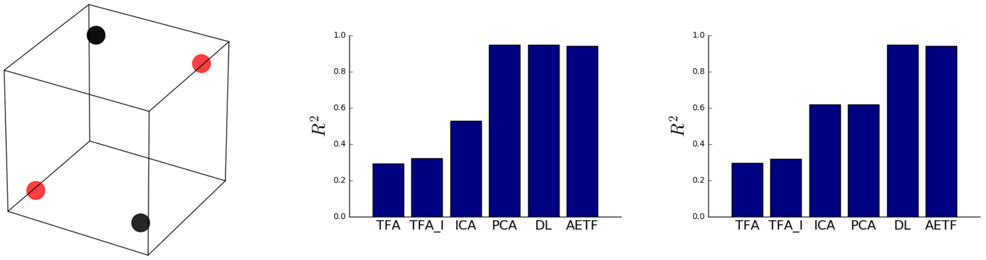

Figure 2. Description of the first simulation: (a) Schematic illustrating two source functions located near the vertices of the lattice. Each source transforms across observations drifting between one of two states (denoted with colors red and black); (b) variance explained for different models using two components. AETF outperforms TFA on the train set; (c) TFA, PCA and ICA underperform on the test set.

the network as it determines the number of output channels for each convolutional layer. Our implementation supports imposing a non-negative factorization in addition to one in which the weights are permitted to take negative values.

We modify the loss from equation (11). Specifically, we introduce a β term in front of the regularizer as suggested in (Liang et al., 2018). Furthermore, they suggest β values less than 1 improve quality. The utilized per-sample loss function for AETF appears in equation (20). In our experiments we set β to zero such that our loss reduces to just the reconstruction error. Here, n represents the n^{th} sample and V is the cardinality of our voxel space such that subscript n, i corresponds to the i^{th} voxel of the n^{th} sample.

$$\begin{aligned}\mathcal{L}(\mathbf{Y}_n) = \frac{1}{V}\sum_{i=1}^{V}\Big[(\hat{\mathbf{Y}}_{n,i} - \mathbf{Y}_{n,i})^2\Big] \\ + \beta D_{KL}(q_\phi(\mathbf{Z}_i|\mathbf{Y}_i) \mid\mid p(\mathbf{Z}_i))\end{aligned} \tag{20}$$

4. Experiments

Three results are presented, each of which illustrates a strength of the AETF model. We discuss $i)$ fitting in-model synthetic data, ii) fitting non-collocated source functions to smooth, unmix and localize spatial dependencies in random fields, and $iii)$ decomposing thousands of functional images into latent source functions and evaluating our ability to generalize on unseen data.

4.1. In-Model Data

We generate a synthetic dataset using $k = 2$ source functions over 1000 observations on a 20× 20×20 lattice. In our experiments, Topographic Factor Analysis (TFA) was unable to handle larger lattice dimensions in $\mathbb{R}^3$. Unlike the generative process specified in TFA (Manning et al., 2014a), the position of each source function may shift across observations and is not restricted to be collocated on the lattice. This design choice is relevant given that the blood oxygen level dependency (BOLD) response is not static and often transforms dynamically as a time series. Figure (2a) provides a schematic illustrating the position of two sources located near the vertices of the cube. Each source function is permitted to drift between one of two possible states which are represented using the red and black colors. Figures (2b) and (2c) provide the variance explained on the training and testing sets respectively using $k = 2$ components. TFA, ICA and PCA underperform relative to Dictionary Learning (DL) and AETF. Unlike DL, AETF is able to parameterize the transforming source functions while maintaining nearly all of the variance explained.

4.2. Gaussian Random Fields

Gaussian random fields (GRFs) are often used in image analysis to model stochastic processes on a lattice and to introduce noise. We illustrate how Auto-Encoding Topographic Factors recovers autocorrelation structure by filtering a sequence of GRFs simulated using spectral methods (Annika & Jrgen, 2011). The spectral density of a fixed covariance kernel is multiplied with a Fourier transformed white noise field before applying an inverse transform. This process introduces a non-smooth signal in which spatial autocorrelations are not explicitly colocated across observations.

Figure (3a) provides a representative sample along with the inferred reconstruction in Figure (3b). We fit 10 source functions to 1000 observations on a cubic lattice. As a visualization, the planar cross-section is provided in Figure (3). The surface is shifted above the image to illustrate the smoothness of the field along with contours presenting the location of the inferred spatial factors. Figure (3c) provides the variance explained across models and fits. Auto-Encoding Topographic Factors outperforms Topographic Factor Analysis both without and with initialization (denoted TFA and TFA$_I$), ICA, Dictionary Learning (DL), and PCA; the canonical method for Gaussian data. It is clear that Topographic Factor Analysis underperforms when the correlation structure is not held fixed.

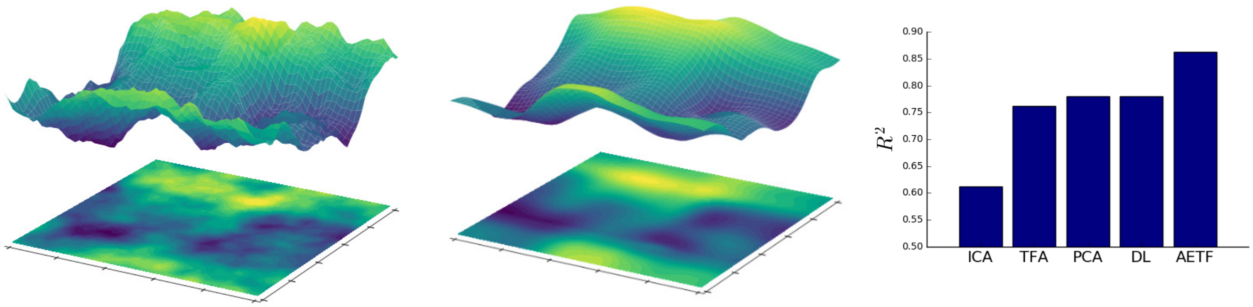

Figure 3. Summary of the AETF fit to the GRF simulation: (a) the cross-section of a single observation and (b) the cross-section of the AETF topographic reconstruction. The surface is shifted above the plane to illustrate the smoothness of the field along with contours presenting the location of the inferred spatial factors; (c) variance explained across models. AETF provides the highest R^2.

4.2.1. IMAGE NOISE

Spatial factor models learn a smooth statistical map in the presence of noise in which the desired signal extends over several lattice points. A good fit should be robust to variation between observations while preserving correlation structure within the data. To achieve this, Auto Encoding Topographic Factors learns a unique decomposition by simultaneously factorizing the observation matrix, inferring the position of spatial dependencies and introducing flexibility for the location of factors across the lattice. This process is analogous to blurring residual differences in location between comparable areas of activation. When two observations are similar, this is captured in their latent spatial representations. For heterogenous data, AETF parameterizes spatial dynamics.

4.2.2. INITIALIZING TFA AND HTFA

Heuristics are often suggested to initialize hyperparameters for Topographic Factor Analysis so that local optima in the source image matrix do not serve as an impediment for non-convex optimization. There exist multiple values of parameters for the location and width of the sources that are equally likely to have generated an observation $\mathbf{y}_n$, due to the rotational invariance of $\mathbf{F}$. One proposed approach is to place hyperparameters a-priori in locations corresponding to high and low activation. Hotspot initialization (Manning et al., 2014a) refers to an iterative process in which the mean image is computed, the mean activation is subtracted and the absolute value is taken of all of the remaining activations. The result is an energy landscape in which peaks correspond to extremum. These peaks are iteratively flattened as source centers are placed on these extremum. Values for $\mathbf{m}_{s_{n,k}}$ the mean of the distribution for source $k's$ width scale are then solved for via Newton's method. Once pre-initialized, the source centers and width scales frequently remain fixed. In our experiments, sources for TFA initialized using both hotspot initialization and k-means outperformed experiments with no initialization. Auto-Encoding Topographic Factors outperformed both methods without being contingent upon any such initialization to perform inference successfully.

4.3. NYU Dataset

We consider the problem of modeling functional images using the NYU Test-Retest dataset (Shehzad et al., 2009). The data was obtained using a Siemens Allegra 3.0 Tesla scanner. The data consists of twenty six participants each with 3 resting-state scans of 197 continuous EPI functional volumes. Each scan consists of 39 slices of a matrix 64×64 with an acquisition voxel size of $3 \times 3 \times 3$ mm. Scans 2 and 3 were conducted 45 minutes apart roughly 5-16 months after Scan 1.

Slice timing correction, spatial normalization, smoothing and noise stripping were performed using the *Nipype* interface to the FSL software library. The sequential dependency of the time series was not accommodated and each time frame was treated independently. An AETF model was trained using all three sessions reserving 20% for the testing set as a performance criteria to evaluate our fit. To test the significance of the lattice dimensions, models were fit to both sagittal cross-section data and full cubic volumes.

4.3.1. SAGITTAL CROSS-SECTIONS IN 2D

Sagittal cross-section data was fit to the 13 subjects using the first session. Figure (4a) provides the variance explained for AETF, PCA, ICA and DL as a function of number of sources on training data. TFA and HTFA implementations are not supported on the 2D lattice. The R^2 approaches 1 the number of sources K increases. Figure (4b) plots the weight values for two randomly selected source functions across a subset of time frames. Dashed vertical lines distinguish subjects. Strong per-subject similarities are visible. Figure (4c) provides the variance explained

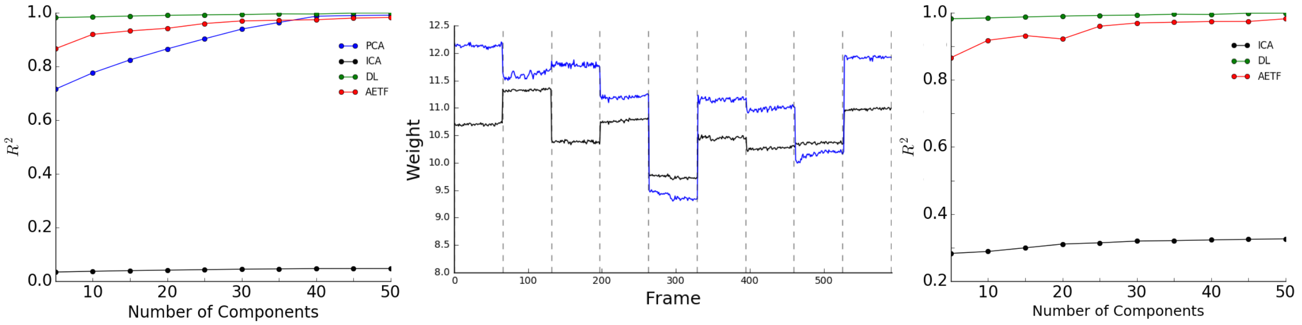

Figure 4. Summary of the Sagittal Cross-Section NYU Data: (a) variance explained for various models as a function of number of sources on the training data; (b) two source weights plotted across time frames illustrating strong subject-specific similarities. Dashed vertical lines denote unique subjects; (c) variance explained as a function of number of sources for test data. AETF returns a near perfect variance explained using $k = 25$ source functions.

by AETF as K increases on the test data. Using $k = 50$ source functions 99% of variance is explained. We find that $K = 25$ anisotropic sources are sufficient for high quality reconstruction. Interestingly, AETF is able to converge without *any* preprocessing to preserve 89% of total variance on the raw NYU data. On the preprocessed dataset 25 source functions preserve 98% of total variance.

4.3.2. Functional Imaging with 3D Volumes

The three-session NYU data on the cubic lattice was modeled using AETF, TFA and HTFA. The $64 \times 64 \times 40$ lattice was divided into eight $20 \times 20 \times 20$ cubic volumes. TFA and HTFA were unable to handle larger lattice dimensions on the full set of 7683 frames. We fit $k = 10$ source functions to each cubic volume and average the cost across the total area. For TFA, one model was fit across subjects whereas 39 subjects were fit using HTFA. Figure (5a) displays a new frame evaluated using the trained model to illustrate the effect of applying the trained model on unseen data. The surface is plotted above the image to highlight the areas of activation above the corresponding factors on the mesh. Figure (5a) and (5b) provide the train and test R^2 respectively. It is clear that AETF outperforms both methods. Unlike HTFA, the hierarchical covariance structure is inferred from the data and not specified a-priori.

5. Discussion

In the context of functional imaging, a spatial model should be able to extract both global and individual characteristics. In examining how the model parameters for centers, widths and weights varied across testing data, we find source centers are not only similar at the per-subject micro-scale but also marginally similar at the global macro-scale. However, we see much more global variability with weight values. Compared to a similar factorization in (Manning et al., 2014b) that constricts learning to globally shared sources and individual per-frame weights, our model naturally learns a similar representation. Namely, through a shared encoder mapping, source variability is less pronounced than weight variability.

Auto-Encoding Topographic Factors offers several advantages over unstructured blind source separation techniques. TFA, HTFA, Dictionary Learning, PCA and ICA explicitly learn factor weights (loadings) for each observation. The number of trainable parameters is therefore linear with respect to N, the number of observations. AETF's parameters ϕ are constant with respect to N. This paradigm reduces memory footprint for large N and allows AETF to handle unseen data. By design, the factor images learned by AETF possess lower complexity than the observed images.

AETF can accommodate any priors but is not contingent upon an a-priori choice of generative model hyperparameters to converge. This is mitigated by choosing uniform priors for the generative model. In this way, AETF is not sensitive to preinitialization issues that plague TFA and HTFA. It is also possible to parameterize the priors of the generative model using a trainable decoder network. Unlike TFA and HTFA, source functions are allowed to transform across individual frames. This is advantageous for time series modeling. In our experiments, Dictionary Learning sometimes provided a comparable R^2. AETF however returns a factorization along with spatially parameterized functions. AETF was written in TensorFlow. The source code and several visualizations are available online.

6. Conclusions

We have presented Auto-Encoding Topographic Factors, a novel variational inference scheme for lattice-based measurements in which each observation is given a unique spatial decomposition. The proposed method is robust to high dimensional data in which sources are not rigidly

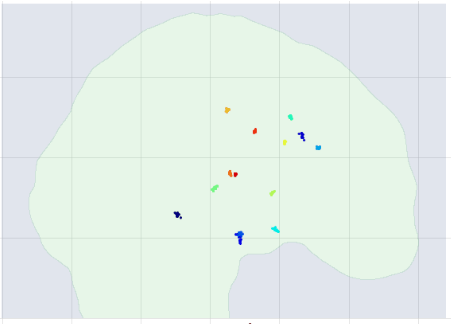

Figure 5. The distribution of the top 15 weighted source center means for subjects and time frames. Colors represent distinct source function center means. The outline of a single frame is provided as a reference. Each point is a parameter for an image in a collection of time series. The latent factors recovered by AETF exhibit strong spatial localization.

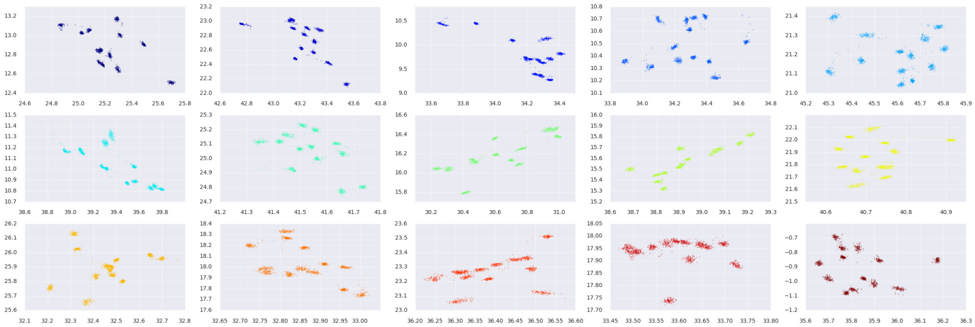

Figure 6. The distribution of center means enhanced for each individual source function. Factors are distinguished by color matching their corresponding representations in Figure 5. Each point is a parameter for an image in a collection of time series. Thirteen clusters appear consistently within each factor recovering the number of subjects. AETF latent representations implicitly preserve hierarchical covariance structure and can be used for clustering.

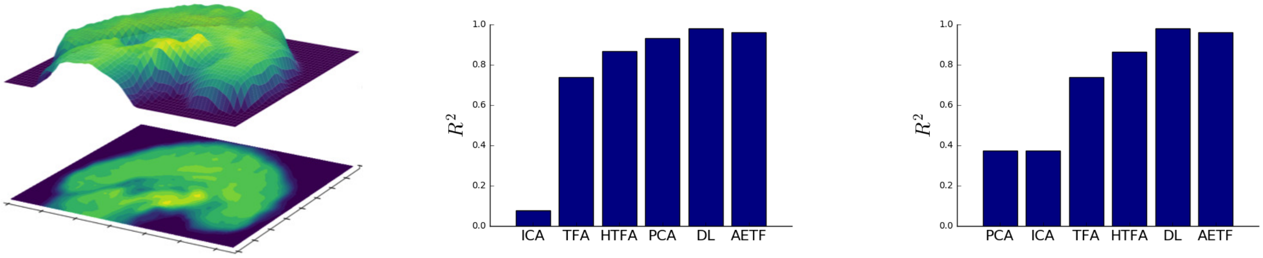

Figure 7. Results for the cubic volume NYU data: (a) a cross section of a frame and the surface highlighting source intensities; R^2 values for training (b) and testing (c) for various models averaged across eight cubic volumes using $k = 10$ source functions. AETF consistently outperforms both Topographic Factor Analysis (TFA) and Hierarchical Topographic Factor Analysis (HTFA).

colocated, introduces non-linearity, supports a family of kernels and the ability to enforce a constrained or non-negative matrix factorization. AETF preserves a large proportion of variance even when factor positions shift dynamically across observations. Highlights include the ability to identify autocorrelation structure in a collection of random fields and the ability to scale to thousands of 3D functional images with a number of training parameters independent of dataset size.

The results, in particular Figure (4b), motivate an explicitly-hierarchical AETF across individuals, as well as a temporally correlated AETF. A natural extension is to explore the method of normalizing flows (Rezende & Mohamed, 2015; Kingma et al., 2016) as an alternative to defining factors by specifying kernels for source functions. We expect that the approximate posterior would remain simple to compute while each source is permitted to undergo a sequence of transformations giving rise to complex and expressive spatial dependencies.

References

Annika, Lang and Jrgen, Potthoff. Fast simulation of gaussian random fields. *Monte Carlo Methods and Applications*, 17(3):195–214, 2011.

Candès, Emmanuel J., Li, Xiaodong, Ma, Yi, and Wright, John. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, June 2011. ISSN 0004-5411. doi: 10.1145/1970392.1970395.

Gershman, Samuel, Blei, David M., Pereira, Francisco, and Norman, Kenneth A. A topographic latent source model for fmri data. *NeuroImage*, 57(1):89–100, 2011. doi: 10.1016/j.neuroimage.2011.04.042.

Gershman, Samuel, Blei, David M., Norman, Kenneth A., and Sederberg, Per B. Decomposing spatiotemporal brain patterns into topographic latent sources. *NeuroImage*, 98:91–102, 2014. doi: 10.1016/j.neuroimage.2014.04.055.

J. Friedrich, et al. Fast constrained non-negative matrix factorization for whole-brain calcium imaging data. In *NIPS workshop on Statistical Methods for Understanding Neural Systems*, 2015.

Kingma, Diederik P. and Ba, Jimmy. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

Kingma, Diederik P. and Welling, Max. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

Kingma, Diederik P., Salimans, Tim, Józefowicz, Rafal, Chen, Xi, Sutskever, Ilya, and Welling, Max. Improving variational autoencoders with inverse autoregressive flow. In *NIPS*, pp. 4736–4744, 2016.

Lee, Daniel D. and Seung, H. Sebastian. Algorithms for non-negative matrix factorization. In Leen, T. K., Dietterich, T. G., and Tresp, V. (eds.), *Advances in Neural Information Processing Systems 13*, pp. 556–562. MIT Press, 2001.

Liang, Dawen, Krishnan, Rahul, Hoffman, Matthew, and Jebara, Tony. Variational autoencoders for collaborative filtering. In *Proceedings of The Web Conference (WWW), 2018*, 2018.

Lopes, Hedibert Freitas, Salazar, Esther, and Gamerman, Dani. Spatial dynamic factor analysis. *Bayesian Anal.*, 3(4):759–792, 12 2008. doi: 10.1214/08-BA329.

Manning, Jeremy R., Ranganath, Rajesh, Keung, Waitsang, Turk-Browne, Nicholas B., Cohen, Jonathan D., Norman, Kenneth A., and Blei, David M. Hierarchical topographic factor analysis. In *International Workshop on Pattern Recognition in Neuroimaging, PRNI 2014, Tübingen, Germany, June 4-6, 2014*, pp. 1–4, 2014a. doi: 10.1109/PRNI.2014.6858530.

Manning, Jeremy R., Ranganath, Rajesh, Norman, Kenneth A., and Blei, David M. Topographic Factor Analysis: A Bayesian Model for Inferring Brain Networks from Neural Data. *PLoS ONE*, 9(5):e94914, may 2014b. ISSN 1932-6203. doi: 10.1371/journal.pone.0094914.

Mukamel, Eran A, Nimmerjahn, Axel, and Schnitzer, Mark J. Automated analysis of cellular signals from large-scale calcium imaging data. *Neuron*, 63:747–760, 2009.

Pnevmatikakis, EftychiosA., Soudry, Daniel, Gao, Yuanjun, Machado, Timothy A., Merel, Josh, Pfau, David, Reardon, Thomas, Mu, Yu, Lacefield, Clay, Yang, Weijian, Ahrens, Misha, Bruno, Randy, Jessell, Thomas M., Peterka, DarcyS., Yuste, Rafael, and Paninski, Liam. Simultaneous denoising, deconvolution, and demixing of calcium imaging data. *Neuron*, 89(2):285 – 299, 2016. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2015.11.037.

Ranganath, Rajesh, Gerrish, Sean, and Blei, David M. Black box variational inference. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, AISTATS 2014, Reykjavik, Iceland, April 22-25, 2014*, pp. 814–822, 2014.

Rezende, Danilo Jimenez and Mohamed, Shakir. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pp. 1530–1538, 2015.

Rezende, Danilo Jimenez, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. In Xing, Eric P. and Jebara, Tony (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1278–1286, Bejing, China, 22–24 Jun 2014. PMLR.

Shehzad, Zarrar, Kelly, A. M. Clare, Reiss, Philip T., Gee, Dylan G., Gotimer, Kristin, Uddin, Lucina Q., Lee, Sang Han, Margulies, Daniel S., Roy, Amy Krain, Biswal, Bharat B., Petkova, Eva, Castellanos, F. Xavier, and Milham, Michael P. The resting brain: Unconstrained yet reliable. *Cerebral Cortex*, 19(10):2209–2229, 2009. doi: 10.1093/cercor/bhn256.

Svensson, Valentine, Teichmann, Sarah A., and Stegle, Oliver. SpatialDE - Identification Of Spatially Variable Genes. *bioRxiv*, pp. 143321+, May 2017. doi: 10.1101/143321.

Worsley, K. J., Marrett, S., Neelin, P., and Evans, A.C. A unified statistical approach for determining significant signals in location and scale space images of cerebral activation, 1996.