TI-VAE: A TEMPORALLY INDEPENDENT VAE WITH APPLICATIONS TO LATENT FACTOR LEARNING IN NEUROIMAGING

Anonymous authors
Paper under double-blind review

ABSTRACT

Functional magnetic resonance imaging (fMRI) data contain complex spatiotemporal dynamics, thus researchers have developed approaches that reduce the dimensionality of the signal while extracting relevant and interpretable dynamics. Recently, the feasibility of latent factor analysis, which can identify the lower-dimensional trajectory of neuronal population activations, has been demonstrated on both spiking and calcium imaging data. In this work, we propose a new framework inspired by latent factor analysis and apply it to functional MRI data from the human somatomotor cortex. Models of fMRI data that can perform whole-brain discovery of dynamical latent factors are understudied. The benefits of approaches such as linear independent component analysis models have been widely appreciated, however, nonlinear extensions are rare and present challenges in terms of identification. Deep learning methods are potentially well-suited, but without adequate inductive biases with respect to spatial weight-sharing may heavily overparameterize the model for the dataset size. Due to the underspecification of neuroimaging approaches, this increases the chances of overfitting and picking up on spurious correlations. Our approach extends temporal ICA to the non-linear case and generalizes weight sharing to non-Euclidean neuroimaging data. We evaluate our model on data with multiple motor sub-tasks to assess whether the model captures disentangled latent factors corresponding to each sub-task. Then, to evaluate the latent factors we find further, we compare the spatial location of each latent factor to the known motor homunculus. Finally, we show that our latent factors correlate better to the task than the current gold standard of source signal separation for neuroimaging data, independent component analysis (ICA).

1 INTRODUCTION

Functional magnetic resonance imaging (fMRI) is an important and widely used imaging method to study the whole-brain dynamics of the human brain. Although it does not directly capture neuronal activity, it can serve as a proxy for measuring neuronal activity non-invasively at high spatial resolutions. Clinicians and researchers have had issues interpreting the signal, however, due to the signal's high dimensionality, poor temporal resolution, and multiple sources of noise leading to a low signal-to-noise ratio. Both to understand the signal itself better and to move towards potentially clinically relevant information, researchers have focused on developing methods that summarize the signal across spatial and temporal scales Descombes et al. (1998); Woolrich et al. (2004). Pre-defined Atlases are also a popular tool to average and increase the signal-to-noise ratio in brain regions of interest (ROI). Averaging can be misleading because spatial regions can have multiple distinct timecourses that overlap within each region, which has led researchers to tools such as independent component analysis (ICA), that decompose the signal into multiple temporal trajectories with corresponding spatial sources McKeown & Sejnowski (1998); Beckmann & Smith (2005); Calhoun & Adali (2006). A promising alternative is emerging with respect to the characterization of neuronal population dynamics using fully-differentiable data-driven approaches. These approaches can scale to large neurological data easily, as well as allow for individualized trainable models. One example of such a technique is latent factor analysis via autoencoders Yu et al. (2008); Everett (2013). Classically, latent factor analysis for fMRI data is done with some form of matrix factorization, such as

principal component analysis Thomas et al. (2002), ICA McKeown & Sejnowski (1998); Beckmann & Smith (2005); Calhoun & Adali (2006), or dictionary learning Lee et al. (2010). Recently these matrix factorizations have been extended to tensor factorizations/analysis Ma et al. (2016), restricted Boltzmann machines (RBM)s Hjelm et al. (2014), and static autoencoders Kim et al. (2021); Geenjaar et al. (2021). In the field of neuronal populations, however, a recent approach finds latent factors using a recurrent autoencoder Pandarinath et al. (2018). Although with different interpretations and under different constraints, low-dimensional latent dynamics underlying the fMRI signal also likely exist. As such, an adapted application of latent factor analysis using recurrent autoencoders is a direction that can help alleviate some of the issues commonly associated with fMRI data. Although this approach does not directly tackle the fact that fMRI data is a proxy for neuronal activity, with careful interpretation, we can still make inferences about functional whole-brain dynamics. Moreover, deep learning approaches allow for the inclusion of relevant constraints, such as geodesic distances or known functional constraints that can further help constrain the solution space to relevant brain dynamics. In this work, we try to find dynamic latent factors underlying the fMRI signal in the context of motor task activations. In this paradigm, the ground truth factors are the motor regions and their associated activations, and the motor homunculus is well-documented across species. For a null model, we utilize the current gold standard of decomposing spatio-temporal fMRI signals, ICA. In addition, we explore the inclusion of constraints to the dynamics by using weight sharing based on hemispheric symmetry, geodesic distances, as well as prior functional activation. We show that our approach combined with these inductive biases better captures task effects than linear ICA when decomposing neural activity. Furthermore, the weight sharing we propose in this work helps expand this work to larger datasets. In fact, as an example we perform a calculation of the weights vs number of subjects in the Appendix A to emphasize our point that neuroscientifically-informed weight sharing is absolutely critical in neuroimaging. Some recent work in this direction also identifies meaningful non-linear dynamical systems from fMRI task data, but they use task information as input to their model Koppe et al. (2019), whereas we do not. Other related work Gao et al. (2020) shows that there exists a non-linear manifold for all tasks in the HCP dataset, but does not look at any specific factors and use ROIs to decrease the dimensionality of the data.

2 METHOD

The methods are organized by first explaining the experimental setup of the motor task fMRI data, as well as the relevant biological data used. The ground truth activations are based on the generalized linear model hemodynamic responses derived from SPM Penny et al. (2011) for each of the sub-motor tasks. We then describe and explain our method. The subsequent sections detail our novel weight sharing method to reduce the number of parameters and more directly incorporate neuroscientific inductive biases. Subsequently, we explain the temporal independence factor we include in our objective function and how we evaluate the temporal factors. The comparisons to null models using ICA are established in the final sections.

Biological data The data we use in this work are cortical surface timeseries from the open-access, under data usage terms, HCP-1200 dataset Van Essen et al. (2013), for all subjects with cortical surface timeseries data (1181). The data is registered using multimodal surface registration (MSM) Robinson et al. (2014; 2018), and surfaces are constructed using Freesurfer Glasser et al. (2013); Fischl (2012). Then, the vertices corresponding to the somatomotor region are extracted using the Yeo-7 atlas Yeo et al. (2011). Each subject's timeseries is band-pass filtered independently ($0.01 - 0.15$Hz) and then linearly detrended using the Nilearn package Abraham et al. (2014). The cortical surface is represented as a set of vertices ($\mathcal{V}$) and each vertex has a blood-oxygen-level-dependent (BOLD) value associated with it at each timestep (t). The number of vertices in this work is 11960, and the number of timesteps is 284. Furthermore, the timings of each of the sub-tasks for the motor task are that the right-hand task occurs at 11 and 132 seconds, the left foot task occurs at 26 and 117 seconds, the tongue task occurs at 41 and 102 seconds, the right-foot task occurs at 56 and 177 seconds, and the left-hand task occurs at 71 and 162 seconds. Each sub-task block lasts 12 seconds. Each timeseries, for each subject, is mapped into a group space, which means that each voxel represents roughly the same location in the brain. This also means that some deformation is not only introduced in the process of obtaining the surface voxels but also during the registration of the timeseries into group space. The locations of the voxels in this work are based on the group-based pial surface, which is the boundary between gray and white matter in the brain. The surface is

thus essentially a graph with a fixed structure, and only the values associated with the voxels change over time. Although cortical surface data has previously been mapped to a sphere and then been mapped to a $2D$ image using polar coordinates Kim et al. (2021), in this work we view the location of each vertex as a graph to retain as much distance information as possible.

Metrics and experimental setting The data consists of 5 motor tasks, left hand, right hand, left foot, right foot, and tongue movements, where the subject moves the respective limbs after hearing an auditory cue. Since the regions associated with these movements are well defined based on the motor homunculus as well as the timing of these events is known due to the event-based scientific paradigm, the ground truth of the spatio-temporal signal is well established in these tasks. Therefore we evaluate our model based on the following metrics: 1) its ability to reproduce the correct spatial maps observed during motor activation, and 2) the exact temporal dynamics associated with the activation during these tasks. The dataset is randomly shuffled and then divided into a training set (70%), a validation set (10%), and a test set (20%) to make sure it generalizes beyond the training data.

The tasks are assumed to last for 12 seconds, and each factor's timeseries is convolved with SPM's simulated hemodynamic response based on the block design of each sub-task. After obtaining the average temporal timeseries for the unseen test set, we find the factors that have the highest absolute average correlation with each sub-task. We then take the average over those absolute correlations to measure how well the model can learn some ground-truth underlying factors in the dataset. Knowing that the model finds underlying factors in the dataset opens up using this model for resting-state data, where underlying factors are often less apparent and no ground truth exists.

Sequential variational autoencoder Sequential autoencoders were developed to learn and model temporal dynamics efficiently. From an information-theoretic perspective, they bottleneck the information and assume that only the most important information is retained in the latent space. As such, sequential autoencoders have been used in a variety of different problems in order to model temporal datasets such as speech processing Graves et al. (2013), to compress high dimensional neuronal population data Keshtkaran et al. (2021), as well as model fMRI dynamics Kashyap & Keilholz (2020).

The sequential variational autoencoder in this methodology consists of a gated recurrent unit (GRU)Cho et al. (2014) and a linear layer. The GRU obtains as input the embeddings from the spatial encoder e_t and outputs its hidden state at each timestep. These hidden states are used to parameterize the mean and standard deviation of the Gaussian distributions at each timestep, see Figure 1c. The distributions are referred to as the factors f_t in this work. The reason we model the factors as distributions is that the loss function of variational autoencoders Kingma & Welling (2013) has been shown to encourage disentanglement of the separate factors in each distribution Graves et al. (2008); Higgins et al. (2016); Burgess et al. (2018); Higgins et al. (2022).

Formally, the problem consists of a dataset $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(\mathrm{N})}\} \in \mathcal{D}$, where each $\mathbf{x}^{(\mathrm{i})}$ is made up of T timesteps $\mathbf{x}^{(\mathrm{i})} = \{\mathbf{x}_1^{(\mathrm{i})}, \mathbf{x}_2^{i}, ..., \mathbf{x}_{\mathrm{T}}^{(\mathrm{i})}\}$. Each timestep for a subject $\mathbf{x}_{\mathrm{t}}^{(\mathrm{i})}$ are the blood-oxygen-level dependent (BOLD) values for each input voxels at that time. The model proposed in this work is based on a variational autoencoder (VAE) Kingma & Welling (2013), which learns both a generative $p_\theta(\mathbf{x}|\mathbf{z})$ and a variational approximation $q_\phi(\mathbf{z}|\mathbf{x})$ of the true posterior. VAEs are optimized using the evidence lower-bound (ELBO) on the expected marginal log-likelihood of $\mathbf{x}$, a more in-depth explanation of the ELBO is provided in previous work Kingma & Welling (2013). In our case we obtain a latent variable for each subject $\mathbf{z}^{(i)}$ and for each timestep $\mathbf{z}^{(i)} = \{\mathbf{z}_1^{(i)}, \mathbf{z}_2^{(i)}, ..., \mathbf{z}_T^{(i)}\}$. In our work we assume the prior for the variational estimation to be a zero-mean, unit-norm diagonal multivariate Gaussian distribution $p(\mathbf{z})$. The resulting ELBO for our problem setting is as follows.

$$\mathcal{L}(\theta, \phi; x_{\mathrm{t}}^{(\mathrm{i})}) := -\mathrm{D}_{\mathrm{KL}}\left(q_\phi(\mathbf{z}_{\mathrm{t}}^{(\mathrm{i})}|\mathbf{x}_{\leq \mathrm{t}}^{(\mathrm{i})})||p(\mathbf{z})\right) + \mathbb{E}_{q_\phi(\mathbf{z}_{\leq \mathrm{t}}^{(\mathrm{i})}|\mathbf{x}_{\mathrm{t}}^{(\mathrm{i})})}\left[\log p_\theta(\mathbf{x}_{\mathrm{t}}^{(\mathrm{i})}|\mathbf{z}_{\mathrm{t}}^{(\mathrm{i})})\right] \tag{1}$$

The two terms can be seen as an encoder $q_\phi(\mathbf{z}_{\mathrm{t}}^{(\mathrm{i})}|\mathbf{x}_{\leq \mathrm{t}}^{(\mathrm{i})})$ and a decoder $p_\theta(\mathbf{x}_{\mathrm{t}}^{(\mathrm{i})}|\mathbf{z}_{\mathrm{t}}^{(\mathrm{i})})$, both parameterized by separate neural networks. If the variance of the input data is assumed to be constant, then optimizing the log-likelihood of the decoder is the same as optimizing the mean-squared error between the input data and the reconstructed data from the decoder. In this case, the parameters of the encoder θ correspond to the spatial encoder and temporal decoder, whereas the generative parameters

ϕ correspond to the spatial decoder and distribution parameters (mean and standard deviations). The optimization of this lower bound is in our case done by taking the mean over the dimensions of the distribution and timesteps for the KL-divergence term. We take the sum over the mean-squared error between the reconstructed and true timesteps within a subject but take the mean over the number of input dimensions.

The importance of weight sharing As mentioned previously, it is important to first perform spatial dimensionality reduction before modeling the timeseries with the temporal encoder and decoder. The flexible weight sharing we propose allows us to do just that and learn lower-dimensional spatial features that we can now use in our temporal encoder and decoder. The weight sharing heavily reduces the number of weights necessary in the model. We compare the number of parameters necessary to train a three-layer encoder and decoder with linear layers, and thus without weight sharing, on fMRI data to our spatial encoder and decoder, see calculation in Appendix A. The model without weight sharing contains 76M parameters, whereas our model contains 44k parameters. The three-layer model may not seem that large, but relative to the number of samples the model is trained on (1200), this is like training an 896B parameter model on ImageNet Deng et al. (2009). Training a neural network with that many parameters on ImageNet is a recipe for overfitting. Our method, on the other hand, only has 44k parameters, which would be equivalent to training on ImageNet with a 513M parameter model. Making sure the model does not have too many parameters for the number of samples is critical to reducing overfitting. On top of this, fMRI data is noisier and thus even more prone to overfitting than natural image datasets like ImageNet. Furthermore, a recent paper has shown the adverse effects of training a neural network with that many parameters because it worsens its underspecification D'Amour et al. (2020). This is exactly why convolutional neural networks (CNNs) became so popular in computer vision initially, and still are for smaller datasets. CNNs can perform effective weight sharing by re-using the same kernel for the full image, which incorporates some of the inductive biases we have about our own vision. This allows us to stack more convolutional layers on top of each other to find highly non-linear features, without having to worry about the number of solutions. In our case, we apply the same MLPs to each cluster and each cluster's features, which is similar to using the same kernel over a full image, except the metric space we define the clusters over is non-Euclidean. We also know that incorporating inductive biases about the data help with underspecification because they constrain the solution space to solutions that are more neuroscientifically feasible D'Amour et al. (2020). In this paper, we specifically evaluate the difference in solutions across two separate inductive biases, namely structural and functional information. Throughout the paper, we will evaluate both inductive biases to get an idea of their effect on the performance of the model.

Weight sharing Both the spatial encoder and spatial decoder make use of many neuroscientifically-inspired forms of weight sharing that draw similarities to weight sharing in convolutional neural networks (CNNs). For example, if we have a $28x28$ image and use a convolutional layer with a kernel and stride size of 3 and padding of 1 along the image, we get 100 $3x3$ patches of the image. Each patch is shared among the 9 weights of the kernel, meaning the same $3x3$ kernel is applied to each patch. The intuition behind using a local kernel is that pixels close together in Euclidean space are similar. Patches for a CNN are based on the Euclidean distance between the pixels in the image, and can thus intuitively be understood as Euclidean clusters. The assumption that points that are closer in Euclidean space are also more similar is not necessarily true in neuroscience. The concept of distance in neuroscience is based on walking across the surface of the brain. This distance is non-Euclidean due to the brain's folds and is called the geodesic distance; hence we have to define what would be Euclidean clusters but using the geodesic distance. To do this, we propose to use graph clustering based on the geodesic distance between nodes on a graph, see Figure 1A. Defining the clusters using the geodesic distance is referred to as the 'structural' inductive bias in our work. However, since graph clustering is general for any distance metric and allows you to incorporate any spatial inductive bias in a model, we also evaluate a 'functional' distance metric that groups vertices together that are similar in terms of their activity patterns.

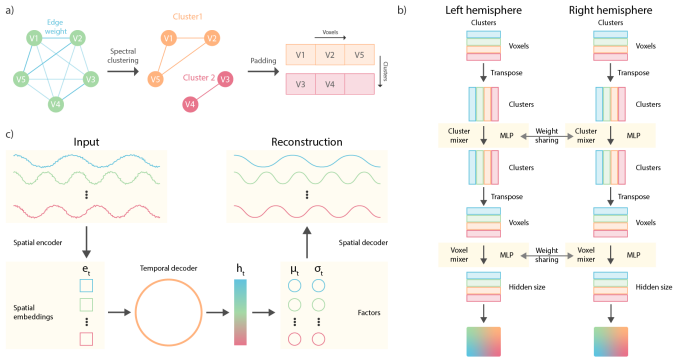

Figure 1: The top left (a) of the figure pictorially illustrates how spectral clustering works. Top right (b) shows the mixer layer we use in both the spatial encoder and decoder. The MLP consists of two linear layers, with an ELU activation on the first layer. The bottom left subfigure (c) shows the full dynamic model.

On top of sharing weights across graph clusters, we also share the weights among the hemispheres. To do this, we pad all other clusters in each hemisphere to have M vertices, where M is the maximum number of vertices in a cluster. Then, for each hemisphere, we use two separate learnable linear layers to map M to a hidden size (in this case, the hidden size is 256) and apply the same layer to each cluster. This gives us C clusters of size 256 per hemisphere. Now, we apply a single spatial encoder to these C clusters for each hemisphere separately and concatenate their output features. Therefore, the spatial encoder shares its weights among the hemispheres. The concatenated output features are then mapped to a desired final feature size using a learnable linear layer and used in our model's temporal encoder and decoder. The spatial encoder is shown in Figure 1C and how it fits into the rest of our model is shown in Figure 1B.

The spatial encoder itself consists of multiple layers, and each layer contains two different MLPs. For every layer, the first MLP mixes the clusters so that for C clusters, the MLP has input size, hidden layer size, and output size C. Cluster mixing means the spatial encoder can learn relationships between features in distant clusters, similar to a CNN's receptive field, see Figure 1B. Thus, the MLP shares its weights across the features in each cluster because it is applied to each of the clusters independently. The second MLP in each layer maps the features (256) in the C clusters and gradually reduces the number of features throughout the layers of the spatial encoder, see Figure 1. This process is similar to the spatial size reduction throughout the layers of a CNN. This second MLP shares its weights across each cluster because it is independently applied to each cluster's features. Our model consists of 3 of these feature mixing layers with hidden sizes $(64, 32, 16, 8, 4, 1)$ for each layer in the MLPs. The spatial decoder is symmetric with the spatial encoder in terms of its hidden sizes.

To conclude, our model shares weights across features and clusters, inspired by the MLPMixer paper Tolstikhin et al. (2021). Although inspired by, we do not use layer normalization and residual connections, but instead use an ELU activation Clevert et al. (2015). However, the patches in our model can be flexibly defined using any distance metric. Hence, our model generalizes the construction of patches to non-Euclidean space, which is critical for efficient neuroscientifically-informed weight sharing in neuroimaging.

Connection to non-linear ICA Variational autoencoders can under some conditions also perform non-linear ICA with identifiability guarantees Khemakhem et al. (2020); Hyvarinen et al. (2019). The way the latent factors are modelled in this work can be considered such a condition, where the additionaly observed variable $\mathbf{u}^{(i)}$ are the previous timesteps in the timeseries. Namely, each factor is a conditional distribution $p_\theta(\mathbf{z}_t^{(i)}|\mathbf{x}_{\leq t}^{(i)})$, where θ correspond to the spatial and temporal encoder, and the temporal decoder. This can be rewritten as $p_\theta(\mathbf{z}_t^{(i)}|\mathbf{x}_t^{(i)}, \mathbf{u}^{i)})$. This is the same formulation for the encoder as in the unifying framework for variational autoencoders and non-linear ICA Khemakhem et al. (2020).

Temporal independence The KL-divergence term in the ELBO effectively acts as a regularization term and in previous works Zhao et al. (2017); Chen et al. (2018) has been shown to be equivalent to the following decomposition with an expectation over the dataset $\mathbb{E}_{\mathcal{D}}$.

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}\left[\mathrm{D_{KL}}\left(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right)\right] &= \mathrm{D_{KL}}\left(q_\phi(\mathbf{z},\mathbf{x})||q_\phi(\mathbf{z})p(\mathbf{x})\right) && \text{(Index-Code MI)}\\ &+ \mathrm{D_{KL}}(q_\phi(\mathbf{z})||\prod_{\mathrm{j}} q_\phi(\mathbf{z}_\mathrm{j})) && \text{(Total correlation)}\\ &+ \sum_{\mathrm{j}} \mathrm{D_{KL}}\left(q_\phi(\mathbf{z}_\mathrm{j})||p(\mathbf{z}_\mathrm{j})\right) && \text{(Dimension-wise KL)}\end{aligned} \tag{2}$$

Where $q_\phi(\mathbf{z}) = \sum_{i=1}^{N} q(\mathbf{z}|\mathbf{x}^{(\mathrm{i})})p(\mathbf{x}^{(\mathrm{i})})$ is the aggregated posterior and $\mathbf{z}_j^{(\mathrm{i})}$ is the j^{th} dimension of the latent factor. Note that with $p(\mathbf{x}^{(\mathrm{i})}$ we refer to the probability that the sample is chosen as a training sample, which is $\frac{1}{N}$. The authors Chen et al. (2018) propose to use minibatch-weighted sampling to get a naïve Monte Carlo estimation of aggregated posterior to compute the total correlation (TC) term and identify the TC term as important to learn disentangled factors. The TC measures the dependency among a set of random variables, in this case, the dimensions of the latent factors. In our case, however, we specifically want to minimize the dependency between the factors over time. Thus, instead of estimating the aggregated posterior using samples in the batch, we estimate it over the timesteps for each subject and take the average TC over the batch. We add the TC term to the ELBO and due to the non-negativity of the KL-divergence, this is still a lower bound.

$$\begin{aligned}\mathcal{L}(\theta,\phi;x_\mathrm{t}^{(\mathrm{i})}) := &-\mathrm{D_{KL}}\left(q_\phi(\mathbf{z}_\mathrm{t}^{(\mathrm{i})}|\mathbf{x}_{\leq \mathrm{t}}^{(\mathrm{i})})||p(\mathbf{z})\right) + \mathbb{E}_{q_\phi(\mathbf{z}_\mathrm{t}^{(\mathrm{i})}|\mathbf{x}_{\leq \mathrm{t}}^{(\mathrm{i})})}\left[\log p_\theta(\mathbf{x}_\mathrm{t}^{(\mathrm{i})}|\mathbf{z}_\mathrm{t}^{(\mathrm{i})})\right]\\ &- \beta\ \mathrm{D_{KL}}(q_\phi(\mathbf{z}_\mathrm{t}^{(\mathrm{i})}||\prod_{j} q_\phi(\mathbf{z}^{(\mathrm{i})})_{\mathrm{t,j}}))\end{aligned} \tag{3}$$

We can now use β to increase or decrease the temporal independence of the factors we learn. This is equivalent to using a TC-VAE Chen et al. (2018) with a minimum β of 1 and a different estimation of the TC.

Implementation The algorithms are implemented using Pytorch Paszke et al. (2017) and are trained on an internal cluster using single NVIDIA GeForce 2800 and NVIDIA V100 GPUs, with a batch size of 8, the Adam optimizerKingma & Ba (2014), a 1E-4 weight decay, a learning rate of 5E-3, 0.1 epsilon, and 0.9, 0.999 as betas. Each instantiation of the algorithm takes about $3-4$ hours to train, based on the graphics card. We also reduce the learning rate when it plateaus using a scheduler, with a 0.95 factor reduction on each plateau, patience of 6 epochs, and a minimum learning rate of 1E-5. L2 norm regularization is also specifically applied to the weight matrix between hidden states in the temporal decoder. We train each model for 150 epochs, across four seeds (42, 1337, 9999, 1212), the epoch with the lowest loss on the validation set is used for the evaluation and/or figures. All necessary code to download and preprocess the data, and run the model will be made publicly available after the double-blind review has concluded on GitHub.

ICA null model Independent Component Analysis (ICA), has been used as a blind source separation to determine different sources of spatial or temporal signals that mix to form the measured signal. The algorithm maximizes the independence of these sources based on either spatial or temporal dissimilarities. ICA has long been used as a gold standard in all neural data, due to its ability to separate sources of neural activity, as well as separate non-neuronal activity, such as motion, respiration, and heartbeat effects. Over time, it has been established as the gold standard in separating spatio-temporal dynamics in EEG, ECOG, MEG, as well as in fMRI datasets Calhoun et al. (2009). We, therefore, utilize ICA as a null model in order to compare our algorithm. The temporal independence results are compared to InfoMax ICA Lee et al. (1999) with the same number of factors as our proposed model. The shortcomings of ICA are that, unlike PCA or other dimensionality techniques, the ICA vectors are unordered and sometimes need manual selection. Moreover, ICA vectors can be noisy for high dimensional data, and prior knowledge, such as in our work, cannot be trivially added to the algorithm.

Comparison with ICA, β-VAE, and varying β The main experiments revolve around comparing our model (TI-VAE) with ICA and a β-VAE, and most importantly across different β values. Both

the functional and structural weight sharing methods are compared as separate models, both for our model and the β-VAE. The experiments with our models, the β-VAE, and ICA are run across 4 different latent dimensions: [5 (ground truth number of factors), 8, 16, 32]. Our model and the β-VAE are run across 4 different seeds, to compute the standard deviation of the performance of the model, and for the following β values: [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 2.0, 3.0]. The β-VAE is run with β values that are common to that model, namely 1.0, 2.0, and 3.0. The goal of this experiment is to test our model's performance against ICA and β-VAE, as well as understand the impact of the independence term.

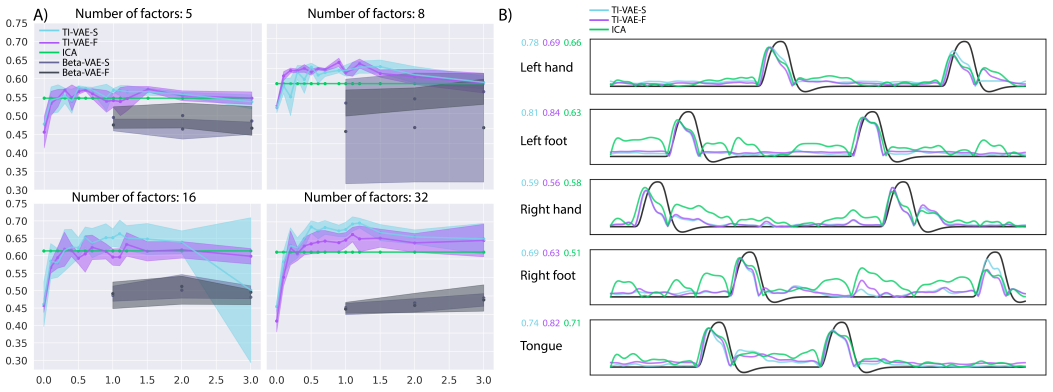

Figure 2: Subfigure A shows the average correlation of the factors corresponding to each sub-task, and for each number of factors. Model name addenda -S and -F correspond to structural or functional weight sharing. The temporal independence term in the loss function (Equation 3) clearly improves the solutions of the learned factors. The right subfigure shows a comparison between the SPM hemodynamic response (black), the best timeseries of our model for each weight sharing regime, and the ICA timecourses. The numbers above the sub-task names correspond to the correlations between each model's timeseries and the SPM hemodynamic response timeseries for that sub-task.

3 RESULTS

Our results show how well our model can correctly identify relevant latent factors from fMRI data. The first section discusses the performance of structural and functional weight sharing to its baseline. We show that the weight sharing we induce is effective and seems to even improve the reconstructions. The algorithm is also demonstrated to outperform the null ICA model, and β-VAE for latent factor identification across all evaluated number of latent dimensions. Lastly, we show how these spatial maps are specific to the motor homunculus and are specific to higher effect sizes in the HCP group maps, and use t-SNE Van der Maaten & Hinton (2008) to show a $2D$ view of the clustering of sub-tasks in the latent factor space.

Temporal independence The indication that the latent factors contain meaningful information regarding the spatio-temporal signal is supported by the high average sub-task correlations, shown in Figure 2 on the left, and the sub-task correlations in the subfigure on the right. Clearly, some sub-tasks are easier to identify for all models than others, but our models both outperform ICA for some values of β. Furthermore, our model with structural weight sharing outperforms ICA more than a standard deviation at multiple beta values for each of the latent dimensions. This result is a clear demonstration that our method is valuable, especially because our model is fully differentiable, non-linear, and can easily be extended to other data, or be combined with other modalities. To get some more insight into the spatial locations that correspond to the factors of the best performing models, we plot them in Figure 3. The spatial maps for our model are created by interpolating each latent factor independently from its minimum value in the training and validation set in the latent space, to its maximum value with 50 steps, and then taking the variance over those steps in the reconstructed surface space. The spatial maps are thus non-negative, whereas the ICA spatial maps can be negative. To deal with this, we use the sign of the correlation for each ICA factor

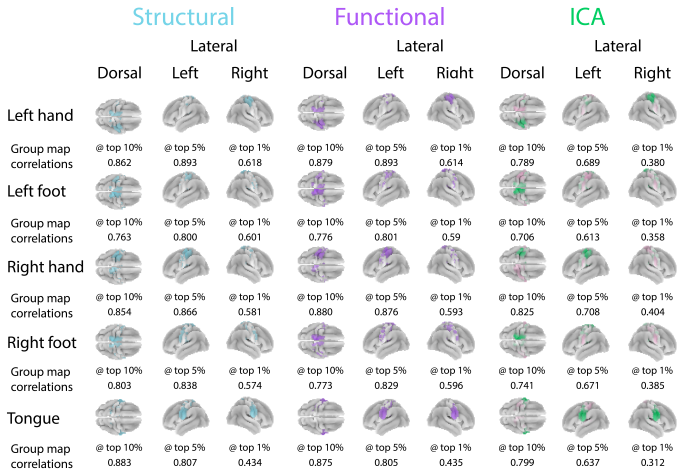

Figure 3: Each model's spatial maps corresponding to the sub-tasks (left), with structural weight sharing on the left, functional weight sharing in the middle, and ICA on the right. Note that for the dorsal view, the bottom hemisphere in the figure corresponds to the right hemisphere. We also show the correlation between the top 10 %, top 5 %, and top 1 % values of the HCP group average effect size maps with the produced spatial maps. Our model is highly specific and sparse, which is reflected by its high correlation to the top values.

with the sub-task it corresponds to and multiply the corresponding spatial map with its sign. For visualization, the bottom 25 % and top 75 % vertices are shown for the ICA, and the top 80 % are shown for our models.

Task dynamics. To get an insight into the dynamics of the factors, we plot a $2D$ t-SNE Van der Maaten & Hinton (2008) projection of the average timeseries over the subjects in the unseen test set, for all of the 32 factors in the best models. Each point in Figure 4 corresponds to a time point from the average timeseries and is colored based on which task it corresponds to, where gray points correspond to time points without a task. Since there is a delay in the BOLD response to a task, the first 5 timepoints at the start of a task are made gradually more opaque, from 0.5 to 1.0, and the last 5 timepoints are made gradually less opaque, from 1.0 to 0.5 for each task. We do not expect the first timepoints after the task starts to elicit a response, so some of the colored points may not be clustered together. The trajectories for both the structural (left) and functional (right) weight sharing are shown in Figure 4. They show clear clusters for each sub-task, which are each performed twice in the timeseries. The same sub-task is not performed subsequently, making the clustering non-trivial. An interesting finding is that the feet seem to be clustered together, and that the right foot and left hand appear close in both the structural and functional inductive bias. In the functional inductive bias, the left and right hand are also close together, which is not necessarily true for the structural bias. Further research into the other, possible non-linear factors that our model finds needs to be done to fully understand these trajectories, but they re-affirm that our model finds meaningful factors, even without selecting the highest correlation ones.

4 Discussion

The spatial maps in Figure 3 correspond to the functional motor homunculus and are correlated with the top highest effect sizes in the HCP group Cohen D value maps. Namely, the spatial maps for the left and right hands are located superior and laterally on the left and right hemispheres, respectively. The locations of spatial maps for the left and right foot are located superior and more medially in the brain, on the left and right hemispheres, respectively. The spatial maps corresponding to the tongue are located inferior to the other sub-tasks, and laterally in both the right and left hemispheres. Given that our model learns these spatial maps over the whole dataset and that individual spatial maps can differ per subject, it is expected that the tongue spatial map occurs in both hemispheres. Another interesting finding is the difference between structural and functional spatial maps, namely that the

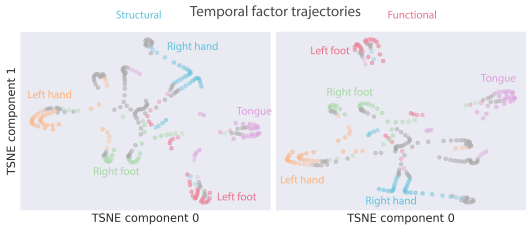

Figure 4: The average latent trajectory projected to $2D$ using t-SNE Van der Maaten & Hinton (2008) for the test set, for both the model with structural (left) and functional (right) inductive biases. Each point in the plot corresponds to a time point and the color corresponds to the task they are in, where gray corresponds to not being in a task. We expect a delay in the reaction due to the hemodynamic response, the first and last 5 timesteps into a task have a gradually increasing and at the end decreasing opacity.

structural inductive bias finds local regions, whereas the functional map sometimes finds regions more spread throughout the brain. This aligns with our expectations because spectral clusters for the functional inductive bias are based on temporal correlations, compared to the geodesic distance for the structural inductive bias. Additionally, both the structural and functional inductive bias find more localized regions that correspond more directly to the functional human motor homunculus, clearly indicating the usefulness of our model. The relationships between the timeseries of each factor and the sub-task may be linear in some cases, which would mean ICA is more appropriate. Given that our model learns those components and can learn non-linear components, our framework opens up a field of future work with non-linear fMRI components.

Limitations. One limitation of the model is that it has only been applied to the somatomotor cortex. This was done to have a good idea of ground-truth spatial and temporal factors we expect to find with our model. The somatomotor cortex is an extensively studied area and has largely been mapped out from a whole-brain perspective. However, it is important to test our model on larger input data in future work to make sure it holds up for whole-brain data. Furthermore, the SPM simulated hemodynamic response is not a perfect model for BOLD activation in the brain and we use a group surface to create the spectral graph clustering, instead of subject-based surfaces.

Broader impact. The current model can have implications for surgical mapping, where functional connectivity based on ICA components is sometimes used. This model does require further and more extensive testing before it can be used in a clinical setting, however. The model's ability to learn non-linear factors can be both a positive and negative aspect of the model. The model can learn subject-specific factors that are not linearly related to group-based factors, as is common in ICA. This is important in a clinical setting, but could potentially lead to learning negative biases in the dataset.

Conclusion The model we propose in this work is a leap toward a fully-differentiable non-linear framework for whole-brain dynamic factor learning. We show that temporal independence is crucial to learning meaningful factors and our model outperforms ICA when the extra term that encourages temporal independence is added to the loss function. Our model can also comfortably scale to larger inputs with its novel weight sharing technique. In fact, weight sharing in our model does not degrade the reconstructions of the data under large dimensionality reduction (from $11k$ voxels to 16 factors) compared to a baseline. In future work, we want to apply this model to more tasks, larger input data, multiple modalities, and resting-state fMRI data.

ACKNOWLEDGMENTS

Data were provided by the Human Connectome Project, WU-Minn Consortium (Principal Investigators: David Van Essen and Kamil Ugurbil; 1U54MH091657) funded by the 16 NIH Institutes and Centers that support the NIH Blueprint for Neuroscience Research; and by the McDonnell Center for Systems Neuroscience at Washington University Other acknowledgments will be added once the double-blind review process has concluded.

REFERENCES

Alexandre Abraham, Fabian Pedregosa, Michael Eickenberg, Philippe Gervais, Andreas Mueller, Jean Kossaifi, Alexandre Gramfort, Bertrand Thirion, and Gaël Varoquaux. Machine learning for neuroimaging with scikit-learn. *Frontiers in neuroinformatics*, 8:14, 2014.

Christian F Beckmann and Stephen M Smith. Tensorial extensions of independent component analysis for multisubject fmri analysis. *Neuroimage*, 25(1):294–311, 2005.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in \beta $-vae. *arXiv preprint arXiv:1804.03599*, 2018.

Vince D Calhoun and Tülay Adali. Unmixing fmri with independent component analysis. *IEEE Engineering in Medicine and Biology Magazine*, 25(2):79–90, 2006.

Vince D Calhoun, Jingyu Liu, and Tülay Adalı. A review of group ica for fmri data and ica for joint inference of imaging, genetic, and erp data. *Neuroimage*, 45(1):S163–S172, 2009.

Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint arXiv:2011.03395*, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Xavier Descombes, Frithjof Kruggel, and D Yves Von Cramon. Spatio-temporal fmri analysis using markov random fields. *IEEE transactions on medical imaging*, 17(6):1028–1039, 1998.

B Everett. *An introduction to latent variable models*. Springer Science & Business Media, 2013.

Bruce Fischl. Freesurfer. *Neuroimage*, 62(2):774–781, 2012.

Siyuan Gao, Gal Mishne, and Dustin Scheinost. Non-linear manifold learning in fmri uncovers a low-dimensional space of brain dynamics. *bioRxiv*, 2020.

Eloy Geenjaar, Tonya White, and Vince Calhoun. Variational voxelwise rs-fmri representation learning: Evaluation of sex, age, and neuropsychiatric signatures. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1733–1740. IEEE, 2021.

Matthew F Glasser, Stamatios N Sotiropoulos, J Anthony Wilson, Timothy S Coalson, Bruce Fischl, Jesper L Andersson, Junqian Xu, Saad Jbabdi, Matthew Webster, Jonathan R Polimeni, et al. The minimal preprocessing pipelines for the human connectome project. *Neuroimage*, 80:105–124, 2013.

Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):855–868, 2008.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6645–6649. Ieee, 2013.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

Irina Higgins, Sébastien Racanière, and Danilo Rezende. Symmetry-based representations for artificial and biological general intelligence. *Frontiers in Computational Neuroscience*, pp. 28, 2022.

R Devon Hjelm, Vince D Calhoun, Ruslan Salakhutdinov, Elena A Allen, Tulay Adali, and Sergey M Plis. Restricted boltzmann machines for neuroimaging: an application in identifying intrinsic networks. *NeuroImage*, 96:245–260, 2014.

Aapo Hyvarinen, Hiroaki Sasaki, and Richard Turner. Nonlinear ica using auxiliary variables and generalized contrastive learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 859–868. PMLR, 2019.

Amrit Kashyap and Shella Keilholz. Brain network constraints and recurrent neural networks reproduce unique trajectories and state transitions seen over the span of minutes in resting-state fmri. *Network Neuroscience*, 4(2):448–466, 2020.

Mohammad Reza Keshtkaran, Andrew R Sedler, Raeed H Chowdhury, Raghav Tandon, Diya Basrai, Sarah L Nguyen, Hansem Sohn, Mehrdad Jazayeri, Lee E Miller, and Chethan Pandarinath. A large-scale neural network training framework for generalized estimation of single-trial population dynamics. *bioRxiv*, 2021.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pp. 2207–2217. PMLR, 2020.

Jung-Hoon Kim, Yizhen Zhang, Kuan Han, Zheyu Wen, Minkyu Choi, and Zhongming Liu. Representation learning of resting state fmri with variational autoencoder. *NeuroImage*, 241:118423, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Georgia Koppe, Hazem Toutounji, Peter Kirsch, Stefanie Lis, and Daniel Durstewitz. Identifying nonlinear dynamical systems via generative recurrent neural networks with applications to fmri. *PLoS computational biology*, 15(8):e1007263, 2019.

Kangjoo Lee, Sungho Tak, and Jong Chul Ye. A data-driven sparse glm for fmri analysis using sparse dictionary learning with mdl criterion. *IEEE Transactions on Medical Imaging*, 30(5): 1076–1089, 2010.

Te-Won Lee, Mark Girolami, and Terrence J Sejnowski. Independent component analysis using an extended infomax algorithm for mixed subgaussian and supergaussian sources. *Neural computation*, 11(2):417–441, 1999.

Guixiang Ma, Lifang He, Chun-Ta Lu, Philip S Yu, Linlin Shen, and Ann B Ragin. Spatio-temporal tensor analysis for whole-brain fmri classification. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pp. 819–827. SIAM, 2016.

Martin J McKeown and Terrence J Sejnowski. Independent component analysis of fmri data: examining the assumptions. *Human brain mapping*, 6(5-6):368–372, 1998.

Chethan Pandarinath, Daniel J O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D Stavisky, Jonathan C Kao, Eric M Trautmann, Matthew T Kaufman, Stephen I Ryu, Leigh R Hochberg, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10):805–815, 2018.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

William D Penny, Karl J Friston, John T Ashburner, Stefan J Kiebel, and Thomas E Nichols. *Statistical parametric mapping: the analysis of functional brain images*. Elsevier, 2011.

Emma C Robinson, Saad Jbabdi, Matthew F Glasser, Jesper Andersson, Gregory C Burgess, Michael P Harms, Stephen M Smith, David C Van Essen, and Mark Jenkinson. Msm: a new flexible framework for multimodal surface matching. *Neuroimage*, 100:414–426, 2014.

Emma C Robinson, Kara Garcia, Matthew F Glasser, Zhengdao Chen, Timothy S Coalson, Antonios Makropoulos, Jelena Bozek, Robert Wright, Andreas Schuh, Matthew Webster, et al. Multimodal surface matching with higher-order smoothness constraints. *Neuroimage*, 167:453–465, 2018.

Christopher G Thomas, Richard A Harshman, and Ravi S Menon. Noise reduction in bold-based fmri using component analysis. *Neuroimage*, 17(3):1521–1537, 2002.

Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems*, 34, 2021.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

David C Van Essen, Stephen M Smith, Deanna M Barch, Timothy EJ Behrens, Essa Yacoub, Kamil Ugurbil, Wu-Minn HCP Consortium, et al. The wu-minn human connectome project: an overview. *Neuroimage*, 80:62–79, 2013.

Mark William Woolrich, Mark Jenkinson, J Michael Brady, and Stephen M Smith. Fully bayesian spatio-temporal modeling of fmri data. *IEEE transactions on medical imaging*, 23(2):213–231, 2004.

BT Thomas Yeo, Fenna M Krienen, Jorge Sepulcre, Mert R Sabuncu, Danial Lashkari, Marisa Hollinshead, Joshua L Roffman, Jordan W Smoller, Lilla Zöllei, Jonathan R Polimeni, et al. The organization of the human cerebral cortex estimated by intrinsic functional connectivity. *Journal of neurophysiology*, 2011.

Byron M Yu, John P Cunningham, Gopal Santhanam, Stephen Ryu, Krishna V Shenoy, and Maneesh Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Advances in neural information processing systems*, 21, 2008.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*, 2017.

A APPENDIX: MODEL PARAMETER CALCULATION

We calculate how many parameters we need in a model with linear layers (and thus no weight sharing) trained on fMRI input volumes of roughly 53-53-53 voxels. The size of the fMRI input volume can differ, but 53-53-53 voxels is a normal size. Let us assume the linear model has hidden hidden sizes of 256 and 128 ,and 16 factors. The following code then allows us to calculate the number of parameters for this three-layer encoder, and three-layer decoder model:

```
import torch
import numpy as np
from torch import nn
from architectures import DoubleMixer

model_a = nn.Sequential(
nn.Linear(53 * 53 * 53, 256),
nn.Linear(256, 128),
nn.Linear(128, 16),
nn.Linear(16, 128),
nn.Linear(128, 256),
nn.Linear(256, 53 * 53 * 53))
print('Number of modelA parameters: '
      f'{sum(p.numel() for p in model_a.parameters() if p.requires_grad)}')
```

The output of the code is 76M parameters. To calculate the number of parameters of an equivalent model with our encoder and decoder, that also has hidden sizes of 256 and 128, and a latent factor size of 16, we would use the following code:

```
model_b = nn.Sequential(
nn.Linear(int(np.ceil(((53 * 53 * 53) / 128) / 2)), 256),
DoubleMixer(in_tokens=128, hid_tokens=128, out_tokens=128,
            in_size=256, hid_size=128, out_size=1),
nn.Linear(128 * 2, 16),
nn.Linear(16, 128 * 2),
DoubleMixer(in_tokens=128, hid_tokens=128, out_tokens=128,
            in_size=1, hid_size=128, out_size=256),
nn.Linear(256, int(np.ceil(((53 * 53 * 53) / 128) / 2))))
print('Number of modelB parameters: '
      f'{sum(p.numel() for p in model_b.parameters() if p.requires_grad)}')
```

The first linear layer is shared between hemispheres, and each hemisphere is clustered into 128 clusters. In this case, we assume that the sizes of the clusters is roughly uniform. The number of parameters for this model is: 44k. Relative to the number of samples the model is trained on (1200), the linear model would be analogous to training an 896B parameter model on ImageNet. Training a neural network with that many parameters on ImageNet is a recipe for overfitting. Our method, on the other hand, only has 44k parameters, which would be equivalent to training on ImageNet with a 513M parameter model.

B APPENDIX: SPECTRAL CLUSTERING

There are three main steps in spectral clustering. First, we create the adjacency matrix ($\mathbf{A}_s$ or $\mathbf{A}_f$) and normalize it between 0 and 1. We assume a fully-connected graph within each hemisphere, so the degree matrix of the graph is the total number of vertices on the diagonal. Then, the graph Laplacian of the graph is computed as $\mathbf{L}_s = \mathbf{D}_s - \mathbf{A}_s$ and similarly for the functional adjacency matrix. Second, the graph Laplacian is decomposed using its eigendecomposition and only the bottom k smallest eigenvalues are used, the others are discarded. The k smallest eigenvalues each correspond to a cluster (eigenvector) of the graph Laplacian.