Cross-subject Mapping of Neural Activity with Restricted Boltzmann Machines

Marko Angjelichinoski[1,†], Suya Wu[1], Joy Putney[2,‡], Simon Sponberg[2], Vahid Tarokh[1]

1 Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA
2 School of Physics and Biological Sciences, Georgia Institute of Technology, Atlanta, GA, USA

†Marko Angjelichinoski contributed to this work when he was at Duke University.
‡Joy Putney contributed to this work when she was at Georgia Institute of Technology.
¶This work was supported in part by Air Force Office of Scientific Research (AFOSR) Grant No. FA9550-22-1-0315. Marko Angjelichinoski was supported in part by the Army Research Office Contract Number W911NF16-1-0368.
* suya.wu@duke.edu

1 Abstract

Subject-to-subject variability is a common challenge both generalizing models of neural data across subjects, discriminating subject-specific and inter-subject features in large neural datasets, and engineering neural interfaces with subject-specific tuning. We study the problem of the cross-subject mapping of neural activity. The objective is to obtain a task-specific representation of the source subject signal into the feature space of the destination subject. We propose to use the Restricted Boltzmann Machine (RBM) with Gaussian inputs and Bernoulli hidden units; once trained over the entire set of subjects, the RBM allows the mapping of source features on destination feature spaces using Gibbs sampling. We also consider a novel computationally efficient training technique for RBMs based on the minimization of the Fisher divergence, which allows the gradients of the RBM to be computed in closed form. We use neural decoding as a downstream application to test the method. Specifically, we test decoding on neuromuscular recordings of spike trains from the ten muscles that primarily control wing motion in an agile flying hawk moth, *Manduca sexta*. The dataset consists of this comprehensive motor program recorded from nine subjects, each driven by six discrete visual stimuli. The evaluations show that the source features can be decoded using the destination classifier with an accuracy of up to 95% when mapped using an RBM trained by Fisher divergence, showcasing the promising potential of the RBMs for cross-subject mapping applications.

2 Author summary

In this study, we address the variability of neural data across subjects, which is a significant obstacle in developing models that can generalize across subjects. Our objective is to create a task-specific representation of the source subject signal in the feature space of the destination subject. To this end, we consider the applications of the Restricted Boltzmann Machine (RBM) with Gaussian inputs and Bernoulli hidden units, trained on the joint feature space of the source subject and destination subject.

The trained RBM can then be used to map source features onto the destination feature spaces using Gibbs sampling. We also present a novel, score-based computationally efficient training technique for RBMs based on Fisher divergence. Using neural decoding as a downstream application, we demonstrate the effectiveness of our method on neuromuscular recordings of spike trains from the ten muscles controlling wing motion in an agile flying hawk moth, *Manduca sexta*, recorded from nine subjects. Numerical evaluations show that the source features can be accurately decoded using the destination classifier with up to 95% accuracy when mapped using an RBM trained by Fisher divergence.

3 Introduction

Learning algorithms in neuroscience are required to generalize well across unseen subjects of a population. Yet, implementing reliable *cross-subject* algorithms in neuroscience is a notoriously challenging problem. An important factor contributing to its difficulty arises from the *non-stationary* nature of the neural activity signals, whose statistical properties vary dramatically even under slight changes in the recording conditions [1, 2]. As a result, the algorithms trained and optimized on data collected from a given subject, fail to perform reliably when directly applied to other subjects. For instance, a neural decoder trained on one subject will perform close to a random choice classifier if applied directly to a different subject, thus failing to identify the correct neurological state or stimulus condition even when the subjects perform the same tasks simultaneously [3].

Problems of this type, i.e., problems where the training and test data originate from different distributions, are common in machine learning and are typically tackled within the sub-field of *transfer learning*. In the context of the cross-subject problem, various approaches have already been considered [2]. In our previous work on this problem, we focused on domain adaptation methods that map the source data onto the destination feature spaces [3]. Generative modeling is another promising approach to cross-subject mapping. More recently, we have also considered using directed graphs, such as conditional variational autoencoder (cVAE), to generate the source data onto the feature space of the destination data [4], where the learning model for the downstream task is trained. This approach, however, requires a separate directed graphical model to be trained each time a new downstream task and/or new destination subject is considered.

In this paper, we propose to use undirected graphs to generate samples for cross-subject mapping. Specifically, we suggest training a Restricted Boltzmann Machine (RBM) [5, 6] for this purpose, as described in Section 4. RBM is a popular generative model that has had notable success in representation learning with applications in a wide variety of tasks in neuroscience [7–11]. We note that the applications of RBMs in transfer learning have been studied before [12–14] in computer vision applications.

The main objective of cross-subject learning is to obtain a task-specific representation of the neural activity of one or more source subjects in the destination spaces of one or multiple destination subjects. Once the RBM is trained using a given set of subjects, it can be used to map signals from any source subject within the same set of subjects onto any destination feature space. We also consider an alternative training method for RBMs based on Fisher divergence minimization [15]. In contrast with the conventional contrastive divergence training (which is equivalent to maximum likelihood, we refer more details to [5]), the Fisher divergence minimization allows the gradient of the RBM to be computed in closed form, fostering efficient implementation that does not require iterative Gibbs sampling during training.

In Section 5, we evaluate the performance of our method for cross-subject decoding

of discrete visual stimulus conditions using the spiking activity of the motor program, specifically the set of spiking motor units, in nine hawk moths [16]. Each moth is exposed to the same set of six visual stimuli and the neuromuscular activity is collected in the form of spike trains extracted from fine wire electromyography (EMGs) of the ten primary flight muscles that control the wings, resulting in a comprehensive, spike-resolved motor program [17]. Unlike vertebrate EMGs, these flight muscles act as effectively single motor units and result in identifiable spike trains comparable to population recordings of individual units elsewhere in the brain or peripheral nervous system. Our results demonstrate the promising potential of the proposed framework, with respectively up to 90% and 95% accuracy in decoding the behavioral state (i.e., the visual stimulus), when using the RBM trained with classical and the new Fisher divergence-based methods.

4 Cross-subject Mapping with Restricted Boltzmann Machines

We divide this section into three parts. In Section 4.1 we present the statistical formulation of the problem of cross-subject mapping as a problem of learning joint distribution between source and destination feature vectors. Next, in Section 4.2, we discuss RBMs and present both the contrastive divergence and Fisher divergence training methods. Section 4.3 presents a simple cross-subject mapping approach that uses Gibbs sampling to draw samples from the joint distribution of the source and destination features.

4.1 Problem Statement

The objective of cross-subject learning is to map *source* subject features to the feature space of one or more *destinations* subjects so that the motor intentions of *source* subject can be decoded by the neural decoder that learned on one or more *destination* subjects. In other words, we aim to obtain the appropriate *destination* space representation of the tasks that the source subjects perform. Technically, the problem boils down to finding a function that maps the feature vectors of the *source* subjects onto the feature space of one or more destination subjects. In principle, the mapping function can be assumed to be purely deterministic. However, in this paper, we adopt a probabilistic approach, which generates task-specific representations of *source* subjects into the feature spaces of one or more *destination* subjects. We outline details below.

Let $\mathcal{M}_\mathrm{S}$ and $\mathcal{M}_\mathrm{D}$ denote the index sets of the *source* and *destination* subjects, respectively. Further, we let $\mathcal{M}$ denote the joint set of all subjects, namely $\mathcal{M} = \mathcal{M}_\mathrm{S} \cup \mathcal{M}_\mathrm{S}$. For simplicity, we assume that the subsets $\mathcal{M}_\mathrm{S}$ and $\mathcal{M}_\mathrm{D}$ are disjoint, namely $\mathcal{M}_\mathrm{S} \cap \mathcal{M}_\mathrm{S} = \emptyset$. Let $\mathbf{x}_m \in \mathbb{R}^{D_m}$ denote the D_m-dimensional vector representing the neural activity of subject $m \in \mathcal{M}$; we refer to $\mathbf{x}_m$ as the *feature vector* of subject m. Furthermore, we use $\mathbf{x}_\mathrm{S} = (\mathbf{x}_i)_{i\in\mathcal{M}_\mathrm{S}}$ and $\mathbf{x}_\mathrm{D} = (\mathbf{x}_j)_{j\in\mathcal{M}_\mathrm{D}}$ to respectively denote the joint feature vectors of the *source* and *destination* subjects. Note that $\mathbf{x}_\mathrm{S}$ and $\mathbf{x}_\mathrm{D}$ are vectors with dimensions $D_\mathrm{S} = \sum_{i\in\mathcal{M}_\mathrm{S}} D_i$ and $D_\mathrm{D} = \sum_{i\in\mathcal{M}_\mathrm{D}} D_i$, respectively. Finally, we use $\mathbf{x}$ to denote the joint vector of features of *all* subjects in the set $\mathcal{M}$; that is $\mathbf{x} = (\mathbf{x}_\mathrm{S}, \mathbf{x}_\mathrm{D}) = (\mathbf{x}_m)_{m\in\mathcal{M}}$, and its dimension $D = \sum_{m\in\mathcal{M}} D_m$.

To learn the cross-subject mapping, we consider a *conditional probability distribution* to generate feature representations in the feature space of *destination* subjects given the feature vector of *source* subjects. One option is to directly parameterize the probability density function $p(\mathbf{x}_\mathrm{D}|\mathbf{x}_\mathrm{S})$ of this conditional distribution. Another approach is to first learn the distribution $p(\mathbf{x}_\mathrm{S})$ and the joint distribution $p(\mathbf{x}_\mathrm{D}, \mathbf{x}_\mathrm{S})$ of all feature vectors

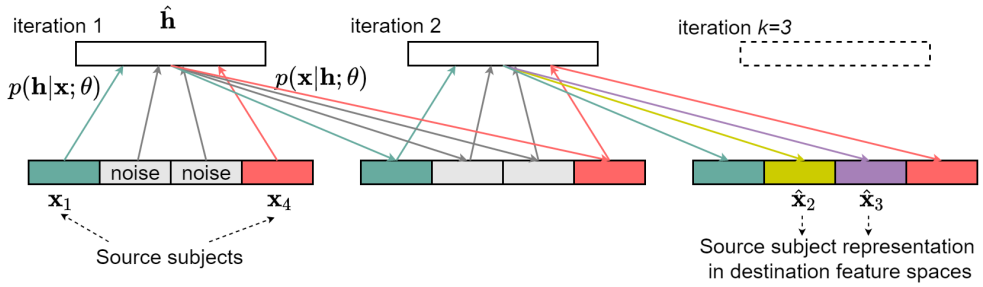


Fig 1. A generative framework for the cross-subject mapping. We iterate k times back and forth through the sampling procedure.

across the entire population of subjects in $\mathcal{M}$, and then to obtain the conditional distribution $p(\mathbf{x}_\mathrm{D}|\mathbf{x}_\mathrm{S})$ by Bayes' theorem.

For the purpose of cross-subject mapping, it is not mandatory to learn an explicit probability density function (pdf) of the conditional distribution. Recall that the objective is to obtain the feature representations of the *source* subject in the feature space of *destination* subject. In this paper, we propose a sampling scheme to obtain this feature representation. Specifically, we consider a generative model $p(\mathbf{x})$ of the concatenated feature vector $\mathbf{x} = (\mathbf{x}_\mathrm{S}, \mathbf{x}_\mathrm{D})$, and we use Gibbs sampling to sample from $p(\mathbf{x})$ with hidden variables $\mathbf{h}$. The Gibbs sampler proceeds as follows: we initialize the visible variables by $\mathbf{x}^{(0)} = (\mathbf{x}_\mathrm{S}^{(0)}, \mathbf{x}_\mathrm{D}^{(0)})$. Here, $\mathbf{x}_\mathrm{S}^{(0)} = \mathbf{x}_\mathrm{S}$ is given, and $\mathbf{x}_\mathrm{D}^{(0)}$ is a dummy and noise-like vector. Next, we sample hidden variables $\hat{\mathbf{h}}$ from $p(\mathbf{h}|\mathbf{x};\theta)$, and then sample visible variables $\mathbf{x}^{(1)}$ from $p(\mathbf{x}|\mathbf{h};\theta)$. After sufficient sampling iterations, we expect to obtain $\mathbf{x}^{(k)} = (\mathbf{x}_\mathrm{S}^{(k)}, \mathbf{x}_\mathrm{D}^{(k)})$ where the $\mathbf{x}_\mathrm{D}^{(k)}$ is our target, the feature representation of $\mathbf{x}_\mathrm{S}$ in the feature space of *destination* subjects. The hidden layer $\mathbf{h}$ is designed to bridge *source* and *destination* feature vectors. We illustrate this generative framework in Fig. 1.

To this end, we aim to learn a generative framework such that we can easily sample from the conditional distributions $p(\mathbf{x}|\mathbf{h};\theta)$ and $p(\mathbf{h}|\mathbf{x};\theta)$. In the previous work [4], we considered cVAE [18], the directed graphs, to map the *source* feature onto the feature space of the *destination* subjects by the decoder of the autoencoder. However, directed graphs are not flexible to adopt new destination subjects, and the architecture of cVAE is built on deep neural networks which are not easy to fine-tune given the limited size of data. In this paper, we consider the class of undirected graphical models and easily adapt to the generative framework. Given its relatively simple architecture and straightforward sampling scheme, RBM is the first generative model we will tackle. The details are elaborated on below.

4.2 Learning Restricted Boltzmann Machines

We will first discuss the Gauss-Bernoulli RBMs and outline the principles of their training. We then review the standard training technique that aims to minimize the contrastive divergence, which is equivalent to minimizing the Kullback-Leibler (KL) divergence between the data-generating distribution and the model. We also consider an alternative training technique that minimizes the Fisher divergence. Unlike the classical method, minimizing the Fisher divergence approach allows the gradients of the loss function to be computed in closed form. This improves the computational efficiency and reliability of the training.

4.2.1 Notation

We adopt the following notation conventions. Recall that in the context of the cross-subject mapping problem outlined in Section 4.1, the vector $\mathbf{x}$ comprises the feature vectors of the entire population of subjects, i.e., $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_M)$. Let $\nabla_{\mathbf{x}}$ and $\Delta_{\mathbf{x}}$ denote the gradient and Laplacian operator with respective to (w.r.t.) the vector $\mathbf{x}$. Let $\mathrm{dg}(\mathbf{x})$ denote a diagonal matrix whose main diagonal is $\mathbf{x}$. For a square matrix $\mathbf{A}$, let $\mathrm{dg}(\mathbf{A})$ denote a diagonal matrix formed by setting all the elements to $\mathbf{A}$ not on the main diagonal to zeroes. We use $\|\mathbf{x}\|$ (respectively $\|\mathbf{A}\|$) to denote the L_2-norm of the $\mathbf{x}$ (respectively the Frobenius norm of $\mathbf{A}$). We further use $p_*(\mathbf{x})$ to denote the true data-generating distribution of $\mathbf{x}$. In practice, $p_*(\mathbf{x})$ is usually unknown; therefore, given a set of observations, a standard problem is to estimate the model density $p(\mathbf{x})$ from some model class that best explains the data under an appropriate evaluation metric. In this paper, we focus on a parametric density model class $p(\mathbf{x}; \boldsymbol{\theta}), \boldsymbol{\theta} \in \Theta$, which is parameterized as an RBM (see Section 4.2.4).

4.2.2 Kullback-Leibler Divergence and the Logarithmic Loss

A common practice to measure the deviation of a postulated probability distribution $p(\mathbf{x})$ from the data-generating distribution $p_*(\mathbf{x})$ is to use the KL divergence defined by

$$D_{\mathrm{KL}}(p_*, p) = -\mathbb{E}_* \left[\log p(\mathbf{x}; \boldsymbol{\theta})\right] + \mathbb{E}_* \left[\log p_*(\mathbf{x})\right], \tag{1}$$

where the expectation is taken w.r.t. $p_*(\mathbf{x})$ (as denoted by the subscript). $D_{\mathrm{KL}}(p_*, p) \geq 0$ with equality if and only if $p = p_*$ almost surely. The minimization of $D_{\mathrm{KL}}(p_*, p)$ is equivalent to the minimization $\mathbb{E}_* \left[\ell(\mathbf{x}; \boldsymbol{\theta})\right]$ where $\ell(\mathbf{x}; \boldsymbol{\theta}) = -\log p(\mathbf{x}; \boldsymbol{\theta})$ is called the logarithmic loss. Let $\boldsymbol{\theta}_*$ denote the data-generating parameter that minimizes the KL divergence in Eq.(1), namely $p(\mathbf{x}; \boldsymbol{\theta}_*)$ is closest to $p_*(\mathbf{x})$ among all distributions over Θ under the KL divergence. It can be shown from the law of large numbers and standard regularity conditions [19] that the maximum likelihood estimate (MLE) $\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \arg\min_{\boldsymbol{\theta}} \bar{\ell}(\mathbf{x}; \boldsymbol{\theta})$ satisfies $\hat{\boldsymbol{\theta}}_{\mathrm{ML}} \to \boldsymbol{\theta}_*$ in probability as the number of data points grows. In other words, the MLE is consistent.

4.2.3 Fisher Divergence and Hyvärinen Score

The Fisher divergence of $p(\mathbf{x}, \boldsymbol{\theta})$ from the data-generating pdf $p_*(\mathbf{x})$ is defined by

$$D_{\mathrm{F}}(p_*, p) = \frac{1}{2}\mathbb{E}_* \left[\|\nabla_{\mathbf{x}} \log p(\mathbf{x}; \boldsymbol{\theta}) - \nabla_{\mathbf{x}} \log p_*(\mathbf{x})\|^2\right], \tag{2}$$

where the expectation is again taken w.r.t. the data-generating pdf $p_*(\mathbf{x})$. We note that $D_{\mathrm{F}}(p_*, p) \geq 0$ with equality if and only if $p_* = p$ almost surely. Under mild regularity conditions, the Fisher divergence Eq.(2) can be written as [15]

$$D_{\mathrm{F}}(p_*, p) = \mathbb{E}_* \left[s_{\mathrm{F}}(\mathbf{x}, \boldsymbol{\theta})\right] + c_*, \tag{3}$$

where c_* is a term that does not depend on $\boldsymbol{\theta}$ and $s_{\mathrm{F}}(\mathbf{x}, \boldsymbol{\theta})$ is the Hyvärinen Score, defined as

$$s_{\mathrm{F}}(\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{2}\|\nabla_{\mathbf{x}} \log p(\mathbf{x}; \boldsymbol{\theta})\|^2 + \Delta_{\mathbf{x}} \log p(\mathbf{x}; \boldsymbol{\theta}). \tag{4}$$

The result Eq.(3) enables to minimize the Fisher divergence over the space of parameters Θ by minimizing the empirical analog of the Hyvärinen Score $\bar{s}_{\mathrm{F}}(\mathbf{x}_n, \boldsymbol{\theta})$. Let $\boldsymbol{\theta}_*$ denote the parameter value that minimizes the Fisher divergence between $p(\mathbf{x}; \boldsymbol{\theta}_*)$ to $p^*(\mathbf{x})$ between all model class candidates. By standard asymptotic analysis, it can be

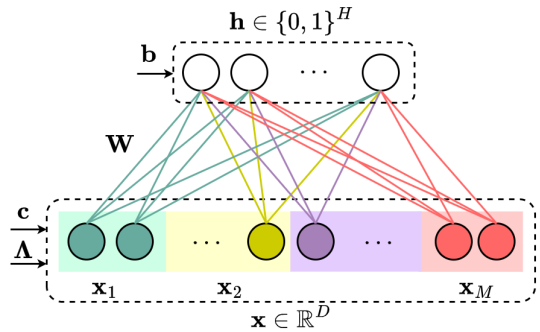


Fig 2. A bipartite undirected graphical model: Gauss-Bernoulli restricted Boltzmann machine with continuous input and binary hidden units.

shown that the estimate $\hat{\boldsymbol{\theta}}_{\mathrm{F}} = \arg\min_{\boldsymbol{\theta}} \bar{s}_{\mathrm{F}}(\mathbf{x}_n, \boldsymbol{\theta})$ satisfies $\hat{\boldsymbol{\theta}}_{\mathrm{F}} \to \boldsymbol{\theta}_*$ in probability as the number of data points grows (we refer details to [15] and references therein). This estimation procedure is known as score matching. It has been proved that score matching using the Langevin Monte Carlo method is equivalent to contrastive divergence in the limit of infinismial step size [20]. Although this result implies that this variant of convergence divergence can retain the consistency on score matching, we note that this equivalence holds only for a particular MCMC method. The actual performance of these two methods are different.

4.2.4 Gauss-Bernoulli Restricted Boltzmann Machines

An RBM is a bipartite undirected graphical model where only the links between visible units and hidden units are allowed. We focus on Gauss-Bernoulli RBM, which consists of continuous inputs $\mathbf{x} \in \mathbb{R}^D$ and binary hidden units $\mathbf{h} \in \{0,1\}^M$ with the pdf

$$p(\mathbf{x}, \mathbf{h}; \boldsymbol{\theta}) = \frac{e^{-E(\mathbf{x},\mathbf{h};\boldsymbol{\theta})}}{Z(\boldsymbol{\theta})}, \; Z(\boldsymbol{\theta}) = \sum_h \int_x e^{-E(\mathbf{x},\mathbf{h};\boldsymbol{\theta})} d\mathbf{x}, \tag{5}$$

where the energy function $E(\mathbf{x}, \mathbf{h}; \boldsymbol{\theta})$ is given by

$$E(\mathbf{x}, \mathbf{h}; \boldsymbol{\theta}) = \frac{1}{2}(\mathbf{x} - \mathbf{c})^\top \mathbf{\Lambda}(\mathbf{x} - \mathbf{c}) - \mathbf{h}^\top \mathbf{W}\mathbf{\Lambda}\mathbf{x} - \mathbf{b}^\top \mathbf{h}. \tag{6}$$

Here, $\mathbf{W} \in \mathbb{R}^{H \times D}$ is the matrix of weights connecting the units from the hidden and input layer, $\mathbf{b} \in \mathbb{R}^H$ and $\mathbf{c} \in \mathbb{R}^D$ are the vectors of hidden and input layer biases, and $\mathbf{\Lambda} = \mathrm{dg}(\boldsymbol{\lambda})$ denotes the diagonal precision matrix of the inputs. All these parameters are freely learnable and they are denoted by $\boldsymbol{\theta}$ in Eq.(5) and Eq.(6). We illustrate a Gauss-Bernoulli RBM model in Fig. 2. It is easy to see that the conditional densities are given by

$$p(\mathbf{h}|\mathbf{x}; \boldsymbol{\theta}) = \sigma\left(\mathbf{W}\mathbf{\Lambda}\mathbf{x} + \mathbf{b}\right), \tag{7}$$

$$p(\mathbf{x}|\mathbf{h}; \boldsymbol{\theta}) = \mathcal{N}\left(\mathbf{W}^T\mathbf{h} + \mathbf{c}, \mathbf{\Lambda}^{-1}\right). \tag{8}$$

The marginal density $p(\mathbf{x}; \boldsymbol{\theta})$ of the visible inputs can be also written in the energy-based form

$$p(\mathbf{x}; \boldsymbol{\theta}) = \frac{e^{-\mathcal{F}(\mathbf{x};\boldsymbol{\theta})}}{Z(\boldsymbol{\theta})}, \tag{9}$$

where $Z(\boldsymbol{\theta})$ is called the normalizing constant, and $\mathcal{F}(\mathbf{x};\boldsymbol{\theta})$ is the free energy:

$$Z(\boldsymbol{\theta}) = \int_x e^{-\mathcal{F}(\mathbf{x};\boldsymbol{\theta})} d\mathbf{x}, \quad \mathcal{F}(\mathbf{x};\boldsymbol{\theta}) = \frac{1}{2}(\mathbf{x}-\mathbf{c})^\top \boldsymbol{\Lambda}(\mathbf{x}-\mathbf{c}) - \mathbf{1}_H^\top \boldsymbol{\gamma}, \tag{10}$$

with $\boldsymbol{\gamma} = \log(\mathbf{1}_H + \exp(\mathbf{W}\boldsymbol{\Lambda}\mathbf{x} + \mathbf{b}))$ denoting the element-wise Softplus function. Unlike Eq.(6), the energy function Eq.(10) associated with the marginal $p(\mathbf{x};\boldsymbol{\theta})$ is no longer linear in the free parameters $\boldsymbol{\theta}$.

A frequently encountered Gauss-Bernoulli RBM in the literature is the one associated with the conditional density $p(\mathbf{x}|\mathbf{h};\boldsymbol{\theta}) = \mathcal{N}\left(\mathbf{W}^T\mathbf{h} + \mathbf{c}, \mathbf{I}_D\right)$ and assumes unit variances for the input units. This is a special case of our model Eq.(6) in which we treat the variances of the inputs as learnable parameters, and all results and discussions in this paper can be applied in a straightforward manner to the special case by replacing $\boldsymbol{\Lambda}$ with the identity matrix.

4.2.5 Learning RBM via Contrastive Divergence

The negative log-likelihood of the parameters of the RBM, i.e., the logarithmic loss can be written as

$$\ell(\mathbf{x};\boldsymbol{\theta}) \equiv -\log p(\mathbf{x};\boldsymbol{\theta}) = \mathcal{F}(\mathbf{x};\boldsymbol{\theta}) + \log Z(\boldsymbol{\theta}). \tag{11}$$

The gradient obtains a particularly interesting form:

$$-\nabla_{\boldsymbol{\theta}} \log p(\mathbf{x};\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}}\mathcal{F}(\mathbf{x};\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \log Z(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}}\mathcal{F}(\mathbf{x};\boldsymbol{\theta}) - \mathbb{E}\left[\nabla_{\boldsymbol{\theta}}\mathcal{F}(\mathbf{x};\boldsymbol{\theta})\right], \tag{12}$$

where the expectation in the second term is taken w.r.t. the marginal density of the visible units given in Eq.(9). Therefore, it is difficult to determine the gradient analytically. In order to make the computation tractable, this expectation is estimated using samples from $p(\mathbf{x};\boldsymbol{\theta})$ which can be obtained by running a Markov chain with Gibbs sampling as the intermediate sampling operator. To speed up the sampling process, Hinton [5] showed that it is not necessary to wait for the Markov chain to converge; instead, if the chain is initialized using training examples, reasonable learning performance might be obtained only after k Gibbs steps. In practice, $k = 1$ is commonly used. However, this corresponds to the approximate minimization of the contrastive divergence (CD), which produces biased estimates of the model parameters [6].

We see that an important implication of approximating MLE-based learning through contrastive divergence minimization is the lack of consistency guarantees. Specifically, minimizing the contrastive divergence is not guaranteed to converge to the data-generating parameter $\boldsymbol{\theta}_*$ that minimizes the KL divergence from $p(\mathbf{x};\boldsymbol{\theta})$ to the data-generating pdf $p_*(\mathbf{x})$. The impediment can be traced back to the computation of the gradient of the logarithmic loss and the analytical intractability of the second term in Eq.(12) which appears due to the intractability of the partition function as a normalizing constant in Eq.(5).

4.2.6 Learning RBM via Fisher Divergence

To overcome the issues associated with the lack of consistency guarantees, instead of aiming to minimize the KL divergence through contrastive divergence approximation, we propose to minimize the Fisher divergence from the marginal density of the visible units $p(\mathbf{x};\boldsymbol{\theta})$ to the data-generating distribution $p_*(\mathbf{x})$. To evaluate the Hyvärinen Score Eq.(4) based on Eq.(9), we derived the following result.

Proposition 1 *The Hyvärinen Score for the Gauss-Bernoulli RBM Eq.*(5) *with energy function Eq.*(6) *is given by*

$$s_{\mathrm{F}}(\mathbf{x},\boldsymbol{\theta}) = \frac{1}{2}\|\boldsymbol{\Lambda}(\mathbf{W}^\top\boldsymbol{\sigma} + \mathbf{c} - \mathbf{x})\|^2 + \mathrm{tr}(-\boldsymbol{\Lambda} + \boldsymbol{\Lambda}\mathbf{W}^\top \mathrm{dg}(\boldsymbol{\sigma}')\mathbf{W}\boldsymbol{\Lambda}), \tag{13}$$

with $\boldsymbol{\sigma} = \sigma(\mathbf{W}\boldsymbol{\Lambda}\mathbf{x} + \mathbf{b})$ *and* $\boldsymbol{\sigma}' = \sigma'(\mathbf{W}\boldsymbol{\Lambda}\mathbf{x} + \mathbf{b})$*, where* σ *and* σ' *respectively denote the element-wise Sigmoid operator and the corresponding first derivative.*

Proof 1 *Taking the derivative of the log-density* $\log p(\mathbf{x};\boldsymbol{\theta})$ *w.r.t.* $\mathbf{x}$*, we obtain*

$$\begin{aligned}
\nabla_{\mathbf{x}} \log p(\mathbf{x};\boldsymbol{\theta}) &= -\nabla_{\mathbf{x}}\mathcal{F}(\mathbf{x};\boldsymbol{\theta}) - \nabla_{\mathbf{x}} \log Z(\boldsymbol{\theta}) \\
&= \boldsymbol{\Lambda}(\mathbf{c} - \mathbf{x}) + \sum_{h=1}^{H} \nabla_{\mathbf{x}} \gamma(\mathbf{W}_{h:}\boldsymbol{\Lambda}\mathbf{x} + b_h) \\
&= \boldsymbol{\Lambda}(\mathbf{c} - \mathbf{x}) + \sum_{h=1}^{H} \sigma(\mathbf{W}_{h:}\boldsymbol{\Lambda}\mathbf{x} + b_h)\boldsymbol{\Lambda}\mathbf{W}_{h:}^{\top} \\
&= \boldsymbol{\Lambda}(\mathbf{c} - \mathbf{x}) + \boldsymbol{\Lambda}\mathbf{W}^T\sigma(\mathbf{W}\boldsymbol{\Lambda}\mathbf{x} + \mathbf{b}),
\end{aligned}$$

which gives the first term in Eq.(13)*. To obtain the second term, we first compute the Hessian matrix of the log-density* $\log p(\mathbf{x};\boldsymbol{\theta})$*; we obtain:*

$$\begin{aligned}
\nabla^2_{\mathbf{xx}} \log p(\mathbf{x};\boldsymbol{\theta}) &= \nabla_{\mathbf{x}}(\nabla_{\mathbf{x}} \log p(\mathbf{x};\boldsymbol{\theta})) \\
&= \nabla_{\mathbf{x}}(\boldsymbol{\Lambda}(\mathbf{c} - \mathbf{x}) + \boldsymbol{\Lambda}\mathbf{W}^T\sigma(\mathbf{W}\boldsymbol{\Lambda}\mathbf{x} + \mathbf{b})) \\
&= -\boldsymbol{\Lambda} + \nabla_{\mathbf{x}}\sigma(\mathbf{W}\boldsymbol{\Lambda}\mathbf{x} + \mathbf{b})\mathbf{W}\boldsymbol{\Lambda} \\
&= -\boldsymbol{\Lambda} + \boldsymbol{\Lambda}\mathbf{W}^T\nabla_{\mathbf{u}}\sigma(\mathbf{u})\mathbf{W}\boldsymbol{\Lambda} \\
&= -\boldsymbol{\Lambda} + \boldsymbol{\Lambda}\mathbf{W}^T \operatorname{dg}(\sigma'(\mathbf{u}))\mathbf{W}\boldsymbol{\Lambda},
\end{aligned}$$

where $\mathbf{u} = \mathbf{W}\mathbf{x} + \mathbf{b}$*. Plugging the Hessian into the Laplacian* $\Delta_{\mathbf{x}} \log p(\mathbf{x};\boldsymbol{\theta}) = \operatorname{tr}(\nabla^2_{\mathbf{xx}} \log p(\mathbf{x};\boldsymbol{\theta}))$ *gives the second term in Eq.*(13)*, which completes the proof.*

We observe that unlike the logarithmic loss in Eq.(11), the Hyvärinen Score can be evaluated explicitly in terms of the parameters $\mathbf{c}$, $\mathbf{b}$ and $\mathbf{W}$ of the Gauss-Bernoulli RBM. Moreover, the calculation does not involve the partition function $Z(\boldsymbol{\theta})$. This simplifies the computation of the gradient w.r.t. the parameters of the RBM, which now can be computed by straightforward application of matrix calculus yielding the following closed-form expressions:

$$\begin{aligned}
\nabla_{\mathbf{c}} s_{\mathrm{F}}(\mathbf{x},\boldsymbol{\theta}) &= \boldsymbol{\Lambda}^2(\mathbf{W}^T\boldsymbol{\sigma} + \mathbf{c} - \mathbf{x}), \\
\nabla_{\mathbf{b}} s_{\mathrm{F}}(\mathbf{x},\boldsymbol{\theta}) &= \operatorname{dg}(\boldsymbol{\sigma}')\mathbf{W}\boldsymbol{\Lambda}^2(\mathbf{W}^T\boldsymbol{\sigma} + \mathbf{c} - \mathbf{x}) + \operatorname{dg}(\mathbf{W}\boldsymbol{\Lambda}^2\mathbf{W}^T)\boldsymbol{\sigma}'', \\
\nabla_{\mathbf{W}} s_{\mathrm{F}}(\mathbf{x},\boldsymbol{\theta}) &= \operatorname{dg}(\boldsymbol{\sigma}')\mathbf{W}\boldsymbol{\Lambda}^2(\mathbf{W}^T\boldsymbol{\sigma} + \mathbf{c} - \mathbf{x})\mathbf{x}^T\boldsymbol{\Lambda} + \boldsymbol{\sigma}(\mathbf{W}^T\boldsymbol{\sigma} + \mathbf{c} - \mathbf{x})^{\top}\boldsymbol{\Lambda}^2 \\
&\quad + \operatorname{dg}(\mathbf{W}\boldsymbol{\Lambda}^2\mathbf{W}^{\top})\boldsymbol{\sigma}''\mathbf{x}^T\boldsymbol{\Lambda} + 2\operatorname{dg}(\boldsymbol{\sigma}')\mathbf{W}\boldsymbol{\Lambda}^2, \\
\nabla_{\boldsymbol{\lambda}} s_{\mathrm{F}}(\mathbf{x},\boldsymbol{\theta}) &= \operatorname{dg}(\mathbf{x})\mathbf{W}^{\top}\operatorname{dg}(\boldsymbol{\sigma}')\mathbf{W}\boldsymbol{\Lambda}^2(\mathbf{W}^{\top}\boldsymbol{\sigma} + \mathbf{c} - \mathbf{x}) \\
&\quad + \operatorname{dg}(\mathbf{W}^{\top}\boldsymbol{\sigma} + \mathbf{c} - \mathbf{x})\boldsymbol{\Lambda}(\mathbf{W}^{\top}\boldsymbol{\sigma} + \mathbf{c} - \mathbf{x}) \\
&\quad + 2\operatorname{dg}(\mathbf{W}^T\operatorname{dg}(\boldsymbol{\sigma}')\mathbf{W})\boldsymbol{\lambda} + \mathbf{1}_N \\
&\quad + \operatorname{dg}(\mathbf{x})\mathbf{W}^{\top}\operatorname{dg}(\mathbf{W}\boldsymbol{\Lambda}^2\mathbf{W}^{\top})\boldsymbol{\sigma}''
\end{aligned}$$

with $\boldsymbol{\sigma} = \sigma(\mathbf{W}\boldsymbol{\Lambda}\mathbf{x} + \mathbf{b})$, $\boldsymbol{\sigma}' = \sigma'(\mathbf{W}\boldsymbol{\Lambda}\mathbf{x} + \mathbf{b})$ (as in Proposition 1), whereas $\boldsymbol{\sigma}'' = \sigma^{''}(\mathbf{W}\boldsymbol{\Lambda}\mathbf{x} + \mathbf{b})$; also, recall that $\boldsymbol{\Lambda} = \operatorname{dg}(\boldsymbol{\lambda})$.[1]

It is evident that, as opposed to the gradient of the logarithmic loss Eq.(12), the gradient of the Hyvärinen Score can be computed explicitly w.r.t. the parameters of the Gauss-Bernoulli RBM, producing closed-form expressions that can be used directly used

[1] We omit the detailed derivation of the gradients for brevity, and we note that the gradients can be derived through straightforward application of matrix calculus.

for training the parameters of the RBM. Indeed, let $\hat{\boldsymbol{\theta}}_{\text{F}}^{(n)}$ denote the parameter estimate at each step n. The new parameter estimate can be obtained through the following update rule (repeated until convergence):

$$\hat{\boldsymbol{\theta}}_{\text{F}}^{(n+1)} = \hat{\boldsymbol{\theta}}_{\text{F}}^{(n)} - \eta \frac{1}{B} \sum_{b} \nabla_{\boldsymbol{\theta}} s_{\text{F}} \left(\mathbf{x}_b, \hat{\boldsymbol{\theta}}_{\text{F}}^{(n)} \right),$$

where η is the learning rate, B is the size of the minibatch of randomly chosen data points $\mathbf{x}_b$, and $b = 1, \ldots, B$ at step n.

4.3 Cross-subject Mapping Algorithm

Recall from Section 4.1 that in cross-subject mapping, the goal is to obtain destination feature representation(s) $\mathbf{x}_{\text{D}}$ from the source feature vector(s) $\mathbf{x}_{\text{S}}$. We will elaborate on how to use an RBM and the Gibbs sampler to sample such representations. We first parameterize the generative model $p(\mathbf{x}, \mathbf{h}; \boldsymbol{\theta})$ of all feature vectors $\mathbf{x} = (\mathbf{x_S}, \mathbf{x_D})$ and hidden variables $\mathbf{h}$ using the Gauss-Bernoulli RBM described in Section 4.2.4, see also Fig. 2. After learning the parameters of the Gauss-Bernoulli model we infer $\mathbf{x}_{\text{D}} = (\mathbf{x}_i)_{i \in \mathcal{M}_{\text{D}}}$ from $\mathbf{x}_{\text{S}} = (\mathbf{x}_j)_{j \in \mathcal{M}_{\text{S}}}$ as follows. First, we initialize the vector $\hat{\mathbf{x}}$ by the features of the source subjects $\mathbf{x}_m,\ m \in \mathcal{M}_{\text{S}}$ and random noise (e.g., with standard normal variables). Then:

1. generate $\hat{\mathbf{h}} \sim p(\mathbf{h}|\hat{\mathbf{x}}; \boldsymbol{\theta})$ via Eq.(7);
2. using $\hat{\mathbf{h}}$ generate $\hat{\mathbf{x}} \sim p(\mathbf{x}|\hat{\mathbf{h}}; \boldsymbol{\theta})$ via Eq.(8).

We obtain the final estimate after repeating the above two steps $k \geq 1$ times; Fig. 1 illustrates an example with $k = 3$. This gives the destination feature space representations of the source feature vectors and they can be further processed using algorithms trained on destination data. Alternatively, in the final step, we can skip sampling from $p(\mathbf{x}|\hat{\mathbf{h}}; \boldsymbol{\theta})$ and we can also infer $\mathbf{x}_{\text{D}}$ as $\hat{\mathbf{x}}_{\text{D}} = (\hat{\mathbf{x}}_j)_{j \in \mathcal{M}_{\text{D}}} = \max_{\mathbf{x}_{\text{D}}} p(\mathbf{x}|\mathbf{h}; \boldsymbol{\theta})$. For simplicity and without loss of generality, we have assumed that source and destination subjects $\mathcal{M}_{\text{S}}$ and $\mathcal{M}_{\text{D}}$ satisfy $\mathcal{M} = \mathcal{M}_{\text{S}} \cup \mathcal{M}_{\text{D}}$ and $\mathcal{M}_{\text{S}} \cap \mathcal{M}_{\text{D}} = \emptyset$. This allows us to skip a tedious step in the algorithm and avoid the marginalization of the joint density $p(\mathbf{x}, \mathbf{h}; \boldsymbol{\theta})$ over subjects that are neither sources nor destinations.

5 Evaluation

Next, we present the results from the evaluations. First, in Section 5.1 we describe the experimental protocol and the acquired data. In Section 5.2 we discuss the evaluation methodology, including evaluation scenarios. In Section 5.3 we present the main findings and observations.

5.1 Experiment, Data and Features

5.1.1 Protocol

We study a comprehensive flight motor program for hawk moths. We will describe the experimental protocol and related procedures only briefly here; the interested reader is referred to [16] where the data set was first published for more details. The subjects, i.e., the moths are tethered inside a three-sided box formed by computer monitors displaying the visual stimuli. Each stimulus is represented by sinusoidal gratings with a spatial frequency of 20° per cycle on 3D spheres projected on the monitors. The spheres drift at a constant velocity of 100° per second, corresponding to a temporal frequency of

5 cycles/second. Moreover, the spheres also drift in opposite directions about the three axes of rotation which result in 6 different visual stimuli also known as pitch (up, down), roll (left, right), and yaw (left, right) [16].

The moth responds to each of the 6 discrete stimuli by producing turning effort as assessed with a 6 degree-of-freedom force/torque transducer. The 10 primary muscles that control the flying motion of the moth are wired and enable spike-resolved EMG signals to be recorded during tethered flight. These key muscles include the flight power muscles (dorsolongitudinal (DLM) and dorsoventral (DVM) muscles), as well as the steering muscles, (third axillary (3AX), basalar (BA), and subalar (SA) muscles) on both the left and the right side of the thorax. The EMG recordings are used to extract the timings of the motor unit spikes in each of the muscles that serve as control commands by means of which the nervous system guides the motion of the moth in response to the different visual stimuli; more details can be found in [16, 17]. Taken together, this dataset is unusual in its near-complete recording of all the spikes the animal can use to control its wings and so is an ideal point of convergence to test for the decodability of stimulus conditions.

The objective is to decode the visual stimulus from the comprehensive motor program recordings, i.e., the spike trains. The subject-specific formulation of the problem where the neural decoder (classifier) is both trained and tested on the same subject was analyzed in [16]. Here, we study the performance of the neural decoder in cross-subject settings, where the test data originates from the source subjects whereas the classifier is trained on destination subject data.

The dataset is collected from 9 subjects over 20 seconds of recording sessions for each visual stimulus. Each session is segmented into *wing strokes*, i.e., trials [16]. The typical duration of a wing stroke is between 50 and 70 milliseconds yielding an average number of trials of ≈ 2500 per moth. It should be noted that some moths in the dataset are missing the recordings from some of their muscles (either one or at most two) due to failures in the recording procedure. Nevertheless, as demonstrated in [16], the absence of some (one or two) muscles does not have a significant aspect on decoding performance; in fact, as shown in [16], high decoding accuracy (higher than 90%) can be achieved even with half of the available muscles due to the completeness of the motor program.

5.1.2 Feature Extraction

Before we delve into more details with respect to the cross-subject neural decoder, we briefly describe our methodology for constructing feature representations from spike trains proposed in [16]. Since the spike trains are given by variable-length vectors of spike timings, we consider Gaussian kernels, a strategy commonly used in neuroscience, to interpolate the spike trains. The Gaussian kernel is given by:

$$x(t) = \sum_n \exp\left(-\frac{(t-t_n)^2}{2\sigma^2}\right), \quad 0 \le t \le \tau,$$

where t_n denotes the timing of the n-th spike collected from an arbitrary muscle, trial, and moth, τ denotes the wing stroke cut-off threshold (we only consider spikes that satisfy $t_n \le \tau$), and σ is the Gaussian kernel bandwidth. The goal is to obtain a smooth multivariate time-series representation of the spike trains in which the spike timing information is conveyed by centering one kernel at each spike and summing the kernels; these yields feature vectors of fixed dimension $\tau \cdot \nu_S$ where ν_S is the sampling frequency. For consistency, the muscles whose recordings are missing are filled with zero vectors of the same length as above. We then flatten the interpolated time series across muscles to obtain one large feature vector. Finally, we apply PCA and retain only the first P

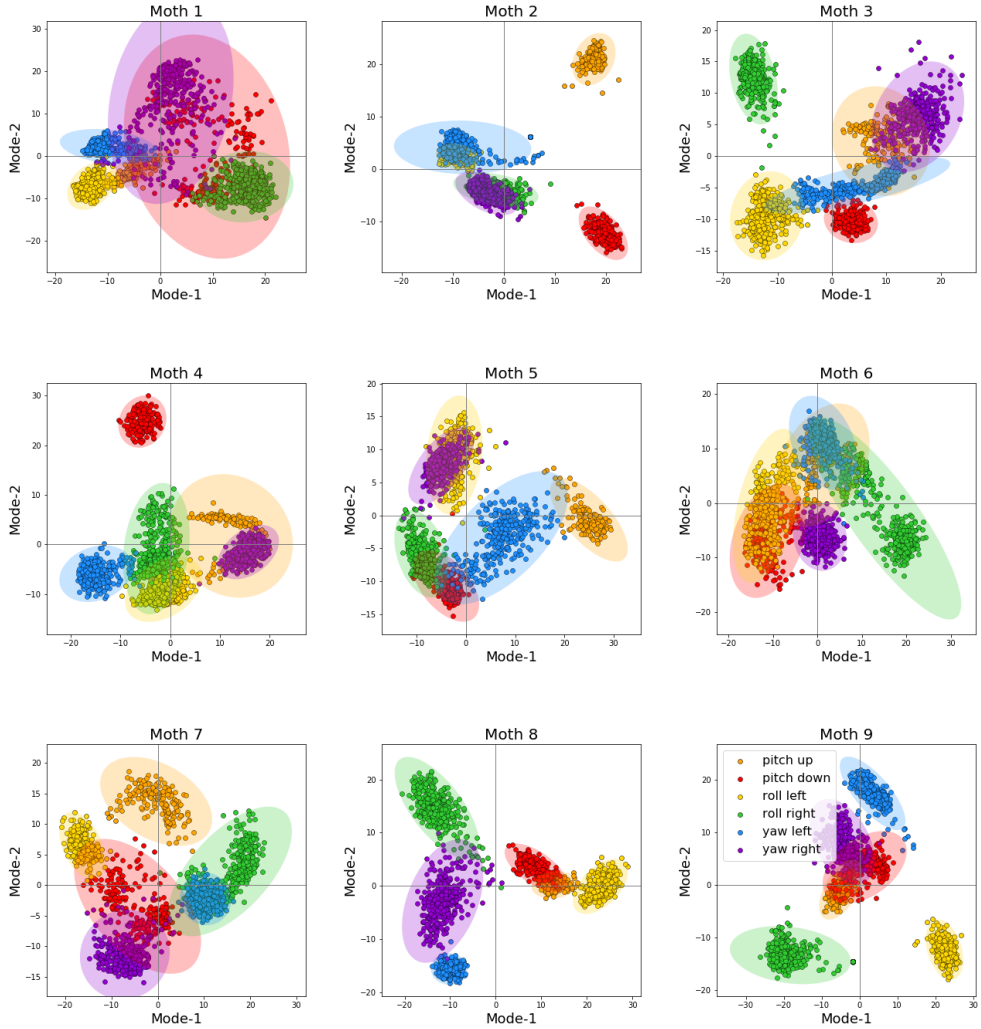


Fig 3. The feature space across the first two principal mode for each moth. For visual clarity, the diagrams also show the confidence ellipse corresponding to one standard deviation for each of the visual conditions.

largest modes; this is our final representation $\mathbf{x}_m$ of the neural activity of subject m with dimension $D_m = P$.

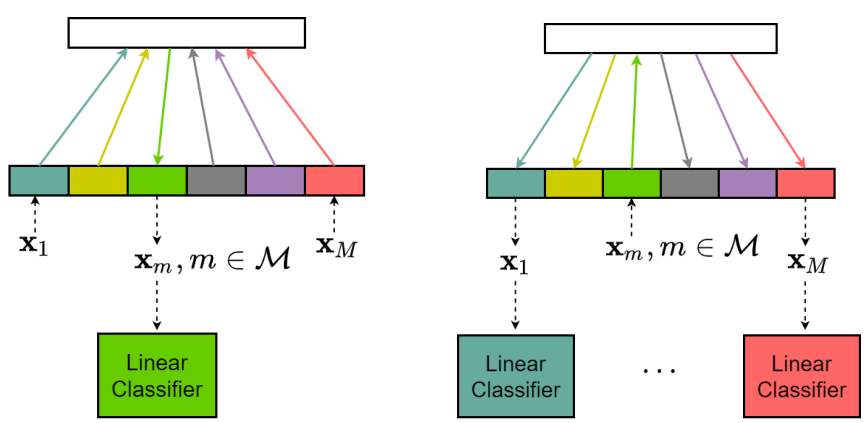
Fig. 3 depicts the feature spaces of each moth and all six visual conditions (tasks) across the first two strongest principal components, i.e., modes after performing PCA. The diagrams also show the confidence ellipses corresponding to one standard deviation for each of the conditions. It can be clearly observed that the data demonstrates strong separability properties even in the first two dimensions of the PCA-based feature space; adding more features (i.e., PCA modes) only increases this separability in the higher-dimensional feature space for each moth as we have observed in our past work [16]; for more details, we advise the interested reader to refer to [16] where the feature extraction procedure was first proposed and its performance thoroughly analyzed.

5.2 Scenarios and Methodology

The moth population set is $\mathcal{M} = \{1, \ldots, 9\}$. We evaluate the performance of the cross-subject neural decoder in the following scenarios:

- *Scenario I* (Fig. 4a). The destination and source sets are $\mathcal{M}_{\mathrm{D}} = \{m\}$ and $\mathcal{M}_{\mathrm{S}} = \mathcal{M} \backslash \{m\}$, respectively. That is, we select a single subject $m \in \mathcal{M}$ as the destination and map the features of *all* remaining subjects $i \neq m$ onto the feature space of the destination subject m. The obtained representation is then decoded using a linear classifier trained on subject m data.

- *Scenario II* (Fig. 4b). The source and destination index sets are $\mathcal{M}_{\mathrm{S}} = \{m\}$ and $\mathcal{M}_{\mathrm{D}} = \mathcal{M} \backslash \{m\}$, respectively. In other words, we select a single subject $m \in \mathcal{M}$ as the source and map the corresponding feature vector onto the feature spaces of *all* remaining subjects $j \neq m$. Similarly, as in scenario I above, the obtained representations of the source feature vector are subsequently decoded using the subject-specific linear classifiers trained on the destination subjects $j \neq m$ individually.



(a) Scenario I **(b)** Scenario II

Fig 4. Evaluation scenarios. (a) One subject is the destination, all other subjects are sources. (b) One subject is the source, all other subjects are destinations.

In both of these scenarios, we evaluate the performance of the destination classifier (trained purely on destination data) on the transferred source features through an RBM model with both standard contrastive divergence minimization and Fisher divergence

minimization; we use RBM-CD and RBM-FD to denote these two RBM models, with CD standing for contrastive divergence (see Section 4.2.5) and FD standing for Fisher divergence (see Section 4.2.6). We compare the performance with two benchmarks:

1. subject-specific neural decoding, when the classifier is trained *and* tested on purely destination data. Equivalently, the data used to train the classifier and the test data respectively come from the same individual subject. In this case, the performance of the neural decoder may be thought of as an upper bound because it represents the best-case scenario where the classifier has access to the most relevant information about feature patterns of the test data.

2. cross-subject neural decoding with *no* transfer, when the source data is directly decoded using the destination neural decoder without transferring the source data into destination feature space. As we discussed in Section 3, in this case, the performance of the neural decoder is usually close to a random guess, since the neural decoder does not take into account the differences between neural activity patterns between the test (source) and the train (destination) data.

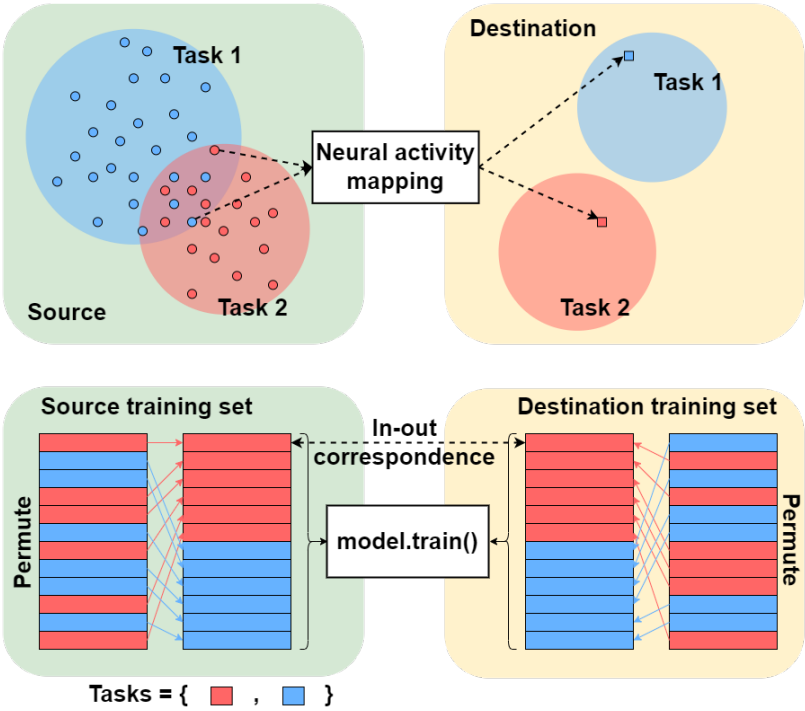


Fig 5. Organization of source and destination training data sets.

The training and testing data for the source and destination data sets in both scenarios are formed by splitting the original data sets of each moth into training and testing data subsets randomly according to $\omega \in (0, 1)$, which denotes the ratio between the number of training samples and the total number of trials. To characterize the performance of the cross-subject neural decoder statistically, the random train/test data splitting procedure is repeated 100 times; the entire model including the RBM is then re-trained using the new training data, and the test performance is recorded.

In cross-subject mapping, we are aiming to find a destination space feature representation of the source feature vector; the mapping should be consistent with the task that the source features are encoding. In other words, the mapping should be such that the transferred representation of the source feature vector should be in the region

of the destination feature space that corresponds to the task that the source is performing, as illustrated in the top row in Fig. 5. In essence, when sampling from the joint pdf $p(\mathbf{x}, \mathbf{h})$, we wish to sample from a stimulus-dependent distribution and this should be guided by the task information the source feature vector is encoding. Hence, the trials in the training data set should be organized such that there is an input-output correspondence with respect to the stimulus. As we are unable to establish correspondence between individual training trials, we assign the correspondences at random. The procedure is schematically depicted in the bottom row in Fig. 5. Namely, for each source feature vector from the training set, we randomly choose a destination feature vector from the same stimulus class and declare the pair to be an input-output pair.

5.3 Results

We used the following values for the free parameters:

Table 1. Experimental Parameters.

Parameter	**Value**
Data split ratio (ω)	0.5
Wing stroke cut-off (τ)	60 ms
Kernel bandwidth (σ)	2.5 ms
Number of principal modes ($D_m = P$)	10
Number of units in hidden layer (H)	15
Optimizer for training RBM model	Adam [21]
Learning rate	0.005
Minibatch size	150
Training epochs	200

In both scenarios, we use the Linear Discriminant Analysis (LDA) for classification which we also used in [16] and has shown to perform exceptionally high decoding accuracy.

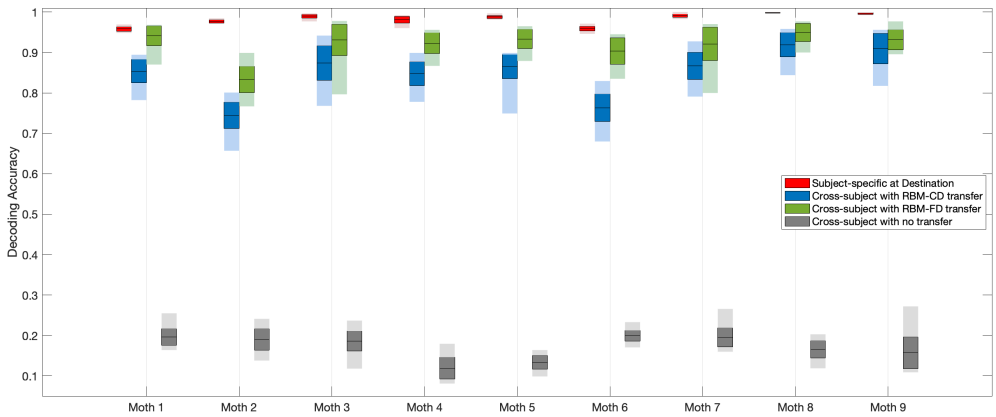


Fig 6. Performance of the cross-subject neural decoder in Scenario I (see Fig. 4a and Section 5.2 for details).

The results are shown in Figs. 6 and 7. We observe that in both scenarios the performance of the cross-subject neural decoder is bounded between the performances of the two benchmarks, with the performance of the subject-specific neural decoder being the upper bound and the performance of the cross-subject neural decoding with

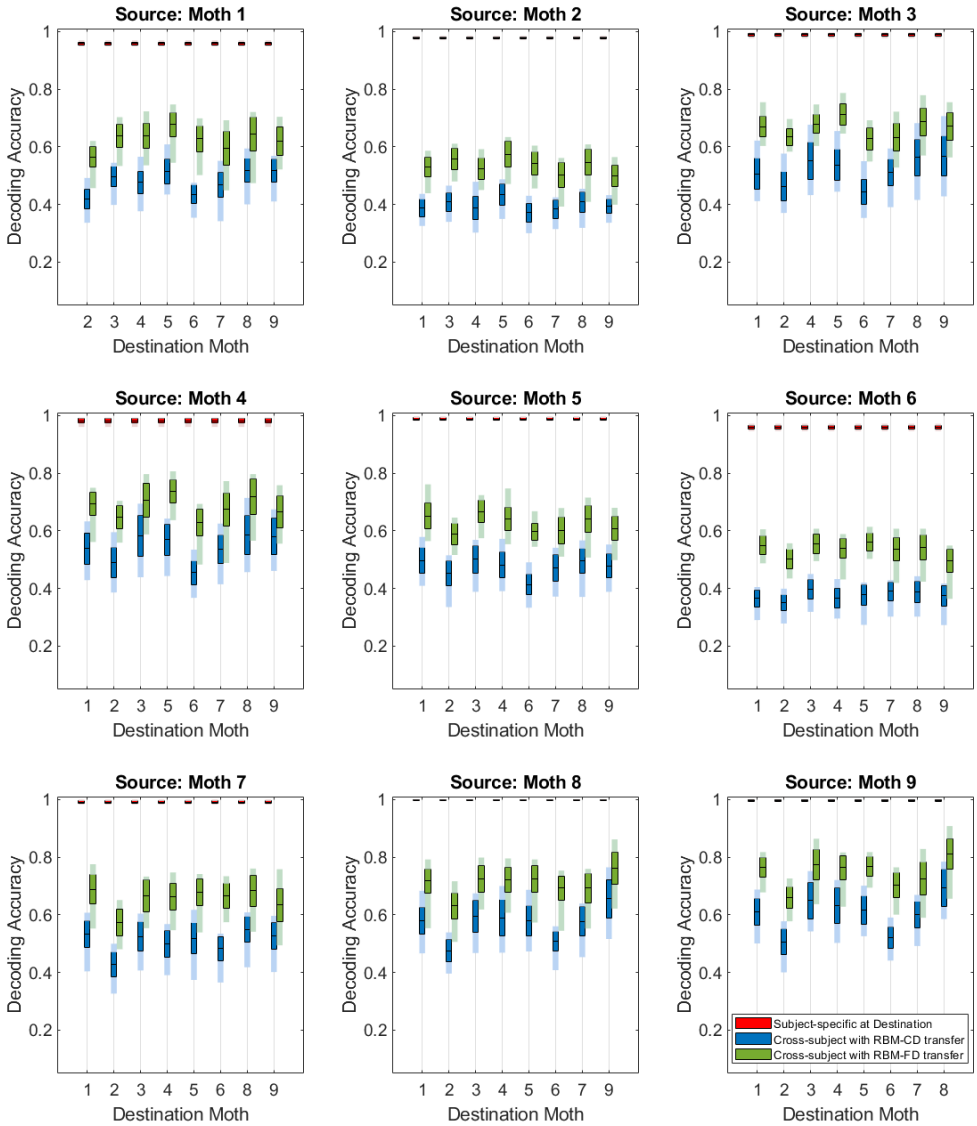


Fig 7. Performance of the cross-subject neural decoder in Scenario II (see Fig. 4b and Section 5.2).

no transfer being the lower bound. The performance of the cross-subject neural decoder being upper-bounded by the performance of the subject-specific neural decoder is intuitively expected. Note that in Fig. 6 the lower bound varies around 0.16, which corresponds to random choice decoding in our case (as there are 6 stimuli in the experiment, see Section 5.1). The performance of the cross-subject neural decoder is similar in Scenario II but we omit to show it in Fig. 7 to avoid clutter. We conclude that the performance of the cross-subject neural decoder without transferring the source features to the destination is very poor, and produces a poor lower bound. The poor decoding performance of the second benchmark when the source test data is directly decoded using a decoder trained on the destination data, without any prior transformation/adaptation of the source data to the destination feature space is also intuitively expected. This can be most easily seen by inspecting Fig. 3 which depicts the feature spaces across the first two modes for each moth. Even though each moth individually exhibits a high degree of class separability (which ultimately results in very reliable subject-specific decoding performance as demonstrated in Fig. 6), there is very little alignment between the geometric distribution of the classes/tasks (i.e., visual conditions) in the space spanned by the first two modes across different moths. In fact, the task-specific representations seem to occupy arbitrary segments of the feature space across the first two dimensions and no discernible pattern can be directly observed; adding even more modes/features which are required to achieve high subject-specific separability, only exacerbate the differences of the feature spaces across moths. As a result, directly decoding any source moth using a neural decoder trained over a different, destination moth results in poor performance as reported in Fig. 6.

The role of the RBM is to serve as a non-linear mapping function that takes source features and adapts them to the destination feature space where the decoder was trained, and this effect can be observed in Fig. 8 which shows the distribution of the source test points in Scenario I after applying FD-RBM cross-subject transfer with trained RBM model to the corresponding destination feature space. Note that the test points are the features coming from a diverse set of sources, namely all remaining (eight) moths; hence, the illustrative result shown in Fig. 8, in addition to the decoding results presented in Fig.6, clearly demonstrate that the FD-RBM model has successfully learned a non-linear transformation that takes a task-specific feature representation from an arbitrary source and maps it into the adequate task-specific region of the destination feature space.

By comparing the results in Fig. 6 with the results in Fig. 7, we also observe that the performance of the neural decoder in Scenario I outperforms the neural decoders in Scenario II. This is an intuitively expected result, as in Scenario II, the size of the joint destination feature vector $\mathbf{x}_{\mathrm{D}}$ is $M-1$ times larger than the source feature vector $\mathbf{x}_{\mathrm{S}}$; that is, in Scenario II we are jointly obtaining the representation of a single source in 8 different destination feature spaces. The opposite reasoning applies to Scenario I. Hence, the drop in the performance from Fig. 6 to Fig. 7 is expected. Furthermore, for a given population of subjects indexed in $\mathcal{M}$, we can view these two scenarios as the two extreme cases that put the upper (Scenario I) and lower (Scenario II) bounds on the performance.

We also observe that the performance of the cross-subject neural decoder with RBM-FD transfer outperforms the performance of the same cross-subject decoder with RBM-CD transfer. This is an interesting finding, further indicating that in the case of RBMs, the training based on Fisher divergence minimization yields better results in comparison with the more conventional approach based on Maximum Likelihood. This result is also consistent with our findings on popular public datasets such as the MNIST where we used RBM for applications such as compression and reconstruction and where we observed that an RBM trained via Fisher divergence minimization yields

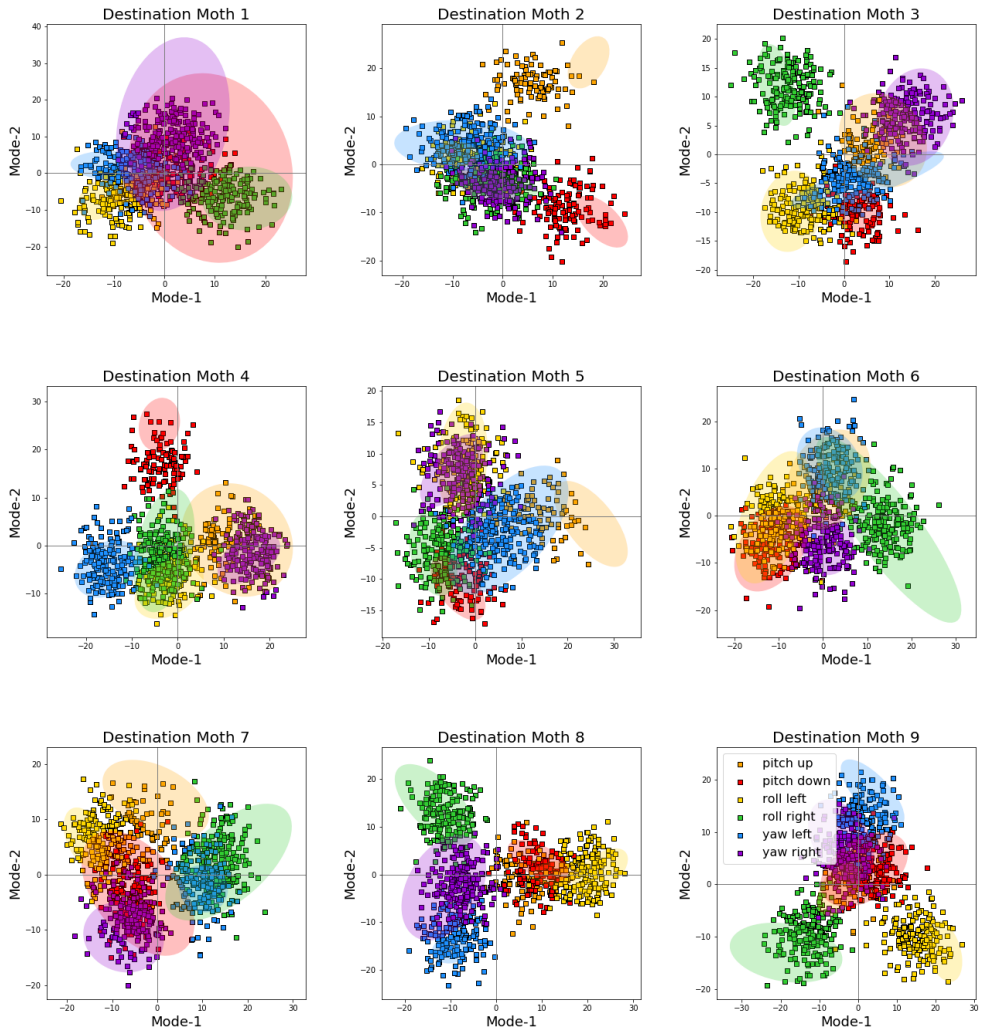


Fig 8. Distribution of source test points (full squares) in the feature spaces of the destination moths (represented by the transparent confidence ellipses from Fig. 3) after mapping them using trained FD-RBM model. The mapping corresponds to Scenario I and, as in Fig. 3, only the first two modes of the feature space are shown.

higher-quality image reconstructions. In addition to the reliability improvement, we note that training an RBM-FD is less computationally demanding as opposed to RBM-CD which requires Gibbs sampling even during training to obtain estimates for the gradients. However, it should be noted that while the RBM-FD model in general tends to outperform the RBM-CD model, the behavior is ultimately determined by the values of the free parameters, and in the case of the parameters we have selected (listed in the beginning of this section), the above observations are valid.

6 Conclusion and Future Work

The design of reliable, robust, and low-cost solutions for cross-subject mapping is a challenging problem in neuroscience. In this paper, we proposed a general framework for learning the joint distribution of source and destination feature representations across a set of subjects using the undirected graphical model and RBM which we evaluated on a neural decoding task and experimental data collected from nine hawk moths during a comprehensive motor program where the moths are subject to a total of six visual stimuli. We also considered an alternative training method for the RBM that minimizes the Fisher divergence and allows the gradient to be computed in closed form, alleviating the need for Gibbs sampling during training. The results verified the viability of the solution. These approaches show promise in generalizing features of complex neural datasets across individuals, tuning neural interfaces to subject-specific features, and leveraging data across multiple subjects when experiments are limited in time or completeness.

Several extensions of the presented approach are possible and are currently part of our ongoing work. First, the derivation of the conditional distribution of the destination features given the source features from the joint distribution represented by an RBM is a direction worthwhile pursuing as it relates to the cross-subject mapping problem and would avoid the issues related to noise-like initialization in the approach presented in this paper. Second, learning subject-invariant RBM models that can also predict feature representations of unseen source subjects is a key direction that should be pursued since it is directly related to the generalization capability of the approach. Finally, we are also investigating the application of the model to other neural signal modalities, including non-invasive modalities such as EEG signals.

References

1. Rao RPN. Brain-Computer Interfacing: An Introduction. Cambridge University Press; 2013.

2. Jayaram V, Alamgir M, Altun Y, Scholkopf B, Grosse-Wentrup M. Transfer Learning in Brain-Computer Interfaces. IEEE Comput Intell Mag. 2016;11(1):20–31. doi:10.1109/MCI.2015.2501545.

3. Angjelichinoski M, Choi J, Banerjee T, Pesaran B, Tarokh V. Cross-subject decoding of eye movement goals from local field potentials. J Neural Eng. 2020;17(1):016067.

4. Angjelichinoski M, Pesaran B, Tarokh V. Deep Cross-Subject Mapping of Neural Activity. ArXiv. 2020;abs/2007.06407.

5. Hinton GE. Training products of experts by minimizing contrastive divergence. Neural Comput. 2002;14(8):1771–1800.

6. Carreira-Perpinan MA, Hinton G. On contrastive divergence learning. In: International workshop on Artificial Intelligence and Statistics (AISTATS). PMLR; 2005. p. 33–40.

7. Plis SM, Hjelm DR, Salakhutdinov R, Allen EA, Bockholt HJ, Long JD, et al. Deep learning for neuroimaging: a validation study. Front Neurosci. 2014;8:229.

8. Kim HC, Jang H, Lee JH. Test–retest reliability of spatial patterns from resting-state functional MRI using the restricted Boltzmann machine and hierarchically organized spatial patterns from the deep belief network. J Neurosci Methods. 2020;330:108451.

9. Li F, Tran L, Thung KH, Ji S, Shen D, Li J. A robust deep model for improved classification of AD/MCI patients. IEEE J Biomed Health Inform. 2015;19(5):1610–1616.

10. Hajinoroozi M, Mao Z, Jung TP, Lin CT, Huang Y. EEG-based prediction of driver's cognitive performance by deep convolutional neural network. Signal Process: Image Commun. 2016;47:549–555.

11. Chai R, Ling SH, San PP, Naik GR, Nguyen TN, Tran Y, et al. Improving EEG-based driver fatigue classification using sparse-deep belief networks. Front Neurosci. 2017;11:103.

12. Wu Y, Ji Q. Constrained deep transfer feature learning and its applications. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016. p. 5101–5109.

13. Farahani HS, Fatehi A, Shoorehdeli MA. Between-domain instance transition via the process of Gibbs sampling in RBM. arXiv preprint arXiv:200614538. 2020;.

14. Wei B, Pal C. Heterogeneous transfer learning with rbms. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). vol. 25; 2011. p. 531–536.

15. Hyvärinen A. Estimation of non-normalized statistical models by score matching. J Mach Learn Res. 2005;6(Apr):695–709.

16. Putney J, Angjelichinoski M, Ravier R, Ferrari S, Tarokh V, Sponberg S. Consistent coordination patterns provide near perfect behavior decoding in a comprehensive motor program for insect flight. bioRxiv. 2021;.

17. Putney J, Conn R, Sponberg S. Precise timing is ubiquitous, consistent, and coordinated across a comprehensive, spike-resolved flight motor program. Proceedings of the National Academy of Sciences. 2019;116(52):26951–26960.

18. Sohn K, Lee H, Yan X. Learning structured output representation using deep conditional generative models. Advances in Neural Information Processing Systems (NIPS). 2015;28.

19. White H. Maximum likelihood estimation of misspecified models. Econometrica. 1982; p. 1–25.

20. Hyvärinen A. Connections Between Score Matching, Contrastive Divergence, and Pseudolikelihood for Continuous-Valued Variables. IEEE Trans Neural Netw. 2007;18:1529–1531.

21. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization; 2014.