Cross-subject Mapping of Neural Activity with Restricted Boltzmann Machines

Haoming Yang[1,*,**], 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

** Equal Contribution

† Marko Angjelichinoski and Suya Wu contributed to this work when they were 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.

Abstract Subject-to-subject variability is a common challenge in generalizing neural data models across subjects. While many methods exist that map one subject to another, it remains challenging to combine many subjects in a computationally efficient manner, especially with features that are highly non-linear such as when considering populations of spiking neurons or motor units. Our objective is to transfer data from one or more target subjects to the data space of one or more source subject(s) such that the neural decoder of the source subject can directly decode the target data when the source(s) is not available during test time. We propose to use the Gaussian-Bernoulli Restricted Boltzmann Machine (RBM); once trained

over the entire set of subjects, the RBM allows the mapping of target features on source 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, in contrast to the more traditional contrastive divergence. We apply our methods to decode turning behaviors from a comprehensive spike-resolved motor program – neuromuscular recordings of spike trains from the ten muscles that control wing motion in an agile flying Manduca sexta. The dataset consists of the comprehensive motor program recorded from nine subjects driven by six discrete visual stimuli. The evaluations show that the target features can be decoded using the source classifier with an accuracy of up to 95% when mapped using an RBM trained by Fisher divergence.

Significant Statement 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 target subject signal in the feature space of the source subject. Our proposed RBM architectures achieve highly flexible and accurate cross-subject mapping with few assumptions. Our Fisher RBM improved the previous state of the art method by 300%. Our methods 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.

1 Introduction

Combining data across individuals or subjects is a ubiquitous need in analyzing neuroscience experiments. Individual variation can produce subject specific noise and different subjects may utilize different strategies to represent the same sensory input or execute the same motor task. Yet, implementing reliable *cross-subject* algorithms in neuroscience is a notoriously challenging problem. Another 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 (Dabagia et al., 2023; Jayaram et al., 2016; Rao, 2013). As a result, the algorithms trained and optimized on data collected from a given subject, have long failed to perform reliably when directly applied to other subjects (Torres-Oviedo and Ting, 2010). Even modern neural decoders 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 (Angjelichinoski, Choi, Banerjee, Pesaran and Tarokh, 2020).

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*. Transfer learning methods could have a significant impact on neuroscience and behavior: for example, a reliable cross-neuron population mapping in the latent domain of neural activities can stabilize and improve brain-computer interface (Degenhart et al., 2020; Karpowicz et al., 2022; Pandarinath et al., 2018). In the context of the cross-subject problem, various approaches have already been considered (Jayaram et al., 2016). One direction to map between subjects is domain adaptation methods that map the target data onto the source feature spaces (Angjelichinoski, Choi, Banerjee, Pesaran and Tarokh, 2020; Harvey et al., 2024). Generative modeling is another promising approach to cross-subject mapping: one can apply a directed graph such as the conditional variational autoencoder (cVAE) to generate the target data onto the feature space of the source data (Angjelichinoski,

Pesaran and Tarokh, 2020), where the learning model for the downstream task is trained These approaches, however, require a separate directed graphical model to be trained each time a new downstream task and/or new source subject is considered.

A more general approach to cross-subject mapping should be independent of downstream tasks, which are generally tackled through distribution alignment (Dyer et al., 2017; Lee et al., 2019). These works aim to find a mapping that aligns the distribution of target features to the distribution of source features. These methods generally require strong assumptions, which ensures their provably robust, accurate result. For example, Procrustes alignment-based methods require both source and target to exist during test time (Degenhart et al., 2020; Haxby et al., 2011). Another example, the Hierarchical Wasserstein Alignment (HiWA) assumes the alignment can be solved through a linear transformation (Lee et al., 2019), and requires cluster structure in the dataset with cluster labels to solve the optimal transport optimization. These alignment-based methods perform well in many decoding problems where aligning the data of the same subject across time and distributional shifts is needed (Dyer et al., 2017; Karpowicz et al., 2022; Lee et al., 2019). However, in a more complex multi-subject setting where relationships across subjects could be highly non-linear, the strong assumptions of alignment methods restrict their flexibility. These alignment methods could also be inefficient with high-dimensional datasets, as the matrix operations become more computationally intensive as dimensionality grows.

While alignment methods are widely applied in neuroscience research, their restrictions call for reliable non-linear cross-subject mapping methods that are more flexible against complex real world data. One instrumental approach is the Deep Learning method that solves the cross-subject mapping problem through non-linear neural networks. The developed methods applicable to cross-subject mapping usually lie in the field of Transfer Learning (Soleimani and Nazerfard, 2021); however, these methods, especially those based on directed graphs with encoding and decoding processes remain inefficient in the multi-subject scenario (Angjelichinoski, Choi, Banerjee, Pesaran and Tarokh, 2020; Angjelichinoski, Pesaran and

Tarokh, 2020). For example, to transfer one subject to 3 different subjects, one would need to train 3 different transfer learning models to achieve a single-direction cross-subject mapping, and the number of models scales exponentially as the number of subjects grows.

In this paper, we propose an efficient, non-linear algorithm that uses undirected graphs to generate samples for cross-subject mapping. Specifically, we leverage the non-linear generative algorithm known as the Restricted Boltzmann Machine (RBM) (Carreira-Perpinan and Hinton, 2005; Hinton, 2002), which allows us to bring the flexible and assumption-less deep learning perspectives and the versatilities of the alignment-based methods together to construct an efficient cross-subject mapping method independent of its downstream decoding tasks RBM is a popular generative model that has had notable success in representation learning with applications in a wide variety of tasks in neuroscience including neuroimaging, classification of temporal events, and EEG analysis (Chai et al., 2017; Hajinoroozi et al., 2016; Kim et al., 2020; Li et al., 2015; Plis et al., 2014). RBMs have also found success in transfer learning, but mainly in computer vision applications (Farahani et al., 2020; Wei and Pal, 2011; Wu and Ji, 2016). The applications of RBM in neuroscience and transfer learning motivate us to adapt RBM for cross-subject mapping, which can serve as a general method with few assumptions on the dataset.

In the neuroscience context, we tackle the general problem of finding a common latent representation of high dimensional spiking data from many neurons across many individual subjects and mapping new subjects to this latent domain. Conceptually what we wish to do is map population recordings from different subjects onto a common latent space and including subjects on which the model has not be previously trained, and then decode from this space. While many other methods have aligned one target to one subject, few have considered the more general problem of combining representations across many subjects even though this is common to many experiments (Hajinoroozi et al., 2016; Kim et al., 2020). Only a few of the current methods consider highly non-linear representations such as spikes but only apply them in a naive environment where only two subjects exist and source and targets both exist

during test time (Degenhart et al., 2020; Lee et al., 2019). Here we utilize RBMs to learn a common decoder either from one source subject applied to many test subjects or from a combination of many source subjects applied to a single target. Therefore, during testing, the neural decoder trained on source data can be directly applied to decode the test data of the target subjects because the cross-subject mapping process is independent of downstream tasks. This particular feature of the undirected graphs allows efficient transfer between all subjects with one model, in comparison of the multi-model method induced by traditional subject mapping

One limitation of RBMs is that they often are computationally expensive because contrastive divergence minimization does not have a closed form and requires Gibbs sampling. Here, we also extend the conventional contrastive divergence training of RBM (which is equivalent to maximum likelihood, we refer more details to (Hinton, 2002)) by proposing an alternative training based on Fisher divergence minimization (Hyvärinen, 2005). The Fisher divergence minimization allows the gradient of the RBM to be computed in closed form, fostering an even more efficient implementation that does not require iterative Gibbs sampling during training.

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 (Putney et al., 2021). 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 (Putney et al., 2019). 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.

In the context of these specific data, the challenge is to enable the decoding of a new target subject moth's motor program into the motor outcome (turn direction) based on

a set of latent space shared in common with the other subjects (i.e. individual moths). Within a single subject, we can already nearly perfectly decode behavior as expected from a comprehensive motor program recording. However, the latent structures within population recordings across the different moths are different, so naive decoding does not generalize. Using our RBM with Fisher divergence we show that we can learn a generalized latent domain from the population of spiking units. We compare the decoding accuracy of this method to other common transfer learning methods most notably HiWA and RBMs with more traditional contrastive divergence. 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; these results thoroughly improve from previous state-of-the-art distribution alignment-based methods like HiWA.

2 Material and Methods

We divide this section into three parts. In Section 2.1 we present the statistical formulation of the problem of cross-subject mapping as a problem of learning joint distribution between target and source feature vectors. Next, in Section 2.2, we discuss RBMs and present both the contrastive divergence and Fisher divergence training methods. Section 2.3 presents a simple cross-subject mapping approach that uses Gibbs sampling to draw samples from the joint distribution of the target and source features. Our code to replicate our study is available in the extended data.

2.1 Problem Statement

The objective of cross-subject learning is to map target subject features to the feature space of source subjects so that the motor intentions of the target subject can be decoded by the neural decoder that is learned on source subjects. In other words, we aim to obtain

the appropriate source space representation of the tasks that the target subjects perform. Technically, the problem boils down to finding a function that maps the feature vectors. This mapping function can be assumed to be purely deterministic. However, in this paper, we adopt a probabilistic approach, which generates task-specific features in source feature space. We outline details below.

Let $\mathcal{M}_{\mathrm{D}}$ and $\mathcal{M}_{\mathrm{S}}$ denote the index sets of the target and source subjects, respectively. Source subject(s) is defined as the subject(s) where the neural decoder is available during test time, and target subject(s) is the subject that we wish to decode using the source decoder Further, we let $\mathcal{M}$ denote the joint set of all subjects, namely $\mathcal{M} = \mathcal{M}_{\mathrm{D}} \cup \mathcal{M}_{\mathrm{S}}$. For simplicity, we assume that the subsets $\mathcal{M}_{\mathrm{D}}$ and $\mathcal{M}_{\mathrm{S}}$ are disjoint, namely $\mathcal{M}_{\mathrm{D}} \cap \mathcal{M}_{\mathrm{S}} = \emptyset$. 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 $\mathbf{x}_{\mathrm{D}} = (\mathbf{x}_i)_{i \in \mathcal{M}_{\mathrm{D}}}$ and $\mathbf{x}_{\mathrm{S}} = (\mathbf{x}_j)_{j \in \mathcal{M}_{\mathrm{S}}}$ to respectively denote the joint feature vectors of the target and source subjects. Note that $\mathbf{x}_{\mathrm{D}}$ and $\mathbf{x}_{\mathrm{S}}$ are vectors with dimensions $D_{\mathrm{D}} = \sum_{i \in \mathcal{M}_{\mathrm{D}}} D_i$ and $D_{\mathrm{S}} = \sum_{i \in \mathcal{M}_{\mathrm{S}}} D_i$, respectively. Finally, we use $\mathbf{x}$ to denote the joint vector of features of *all* subjects in the set $\mathcal{M}$; that is $\mathbf{x} = (\mathbf{x}_{\mathrm{D}}, \mathbf{x}_{\mathrm{S}}) = (\mathbf{x}_m)_{m \in \mathcal{M}}$, 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 source subjects given the feature vector of target subjects. One option is to directly parameterize the probability density function $p(\mathbf{x}_{\mathrm{S}}|\mathbf{x}_{\mathrm{D}})$ of this conditional distribution. Another approach is to first learn the distribution $p(\mathbf{x}_{\mathrm{D}})$ and the joint distribution $p(\mathbf{x}_{\mathrm{S}}, \mathbf{x}_{\mathrm{D}})$ of all feature vectors across the entire population of subjects in $\mathcal{M}$, and then to obtain the conditional distribution $p(\mathbf{x}_{\mathrm{S}}|\mathbf{x}_{\mathrm{D}})$ 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 target subject in the feature space of the source subject. Therefore, 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 $p(\mathbf{h}|\mathbf{x};\theta)$. Generative algorithm for cross subject mapping in previous works, generally applies directed graphs such as cVAE (Angjelichinoski, Pesaran and Tarokh, 2020; Sohn et al., 2015), which map the target feature onto the feature space of the source subjects by the decoder of the autoencoder. However, directed graphs are not flexible to adopt new source 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, an RBM is the first generative model we will tackle. In our application, the source data is unknown during test time, and our following construction of the RBM allows the mapping of multiple subjects from targets to sources with only one trained model.

This RBM is constructed such that a joint distribution of all features is learned. Specifically, we consider a generative model $p(\mathbf{x})$ of the concatenated feature vector $\mathbf{x} = (\mathbf{x}_{\mathrm{D}}, \mathbf{x}_{\mathrm{S}})$. We first learn the joint distribution $p(\mathbf{x}) = p(\mathbf{x}_{\mathrm{D}}, \mathbf{x}_{\mathrm{S}})$. Once the joint distribution is learned by the RBM, 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 $\mathbf{x}^{(0)} = (\mathbf{x}_{\mathrm{D}}^{(0)}, \mathbf{x}_{\mathrm{S}}^{(0)})$. Here, $\mathbf{x}_{\mathrm{D}}^{(0)} = \mathbf{x}_{\mathrm{D}}$ is given, and $\mathbf{x}_{\mathrm{S}}^{(0)}$ 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 $\mathbf{x}^{(k)} = (\mathbf{x}_{\mathrm{D}}^{(k)}, \mathbf{x}_{\mathrm{S}}^{(k)})$ where the $\mathbf{x}_{\mathrm{S}}^{(k)}$ is the feature representation of $\mathbf{x}_{\mathrm{D}}$ in the feature space of source subjects. The hidden layer $\mathbf{h}$ is designed to bridge target and source feature vectors. Note that this architecture setup allows us to only train one RBM that covers the need for any multi-subject mapping during test time. This flexibility follows from learning the joint distribution during training, allowing us to efficiently apply Bayes rule during sampling. During the training of the RBM, there is no distinction between $\mathbf{x}_{\mathrm{D}}$ and $\mathbf{x}_{\mathrm{S}}$ as all subjects are available. During test time however, only $\mathcal{M}_{\mathrm{S}}$'s neural decoder and only $\mathcal{M}_{\mathrm{D}}$'s data is available. With reliable conditional sampling from the learned conditional distribution, we can now map new data from target to source

without any source data available during testing, and apply the sampled data to the source's neural decoder. We illustrate this generative framework in Fig. 1.

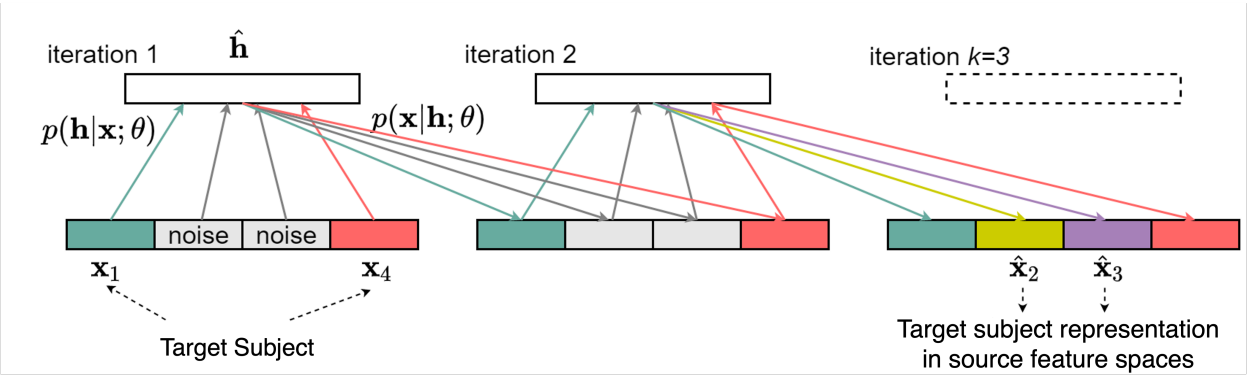


Fig 1: A generative framework for the cross-subject mapping during testing. We assume that both the source and target data are available during training. Neural decoders, which in our case are linear classifiers (see details in section 3), are trained for the source subjects. Only the target subjects are available during testing, and we replace the source representation with randomly sampled standard Gaussian noise. The trained RBM, after k iterations, will sample from the conditional distribution $p(\mathbf{x}_{\mathrm{S}}|\mathbf{x}_{\mathrm{D}})$, which can be seen as the target representation in source space. In this figure $k = 3$.

2.2 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 (Carreira-Perpinan and Hinton, 2005; Hinton, 2002). We also consider an alternative training technique that minimizes the Fisher divergence (Hyvärinen, 2005). 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.

2.2.1 Notation

We adopt the following notation conventions. Recall that in the context of the cross-subject mapping problem outlined in Section 2.1, 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 2.2.4).

2.2.2 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 (White, 1982) 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.

2.2.3 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 (Hyvärinen, 2005)

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

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 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 (Hyvärinen, 2005) 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 infinitesimal step size (Hyvärinen, 2007). 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

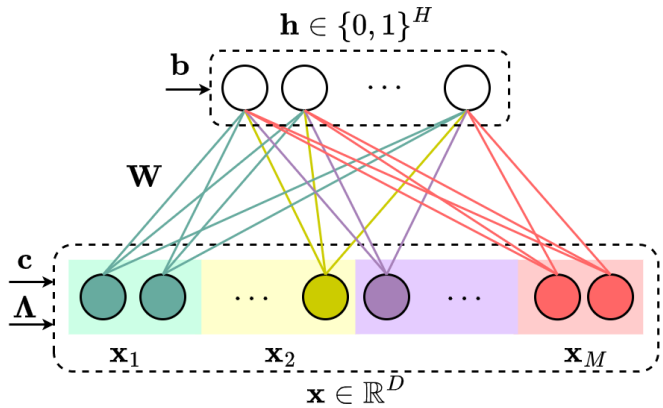


Fig 2: A bipartite undirected graphical model: Gauss-Bernoulli restricted Boltzmann machine with continuous input and binary hidden units.

different.

2.2.4 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

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

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

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 $\boldsymbol{\gamma} = \log(\mathbf{1}_H + \exp\left(\mathbf{W}\boldsymbol{\Lambda}\mathbf{x} + \mathbf{b}\right))$ 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.

2.2.5 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 (2002) 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 during training (Hinton, 2002). However, this corresponds to the approximate minimization of the contrastive divergence (CD), which produces biased estimates of the model parameters (Carreira-Perpinan and Hinton, 2005). While obtaining an expected value may require a large k for a contrastive divergence to converge (van der Plas et al., 2023), in our application, we need to sample from the conditional distribution, therefore we chose $k = 1$ during test time where we sample from the distribution of source conditioned on target

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).

2.2.6 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:

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

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

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

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.

2.3 Cross-subject Mapping Algorithm

Recall from Section 2.1 that in cross-subject mapping, the goal is to obtain source feature representation(s) $\mathbf{x}_{\mathrm{S}}$ from the target feature vector(s) $\mathbf{x}_{\mathrm{D}}$. 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 $\mathbf{x} = (\mathbf{x}_{\mathrm{S}}, \mathbf{x}_{\mathrm{D}})$ and hidden variables $\mathbf{h}$ using the Gauss-Bernoulli RBM described in Section 2.2.4, see also Fig. 2. After learning the parameters of the Gauss-Bernoulli model we infer $\mathbf{x}_{\mathrm{S}} = (\mathbf{x}_i)_{i \in \mathcal{M}_{\mathrm{S}}}$ from $\mathbf{x}_{\mathrm{D}} = (\mathbf{x}_j)_{j \in \mathcal{M}_{\mathrm{D}}}$ as follows. First, we initialize the vector $\hat{\mathbf{x}}$ by the features of the target subjects $\mathbf{x}_m,\ m \in \mathcal{M}_{\mathrm{D}}$ 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

[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.

an example with $k = 3$. This gives the source feature space representations of the target feature vectors and they can be further processed using algorithms trained on source data. Alternatively, in the final step, we can skip sampling from $p(\mathbf{x}|\hat{\mathbf{h}};\boldsymbol{\theta})$ and we can also infer $\mathbf{x}_\mathrm{S}$ as $\hat{\mathbf{x}}_\mathrm{S} = (\hat{\mathbf{x}}_j)_{j\in\mathcal{M}_\mathrm{S}} = \max_{\mathbf{x}_\mathrm{S}} p(\mathbf{x}|\mathbf{h};\boldsymbol{\theta})$. For simplicity and without loss of generality, we have assumed that target and source subjects $\mathcal{M}_\mathrm{D}$ and $\mathcal{M}_\mathrm{S}$ satisfy $\mathcal{M} = \mathcal{M}_\mathrm{D} \cup \mathcal{M}_\mathrm{S}$ and $\mathcal{M}_\mathrm{D} \cap \mathcal{M}_\mathrm{S} = \emptyset$. 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 targets nor sources.

3 Evaluation and Results

Next, we present the results from the evaluations. First, in Section 3.1 we describe the experimental protocol and the acquired data. In Section 3.2 we discuss the evaluation methodology, including evaluation scenarios. In Section 3.3 we present the main findings and observations.

3.1 Experiment, Data and Features

3.1.1 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 (Putney et al., 2021) 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) (Putney et al., 2021).

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 Putney et al. (2019) and Putney et al. (2021). 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 Putney et al. (2021). Here, we study the performance of the neural decoder in cross-subject settings, where the test data originates from the target subjects whereas the classifier is trained on source 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 (Putney et al., 2021). 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 Putney et al. (2021), the absence of some (one or two) muscles does not have a significant effect on decoding performance; in fact, as shown in Putney et al. (2021), high decoding accuracy (higher than

90%) can be achieved even with half of the available muscles due to the completeness of the motor program.

3.1.2 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 Putney et al. (2021). 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 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

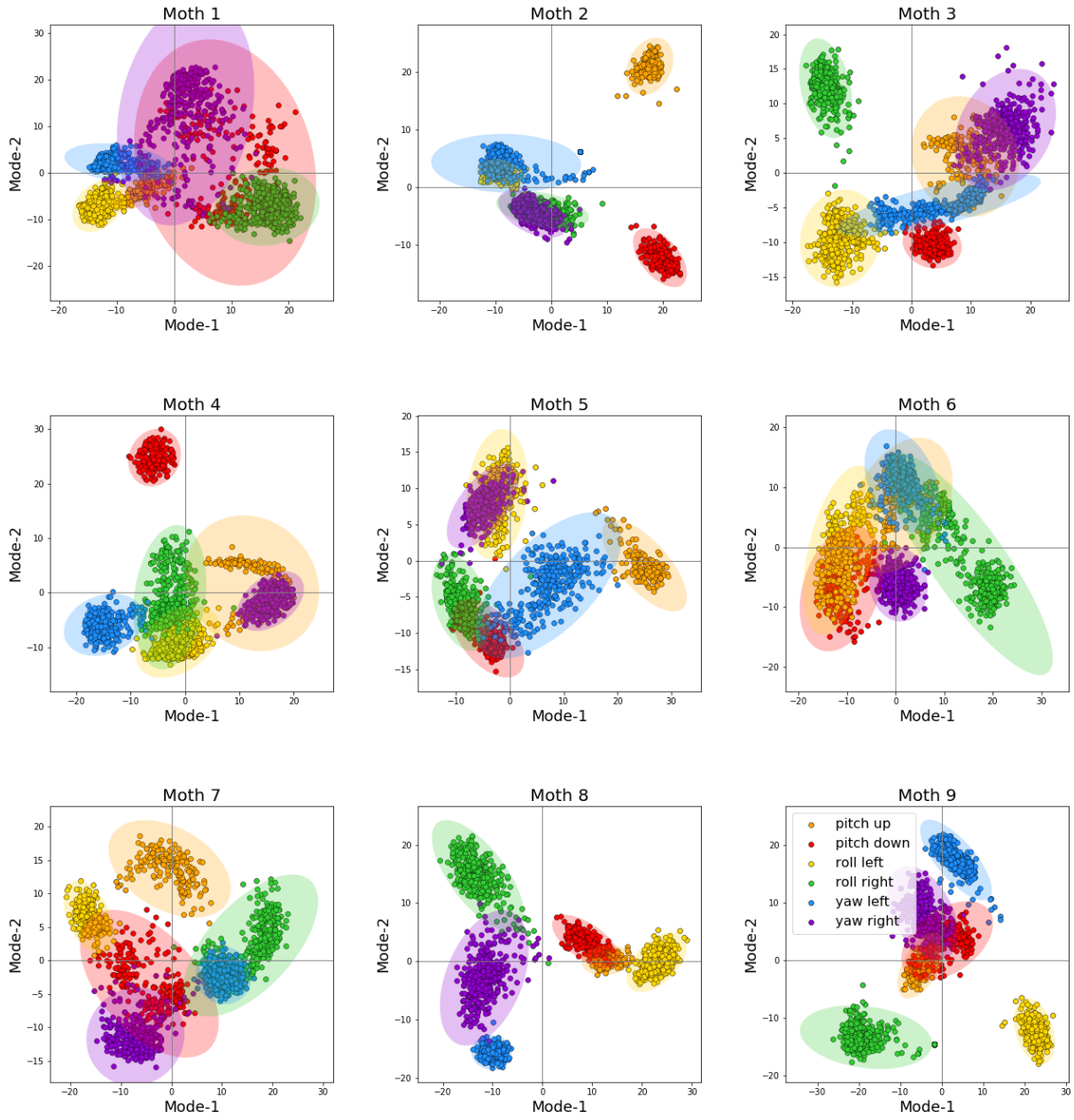


Fig 3: The feature space across the first two principal modes for each moth. For visual clarity, the diagrams also show the confidence ellipse corresponding to one standard deviation for each of the visual conditions.

modes) only increases this separability in the higher-dimensional feature space for each moth as we have observed in past work (Putney et al., 2021); for more details, we advise the interested reader to refer to Putney et al. (2021) where the feature extraction procedure was first proposed and its performance thoroughly analyzed.

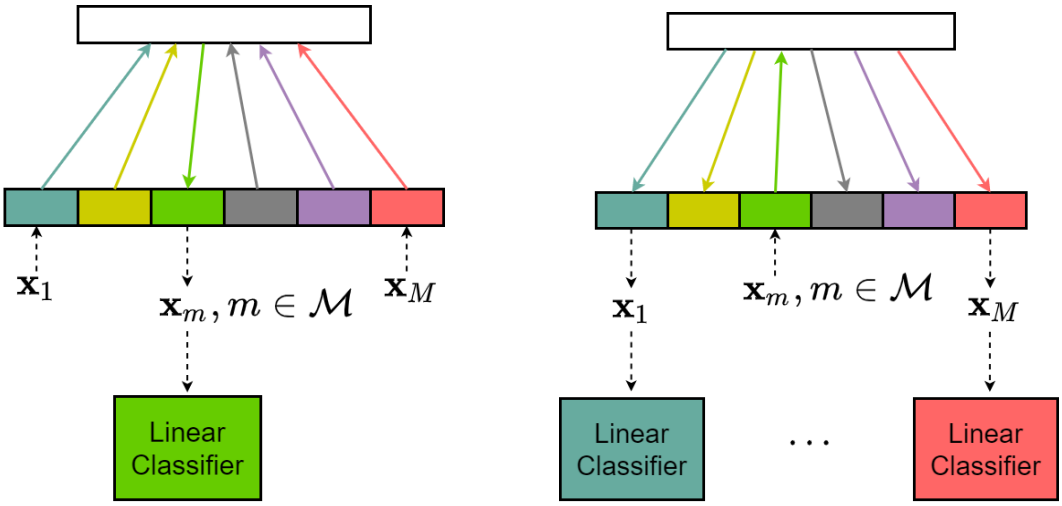
3.2 Scenarios and Methodology

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

- *Scenario II* (Fig. 4b). The target and source index sets are $\mathcal{M}_{\mathrm{D}} = \{m\}$ and $\mathcal{M}_{\mathrm{S}} = \mathcal{M} \backslash \{m\}$, respectively. In other words, we select a single subject $m \in \mathcal{M}$ as the target 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 target feature vector are subsequently decoded using the subject-specific linear classifiers trained on the source subjects $j \neq m$ individually.

In both of these scenarios, we evaluate the performance of the source classifier (trained purely on source data) on the transferred target 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 2.2.5) and FD standing for Fisher divergence (see Section 2.2.6). We compare the performance with three benchmarks:



(a) Scenario I (b) Scenario II

Fig 4: Evaluation scenarios. (a) One subject is the source, all other subjects are targets. (b) One subject is the target, and all other subjects are sources.

1. subject-specific neural decoding, when the classifier is trained *and* tested on purely source 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 target data is directly decoded using the source neural decoder without transferring the target data into source feature space. As we discussed in Section 1, 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 (target) and the train (source) data.

3. cross-subject neural decoding using HiWA, as described in Section 1, is a supervised state-of-the-art linear alignment technique that exploits the cluster structure to solve a hierarchical optimal transport optimization (Lee et al., 2019). This method is widely

applied in neural settings (Dabagia et al., 2023).

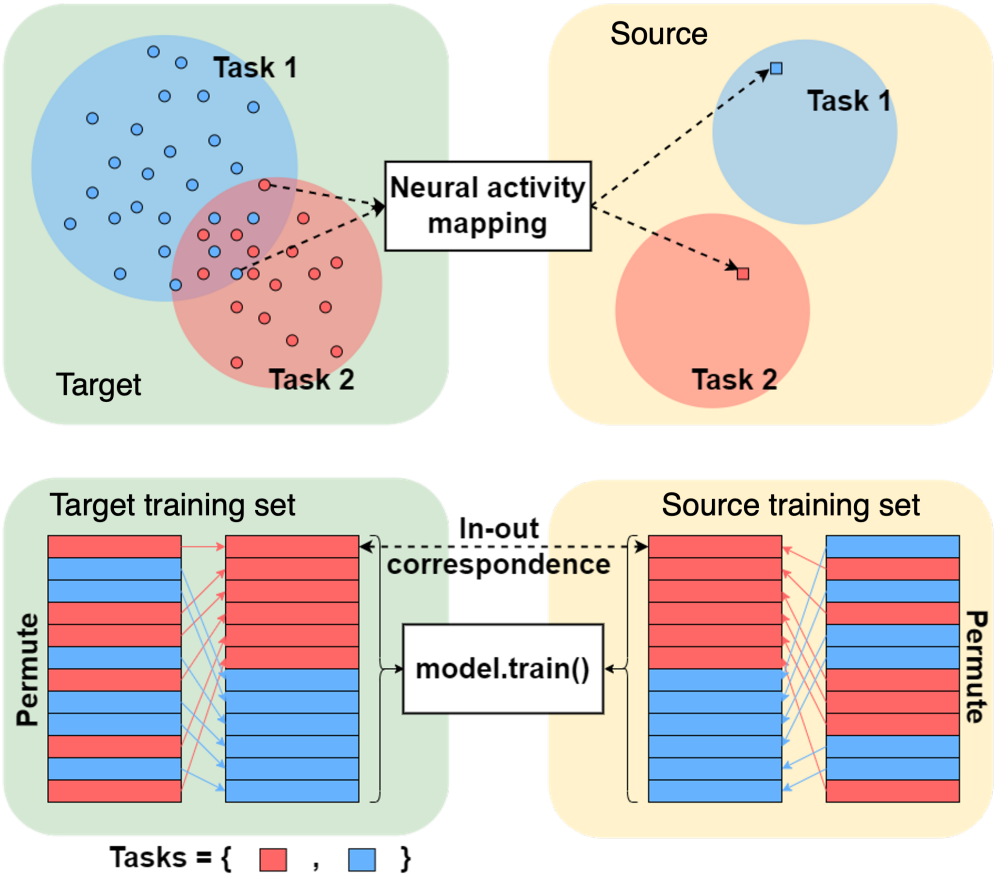


Fig 5: Organization of target and source training data sets.

The training and testing data for the target and source 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 source space feature representation of the target feature vector; the mapping should be consistent with the task that the target features are encoding. In other words, the mapping should be such that the transferred

representation of the target feature vector should be in the region of the source feature space that corresponds to the task that the target 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 target 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 target feature vector from the training set, we randomly choose a source feature vector from the same stimulus class and declare the pair to be an input-output pair.

3.3 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 (Kingma and Ba, 2014)
Learning rate	0.005
Minibatch size	150
Training epochs	350

In both scenarios, we use the Linear Discriminant Analysis (LDA) for classification which is also applied in Putney et al. (2021) and has shown to perform exceptionally high decoding accuracy. Only one RBM is trained for both scenarios

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 first and

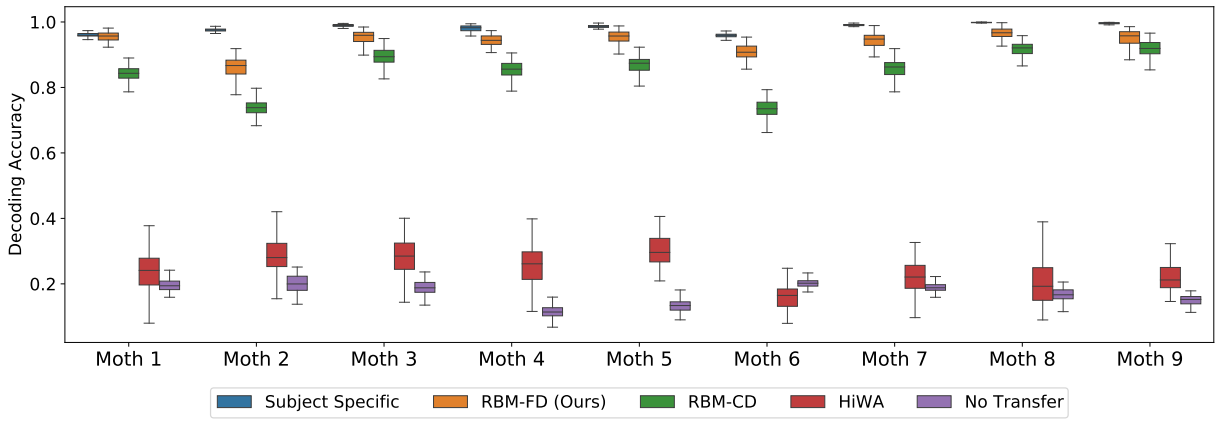


Fig 6: Performance of the cross-subject neural decoder in Scenario I, the error bars are the 95% confidence interval computed using the 100 random initialization (see Fig. 4a and Section 3.2 for details).

second benchmarks, with the performance of the subject-specific neural decoder being the upper bound and the performance of the cross-subject neural decoding with no transfer being the lower bound. 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 3.1). 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 target features to the source is very poor, and produces a poor lower bound, which 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 exacerbates the differences of the

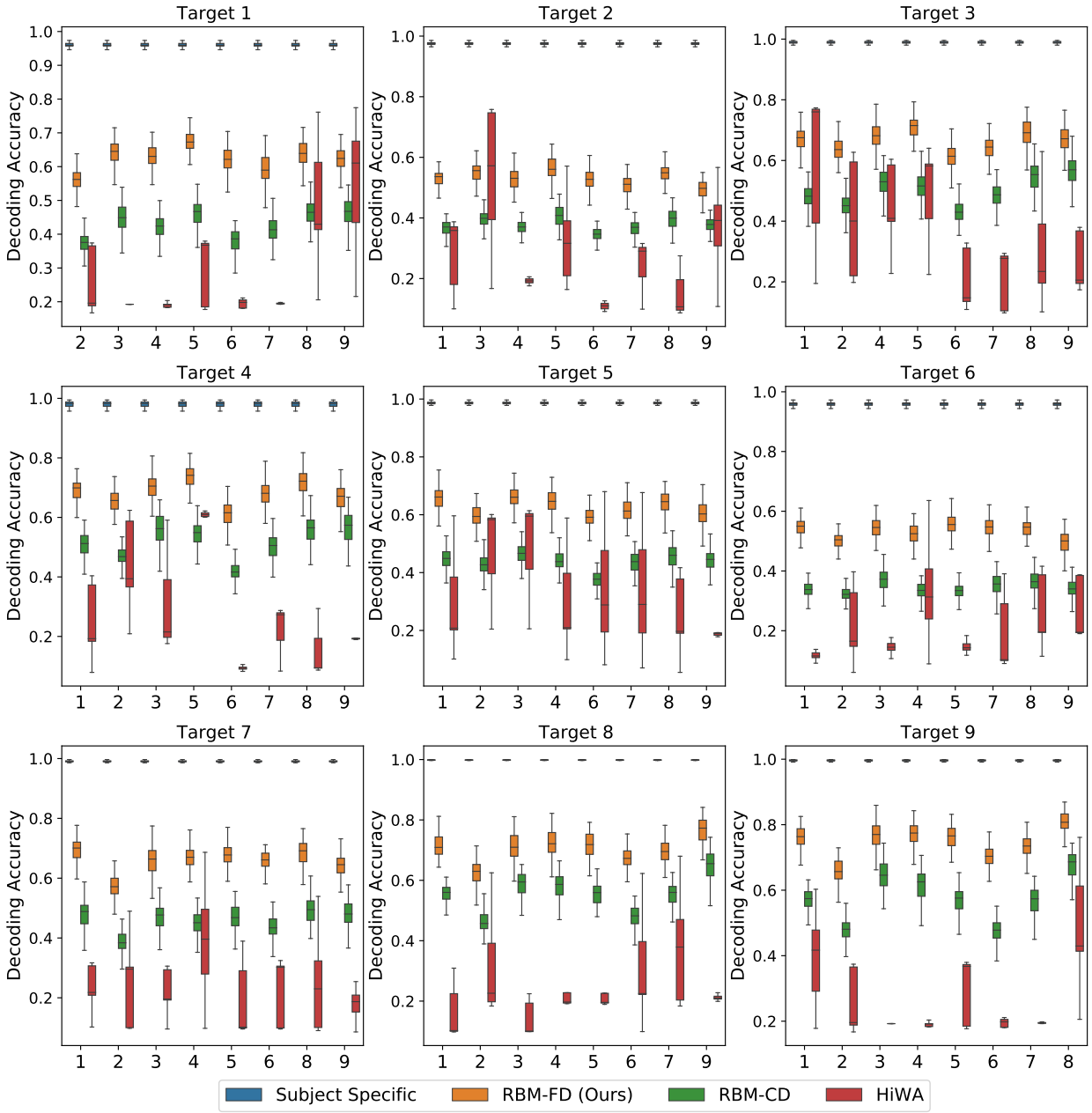


Fig 7: Performance of the cross-subject neural decoder in Scenario II, the error bars are the 95% confidence interval computed using the 100 random initialization (see Fig. 4b and Section 3.2).

.

feature spaces across moths. As a result, directly decoding any target moth using a neural decoder trained over a different, source moth results in poor performance as reported in Fig. 6.

While HiWA does improve upon the lower bound (cross-subject decoding with no transfer), RBM methods consistently outperform HiWA with higher downstream classification accuracy and smaller variances under different random splitting and initialization in both scenarios. With the complex nature of neural data across subjects, there is no guarantee that different subjects can be mapped through a linear alignment, which could be the reason for HiWA's limited performance in this dataset. RBM learns a non-linear mapping function that adapts the target to the source feature space where the decoder was trained with few assumptions; in a highly complex setting with multiple subjects and high dimensionality, which is a common theme in neural datasets, our RBM methods could provide a more adaptable alternative to the widely applied alignment methods

The goal of cross-subject transfer learning is to learn a mapping function that takes target features and adapts them to the source feature space where the decoder was trained. We present the RBM-adapted feature space in Fig. 8 which shows the distribution of the target test points in Scenario I after applying the FD-RBM cross-subject transfer with trained RBM model to the corresponding source feature space. Note that the test points are the features coming from a diverse set of targets, 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 target and maps it into the adequate task-specific region of the source 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 source feature vector $\mathbf{x}_{\mathrm{S}}$ is $M-1$ times larger than the target feature vector $\mathbf{x}_{\mathrm{D}}$; that is, in Scenario II we

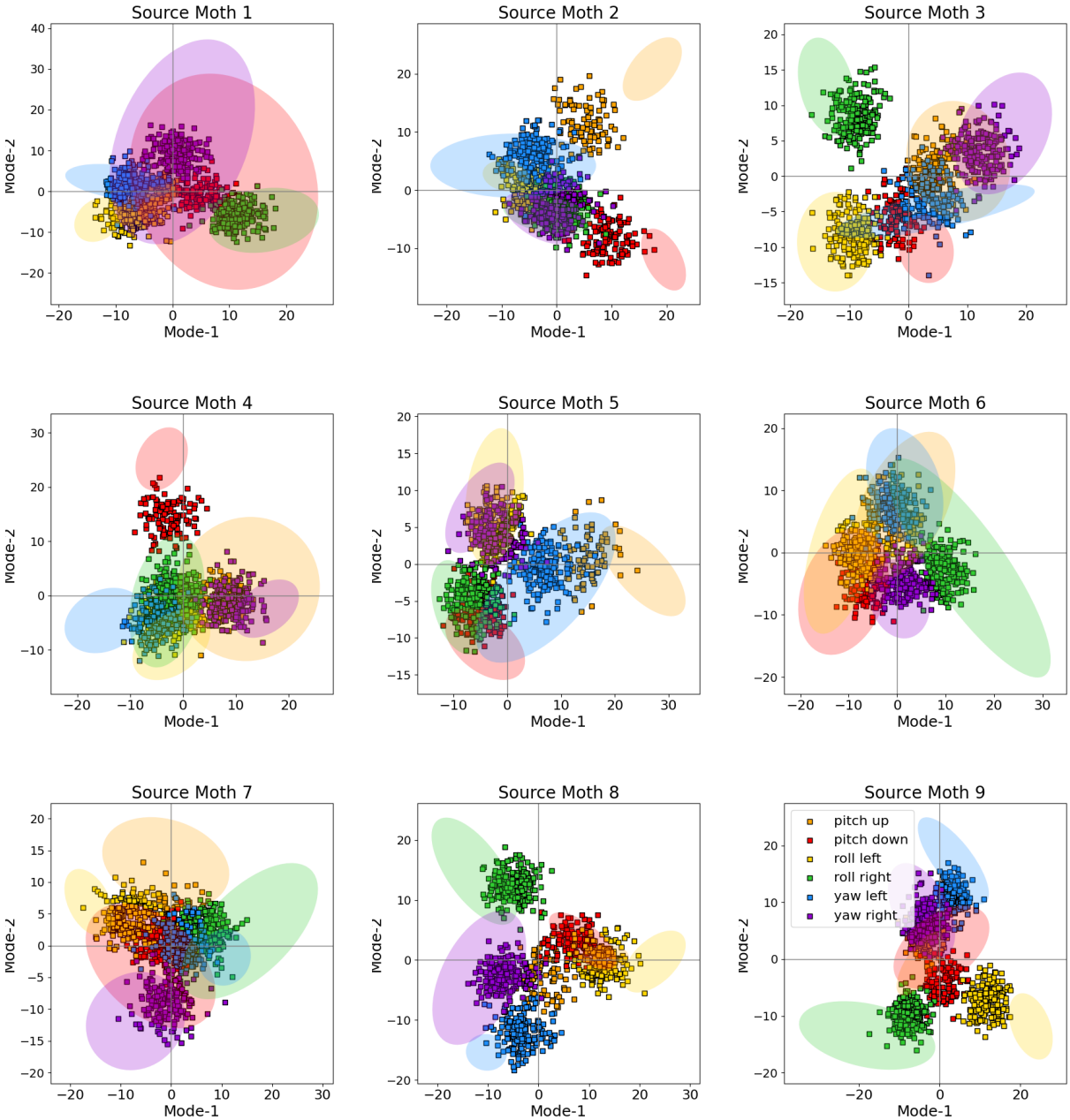


Fig 8: Distribution of target test points (full squares) in the feature spaces of the source moths (represented by the transparent confidence ellipses from Fig. 3) after mapping them using trained FD-RBM model. The mapping corresponds to Scenario I and, as in Fig. 3, only the first two modes of the feature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

are jointly obtaining the representation of a single target in 8 different source 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 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.

4 Discussion

The design of reliable, robust, and computationally efficient solutions for cross-subject mapping and combining data across subjects is a challenging problem in neuroscience. In this paper, we proposed a novel framework for learning the joint distribution of target and source feature representations across a set of subjects using the undirected graphical model and the flexible non-linear generative model RBM. To improve the efficiency of traditional RBM, we

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. We evaluated our proposed scheme against recently developed distribution alignment methods, HiWA, 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, and we found substantial improvement over HiWA in two transfer scenarios. The results verified the viability of the solution. These approaches, which are highly flexible with few assumptions, 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.

Application of Restricted Boltzmann Machine in Neuroscience Restricted Boltzmann machines are widely applied as a tool to study neuroscience problems. The bipartite network structure has long attracted neural science interests to study the structural connection and functional activites in the brain (Hjelm et al., 2014). Recently, RBM has been applied to study the structural connectivity of zebrafish larvae and identified coarse-grained neural grouping (van der Plas et al., 2023). Combining a recurrent element with the RBM for temporal analysis, a variant of RBM, the recurrent temporal RBM was shown to capture the temporal dynamic of the whole brain activities of zebrafish larvae (Monnens et al., 2024).

Our work considers a different application in neural science, where we leverage the undirected graph structure of the RBM to learn joint distributions of neural activities, and then utilize this joint distribution to map between multiple subjects for independent downstream tasks. In general, with its special bipartite network construction and its undirected probabilistic graphical model structure, RBM is a highly flexible tool to further explore neural science for understanding brain structures, and dynamical neural activities, or apply to cross subject mapping.

Efficient and versatile cross-subject mapping Many cross-subject mapping methods have been proposed to improve cross-subject mapping methods to be more accurate, robust, and versatile. One way to map between subjects is through alignment, which aligns the neural representation between two subjects directly after simple preprocessing such as PCA (Gallego et al., 2020; Lee et al., 2019); one can also map the neural representation to a latent manifold before learning a mapping between the source and the target (Dabagia et al., 2023; Degenhart et al., 2020; Herrero-Vidal et al., 2021). More recently, generative algorithms emerging from deep learning were also applied to cross-subject mapping. For example, a Generative Adversarial Network was applied to transfer human activities between two subjects (Soleimani and Nazerfard, 2021) while a conditional VAE was applied to transfer latent neural representation between monkeys (Angjelichinoski, Pesaran and Tarokh, 2020).

However limited studies focus on improving the general efficiency of cross-subject mapping methods, especially when multiple subjects are present in a complete dataset. Most of the cross-subject mapping methods deal with two subjects where at most two transformations are required to enable cross-subject mapping (Herrero-Vidal et al., 2021; Lee et al., 2019). As the number of subjects grows, the number of models needed to transfer between all subjects grows exponentially. To reduce the number of mappings when multiple subjects exist, researchers have developed a feature transfer that needs to be learned in a supervised manner (Zhao et al., 2021). Our work approaches a similar multi-subject/multi-target mapping problem from a probabilistic standpoint: by using RBM and estimating the joint distribution of all subjects at the same time, we learn the conditional probability that allows the mapping of an arbitrary number of targets to an arbitrary number of sources. This process improves the efficiency of cross-subject mapping with no required labels as we directly learn the distribution of features without a decoder.

Cross-subject mapping in a generative, probabilistic way through RBM also allows the

RBM to be more versatile to apply in various settings. A common drawback of current methods of cross-subject mapping requires the availability of the source subject's data during test time. For example, the alignment-based algorithms can only correctly compute the alignment matrix when both the source and test are available (Degenhart et al., 2020; Lee et al., 2019). However, the RBM allows us to transfer different target subjects to the space of the source subject without additional knowledge regarding the source beyond its training data. This is enabled by the probabilistic modeling of RBM. With the joint distribution of the subjects represented by the RBM, conditional distribution can be easily computed for unknown subjects, allowing cross-subject mapping when a subject is not available. The probabilistic framework is also applied to align multiple subjects' neural responses for training a pooled decoder to classify odor stimuli (Herrero-Vidal et al., 2021).

Limitations While the RBM architecture with fisher divergence proposed in this study greatly improves the efficiency of cross-subject mapping over many subjects, limitations remain from several perspectives: its asymmetrical transformation, limited extensions to quantify similarities between subjects, and its lack of structure in the hidden layer that removes interpretability from this architecture.

Many cross-subject mapping algorithms are symmetrical, meaning that transforming the one subject to another subject can be directly applied to transform in the opposite directions with limited changes to the transformation function. Usually, the symmetric methods map two subjects onto a shared embedding space, which is widely applied in neural alignment methods such as (Harvey et al., 2024; Kriegeskorte et al., 2008). The RBM cross-subject mapping method is directional: the transformation from one subject to all of the other subjects learned by an RBM is characterized through complex neural networks that are non-linear. Directional transformations inevitably will make the transformation process less efficient, but the capability of mapping between multiple subjects at once compensates for

the lack of symmetry in the transformation.

Another drawback of our RBM methods is that we cannot easily derive a distance or similarity metric to quantify the similarities between two (groups) of subjects. Many alignment-based methods, such as Harvey et al. (2024), can directly extract the similarity information between two subjects once the alignment is complete. We refer to Klabunde et al. (2023) for a more comprehensive review of similarity metrics derived from cross-subject mapping or alignment. With RBM being an estimator of the joint distribution of all subjects capable of sampling from conditional distributions, it is possible to compute information-based metrics such as the conditional entropy between two (groups of) subjects. One issue with conditional entropy, however, is that this metric is not symmetric, and thus cannot be used explicitly to quantify similarities between subjects. Symmetrical information metrics like the mutual information between two (groups) of subjects cannot be computed with the current architecture as marginal distribution of subjects is required for that computation. However, the conditional entropy is not strictly a distance metric because it is not symmetric. Modifying the current RBM architecture could lead to a system that can quantify similarities between subjects through mutual information, which is symmetric, but that is beyond the scope of this current study.

The lack of interpretability is another limitation of our current architecture. Cross-subject mapping methods that utilize algebraic solutions can identify how each feature of the target subject is mapped to the space of the source subject. For example, the Procrustes Alignment method computes a simple transformation matrix that maps the target data onto source space (Harvey et al., 2024). However, it is difficult to understand the operations applied to the target data by the RBM to transform them into the source space due to their non-linear nature. Various network analysis methods could be used to test the sensitivity of the representation and decoding to specific inputs (Yang et al., 2022). So where the goal of an analysis is

prediction, classification, or other types of questions amenable to non-linear ML approaches, our RBM with Fisher divergence method provides a flexible, efficient framework for combining neural data across subjects even when the recordings are sparse spiking representations

Future work 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 source features given the target 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 target 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

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

Angjelichinoski, M., Pesaran, B. and Tarokh, V. (2020). Deep cross-subject mapping of neural activity, *ArXiv* **abs/2007.06407**.

Carreira-Perpinan, M. A. and Hinton, G. (2005). On contrastive divergence learning, *International workshop on Artificial Intelligence and Statistics (AISTATS)*, PMLR, pp. 33–40.

Chai, R., Ling, S. H., San, P. P., Naik, G. R., Nguyen, T. N., Tran, Y., Craig, A. and Nguyen, H. T. (2017). Improving eeg-based driver fatigue classification using sparse-deep belief networks, *Front. Neurosci.* **11**: 103.

Dabagia, M., Kording, K. P. and Dyer, E. L. (2023). Aligning latent representations of neural activity, *Nature Biomedical Engineering* **7**(4): 337–343.

Degenhart, A. D., Bishop, W. E., Oby, E. R., Tyler-Kabara, E. C., Chase, S. M., Batista, A. P. and Yu, B. M. (2020). Stabilization of a brain–computer interface via the alignment of low-dimensional spaces of neural activity, *Nature biomedical engineering* **4**(7): 672–685.

Dyer, E. L., Gheshlaghi Azar, M., Perich, M. G., Fernandes, H. L., Naufel, S., Miller, L. E. and Körding, K. P. (2017). A cryptography-based approach for movement decoding, *Nature biomedical engineering* **1**(12): 967–976.

Farahani, H. S., Fatehi, A. and Shoorehdeli, M. A. (2020). Between-domain instance transition via the process of gibbs sampling in rbm, *arXiv preprint arXiv:2006.14538* .

Gallego, J. A., Perich, M. G., Chowdhury, R. H., Solla, S. A. and Miller, L. E. (2020). Long-term stability of cortical population dynamics underlying consistent behavior, *Nature neuroscience* **23**(2): 260–270.

Hajinoroozi, M., Mao, Z., Jung, T.-P., Lin, C.-T. and Huang, Y. (2016). Eeg-based prediction of driver's cognitive performance by deep convolutional neural network, *Signal Process.: Image Commun.* **47**: 549–555.

Harvey, S. E., Larsen, B. W. and Williams, A. H. (2024). Duality of bures and shape distances with implications for comparing neural representations, *Proceedings of UniReps: the First Workshop on Unifying Representations in Neural Models*, PMLR, pp. 11–26.

Haxby, J. V., Guntupalli, J. S., Connolly, A. C., Halchenko, Y. O., Conroy, B. R., Gobbini, M. I., Hanke, M. and Ramadge, P. J. (2011). A common, high-dimensional model of the representational space in human ventral temporal cortex, *Neuron* **72**(2): 404–416.

Herrero-Vidal, P., Rinberg, D. and Savin, C. (2021). Across-animal odor decoding by probabilistic manifold alignment, *Advances in Neural Information Processing Systems* **34**: 20360–20372.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence, *Neural Comput.* **14**(8): 1771–1800.

Hjelm, R. D., Calhoun, V. D., Salakhutdinov, R., Allen, E. A., Adali, T. and Plis, S. M. (2014). Restricted boltzmann machines for neuroimaging: an application in identifying intrinsic networks, *NeuroImage* **96**: 245–260.

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

Hyvärinen, A. (2007). Connections between score matching, contrastive divergence, and pseudolikelihood for continuous-valued variables, *IEEE Trans. Neural. Netw.* **18**: 1529–1531.

Jayaram, V., Alamgir, M., Altun, Y., Scholkopf, B. and Grosse-Wentrup, M. (2016). Transfer learning in brain-computer interfaces, *IEEE Comput. Intell. Mag.* **11**(1): 20–31.

Karpowicz, B. M., Ali, Y. H., Wimalasena, L. N., Sedler, A. R., Keshtkaran, M. R., Bodkin, K., Ma, X., Miller, L. E. and Pandarinath, C. (2022). Stabilizing brain-computer interfaces through alignment of latent dynamics, *bioRxiv* pp. 2022–04.

Kim, H.-C., Jang, H. and Lee, J.-H. (2020). 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* **330**: 108451.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization.

Klabunde, M., Schumacher, T., Strohmaier, M. and Lemmerich, F. (2023). Similarity of neural network models: A survey of functional and representational measures, *arXiv preprint arXiv:2305.06329* .

Kriegeskorte, N., Mur, M. and Bandettini, P. A. (2008). Representational similarity analysis-connecting the branches of systems neuroscience, *Frontiers in systems neuroscience* **2**: 249.

Lee, J., Dabagia, M., Dyer, E. and Rozell, C. (2019). Hierarchical optimal transport for multimodal distribution alignment, *Advances in neural information processing systems* **32**.

Li, F., Tran, L., Thung, K.-H., Ji, S., Shen, D. and Li, J. (2015). A robust deep model for improved classification of ad/mci patients, *IEEE J. Biomed. Health Inform.* **19**(5): 1610–1616.

Monnens, S. Q., Peters, C., Hesselink, L. W., Smeets, K. and Englitz, B. (2024). The recurrent temporal restricted boltzmann machine captures neural assembly dynamics in whole-brain activity, *eLife* **13**: RP98489.

Pandarinath, C., Ames, K. C., Russo, A. A., Farshchian, A., Miller, L. E., Dyer, E. L. and Kao, J. C. (2018). Latent factors and dynamics in motor cortex and their application to brain–machine interfaces, *Journal of Neuroscience* **38**(44): 9390–9401.

Plis, S. M., Hjelm, D. R., Salakhutdinov, R., Allen, E. A., Bockholt, H. J., Long, J. D., Johnson, H. J., Paulsen, J. S., Turner, J. A. and Calhoun, V. D. (2014). Deep learning for neuroimaging: a validation study, *Front. Neurosci.* **8**: 229.

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

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

Rao, R. P. N. (2013). *Brain-computer interfacing: an introduction*, Cambridge University Press.

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

Soleimani, E. and Nazerfard, E. (2021). Cross-subject transfer learning in human activity recognition systems using generative adversarial networks, *Neurocomputing* **426**: 26–34.

Torres-Oviedo, G. and Ting, L. H. (2010). Subject-specific muscle synergies in human balance control are consistent across different biomechanical contexts, *Journal of neurophysiology* **103**(6): 3084–3098.

van der Plas, T. L., Tubiana, J., Le Goc, G., Migault, G., Kunst, M., Baier, H., Bormuth, V., Englitz, B. and Debrégeas, G. (2023). Neural assemblies uncovered by generative modeling explain whole-brain activity statistics and reflect structural connectivity, *Elife* **12**: e83139.

Wei, B. and Pal, C. (2011). Heterogeneous transfer learning with rbms, *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 25, pp. 531–536.

White, H. (1982). Maximum likelihood estimation of misspecified models, *Econometrica* pp. 1–25.

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

Yang, H., Winter, S., Zhang, Z. and Dunson, D. (2022). Interpretable ai for relating brain structural and functional connectomes, *arXiv preprint arXiv:2210.05672* .

Zhao, Y., Dai, G., Borghini, G., Zhang, J., Li, X., Zhang, Z., Aricò, P., Di Flumeri, G., Babiloni, F. and Zeng, H. (2021). Label-based alignment multi-source domain adaptation for cross-subject eeg fatigue mental state evaluation, *Frontiers in Human Neuroscience* **15**: 706270.