Extracting Semantic-Dynamic Features for Long-Term Stable Brain Computer Interface

Tao Fang[12], Qian Zheng[12], Yu Qi[13*], Gang Pan[12*]

[1]The State Key Lab of Brain-Machine Intelligence, Zhejiang University, Hangzhou, China
[2]College of Computer Science and Technology, Zhejiang University, Hangzhou, China
[3] MOE Frontier Science Center for Brain Science and Brain-machine Integration, Zhejiang University, Hangzhou, China
{duolafang, qianzheng, qiyu, gpan}@zju.edu.cn

Abstract

Brain-computer Interface (BCI) builds a neural signal to the motor command pathway, which is a prerequisite for the realization of neural prosthetics. However, a long-term stable BCI suffers from the neural data drift across days while retraining the BCI decoder is expensive and restricts its application scenarios. Recent solutions of neural signal recalibration treat the continuous neural signals as discrete, which is less effective in temporal feature extraction. Inspired by the observation from biologists that low-dimensional dynamics could describe high-dimensional neural signals, we model the underlying neural dynamics and propose a semantic-dynamic feature that represents the semantics and dynamics in a shared feature space facilitating the BCI recalibration. Besides, we present the joint distribution alignment instead of the commonly used marginal alignment strategy, dealing with the various complex changes in neural data distribution. Our recalibration approach achieves state-of-the-art performance on the real neural data of two monkeys in both classification and regression tasks. Our approach is also evaluated on a simulated dataset, which indicates its robustness in dealing with various common causes of neural signal instability.

Introduction

The Brain-computer Interface, BCI (Chapin et al. 1999; Hochberg et al. 2006; Zhang et al. 2019a; Fang, Qi, and Pan 2020) provides direct brain control of external devices by decoding the motor intentions from neural activities, which has demonstrated the potential in motor rehabilitation and restoration. However, long-term available BCI suffers from the intrinsic instability caused by several inevitable problems (*e.g.*, elusive flow of the tissues, neuron necrosis, and electrodes displacement (Barrese et al. 2013; Degenhart et al. 2020)), which restricts the promotion of BCI from the lab to real-life applications. A widely adopted strategy for alleviating such degradation is daily recalibration (Ajiboye et al. 2017). Recalibration with supervised learning approaches (Wen et al. 2021; Brandman et al. 2018) enables high performance. However, labeling the newly-collected data is often expensive, raising restrictions for promoting BCI to real-life applications. Recently, leveraging unlabelled data for recalibration has received increasing interest.

*Corresponding authors.

Motor-related neural signals (*e.g.*, cortical signals in dorsal premotor cortex, PMd or primary motor cortex, M1) have been widely used in mature BCI experiments and even neuroprosthesis (Hochberg et al. 2012; Wodlinger et al. 2014; Pandarinath et al. 2017; Pan et al. 2018). Researchers have revealed motor-related neural signals to be a high-dimensional mapping of low-dimensional underlying dynamics (Pandarinath et al. 2018b; Churchland et al. 2012). Therefore, recalibrating motor-related neural signals with unlabeled data could be achieved by unsupervised domain adaptation (UDA) considering the dynamics. However, prevalent UDA methods are designed for non-sequential data (*e.g.*, images (Long et al. 2014; Sun, Feng, and Saenko 2016)) and primarily focus on spatial features. Existing UDA methods for sequential data (*e.g.*, texts (Zhang et al. 2019b), audios (Drossos, Magron, and Virtanen 2019), cortical signals (Dyer et al. 2017; Farshchian et al. 2018)) also fail to extract dynamic features. That is, they treat the sequence to be discrete and extract features by aggregating local ones, which ignores the structure (Wang et al. 2018) or dynamics (top row, Fig. 1). Besides, these methods only align the marginal distribution of the features while do not explicitly align decision boundaries, which may fuse semantics in the target domain and degrade the classification performance (bottom row, Fig. 1).

To extract dynamic features for recalibration, our basic idea is to force the extracted semantic features across different classes embedded in the initial-point subspace of a manifold determined by an autonomous linear dynamical system. The embedding representation contains the semantics, while its spatial information in the manifold reflects the dynamic features (as the dynamic trajectories are decided by its initial points). By compressing the dynamic feature as an initial point, the intrinsic dynamics instead of discrete features for each time stamp could be comprehensively considered. The unified representation of **S**emantic and **D**ynamic (SD) features facilitates unsupervised recalibration in two aspects. 1) The manifold determined by the dynamical system is long-term stable as discovered in (Gallego et al. 2020), which is free from recalibration across domains. 2) In this manifold, aligning the unified representation aligns semantics and dynamics autonomously, and we only need to focus on the SD feature alignment in the recalibration.

Concretely, we use an observation module to extract the

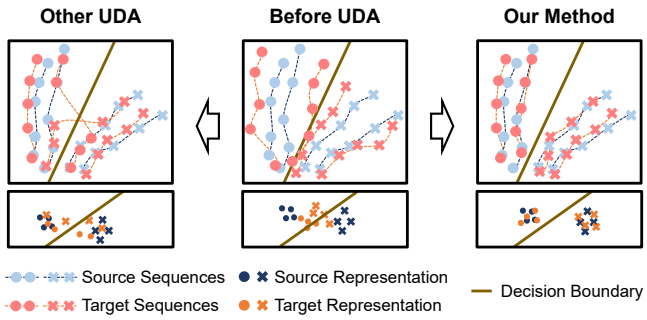

Figure 1: Illustration of the difference between other UDA methods and ours when aligning sequence data. Top row: other methods destroy the integrity of sequences (left) while ours maintains the dynamics (right). Bottom row: in feature space, other methods only consider marginal alignment, neglecting semantics and confusing decision boundaries (left), while ours additionally considers conditional alignment to explicitly align decision boundaries (right).

feature of input sequences. To obtain the SD features, we tailor 1) a classifier module that forces the extracted features to encode the task-related semantics and 2) a motor decoding module that forces the extracted features to embed in the initial-point subspace of a dynamical system. We use semantic labels and real motion trajectories in the source domain as the supervision to train these three modules. As the underlying dynamics of the motor cortex (MC) lie in a long-term stable manifold and are invariant across days (Pandarinath et al. 2018b; Gallego et al. 2020), the recalibration of motor-related neural signals could be achieved by fixing the dynamical system (or the motor decoding module) and fine-tuning the observation module (using data in the target domain). The classifier module is also fixed to keep the hypothesis shared across domains. To avoid the semantic confusion arising from marginal alignments (fuzzy decision boundary as (Kang et al. 2019)), we adopt a joint distribution alignment, *i.e.*, marginal and conditional alignments, explicitly consider the semantics during alignment. Our contributions can be summarized as follows.

- We propose an unsupervised recalibration method based on the stable circuit assumption, which is robust to daily changed motor-related neural signals.
- We present a feature extraction method that tailors a classifier and a motor decoding module, which successfully embeds the semantics and dynamics into a unified feature space.
- We introduce a feature alignment method that jointly considers marginal and conditional alignments, which alleviates the fuzzy semantics after recalibration of motor-related neural signals.

Related Work

The underlying dynamics extraction in brain signals. The motor cortex signals, which have shown a high correlation with the preparation and execution of the muscle movements, have been widely used in BCI and neuroprosthesis for intention or instruction estimation (Qi et al. 2022), and research finds it strongly governed by the intrinsic dynamics (Pandarinath et al. 2018a; Churchland et al. 2012). There are several studies modeling the internal dynamical system on the observed neural data (Pandarinath et al. 2018b; Kao, Ryu, and Shenoy 2015), extracting low-dimensional latent states as de-noised dynamics and trying to estimate the motion-related parameters based on the dynamics. Such linear dynamical system based approaches suppose that the current latent state could be modeled as a linear function of the previous states. With the internal factors interpreted as smooth dynamics, some methods achieved more accurate and robust BCI (Pandarinath et al. 2018b). However, these methods still require a number of manual labels to fit parameters when facing new sessions or environments, and cannot fully utilize the dynamic structure of the neural data itself for future generalization.

Unsupervised domain adaptation (UDA). The prevalent UDA methods could be roughly categorized as the discrepancy-minimizing methods and the adversarial domain adaptation methods. The discrepancy-based methods adopt various domain distance measures and try to minimize such domain discrepancy by matching the statistical moments of different domains' distributions (Long et al. 2014; Borgwardt et al. 2006; Sun, Feng, and Saenko 2016). The adversarial domain adaptation (Ganin et al. 2016) is based on the deep networks, and introduces an additional discriminator to confuse features of different domains by training the model as a two-player adversarial game. In addition to merely matching the marginal distributions, recently, researchers have introduced joint distribution alignment or conditional distribution alignment (Kang et al. 2019; Xie et al. 2018; Chen et al. 2019) to handle more complex domain shift situations like concept shift (Zhao et al. 2020). For the domain shift in brain signals, some related works accomplish the recalibration for BCIs by supervised retraining (Ajiboye et al. 2017) or retraining with different kinds of auxiliary information (Degenhart et al. 2020; Kao, Ryu, and Shenoy 2015; Gallego et al. 2020; Wen et al. 2021). To make BCIs close to practical application, recently, researchers focus on unlabeled neural decoder recalibration on different days (*e.g.*, without motion trajectories) and taking it as a UDA task (Dyer et al. 2017; Farshchian et al. 2018). However, these methods focus on point-to-point alignment without considering the global-sequence features, and only use marginal alignment without using the intrinsic semantics, still having limitations for complex drift scenarios.

Preliminary

Problem Definition

The recalibration of neural data is a UDA problem. We treat the neural and motion data from different sessions (usually from different days) as different domains. We have a labeled source dataset $D^{\mathrm{s}} = (X^{\mathrm{s}}, Y^{\mathrm{s}})$ where the neural data X^{s} and motion data Y^{s} are composed of several sequences $(\mathbf{x}_i^{\mathrm{s}}, \mathbf{y}_i^{\mathrm{s}})$ that $X^{\mathrm{s}} = \{\mathbf{x}_i\}_{i=1}^{N^{\mathrm{s}}}, Y^{\mathrm{s}} = \{\mathbf{y}_i\}_{i=1}^{N^{\mathrm{s}}}$ and an unlabeled target dataset $D^{\mathrm{t}} = X^{\mathrm{t}}$ consisting of sequences $\mathbf{x}_i^{\mathrm{t}}$

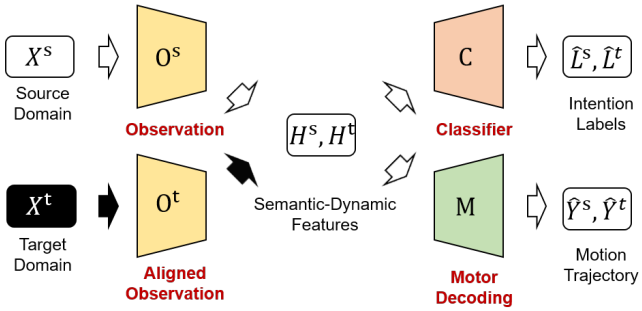

Figure 2: The flow chart of our recalibration method.

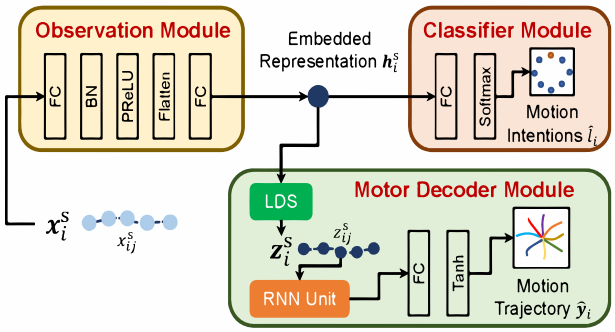

Figure 3: The schematic diagram of the feature extraction phase. Input $\mathbf{x}_i^s$ is a sequence that describes a complete motion, and x_{ij} is the recording at timestamp j. SD-feature h_i^s is extracted for motion intention classification (top right) and motion trajectory regression (bottom right), respectively. Abbreviations: FC, Fully Connected layer; BN, Batch Normalization layer; LSD, Linear Dynamical system.

that $X^t = \{\mathbf{x}_i\}_{i=1}^{N^t}$ without motion signals. In the following, we organize the neural and motion data as a collection of sequences $X^s \in \mathcal{R}^{N^s \times T \times c}$, $X^t \in \mathcal{R}^{N^t \times T \times c}$ and $Y^s \in \mathcal{R}^{N^s \times T \times 2}$, where each sequence lasts for T time stamps (a complete motion trial), the inputs' dimension is c and the total sequence numbers are N^s and N^t in source and target domains respectively. Note that T and c are shared parameters across domains. Besides, we have a sequence-level label $L^s = \{l_i^s\}_{i=1}^{N^s}$ which indicates the intention of the whole trial (like the target of the reaching movement) for the classifier module. In the domain shift scenarios, we supposed the (X^s, Y^s, L^s) and (X^t, Y^t, L^t) are sampled from different distributions that $\mathrm{P}(X^s, Y^s, L^s) \neq \mathrm{P}(X^t, Y^t, L^t)$.

For the estimated latent states, we have SD features $H^s = \{h_i^s\}_{i=1}^{N^s}$ and $H^t = \{h_i^t\}_{i=1}^{N^t}$ in bottleneck and the corresponding latent dynamics as $Z^s = \{\mathbf{z}_i^s\}_{i=1}^{N^s}$ and $Z^t = \{\mathbf{z}_i^t\}_{i=1}^{N^t}$, where $h^s, h^t \in R^d$, $\mathbf{z}_i^t$ and $\mathbf{z}_i^t \in R^{T \times d}$ where d denotes the hidden states' dimension. The UDA task indicates mapping the source and target data into one shared feature space in which $\mathrm{P}(H^s, Y^s, L^s) = \mathrm{P}(H^t, Y^t, L^t)$ ideally and making the hypothesis learned on the source features work on the target features.

Theoretical Analysis

Theoretically summarized in (Ben-David et al. 2010), given the source and target domains D_s, D_t, we could measure the error ϵ_t of a hypothesis $\pi \in \mathcal{H}$ as a summary of:

$$\epsilon_t(\pi) \leq \epsilon_s(\pi) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\overline{D}_s, \overline{D}_t) + \lambda_{\mathcal{H}}, \qquad (1)$$

which could be roughly divided as the error of the hypothesis on the source domain, the discrepancy between the source and target domain, and the joint hypothesis error $\lambda_{\mathcal{H}}$ respectively. Traditional marginal alignment methods measure the discrepancy $d_{\mathcal{H}\Delta\mathcal{H}}$ by the distance of statistical momentum or an additional discriminator and minimize it, taking $\lambda_{\mathcal{H}}$ as an ignorable term. However, in the neural data shift case only aligning the marginal distribution means ignoring the conditional or label shift (Zhao et al. 2020) situations (*e.g.*, changes of neural tuning functions) which are common in BCI (Degenhart et al. 2020), causing great inter-class overlapping areas in the feature space and degrading the hypothesis' performance on the aligned target features.

Method

Our recalibration method is summarized in the flow chart Fig. 2. In the feature extraction phase, we take the source data X^s as inputs and extract the embeddings H^s by an observation module $H^s = \mathrm{O}^s(X^s)$. Then two decoders including the classifier module C and the motor decoding module M predict the motion intention labels $\hat{L}^s$ and the motion trajectories $\hat{Y}^s$ from H^s, respectively. The whole process is trained end-to-end with the supervision of intention labels and motion trajectories. In the feature alignment phase, we fine-tune the observation module O^s and obtain the aligned observation one O^t, which uses data X^t from the target domain to align its feature space H^t to H^s.

Feature Extraction

As discussed in (Pandarinath et al. 2018b,a), the neural activity in MC could be reasonably translated into the intrinsic dynamics, which is robust, noise-free, and more accurate for its downstream tasks (like motion prediction). Some studies (Tanji and Evarts 1976) indicate that given enough preparation time, signals like PMd may encode a complete motion intention for the following piece of time. Considering the brain as a dynamical system, such a preparatory state works as the initial state for the following dynamics. Based on such discovery, we build a model to extract the preparatory state as a semantic embedding to encode the motion intention and assume that the future neural dynamics are predictable based on it. The schematic diagram of the feature extraction is displayed in Fig. 3.

Observation module O^s. As shown in Fig. 3, the observation module is a multi-layer perception (MLP) network. It first uses a fully connected (FC) layer for dimension reduction to extract spatial features. It then uses a flattened layer to combine the channel dimension with the temporal dimension as one feature dimension and introduce another

FC layer to extract a fixed-length embedding h_i^{s} (*i.e.*, SD features) to retain the temporal information of the sequence.

Motor decoding module M**.** Based on the assumption that the underlying dynamics of short-term MC signals are predictable (Pandarinath et al. 2018a) when well prepared, we assume h_i^{s} has taken the preparatory information. By combining an autonomous linear dynamical system (LDS), *i.e.*, taking the embedding h_i^{s} as the initial state, h_i^{s} could be used for complete motion trajectory extraction. An autonomous LDS that let the dynamic trajectory $\mathbf{z}_i^{\mathrm{s}}$ evolve from h_i^{s} is constructed as:

$$\dot{Z}^s = \mathrm{F}(Z^s)W, z_{i0}^s = h_i^s, \tag{2}$$

where $\dot{Z}^s$ is the first derivative of Z^s and $F(Z^s)$ indicates the collection of candidate functions $F(Z^s) = [f_1(Z^s), f_2(Z^s), ..., f_k(Z^s)]$ (*e.g.*, $f_i(Z^s) = (Z^s)^2$). $W \in R^{T\times d}$ denotes the coefficients that is assigned to each term in $\mathrm{F}(Z)$ where d is the dimension of the latent state $\mathbf{z}_i^s$ and h_i^s. For complex scenarios, a combination of different functions $f_i(Z)$ is preferred (Luan, Liu, and Sun 2022). While for the regular and short-term (about 1 second) movements in our case, we use a simpler formation $\mathrm{F}(Z) = Z$.

To estimate the parameter W, we introduce an RNN ($\mathrm{M_{rnn}}$) to predict the motion trajectory from $\mathbf{z}_i^s$. $\mathrm{M_{rnn}}$ is trained by a reconstruction loss. The parameters of $\mathrm{O^s}$, W and $\mathrm{M_{rnn}}$ are estimated together by minimizing:

$$\mathcal{L}_{\mathrm{rec}} = \sum_{i=0}^{N_{\mathrm{s}}} ||\mathbf{y}_i^s - \mathrm{M_{rnn}}(\mathbf{z}_i^s)||^2. \tag{3}$$

Additionally, we introduce a regularization term on W to restrict $|\dot{Z}^s|$ because a large $|\dot{Z}^s|$ makes the dynamic trajectory move rapidly in the unit feature space, causing lots of points to pile up at the activation function boundary thus reducing performance. Here we minimize an L2 loss $\mathcal{L}_{\mathrm{reg}} = ||W||^2$.

The autonomous LDS is simple and has only one equilibrium point, making the model explainable and easy to analyze. During experiments, the learned parameter W tends to be negative-definite and $\mathbf{z}_i^{\mathrm{s}}$ tends to move towards the equilibrium point in the vector field[1]. As the initial point, the spatial information of h_i^{s} represents the entire dynamic trajectory $\mathbf{z}_i^{\mathrm{s}}$ evolved from it and due to the unique correspondence between h_i^{s} and $\mathbf{z}_i^{\mathrm{s}}$ in LDS, we will not get the same h_i^{s} for different trajectories to avoid semantic confusion. Therefore, aligning (H^s, H^t) aligns (Z^{s}, Z^{t}). Moreover, the linear dynamics cannot well describe the real complex movements (*e.g.*, motion perturbation) and the mapping from Z^{s} to Y^{s} is unlikely to be directly fitted, so the non-linear RNN $\mathrm{M_{rnn}}$ is introduced to link the linear dynamics to nonlinear observations in this module.

Classifier module C**.** Further, we propose to optimize the semantic representation H^{s} by assigning it semantic information. Concretely, we introduce an auxiliary classification task on H^{s} to make it clustered and semantically meaningful. The classifier C is an MLP with a softmax layer for posterior probability inference, which is trained by the cross entropy loss:

$$\mathcal{L}_{\mathrm{CLF}} = \sum_{i=1}^{N_{\mathrm{s}}}\sum_{j=1}^{N_{\mathrm{c}}} l_{i,j}^{\mathrm{s}} \log \hat{l}_{i,j}^{s}, \tag{4}$$

where $\hat{l}_i^s = C(h_i^{\mathrm{s}})$ and N_{c} is the number of classes. We decode L^s and Y^s from two modules M and C, respectively, based on the assumption that $\mathrm{P}(Y^s, L^s, H^s) = \mathrm{P}(Y^s|H^s)\mathrm{P}(L^s|H^s)\mathrm{P}(H^s)$ but ignore the correlation of $\mathrm{P}(L^s|Y^s)$ or $\mathrm{P}(Y^s|L^s)$ when L^s and Y^s have been collected.

Overall, the parameters of the encoder-decoder model ($\mathrm{O}^s, W, \mathrm{M_{rnn}}, \mathrm{C}$) are optimized end-to-end by $\mathcal{L}_{\mathrm{pretrain}}$:

$$\mathcal{L}_{\mathrm{pretrain}} = \mathcal{L}_{\mathrm{rec}} + \lambda_1\mathcal{L}_{\mathrm{CLF}} + \lambda_2\mathcal{L}_{\mathrm{reg}}. \tag{5}$$

Feature Alignment

Several causes have been found behind the domain shift in BCI (Degenhart et al. 2020) including neuron death, electrode movement (marginal shift), and changes in neural tuning functions (conditional shift). Instead of only aligning the marginal distribution, we first separate the joint distribution alignment problem into two subtasks: marginal alignment and conditional alignment, and try to solve them separately.

The alignment of the joint distribution could be divided as reducing the marginal and conditional distribution discrepancy respectively (Tanwani 2021):

$$\begin{aligned}
&|\log \mathrm{P}(X^{\mathrm{s}}, L^{\mathrm{s}}) - \log \mathrm{P}(X^{\mathrm{t}}, L^{\mathrm{t}})| = \\
&|\log \mathrm{P}(L^{\mathrm{s}}|X^{\mathrm{s}})\mathrm{P}(X^{\mathrm{s}}) - \log \mathrm{P}(L^{\mathrm{t}}|X^{\mathrm{t}})\mathrm{P}(X^{\mathrm{t}})| = \\
&|\log \mathrm{P}(X^{\mathrm{s}}) - \log \mathrm{P}(X^{\mathrm{t}})| + |\log \mathrm{P}(L^{\mathrm{s}}|X^{\mathrm{s}}) - \log \mathrm{P}(L^{\mathrm{t}}|X^{\mathrm{t}})|,
\end{aligned} \tag{6}$$

where we explain the $|\log \mathrm{P}(X^{\mathrm{s}}) - \log \mathrm{P}(X^{\mathrm{t}})|$ term as the marginal discrepancy and measure it approximately by $\mathcal{L}_{\mathrm{marginal}}$ loss, and $|\log \mathrm{P}(L^{\mathrm{s}}|X^{\mathrm{s}}) - \log \mathrm{P}(L^{\mathrm{t}}|X^{\mathrm{t}})|$ as the conditional discrepancy and representing it by $\mathcal{L}_{CD}$ loss, and align by mapping X^{s}, X^{t} into a shared feature space that minimizes the distribution discrepancy.

Marginal alignment The initial target model parameters $\mathrm{O^t}$ are copied from the pretrained $\mathrm{O^s}$, and we fix $\mathrm{O_s}$ and the hypothesis C, M (including W and $\mathrm{M_{rnn}}$) in the following. The marginal discrepancy is computed on the semantic features $H^{\mathrm{s}} = \mathrm{O^s}(X^{\mathrm{s}})$, $H^{\mathrm{t}} = \mathrm{O^t}(X^{\mathrm{t}})$ by the Kernel Maximum Mean Discrepancy (KMMD) (Borgwardt et al. 2006):

$$\mathcal{L}_{\mathrm{marginal}} = ||\mathrm{E}[\phi(H^{\mathrm{s}})] - \mathrm{E}[\phi(H^{\mathrm{t}})]||^2_{\mathcal{H}}, \tag{7}$$

where E means the expectation and $\phi(H)$ is the feature mapping that $\phi : \mathcal{X} \to \mathcal{H}$. Here we avoid directly defining ϕ by using the known kernel trick: $k(h_i^{\mathrm{s}}, h_i^{\mathrm{t}}) =< \phi(h_i^{\mathrm{s}}), \phi(h_i^{\mathrm{t}}) >_{\mathcal{H}}$ where the kernel matrix is a positive-definite matrix decided by $k(h_i^{\mathrm{s}}, h_i^{\mathrm{t}}) = exp(-||h_i^{\mathrm{s}} - h_i^{\mathrm{t}}||^2/(2\sigma^2)$.

Conditional alignment For conditional alignment, ideally, O^{t} is optimized to get $\mathrm{P}(L^{\mathrm{s}}|H^{\mathrm{s}}) = \mathrm{P}(L^{\mathrm{t}}|H^{\mathrm{t}})$ while

[1]Please find the experimental validation in our supplementary material (sec. A)

training it directly is intractable. Based on the Bayesian formula, we convert the conditional distribution to:

$$\mathrm{P}(L^s|H^s) = \frac{\mathrm{P}(H^s|L^s)\mathrm{P}(L^s)}{\mathrm{P}(H^s)} \propto \mathrm{P}(H^s|L^s)\mathrm{P}(L^s) \quad (8)$$

We make an assumption that the movements across different days approximately obey the same distribution that $\mathrm{P}(Y^{\mathrm{s}}) = \mathrm{P}(Y^{\mathrm{t}})$, $\mathrm{P}(L^{\mathrm{s}}) = \mathrm{P}(L^{\mathrm{t}})$. This is reasonable as in daily use of BCI, subjects generally do not have a significant preference for a specific direction. Then we get:

$$\begin{aligned} &|\mathrm{P}(H^{\mathrm{s}}|L^{\mathrm{s}})\mathrm{P}(L^{\mathrm{s}}) - \mathrm{P}(H^{\mathrm{t}}|L^{\mathrm{t}})\mathrm{P}(L^{\mathrm{t}})| \\ &\quad \propto |\mathrm{P}(H^{\mathrm{s}}|L^{\mathrm{s}}) - \mathrm{P}(H^{\mathrm{t}}|L^{\mathrm{t}})|. \end{aligned} \quad (9)$$

To minimize the discrepancy between $\mathrm{P}(H^{\mathrm{s}}|L^{\mathrm{s}})$ and $\mathrm{P}(H^{\mathrm{t}}|L^{\mathrm{t}})$, we construct a conditional alignment loss $\mathcal{L}_{CD}$ to make the semantic features H^{s} and H^{t} that from the same category aligned to the same distribution: $\mathrm{P}(H^{\mathrm{s}}|L^{\mathrm{s}} = i) = \mathrm{P}(H^{\mathrm{t}}|L^{\mathrm{t}} = i)$. Still, we use KMMD to measure the conditional domain distance and accumulate the distance in each subspace as $\mathcal{L}_{\mathrm{CD}}$:

$$\mathcal{L}_{\mathrm{CD}} = \sum_{i=1}^{N_c} ||\mathrm{E}[\phi(H^{\mathrm{s}}_{L_{\mathrm{s}}=i})] - \mathrm{E}[\phi(H^{\mathrm{t}}_{\hat{L}_{\mathrm{t}}=i})]||^2_{\mathcal{H}}. \quad (10)$$

Considering that the target labels L^{t} are inaccessible, pseudo labels $\hat{L}^{\mathrm{t}} = \mathrm{C}(H^{\mathrm{t}})$ could be used as an alternative. With $\hat{L}^{\mathrm{t}}$, the feature space of target domain could be divided into several subspaces and conditional alignment will be achieved by aligning samples in each subspace.

Selection for high confidence samples. However, directly using C and its hyperplane on target X^{t} and trusting the divided subspace for alignment is risky and may even get a negative transfer result. Before alignment, we assess the confidence level for $\hat{L}^{\mathrm{t}}$ and remove the samples with low confidence. We assess the posterior probability inferred by $\mathrm{C}(X_{\mathrm{t}})$, sorting it and selecting the top n samples with the highest posterior probability in each subspace. By such a strategy, only the samples that are far from the decision hyperplane are trusted for alignment and the samples that are easily misclassified are ignored. As the training of $\mathrm{O_t}$, gradually more samples will be trusted.

To make sure the shared feature space H^s, H^t still works on the source data, similar to equation (3)(4) the intention classification loss $\mathcal{L}'_{\mathrm{CLF}}$ and trajectory reconstruction loss $\mathcal{L}'_{\mathrm{rec}}$ for H^{t} are added to maintain the performance of O^{t} on the source data. In summary, we align the joint distribution by fine-tuning $\mathrm{O_t}$ on the target domain by minimizing:

$$\mathcal{L}_{\mathrm{align}} = \lambda_3\mathcal{L}'_{\mathrm{reg}} + \lambda_4\mathcal{L}'_{\mathrm{CLF}} + \lambda_5\mathcal{L}_{\mathrm{marginal}} + \lambda_6\mathcal{L}_{\mathrm{CD}}. \quad (11)$$

The process can be summarized as Algorithm 1.

Algorithm 1: Getting alignment loss terms of SD-Net at per single loop.

Input:
batches of source samples: $\tilde{X^s}$, $\tilde{Y^s}$, $\tilde{L^s}$,
batches of target samples: $\tilde{X^t}$
Parameter:
the fixed $\mathrm{O^s}$, $\mathrm{C^s}$, $\mathrm{M^s}$ and the non-fixed $\mathrm{O^t}$,
total number of batches N_{b}, sequence length T, number of categories N_c
Output:
$\mathcal{L}_{\mathrm{marginal}}$, $\mathcal{L}_{\mathrm{CD}}$

Let $b = 0$, $\mathcal{L}_{\mathrm{marginal}} = 0$, $\mathcal{L}_{\mathrm{CD}} = 0$,
while $b \leq N_{\mathrm{b}}$ **do**
  $b \leftarrow b + 1$
  $\hat{L}^t_b = C^s(O^t(\tilde{X^t_b}))$ (Get pseudo labels)
  Selection for high confidence samples.
  $\mathcal{L}_{\mathrm{marginal}} \leftarrow \mathcal{L}_{\mathrm{marginal}} + \mathrm{MMD}(O^s(\tilde{X}^s_b), O^t(\tilde{X}^t_b))$
  for $i = 0 \rightarrow N_c - 1$ **do**
    $\hat{X}^s_b \leftarrow \tilde{X}^s_b[\tilde{L}^s_b = i]$ (Select by labels)
    $\hat{X}^t_b \leftarrow \tilde{X}^t_b[\hat{L}^t_b = i]$ (Select by pseudo labels)
    $\mathcal{L}_{\mathrm{CD}} \leftarrow \mathcal{L}_{\mathrm{CD}} + \mathrm{MMD}(O^s(\hat{X}^s_b), O^t(\hat{X}^t_b))$
  end for
end while

Implementing Details

Encoder-Decoder model structure. The encoder $\mathrm{O_s}$ and $\mathrm{O_t}$ share the same structure: 1) one FC layer with d hidden neurons and mapping X^{s} or X^{t} into $[B, T, d]$ size features, where B denotes the batch size; 2) one flatten layer reorganizing the features as the size of $[B, T \times d]$; 3) another FC layer mapping the features into the $[B, d]$ size semantic vectors H_{s} or H_{t}. Taking the SD features as the initial states, the autonomous dynamical system W interprets H^s, H^t into the corresponding dynamics Z_{s} or $Z_{\mathrm{t}} \in R^{B\times T\times d}$ by evolving itself from such initial points H^{s} or H^{t}. Then the decoder M_{rnn} takes Z^{s} and Z^{t} as inputs to induce the corresponding movement trajectories. The decoder is a non-linear RNN consisting of one layer and Tanh activation functions, with totally d hidden neurons. The RNN is followed by an FC layer, interpreting the RNN hidden states non-linearly into the motion signals. In addition, a classifier C is trained on H^{s}, consisting of only one FC layer and a softmax layer, and tries to predict the labels of movement targets L^{s}. The architecture is pretrained end-to-end by $\mathcal{L}_{\mathrm{pretrain}}$, optimized by Adam with the learning rate of 0.002. For other settings we select $d = 50$, $b = 10$ and $T = 60$, and $\lambda_1 = 0.1, \lambda_2 = 0.1$.

Domain adaptation strategy. We fine-tune the target domain observation module O^{t} to make the target feature distribution approach that of the source to construct a shared feature space. Here we align the joint distribution of the internal features by minimizing $\mathcal{L}_{\mathrm{align}}$ on O^{t} with other models fixed. O^{t} is trained with 10 batch size, 60 epochs, 0.002 learning rate and optimized by Adam. For $\mathcal{L}_{\mathrm{align},}$, we set $\lambda_3 = 0.1$, $\lambda_4 = 0.1$, $\lambda_5 = 0.1$, $\lambda_6 = 1$. We select the pseudo labels with the top 75% highest posterior probability and eliminate the others. Because there are fewer samples in the target domain after such selection, we randomly resample on the high-confidence target samples to get the same number of samples as the source domain, preventing the model from preferring the source domain or even overfitting on it caused by the asymmetric domain size.

R^2/acc	Linear decoder	LSTM	Dyer et al.	WGAN	Ours (not aligned)	Ours (full method)	Ours (Sup.)
D1 (C)	$0.392/0.354$	$0.689/0.402$	$-/-$	$-/0.75$	**0.845/0.867**	$-/-$	$0.845/0.867$
D2 (C)	$0.175/0.245$	$0.595/0.338$	$-/-$	$-/0.72$	$0.773/0.859$	**0.875/0.924**	$0.883/0.811$
D3 (C)	$< 0/0.262$	$0.323/0.219$	$-/-$	$-/0.54$	$0.725/0.747$	**0.792/0.771**	$0.933/0.810$
D1 (J)	$0.386/0.349$	$0.887/0.500$	$0.44/-$	$-/0.83$	**0.730/0.925**	$-/-$	$0.730/0.925$
D2 (J)	$0.340/0.261$	$0.442/0.292$	$0.42/-$	$-/0.74$	$0.43/0.75$	**0.455/0.8**	$0.763/0.975$
D3 (J)	$0.199/0.261$	$0.338/0.204$	$< 0/-$	$-/0.62$	$0.438/0.57$	**0.471/0.68**	$0.749/0.9$
D4 (J)	$0.423/0.317$	$0.431/0.227$	$< 0/-$	$-/$**0.64**	$0.433/0.49$	**0.447**$/0.56$	$0.768/0.8$

Table 1: Performance of our methods and several candidate methods on NHP dataset measured by R^2 and classification accuracy. The dataset contains three-day (D1 $\sim$ D3) recordings of CHEWIE (C) and four-day (D1 $\sim$ D4) recordings of Jango (J). All the models are only trained on the source domain including D1(C) and D2(J). The best results in each domain are bolded.

R^2/acc	Neuron death	Electrode/tissue shift	Tuning function changes	Combination
w/o alignment	**0.99**$/0.755$	$0.48/0.050$	$0.44/-0.017$	$0.92/0.123$
Alignment	$0.98/$**0.865**	**0.96/0.839**	**0.96/0.838**	**0.98/0.779**

Table 2: Our model's performance on different neural shift scenarios on the simulation dataset.

Experiments

Datasets. For experiments we use the neural data published in (Dyer et al. 2017), as well as its corresponding movement trajectories Y^{s}, Y^{t}, and target point category L^{s}, L^{t} as labels. The dataset contains neural data from two non-human primates (NHPs) named 'CHEWIE' and 'JANGO' respectively. In (Dyer et al. 2017) 'CHEWIE' performs an 8-direction centre-out reaching task and 'JANGO' applies its force of the wrist to move a cursor on the screen. The published data were recorded by the same device across different days (with a maximum span of one month). We apply our method to the cross-session data recorded on three different days of CHEWIE and four days of JANGO, and treat the first-day recording as the source domain and the other days of data as the target domain. More experimental protocols are the same as (Dyer et al. 2017). The data preprocessing details are described in supplementary materials.

Simulation neural signals. To explore our method's performance on the various factors that lead to shifts in neural data, we designed a simulation dataset based on the cosine tuning curve model (Gilja et al. 2012). The tuning function means the function relating the motor cortex signals to the hand movement direction. Concretely some neurons prefer a specific direction (named the preferred direction, PD) and perform actively when the subject moves toward this direction. The biological tuning function could be simulated by a cosine function:

$$fr_i(t) = a_{i,0} + a_{i,x}\cos\theta_{\mathrm{t}} + a_{i,y}\sin\theta_{\mathrm{t}} + \epsilon_i, \qquad (12)$$

where fr denotes the estimated firing rates, $a_{i,0}, a_{i,x}, a_{i,y}$ are the coefficients including baseline, $\mathrm{cosPD_i}$ and $\mathrm{sinPD_i}$ respectively, the ϵ means the noise term and θ_{t} means the moving direction at time t.

Referring to the scenarios summarized in (Degenhart et al. 2020), the neural instability could be divided as 1) baseline shift; 2) neuron death; 3) electrode/tissue shift;4) changes of tuning function; 5) complex situation. The baseline shift could be removed by the normalization during preprocessing, which will not be discussed in the following. The other situations could be simulated by: 2) deleting part of the input channels; 3) randomly shuffling the input channels; 4) replacing part of the neurons with some reserved neurons that have different tuning curves; 5) combing the mentioned strategies.

Compared Methods. We compared our method with several classical neural decoders used in BCI and some latest methods for unsupervised sequence data domain adaptation. 1) Traditional methods including linear regression and LSTM that supervised trained on the source domain. 2) An unsupervised alignment method in (Dyer et al. 2017), which use a brutal searching strategy to align the distribution of the low-dimensional neural representations with that of the movements by minimizing their KL-divergence; 3) an adversarial domain adaptation method WGAN (Drossos, Magron, and Virtanen 2019), which is designed for sequence data (acoustic data). 4) our method that supervised trained on each domain as a reference for the ideal neural decoder (upper bound) after alignment, whose results are shown in the rightmost column of Table 1. Here we do not compare with (Azabou et al. 2021) since it does not focus on solving cross-domain issues.

Metrics. As our model is a multi-task decoder that predicts both the motion trajectories (regression task) and the intention labels (auxiliary classification task), we adopt the classification accuracy for target prediction tasks and coefficient of determination (R^2) for motion prediction tasks separately.

Overall Performance

Regression task. We compare our method with the candidate methods on the data of $Chewie$ and $Jango$ and the results are shown in Table. 1 except WGAN (as is specially designed for classification tasks). For $(Dyeret\ al.\ 2017)$ we only use the results reported in their paper. All methods are trained on the first-day data of each subject (source) and only accessible to the neural data in the following days (target). The reconstructed trajectories with and without alignment

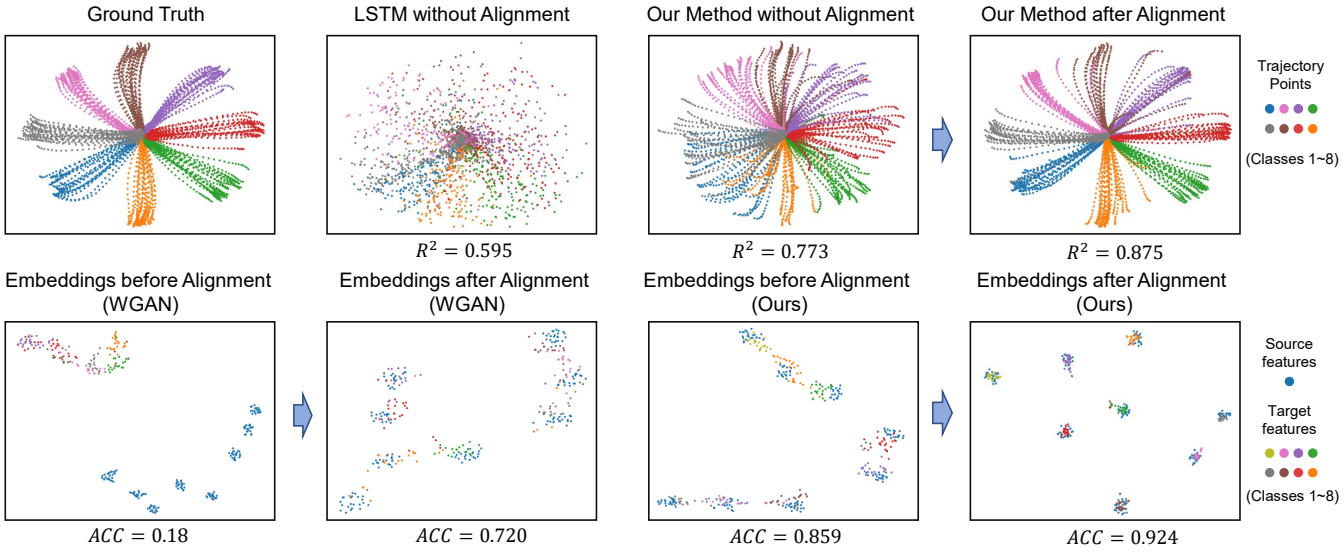

Figure 4: The UDA results of our method. On the top row, we show the trajectory points predicted by aligned/non-aligned models on the target data (D2 of CHEWIE). In the lower row, we show the distribution change of the embeddings (dimension reduction by TSNE) before and after alignment by traditional WGAN and our method respectively.

are shown in Fig. 4 and can be seen that the chaotic trajectory predicted on D2 is repaired by our method (Fig.4 right) and the fixed trajectory shows a high correlation with the ground truth $R^2 = 0.875$. The results show that when applied to neural signals, our method is able to rescue the performance degradation of BCI and performs significantly better than other methods when the neural data drifts (more results in supplementary materials).

Classification task. We also applied our method and the candidate methods to the 8-direction classification task with the same procedure as the regression task and show the classification accuracy in Table. 1. Visualized by TNSE(Van der Maaten and Hinton 2008) in Fig. 4 (right) the embeddings of the target domain data approach the source domain by the joint distribution alignment strategy and perform better (0.924 vs. 0.720 in accuracy) than that of the WGAN's (Fig. 4 left). It shows that our method corrects several labels of the misclassified trials and outperforms other methods. Because both of the movement tasks (arm moving for C but wrist moving for J) and the neural cortices of signal acquisition (arm region for C but hand region for J) are different, we get slightly different performance between the subjects.

Ablation Study

An ablation study is conducted to explore the influence of our alignment strategy. For the method without alignment, we only train our model on the source domain and directly test on the following days' data without recalibration. The cross-day results of our model trained with and without alignment are shown in Table. 1, which indicates that the alignment strategy brings improvement in our model's generality on the target domain. Note that owing to the biologically plausible modeling, our model's stability is still better than some traditional methods even without alignment.

Results on Simulation Dataset

We design a simulation dataset to simulate a variety of factors that are likely to cause domain drift in neural data. The performance of our method on these factors is shown in Table.2. The death of neurons reduces the Signal-to-Noise-Ratio (SNR) of the data such that the alignment has a limited role but still improves the BCI performance. For the situations like disrupting the sequence of neural channels or changes in the tuning functions, our model performs extremely well and is able to eliminate hypothesis failure in the target domain. To get closer to the real scenario we combine all the factors and show our method's effect in Table 2 (right). Results show that our method successfully handles a variety of distribution drifts on the neural data.

Discussion

In this work, we propose an unsupervised domain adaptation method for the unlabeled BCI recalibration scenario. Instead of matching the data points discretely, we align the neural sequences at a semantic level. Concretely we extract the intrinsic dynamics of the sequence, mapping a semantic vector (SD feature) as the dynamic trajectory's initial point and assigning them into the same manifold (the vector field of the dynamical system) eventually. Based on this we introduce a joint distribution alignment strategy to align the SD features, and the dynamic trajectories are also aligned automatically. The promising results indicate that we could model most neural data misalignment causes by adjusting the observation matrix and by aligning it, we obtain a stable neural manifold, being able to build a long-term stable BCI. Despite some existing limitations (refer to supplementary material), we have successfully achieved a stable neural manifold and developed a long-term stable BCI.

Acknowledgments

This work was supported by STI 2030 Major Projects (2021ZD0200400), Natural Science Foundation of China (U1909202, 61925603, 62276228), Key Realm R&D Program of Guangzhou (202007030005) and Lingang Laboratory (LG-QS-202202-04).

References

Ajiboye, A. B.; Willett, F. R.; Young, D. R.; Memberg, W. D.; Murphy, B. A.; Miller, J. P.; Walter, B. L.; Sweet, J. A.; Hoyen, H. A.; Keith, M. W.; et al. 2017. Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: a proof-of-concept demonstration. *The Lancet*, 389(10081): 1821–1830.

Azabou, M.; Azar, M. G.; Liu, R.; Lin, C.-H.; Johnson, E. C.; Bhaskaran-Nair, K.; Dabagia, M.; Avila-Pires, B.; Kitchell, L.; Hengen, K. B.; et al. 2021. Mine your own view: Self-supervised learning through across-sample prediction. *arXiv preprint arXiv:2102.10106*.

Barrese, J. C.; Rao, N.; Paroo, K.; Triebwasser, C.; Vargas-Irwin, C.; Franquemont, L.; and Donoghue, J. P. 2013. Failure mode analysis of silicon-based intracortical microelectrode arrays in non-human primates. *Journal of neural engineering*, 10(6): 066014.

Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; and Vaughan, J. W. 2010. A theory of learning from different domains. *Machine learning*, 79(1): 151–175.

Borgwardt, K. M.; Gretton, A.; Rasch, M. J.; Kriegel, H.-P.; Schölkopf, B.; and Smola, A. J. 2006. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14): e49–e57.

Brandman, D. M.; Hosman, T.; Saab, J.; Burkhart, M. C.; Shanahan, B. E.; Ciancibello, J. G.; Sarma, A. A.; Milstein, D. J.; Vargas-Irwin, C. E.; Franco, B.; et al. 2018. Rapid calibration of an intracortical brain–computer interface for people with tetraplegia. *Journal of neural engineering*, 15(2): 026007.

Chapin, J. K.; Moxon, K. A.; Markowitz, R. S.; and Nicolelis, M. A. 1999. Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. *Nature neuroscience*, 2(7): 664–670.

Chen, C.; Chen, Z.; Jiang, B.; and Jin, X. 2019. Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, 3296–3303.

Churchland, M. M.; Cunningham, J. P.; Kaufman, M. T.; Foster, J. D.; Nuyujukian, P.; Ryu, S. I.; and Shenoy, K. V. 2012. Neural population dynamics during reaching. *Nature*, 487(7405): 51–56.

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.

Drossos, K.; Magron, P.; and Virtanen, T. 2019. Unsupervised adversarial domain adaptation based on the wasserstein distance for acoustic scene classification. In *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 259–263. IEEE.

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.

Fang, T.; Qi, Y.; and Pan, G. 2020. Reconstructing perceptive images from brain activity by shape-semantic GAN. *Advances in Neural Information Processing Systems*, 33: 13038–13048.

Farshchian, A.; Gallego, J. A.; Cohen, J. P.; Bengio, Y.; Miller, L. E.; and Solla, S. A. 2018. Adversarial domain adaptation for stable brain-machine interfaces. *arXiv preprint arXiv:1810.00045*.

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.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. 2016. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1): 2096–2030.

Gilja, V.; Nuyujukian, P.; Chestek, C. A.; Cunningham, J. P.; Yu, B. M.; Fan, J. M.; Churchland, M. M.; Kaufman, M. T.; Kao, J. C.; Ryu, S. I.; et al. 2012. A high-performance neural prosthesis enabled by control algorithm design. *Nature neuroscience*, 15(12): 1752–1757.

Hochberg, L. R.; Bacher, D.; Jarosiewicz, B.; Masse, N. Y.; Simeral, J. D.; Vogel, J.; Haddadin, S.; Liu, J.; Cash, S. S.; Van Der Smagt, P.; et al. 2012. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature*, 485(7398): 372–375.

Hochberg, L. R.; Serruya, M. D.; Friehs, G. M.; Mukand, J. A.; Saleh, M.; Caplan, A. H.; Branner, A.; Chen, D.; Penn, R. D.; and Donoghue, J. P. 2006. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442(7099): 164–171.

Kang, G.; Jiang, L.; Yang, Y.; and Hauptmann, A. G. 2019. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4893–4902.

Kao, J. C.; Ryu, S. I.; and Shenoy, K. V. 2015. Leveraging historical knowledge of neural dynamics to rescue decoder performance as neural channels are lost:"Decoder hysteresis". In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 1061–1066. IEEE.

Long, M.; Wang, J.; Ding, G.; Sun, J.; and Yu, P. S. 2014. Transfer joint matching for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1410–1417.

Luan, L.; Liu, Y.; and Sun, H. 2022. Distilling Governing Laws and Source Input for Dynamical Systems from Videos. *arXiv preprint arXiv:2205.01314*.

Pan, G.; Li, J.-J.; Qi, Y.; Yu, H.; Zhu, J.-M.; Zheng, X.-X.; Wang, Y.-M.; and Zhang, S.-M. 2018. Rapid decoding of hand gestures in electrocorticography using recurrent neural networks. *Frontiers in neuroscience*, 12: 555.

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

Pandarinath, C.; Nuyujukian, P.; Blabe, C. H.; Sorice, B. L.; Saab, J.; Willett, F. R.; Hochberg, L. R.; Shenoy, K. V.; and Henderson, J. M. 2017. High performance communication by people with paralysis using an intracortical brain-computer interface. *Elife*, 6: e18554.

Pandarinath, C.; O'Shea, D. J.; Collins, J.; Jozefowicz, R.; Stavisky, S. D.; Kao, J. C.; Trautmann, E. M.; Kaufman, M. T.; Ryu, S. I.; Hochberg, L. R.; et al. 2018b. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10): 805–815.

Qi, Y.; Zhu, X.; Xu, K.; Ren, F.; Jiang, H.; Zhu, J.; Zhang, J.; Pan, G.; and Wang, Y. 2022. Dynamic ensemble bayesian filter for robust control of a human brain-machine interface. *IEEE Transactions on Biomedical Engineering*, 69(12): 3825–3835.

Sun, B.; Feng, J.; and Saenko, K. 2016. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Tanji, J.; and Evarts, E. V. 1976. Anticipatory activity of motor cortex neurons in relation to direction of an intended movement. *Journal of neurophysiology*, 39(5): 1062–1068.

Tanwani, A. 2021. DIRL: Domain-invariant representation learning for sim-to-real transfer. In *Conference on Robot Learning*, 1558–1571. PMLR.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Wang, Y.; Lin, K.; Qi, Y.; Lian, Q.; Feng, S.; Wu, Z.; and Pan, G. 2018. Estimating brain connectivity with varying-length time lags using a recurrent neural network. *IEEE Transactions on Biomedical Engineering*, 65(9): 1953–1963.

Wen, S.; Yin, A.; Furlanello, T.; Perich, M.; Miller, L.; and Itti, L. 2021. Rapid adaptation of brain–computer interfaces to new neuronal ensembles or participants via generative modelling. *Nature Biomedical Engineering*, 1–13.

Wodlinger, B.; Downey, J.; Tyler-Kabara, E.; Schwartz, A.; Boninger, M.; and Collinger, J. 2014. Ten-dimensional anthropomorphic arm control in a human brain- machine interface: difficulties, solutions, and limitations. *Journal of neural engineering*, 12(1): 016011.

Xie, S.; Zheng, Z.; Chen, L.; and Chen, C. 2018. Learning semantic representations for unsupervised domain adaptation. In *International conference on machine learning*, 5423–5432. PMLR.

Zhang, S.; Yuan, S.; Huang, L.; Zheng, X.; Wu, Z.; Xu, K.; and Pan, G. 2019a. Human mind control of rat cyborg's continuous locomotion with wireless brain-to-brain interface. *Scientific reports*, 9(1): 1–12.

Zhang, Y.; Nie, S.; Liu, W.; Xu, X.; Zhang, D.; and Shen, H. T. 2019b. Sequence-to-sequence domain adaptation network for robust text image recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2740–2749.

Zhao, S.; Yue, X.; Zhang, S.; Li, B.; Zhao, H.; Wu, B.; Krishna, R.; Gonzalez, J. E.; Sangiovanni-Vincentelli, A. L.; Seshia, S. A.; et al. 2020. A review of single-source deep unsupervised visual domain adaptation. *IEEE Transactions on Neural Networks and Learning Systems*.