THE UNIVERSITY OF CHICAGO

DOMAIN ADAPTATION WITH CONTRASTIVELY LEARNED SUPPORT: CASE STUDY WITH BRAIN COMPUTER INTERFACING

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
BACHELOR OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

BY
MENGZHAN LIUFU

CHICAGO, ILLINOIS
APRIL 2025

Copyright © 2025 by Mengzhan Liufu

All Rights Reserved

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to take this opportunity to acknowledge individuals who critically enabled my work and growth, both within and outside the scope of this thesis. First and foremost, I want to sincerely thank my thesis supervisor Professor Henry Hoffmann, a dedicated martial artist, virtuosic computer scientist, and an outstanding department head for the CS department in this difficult time. In the context of grappling sports, Dr. Hoffmann has been an incredibly careful, thoughtful and skilled companion to practice Jiujitsu with. As a research mentor, he always aims to train me to be an efficient problem solver, as well as an effective communicator who can get his ideas across. I'm sure I will keep benefiting from those in my upcoming career. I thank Ryien Hosseini and Mingyuan Xiang for providing invaluable feedback and help. Their expertise in machine learning and programming, and more importantly their generosity, have helped me overcome one hurdle after another. I also thank Professor Yanjing Li, during my discussion with whom the idea of self-calibrating BCI first emerged. Outside the thesis, I'd like to thank Professor Jai Yu and Professor Bozhi Tian for offering me wonderful research experiences as an undergraduate. My opportunity to continue research in graduate school is also critically enabled by their support. Jai and Bozhi demonstrate to me the best qualities of top-rate scientists, and I aspire to be as passionate, consistent, meticulous and creative as they always are. I thank my parents (Hui Zeng and Guoqing Liufu) for supporting my education; although remote, their unconditional love and support is always with me by heart. I thank Audrey Kaye for being dear to me, cheerful, encouraging and caring on the best and worst days alike. My thanks also go to other friends - including but not limited to Yuanyuan (Michelle) Hu, Bojun (David) Feng, Zichong (Terry) Yuan and Chuyun (Joyce) Zhang - for making my college life colorful.

ABSTRACT

Brain-Computer Interfaces (BCIs) promise to enable seamless communication between the human brain and external computing devices. However, current deep learning-based BCI models struggle to generalize across individuals due to subject-specific neural variability. While supervised fine-tuning can recalibrate models, it demands an impractical amount of labeled data post-deployment. In this thesis, I propose, implement and evaluate a self-calibrating BCI framework designed for unsupervised and semi-supervised domain adaptation without access to source data. The framework introduces a contrastively trained support encoder that extracts subject-specific, task-irrelevant information from neural signals. Task embeddings are transformed by attending over the support embeddings or class prototypes to achieve an individual-invariant representation space. This design would enable a common classifier to operate across subjects with minimal to no labeled target data. I evaluate the proposed approach on two canonical BCI tasks — motor imagery decoding and sleep staging — and benchmark fine tuning and state-of-the-art domain adaptation methods. Although the proposed method does not yet consistently surpass baselines, my evaluation of the baseline methods furthers understanding of their limitations, and analysis of intermediate embeddings shows promising structure. Future improvements will potentially enable truly unsupervised, source-free calibration for BCIs.

CHAPTER 1

INTRODUCTION

Brain computer interfaces (BCIs) are real-time systems that enable direct communication between human brains and external computing devices. They aim to provide online, instant feedback to user's neural activity, like outputting text based on user's unspoken words and controlling prosthetics based on user's motor intentions. They typically use learning models to map brain activity to human interpretable, actionable concepts such as movements and texts (Fig. 1.1).

In early BCI setups, a neural recording device interfaces with the brain and streams neural data, and then a wired or wireless connection sends data to a nearby computer to process data with deep learning-based BCI model. More recently, to achieve autonomous BCI operation in daily life setting, highly integrated brain implantable devices have been invented to avoid frequent, long-range data transfers. These devices offer both neural data streaming and limited computing resources to support on-device data processing, and compressed version of BCI models are deployed on the device. This setup enables untethered BCI on edge. Although many deep learning-based BCIs have been implemented, it remains challenging to calibrate BCIs across time and individuals. It has been shown that BCI models fail to generalize well across different people due to individual differences in neural patterns [1].). They even fail to perform well across time within the same individual because neural data is highly non-stationary. While supervised strategies such as model retraining, fine tuning and supervised transfer learning have been applied to address this issue, these supervised methods typically require a significant amount of new, labeled data to calibrate effectively (Fig. 1.1). Data labels are impossible to access when the BCI operates autonomously on an edge device such as a brain implant, with little to no supervision. On the other hand, existing unsupervised strategies often achieve calibration by aligning new data and pre-existing training data. Such approach is also infeasible for edge-deployed BCIs because the

size of training dataset is likely larger than the storage space offered by these edge devices.

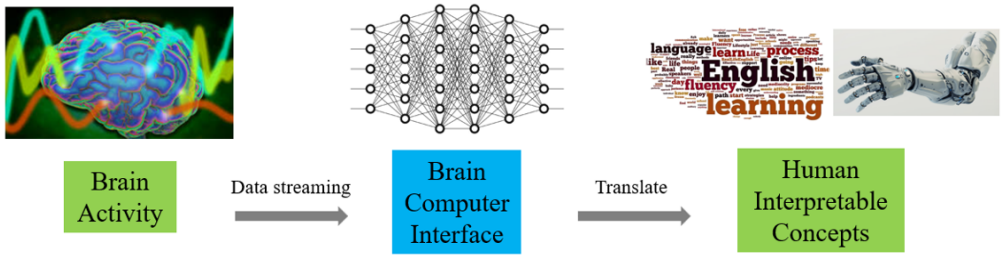

Figure 1.1: Brain computer interfaces (BCIs) enables communication between nervous system and external computing devices. BCIs stream raw neural data in real time and leverage deep learning methods to translate neural data to human interpretable concepts.

The aim of this thesis is to develop a self-calibrating BCI framework potentially suitable for edge deployment. The framework I explored consists of a contrastively trained support encoder that extracts task-irrelevant information from each individual in the form of support embeddings, and a task encoder which produces task-specific embeddings for each input data sample. I explored using an attention module to affine transform individual-specific task embeddings by attending over the support embeddings of the individual. The transformed task embeddings are supposed to live in an individual-agnostic feature space, ready for a common task classifier (see Methods section for more detail). This calibration framework is designed to enable adaptation in an unsupervised / semi-supervised and source-free manner. This work will potentially bring BCI technologies closer to everyday use, facilitating personalized and precise neural healthcare. The BCI calibration setting I chose to focus on is an example of the general problem of Unsupervised Domain Adaptation (UDA), which spans various domains including vision, natural language processing and language generation [2].

CHAPTER 2

RELATED WORK

The notion of self-calibrating brain computer interfaces involves two parts, self brain computer interfacing (BCI), and self-calibration or unsupervised domain adaptation (UDA). The framework I designed and explored involves contrastive learning (CL), attention mechanism and meta learning. In this chapter, I will introduce each relevant concept and the latest advances there.

2.1 Brain Computer Interfacing

Brain-Computer Interfaces (BCIs) enable direct communication between the brain and external devices. This paradigm enables opportunities such as motor function restoration, cognition enhancement, and machine interfacing. Powered by advancements in both neural interface hardware and deep learning algorithms, BCIs have evolved significantly in capability and its range of applications.

Modern BCIs span a wide range of applications. In motor restoration, the late Krishna Shenoy's work with intracortical BCIs allowed paralyzed individuals to control computer cursors or robotic limbs by decoding motor intent from the motor cortex in real time [3]. In the domain of communication, Edward Chang's team developed a BCI that reconstructs spoken language from neural activity recorded in speech-related cortical areas, restoring speech capability for individuals with anarthria [4]. Beyond clinical use, more unconventional BCIs are emerging. Rather than passively listening to the brain, innovative BCI paradigms started "writing back" to the brain to enhance our conscious and unconscious experiences. For instance, Elemind's non-invasive headband modulates brain rhythms via transcranial electrical stimulation to improve sleep onset and memory consolidation [5].

BCIs operate across a spectrum of invasiveness and material interfaces, reflecting a trade-off

between signal fidelity, spatial resolution, and usability. Non-invasive systems such as EEG caps remain widely used in research and consumer technologies for their safety and ease of use. These caps measure scalp electrical potentials, typically using rigid electrodes. Electromyographic (EMG) devices like Meta's wristband captures muscle activity as a proxy for neural intent to enable gesture recognition and motor control [6]. Despite their lower spatial resolution, such interfaces are promising for real-world deployment due to their non-intrusive nature. Invasive systems such Utah arrays and ECoG grids need to be surgically implanted and live chronically on cortical surface or inside deep brain tissue until removal. Significantly closer to the source of neural activity, such invasive systems offer high-fidelity recordings and have powered many landmark demonstrations in motor and speech BCIs [3, 4, 7]. However, they often face challenges related to long-term stability, biocompatibility, and surgical risk. To address these issues, the field of flexible and soft bioelectronics offered conformable neural interfaces that match the mechanical properties of brain tissue. These interfaces reduce foreign body response and improve long-term signal stability. For instance, the work by Kim et al. introduced ultrathin, flexible electronic arrays capable of chronic cortical recording with minimal immune response [8]. Similarly, Lu et al. demonstrated injectable mesh electronics that integrate seamlessly with neural tissue, enabling stable electrophysiological recordings over months [9]. These innovations are paving the way toward chronic, high-resolution, and minimally invasive BCIs, especially relevant for long-term brain health monitoring and therapeutics.

The rise of deep learning has transformed BCI research by enabling robust, scalable decoding of neural signals. Prior to deep learning, Kalman filters and LSTMs are used to map intracortical spiking activity to arm trajectories for motor decoding [3]. Then, Recurrent neural networks (RNNs) and convolutional neural networks (CNNs) are widely used to extract spatiotemporal patterns from neural time series. For example, in speech BCIs, Anumanchipalli et al. used a sequence-to-sequence deep learning model to map ECoG signals to synthetic

speech [10]. In non-invasive BCIs, CNNs have enabled end-to-end decoding of motor imagery and P300 signals from EEG data [11]. The more recent transformer architecture has also showed its impact in BCI. Willett et al. used transformer-based architectures to decode attempted handwriting from neural signals, enabling text communication at 90 characters per minute in a paralyzed individual [12]. Recent advances in generative deep learning have demonstrated their relevance to brain interfacing as well, especially to externalize visual imageries in the brain. Diffusion models have been used to reconstruct photorealistic images from fMRI signals, guided by image-text priors from Stable Diffusion [13]. Chen et al. proposed MinD-Video, a transformer-based model that decodes continuous visual experience from brain signals, generating temporally coherent video-like reconstructions [14]. Overall, neural signals are high-dimensional, non-stationary and noisy; deep learning methods offer solutions that have comparable sophistication to the problem itself.

2.2 Unsupervised Domain Adaptation

In real-world machine learning applications, models trained on one dataset (source domain) often perform poorly when deployed on data from a different distribution (target domain). Unsupervised Domain Adaptation (UDA) addresses this challenge by leveraging labeled source data and unlabeled target data to bridge the domain gap. UDA techniques typically aim to learn domain-invariant representations that generalize across both domains. Less commonly explored, some have taken the generative approach to adapt task model based on input data.

A central strategy in UDA is to align the distributions of source and target features in a shared latent space. Earliest approaches involve statistical distance metrics, such as the Maximum Mean Discrepancy (MMD), which penalizes differences between the empirical distributions of source and target embeddings. MMD-based approaches are non-parametric and have been successfully integrated into deep networks (e.g., DDC, DAN) by adding an

MMD loss to intermediate layers [15].

Adversarial training can also align source and target domains. Inspired by GANs, these methods train a domain discriminator to distinguish between source and target features, while the feature encoder tries to fool the discriminator into thinking both domains are indistinguishable. This min-max game encourages the model to learn domain-invariant features. Domain-Adversarial Neural Networks (DANN) exemplify this approach and have become a foundational method in the field [16].

Contrastive learning (CL) also enables domain adaptation. CL is originally proposed to train robust visual feature extractors without supervision by making different views of the same image (or same class of images) represented similarly and different images represented as dissimilar as possible [17]. When adapted for UDA, CL treats source and target examples are treated as different views of the same underlying distribution and learn embeddings where same-class or similar features are pulled together while dissimilar ones are pushed apart. Some works combine contrastive losses with adversarial alignment to improve both class separation and domain invariance [18]. When no true labels are available from the target domain, as is the case in UDA, some have given target samples pseudo-labels and used contrastive learning to align the source and target distributions [19].

Beyond alignment, other strategies include self-training, domain-specific batch normalization and entropy minimization. Self-training methods use the model's predictions on the target domain to generate pseudo-labels, which are then used to iteratively refine the model. Notable examples include Mean Teacher and FixMatch-style approaches adapted for domain adaptation [20]. Domain-specific batch normalization uses domain-specific statistics in batch norm layers to isolate and adapt domain-specific shifts while sharing the rest of the model's parameters [21]. Entropy minimization method focuses on minimizing the entropy of the model's predictions on the target domain, the network is encouraged to make more confident (and ideally correct) predictions [22].

Many UDA methods rely on aligning source and target domain distributions to find a domain-invariant feature space. However, in reality when adaptation is needed, only a trained source model is available for adaptation, and source data is not accessible for privacy or practical reasons. Source-free domain adaptation (SFDA) addresses this very limit, achieving adaptation without access to source data. Adaptation is achieved by either (a) refining the decision boundary using target-domain data alone (e.g., via entropy minimization or pseudo-labeling), or (b) generating prototypes or clustering targets in the feature space to realign categories. Approaches like SHOT (Source Hypothesis Transfer) perform self-supervised clustering on target features while preserving the classifier structure from the source model [23]. Specifically for time series processing, Ragab et.al uses a temporal imputation task to capture inherent temporal dynamics and ensure its consistency between source and target domains, assuming that they share the same inherent dynamics [24].

All methods mentioned (except for batch normalization) focuses on adjusting the data samples themselves, so one common task model can perform equally well across domains. Domain-specific batch normalization is an early attempt to dynamically adapt the parameters of the model itself based on the data characteristics of each domain. In natural language processing, Volk et.al uses a hypernetwork to generate the weights of the task classifier based on the unique signature of each input sample [25, 26].

2.3 Meta-learning

Meta-learning is often described as "learning to learn". This paradigm aims to train models to rapidly adapt to new tasks with minimal data. This is particularly valuable for applications where collecting large, labeled datasets is impractical. Meta-learning enables models to generalize across domains and adapt efficiently to new domains with few or no labeled samples.

In the context of UDA, meta-learning can be deployed in various ways. One common ap-

proach is model-agnostic meta-learning (MAML), where a model is trained over multiple simulated tasks (e.g., classification across different EEG subjects) to find an initialization of parameters that can be quickly fine-tuned on a new target domain [27]. Duan et al. applied this concept to EEG-based BCI by dividing data from multiple source subjects into meta-train and meta-test groups, effectively simulating the domain shift during training [1]. While effective in adaptation, MAML-based methods can suffer from overfitting, especially when the base network lacks sufficient representational power.

An alternative strategy is metric-based meta-learning, such as the relation network framework [28]. This method learns a similarity function between a set of labeled support samples and unlabeled query samples by constructing class-representative embeddings. An attention mechanism identifies the most relevant class features from the support set, and the relation module evaluates the similarity between these class representations and the query embeddings. This approach has shown promising results in few-shot BCI settings, improving performance even with as few as 5 labeled samples per class [1]. Meta-learning in this context aims to create a subject-invariant feature space, where the learned representation is robust to inter-subject variability and can be efficiently adapted to unseen users. While these methods do not minimize domain discrepancy as explicitly as adversarial approaches, they implicitly promote generalization through episodic training across domains [29].

CHAPTER 3

METHODS

3.1 Problem Definition

For this thesis, I consider the discriminative BCI problems, where neural signals are classified into finite number of different categories, as is the case in motor imagery decoding and sleep stage classification [3, 30]. Therefore, without loss of generality, we consider a classification task for which we aim to perform UDA of time series. We have two distributions over the time series from the source domains $D_{\mathcal{S}} = \{D_i\}_{i=1}^{N_{\mathcal{S}}}$ where $N_{\mathcal{S}}$ is the number of source subjects, and the target domain $D_{\mathcal{T}}$ which I assume to only include data from one subject. We have **labeled** *i.i.d* samples from the source domain given by $\mathcal{S} = \{(x_i^{\mathcal{S}}, y_i^{\mathcal{S}})\}_{i=1}^{n_{\mathcal{S}}} \sim D_{\mathcal{S}}$ where $n_{\mathcal{S}}$ is the total number of samples from $D_{\mathcal{S}}$. From the target domain $D_{\mathcal{T}}$, when adaptation is fully unsupervised, we have **unlabeled** *i.i.d* samples given by $\mathcal{T} = \{x_i^{\mathcal{T}}\}_{i=1}^{n_{\mathcal{T}}} \sim D_{\mathcal{T}}$ where $n_{\mathcal{T}}$ is the number *i.i.d* samples available from the target domain. In many BCI settings, while it is impractical to collect a large labeled dataset, it is possible to collect a small number of labeled samples for calibration (think of when a user puts on a VR headset, the headset usually requires the user to make a few standard gestures for calibration purposes). This motivates the few-shot adaptation scenario, in which we assume access to a small labeled support set $\tilde{\mathcal{T}} = \{(x_i^{\mathcal{T}}, y_i^{\mathcal{T}})\}_{i=1}^{k \cdot C} \sim D_{\mathcal{T}}$ where C is the number of label classes and k is the number of labeled samples available per class (i.e the k-shot adaptation setting). I assume $\tilde{\mathcal{T}}$ are drawn *i.i.d* from $\mathcal{T}$ and thus constitute a representative subset of the domain although $|\tilde{\mathcal{T}}| \ll |\mathcal{T}|$.

I consider multidimensional time series, as neural recordings usually come in multiple channels; commercial EEG caps have 32 to 64 channels, and state-of-the-art neural recording probes in animal research have almost 400 channels and thousands of independent recording sites (as a side note, latest effort in neuroscience attempts whole-brain recording using

$\sim 6 \times 10^3$ recording sites, the sheer dimensionality of which poses non-trivial challenges). Each multidimensional sample x_i (either from the source or target domain) is denoted by $x_i = \{x_{it}\}_{t=1}^{T} \in R^{M \times T}$ where T is the number of uniform timestamps and M is the number of recording channels.

The goal is to build a classification network that (1) gets trained using labeled source samples $\mathcal{S}$, and (2) generalizes well over target samples $\mathcal{T}$. Target samples $\mathcal{T}$ are not used during training nor inference-time adaptation, except in the few-shot adaptation setting, $\tilde{\mathcal{T}}$ can be used for adaptation.

3.2 Baselines

It has been commonly assumed that supervised fine tuning requires too many labeled samples to be practical; however, at least in the BCI setting, it is unclear exactly how many labeled samples are needed to achieve what post-adaptation accuracy. Therefore, I performed an experiment to determine the sample efficiency of supervised fine tuning approach for adaptation. This knowledge enables comparison of labeled sample efficiency between fine tuning and semi-supervised, weakly supervised, and few-shot adaptation methods.

As outlined in Related Works, many attempts at the UDA problem have been made. In this thesis, I evaluate two state-of-the-art unsupervised baselines, CLUDA and MAPU [19, 24]. CLUDA achieves UDA by aligning source domain distribution $\mathcal{S}$ and target domain distribution $\mathcal{T}$ at adaptation time, so CLUDA is **not** source free. MAPU is source free; it uses a temporal imputer network to capture inherent temporal dynamics shared between $\mathcal{S}$ and $\mathcal{T}$, and the imputer network replaces the need of source data during adaptation.

3.3 Proposed Framework

In this section, I describe the proposed framework for self-calibrating BCI. I will first provide an overview then describe how I (1) contrastively train a support encoder to represent individual-specific, task-irrelevant information, (2) calculate class prototypes from the support embeddings of the labeled target support set $\tilde{\mathcal{T}}$ (when available), and (3) use attention to project task embeddings into a generalized subspace by attending over support embeddings (when fully unsupervised) or class prototypes (when labeled support set is available).

3.3.1 Architecture

The proposed architecture for self-calibrating BCI is shown in Figure 3.3. The **support encoder** $F_{\text{support}}(\cdot)$ is trained using contrastive learning to represent data samples from the same individual (across label classes) similarly and samples from different individuals dissimilarly. It takes time series data samples x^s and x^t from both domains and outputs support embeddings $z^s_{\text{support}} = F_{\text{support}}(x^s)$ and $z^t_{\text{support}} = F_{\text{support}}(x^t)$. The **task encoder** $F_{\text{task}(\cdot)}$ produces embeddings z^s_{task} and z^t_{task} containing task-specific information. F_{task} may or may not have the same structure as F_{support}; in principle, since F_{support} is task agnostic, F_{support} should be generalizable across different BCI applications and F_{task} needs to be tailored to the task at hand. During both training and inference time adaptation, I use an **attention** $\mathcal{A}(\cdot)$ to project z^s_{task} and z^t_{task} to $\hat{z}^s_{\text{task}}$ and $\hat{z}^t_{\text{task}}$ by attending over z^s_{support} and z^t_{support}. z^s_{task} and z^t_{task} are transformed task embeddings that live in a subject-invariant space. Finally, a **classifier network** $\mathcal{C}(\cdot)$ is trained to predict the labels y^s, y^t. As mentioned above, the target domain dataset $\mathcal{T} = \{(x^t_i, y^t_i)\}$ is not used during training.

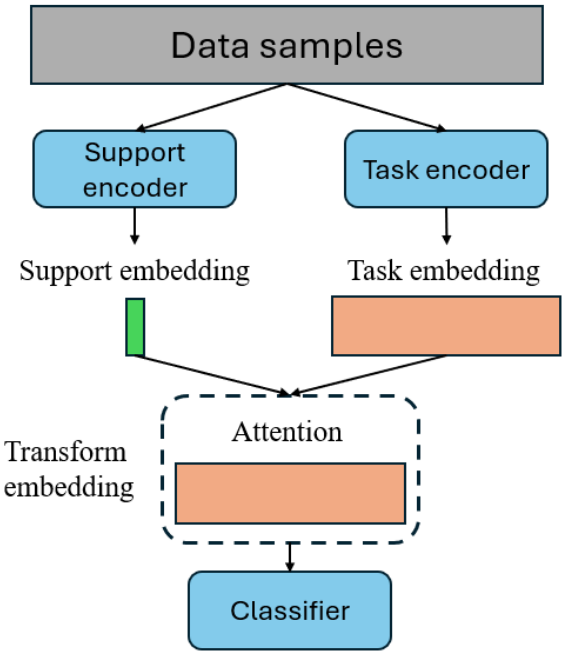

Figure 3.1: $z^{\mathcal{T}}_{\text{task}}$ is projected to $\hat{z}^{\mathcal{T}}_{\text{task}}$ by attending over $z^{\mathcal{T}}_{\text{support}}$

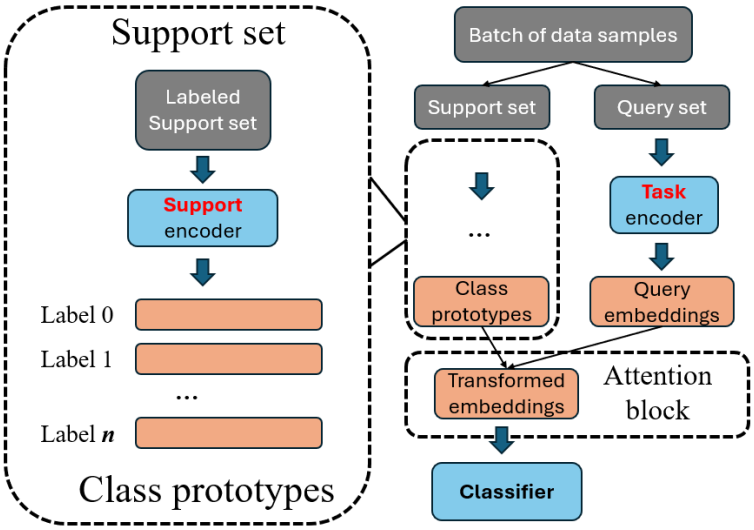

Figure 3.2: When a labeled support set $\tilde{\mathcal{T}}$ is available from $D_{\mathcal{T}}$, class prototypes $\tilde{z}^{\mathcal{T}}_{\text{support}}$ are calculated from $\tilde{T}$ and used to project $z^{\mathcal{T}}_{\text{task}}$ to $\hat{z}^{\mathcal{T}}_{\text{task}}$.

Figure 3.3: The Proposed architecture consists of support encoder $F_{\text{support}}(\cdot)$, task encoder $F_{\text{task}(\cdot)}$, an attention block and a classifier $\mathcal{C}(\cdot)$. Class prototypes are used to transform task embeddings z_{task} .

3.3.2 Contrastive Learning for Support Encoder

I trained a support encoder F_{support} to produce embeddings that capture subject-specific patterns in neural activity. The goal was to group samples from the same subject together and push apart samples from different subjects, regardless of task label (Fig. 3.4). These subject-discriminative embeddings were later used as part of the attention-based adaptation mechanism.

At each iteration, we assembled a minibatch by sampling n examples from each subject $s \in \{1, 2 \cdots, N_{\mathcal{S}}\}$, forming a batch of size $n \cdot N_{\mathcal{S}}$. Let $\mathbf{z}_i = F_{\text{support}}(\mathbf{x}_i) \in \mathbb{R}^d$ denote the embedding for the i-th input $\mathbf{x}_i$ where d is the embedding dimension. I used a contrastive loss to bring together embeddings from the same subject and repel those from different subjects. For a given anchor $\mathbf{z}_i$, the positive sample $\mathbf{z}_{i^+}$ is drawn from the same subject as $\mathbf{x}_i$ while negatives come from different subjects. I used the NT-Xent (normalized temperature-scaled cross-entropy) loss:

$$\mathcal{L}_{\text{contrastive}} = \sum_{i=1}^{N} -\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_{i^+})/\tau)}{\sum_{j \neq i} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)} \tag{3.1}$$

where:

$$\text{sim}(\mathbf{z}_i, \mathbf{z}_j) = \frac{\mathbf{z}_i^\top \mathbf{z}_j}{\|\mathbf{z}_i\| \, \|\mathbf{z}_j\|} \tag{3.2}$$

and τ is a temperature hyperparameter.

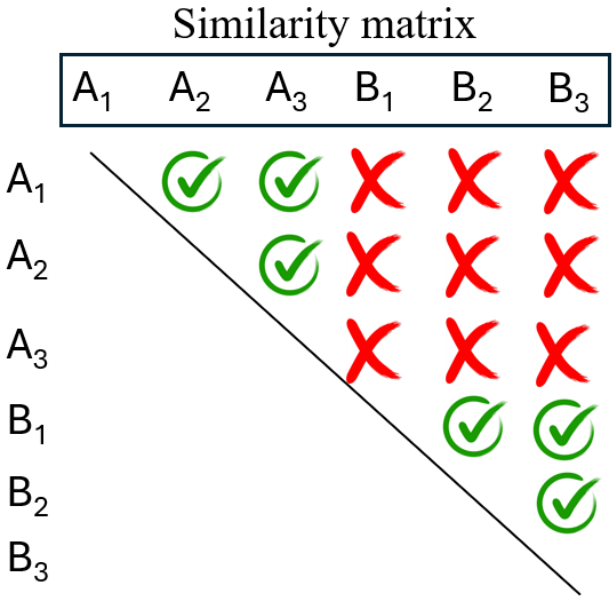

Figure 3.4: Positive samples are drawn from the same subject as the anchor sample, and negative samples come from other subjects.

3.3.3 Attention for Transforming Task Embeddings

To improve subject-invariant generalization, I use an attention block to transform task embeddings using a support set from $\mathcal{T}$. The goal of this transformation is to project embeddings from $\mathcal{T}$ into a shared subspace aligned with the source distribution $\mathcal{S}$. We consider two settings, when the labeled support set $\tilde{\mathcal{T}}$ is available versus when it is not available (fully unsupervised).

In the unsupervised setting, each task embedding attends directly over the set of support embeddings. Let $\mathbf{z}^{\text{task}} \in \mathbb{R}^{B\times T\times D}$, $\mathbf{z}^{\text{support}} \in \mathbb{R}^{B\times T\times D}$ where B is the batch size, T is the time dimension and D the feature dimension. The transformation is done independently at each timestep t to properly handle temporal dependency. At each time step $t \in \{1, 2, \cdots, T\}$, I computed attention between each task embedding $\mathbf{z}_t^{\text{task}}$ and all support embeddings at that

time step $\{\mathbf{z}_t^{\text{support}}(j)\}_{j=1}^{B}$. I first calculated the queries (Q), keys (K) and values (V):

$$\begin{aligned} \mathbf{q}_t &= W_q \cdot \mathbf{z}_t^{\text{task}} \in \mathbb{R}^d \\ \mathbf{k}_t^{(j)} &= W_k \cdot \mathbf{z}_t^{\text{support}}(j) \in \mathbb{R}^d \\ \mathbf{v}_t^{(j)} &= W_v \cdot \mathbf{z}_t^{\text{support}}(j) \in \mathbb{R}^d \end{aligned} \tag{3.3}$$

I then calculated the attention scores by calculating the similarity between queries $\mathbf{q}_t$ and each key $\mathbf{k}_t^{(j)}$

$$\alpha_t^{(j)} = \frac{\exp(\mathbf{q}_t^\top \mathbf{k}_t^{(j)}/\sqrt{d})}{\sum_{j'} \exp(\mathbf{q}_t^\top \mathbf{k}_t^{(j')}/\sqrt{d})} \tag{3.4}$$

Finally, I calculated the attended embeddings and combined it with the original task embedding as residual

$$\hat{\mathbf{z}}_t^{\text{task}} = \mathbf{z}_t^{\text{task}} + \sum_{j=1}^{B} \alpha_t^{(j)} \mathbf{v}_t^{(j)} \tag{3.5}$$

When a labeled support set $\mathcal{T}$ is available, I calculated class prototypes and allowed each task embedding to attend over them. Let $\mathcal{C}$ be the number of label classes, the class prototype at time t for class c is

$$\mathbf{p}_t^{(c)} = \frac{1}{|\tilde{\mathcal{T}}_{\mathcal{C}}|} \sum_{j \in \tilde{\mathcal{T}}_{\mathcal{C}}} \mathbf{z}_t^{\text{support}}(j) \tag{3.6}$$

where $\tilde{\mathcal{T}}_{\mathcal{C}}$ is the subset of $\mathcal{C}$ that belongs to label class $\mathcal{C}$. I then let each task embedding at

each time t attend over each class prototype at time t as follows

$$\begin{aligned}
\mathbf{q}_t &= W_q \mathbf{z}_t^{\text{task}} \\
\mathbf{k}_t^{(c)} &= W_k \mathbf{p}_t^{(c)} \\
\mathbf{v}_t^{(c)} &= W_v \mathbf{p}_t^{(c)} \\
\alpha_t^{(c)} &= \frac{\exp(\mathbf{q}_t^\top \mathbf{k}_t^{(c)} / \sqrt{d})}{\sum_{c'=1}^{C} \exp(\mathbf{q}_t^\top \mathbf{k}_t^{(c')} / \sqrt{d})} \\
\hat{\mathbf{z}}_t^{\text{task}} &= \mathbf{z}_t^{\text{task}} + \sum_{c=1}^{B} \alpha_t^{(c)} \mathbf{v}_t^{(c)}
\end{aligned} \tag{3.7}$$

The key difference between this and the unsupervised version is that, instead of attending over every support embedding $\mathbf{z}_t^{\text{support}}(j)$ from $\tilde{\mathcal{T}}$, attending over known class prototypes more structured and concrete information.

3.3.4 Meta-learning and Episodic Training

To train the model for cross-subject generalization, I used a meta-learning approach inspired by few-shot episodic training. Each subject is treated as a separate "task," and the model is trained to generalize from a small support set to a query set within that subject. This episodic structure mimics the test-time scenario, where the model must adapt to a new subject given only a small number of labeled support examples. The training procedure operates on a list of per-subject data loaders, ensuring each episode samples data from a single individual. A pseudocode of the episodic training code is shown below.

Algorithm 1 Episodic Training for Meta-Learning

Input: Per-subject dataloader list: $\{\mathcal{L}_1, \ldots, \mathcal{L}_S\}$
for each training epoch **do**
    for each subject $s = 1, \ldots, N_{\mathcal{S}}$ **do**
        $\mathcal{B}_s \leftarrow$ next batch from dataloader $\mathcal{L}_s$
        Split $\mathcal{B}_s$ into support set $\mathcal{S}_s$ and query set $\mathcal{Q}_s$
        $\hat{y}_s \leftarrow \text{Model}(\mathcal{S}_s, \mathcal{Q}_s)$
        Compute loss $\mathcal{L}_s = \text{Loss}(\hat{y}_s, y_s^{\text{query}})$
        Backpropagate loss and update model parameters
    end for
end for

3.4 Evaluation

I evaluated the baselines and the proposed framework with two canonical datasets, a motor imagery decoding dataset called *Schirrmeister's* ($N = 14, N = 13440, M = 128$, sampling rate = 500 Hz) [31] and a sleep staging dataset called *SleepPhysioNet* ($N = 197, M = 32$, sampling frequency = 100 Hz) [32]. In each 4 second trial of motor decoding dataset, the participant imagines making one of four movements (moving left arm, right arm, leg or tongue) while staying physically still. The goal of motor decoding is to classify each trial of neural data as one of the four movement imageries. The *SleepPhysioNet* dataset consists of 197 whole-night PolySomnoGraphic sleep recordings, and the goal of sleep staging is to classify each window of neural data during sleep into one of four sleep stages or wakefulness. I used ShallowFBCSPNet [31] and the SleepStagerNet from Chambon et.al [30] to perform the motor decoding and sleep staging tasks. While there are alternative deep learning models proposed for each of the tasks, the focus of the thesis is UDA methods which should be generalizable across different task models.

For each dataset, I evaluated each adaptation method using the leave-one-out approach. Namely, for each dataset with N subjects, I trained the adaptation method with data from $N-1$ subjects as the source domain $D_{\mathcal{S}}$, and at inference time the method adapted the model to the left-out subject i.e target domain $D_{\mathcal{T}}$. For the fine tuning baseline, I fine tuned the trained model for 30 epochs with varying amounts of labeled target samples from 4 samples (one per label class) to the entire $\mathcal{T}$. For unsupervised methods, I feed the entire $\mathcal{T}$ through the adaptation pipeline.

CHAPTER 4

RESULTS

4.1 Labeled Sample Efficiency with Fine Tuning

Results of the labeled sample efficiency experiment (fine-tuning) is shown in Figure 4.3. We see that for both motor decoding (ShallowFBCSPNet) and the Sleep Stager, fine tuning a trained model starts off at significantly higher accuracy than a randomly initialized and untrained network. For both tasks, fine tuning a trained model helps achieve higher final accuracy than training a model from scratch. That may be because a model pre-trained on multiple other subjects better captures task-relevant information that generalizes across subjects. For ShallowFBCSPNet and the motor decoding task, fine tuning requires approximately 30 minutes worth of labeled data to reach optimal accuracy, and the number is roughly 200 minutes for sleep staging. This amount is indeed impractical to collect after a BCI model is deployed. Therefore, we confirm that effective UDA method is necessary for efficient calibration of edge-depliyed BCIs.

4.2 Model Performance after Adaptation by Unsupervised Baseline Methods

The performance of the two unsupervised baselines, CLUDA and MAPU, are shown in Figure 4.6 and 4.7 respectively. CLUDA is not a source-free approach, so we tested whether there is significant difference in adaptation effects when the source domain dataset $\mathcal{S}$ includes one versus multiple subjects. If adapting from one subject offers similar or better adaptation than multiple subjects, the non source-free nature of CLUDA becomes less disadvantageous. When $\mathcal{S}$ includes data from one subject only, CLUDA fails to improve model performance on $\mathcal{T}$ by aligning the two domains 4.4. On the other hand, when $\mathcal{S}$ includes data from multiple

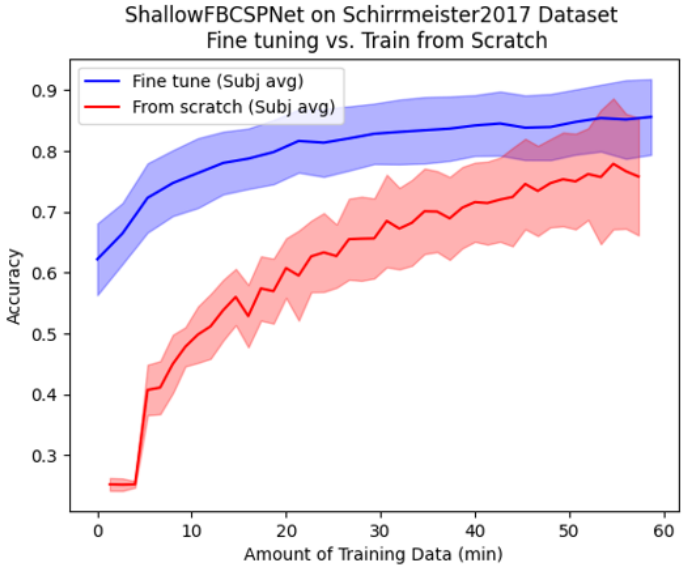

Figure 4.1: Final classification accuracy for the motor decoding task reached after 30 epochs of training from scratch (red) or fine tuning (blue) using varying amount of labeled samples up to 60 minutes.

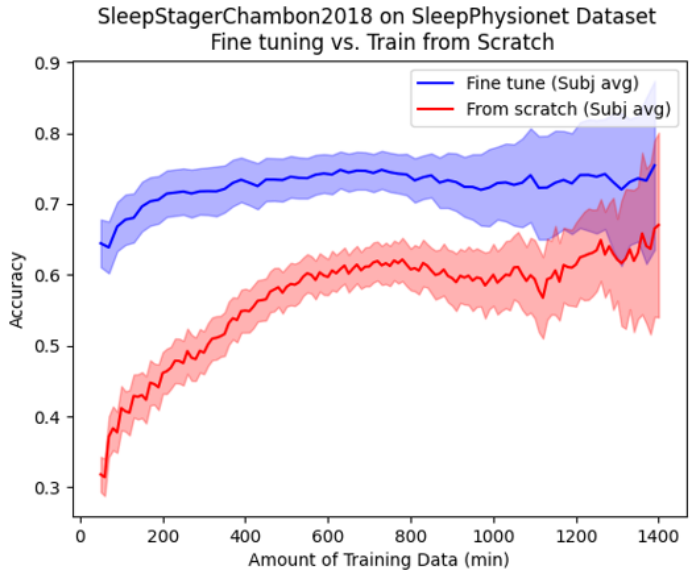

Figure 4.2: Final classification accuracy over target domain dataset $\mathcal{T}$ for the sleep staging task reached after 30 epochs of training from scratch (red) or fine tuning (blue) using varying amount of labeled samples up to 1400 minutes.

Figure 4.3: Fine tuning is more sample efficient than training from scratch but still requires significant labeled samples.

$(N-1)$ subjects, performance in $\mathcal{T}$ improves as alignment goes and reaches optimal accuracy after 20 epochs 4.5. Notably, the optimal accuracy in $\mathcal{T}$ after adaptation is higher than the highest accuracy in $\mathcal{S}$. This comparison suggests that, pre-training models on a single subject may not lead the models to converge to a subject-generalizable space. That would make the pre-trained models intrinsically less open to adaptation to new data.

We made a similar observation with MAPU (Fig. 4.7). With MAPU, adaptation with $N_{\mathcal{S}} = 1$ fails to improve accuracy in $\mathcal{T}$ to above chance level, while adaptation from a multi-subject source dataset $\mathcal{S}$ achieves significanly higher post-adaptation accuracy (approximately 70%). Both fail to outperform the fine tuning baseline. One noticeable shortcoming of MAPU is that its adaptation curve plateaus quickly and stops responding to more input data from $\mathcal{T}$ (Fig. 4.7). One possibility is that the small size of the temporal imputer network. While this property helps achieve light-weight, source-free UDA, it may limit the expressiveness of the imputer network and the amount of temporal dynamics the imputer captures. The original proposition of MAPU was tested using ECG signals, which come with less noise and lower dimensionality than neural signals.

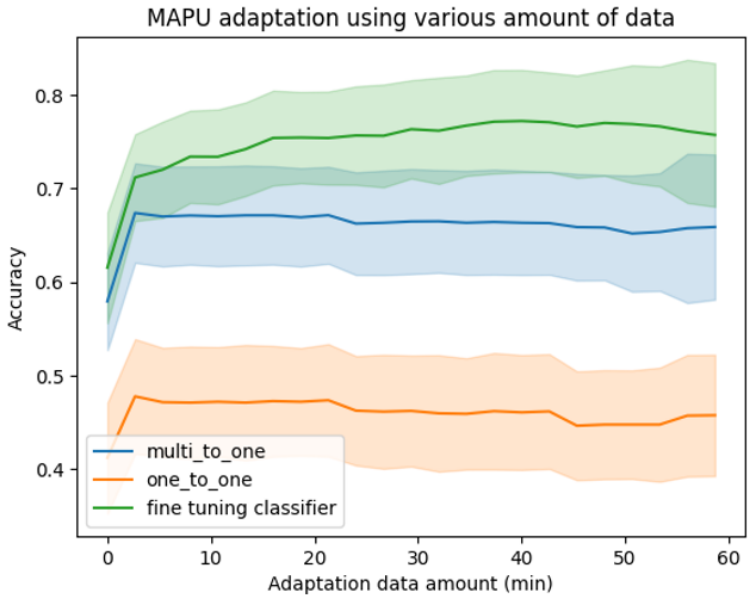

Figure 4.7: Accuracy in $\mathcal{T}$ after adaptation by MAPU, when $D_{\mathcal{S}}$ includes one subject (orange) or multiple subjects (blue). The curve of fine tuning (green) is included as a baseline.

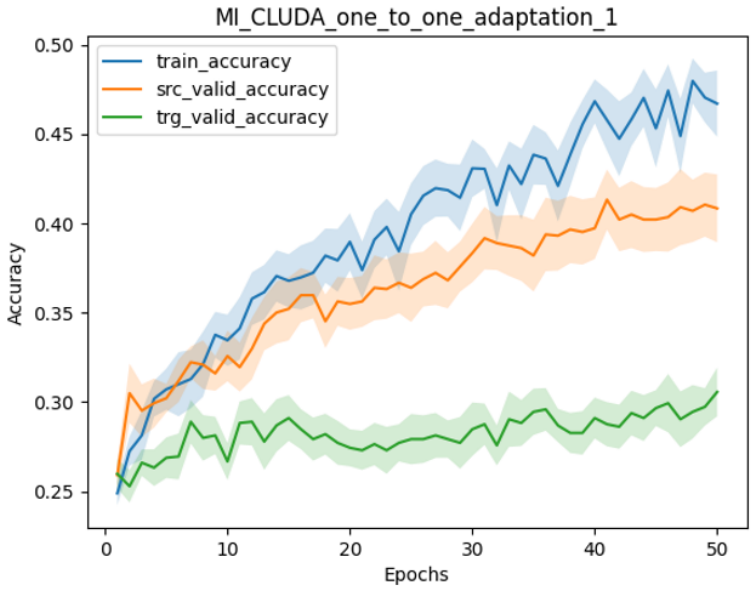

Figure 4.4: Training accuracy in $\mathcal{S}$ (blue), validation accuracy in $\mathcal{S}$ (orange) and accuracy after adaptation in $\mathcal{T}$ (green). $D_{\mathcal{S}}$ includes one subject i.e $N_{\mathcal{S}} = N_{\mathcal{T}} = 1$.

Figure 4.5: Training accuracy in $\mathcal{S}$ (blue), validation accuracy in $\mathcal{S}$ (orange) and accuracy after adaptation in $\mathcal{T}$ (green). $D_{\mathcal{S}}$ includes multiple subjects i.e $N_{\mathcal{S}} = N - 1, N_{\mathcal{T}} = 1$.

Figure 4.6: The adaptation benefit of CLUDA is significantly better when source domain $D_{\mathcal{S}}$ includes multiple subjects (Fig. 4.5)

4.3 Model Performance after Adaptation by Proposed Framework

Model accuracy in $\mathcal{T}$ with and without adaptation by the proposed framework is shown in Figure 4.8. Unfortunately, the proposed framework does not yet consistently offer increased accuracy. For certain individuals, it decreases post-adaptation accuracy to significantly lower than without adaptation. I have been troubleshooting the framework by inspecting the outputs from each component.

The support encoder F_{support} produces support embeddings z_{support} that clearly distinguish between subjects. When reduced and visualized on a 2-dimensional plane using tSNE [33], the support embeddings clearly cluster by subject identity. The clusters do not split into four lobes from within, meaning the support embeddings do capture significantly more subject-specific information than task-specific information as designed to.

As explained in the Methods section, transformed task embeddings $\hat{z}_{\text{task}}$ live in the same metric space as z_{task} yet are less subject-specific, therefore facilitating adaptation to unseen subjects. The transformed task embeddings are visualized using tSNE (Fig. 4.10). There is clear overlap and alignment between subjects, suggesting that the transformed space abstracts away subject-specific variability. Despite overlap, there is noticeable local cohesion (subjects cluster into "soft neighborhoods"). This structure is more organized than the embeddings produced by an encoder blindly trained across data from multiple subjects. The transformed task embeddings reduces inter-subject variation yet preserves task-relevant structure, so they are promising for the goal outlined in this thesis.

I will further investigate the cause of the framework's suboptimal performance. Specifically, I will inspect the weights in the attention module to understand how it transforms the task embeddings. I will also inspect how samples in $\mathcal{T}$ distribute around the decision boundary of the classifier. Another potential cause of issues comes from the episodic training strategy. Calculating loss from the data of only one subject in each training iteration, training may have encountered issues such as instable gradients, which would prevent the framework from

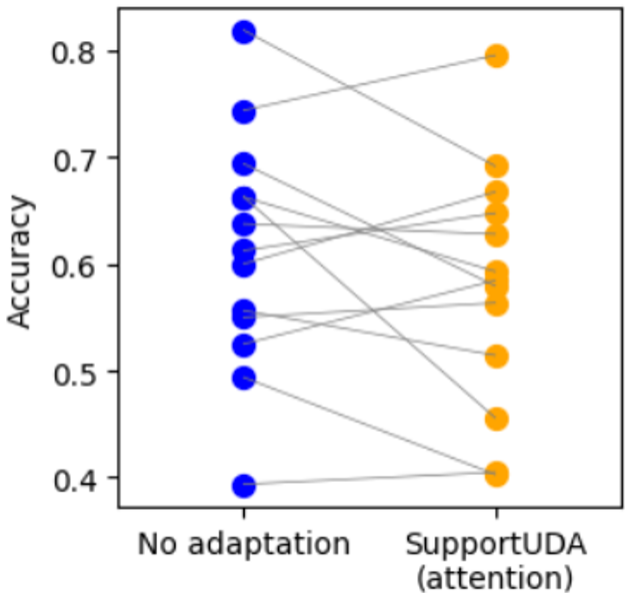

Figure 4.8: Performance in target domain dataset $\mathcal{T}$, after adaptation by proposed framework (yellow dots) vs. no adaptation baseline (blue dots). The two dots with and without adaptation for each subject is connected by a gray line.

converging to optimal minima.

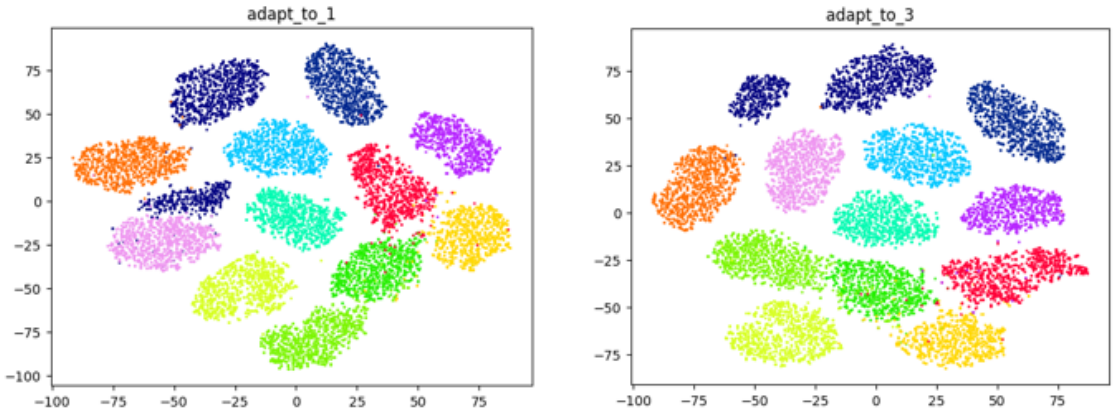

Figure 4.9: Support embeddings $\hat{z}_{\text{support}}$ reduced to a 2D plane; each dot is a support embedding $\hat{z}_{\text{support},i}$ from a data sample x_i, and each color represents one subject.

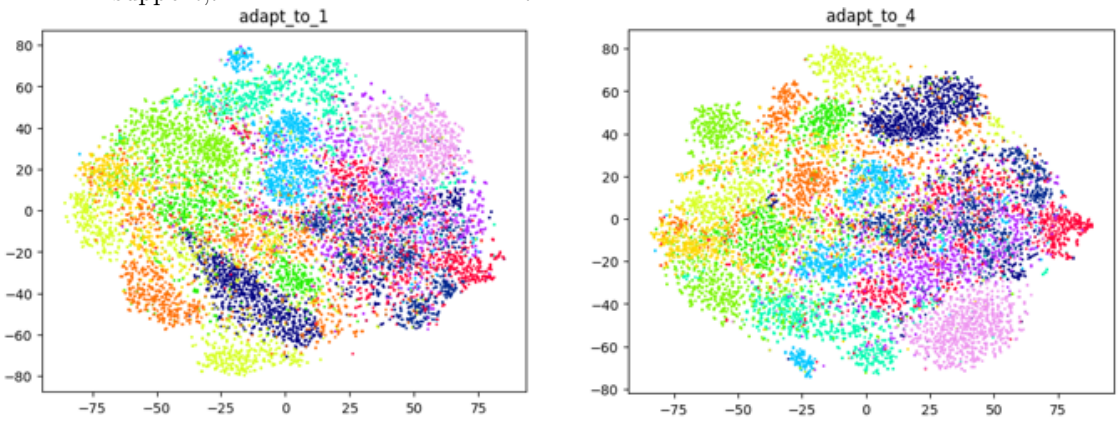

Figure 4.10: Transformed task embeddings $\hat{z}_{\text{task}}$ reduced to a 2D plane; each dot is a transformed task embedding $\hat{z}_{\text{task},i}$ from a data sample x_i, and each color represents one subject.

Figure 4.11: Support embeddings z_{support} distinguish between subject identity, and transformed task embeddings $\hat{z}_{\text{task}}$ reflect both task structure and similarity between subjects.

CHAPTER 5

CONCLUSION

To conclude this thesis, I showed that brain computer interfaces (BCI) require effective UDA methods to offer reliable, accurate service. As fine tuning pre-trained BCI models requires impractical amount of labeled samples from the new subject to calibrate effectively, I evaluated SOTA UDA methods such as CLUDA and MAPU in the context of this BCI exercise. I found that existing source-free UDA methods have limited expressiveness to effectively transfer the knowledge learned from the source subjects to the new target subject; the non source-free methods, on the other hand, require access to the whole source dataset for inference time adaptation. The framework proposed in this thesis aims to use contrastive learning, attention mechanism and meta-learning to dynamically project subject-specific data samples to subject-invariant representations. Although not yet fully functioning, analysis of the intermediate products in the pipeline shows promise to soon fulfill the design goals. I will make future attempts to remove potential issues and achieve the goal of source-free, unsupervised UDA for BCI.

REFERENCES

[1] W. KO, E. JEON, S. JEONG, J. PHYO, and H. I. SUK. A survey on deep learning-based short/zero-calibration approaches for eeg-based brain-computer interfaces. *Front Hum Neurosci*, 2021.

[2] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189, Lille, France, 07–09 Jul 2015. PMLR.

[3] Chethan Pandarinath, Paul Nuyujukian, Christopher H Blabe, Brittany L Sorice, John Saab, Francis R Willett, Leigh R Hochberg, Krishna V Shenoy, and Jaimie M Henderson. High performance communication by people with paralysis using an intracortical brain-computer interface. *eLife*, 6:e18554, 2017.

[4] Sarah L Metzger, Francis R Willett, et al., and Edward F Chang. A high-performance speech neuroprosthesis. *Nature*, 620:544–552, 2023.

[5] Elemind Labs. Sleep enhancement through real-time phase-locked stimulation. `https://www.elemindtech.com`, 2024. Accessed: 2025-04-24.

[6] Meta Reality Labs. Meta's emg wristband: Neural interface technology. `https://tech.fb.com/reality-labs/2021/03/introducing-our-wrist-based-input-device/`, 2021. Accessed: 2025-04-24.

[7] Leigh R Hochberg, David Bacher, Beata Jarosiewicz, Nickiel Y Masse, Jonathan D Simeral, John Vogel, Sami Haddadin, Jian Liu, Sydney S Cash, Patrick van der Smagt, et al. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature*, 485(7398):372–375, 2012.

[8] Dae-Hyeong Kim, Nanshu Lu, Rui Ma, Yihui Kim, Roozbeh H Kim, Shuodao Wang, Jian Wu, Sunghoon M Won, Huanyu Tao, Abu Javed Islam, et al. Epidermal electronics. *Science*, 333(6044):838–843, 2011.

[9] Lu Lu, Philipp Gutruf, Lin Xia, Darshil L Bhatti, Jae-Woong Kim, Xiaochuan Wang, Xiaogang Feng, Ryan Lobell, Ponnusamy Rajendran, Mark S Haney, et al. Injectable, cellular-scale optoelectronics with applications for wireless optogenetics. *Science Advances*, 3(6):e1601473, 2017.

[10] Gopala K Anumanchipalli, Josh Chartier, and Edward F Chang. Speech synthesis from neural decoding of spoken sentences. *Nature*, 568(7753):493–498, 2019.

[11] Robin T Schirrmeister, Jost T Springenberg, Lukas DJ Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and Tonio Ball. Deep learning with convolutional neural networks for eeg decoding and visualization. *Human brain mapping*, 38(11):5391–5420, 2017.

[12] Francis R Willett, David T Avansino, Leigh R Hochberg, Jaimie M Henderson, and Krishna V Shenoy. Brain-to-text communication via handwriting decoding. *Nature*, 593(7858):249–254, 2021.

[13] Yusuke Takagi and Shinji Nishimoto. High-resolution image reconstruction with latent diffusion models from human brain activity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23198–23208, 2023.

[14] Yujia Chen, Yixuan Yang, Tianyi Xue, Hao Wu, Hao Peng, Yanting Liu, Zhicheng Zuo, Jiajun Wang, Han Lu, Guosheng Hu, et al. Mind-video: Reconstructing video frames from brain activity. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2023.

[15] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[16] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[17] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020.

[18] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019.

[19] Yilmazcan Ozyurt, Stefan Feuerriegel, and Ce Zhang. Contrastive learning for unsupervised domain adaptation of time series. In *The Eleventh International Conference on Learning Representations*, 2023.

[20] Geoffrey French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. In *International Conference on Learning Representations*, 2018.

[21] Hyeonseob Chang, SeongUk Park, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7354–7362, 2019.

[22] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*, volume 17, pages 529–536, 2005.

[23] Jian Liang, Dapeng Hu, and Jiashi Feng. We don't need source data: Source-free domain adaptation via entropy minimization. In *International Conference on Learning Representations*, 2020.

[24] Mohamed Ragab, Peiliang Gong, Emadeldeen Eldele, Wenyu Zhang, Min Wu, Chuan-Sheng Foo, Daoqiang Zhang, Xiaoli Li, and Zhenghua Chen. Evidentially calibrated source-free time-series domain adaptation with temporal imputation. *CoRR*, abs/2406.02635, 2024.

[25] Tomer Volk, Eyal Ben-David, Ohad Amosy, Gal Chechik, and Roi Reichart. Example-based hypernetworks for multi-source adaptation to unseen domains. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9096–9113, Singapore, December 2023. Association for Computational Linguistics.

[26] David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *International Conference on Learning Representations*, 2017.

[27] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[28] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 1199–1208, 2018.

[29] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*, 2020.

[30] S. CHAMBON, M. N. GALTIER, P. J. ARNAL, G. WAINRIB, and A. GRAMFORT. A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series. *IEEE Trans Neural Syst Rehabil Eng*, 2018.

[31] R. T. SCHIRRMEISTER, J. T. SPRINGENBERG, L. D. J. FIEDERER, M. GLASSTETTER, K. EGGENSPERGER, M. TANGERMANN, F. HUTTER, W. BURGARD, and T. BALL. Deep learning with convolutional neural networks for eeg decoding and visualization. *Hum Brain Mapp*, 2017.

[32] B Kemp, AH Zwinderman, B Tuk, HAC Kamphuisen, and JJL Oberyé. Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the eeg. *IEEE-BME*, pages 1185–1194, 2000.

[33] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.