Generalizable Sample-efficient Siamese Autoencoder for Hidden Hearing Loss Diagnosis in Listeners with Subjective Tinnitus

Zhe Liu, *Member, IEEE,* Lina Yao, *Member, IEEE* Xianzhi Wang, *Member, IEEE*
Jessica Monaghan, *Member, IEEE* Roland Schaette, *Member, IEEE* Zihuai He, *Member, IEEE*
and David McAlpine, *Member, IEEE*

***Abstract*—Electroencephalogram (EEG)-based neurofeedback has been widely studied for tinnitus therapy in recent years. Most existing research relies on experts' cognitive prediction, and studies based on machine learning and deep learning are either data-hungry or not well generalizable to new subjects. In this paper, we propose a robust, data-efficient model for distinguishing tinnitus from the healthy state based on EEG-based tinnitus neurofeedback. We propose trend descriptor, a feature extractor with lower fineness, to reduce the effect of electrode noises on EEG signals, and a Siamese encoder-decoder network boosted in a supervised manner to learn accurate alignment and to acquire high-quality transferable mappings across subjects and EEG signal channels. Our experiments show the proposed method significantly outperforms state-of-the-art algorithms when analyzing subjects' EEG neurofeedback to 90dB and 100dB sound, achieving an accuracy of 91.67%-94.44% in predicting tinnitus and control subjects in a *subject-independent* setting. Our ablation studies on mixed subjects and parameters show the method's stability in performance.**

***Index Terms*—EEG, subject-independent, Siamese autoencoder, domain alignment, trend descriptor, tinnitus.**

I. Introduction

Tinnitus is a type of phantom perception caused by neural activities related to auditory system disorder. It is a common disease in large populations, covering over 17% of the general population and up to 33% of the elderly [1], and has been widely studied over the last decades. Tinnitus may lead to hearing loss if not treated timely but, meanwhile, can be difficult to be diagnosed. Currently, the diagnosis of tinnitus remains largely relying on patients' cognitive reactions to questionnaires and auditory tests [2].

Extensive experiments and studies on exploring the causes of tinnitus have led to the widely accepted opinion that tinnitus may be triggered under the stressful and annoying situations temporarily but turned into permanence by the mechanism of central auditory system [3], [4]. The mechanism enlarges and reinforces the relationship between the unpleasant situations and the tinnitus unintentionally, and turns out to be a persisting state at last. Some research proves that patients' responses to tinnitus will decrease if they turn their attention from tinnitus to other irrelevant tasks [5]. This discovery suggests that we may alleviate and terminate the aberrant neural activity in the central auditory system by breaking the association between tinnitus and negative emotions or situations by habituation. Following this idea, sound therapy aims to alleviate tinnitus via training patients using session sounds. Until now, sound controlled tinnitus therapy has proven effective [6] and become the principal treatment to tinnitus. Considering the variance of subjects in the phantom type, it is crucial to schedule treatment based on individual patients' feedback in such therapy.

Compared with traditional auditory tests that investigate patients' tinnitus, electroencephalogram (EEG), or more specifically, Auditory brainstem response (ABR), allows obtaining real-time feedback from the nervous system using non-invasive wearable devices. While the neurofeedback can be an effective data source for experts—who analyze and decides the proper sound treatment for patients manually [7], [8], [9]—machine learning and deep learning methods, e.g., support vector machine (SVM) [10], [11], neural network [12], [13], and autoencoder [14], [15], have achieved extraordinary performance in EEG-based neurofeedback analysis. Recently, generative models have shown the potential for overcoming subject variances in tinnitus neurofeedback analysis [16], [17], given it capability in domain alignment and domain transfer. As subject variance can be viewed as characteristic information of a domain, generative models can learn how to transfer the information from one domain to another. Besides, generative models can adopt domains alignment to embed the samples from different subjects into a unified space and thus achieve better classification.

This paper introduces a novel hand-engineered descriptor and an automatic representation learning model named Siamese Autoencoder for small-scale datasets and subject-independent experiments. Our designed features achieve remarkable performance in distinguishing tinnitus patients from control subjects. They can be used for either determining whether a patient has recovered to a healthy state or predicting the usefulness of certain sound in neurofeedback-based sound therapy for tinnitus treatment. We make the following contributions in this paper:

- We design a novel Siamese Autoencoder with extra

Zhe Liu and Lina Yao are with the School of Computer Science and Engineering, University of New South Wales, Austrlaia e-mails: zhe.liu@student.unsw.edu.au, lina.yao@unsw.edu.au.

Xianzhi Wang is with School of Computer Science, University of Technology Sydney.

Jessica Monaghan and David MaAlpine are with Department of Linguistics, Macquarie University.

Roland Schaette is with Ear Institute, University College London.

Zihuai He is with School of Medicine, Stanford University.

auxiliary accuracy loss, domain loss and alignment loss for better autoencoder optimization, domain transfer, and domain alignment, respectively. We further propose a trend descriptor that can reduce the effect of electrode noise to complement the autoencoder representations. The proposed two features, in combination, achieve good generalization in different experiments.
- We introduce a new split method, i.e., the Anchor Split training procedure, for our Siamese Autoencoder. The new split method contains the required information for domain alignment, domain transfer, and class prediction, and is a more suitable split method than tradition randomly cross-validation split methods.
- We experimentally show our approach outperforms state-of-the-art algorithms by 12.5% and 5.56% under two different conditions, respectively. We also study the impact of our proposed two features on the final performance and present ablation studies on the hyper-parameters and sound classification.

II. Related Work

Much research has been conducted about EEG-based neurofeedback neural activity training to alleviate brain system disorders. This treatment has been widely used in treating epilepsy [18] and attention deficit hyperactivity disorder [19], and has achieved outstanding performance. Tinnitus researchers similarly used neurofeedback to monitor EEG status and applied brain training to keep brain maintaining in a healthy state.

The early work on judging the health of neurofeedback mainly relied on the cognitive judgement of experts with little statistics information. The first attempt of neurofeedback training on tinnitus treatment was carried out by [7]. They researched on 40 patients and 15 control subjects. The treatment protocol was to increase the Alpha-band activity and decrease the Beta-band activity of EEG, which was considered to be related to tinnitus. The experiment lasted for seven years, and all the patients expressed a significant decrement of their tinnitus by the Tinnitus Questionnaire (TQ). Schenk et al. [9] further replicated the experiment in 40 patients. They conducted stress-test while monitoring the EEG of patients and control subjects, and the experiment results showed that all the patients scored tinnitus with less annoying and stressful in TQ by tinnitus after training. Among the patients, 23 of 40 patients succeed in increasing Alpha-band activities, but 13 of them failed to decrease Beta-band activities. The experiment confirmed the effectiveness in treating tinnitus with neurofeedback based EEG training but also showed that simple lifting of tinnitus patients' EEG did not equal to the healthy EEG state.

Researchers further began to use some statistical tools to help judge whether patients' EEG recover to a healthy state. Weiler et al. [8] extracted the power value of Alpha-band, Beta-band, Theta-band and Delta-band activities and compared the z-score of the patient with 20 control subjects. The protocol was to ameliorate the waves of patients to have similar z-score with the other observed 20 control subjects. The experiment showed that tinnitus might be correlated to the above all four bands. Milner et al. [20] applied slow cortical potential (SCP) in neurofeedback tinnitus treatment, which was a descriptor of overall cortical distribution shifts in neural activities. The case report of this SCP-based neurofeedback training showed that the reduction of frequency in Delta-band and Theta-band could effectively decrease the tinnitus loudness and pitch. These studies showed that tinnitus might have a complicated relationship with EEG patterns, which was hard for simple models to catch. The subject variance might lead to the different patterns of the EEG. All the above-mentioned research stayed on the stage of relying on the cognitive judgements of experts and case-specific analysis. Therefore, the research did not have a clear criterion, and the experiment results differed from each other.

Recently, researchers sought the help of machine learning and deep learning methods for a solid criterion in the analysis of tinnitus. Some efforts aim to distinguish tinnitus patients from control subjects by machine learning. Sun et al. [13] extracted Principal Components Analysis (PCA), Fast Fourier Transformation (FFT), and frequency-domain statistics features for analysis. Similarly, Li et al. [10] preprocessed data in the frequency domain, and further extracted the features by cosine mapping and main-phase computing. Both of the work received good performance in the experiments. However, these studies were subject-dependent, which meant that some of the test samples come from the same subjects in training. Then, short-time sampling from the same subjects would produce some similar samples, so subject-dependent experiments may contain similar samples in both train and test samples, which would overestimate the performance of models. Wang et al. [11] studied the subject-independent experiments in classifying tinnitus patients from control subjects. It adopted FFT and concatenate the multi-view information from multiple channels and bands, which achieved good performance with least squares support vector machine in a dataset of 29 volunteers.

Unfortunately, these tinnitus-related work only considered how to distinguish patients from control subjects in a quiet environment: control subjects would not hear anything, but tinnitus patients would hear the phantom sound, which might lead to a clear difference in EEG. Moreover, their model failed to consider to help sound therapy for tinnitus, which needed models to predict tinnitus patients from control subjects while hearing a sound. Therefore, their models could not handle the job of neurofeedback based sound therapy for tinnitus.

Other EEG study fields have attracted many advanced models and analysis tools. Some studies [21], [12] utilized deep learning to predict the sleep stages and activities in subject-independent experiments; others [15], [14] applied generative models to generate low-dimension features; finally, recent studies [22], [23] adopted domain transfer by generative models to enhance the EEG analysis. However, these powerful generative models would be limited in the data scale, which needed a large scale data to optimize the massive network parameters.

In summary, most studies [24], [25] only took statistics features (e.g. power value) and relied on the large scale dataset for training; in contrast, most tinnitus EEG dataset

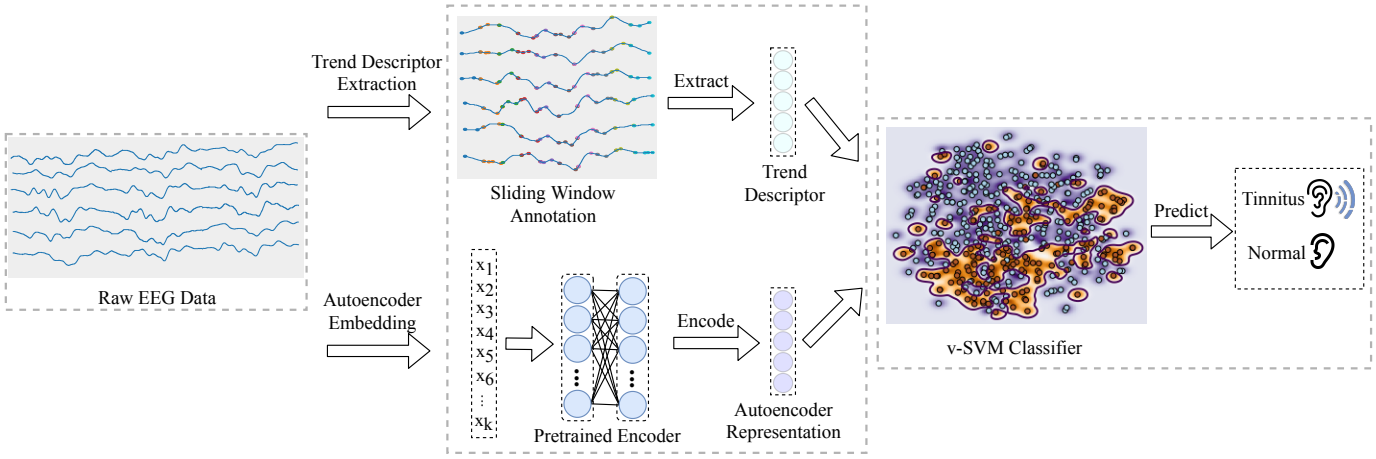

Fig. 1. Model Overview. We create two low-dimensional representations of raw data by a) using a trend descriptor and b) encoding the raw data with a pre-trained encoder from our Siamese autoencoder. The sliding window annotation highlights the peaks and troughs in consecutive sliding-windows in different colors. For the encoder input, x_k denotes the kth sample in an EEG trial. We depict the trend descriptor and the encoded representation with blue and purple circles, respectively. v-SVM takes the combination of two features to predict whether the EEG data belong to a tinnitus patient or a healthy subject.

had a limited scale. Little work has been done to provide a reliable subject-independent tool to assist sound therapy for tinnitus. Therefore, we propose a novel trend descriptor which contains better overall shape information, and we design a Siamese autoencoder intended for small-scale datasets and subject variance learning in sound therapy.

III. Methodology

A. Model Overview

Fig. 1 shows an overview of our proposed model with feature construction and class prediction as two main steps. The model takes the raw EEG signals as inputs and generates two sources of features: a trend descriptor (which separates a single trial into sliding windows and summarizes the trend in each) and a pre-trained encoder of the Siamese autoencoder (which learns subject differences and align samples from different subjects into a unified domain). The classifier operates on these two feature sources to predict the class: tinnitus or control. Throughout this section, we define the EEG trial set as $X = \{X_1, X_2, ..., X_n\}$, where X_i denotes a single trial.

B. Trend Descriptor Extraction

We propose a lower fineness trend descriptor to solve the electrode noise problem in EEG signals. The trend descriptor first finds the peak and trough points in sliding windows and then collects the magnitude and number of peaks and troughs in each window.

First, we introduce our lower fineness extreme point searching method. Given an EEG trial $X_i \in X$, we define the slice set of X_i as $Q_i = \{q_1, q_2, ..., q_w\}$, where q_i denotes the first sliding window of Q_i. Then, we search the following points of a time point n to sign it. Given a certain searching scope, we have the sign function:

$$Trend(n) = \begin{cases} Up & (p > d + \mu) \\ Stable & (d + \mu \geq p \geq d - \mu) \\ Down & (p < d - \mu) \end{cases} \quad (1)$$

where n denotes a time point in a sliding window; p, d denote the larger point number and the smaller point number than n in the searching scope; μ denotes a hyperparameter which lowers the fineness of searching extreme points and ignores the abnormal fluctuation caused by electrode noise. Then, the peaks and troughs in the sliding window are readily identified by observing when the sign function changes from Up to Down, or vice versa. The position of a peak will be the last sample with a Up or Stable sign before a sample with a Down sign. Similarly, troughs can be identified from the last sample with a Down or Stable sign before a change to an Up sign. The peaks and troughs are collected in temporal order as E, and we can express the trend character of an arbitrary sub-window q_i by a 4-dimension vector $Trend(q_i)$, where:

$$\begin{aligned} R &= \{E_j : E_j < E_{j+1}, E_j \in q_i\} \\ D &= \{E_j : E_j > E_{j+1}, E_j \in q_i\} \\ Trend(q_i) &= \{|R|, \sum_{i \in R} abs(E_i - E_{i+1}), \\ &\qquad |D|, \sum_{i \in D} abs(E_i - E_{i+1})\} \end{aligned} \quad (2)$$

Since E is time-ordered, a peak can only be followed by a trough and vice-versa. Therefore, R, D denote the sets of peak and trough points. We denote by $|R|$,$|D|$ the number of elements in R, D, and by abs the absolute value. Then, we can obtain the trend descriptor of a trial $Trend(X_i)$ by concatenating the trend descriptor of sliding windows belongs to its corresponding Q_i:

$$Trend(X_i) = \{Trend(q_i) : q_i \in Q_i\} \quad (3)$$

The trend descriptor will have $4 * w$ dimensions, depending on sliding-window-size and step. We show the procedure in Algorithm 1. Finally, we apply z-score normalization:

$$\begin{aligned} Trend(X_i) &= Normalize(\{Trend(q_i) : q_i \in Q_i\}) \\ &= \{\frac{Trend(q_i) - \overline{Trend(q_i^X)}}{\sigma(Trend(q_i^X))} : q_i \in Q_i\} \end{aligned} \quad (4)$$

Algorithm 1 Trend Descriptor Extraction with default values: window size = 50, step = 50, search scope = 10, $\mu = 2$

Require: EEG trial set $X = \{X_1, X_2, ..., X_n\}$

for $i \in [1, n]$ **do**
  Split X_i into slices $Q_i = \{q_1, q_2, ..., q_w\}$
  for $j \in [1, w]$ **do**
    for $m \in [1, size]$ **do**
      Sign $n_m \in q_j$ by Eq. (1)
    end for
    Calculate $Trend(q_j)$ by Eq. (2)
  end for
  Concatenate and obtain $Trend(X_i)$ by Eq. (3)
end for
for $i \in [1, n]$ **do**
  Normalize $Trend(X_i)$ by Eq. (4)
end for

where $q_i^X = \{q_i^{X_i} : X_i \in X\}$ denotes the set of sliding windows from the same time of all trials; $\overline{Trend(q_i^X)}$ denotes the mean value; and $\sigma(Trend(q_i^X))$ denotes the standard deviation.

C. Siamese Autoencoder Representation Learning

Autoencoders are a widely used tool to extract the significant information from low signal-to-noise ratio EEG data by encoding raw data into a low-dimension representation. Here, we use a Siamese structure and domain adaptation methods to enhance the autoencoder. Fig. 2 illustrates the architecture of the proposed Siamese autoencoder, which uses two parallel autoencoders with shared weight parameters as the analysis tool as contains three types of loss function: L_{AE}, $L_{Siamese}$ and L_{align}. The proposed Siamese autoencoder takes pairwise raw data, meaning that for N input samples, we have $\binom{N}{2}$ training pairs. Therefore, we can generate enough training samples to optimize the deep-learning model even with the small-scale EEG dataset. Given a raw trial data set X, we compose the pairwise training batch $B = \{B_1, B_2, ..., B_n\}$ by $B_i = \{X_a, X_b : X_a, X_b \in X\}$ (details of construction of B in Section III-E). We define s as the autoencoder representation, and the representation pair of B_i as $S_i = \{s_a^i, s_b^i\}$, where s_a^i, s_b^i denote two corresponding representations of B_i, respectively.

In the following, we introduce the autoencoder structure, the Siamese structure, and domain alignment representation learning for the proposed Siamese autoencoder, respectively.

1) Autoencoder Structure: An autoencoder consists of two components: an *Encoder* and a *Decoder*.The *Encoder* extracts the low-dimension representation, and *Decoder* reconstructs the raw data from the representation. The representation should contain all the essential information from the raw data to allow the *Decoder* to recover the raw data.

Given an arbitrary trial data X_i, we construct the *Encoder* with two fully connected (FC) layers activated by the hyperbolic tangent (Tanh) function. Here, we use the Tanh function because it can convert the input data into [-1,1] and help avoid the vanishing and exploding gradient problems. Then we can obtain the representation s_i of X_i by:

$$\begin{aligned} Layer(X_i) &= Tanh(W * X_i + b) \\ s_i &= Encoder(X_i) \\ &= Layer(Layer(X_i)) \end{aligned} \tag{5}$$

where W, b denote the weights and bias of the *Encoder*, respectively. Similarly, *Decoder* consists of two FC layers, but only the first layer has the Tanh activation function. Our raw data scope is out of the range of the Tanh function, so we

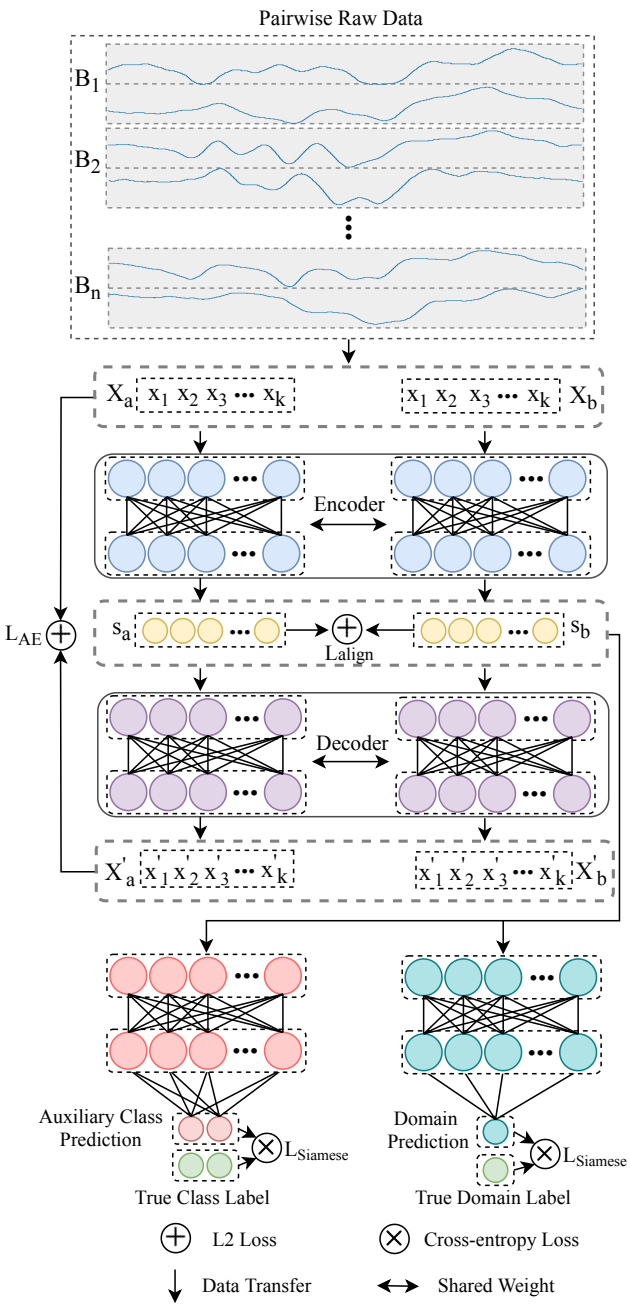

Fig. 2. Siamese Autoencoder Architecture. X_a, X_b represent the raw data from a train pair and $X_a^{'}, X_b^{'}$ denote the corresponding generated data. s_a, s_b are the encoded representation from *Encoder*. Specially, $L_{Siamese}$ consists of two parts: auxiliary class loss and domain loss.

drop the Tanh function in the decoder output layer. Then, we have the generated raw data $X_i^{'}$ by:

$$\begin{aligned} X_i^{'} &= Decoder(s_i) \\ &= W * (Layer(s_i)) + b \end{aligned} \tag{6}$$

where W, b denote weights and bias of $Decoder$, respectively. We want the generated $X_i^{'}$ to be as similar as possible to the raw data X_i. Therefore, we apply the L2 loss function to ensure the $Encoder$ and the $Decoder$ preserve the necessary information from the raw data. This leads to the pairwise autoencoder loss function, L_{AE}, for an arbitrary pairwise data B_i:

$$L_{AE} = \sum_{X_a \in B_i} ||X_a - X_a^{'}||_2^2 + \sum_{X_b \in B_i} ||X_b - X_b^{'}||_2^2 \tag{7}$$

where $||X_a - X_a^{'}||_2^2$ denotess the L2 loss between the raw data and the generated data. L_{AE} will ensure the representation S_i carries the essential information of raw data B_i during the training.

2) Siamese Structure: This section introduces the auxiliary class enhanced Siamese structure for domain transfer leaning. Tradition Siamese structure compares two samples and determines whether they are from the same class. In our model, we compare the pairwise representations S_i and predict whether they belong to the same subject domain. This domain prediction let $Encoder$ transfer the domain information from different subjects and learn how to generate the representation with discriminative subject identity information. We also propose an auxiliary class loss, which prevents $Encoder$ from generating white noise representations.

Given an arbitrary pairwise representations $S_i = \{s_a^i, s_b^i\}$ from $Encoder$, we define the subject domain label D as 1 when s_a^i, s_b^i are from the same subject otherwise 0. Siamese structure consists of two classifiers $Class$ and $Domain$ for predicting class labels Y and domain labels D of a pairwise representation S_i, respectively. Both classifiers contain two FC layers and take Sigmoid Function as the activation function of outputs. We apply the Cross-entropy Loss as our classification loss function. Then, Siamese structure loss $L_{Siamese}$:

$$\begin{aligned} L_{cls}(y, \hat{y}) &= -y \log \hat{y} - (1-y) log(1-\hat{y}) \\ L_{Siamese} &= \sum_{S_i \in S} L_{cls}(Y_a, Class(s_a^i)) + L_{cls}(Y_b, Class(s_b^i)) \\ &\quad + L_{cls}(D, Domain(S_i)) \end{aligned} \tag{8}$$

where $L_{cls}(y, \hat{y})$ denotes the Cross-entropy Loss function between true label y and predicted probability $\hat{y}$; Y_a, Y_b denote the class labels of s_a^i, s_b^i, respectively. The first two terms represent the accuracy loss, which offers the label information during the training of autoencoder. Label information could assist the optimization of autoencoder towards better prediction results, and keep the key information for class prediction. The last term is the domain loss, which hopes the representations can contain the subject identity information. Subject identity information will make the representations more diverse and enhance the robustness of algorithms on predicting class labels from different subjects. Therefore, $L_{Siamese}$ will make the representations produced by autoencoder more easily to predict the class labels and the subject source-domain.

3) Domain Alignment Representation Learning: The Siamese structure improves the subject information and class information carried by the representations, but it fails to align the sample domain into a unified classification space. Therefore, we propose L_{align} to adopt domain alignment of the generated representations from $Encoder$.

Given an arbitrary training batch set $S = \{S_1, S_2, ..., S_n\}$, where $S_i = \{s_a^i, s_b^i\}$, then we have:

$$\begin{aligned} L_{align} &= \sum_{S^i \in U} ||s_a^i - s_b^i||_2^2 - \sum_{s^i \in T} ||s_a^i - s_b^i||_2^2 \\ s.t. \quad U &= \{S^i : Y_{s_a^i} = Y_{s_a^i}\}, \quad T = \{S^i : Y_{s_a^i} \neq Y_{s_a^i}\} \end{aligned} \tag{9}$$

where U, T denote the set of pairwise data from same and different classes, respectively; s_a^i, s_b^i denote the corresponding two representations of S_i. L_{align} is designed to minimize the pairwise representations difference from the same class, and maximize the representation difference from the different classes, in which way the class domain of different subjects will be adapted into a unified domain after optimization. Based on Eq. (7), Eq. (8), and Eq. (9), we could get the loss function of our Siamese Autoencoder:

$$L = L_{AE} + L_{Siamese} + L_{align} \tag{10}$$

where L_{AE} is the basic autoencoder loss function; $L_{Siamese}$ is intended for subject variance learning by domain transfer; L_{align} is designed for domain alignment in classification.

D. Classifier

We use v-Support Vector Machine (v-SVM) with default settings in scikit-learn [26] to predict class labels. v-SVM takes both trend descriptors and autoencoder representations as the inputs of the classifier. It aims to find a hyper-plane that maximizes the sample distance to this hyper-plane. In our tinnitus prediction, which is a binary classification problem, we define the binary labels as {-1,1}. Then, we denote the classification hyper-plane by

$$0 = w^T r + b \tag{11}$$

where w denotes the normal vector of the hyper-plane; b denotes a real number; r denotes the mapped point by kernel function (e.g. radial basis function kernel). Note, w and r are vectors with multi-dimensions. Then, we have the sign function for class label y:

$$y_{r_i} = \begin{cases} 1 & w^T r_i + b \geq 1 \\ -1 & w^T r_i + b \leq -1 \end{cases} \tag{12}$$

Now, our goal transforms into maximizing the gap between classification margin 1 and -1, and we figure out the distance of an arbitrary sample r_i to the hyper-plane by

$$\gamma_i = \frac{y_i(w^T r_i + b)}{||w||} \tag{13}$$

where $||w||$ denotes the norm of hyper-plane normal vector, and y_i denotes the class label of sample r_i.

Finally, we have the loss function for v-SVM [27]:

$$\begin{aligned} &\arg\min \frac{||w||^2}{2} - \rho v + \frac{1}{l}\sum_{i}^{l}\zeta_i \\ s.t. \quad &y_i[w^T r_i + b] \geq \rho - \zeta_i, i \in [1, l] \\ &\rho = y_i - \varepsilon, \rho \geq 0 \\ &\zeta_i \geq 0, i \in [1, l] \end{aligned} \tag{14}$$

where v denotes a hyper-parameter between 0 and 1; ε denotes the error tolerance of prediction; ρ denotes the tolerated sample sign; l denotes the sample number; ζ_i denotes the relaxation variable. The optimization details are in [27].

E. Anchor Split Training and Procedure

Training batch split is a critical factor of generative models' performance. Therefore, we propose Anchor split for our Siamese autoencoder that includes subject-difference and class difference in each training batch. The proposed Anchor-split method allows the autoencoder to learn the characters of subjects and classes evenly. First, we select a random sample from each subject as Anchor to ensure each training batch will contain the information of all training subjects. Second, we randomly pick a batch size number of positive and negative unique samples belonging to each Anchor, respectively. The selected samples will be combined as pairs with the corresponding Anchors that carry the class information. Then, we further choose a random batch size number of positive and negative unique samples belonging to the different subjects from Anchors and merge them as pairs to transfer the subject variance information to the autoencoder. Third, we keep the selected pairs to build a data warehouse for each Anchor and shuffle them randomly. For each iteration of an epoch, we will select a certain number of pairwise data from each Anchors data warehouses to obtain a training batch. In each epoch, we repeat the three steps to relocate Anchors and rebuild the corresponding data warehouses to improve the training diversity. We illustrate the detailed training procedure in Algorithm 2.

Algorithm 2 Training procedure with default values: batch size = 128, max epoch = 300, representation dimension = 128, learning rate (lr) = 0.001

Require: EEG trial set $X = \{X_1, X_2, ..., X_n\}$
1: Split train/test subjects
2: Extract trend descriptor by Algorithm 1
3: **while** epoch $\leq$ max epoch **do**
4: Set Anchors in train subjects
5: Build data warehouse within train subjects
6: **while** Data warehouse not empty **do**
7: Pick up a pairwise train batch
8: Calculate L by Eq. (10)
9: $W, b \leftarrow Adam(lr, L)$
10: **end while**
11: **end while**
12: Encode samples of test subjects by Eq. (5)
13: Use v-SVM to predict classes with trend descriptors and representations of test subjects by Eq. (14)

IV. Experiment

A. Experiment Setting

Our experimental dataset [28] contains 456 trials from 43 volunteers aged from 19 to 61, including 19 tinnitus patients and 24 control subjects. The EEGs were recorded with a 50 kilohertz (kHz) sampling frequency, and the stimuli were click trains with 90 decibels (dB) and 100dB sound pressure level (SPL). We conduct subject-independent experiments in distinguishing tinnitus subjects from control subjects for 90dB SPL and 100dB SPL click-evoked EEG data, respectively. We selected trials from two tinnitus patients and two control subjects for testing and used the remaining trials for training. This provided 216 trials for training and 49 trials for testing in the 90dB SPL data, generating 23,220 pairs of training data; we also have 148 trials for training and 43 trials for testing in 100dB SPL experiment, which contains 10,878 pairs of train samples. By this means, we generated sufficient data for training our Siamese autoencoder.

For both experiments, we define tinnitus patients as 1 and control subjects as 0, and set our Siamese autoencoder with the same parameters: lr = 0.005, batch size = 128, max epoch = 300, representation dimension = 64; we also set window-size = 40, step = 40, search scope = 10, $\mu = 2$ for our trend descriptor. We compared our method with several competitive methods: (**a**) v-Support Vector Machine (v-SVM) [29] using raw data as the baseline, (**b**) Improved Covariance Matrix Estimators (nCSP) [30], (**c**) Convolutional Recurrent Attention Model (CRAM) [12], (**d**) Autoencoder enhanced Extreme Gradient Boosting (AEXGB) [14], (**e**) Compact Convolutional Neural Network for EEG (EEGNet) [31], (**f**) Shallow Convolutional Network (ShallowNet) [32], and (**g**) Deep Convolutional Network (DeepNet) [32].

We define Macro and Weighted criteria for performance evaluation in our experiments:

$$\begin{aligned} Macro &: \frac{1}{|\mathcal{L}|}\sum_{l\in\mathcal{L}}\phi(y_l, \hat{y}_l) \\ Weighted &: \frac{1}{\sum_{l\in\mathcal{L}}|\hat{y}_l|}\sum_{l\in\mathcal{L}}|\hat{y}_l|\phi(y_l, \hat{y}_l) \end{aligned} \tag{15}$$

where $\mathcal{L}$ denotes the set of labels; $y_l, \hat{y}_l$ denote the predicted label and true labels; $|y_l|$ denotes the number of predicted labels which have the label l; $\phi(y_l, \hat{y}_l)$ denotes the function to compute Precision, Recall, or F1-Score for the true and predicted labels.

B. Results

Table I and Table II show the algorithms' best performance in distinguishing tinnitus subjects from control subjects on 90dB SPL and 100dB SPL EEG data, respectively. Note, 90dB SPL experiment has the same number of positive and negative samples, so Weighted-F1 has the same value as Macro-F1. v-SVM cannot distinguish the tinnitus patients with raw data; AE-XGB shows the autoencoder can effectively improve the quality of the raw data; while tradition machine learning methods like nCSP fail to learn the difference between tinnitus and

TABLE I
TINNITUS PATIENTS PREDICTION FROM CONTROL OBJECTS UNDER 90DB SPL.

Model	0-Precision	0-F1	1-Precision	1-F1	Macro-Precision	Macro-Recall	Macro-F1 (Weighted-F1)	Accuracy
v-SVM	0.5333	0.5926	0.5556	0.4762	0.5444	0.5417	0.5344	0.5417
nCSP	0.6364	0.6087	0.6154	0.6400	0.6259	0.6250	0.6243	0.6250
CRAM	0.5202	0.6844	1.0000	0.5030	0.7601	0.6680	0.5937	0.6140
AE-XGB	0.6923	0.7200	0.7273	0.6957	0.7098	0.7083	0.7078	0.7083
EEGNet	0.8571	0.6316	0.6471	0.7586	0.7521	0.7083	0.6951	0.7083
ShallowNet	0.7059	0.8276	1.0000	0.7368	0.8529	0.7917	0.7822	0.7917
DeepNet	0.8000	0.4706	0.5789	0.7097	0.6895	0.6250	0.5901	0.6250
SiameseAE	0.8571	0.9231	1.0000	0.9091	0.9286	0.9167	0.9161	0.9167

TABLE II
TINNITUS PATIENTS PREDICTION FROM CONTROL OBJECTS UNDER 100DB SPL.

Model	0-Precision	0-F1	1-Precision	1-F1	Macro-Precision	Macro-Recall	Macro-F1	Weighted-F1	Accuracy
v-SVM	0.4000	0.3077	0.5385	0.6087	0.4692	0.4750	0.4582	0.4749	0.5000
nCSP	0.5833	0.6364	0.6250	0.5556	0.6042	0.6000	0.5960	0.5960	0.6000
CRAM	0.3807	0.5514	1.0000	0.4913	0.6903	0.6628	0.5214	0.5089	0.5233
AE-XGB	0.5556	0.5882	0.6667	0.6316	0.6111	0.6125	0.6099	0.6123	0.6111
EEGNet	0.6667	0.8000	1.0000	0.7500	0.8333	0.8000	0.7750	0.7722	0.7778
ShallowNet	0.6154	0.7619	1.0000	0.6667	0.8077	0.7500	0.7143	0.7090	0.7222
DeepNet	0.8750	0.8750	0.9000	0.9000	0.8875	0.8875	0.8875	0.8889	0.8889
SiameseAE	0.8889	0.9412	1.0000	0.9474	0.9444	0.9500	0.9443	0.9446	0.9444

TABLE III
TREND DESCRIPTOR (TD) AND SIAMESE AUTOENCODER (SA) INDEPENDENT PERFORMANCE UNDER 90DB SPL AND 100DB SPL.

SPL	Feature	0-F1	1-F1	Weighted-F1	Accuracy
90dB	TD	0.500	0.500	0.497	0.500
90dB	SA	0.811	0.792	0.788	0.792
100dB	TD	0.775	0.775	0.778	0.778
100dB	SA	0.762	0.667	0.709	0.722

healthy EEG in subject-independent experiments. Deep learning models have better performance in both experiments—ShallowNet achieves 79.17% accuracy under 90dB SPL, and DeepNet obtains 88.89% accuracy under 100dB SPL, but neither of them has good performance in both experiments.

Our proposed method outperforms other algorithms in both experiments, which shows the Siamese generative model's stronger ability to catch the subject variance and the robustness of our algorithm. We achieve 100% precision and over 0.90 F1-score in the prediction of tinnitus patients, which means our model can find the most patients' EEG accurately. The overall accuracy of our model is over 90%, which shows the eligibility in the EEG-based sound therapy.

C. Trend Descriptor and Siamese Autoencoder Analysis

We use the same data split as in former experiments to explore the impact of the trend descriptor and the Siamese autoencoder, respectively. Specifically, we use v-SVM to show the performance of hand-crafted features and the output of the classifier in Siamese network to show the effectiveness of autoencoder-generated features.

Table III shows either of the two features cannot compete with their combination in performance in either experiment. Trend descriptor has better performance in 100dB SPL while merely provides useful independent information in 90dB SPL; Siamese autoencoder can extract meaningful information from low signal-to-noise ratio data but has worse performance than trend descriptor in 100dB SPL. Compared with the previous experiment, combining the two features delivers much better performance in both experiments, meaning the two features can complement and improve each other.

D. Sound Classification

We further test our model in predicting sound loudness in mixed subjects. We define 90dB as label 0 and 100dB as label 1. We select three tinnitus patients and two control subjects for the test and use the remaining samples for the train. Then we have 400 trials for train and 56 trials for the test, which means 79,800 pairwise samples for the train. We compare our method with the same algorithms in the prediction of tinnitus in Table IV. Prediction of sound level is easier for algorithms, while we get a baseline of 81.35%. The proposed Siamese autoencoder only has lower 100dB precision than some of the state-of-the-art algorithms but outperforms them in other criterion scores. This exploration reveals our model's powerful ability to analyze EEG-related tasks.

E. Ablation Study on Hyper-parameters

We investigate the robustness of our Siamese autoencoder, by testing the impact of learning rate (lr), batch size, and representation dimension on our model's performance, separately, with the other parameters default to lr = 0.001, batch size = 128, and representation dimension = 64. We run the model for 10 times and calculate the mean performance of the best five times to reduce the adverse impact caused by randomization in our Anchor split.

TABLE IV
SOUND LOUDNESS PREDICTION ON MIXED SUBJECTS.

Model	0-Precision	0-F1	1-Precision	1-F1	Macro-Precision	Macro-Recall	Macro-F1	Weighted-F1	Accuracy
v-SVM	0.7955	0.8642	0.8667	0.7027	0.8311	0.7684	0.7835	0.8040	0.8136
nCSP	0.6250	0.6494	0.6471	0.6197	0.6360	0.6351	0.6345	0.6345	0.6351
CRAM	0.9105	0.9379	0.9398	0.8897	0.9252	0.9058	0.9138	0.9196	0.9205
AE-XGB	0.8537	0.8974	0.8889	0.8000	0.8713	0.8366	0.8487	0.8611	0.8644
EEGNet	0.8182	0.8889	0.9333	0.7568	0.8758	0.8047	0.8228	0.8396	0.8475
ShallowNet	0.9722	0.9589	0.9130	0.9333	0.9426	0.9502	0.9461	0.9494	0.9492
DeepNet	0.8780	0.9231	0.9444	0.8500	0.9112	0.8729	0.8865	0.8958	0.8983
SiameseAE	1.0000	0.9722	0.9167	0.9565	0.9583	0.9730	0.9644	0.9664	0.9661

1) Learning Rate: Table V shows the results of lr that larger lr leads to the better performance in 90dB experiments but the worse performance in 100dB experiments. When $lr = 0.005$, our model achieves the most balanced best performance in both experiments. Our model achieves worst performance when $lr = 0.003$ in 90dB and $lr = 0.007$ in 100dB. The performance of our model will be largely influenced by lr.

2) Batch Size: We test seven commonly-used batch sizes. Since we have more than 32 subjects, we only test the batch sizes larger than 32. Batch size has only a small influence on the 100dB experiments but a larger influence on the 90dB experiments. 90dB experiments need a relatively small or a large enough batch size to allow model to learn the representation better, while 100dB experiments need a small batch size. Therefore, the best option for batch size is 128.

3) Representation Dimension: Both experiments have the worst performance when we use the 16-dimension representation, which means the subject information has much more information than the expression ability of a 16-dimension vector. Moreover, the too large dimension will decrease the noise compressed ability of autoencoder, which will also lead to performance reduction in representation learning. Therefore, we pick 64 as the balanced representation dimension for our autoencoder.

Overall, the standard deviation in most criteria is no greater than 0.05, demonstrating the model's robustness in different parameters. lr is the parameter that most significantly influences the performance of our model. 90dB experiments are more sensitive to the parameter choice, while 100dB experiments are more stable with respect to changes in batch size and representation dimension. The parameters have different optimal values and differed influences in the two experiments. Our model achieves the most balanced performance when $lr = 0.005$, batch size = 128, and representation = 64.

V. CONCLUSION

We propose a novel model that seamlessly integrates a lower fineness trend descriptor and a powerful Siamese autoencoder to distinguish tinnitus patients from control subjects based on EEG signals. In particular, the Siamese autoencoder can learn subject variances in small-scale datasets, and our proposed model outperforms state-of-the-art algorithms under different SPLs. We further prove the effectiveness of our model in sound loudness prediction and through ablation studies.

REFERENCES

[1] P. J. Jastreboff, W. C. Gray, S. L. Gold *et al.*, "Neurophysiological approach to tinnitus patients," *American Journal of Otology*, vol. 17, no. 2, pp. 236–240, 1996.

[2] P. H. Wilson, J. Henry, M. Bowen, and G. Haralambous, "Tinnitus reaction questionnaire: psychometric properties of a measure of distress associated with tinnitus," *Journal of Speech, Language, and Hearing Research*, vol. 34, no. 1, pp. 197–201, 1991.

[3] F. Gonzalez-Lima and H. Scheich, "Neural substrates for tone-conditioned bradycardia demonstrated with 2-deoxyglucose. ii. auditory cortex plasticity," *Behavioural brain research*, vol. 20, no. 3, pp. 281–293, 1986.

[4] D. J. Strauss, W. Delb, R. D'Amelio, Y. F. Low, and P. Falkai, "Objective quantification of the tinnitus decompensation by synchronization measures of auditory evoked single sweeps," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 16, no. 1, pp. 74–81, 2008.

[5] L. Oatman and B. Anderson, "Suppression of the auditory frequency following response during visual attention," *Electroencephalography and clinical Neurophysiology*, vol. 49, no. 3-4, pp. 314–322, 1980.

[6] P. J. Jastreboff, M. M. Jastreboff *et al.*, "Tinnitus retraining therapy (trt) as a method for treatment of tinnitus and hyperacusis patients," *Journal of the American Academy of Audiology*, vol. 11, no. 3, pp. 162–177, 2000.

[7] K. Gosepath, B. Nafe, E. Ziegler, and W. Mann, "Neurofeedback in der therapie des tinnitus," *Hno*, vol. 49, no. 1, pp. 29–35, 2001.

[8] E. W. Weiler, K. Brill, K. H. Tachiki, and D. Schneider, "Neurofeedback and quantitative electroencephalography," *International Tinnitus Journal*, vol. 8, no. 2, pp. 87–93, 2002.

[9] S. Schenk, K. Lamm, H. Gündel, and K.-H. Ladwig, "Neurofeedback-gestütztes eeg-α-und eeg-β-training," *Hno*, vol. 53, no. 1, pp. 29–38, 2005.

[10] P.-Z. Li, J.-H. Li, and C.-D. Wang, "A svm-based eeg signal analysis: an auxiliary therapy for tinnitus," in *International Conference on Brain Inspired Cognitive Systems*. Springer, 2016, pp. 207–219.

[11] S.-J. Wang, Y.-X. Cai, Z.-R. Sun, C.-D. Wang, and Y.-Q. Zheng, "Tinnitus eeg classification based on multi-frequency bands," in *International Conference on Neural Information Processing*. Springer, 2017, pp. 788–797.

[12] D. Zhang, L. Yao, K. Chen, and J. Monaghan, "A convolutional recurrent attention model for subject-independent eeg signal analysis," *IEEE Signal Processing Letters*, vol. 26, no. 5, pp. 715–719, 2019.

[13] Z.-R. Sun, Y.-X. Cai, S.-J. Wang, C.-D. Wang, Y.-Q. Zheng, Y.-H. Chen, and Y.-C. Chen, "Multi-view intact space learning for tinnitus classification in resting state eeg," *Neural Processing Letters*, vol. 49, no. 2, pp. 611–624, 2019.

[14] X. Zhang, L. Yao, D. Zhang, X. Wang, Q. Z. Sheng, and T. Gu, "Multi-person brain activity recognition via comprehensive eeg signal analysis," in *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. ACM, 2017, pp. 28–37.

[15] T. Nguyen, S. Nahavandi, A. Khosravi, D. Creighton, and I. Hettiarachchi, "Eeg signal analysis for bci application using fuzzy system," in *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015, pp. 1–8.

[16] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 2642–2651.

[17] Y. Taigman, A. Polyak, and L. Wolf, "Unsupervised cross-domain image generation," *arXiv preprint arXiv:1611.02200*, 2016.

TABLE V
LEARNING RATE PERFORMANCE ON MEAN SCORE (STANDARD DEVIATION).

	90dB SPL				100dB SPL			
lr	0-F1	1-F1	Weighted-F1	Accuracy	0-F1	1-F1	Weighted-F1	Accuracy
0.001	0.846(0.038)	0.836(0.025)	0.841(0.031)	0.842(0.031)	0.917(0.046)	0.926(0.044)	0.922(0.044)	0.922(0.044)
0.002	0.846(0.035)	0.818(0.041)	0.832(0.038)	0.833(0.037)	0.938(0.056)	0.949(0.045)	0.944(0.050)	0.944(0.050)
0.003	0.831(0.010)	0.797(0.040)	0.814(0.023)	0.817(0.020)	0.899(0.021)	0.901(0.023)	0.900(0.022)	0.900(0.022)
0.004	0.888(0.027)	0.878(0.037)	0.883(0.031)	0.883(0.031)	0.886(0.006)	0.891(0.004)	0.889(0.000)	0.889(0.000)
0.005	0.891(0.013)	0.874(0.022)	0.882(0.017)	0.883(0.017)	0.919(0.046)	0.925(0.043)	0.922(0.044)	0.922(0.044)
0.006	0.879(0.048)	0.870(0.058)	0.875(0.053)	0.875(0.053)	0.886(0.037)	0.891(0.033)	0.889(0.035)	0.889(0.035)
0.007	0.871(0.035)	0.858(0.035)	0.864(0.035)	0.865(0.035)	0.863(0.032)	0.870(0.023)	0.867(0.027)	0.867(0.027)
0.008	0.913(0.032)	0.898(0.039)	0.906(0.035)	0.906(0.035)	0.873(0.045)	0.882(0.039)	0.878(0.041)	0.878(0.042)
0.009	0.879(0.030)	0.846(0.042)	0.862(0.036)	0.865(0.035)	0.926(0.049)	0.934(0.044)	0.931(0.046)	0.931(0.046)
0.010	0.906(0.021)	0.906(0.015)	0.906(0.018)	0.906(0.018)	0.869(0.027)	0.880(0.022)	0.875(0.024)	0.875(0.024)

TABLE VI
BATCH SIZE PERFORMANCE ON MEAN SCORE (STANDARD DEVIATION).

	90dB SPL				100dB SPL			
Batch	0-F1	1-F1	Weighted-F1	Accuracy	0-F1	1-F1	Weighted-F1	Accuracy
64	0.825(0.027)	0.797(0.049)	0.811(0.037)	0.813(0.036)	0.924(0.021)	0.935(0.027)	0.930(0.024)	0.931(0.024)
96	0.832(0.047)	0.832(0.047)	0.832(0.043)	0.833(0.042)	0.926(0.049)	0.934(0.044)	0.931(0.046)	0.931(0.046)
128	0.846(0.038)	0.836(0.025)	0.841(0.031)	0.842(0.031)	0.917(0.046)	0.926(0.044)	0.922(0.044)	0.922(0.044)
160	0.780(0.039)	0.737(0.028)	0.758(0.033)	0.760(0.035)	0.913(0.024)	0.919(0.031)	0.917(0.028)	0.917(0.028)
192	0.797(0.028)	0.762(0.043)	0.780(0.035)	0.781(0.035)	0.911(0.022)	0.921(0.032)	0.916(0.028)	0.917(0.028)
224	0.848(0.040)	0.812(0.070)	0.830(0.054)	0.833(0.051)	0.913(0.050)	0.919(0.047)	0.917(0.048)	0.917(0.048)
256	0.774(0.045)	0.745(0.067)	0.759(0.055)	0.760(0.054)	0.900(0.019)	0.905(0.027)	0.903(0.024)	0.903(0.024)

TABLE VII
REPRESENTATION DIMENSION PERFORMANCE ON MEAN SCORE (STANDARD DEVIATION).

	90dB SPL				100dB SPL			
Dimension	0-F1	1-F1	Weighted-F1	Accuracy	0-F1	1-F1	Weighted-F1	Accuracy
16	0.770(0.027)	0.761(0.027)	0.765(0.022)	0.767(0.021)	0.897(0.022)	0.908(0.026)	0.903(0.024)	0.903(0.024)
32	0.806(0.033)	0.774(0.038)	0.790(0.034)	0.792(0.035)	0.941(0.040)	0.947(0.039)	0.944(0.039)	0.944(0.039)
48	0.796(0.019)	0.786(0.026)	0.791(0.019)	0.792(0.018)	0.913(0.050)	0.919(0.047)	0.917(0.048)	0.917(0.048)
64	0.846(0.038)	0.836(0.025)	0.841(0.031)	0.842(0.031)	0.917(0.046)	0.926(0.044)	0.922(0.044)	0.922(0.044)
80	0.839(0.025)	0.800(0.053)	0.820(0.037)	0.823(0.035)	0.912(0.030)	0.921(0.027)	0.917(0.028)	0.917(0.028)
96	0.812(0.022)	0.812(0.022)	0.812(0.021)	0.813(0.021)	0.909(0.054)	0.922(0.046)	0.916(0.049)	0.917(0.048)
112	0.828(0.017)	0.817(0.021)	0.823(0.018)	0.823(0.018)	0.910(0.028)	0.922(0.028)	0.917(0.028)	0.917(0.028)
128	0.787(0.068)	0.746(0.086)	0.767(0.072)	0.771(0.069)	0.898(0.025)	0.906(0.024)	0.903(0.024)	0.903(0.024)

[18] M. B. Sterman and T. Egner, "Foundation and practice of neurofeedback for the treatment of epilepsy," *Applied psychophysiology and biofeedback*, vol. 31, no. 1, p. 21, 2006.

[19] N. Lofthouse, L. E. Arnold, S. Hersch, E. Hurt, and R. DeBeus, "A review of neurofeedback treatment for pediatric adhd," *Journal of attention disorders*, vol. 16, no. 5, pp. 351–372, 2012.

[20] R. Milner, M. Lewandowska, M. Ganc, K. Cieśla, I. Niedziałek, and H. Skarżyński, "Slow cortical potential neurofeedback in chronic tinnitus therapy: a case report," *Applied psychophysiology and biofeedback*, vol. 41, no. 2, pp. 225–249, 2016.

[21] A. Supratak, H. Dong, C. Wu, and Y. Guo, "Deepsleepnet: A model for automatic sleep stage scoring based on raw single-channel eeg," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 11, pp. 1998–2008, 2017.

[22] K. G. Hartmann, R. T. Schirrmeister, and T. Ball, "Eeg-gan: Generative adversarial networks for electroencephalograhic (eeg) brain signals," *arXiv preprint arXiv:1806.01875*, 2018.

[23] S. Hwang, K. Hong, G. Son, and H. Byun, "Ezsl-gan: Eeg-based zero-shot learning approach using a generative adversarial network," in *2019 7th International Winter Conference on Brain-Computer Interface (BCI)*. IEEE, 2019, pp. 1–4.

[24] A. Phinyomark, F. Quaine, S. Charbonnier, C. Serviere, F. Tarpin-Bernard, and Y. Laurillau, "Emg feature evaluation for improving myoelectric pattern recognition robustness," *Expert Systems with applications*, vol. 40, no. 12, pp. 4832–4840, 2013.

[25] D. Coyle, G. Prasad, and T. M. McGinnity, "A time-series prediction approach for feature extraction in a brain-computer interface," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 13, no. 4, pp. 461–467, 2005.

[26] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.

[27] C.-C. Chang and C.-J. Lin, "Training v-support vector regression: theory and algorithms," *Neural computation*, vol. 14, no. 8, pp. 1959–1977, 2002.

[28] R. Schaette and D. McAlpine, "Tinnitus with a normal audiogram: physiological evidence for hidden hearing loss and computational model," *Journal of Neuroscience*, vol. 31, no. 38, pp. 13 452–13 457, 2011.

[29] D. J. Crisp and C. J. Burges, "A geometric interpretation of v-svm classifiers," in *Advances in neural information processing systems*, 2000, pp. 244–250.

[30] J. Olias, R. Martín-Clemente, M. A. Sarmiento-Vega, and S. Cruces, "Eeg signal processing in mi-bci applications with improved covariance matrix estimators," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 5, pp. 895–904, 2019.

[31] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "Eegnet: a compact convolutional neural

network for eeg-based braincomputer interfaces," *Journal of Neural Engineering*, vol. 15, no. 5, p. 056013, 2018. [Online]. Available: http://stacks.iop.org/1741-2552/15/i=5/a=056013

[32] S. R. Tibor, S. J. Tobias, F. L. D. Josef, G. Martin, E. Katharina, T. Michael, H. Frank, B. Wolfram, and B. Tonio, "Deep learning with convolutional neural networks for eeg decoding and visualization," *Human Brain Mapping*, vol. 38, no. 11, pp. 5391–5420. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/hbm.23730