

PAPER

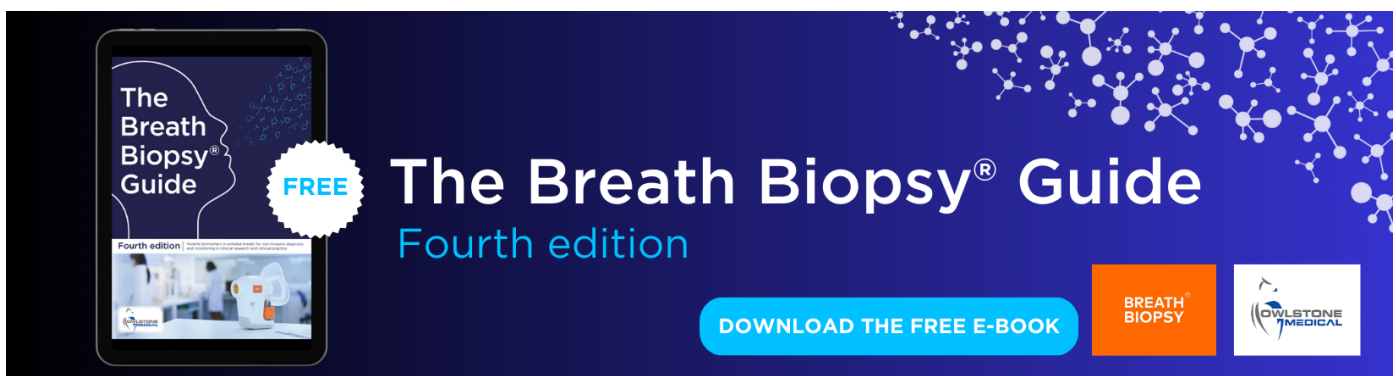
Fixed template network and dynamic template network: novel network designs for decoding steady-state visual evoked potentials

To cite this article: Xiaolin Xiao *et al* 2022 *J. Neural Eng.* **19** 056049

View the [article online](#) for updates and enhancements.

You may also like

- [To train or not to train? A survey on training of feature extraction methods for SSVEP-based BCIs](#)
R Zerafa, T Camilleri, O Falzon *et al.*
- [The effect of distractors on SSVEP-based brain-computer interfaces](#)
R Zerafa, T Camilleri, K P Camilleri *et al.*
- [A new multivariate empirical mode decomposition method for improving the performance of SSVEP-based brain-computer interface](#)
Yi-Feng Chen, Kiran Atal, Sheng-Quan Xie *et al.*



The Breath Biopsy® Guide
Fourth edition

FREE

DOWNLOAD THE FREE E-BOOK

BREATH BIOPSY

OWLSTONE MEDICAL



PAPER

Fixed template network and dynamic template network: novel network designs for decoding steady-state visual evoked potentials

Xiaolin Xiao^{1,2} , Lichao Xu² , Jin Yue², Baizhou Pan², Minpeng Xu^{1,2,*} and Dong Ming^{1,2,*}¹ Department of Biomedical Engineering, College of Precision Instruments and Optoelectronics Engineering, Tianjin University, Tianjin 300072, People's Republic of China² Academy of Medical Engineering and Translational Medicine, Tianjin University, Tianjin 300072, People's Republic of China

* Authors to whom any correspondence should be addressed.

E-mail: minpeng.xu@tju.edu.cn and richardming@tju.edu.cn**Keywords:** brain–computer interfaces, deep learning, SSVEP, EEG**Abstract**

Objective. Decomposition methods are efficient to decode steady-state visual evoked potentials (SSVEPs). In recent years, the brain–computer interface community has also been developing deep learning networks for decoding SSVEPs. However, there is no clear evidence that current deep learning models outperform decomposition methods on the SSVEP decoding tasks. Many studies lacked the comparison with state-of-the-art decomposition methods in a fair environment.

Approach. This study proposed a novel network design motivated by the works of decomposition methods. Fixed template network (FTN) and dynamic template network (DTN) are two novel networks combining the advantages of fixed templates and subject-specific templates. This study also proposed a data augmentation method for SSVEPs. This study compared the intra-subject classification performance of DTN and FTN with that of state-of-the-art decomposition methods on three public SSVEP datasets. **Main results.** The results show that both FTN and DTN achieved the suboptimal classification performance compared with state-of-the-art decomposition methods. **Significance.** Both network designs could enhance the decoding performance of SSVEPs, making them promising networks for improving the practicality of SSVEP-based applications.

1. Introduction

Brain–computer interfaces (BCIs) provide a new way to control machines without any physical intervention based on decoding information from brain activities [1, 2]. Electroencephalography (EEG) has been used in BCIs for decades due to its low-cost, non-invasive, and high temporal resolution characteristics. Advances in EEG-based BCIs promote practical BCI applications in many areas, e.g. quadcopter control [3], brain speller [4], stroke rehabilitation [5], and glaucoma detection [6]. EEG-based BCIs are promising to bring the next generation of human–machine interaction.

In recent years, steady-state visual evoked potentials (SSVEPs) have been widely used in vision-related BCIs. SSVEP is a type of visual evoked potentials (VEPs) that reflect the visual information processing

mechanism in the brain [7]. A flashing visual stimulus elicits an SSVEP response which oscillates at the same frequency of the stimulus and its harmonics [8]. Compared with other BCI paradigms, e.g. motor imagery (MI), SSVEP has a relatively high signal-to-noise ratio and less requirement for the training process [9, 10]. Researchers have developed many efficient decoding methods for SSVEP-based BCIs to achieve a high information transfer rate (ITR). These methods generally depend on matrix factorization. Canonical correlation analysis (CCA) and its variants have been widely used to detect frequency components in SSVEP with pre-generated templates (usually sine waves) [11–13]. These common templates can be replaced with templates generated in the training process by considering subject-specific information, e.g. task-related component analysis (TRCA) [14, 15] and its variant (TRCA-R) [16] which integrated the phase

information of stimulus into the pre-defined sinusoidal template. In addition, aiming at the redundancy of the spatial filters in TRCA, an optimized algorithm based on spatial filters derived from discriminative spatial patterns (DSP) and augmented EEG signals has been proposed as task discriminant component analysis (TDCA) [17]. A filter bank strategy could also further improve the ITRs of SSVEP-based BCIs [14, 18].

Besides methods based on matrix factorization, the BCI community has also been developing deep learning networks for decoding brain signals in recent years. Schirrmester *et al* designed ShallowConvNet for MI classification tasks [19], which imitated temporal and spatial filters that are usually used in decomposition methods, e.g. filter bank common spatial patterns (FBCSP) [20]. EEGNet further inherited the above architecture, showing better classification performance than FBCSP in multiple MI datasets [21, 22]. However, most networks were designed for MI classification tasks, and only a few of them were validated on the SSVEP classification tasks [23–27]. These works either used artificial features as inputs, e.g. spectrum features [27], or lacked the comparison with state-of-the-art decomposition methods [26]. Moreover, these works were usually validated on relatively small SSVEP datasets, which lack a comparative benchmark for evaluating the performance of deep learning networks. To our knowledge, there is no clear evidence that current deep learning models outperform decomposition methods on the SSVEP decoding tasks.

In this study, we proposed two novel network designs motivated by the works of decomposition methods. Dynamic template network (DTN) and fixed template network (FTN) are the novel networks combining the advantages of fixed templates (e.g. CCA) and subject-specific templates (e.g. TRCA). We also noticed that Li *et al* proposed a convolutional correlation analysis (Conv-CA) which had similar thinking about fusing the template design into the convolution network [28]. However, Conv-CA can not learn subject-specific templates from the training data without human intervention. Conv-CA also lacks fair comparison with decomposition methods, since the training of the network used data generated by the sliding window method from the full length of a trial (e.g. 5 s) while TRCA only used the data extracted from the beginning of the stimulus onset with the same length of the sliding window. This study will systematically test if the two novel methods could effectively improve the decoding performance of SSVEPs among three public datasets, compared with state-of-the-art decomposition methods in a fair environment. If so, the proposed network structures may further improve the practicality of SSVEP-based applications.

The organization of the rest of the work is as follows. Section 2 describes the related work and

section 3 describes the network architecture used in this work. Section 4 presents the study results. Section 5 discusses the results. Finally, section 6 concludes this study.

2. Related work

2.1. Extended canonical correlation analysis

Canonical correlation analysis (CCA) is a widely used unsupervised method in SSVEP-based BCIs [4, 11, 12]. CCA measures the underlying associations of two sets of variables via finding orthogonal linear combinations of the variables which have the maximum correlation with each other. Considering EEG data $\mathbf{X} \in \mathbb{R}^{N_c \times N_t}$ and reference signals $\mathbf{Y} \in \mathbb{R}^{M \times N_t}$, which are both zero-centered, CCA finds the weight vectors $\mathbf{u} \in \mathbb{R}^{N_c}$ and $\mathbf{v} \in \mathbb{R}^M$ to maximize the following objective function:

$$\hat{\mathbf{u}}, \hat{\mathbf{v}} = \underset{\mathbf{u}, \mathbf{v}}{\operatorname{argmax}} \frac{\mathbf{u}^T \mathbf{X} \mathbf{Y}^T \mathbf{v}}{\sqrt{\mathbf{u}^T \mathbf{X} \mathbf{X}^T \mathbf{u}} \sqrt{\mathbf{v}^T \mathbf{Y} \mathbf{Y}^T \mathbf{v}}}, \quad (1)$$

where N_c is the number of channels, M is the number of reference signals, and N_t is the number of time points. In SSVEPs, \mathbf{Y} is usually a pre-defined sinusoidal template $\mathbf{Y}_k \in \mathbb{R}^{2N_h \times N_t}$ with respect to the k th class:

$$\mathbf{Y}_k = \begin{bmatrix} \sin(2\pi f_k t) \\ \cos(2\pi f_k t) \\ \vdots \\ \sin(2\pi N_h f_k t) \\ \cos(2\pi N_h f_k t) \end{bmatrix}, \quad (2)$$

where f_k is the k th target frequency and N_h is the number of harmonics.

To utilize the class-specific templates $\bar{\mathbf{X}}_k$, Extended CCA (eCCA) designs the following feature extraction step instead of the normal canonical correlation coefficient [4]:

$$\bar{\mathbf{X}}_k = \frac{1}{N_k} \sum_{i=1}^{N_k} \mathbf{X}_k^i, \quad (3)$$

$$\mathbf{r}_k = \begin{bmatrix} \rho(\mathbf{X}^T \hat{\mathbf{u}}_{\mathbf{X}\mathbf{Y}_k}, \mathbf{Y}_k^T \hat{\mathbf{v}}_{\mathbf{Y}\mathbf{Y}_k}) \\ \rho(\mathbf{X}^T \hat{\mathbf{u}}_{\mathbf{X}\bar{\mathbf{X}}_k}, \bar{\mathbf{X}}_k^T \hat{\mathbf{u}}_{\mathbf{X}\bar{\mathbf{X}}_k}) \\ \rho(\mathbf{X}^T \hat{\mathbf{u}}_{\mathbf{X}\mathbf{Y}_k}, \bar{\mathbf{X}}_k^T \hat{\mathbf{u}}_{\mathbf{X}\mathbf{Y}_k}) \\ \rho(\mathbf{X}^T \hat{\mathbf{u}}_{\bar{\mathbf{X}}_k \mathbf{Y}_k}, \bar{\mathbf{X}}_k^T \hat{\mathbf{u}}_{\bar{\mathbf{X}}_k \mathbf{Y}_k}) \\ \rho(\bar{\mathbf{X}}_k^T \hat{\mathbf{u}}_{\mathbf{X}\bar{\mathbf{X}}_k}, \bar{\mathbf{X}}_k^T \hat{\mathbf{v}}_{\mathbf{X}\bar{\mathbf{X}}_k}) \end{bmatrix}, \quad (4)$$

where N_k is the number of available trials for the k th class and \mathbf{X}_k^i is the i th trial of the training data for the k th class. $\hat{\mathbf{u}}_{\mathbf{X}\mathbf{Y}_k}$ denotes the CCA weight vector \mathbf{u} computed in equation (1) with respect to \mathbf{X} and \mathbf{Y}_k and $\hat{\mathbf{v}}_{\mathbf{Y}\mathbf{Y}_k}$ denotes the CCA weight vector \mathbf{v} computed in equation (1) with respect to \mathbf{X} and \mathbf{Y}_k . $\rho(\cdot, \cdot)$ computes the Pearson's correlation coefficient.

For any test data \mathbf{X} , the predicted label \hat{k} would be the label of the maximum correlation coefficients among K classes:

$$\rho_k = \sum_{i=1}^5 \text{sign}(\mathbf{r}_{ki}) \mathbf{r}_{ki}^2$$

$$\hat{k} = \underset{k}{\text{argmax}} \rho_k, \quad (5)$$

where $\text{sign}(\cdot)$ is the sign function and \mathbf{r}_{ki} is the i th component of the vector \mathbf{r}_k .

2.2. Task-related component analysis

Task-related component analysis (TRCA) extracts task-related components from the test data and predicts the labels by comparing these components with the class-specific templates [14]. TRCA is a supervised method that requires labeled training data of a single subject. Specifically, TRCA computes the inter-trial similarity of the k th class as follows:

$$\mathbf{S}_k = \sum_{\substack{i_1 \neq i_2 \\ i_1, i_2=1}}^{N_k} \mathbf{X}_k^{i_1} (\mathbf{X}_k^{i_2})^T, \quad (6)$$

where i_1 and i_2 are the available trial indices for the k th class.

TRCA finds the weight vector $\mathbf{w}_k \in \mathbb{R}^{N_c}$ to maximize the following objective function:

$$\hat{\mathbf{w}}_k = \underset{\mathbf{w}_k}{\text{argmax}} \frac{\mathbf{w}_k \mathbf{S}_k \mathbf{w}_k^T}{\mathbf{w}_k \mathbf{Q}_k \mathbf{w}_k^T}, \quad (7)$$

where \mathbf{Q}_k is the sum of covariances for the k th class:

$$\mathbf{Q}_k = \sum_{i=1}^{N_k} \mathbf{X}_k^i (\mathbf{X}_k^i)^T. \quad (8)$$

In the test phase, TRCA requires the class-specific template $\bar{\mathbf{X}}_k$ in equation (3) and the predicted label \hat{k} of the test data \mathbf{X} would be:

$$\rho_k = \rho(\mathbf{X}^T \hat{\mathbf{w}}_k, \bar{\mathbf{X}}_k^T \hat{\mathbf{w}}_k)$$

$$\hat{k} = \underset{k}{\text{argmax}} \rho_k. \quad (9)$$

Nakanishi *et al* [14] also proposed ensemble TRCA (eTRCA), which had better performance than TRCA, by combining $\hat{\mathbf{w}}_k$ of all classes into an ensemble spatial filter $\hat{\mathbf{W}} \in \mathbb{R}^{N_c \times K}$, where K is the number of classes:

$$\hat{\mathbf{W}} = [\hat{\mathbf{w}}_1, \dots, \hat{\mathbf{w}}_k, \dots, \hat{\mathbf{w}}_K], \quad (10)$$

and the predicted label \hat{k} of the test data \mathbf{X} would be:

$$\rho_k = \rho(\mathbf{X}^T \hat{\mathbf{W}}, \bar{\mathbf{X}}_k^T \hat{\mathbf{W}})$$

$$\hat{k} = \underset{k}{\text{argmax}} \rho_k. \quad (11)$$

Wong *et al* proposed an improved algorithm named eTRCA-R, projecting \mathbf{X}_k^i onto the subspace of \mathbf{Y}_k firstly and then extracting eTRCA spatial filters with projected signals [16]. The projected signal $\tilde{\mathbf{X}}_k^i$ is:

$$\tilde{\mathbf{X}}_k^i = \mathbf{X}_k^i \mathbf{P}_k, \quad (12)$$

where $\mathbf{P}_k = \mathbf{Y}_k^T (\mathbf{Y}_k \mathbf{Y}_k^T)^{-1} \mathbf{Y}_k$. Spatial filters could be computed according to the above eTRCA procedures via replacing \mathbf{X}_k^i with $\tilde{\mathbf{X}}_k^i$ in equation (6).

2.3. Discriminative spatial patterns

Discriminative spatial patterns (DSP) was initially used for extracting spatial filters of movement-related potentials [29]. However, DSP can also be used for SSVEPs with proper feature extraction steps. Like TRCA, DSP requires labeled training data of a single subject for estimating class-specific templates $\bar{\mathbf{X}}_k$ in equation (3) and the mean template $\bar{\mathbf{X}}$:

$$\bar{\mathbf{X}} = \frac{1}{N} \sum_{i=1}^N \mathbf{X}^i, \quad (13)$$

where $N = \sum_{k=1}^K N_k$ is the total number of trials.

Different from TRCA that estimates \mathbf{w}_k for the k th class, DSP finds the global weight vector $\mathbf{w} \in \mathbb{R}^{N_c}$ to maximize the following objective function:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmax}} \frac{\mathbf{w} \left(\sum_{k=1}^K \mathbf{S}_k \right) \mathbf{w}^T}{\mathbf{w} \left(\sum_{k=1}^K \mathbf{Q}_k \right) \mathbf{w}^T}, \quad (14)$$

where \mathbf{S}_k and \mathbf{Q}_k are:

$$\mathbf{S}_k = N_k (\bar{\mathbf{X}}_k - \bar{\mathbf{X}}) (\bar{\mathbf{X}}_k - \bar{\mathbf{X}})^T$$

$$\mathbf{Q}_k = \sum_{i=1}^{N_k} (\mathbf{X}_k^i - \bar{\mathbf{X}}_k) (\mathbf{X}_k^i - \bar{\mathbf{X}}_k)^T. \quad (15)$$

DSP maximizes the discrepancy between classes meanwhile minimizes the dissimilarity between the data and the template for each class. For any test data \mathbf{X} , the predicted label \hat{k} would be:

$$\rho_k = \rho \left((\mathbf{X} - \bar{\mathbf{X}})^T \hat{\mathbf{w}}, (\bar{\mathbf{X}}_k - \bar{\mathbf{X}})^T \hat{\mathbf{w}} \right)$$

$$\hat{k} = \underset{k}{\text{argmax}} \rho_k. \quad (16)$$

2.4. Task discriminant component analysis

Recently, Liu *et al* designed an optimized method named TDCA [17]. This algorithm aims at reducing the redundancy of the spatial filters in TRCA. Essentially, TDCA augments original EEG trials and constructs spatial filters based on DSP. Specially, for any training sample $\mathbf{X} \in \mathbb{R}^{N_c \times N_t}$, there is the augmented EEG trial $\tilde{\mathbf{X}}$:

$$\tilde{\mathbf{X}} = [\mathbf{X}^T, \mathbf{X}_1^T, \dots, \mathbf{X}_m^T]^T, \quad (17)$$

where $\tilde{\mathbf{X}} \in \mathbb{R}^{(m+1)N_c \times N_t}$ is the augmented EEG trial. $\mathbf{X}_m \in \mathbb{R}^{N_c \times N_t}$ denotes the EEG trial delayed by m points and represents the data copy from time $1 + m$ to $N_t + m$ in data points. N_t is the upper limit of the data points in the testing sample. The testing sample delayed by m points is deducted as:

$$\mathbf{X}_m = [\mathbf{X}'_m, \mathbf{O}], \quad (18)$$

where \mathbf{X}'_m denotes the data copy from time point $m + 1$ to N_t , and \mathbf{O} is the matrix padded with zeros. $\mathbf{X}_m \in \mathbb{R}^{N_c \times N_t}$ is guaranteed.

TDCA method then projects the augmented EEG trial $\tilde{\mathbf{X}}$ onto the subspace spanned by the reference signal defined by the equation (2). Based on \mathbf{Y}_k , the projected sample $\tilde{\mathbf{X}}_p^k$ is decomposed as:

$$\tilde{\mathbf{X}}_p^k = \tilde{\mathbf{X}}(\mathbf{Y}_k)^T(\mathbf{Y}_k(\mathbf{Y}_k)^T)^{-1}(\mathbf{Y}_k), \quad (19)$$

here decompose $(\mathbf{Y}_k)^T$ into $\mathbf{Q}_k\mathbf{R}_k$ by the QR decomposition, then equation (19) can be further simplified as:

$$\tilde{\mathbf{X}}_p^k = \tilde{\mathbf{X}}\mathbf{Q}_k(\mathbf{Q}_k)^T. \quad (20)$$

The final EEG augmented trial $\mathbf{X}_a^k \in \mathbb{R}^{(m+1)N_c \times 2N_t}$ for the k th class is constructed as:

$$\mathbf{X}_a^k = [\tilde{\mathbf{X}}, \tilde{\mathbf{X}}_p^k]. \quad (21)$$

For each training trial, there is a corresponding augmented EEG sample with respect to the k th class. TDCA utilizes those augmented EEG samples to obtain the spatial filter bank $\hat{\mathbf{W}}^k \in \mathbb{R}^{(m+1)N_c \times (m+1)N_c}$ and the average augmented template $\bar{\mathbf{X}}_a^k$ for the k th class from the k th EEG sample.

The first L components of $\hat{\mathbf{W}}^k$ are extracted for constructing a new spatial filter $\tilde{\mathbf{W}}^k \in \mathbb{R}^{(m+1)N_c \times L}$, and the correlation coefficient ρ_k of the testing sample \mathbf{X} with respect to the k th class is calculated as:

$$\rho_k = \rho((\tilde{\mathbf{W}}^k)^T \mathbf{X}_a^k, (\tilde{\mathbf{W}}^k)^T \bar{\mathbf{X}}_a^k), \quad (22)$$

where \mathbf{X}_a^k is the augmented EEG sample with respect to the k th class, the \hat{k} corresponding to the max ρ_k is chosen as the predicted label for the testing sample.

2.5. Filter bank framework

Due to the characteristics of SSVEPs that elicit the base frequency and its harmonics, the filter bank analysis is quite efficient to improve the accuracy in the decoding of SSVEPs. The basic idea is to decompose original data \mathbf{X} to several sub-band components $\tilde{\mathbf{X}}_b$ and combine outputs of sub-bands to predict the final label. eCCA [4], eTRCA [14], DSP [29], and TDCA [17] could be embedded into this framework and named FBeCCA, FBeTRCA, FBDSP and FBDTCA

in this study, respectively. Considering ρ_k^b the correlation coefficient of the b th sub-band for the k th class, the final correlation coefficient $\tilde{\rho}_k$ is:

$$\tilde{\rho}_k = \sum_{b=1}^{N_b} w(b) \rho_k^b$$

$$w(b) = b^{-m} + n, \quad (23)$$

where N_b is the total number of sub-bands and $w(\cdot)$ computes the weight for each sub-band component. The parameters m and n are usually determined with a grid-search method using an offline analysis.

The predicted label \hat{k} of the test data \mathbf{X} would be:

$$\hat{k} = \underset{k}{\operatorname{argmax}} \tilde{\rho}_k. \quad (24)$$

Authors in [18] analyzed three filter bank designs and suggested a bottom-up way to generate all sub-bands. Each sub-band covers multiple harmonics with a high cutoff frequency at the upper-bound frequency f_{ub} of the SSVEP components. For example, suppose the lowest target stimulus frequency is f_1 , the n th sub-band would be within $[nf_1, f_{ub}]$.

2.6. EEGNet

EEGNet was chosen as the baseline neural network model in this work. This network retains temporal and spatial convolution layers proposed in [19] that imitate temporal and spatial filters in FBCSP [20]. It also introduces depthwise separable convolution to reduce the number of training parameters. EEGNet was initially designed for MI tasks [21] and some works also showed that it can be used for decoding SSVEPs [23, 30]. However, these works were validated on small SSVEP datasets and lacked comparisons with state-of-the-art decomposition methods. The effectiveness of EEGNet in decoding SSVEPs still needs further verification. More details of EEGNet can be found in <https://github.com/vlawnern/arleeegmodels>.

3. Methods

3.1. Fixed template network

The FTN was motivated by the design of eCCA that compares the data with pre-defined templates. Figure 1 illustrates the architecture of the FTN, considering the 3-class classification problem as an example. Three fixed templates and an input were provided to the FTN. The shape of the input was $N_c \times N_t$, where N_c is the number of channels and N_t is the number of time points. The shape of the template was $2N_h \times N_t$, where N_h is the number of harmonics and the template is formulated as equation (2). All templates and input were applied with the feature extraction module, which was designed for the template and input respectively. Next, cosine similarity between the input and templates were computed along the last dimension and their return values were

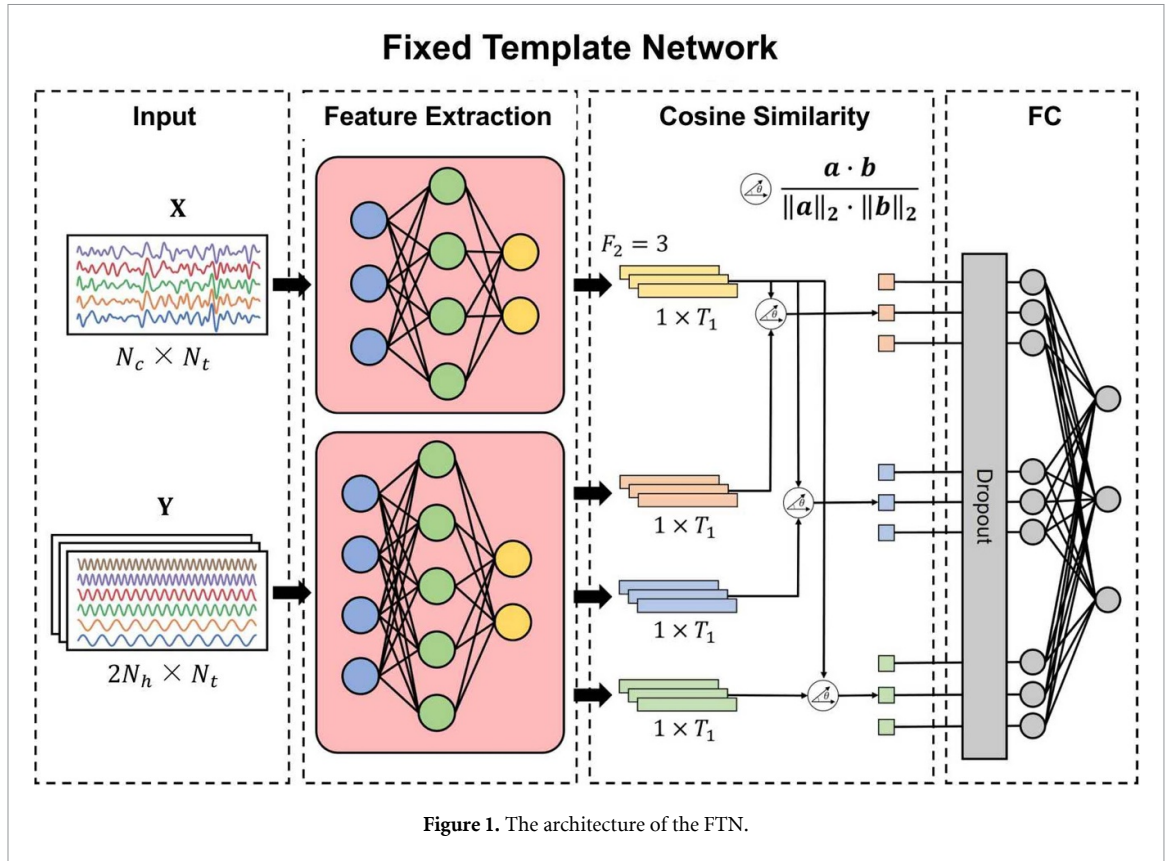


Figure 1. The architecture of the FTN.

Table 1. Feature extraction module for the input.

| Layer | Input | Output | Out_channels | Kernel | Stride | Padding |
|----------------|-------------------|-------------------|--------------|------------|------------|---------|
| InstanceNorm2d | $(1, N_c, N_t)$ | $(1, N_c, N_t)$ | | | | |
| Conv2d | $(1, N_c, N_t)$ | (F_1, N_c, N_t) | F_1 | $(1, K_1)$ | $(1, 1)$ | SAME |
| Conv2d | (F_1, N_c, N_t) | $(F_2, 1, N_t)$ | F_2 | $(N_c, 1)$ | $(1, 1)$ | VALID |
| Conv2d | $(F_2, 1, N_t)$ | $(F_2, 1, T_1)$ | F_2 | $(1, K_2)$ | $(1, K_2)$ | VALID |
| BatchNorm2d | $(F_2, 1, T_1)$ | $(F_2, 1, T_1)$ | | | | |
| Tanh | $(F_2, 1, T_1)$ | $(F_2, 1, T_1)$ | | | | |
| Conv2d | $(F_2, 1, T_1)$ | $(F_2, 1, T_1)$ | F_2 | $(1, K_3)$ | $(1, 1)$ | SAME |

Table 2. Feature extraction module for the template.

| Layer | Input | Output | Out_channels | Kernel | Stride | Padding |
|-------------|--------------------|--------------------|--------------|-------------|------------|---------|
| Conv2d | $(1, 2N_h, N_t)$ | $(F_1, 2N_h, N_t)$ | F_1 | $(1, K_1)$ | $(1, 1)$ | SAME |
| Conv2d | $(F_1, 2N_h, N_t)$ | $(F_2, 1, N_t)$ | F_2 | $(2N_h, 1)$ | $(1, 1)$ | VALID |
| Conv2d | $(F_2, 1, N_t)$ | $(F_2, 1, T_1)$ | F_2 | $(1, K_2)$ | $(1, K_2)$ | VALID |
| BatchNorm2d | $(F_2, 1, T_1)$ | $(F_2, 1, T_1)$ | | | | |
| Tanh | $(F_2, 1, T_1)$ | $(F_2, 1, T_1)$ | | | | |
| Conv2d | $(F_2, 1, T_1)$ | $(F_2, 1, T_1)$ | F_2 | $(1, K_3)$ | $(1, 1)$ | SAME |

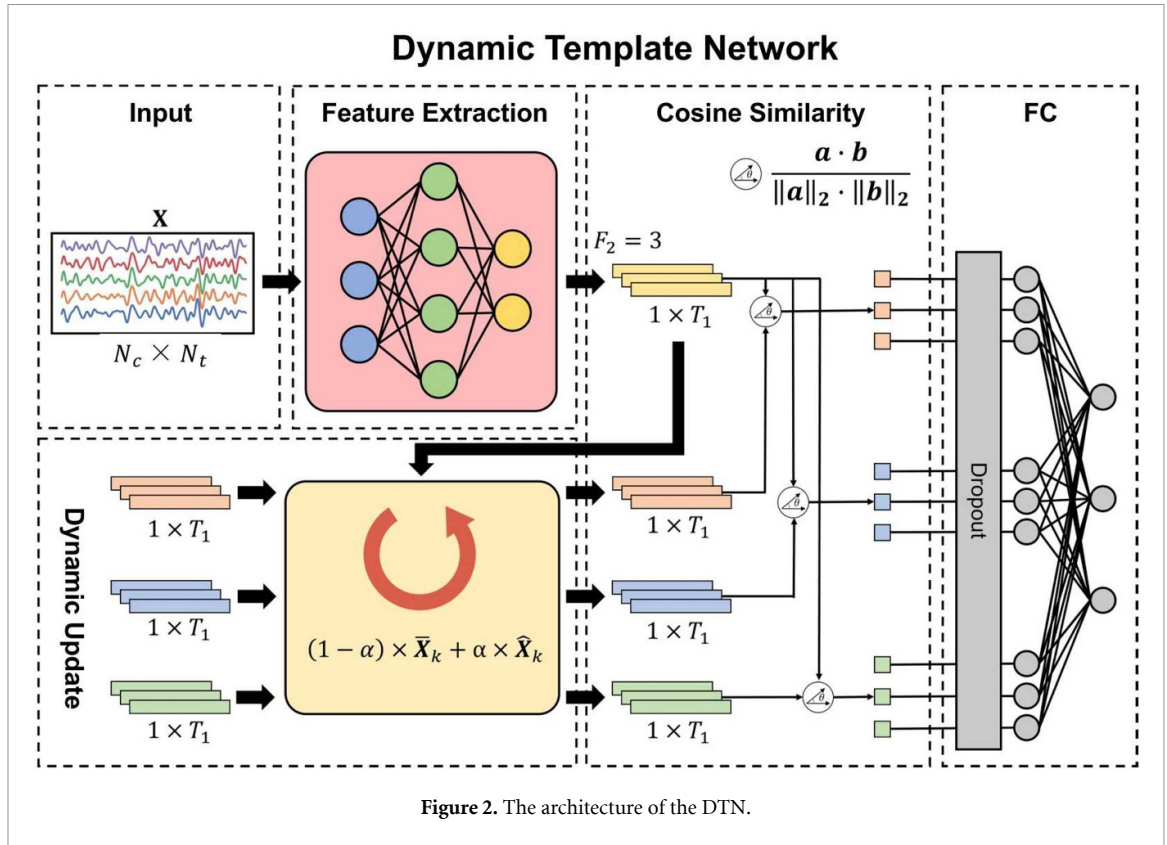
stacked together. A fully connected layer was finally used to classify the input label.

The architecture of feature extraction modules designed for the input and template are shown in tables 1 and 2, respectively. In table 1, the input was firstly applied with an instance normalization layer to remove any instance-related bias. Then, a convolution layer with a kernel $(1, K_1)$ was used to imitate temporal filters. The following convolution layer with a kernel $(N_c, 1)$ was to imitate spatial filters. The

instance normalization layer was removed in table 2 since templates are artificially generated.

3.2. Dynamic template network

The DTN was motivated by the design of TRCA and DSP that compares the data with subject-specific templates. Figure 2 illustrates the architecture of the DTN, considering the 3-class classification problem as an example. In the DTN, all templates and input were applied with the feature extraction module



depicted in table 1 instead of their separate feature extraction modules in the FTN. The rest of operations were the same as those in the FTN.

The most important part of the DTN is how to dynamically update templates based on each batch of the inputs in the training process. Figure 3 illustrates the template update process of the DTN, considering the 3-class classification problem as an example. The initial dynamic templates were set to zero. A batch of the inputs (two inputs colored in orange and one input colored in blue) was provided to the DTN. First, their batch templates were calculated by averaging the corresponding samples in the batch. Then, dynamic templates were updated with the exponential moving average as follows:

$$\bar{\mathbf{X}}_k = (1 - \alpha) \times \bar{\mathbf{X}}_k + \alpha \times \hat{\mathbf{X}}_k, \quad (25)$$

where $\bar{\mathbf{X}}_k$ is the dynamic template of the k th class and $\hat{\mathbf{X}}_k$ is the mean batch template of the k th class. The default value of α is 0.1. However, α can be $1/N_{batches}$ where $N_{batches}$ is the tracked cumulative number of batches. In that case, the cumulative moving average is used which is nearly the simple average of all inputs of the k th class. The template colored in green in figure 3 remains unchanged since there is no corresponding sample in the batch.

3.3. Data augmentation

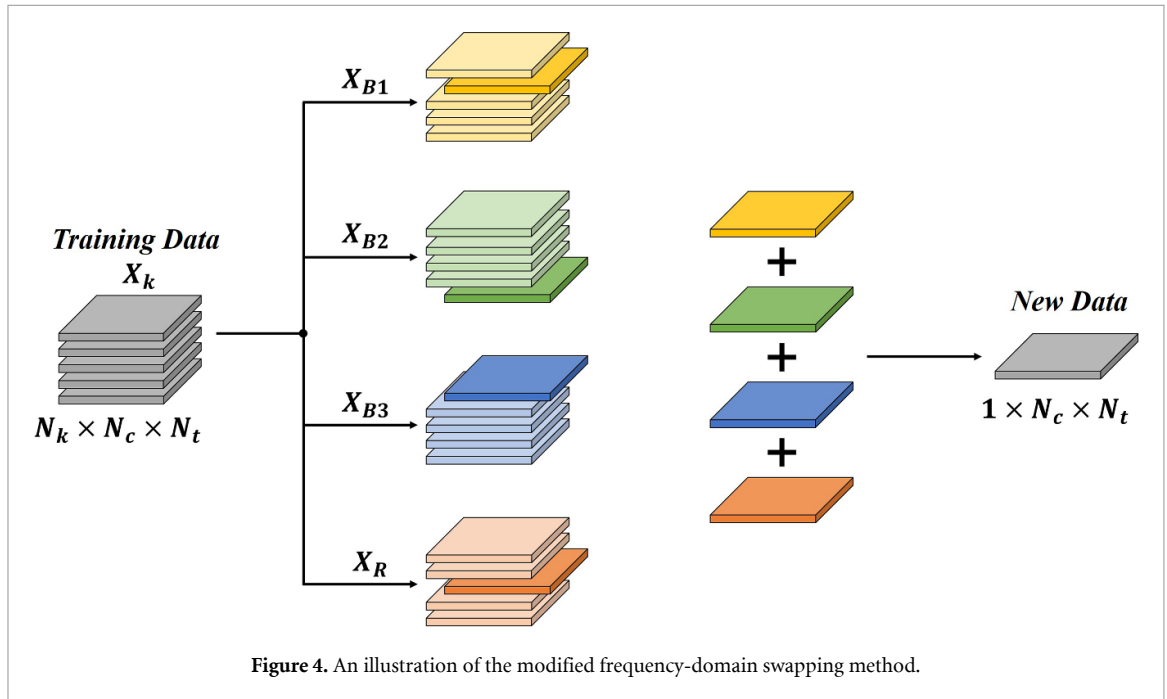
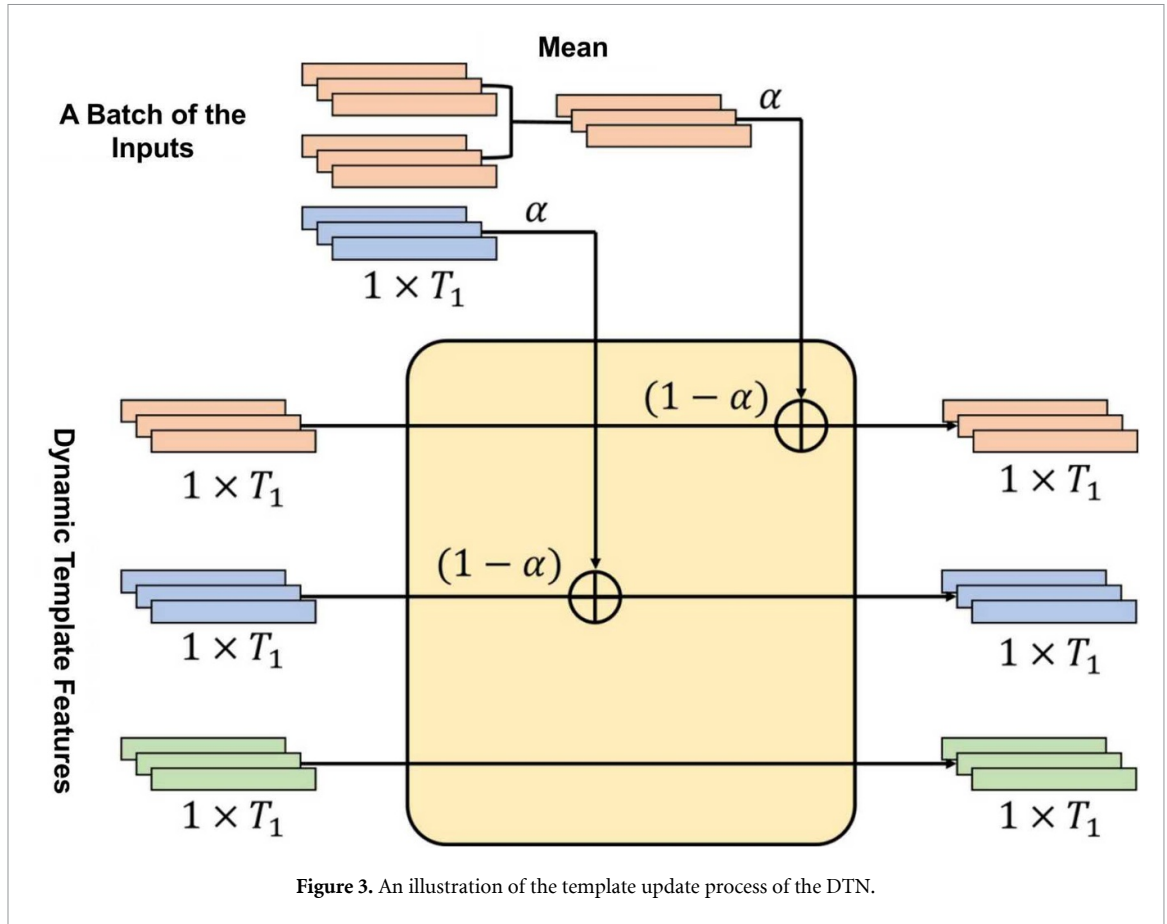
Since the amount of training data of each subject is limited, this study used the modified frequency-domain swapping method to augment the

available data for the designed template networks. The frequency-domain swapping method was originally proposed for MI tasks [31], which combines the sub-bands of the original data to generate the new data. The random exchange in the original method is only carried out in the specified sub-bands while the residual part of the signal is not included. The modified frequency-domain swapping method considers keeping the residual part of the signal since it contains information not included in the sub-bands. Figure 4 illustrates the modified frequency-domain swapping method with three sub-bands as an example. Considering the training data of the k th class $\mathbf{X}_k \in \mathbb{R}^{N_k \times N_c \times N_t}$, three sub-band data \mathbf{X}_{B1} , \mathbf{X}_{B2} , and \mathbf{X}_{B3} were generated with the bandpass filtering. \mathbf{X}_R were the residual part of \mathbf{X}_k such that $\mathbf{X}_R = \mathbf{X}_k - \mathbf{X}_{B1} - \mathbf{X}_{B2} - \mathbf{X}_{B3}$. A new sample could be generated from the above sub-band database by combining randomly selected sub-band components together.

3.4. Datasets

This study used the EEG data from three public SSVEP datasets Nakanishi2015 (figure 5) [32], Yijun2016 (figure 6) [33] and BETA (figure 7) [34]. Nakanishi2015 and Yijun2016 can be downloaded using the MOABB package [35]. BETA can be downloaded from <http://bci.med.tsinghua.edu.cn/download.html>. Table 3 lists the details of the datasets.

Nakanishi2015 is a dataset containing 12 SSVEP stimuli, coded in a joint frequency and phase



modulation (JFPM) approach. The frequencies of the 12 stimuli were from 9.25 Hz to 14.75 Hz with 0.5 Hz as an offset between the two stimuli. The phases of the 12 stimuli were from 0 to 2π with 0.5π as an offset between the two stimuli. The visual stimuli are presented in the form of the square wave, and

SSVEPs are elicited by the alternating white and black flickering frames. For specified frequency f with an initial phase ϕ , the stimulus sequence $s(f, \phi, i)$ can be defined as:

$$s(f, \phi, i) = \text{square}[2\pi f(i/\text{RefreshRate}) + \phi], \quad (26)$$

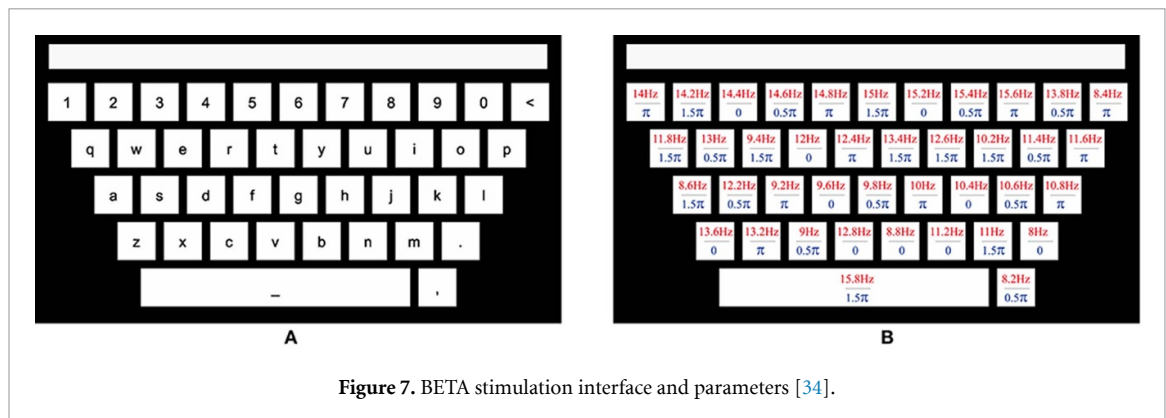
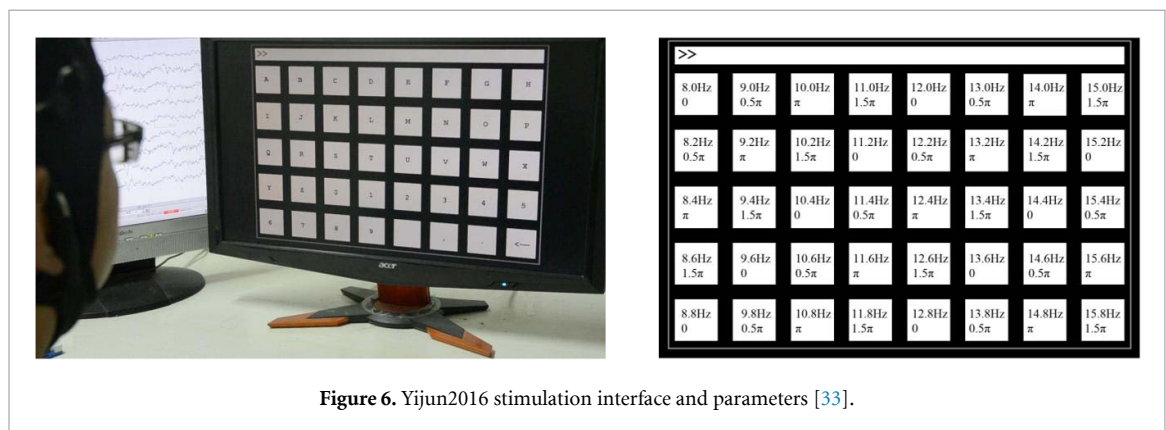
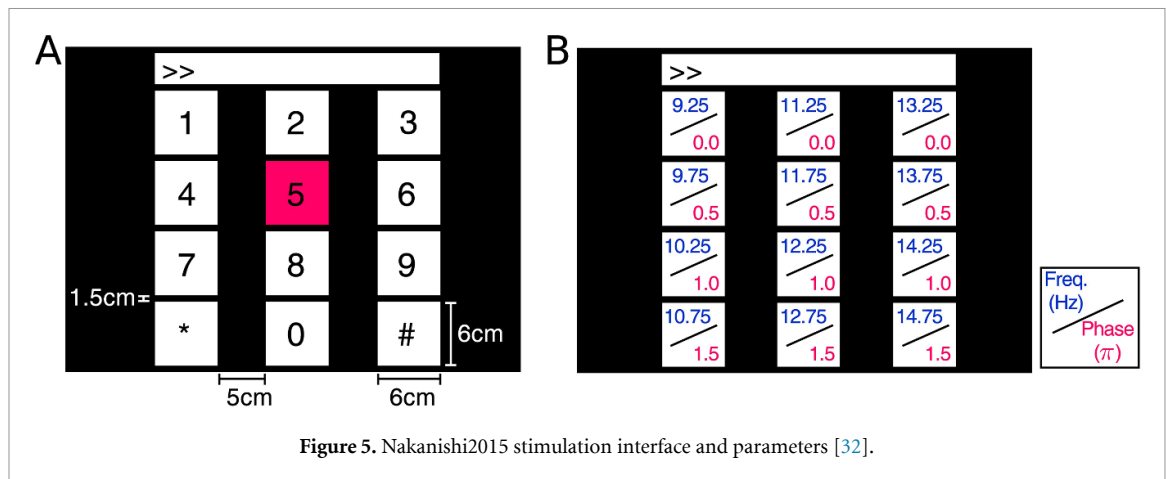


Table 3. Details of SSVEP datasets.

| Dataset | # of Classes | # of Subjects | Trial duration (s) | # of Channels | Sampling rate (Hz) | # of trials per subject |
|---------------|--------------|---------------|--------------------|---------------|--------------------|-------------------------|
| Nakanishi2015 | 12 | 10 | 4 | 8 | 256 | 180 |
| Yijun2016 | 40 | 35 | 5 | 60 | 250 | 240 |
| BETA | 40 | 70 | 2–3 | 60 | 250 | 160 |

where the square wave has a duty cycle of 50% with levels 0 and 1, and i indicates the frame index. 10 participants are recruited for this dataset. For each participant, there are 15 blocks with 12 trials for each block (one trial for each class of target). Each trial begins with a 1 s cue and the stimuli flicker for 4 s. All data epochs were bandpass filtered from 6 to 80 Hz

with an infinite impulse response filter, using the `filtfilt` function in MATLAB.

Yijun2016 is a dataset containing 40 SSVEP stimuli modulated with the JFPM approach. The frequencies of the 40 stimuli were from 8 to 15.8 Hz with 0.2 Hz as an offset between the two stimuli. The phases of the 40 stimuli were from 0 to 2π with 0.5π as an

offset between the two stimuli. The 5×8 stimulus matrix of the speller includes 40 characters including 26 English alphabets, 10 digits, and four symbols (dot, comma, backspace, and space). The visual flickers on the LCD monitor are presented by a sampled stimulation method. For specified frequency f with an initial phase ϕ , the stimulus sequence $s(f, \phi, i)$ can be defined as:

$$s(f, \phi, i) = \frac{1}{2}(1 + \sin[2\pi f(i/\text{RefreshRate}) + \phi]), \quad (27)$$

where $\sin()$ generates a sine wave and i indicates the frame index. The refresh rate of the LCD monitor is 60 Hz. 35 participants are recruited for this dataset. For each participant, there are six blocks with 40 trials for each block (one trial for each class of target). The participants are instructed to focus on the target for 5 s after a 0.5 s cue at the start of each trial and avoid eyes blinking. Then before the next trial, the screen is blank for 0.5 s. EEG data were acquired with an amplifier frequency passband ranged from 0.15 to 200 Hz and a notch filter at 50 Hz was applied in the recording.

BETA is a dataset containing 40 SSVEP stimuli modulated with the JFPM approach. The frequencies of the 40 stimuli were from 8 to 15.8 Hz with 0.2 Hz as an offset between the two stimuli. The phases of the 40 stimuli were from 0 to 2π with 0.5π as an offset between the two stimuli. The graphic interface is designed to resemble the traditional QWERT keyboard. The 40 targets are the same as those in the Yijun2016 dataset. The stimulus sequence of each flicker is the same as equation (27). Each trial begins with a 0.5 s cue and ends with a 0.5 s blank for rest. The flickering lasts for at least 2 s for the first 15 participants (S1–S15), and at least 3 s for the remaining 55 participants (S16–S70). EEG data were acquired with an amplifier frequency passband ranged from 0.15 to 200 Hz and a notch filter at 50 Hz was applied in the recording. All data epochs were bandpass filtered from 3 to 100 Hz using the eegfilt function in EEGLAB [36].

3.5. Experiment settings

The decoding experiment aims at comparing the classification performance of the decomposition methods and the deep learning models in the supervised situation. Eight channels of Nakanishi2015 (PO7, PO3, POz, PO4, PO8, O1, Oz, O2) and nine channels of Yijun2016 and BETA (Pz, PO5, PO3, POz, PO4, PO6, O1, Oz, O2) were used in this study. The data epochs were extracted in $[0s + t_{\text{delay}}, 1s + t_{\text{delay}}]$ where time 0 indicated the stimulus onset. The parameter t_{delay} , which reflected a latency delay in the visual pathway, was selected based on previous studies with slight differences among the selected datasets (Nakanishi2015: 0.135 s; Wang2016: 0.14 s; BETA: 0.13 s) [32–34]. All data epochs were resampled to 250 Hz. A leave-one-out cross-validation scheme was used for

evaluating the performance of methods. For example, a subject of Nakanishi2015 had 15 blocks and a block included 12 trials covering 12 classes. In one fold of the cross-validation, a block was selected as the testing set. Another block from the remaining blocks was selected as the validation set and the rest 13 blocks were selected as the training set.

Decomposition methods in section 2 were implemented in Python. N_h in equation (2) was 5. The training set and validation set were concatenated to establish models for decomposition methods since these methods do not need the validation set to avoid overfitting. The maximum value of N_h in section 2.5 was 3. All sub-band data were filtered with a Chebyshev Type I filter by the filtfilt function in Scipy [37]. Sub-bands were selected as 8–90, 16–90, and 24–90 Hz. The parameters of sub-bands are inherited from the code of original authors of TRCA, which is available at <https://github.com/mnakanishi/TRCA-SSVEP/blob/master/src/filterbank.m>. For TDCA, the delay point m in the equation (17) is set as 5 and the number of chosen components L in the equation (22) is set as 8.

Due to the limit of the training data, the modified frequency-domain swapping method in section 3.3 was used to augment the training data for FTN and DTN. The sub-bands for the data augmentation were 8–15.8, 16–31.6, 24–47.4, 32–63.2, and 40–79 Hz. We augmented 20 samples per class for each subject, combining the training set and validation set as the basic database. The augmented data were used as the new validation set while the original training set and validation set were concatenated as the new training set. For example, the size of the training set, validation set, and test set of a subject in BETA was 80, 40, and 40, respectively. Thus, the size of augmented validation set was 800 (40 classes). The size of new training set was 120 (80 + 40) and the size of test set remains unchanged. In this case, decomposition methods and neural networks have the same training data for each subject.

EEGNet, FTN and DTN were implemented in the PyTorch framework. The parameters of EEGNet were consistent with those in [23]. For the feature extraction module in FTN and DTN, F_1 was 3, F_2 was 120, K_1 and K_3 were 9, and K_2 was 2. The dropout rate was 0.95. α in equation (25) was 0.1. Two-stage training scheme was employed in this study. The model was firstly trained on the data of all subjects and then was fine-tuned on the data of the specific subject. In the first stage, the optimizer was Adam with the initial learning rate set to 0.001 (0.01 for EEGNet), dynamically adjusted by the CosineAnnealingLR scheduler. The training batch size was 256. The model was trained for a maximum of 600 epochs with early stopping and the best model parameters on the validation set were kept for the next stage. In the fine-tuning stage, the batch size was 32 for each subject and the model was trained for a maximum of 100 epochs.

Models were trained and tested on a personal computer with an AMD Ryzen 5 1600 and an NVIDIA GTX 1080. The code is available for reproducibility at <https://github.com/Mrswolf/template-networks>.

The balanced accuracy (BA) was calculated for evaluating the performance of methods, such an indicator avoids the classification accuracy bias caused by the imbalance of the data set categories. If there are K number of classes in total, the BA is defined as:

$$BA = \frac{1}{K} \sum_{k=1}^K \frac{\hat{N}_k}{N_k}, \quad (28)$$

where N_k is the total number of observations in the k th class, and \hat{N}_k is the number of observations correctly identified as the same class label.

3.6. Statistical analysis

This study compared accuracies of decomposition methods and novel designed template networks with statistical methods introduced in [35]. For intra-subject classification results, either a one-tailed permutation-based t-test (for datasets with less than 20 subjects) or a Wilcoxon signed-rank test is used for each pair of methods within each dataset, generating a p -value for the null hypothesis that one algorithm is not better than the other. Stouffer's method for meta-analysis is used here to generate a final p -value for each pair of algorithms by combining the p -values from datasets with different conditions (e.g. N_t and N_b). The significance level in this study was 0.001. Considering the possible problem of false positives for multiple comparisons, all the significances were calculated after Bonferroni corrected.

4. Results

This section presents the results of the BA of matrix-decomposition methods and deep learning models, and illustrates the statistical analysis results among those methods.

4.1. Decoding algorithms based on matrix decomposition

Figure 8 illustrates the results of accuracy from different matrix decomposition algorithms under different data lengths (N_t) among Nakanishi2015, Yijun2016, and BETA dataset. FBDSP, FBeTRCA, FBeTRCA-R, and FBTDCA has a accuracy as 89.7%, 91.9%, 92.8%, and 92.2%, respectively for the results of Nakanishi2015 with $N_t = 0.5$ s, and 97.3%, 97.7%, 97.8%, and 92.4%, respectively when $N_t = 1.0$ s for the same dataset. For the results of Yijun2016, FBDSP, FBeTRCA, FBeTRCA-R, and FBTDCA has a accuracy as 75.0%, 77.7%, 78.0%, and 84.1%, respectively with $N_t = 0.5$ s, and 92.4%, 93.4%, 93.8%, and 96.2%, respectively with $N_t = 1.0$ s. For the results of BETA, FBDSP, FBeTRCA, FBeTRCA-R, and FBTDCA has

an accuracy as 55.9%, 59.6%, 60.6%, and 67.1%, respectively with $N_t = 0.5$ s, and 73.1%, 74.7%, 75.8%, and 81.8%, respectively with $N_t = 1.0$ s. With good reason, the BA of FBTDCA, FBeTRCA-R, FBeTRCA, and FBDSP decrease in the listed sequence under most scenarios. Additionally, FBTDCA has the highest BA under Yijun2016 and BETA with a relatively obvious advantage compared to the other matrix decomposition methods.

The results of statistical analysis described from method section 3.6 in figure 9 show that FBTDCA outperforms other decomposition methods under supervised learning situations, and this method is one of the optimal methods in the current matrix-decomposition-based SSVEP supervised decoding algorithm. FBeTRCA-R was the suboptimal method comparing to FBeTRCA and FBDSP. FBTDCA and FBeTRCA-R are chosen as the benchmark algorithms for comparison with the deep learning models in the following sections.

4.2. Comparison of deep learning models and the benchmark algorithms

Figure 10 shows the classification accuracy of the deep learning models (EEGNet, DTN, FTN) and the benchmark algorithms (FBTDCA, FBeTRCA-R) under different data lengths without the data augmentation. Compared to the benchmark methods, the decoding effect of EEGNet performs worse significantly, while the two network structures DTN and FTN proposed in the previous section have close or better results. Under different data lengths from the BETA dataset, the classification effect of FTN is better than that of FBeTRCA-R while slightly worse than that of FBTDCA. FBTDCA, FBeTRCA-R, DTN, and FTN achieves a accuracy as 67.1%, 60.6%, 60.1%, and 66.2%, respectively with $N_t = 0.5$ s, and 81.8%, 75.7%, 76.3%, and 78.3% respectively with $N_t = 1$ s.

Figure 11 illustrates the results of deep learning models with the data augmentation method under different datasets. For dataset Nakanishi2015, DTN and DTN-A has the BA as 90.2% and 90.3% respectively with $N_t = 0.5$ s, where DTN-A has a 0.1% improvement. In the same scenario, FTN and FTN-A has the BA as 85.0% and 87.0% respectively, where FTN-A has a 2.0% improvement. When $N_t = 1$ s, DTN and DTN-A achieves the accuracy as 95.9% and 96.5%, with a 0.6% increase of accuracy for DTN-A, and FTN and FTN-A achieves the accuracy as 93.0% and 93.7%, with an approximately 0.7% increase of accuracy for FTN-A.

For dataset Yijun2016, DTN and DTN-A has the BA as 78.3% and 79.7% respectively with $N_t = 0.5$ s, where DTN-A has a 1.4% improvement. In the same scenario, FTN and FTN-A has the BA as 78.8% and 81.0% respectively, where FTN-A has a 2.2% improvement. When $N_t = 1$ s, DTN and DTN-A achieves the accuracy as 92.7% and 93.7%, with a 1.0% increase of accuracy for DTN-A, and FTN and

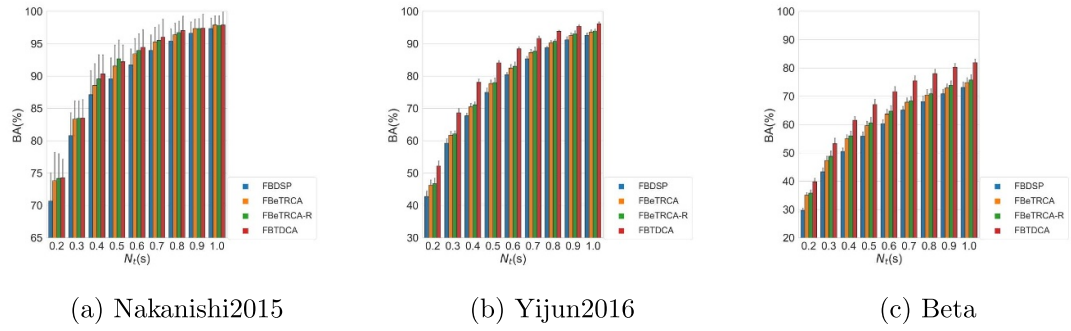


Figure 8. Comparing the BA (%) of different matrix decomposition algorithms under different data lengths (N_t).

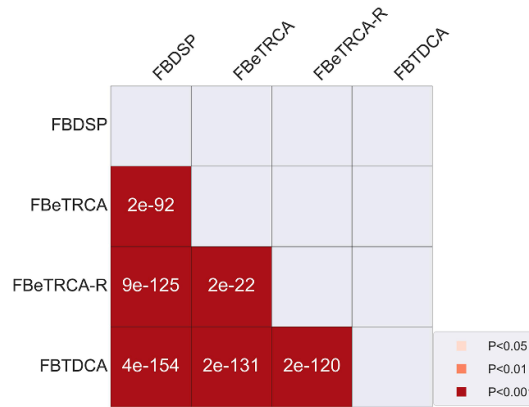


Figure 9. Statistical analyses of classification results for listed decomposition methods. The red squares indicate that the method in the row outperforms the method in the column, the number in each square is the p -value.

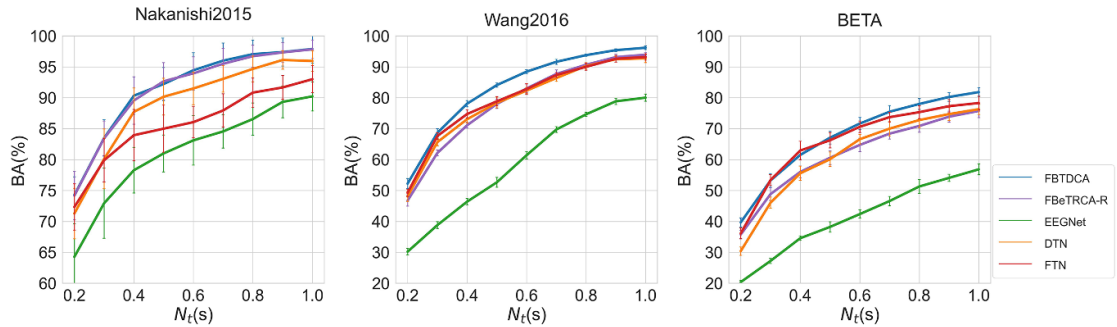


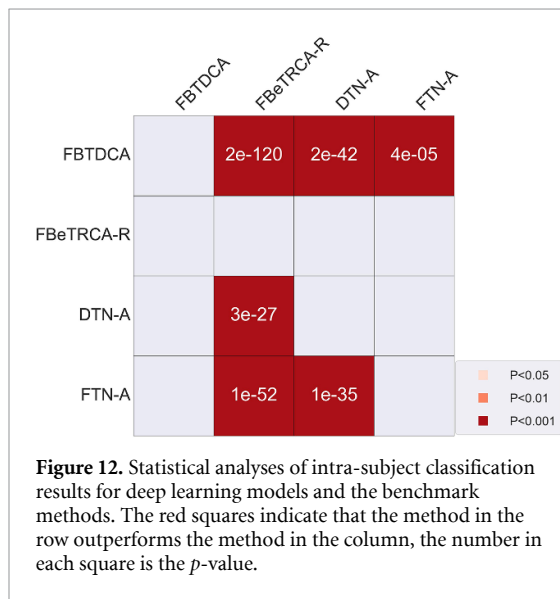
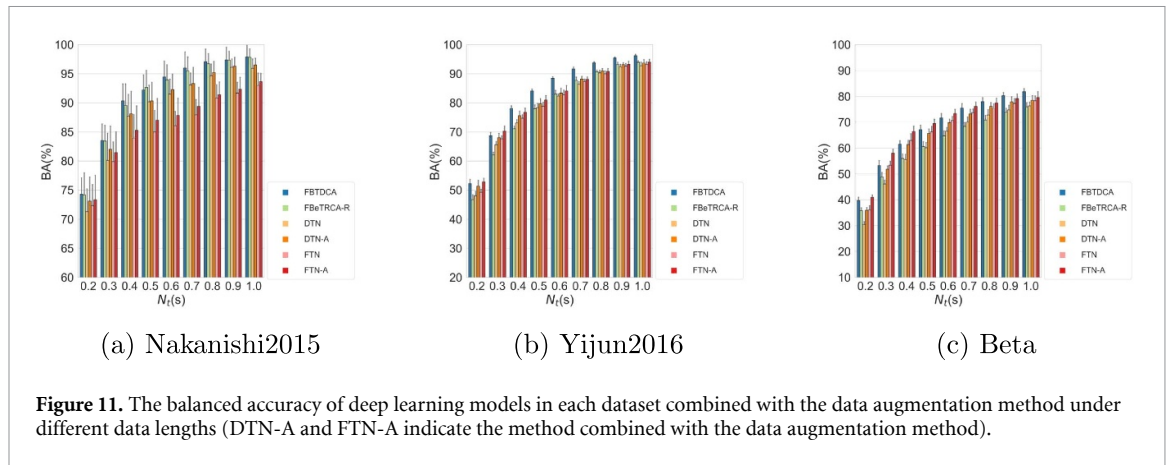
Figure 10. Comparing the BA (%) of different deep learning models under different data lengths (N_t).

FTN-A achieves the accuracy as 93.2% and 94.0%, with an approximately 0.8% increase of accuracy for FTN-A.

For dataset BETA, DTN and DTN-A has the BA as 60.1% and 65.6% respectively with $N_t = 0.5$ s, where DTN-A has a 5.5% improvement. In the same scenario, FTN and FTN-A has the BA as 66.2% and 69.4% respectively, where FTN-A has a 3.2% improvement. When $N_t = 1$ s, DTN and DTN-A achieves the accuracy as 76.3% and 78.5%, with a 2.2% increase of accuracy for DTN-A, and FTN and FTN-A achieves the accuracy as 78.3% and 79.6%, with an approximately 1.3% increase of accuracy for FTN-A.

Figure 12 is the meta-analysis results among deep learning models and the benchmark methods, showing that FBTDCa significantly outperforms DTN and FTN. DTN significantly outperforms FBeTRCA-R. FTN significantly outperforms FBeTRCA-R and DTN.

Figures 13 and 14 show the analyses of DTN templates and average templates for subjects 8 and 56 in the BETA dataset, respectively. Specifically, the stored template features in DTN were replaced with the subject's average template features, computed via the feature extraction module in the DTN. Figures 13(a) and 14(a) show that DTN template features are



highly correlated with the subject's average template features. Figures 13(b) and 14(b) illustrate the power spectral densities of these template features, indicating that most template features contain the corresponding template frequencies. Figure 15 illustrates the classification results of DTN with replaced template features under different datasets, showing that the average template features do not improve the classification accuracy.

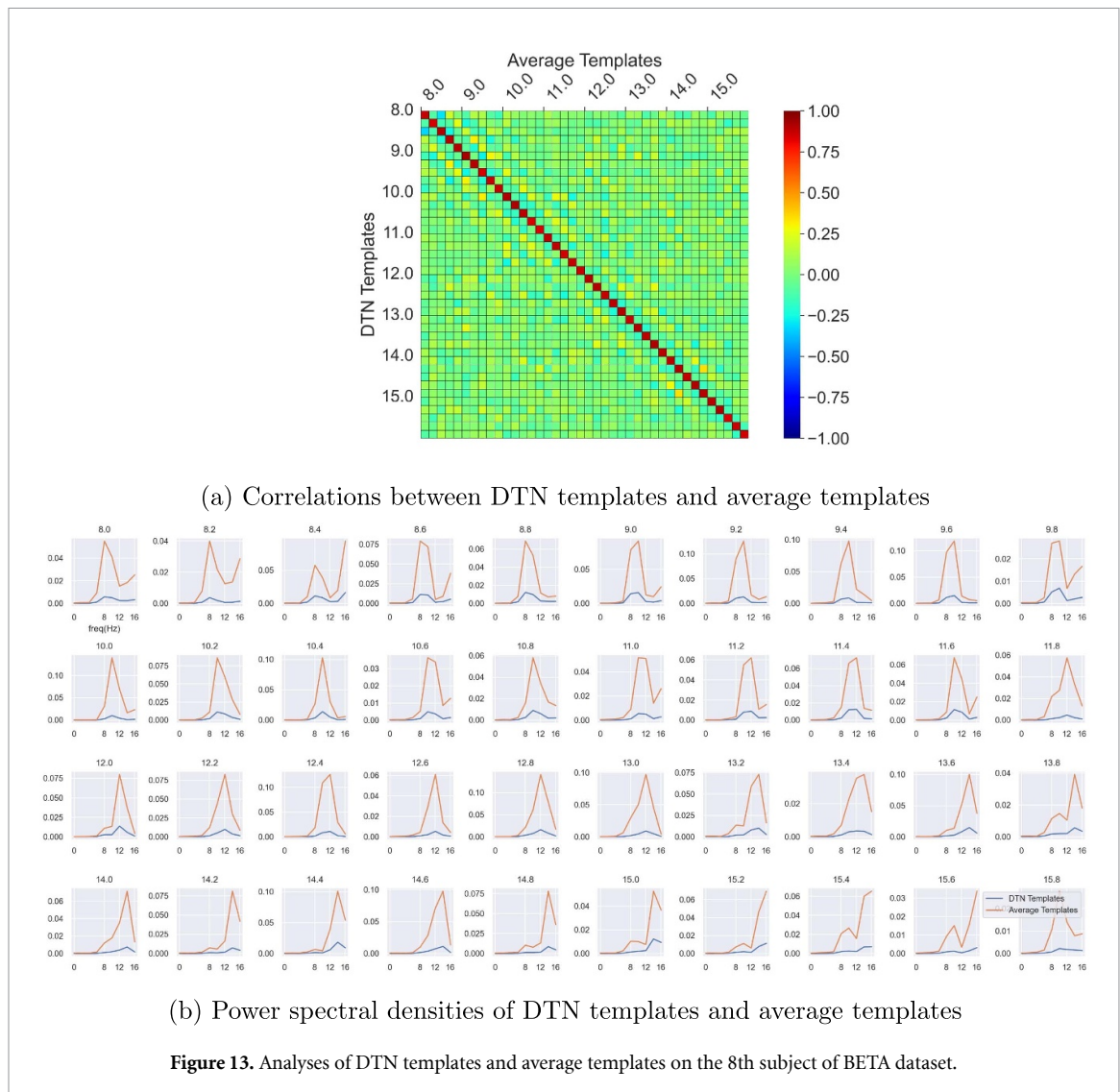
5. Discussion

Deep learning models, e.g. ShallowConvNet and EEGNet, have achieved better decoding performance on the MI decoding tasks in recent years. However, to our knowledge, there is no clear evidence that current deep learning models outperform decomposition methods on the SSVEP decoding tasks. Decomposition methods, e.g. eTRCA, still have state-of-the-art performance in decoding SSVEPs. Although some studies validated networks on the SSVEP decoding task, there are still some problems that need to be addressed [23–27]. First, these studies lacked

the comparison with state-of-the-art decomposition methods. Second, these studies were usually validated on relatively small SSVEP datasets which may not be enough to compare the performance of different methods. Moreover, due to the small number of available training trials, some studies used the data generated by the sliding window method from the full length of a trial, which may not be a fair comparison for decomposition methods since most of them only used the data extracted from the beginning of the stimulus onset with the same length of the sliding window [23, 27, 28].

The successful design of ShallowConvNet and EEGNet makes us think about fusing the advantages of decomposition methods into the design of the network for SSVEPs. The two novel network structures are motivated by the most important part in these decomposition methods, which is the usage of templates for SSVEPs. The advantages of fixed templates (e.g. CCA) and subject-specific templates (e.g. eTRCA) are combined in the structures. FTN and DTN are the corresponding implementations for these two types of template. We noticed that Li *et al* proposed Conv-CA which had similar thinking about fusing the template design into the convolution network [28]. However, Conv-CA lacks the ability to learn subject-specific templates from the training data without the human intervention. This problem has been solved in the dynamic template update process described in section 3.2.

Results in sections 4.1 and 4.2 show the balanced accuracies of decomposition methods and network structures across different time windows. The accuracies of all methods were improved with longer time windows. Section 4.1 compares the BA of four decomposition methods (FBTDSP, FBETRCA, FBETRCA-R, and FBTDCA). On dataset Nakanishi2015 which is a small dataset with only nine subjects, all the traditional methods could achieve an accuracy above 80% even with a short time window ($N_t = 0.3$ s), meaning most subjects in Nakanishi2015 are familiar with SSVEPs. Besides, the decoding performance between the four methods was



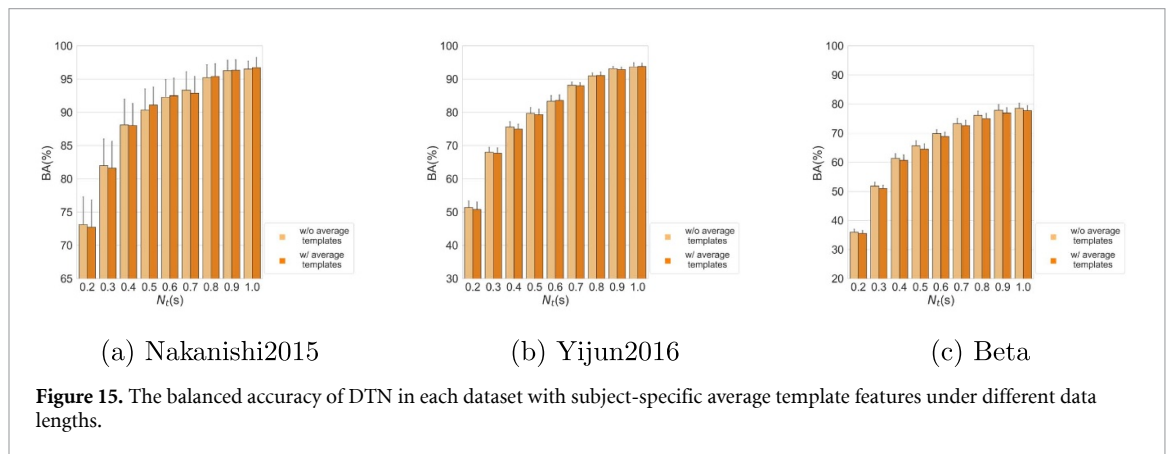
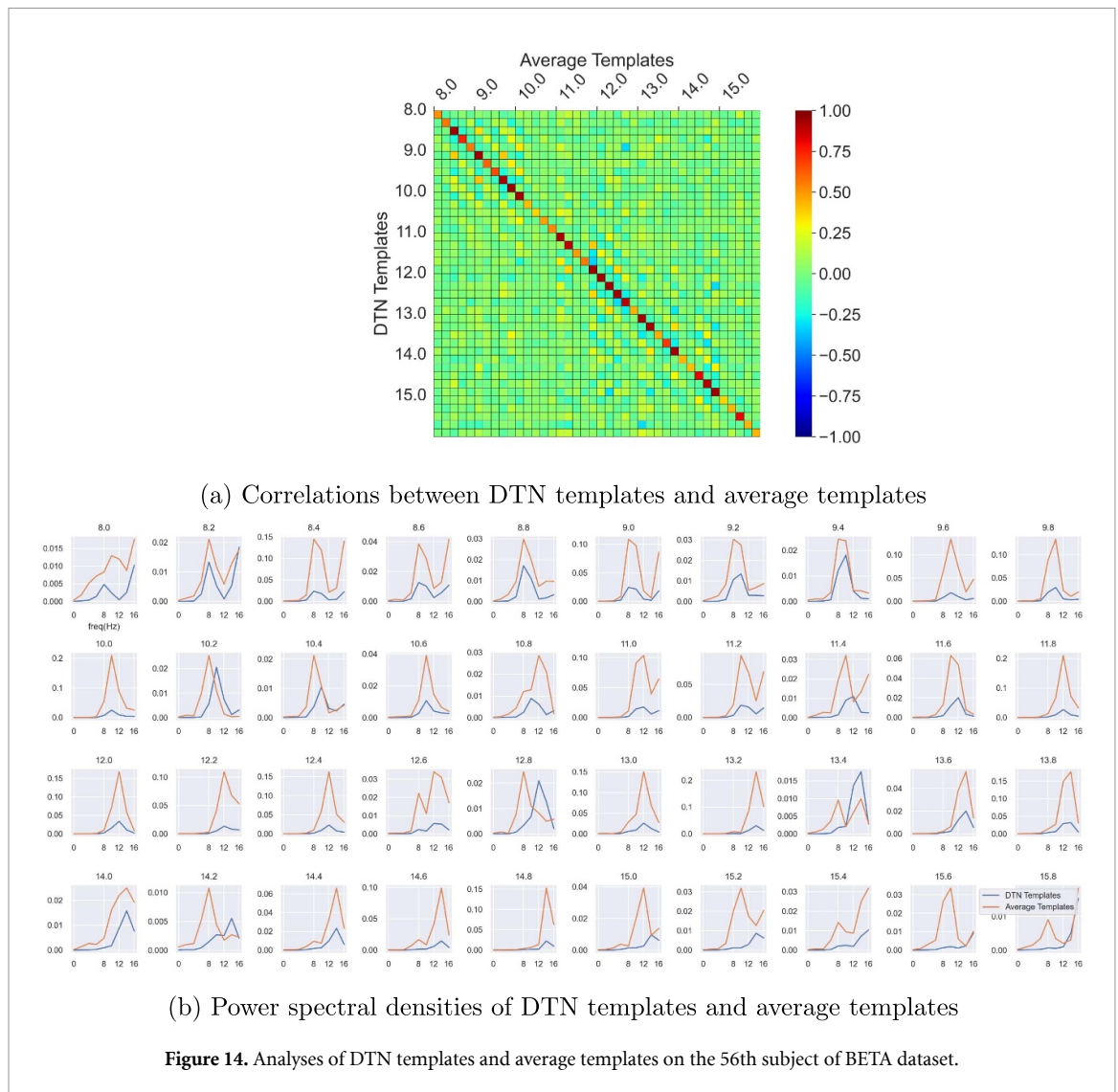
close, suggesting that this dataset may not be enough to compare the performance of different methods. Figures 8(b) and 8(c) show the results on dataset Yijun2016 and BETA, respectively. These datasets have larger numbers of subjects than that of Nakanishi2015. FBTDCa shows better decoding performance across all time windows on these two datasets, and figure 9 proves that FBTDCa outperforms other decomposition methods. FBTDCa and the suboptimal FBeTRCA-R were selected as the benchmark methods in section 4.2.

Section 4.2 compares the BA of benchmark decomposition methods and deep learning models. Results in figure 10 show that FTN and DTN are significantly better than EEGNet, suggesting the effectiveness of fusing the template idea into the network designs. Although the available training data of deep learning models for each subject is less than that of decomposition methods (1 block was selected as the validation set in the deep learning training process), FTN and DTN still achieve the comparable classification performance, especially on Wang2016 and BETA. The relatively large difference between

deep learning models and decomposition methods on dataset Nakanishi2015 may contribute to the overfitting problem. The parameters of FTN and DTN chosen here may be too large for the 12-class classification problem.

Results in figure 11 show the performance of template networks with the data augmentation method. On the one hand, the available training data of deep learning models in the fine-tuning phase for each subject is consistent with that of decomposition algorithms. On the other hand, the augmented validation set combined with the early stopping strategy can avoid the overfitting of deep learning models. The increased training sample size from the data augmentation method leads to an improvement in the BA of DTN and FTN compared to the models without the data augmentation in most scenarios, and such improvement is most obvious in the BETA dataset. The classification performance of FTN-A was even better than that of FBTDCa in small data lengths (e.g. 0.4 s).

Results in figures 13 and 14 show that the DTN template features are quite similar with the



average template features. The difference in the amplitude of power spectral densities suggests that there exist biases between these templates, which may be introduced via the batch normalization layer in the feature extraction module. Thus, replacing the DTN template features with the subject's average template features directly may decrease the decoding performance of the network, which indicated in figure 15.

The statistical results in figure 12 show that the proposed template networks significantly outperformed FBeTRCA-R which is suboptimal among decomposition methods. FBeTRCA is still better than other methods. The effectiveness of FBeTRCA suggests that decomposition methods are still important in decoding signals with a small sample size. Decomposition methods may also have

advantages in training time. On our platform, FBT-DCA requires about 1.14 s training time for a subject under 0.5 s data length while FTN and DTN trained on the GPU require about 23 and 22 s, respectively.

However, template networks could also achieve similar decoding performance for SSVEPs in a fair comparable environment, suggesting they have many possibilities for performance improvement in the future. Temporal convolution that mimics temporal filtering could be easily embedded into the novel algorithms so as to get rid of the limitation of fixed filter banks. It is also possible to group samples in the batch by their similarities with dynamic templates, which would extend the template networks to the scenario of training on the data of multiple subjects. Meanwhile, the inference time of DTN and FTN on the CPU under 0.5 s data length is about 0.0035 s and 0.01 s, respectively, which enables them to decode online BCI commands. It could be expected that a novel template network that combines DTN and FTN will be applied in the future to further increase the efficiency in decoding SSVEPs.

6. Conclusion

This study proposed a novel network design for decoding SSVEPs. This study systematically evaluated the intra-subject classification performance of FTN, DTN, EEGNet, and state-of-the-art decomposition methods on three public datasets. The results show that two novel template networks achieved the sub-optimal classification performance among decomposition methods. FTN and DTN could enhance the decoding performance of SSVEPs, making them promising for improving the practicality of SSVEP-based applications.

Acknowledgments

L Xu and X Xiao contributed equally to the study conception, literature search, and writing. All authors contributed to manuscript revision, read and approved the submitted version.

This research was funded by the Ministry of Science and Technology of China (2022ZD0210200); National Natural Science Foundation of China (Nos. 62106170, 62122059, 61976152, 81925020); Introduce Innovative Teams of 2021 'New High School 20 Items' Project (2021GXRC071).

ORCID iDs

Xiaolin Xiao  <https://orcid.org/0000-0002-3516-561X>

Lichao Xu  <https://orcid.org/0000-0003-2717-2809>

References

- [1] Wolpaw J R, Birbaumer N, McFarland D J, Pfurtscheller G and Vaughan T M 2002 Brain-computer interfaces for communication and control *Clin. Neurophysiol.* **113** 767–91
- [2] Gu X, Cao Z, Jolfaei A, Xu P, Wu D, Jung T-P and Lin C-T 2021 EEG-based brain-computer interfaces (BCIs): a survey of recent studies on signal sensing technologies and computational intelligence approaches and their applications *IEEE/ACM Trans. Comput. Biol. Bioinform.* **18** 1645–66
- [3] LaFleur K, Cassady K, Doud A, Shades K, Rogin E and He B 2013 Quadcopter control in three-dimensional space using a noninvasive motor imagery-based brain-computer interface *J. Neural Eng.* **10** 046003
- [4] Chen X, Wang Y, Nakanishi M, Gao X, Jung T-P and Gao S 2015 High-speed spelling with a noninvasive brain-computer interface *Proc. Natl Acad. Sci.* **112** E6058–67
- [5] Jure F A, Carrere L C, Gentiletti G G and Tabernig C B 2016 BCI-FES system for neuro-rehabilitation of stroke patients *J. Phys.: Conf. Ser.* **705** 012058
- [6] Nakanishi M, Wang Y-T, Jung T-P, Zao J K, Chien Y-Y, Diniz-Filho A, Daga F B, Lin Y-P, Wang Y and Medeiros F A 2017 Detecting glaucoma with a portable brain-computer interface for objective assessment of visual function loss *JAMA Ophthalmol.* **135** 550–7
- [7] Wang Y, Wang R, Gao X, Hong B and Gao S 2006 A practical VEP-based brain-computer interface *IEEE Trans. Neural Syst. Rehabil. Eng.* **14** 234–40
- [8] Müller-Putz G R, Scherer R, Brauneis C and Pfurtscheller G 2005 Steady-state visual evoked potential (SSVEP)-based communication: impact of harmonic frequency components *J. Neural Eng.* **2** 123
- [9] Wang Y, Gao X, Hong B, Jia C and Gao S 2008 Brain-computer interfaces based on visual evoked potentials *IEEE Eng. Med. Biol. Mag.* **27** 64–71
- [10] Vialatte F-B, Maurice M, Dauwels J and Cichocki A 2010 Steady-state visually evoked potentials: focus on essential paradigms and future perspectives *Prog. Neurobiol.* **90** 418–38
- [11] Lin Z, Zhang C, Wu W and Gao X 2006 Frequency recognition based on canonical correlation analysis for SSVEP-based BCIs *IEEE Trans. Biomed. Eng.* **53** 2610–4
- [12] Zhang Y U, Zhou G, Jin J, Wang X and Cichocki A 2014 Frequency recognition in SSVEP-based BCI using multisets canonical correlation analysis *Int. J. Neural Syst.* **24** 1450013
- [13] Jiao Y, Zhang Y, Wang Y, Wang B, Jin J and Wang X 2018 A novel multilayer correlation maximization model for improving CC-based frequency recognition in SSVEP brain-computer interface *Int. J. Neural Syst.* **28** 1750039
- [14] Nakanishi M, Wang Y, Chen X, Wang Y-T, Gao X and Jung T-P 2017 Enhancing detection of SSVEPs for a high-speed brain speller using task-related component analysis *IEEE Trans. Biomed. Eng.* **65** 104–12
- [15] Zhang Y, Guo D, Li F, Yin E, Zhang Y, Li P, Zhao Q, Tanaka T, Yao D and Xu P 2018 Correlated component analysis for enhancing the performance of SSVEP-based brain-computer interface *IEEE Trans. Neural Syst. Rehabil. Eng.* **26** 948–56
- [16] Wong C M, Wang B, Wang Z, Lao K F, Rosa A and Wan F 2020 Spatial filtering in SSVEP-based BCIs: Unified framework and new improvements *IEEE Trans. Biomed. Eng.* **67** 3057–72
- [17] Liu B, Chen X, Shi N, Wang Y, Gao S and Gao X 2021 Improving the performance of individually calibrated SSVEP-BCI by task-discriminant component analysis *IEEE Trans. Neural Syst. Rehabil. Eng.* **29** 1998–2007
- [18] Chen X, Wang Y, Gao S, Jung T-P and Gao X 2015 Filter bank canonical correlation analysis for implementing a high-speed SSVEP-based brain-computer interface *J. Neural Eng.* **12** 046008
- [19] Schirrmester R T, Springenberg J T, Fiederer I D J, Glasstetter M, Eggensperger K, Tangermann M, Hutter F, Burgard W and Ball T 2017 Deep learning with

- convolutional neural networks for EEG decoding and visualization *Hum. Brain Mapp.* **38** 5391–420
- [20] Ang K K, Chin Z Y, Zhang H and Guan C 2008 Filter bank common spatial pattern (FBCSP) in brain-computer interface 2008 *IEEE Int. Joint Conf. on Neural Networks (IEEE World Congress on Computational Intelligence)* (IEEE) pp 2390–7
- [21] Lawhern V J, Solon A J, Waytowich N R, Gordon S M, Hung C P and Lance B J 2018 EEGNet: a compact convolutional neural network for EEG-based brain-computer interfaces *J. Neural Eng.* **15** 056013
- [22] Xu L, Xu M, Ke Y, An X, Liu S and Ming D 2020 Cross-dataset variability problem in EEG decoding with deep learning *Front. Hum. Neurosci.* **14** 00103
- [23] Waytowich N, Lawhern V J, Garcia J O, Cummings J, Faller J, Sajda P and Vettel J M 2018 Compact convolutional neural networks for classification of asynchronous steady-state visual evoked potentials *J. Neural Eng.* **15** 066031
- [24] Attia M, Hettiarachchi I, Hossny M and Nahavandi S 2018 A time domain classification of steady-state visual evoked potentials using deep recurrent-convolutional neural networks 2018 *IEEE 15th Int. Symp. on Biomedical Imaging (ISBI 2018)* (IEEE) pp 766–9
- [25] Nguyen T-H and Chung W-Y 2018 A single-channel SSVEP-based BCI speller using deep learning *IEEE Access* **7** 1752–63
- [26] Xing J, Qiu S, Ma X, Wu C, Li J, Wang S and He H 2020 A CNN-based comparing network for the detection of steady-state visual evoked potential responses *Neurocomputing* **403** 452–61
- [27] Ravi A, Beni N H, Manuel J and Jiang N 2020 Comparing user-dependent and user-independent training of CNN for SSVEP BCI *J. Neural Eng.* **17** 026028
- [28] Li Y, Xiang J and Kesavadas T 2020 Convolutional correlation analysis for enhancing the performance of SSVEP-based brain-computer interface *IEEE Trans. Neural Syst. Rehabil. Eng.* **28** 2681–90
- [29] Liao X, Yao D, Wu D and Li C 2007 Combining spatial filters for the classification of single-trial EEG in a finger movement task *IEEE Trans. Biomed. Eng.* **54** 821–31
- [30] Ko W, Jeon E, Jeong S and Suk H-I 2020 Multi-scale neural network for EEG representation learning in BCI (arXiv:2003.02657)
- [31] Dai G, Zhou J, Huang J and Wang N 2020 HS-CNN: a CNN with hybrid convolution scale for EEG motor imagery classification *J. Neural Eng.* **17** 016025
- [32] Nakanishi M, Wang Y, Wang Y-T and Jung T-P 2015 A comparison study of canonical correlation analysis based methods for detecting steady-state visual evoked potentials *PLoS One* **10** e0140703
- [33] Wang Y, Chen X, Gao X and Gao S 2016 A benchmark dataset for SSVEP-based brain-computer interfaces *IEEE Trans. Neural Syst. Rehabil. Eng.* **25** 1746–52
- [34] Liu B, Huang X, Wang Y, Chen X and Gao X 2020 Beta: a large benchmark database toward SSVEP-BCI application *Frontiers neurosci.* **14** 627
- [35] Jayaram V and Barachant A 2018 MOABB: trustworthy algorithm benchmarking for BCIs *J. Neural Eng.* **15** 066011
- [36] Delorme A and Makeig S 2004 EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis *J. Neurosci. Methods* **134** 9–21
- [37] Virtanen P et al 2020 SciPy 1.0: fundamental algorithms for scientific computing in python *Nat. Methods* **17** 261–72