

PAPER • OPEN ACCESS

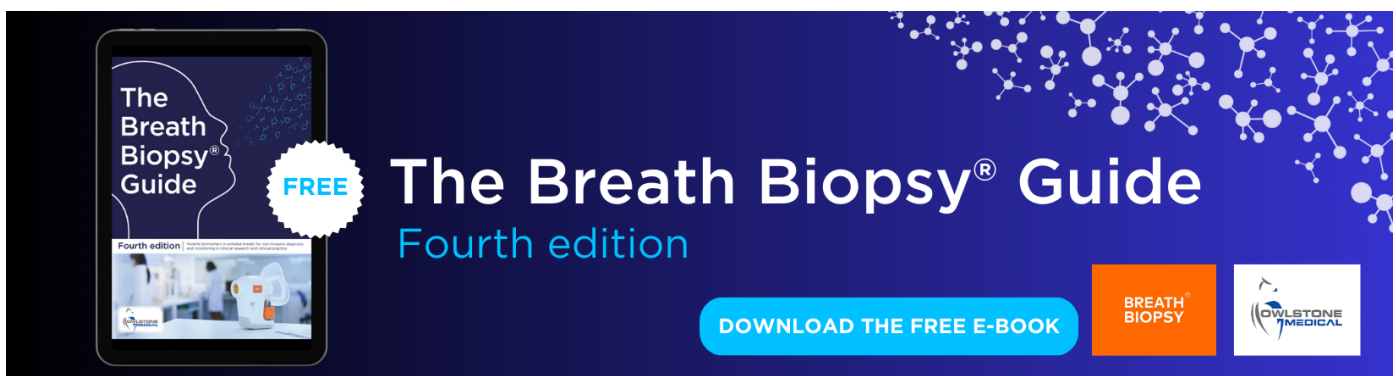
From full calibration to zero training for a code-modulated visual evoked potentials for brain–computer interface

To cite this article: J Thielen *et al* 2021 *J. Neural Eng.* **18** 056007

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

You may also like

- [Nonlinear point-process estimation of neural spiking activity based on variational Bayesian inference](#)
Ping Xiao and Xincheng Liu
- [Brain–computer interfaces based on code-modulated visual evoked potentials \(c-VEP\): a literature review](#)
Víctor Martínez-Cagigal, Jordy Thielen, Eduardo Santamaría-Vázquez et al.
- [Multiscale modeling and decoding algorithms for spike-field activity](#)
Han-Lin Hsieh, Yan T Wong, Bijan Pesaran et al.



The Breath Biopsy® Guide
Fourth edition

FREE

DOWNLOAD THE FREE E-BOOK

BREATH BIOPSY

OWLSSTONE MEDICAL



PAPER

OPEN ACCESS

RECEIVED
20 November 2020

REVISED
17 January 2021

ACCEPTED FOR PUBLICATION
9 March 2021

PUBLISHED
6 April 2021

Original Content from
this work may be used
under the terms of the
[Creative Commons
Attribution 4.0 licence](#).

Any further distribution
of this work must
maintain attribution to
the author(s) and the title
of the work, journal
citation and DOI.



From full calibration to zero training for a code-modulated visual evoked potentials for brain–computer interface

J Thielen^{1,2,*} , P Marsman², J Farquhar¹ and P Desain^{1,2}

¹ MindAffect, Nijmegen, The Netherlands

² Radboud University, Donders Institute for Brain, Cognition and Behaviour, Nijmegen, The Netherlands

* Author to whom any correspondence should be addressed.

E-mail: jordy.thielen@donders.ru.nl

Keywords: brain–computer interface (BCI), electroencephalography (EEG), code-modulated visual evoked potentials (cVEPs), reconvolution, zero training, spread spectrum communication

Abstract

Objective. Typically, a brain–computer interface (BCI) is calibrated using user- and session-specific data because of the individual idiosyncrasies and the non-stationary signal properties of the electroencephalogram (EEG). Therefore, it is normal for BCIs to undergo a time-consuming passive training stage that prevents users from directly operating them. In this study, we systematically reduce the training data set in a stepwise fashion, to ultimately arrive at a calibration-free method for a code-modulated visually evoked potential (cVEP)-based BCI to fully eliminate the tedious training stage. **Approach.** In an extensive offline analysis, we compare our sophisticated encoding model with a traditional event-related potential (ERP) technique. We calibrate the encoding model in a standard way, with data limited to a single class while generalizing to all others and without any data. In addition, we investigate the feasibility of the zero-training cVEP BCI in an online setting. **Main results.** By adopting the encoding model, the training data can be reduced substantially, while maintaining both the classification performance as well as the explained variance of the ERP method. Moreover, with data from only one class or even no data at all, it still shows excellent performance. In addition, the zero-training cVEP BCI achieved high communication rates in an online spelling task, proving its feasibility for practical use. **Significance.** To date, this is the fastest zero-training cVEP BCI in the field, allowing high communication speeds without calibration while using only a few non-invasive water-based EEG electrodes. This allows us to skip the training stage altogether and spend all the valuable time on direct operation. This minimizes the session time and opens up new exciting directions for practical plug-and-play BCI. Fundamentally, these results validate that the adopted neural encoding model compresses data into event responses without the loss of explanatory power compared to using full ERPs as a template.

1. Introduction

A brain–computer interface (BCI) enables the use of a non-muscle channel to communicate with the external world by extracting intentions from measured brain activity and by converting these to a computer output [1]. A BCI involves a user who performs a specific task that evokes a clear pattern of brain activity as recorded by some measuring device. For practical reasons, electroencephalography (EEG) is commonly used for BCI. The recorded brain activity goes through several data processing steps in order to decode the user's intended action, which is used

to issue a command to a device, which often can be observed by the user as feedback [2]. Before any output can be generated, the machine learning method needs calibration data that are collected in a supervised way (i.e. the user is instructed so that the data are labeled). This training phase is essential for the classifier to capture the user-specific response, but is often ignored in system performance evaluation.

1.1. Visual evoked potentials for speller BCIs

A commonly used brain response for BCI is the exogenous visual evoked potential (VEP) [3], which is triggered by and time-locked to an external visual

stimulus. Using the VEP for communication was already proposed in 1984 [4]. A common categorization of the VEP defines three kinds: time modulated (tVEP), frequency modulated (fVEP) and code modulated (cVEP). These concepts can be related to multiple access technology in telecommunication realized by coding in these three domains. The most well-known is code-division multiple access. While this approach formally casts BCI in terms of information theory as communication over a noisy channel (i.e. the brain) [3], the concepts directly relate to cognitive neuroscience: tVEPs are transient responses to isolated individual events such as event-related potentials (ERPs) such as the P300; fVEPs are steady-state responses (oscillations) to periodic sequences of events such as the steady-state visual evoked potential (SSVEP); and cVEPs are broad-band responses to fast (i.e. overlapping) stochastic sequences of events such as the broadband visual evoked potential.

In this work, we focus on cVEP BCI, which uses optimized pseudo-random sequences to encode stimuli. A cVEP BCI can be robust to narrow-band interference due to its spread-spectrum nature, just as conversely an fVEP can be more robust to broadband noise. In addition, the demodulation of cVEP signals as a sequence of a few event types (as we do in our work using an encoding model) can cause a processing gain, i.e. a much-needed increase in the signal-to-noise ratio (SNR). The first realization of a cVEP-based BCI was proved to be reliable for an amyotrophic lateral sclerosis (ALS) patient with invasive recordings [4, 5]. Only later were reliable performances shown with non-invasive recordings [6, 7]. An important improvement was the use of canonical correlation analysis (CCA) for spatial filtering [8] and a combined spatio-temporal characterization [9, 10]. Furthermore, several techniques have shown high performance such as with the use of a support vector machine [11, 12] and convolutional neural network [13, 14]. Finally, this non-invasive cVEP method has also recently proved to be feasible for patient applications [15].

A common application of BCI is communication. This was pioneered by the seminal work of Farwell and Donchin, who proposed the visual matrix speller [16]. This classic tVEP-based BCI (using the P300 ERP) is reliable and is one of the few BCIs to be adopted in practice. Higher communication speeds have been reported by using fVEP-based spellers [17, 18] or cVEP-based spellers [12, 14]. For an excellent review of speller BCIs, see [19].

1.2. Improving BCI at test time

A common measure used in the BCI field to estimate the performance of a (speller) BCI is the information transfer rate (ITR) [20]. This measure takes into account the classification accuracy, the time it takes to make a classification (including trial and inter-trial time) and the total number of selections possible.

The ITR makes it possible to compare the performance of very different BCIs and captures the trade-off between the three components that interact with each other: size of the transmitted alphabet (i.e. the number of classes), accuracy (i.e. the proportion of correct transmissions) and time (i.e. the duration of a single trial, the number of samples needed for detection).

In the BCI field, significant attention goes into optimizing the classification accuracy or classification speed that together improve the ITR [20]. This is important to reduce the testing time as much as possible. Here, with testing time, we refer to the average time it takes to make a selection, which can be optimized by reducing the number of errors or by reducing the time for a single selection. To reduce the selection time, several static as well as dynamic stopping methods exist. In static stopping procedures, the overall optimal stopping time is estimated from the set of training trials and all testing trials thus take the same amount of data. Instead, a dynamic stopping rule exploits between-trial differences and, after picking a criterion (such as the probability of error to be below a certain level) and characterizing the parameters for it on the training data, allows the trial cutoff (i.e. the time point where the system decides and emits the output) to be chosen dynamically according to a run-time criterion. In our previous work, we used a dynamic stopping rule based on the margin between the correlation of the trial with the best and second best template [9]. The downside of this method is that it indeed needs to be calibrated with training data.

1.3. Improving BCI at training time

The ITR concerns only the test performance and thus ignores the time it took to calibrate the BCI. Clearly, a BCI that is calibrated within a few seconds is preferred over one that requires minutes or even hours, all other things being equal. We would like to stress here that the BCI field should incorporate a session performance that takes into account training alongside testing performance, in addition to the ITR.

Specifically, instead of focusing on testing time, the main aim of this study is to reduce the training time of a cVEP BCI, ultimately to none at all. As noted before, (speller) BCIs require a calibration stage to learn the user-specific data distribution that ultimately allows the BCI to decode the user's intention from brain activity. It is well known throughout the BCI literature that calibration is necessary for every new user and often even for every new session of the same user, because of variations between users as well as within users over time (non-stationarity). The downside of such a calibration period is that it takes away valuable time for the online testing phase. This issue becomes even more prevalent for patient applications, where the patients might have a limited attention span or become fatigued more easily due to their condition, more so as it has been shown

that workload and fatigue can affect ERP responses in general [21].

To overcome the above-mentioned issue, the calibration time should be reduced as much as possible. A popular method to speed up or even eliminate calibration time is to apply transfer learning. The goal of transfer learning is to find a session-independent classifier using cross-participant data (for an excellent overview, see [22]). Despite theoretical advances, practical applications of transfer learning remain scarce, especially for evoked-response BCIs. For the P300 speller, several attempts have been made using unsupervised learning and online adaptation [23, 24] and learning from label proportions [25]. An explicit attempt to reduce the training time for cVEP BCIs was made using an automatic repeat request, which stops acquiring more training trials when a certain reliability measurement reaches a threshold during calibration [26]. So far, only one study suggested a zero-training approach for a cVEP BCI, making use of a language model of likely output symbols that post hoc corrects previously made classifications [27]. In addition, one study explored transfer learning for a cVEP BCI [28].

When available, an encoding model that models and predicts the EEG signal (the input to the classifier) based on the (attended) stimulus sequence, has even more potential in reducing calibration time. In a previous work, we introduced reconvolution as a novel (patented [29]) framework to model EEG responses to sequences of events for BCI [9, 10]. Reconvolution is an encoding model that models responses to sequences of events simply as the linear sum of the responses to the individual events. It is similar to, but not the same as the true deconvolution approach, for example, taken by Lalor and colleagues [30]. This is because the input is not treated as a sequence of Dirac pulses that in the limit make up a continuous function, but as an asynchronous sequence of discrete events, each with its response. A similar decoding model was proposed by [13] and the recently proposed EEG2Code model [14, 31], both of which use a similar decomposition to learn responses to events instead of full sequences. In this way, the number of parameters reduces from the number of samples in the response to full sequences of events, to the number of samples in the responses to a few individual events. In addition, there are many more events in a sequence than there are sequences themselves. Together, this allows for a processing gain that can lead to faster recognition in the testing phase, but may in principle also reduce the training time.

1.4. Aims of the current study

In this study, we use a large offline data set to compare four training regimes that stepwise reduce the richness of the training data: e-train, n-train, 1-train and 0-train. First, as a baseline, we investigate ERPs to full sequences to be used as templates for

a template-matching classifier (e-train). This standard approach (e.g. see [6, 7]) requires many repetitions for each possible sequence to compute robust averages and therefore requires a large training data set. Second, we apply an encoding model (reconvolution) trained with data from all n classes available as the target (n-train), to investigate how much such a modeling approach can reduce the training data compared to e-train. Third, we apply the encoding model to data from only one class (1-train) while generating the templates for all n sequences, to investigate whether the encoding model generalizes well. Fourth, we use the encoding model for a semi-supervised zero-training approach (0-train), which requires no calibration data at all. Note that we call it semi-supervised because, while it does not require the trial to be labeled with the actual stimulus class during training, it does require the codebook of all allowed stimulus sequences to be known. Such a zero-training approach becomes feasible when the response components are known beforehand, as in fVEP where the SSVEP response is known to be an oscillation with the stimulation frequency as its fundamental frequency plus harmonics. Zero training can then be implemented as the best model fit (e.g. see [32]).

In addition to the important pragmatic benefit of a faster (i.e. zero) calibration, the comparison between the methods also reveals to what extent the underlying neuroscientific model, a decomposition into independent responses to a few event types, holds. Specifically, an encoding model (n-train) that is as successful as the ERP model (e-train) demonstrates that higher-order patterns in the group of stimuli, such as the ones allowing for entrainment, do not evoke an important component in the EEG. Thus, the variance can be explained well by the much more parsimonious event responses in the encoding model. This assumption is put to the test even further when the encoding model is required to generalize to stimulation patterns it has not seen before (1-train) and when the model is used to fit any response and perform classification according to the model fit (0-train).

Finally, we complement the offline analyses with an evaluation of the zero-training approach in a full online system using only a few water-based electrodes to prove the feasibility and performance during real use of a practical zero-training cVEP BCI. Together, these methods allow for a plug-and-play calibration-free BCI making full use of the reconvolution encoding model to allow for high performance over a fast full train-test session.

2. Methods

2.1. Data availability

All raw data and analysis scripts will be made available at the Donders Data Repository

<https://data.donders.ru.nl/> at <https://doi.org/10.34973/9txv-z787>.

2.2. Participants

Thirty participants (aged 19–62 years, average 25 years; 17 females) participated in the offline experiment. Eleven participants (aged 20–63, average 28 years; 3 females) participated in the online experiment. Two of the participants in the online study were excluded from the analysis due to poor signal quality throughout the experiment. Four of these participants took part in both experiments. Exclusion criteria were any history of epilepsy or claustrophobia. All participants had normal or corrected to normal vision and reported no central nervous system abnormalities. All participants gave written informed consent prior to the experiment and received payment or course credit after the experiment. The experimental procedure and methods were approved by and performed in accordance with the guidelines of the local ethical committee of the Faculty of Social Sciences of Radboud University.

2.3. Materials

The EEG data of the offline experiment were recorded at 512 Hz with eight sintered Ag/AgCl active electrodes placed according to the 10–20 system (Fz, T7, T8, POz, O1, Oz, O2, Iz) and amplified by a Biosemi ActiveTwo amplifier. The EEG data were offline preprocessed using FieldTrip [33], following a high-pass filter at 2 Hz using a second-order Butterworth filter, low-pass filter at 30 Hz using a sixth-order Butterworth filter and downsampling to 120 Hz.

The EEG data of the online experiment were recorded at 512 Hz with eight water-based electrodes placed according to the 10–20 system (Fz, T7, T8, POz, O1, Oz, O2, Iz) and amplified by a TMSi Porti amplifier. The EEG data were in real-time high-pass filtered at 2 Hz using a second-order Butterworth filter, low-pass filtered at 50 Hz using a fourth-order Chebyshev type II filter with 50 dB stop-band attenuation and downsampled to 180 Hz.

In both experiments, we specifically chose those eight electrodes to optimize signal acquisition around the occipital cortex, while capturing some irrelevant signals from other sources for noise cancellation. For a larger analysis of electrode positioning for cVEP BCIs, see the work by Ahmadi and colleagues [34].

In both offline and online experiments, stimuli were presented on a 12.9 inch iPad Pro (Apple Inc.) with a 60 Hz refresh rate and 1920 × 1080 pixel resolution. The tablet was placed in a tablet holder in front of the participant at a viewing distance of 60 cm. Both experiments were conducted in a room with constant ambient light, mimicking everyday conditions. In the offline experiment, the stimuli were arranged in a 4 × 5 ($n = 20$, figure 1(a)) calculator grid to limit the number of symbols, while in the online experiment a keyboard layout ($n = 29$, figure 1(b)) was presented to

allow participants to spell sentences. The stimuli were presented on a mean luminance gray background. Individual cells were 3.1 × 2.8 cm (i.e. 2.96 × 2.67 visual degrees) with 0.4 cm (i.e. 0.38 visual degrees) space in between both horizontally as well as vertically. At the right top of the screen, a black box was placed that turned white at the start of stimulation, triggering a synchronization signal via an optosensor that marks the onset of stimulation in the EEG data. The processing of the continuous data stream was handled by the BrainStream software package (<https://tsgdoc.socsci.ru.nl/index.php?title=BrainStream>).

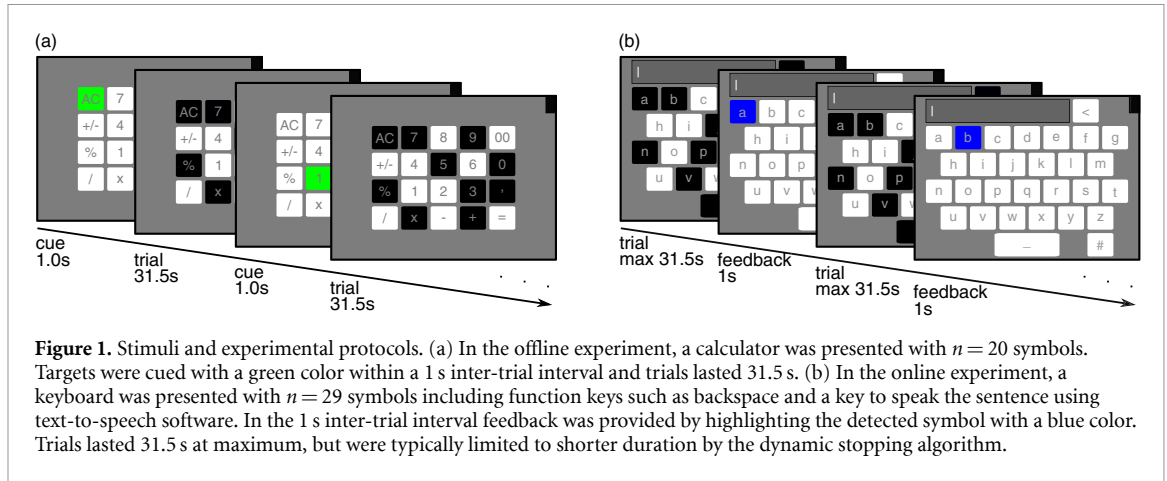
Each cell was luminance modulated with a unique pseudo-random bit-sequence, where ones represent a white color and zeros a black color. We generated a preferred pair of 63-bit m-sequences [35] with feedback tap positions at [6, 5, 2, 1] and [6, 1], and combined these using the XOR operator at any time delay to generate a set of $2^6 - 1 = 63$ Gold codes [36]. Gold codes are a set of binary sequences that exhibit minimal auto- and cross-correlation between them and are therefore widely used in telecommunication and navigation systems. It is common to add the original two m-sequences to this set to generate a set of $2^6 + 1 = 65$ Gold codes [37].

We further modulated these codes by multiplying (XOR-ing) with a double frequency bit-clock. The resulting transition in the middle of each bit causes the modulated bit-sequences (seen with respect to the doubled clock) to contain only short events (i.e. one one, followed by one or two zeros (10, 100), with a 16.67 ms flash) and long events (two ones, followed by one or two zeros (110, 1100), with a 33.34 ms flash) as in our previous work [9]. These modulated Gold codes had a length of $2 \times (2^6 - 1) = 126$ bits for a duration of 2.1 s. They exhibit good auto- and cross-correlation properties together with limited low-frequency content (i.e. they contain no long runs of ones or zeros).

From this set of modulated Gold codes we selected the best subset of 20 sequences for the offline experiment and 29 for the online experiment. We did so using the subset optimization procedure as presented in our previous work [9]. In short, we first computed grand averaged modeled responses for all modulated Gold codes given the data from [9]. Subsequently, we selected the subset of 20 or 29 sequences that minimized the maximum pairwise within subset cross-correlation in their grand averaged modeled responses.

2.4. Offline experiment

The offline experiment started with a practice block in which participants used the zero-training approach (0-train) to select symbols of their choice and could familiarize themselves with the system. Data recorded during this practice block were not used for offline analysis. Afterwards, participants completed five identical blocks. Each block consisted of 20 trials, one



for each of the 20 cells presented in random order. At the start of the trial, the target cell was highlighted in green for 1 s. After this cue, all cells started flashing for 31.5 s, while the participant maintained fixation at the target cell (see figure 1(a)). After the trial has ended, the next trial was initiated directly, with no feedback being given. We presented trials for this long duration of 31.5 s to allow many code repetitions needed for e-train as well as having longer trials that may be required for the first few trials for 0-train (i.e. a warm-up period).

In summary, a total of 100 single trials of EEG data were collected for offline analysis. Within these 100 trials, there were five trials for each of the 20 bit-sequences. Each trial was 31.5 s long and had an inter-trial interval of 1 s. Within this 31.5 s trial, a bit-sequence of 2.1 s was repeated 15 times.

2.5. Online experiment

Participants completed three short blocks in the online experiment. Each block consisted of a maximum of 60 trials with a maximum trial duration of 31.5 s and an inter-trial interval of 1 s for feedback (see figure 1(b)). Within a trial, all cells flashed concurrently with their respective bit-sequences, while the participant kept fixation at the targeted cell. After the trial had ended, feedback on the selection was given by coloring the selected symbol blue and adding the selected symbol to the sentence at the top of the keyboard.

During each of the three online blocks, the BCI operated in the zero-training mode (0-train). The classifier was always reinitialized prior to every block, so there was no data transfer between blocks. In addition, the classifier was running the dynamic stopping algorithm (see section 2.7), which means that trials were classified as quickly as possible, with a maximum duration of 31.5 s, upon which classification was enforced.

In the first block, participants were asked to spell all characters one by one in alphabetical order (26 symbols plus stop button). In the second block, participants were asked to spell the sentence, ‘plug and play brain

computer interface’ (38 symbols plus stop button). In the third block, participants spelled a sentence of their choice (on average 38 symbols plus a stop button). In all three blocks, participants were asked to correct mistakes by selecting the backspace and to stop the application by selecting the speak button.

2.6. Classification

We use a template-matching classifier to predict the attended cell given the recorded brain activity. Let $\mathbf{X} \in \mathbb{R}^{c,m}$ be a single trial of c channels and m samples and $\mathbf{w} \in \mathbb{R}^c$ a spatial filter with a weight for each of the c channels. The first step in the classification is to spatially filter the single trial:

$$\mathbf{x} = \mathbf{w}^\top \mathbf{X}, \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^m$ is the weighted sum of the channels. Subsequently, we evaluate the similarity of the single trial with each of the n template responses $\mathbf{t}_i \in \mathbb{R}^m$, i.e. one for each of n possible output classes. We compute the similarity as the Pearson’s correlation and maximize it to obtain a classification:

$$\hat{y} = \arg \max_i \frac{\mathbf{x}^\top \mathbf{t}_i}{\sqrt{\mathbf{x}^\top \mathbf{x} \cdot \mathbf{t}_i^\top \mathbf{t}_i}}. \quad (2)$$

2.7. Stopping rule

To emit the decision as soon as possible, we implemented a novel dynamic stopping procedure. The stopping procedure, as described in our earlier work, requires the learning of margins using training data [9]. Here, we present a stopping rule that does not need to be calibrated and can therefore also be used for a zero-training approach.

Before emitting \hat{y} , a confidence level is estimated based on the correlation of the single trial with the best fitting template and the (Beta) distribution of the correlations with the other templates. Since the Beta distribution ranges from 0 to 1 while correlations range from -1 to 1 , we first transform the correlations according to $z_i = (\rho_i + 1)/2$. Subsequently, we fit the Beta distribution over all but the maximum correlation yielding the parameters α and β of the

Beta distribution $f(z_{i\neq\hat{y}}; \alpha, \beta)$. Next, we compute the probability that the maximum correlation is higher than a random maximum from the Beta distribution $p = f(z_{\hat{y}}; \alpha, \beta)$. The classification \hat{y} is emitted when p exceeds a targeted probability threshold of 0.95.

In principle, the stopping rule could be checked on every EEG sample added to the single trial. However, this was computationally too expensive, so instead a lower time resolution was chosen pragmatically. Data segments were 500 ms long in the online experiment, while they were 100 ms in the offline experiment to allow for smoother learning and decoding curves. Thus, the dynamic stopping procedure is applied to an incrementally growing size of the single trial by adding these data segments and is corrected for multiple comparisons using a Bonferroni correction.

2.8. Calibration

The objective is to build a template-matching classifier, as defined above, to optimally decode the attended class using a spatial filter \mathbf{w} and template responses \mathbf{t}_i for all $i = 1, \dots, n$ classes. To do so, we present four methods for calibrating such a template-matching classifier: e-train, n-train, 1-train and 0-train. These four methods step-wise decrease the richness of the training data. Specifically, the e-train method uses the conventional averaging approach to compute ERPs, which requires many repetitions of single trials for each of the n classes and therefore requires a large training data set. Instead, the n-train, 1-train and 0-train approaches use an encoding model (reconvolution) from our earlier work [9, 10, 29]. This model reduces the amount of training data substantially as it is based on responses to a handful of event types instead of responses to full sequences of events. Specifically, the model can be used to estimate templates given limited data of all classes (n-train), data of only one class (1-train) or no training data at all (0-train). We provide details of all four methods below.

2.8.1. e-train

The e-train method builds the templates by computing ERPs. Typically, these are measured as the averaged response to repeated stimulation with the same stimulus sequence:

$$\mathbf{T}_i = \frac{1}{J} \sum_j \mathbf{X}_j, \quad (3)$$

where $\mathbf{T}_i \in \mathbb{R}^{c,m}$ and $\mathbf{X}_j \in \mathbb{R}^{c,m}$ with $j = 1, \dots, J$ are the single-trials of the same i th class. We then require spatial filters for the raw data in \mathbf{X} as well as templates in \mathbf{T} . For this, we employ CCA. We stack and repeat the ERPs \mathbf{T}_i following the order in \mathbf{X} for all k training trials:

$$\mathbf{T} = [\mathbf{T}_{y_1}, \mathbf{T}_{y_2}, \dots, \mathbf{T}_{y_k}], \quad (4)$$

where \mathbf{T}_{y_i} is the template belonging to the class as specified by the label y_i of the i th trial and so that $\mathbf{T} \in \mathbb{R}^{c,m \cdot k}$. We concatenate all trials in \mathbf{X} to $\mathbf{S} \in \mathbb{R}^{c,m \cdot k}$ so that we can find optimized spatial filters using CCA:

$$\max_{\mathbf{w}, \mathbf{v}} \rho(\mathbf{w}^\top \mathbf{S}, \mathbf{v}^\top \mathbf{T}) = \frac{\mathbf{w}^\top \mathbf{S} \mathbf{T}^\top \mathbf{v}}{\mathbf{w}^\top \mathbf{S} \mathbf{S}^\top \mathbf{w} \cdot \mathbf{v}^\top \mathbf{T} \mathbf{T}^\top \mathbf{v}}, \quad (5)$$

where $\mathbf{w} \in \mathbb{R}^c$ can be used to spatially filter single trials and $\mathbf{v} \in \mathbb{R}^c$ can be used to spatially filter the multi-channel templates to obtain the \mathbf{t}_i required for the template-matching classifier:

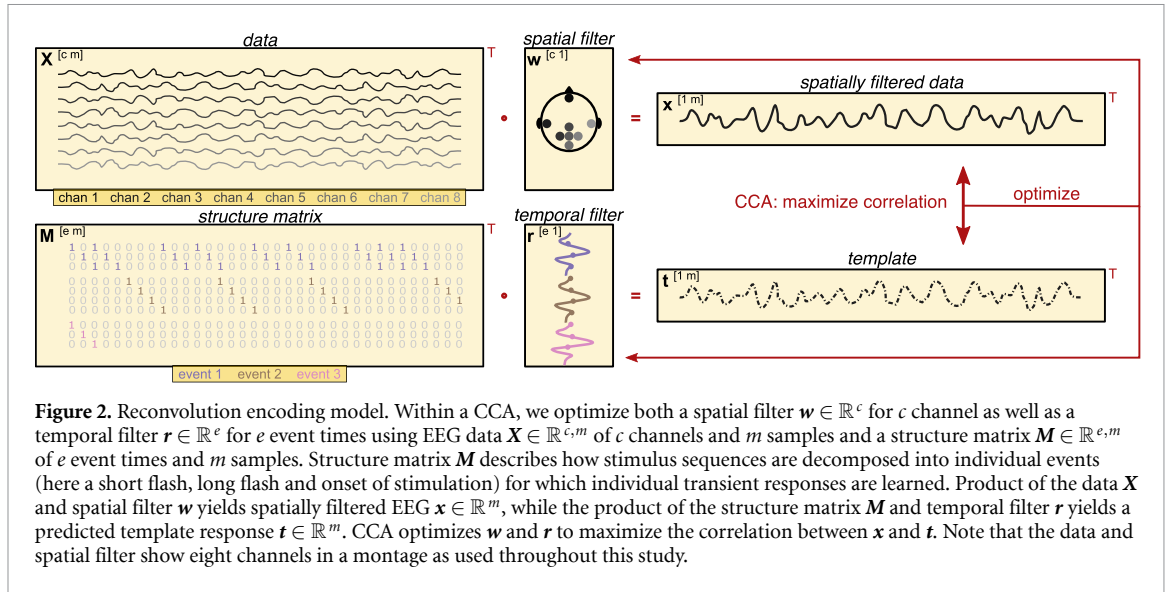
$$\mathbf{t}_i = \mathbf{v}^\top \mathbf{T}_i. \quad (6)$$

2.8.2. n-train

The n-train method builds templates by using the reconvolution encoding model proposed in our previous work [9, 10] (also patented [29]). Instead of learning responses to full stimulus sequences (e.g. e-train), this encoding model exploits the fact that all bit-sequences are sequences of events. Specifically, following the linear superposition hypothesis, the response to a sequence of events is the linear summation of the responses to the individual events (see e.g. [38], but also see [39]). Hence, we can apply the averaging at the level of events, instead of at the level of sequences. Since there are many events within sequences, exploiting the repetition at the level of events can reduce the training data substantially. A graphical representation of reconvolution is depicted in figure 2.

Reconvolution requires a mapping of a bit-sequence $\mathbf{V}_i \in \mathbb{R}^m$ to events $\mathbf{E}_i \in \mathbb{R}^{e,m}$, where m is the number of samples and e the number of events. The event matrix \mathbf{E} denotes the i th code at which sample a certain event happens with a one, and is zero elsewhere. A straightforward mapping could be used to define a single flash (i.e. one 'on' bit) as an event (e.g. [14]). Another mapping could differentiate between short (i.e. one 'on' bit) and long (i.e. two 'on' bits) flashes to allow for some form of non-linearity (e.g. [9]). Here, we use a mapping to short and long flashes and also include an event that marks the onset of stimulation only to capture the large transient response to the start of stimulation at the beginning of a single trial.

Subsequently, we use a so-called structure matrix $\mathbf{M}_i \in \mathbb{R}^{l,m}$ that maps an event to an impulse response function (i.e. a matrix representation of convolution). This matrix is of a Toeplitz-like structure and denotes when certain events happen and how they overlap. The first row of \mathbf{M}_i is an exact copy of \mathbf{E}_i for a particular event. The next row lists a one one sample later and so ones go down the rows of l columns. Naturally, l denotes the length of the impulse response function that is estimated for this event. In the case of multiple events ($e > 1$), the structure matrix is constructed for all individual events (possibly with varying l)



and then concatenated to form a final structure matrix $\mathbf{M}_i \in \mathbb{R}^{e,l,m}$.

The estimation of both the temporal filter as well as the spatial filter can then be performed using a single CCA decomposition. For this, we stack the structure matrices following the order in \mathbf{X} for all k training trials:

$$\mathbf{M} = [\mathbf{M}_{y_1}, \mathbf{M}_{y_2}, \dots, \mathbf{M}_{y_k}], \quad (7)$$

where \mathbf{M}_{y_i} is the structure matrix belonging to the class as specified by the label y_i of the i th trial, so that $\mathbf{M} \in \mathbb{R}^{l,m,k}$ and we concatenate all trials in \mathbf{X} to $\mathbf{S} \in \mathbb{R}^{c,m,k}$. Now, we find an optimized spatial filter \mathbf{w} and the temporal response vector \mathbf{r} using CCA:

$$\max_{\mathbf{w}, \mathbf{r}} \rho(\mathbf{w}^\top \mathbf{S}, \mathbf{r}^\top \mathbf{M}) = \frac{\mathbf{w}^\top \mathbf{S} \mathbf{M}^\top \mathbf{r}}{\sqrt{\mathbf{w}^\top \mathbf{S} \mathbf{S}^\top \mathbf{w} \cdot \mathbf{r}^\top \mathbf{M} \mathbf{M}^\top \mathbf{r}}}, \quad (8)$$

where $\mathbf{w} \in \mathbb{R}^c$ and $\mathbf{r} \in \mathbb{R}^{e,l}$. The spatial filter \mathbf{w} can be used to spatially filter single trials, while the response vector \mathbf{r} can be used to predict template responses to any stimulus for which we have a structure matrix:

$$\mathbf{t}_i = \mathbf{r}^\top \mathbf{M}_i. \quad (9)$$

2.8.3. 1-train

The 1-train method is similar to the n-train method, but is trained on data from one class only. Reconvolution is able to predict responses to stimuli it has not seen before, as long as these unseen stimuli are built up from the same events [9]. This may even further reduce the richness of the training data.

2.8.4. 0-train

Here, we present a novel zero-training method that reduces the training data to none at all. For this, we again use reconvolution and prior knowledge that only n stimuli are possible by design. To classify an unknown single trial, an encoding model is fit for

each possible class [10]. Specifically, given a single-trial $\mathbf{X} \in \mathbb{R}^{c,m}$ of m samples and c channels and given all n possible structure matrices $\mathbf{M}_i \in \mathbb{R}^{e,l,m}$ we first optimize class-specific CCAs:

$$\max_{\mathbf{w}_i, \mathbf{r}_i} \rho(\mathbf{w}_i^\top \mathbf{X}, \mathbf{r}_i^\top \mathbf{M}_i) = \frac{\mathbf{w}_i^\top \mathbf{X} \mathbf{M}_i^\top \mathbf{r}_i}{\sqrt{\mathbf{w}_i^\top \mathbf{X} \mathbf{X}^\top \mathbf{w}_i \cdot \mathbf{r}_i^\top \mathbf{M}_i \mathbf{M}_i^\top \mathbf{r}_i}}. \quad (10)$$

Subsequently, we can spatially filter the single trial with the class-specific spatial filter and predict the template response using the class-specific structure matrix and response vector:

$$\mathbf{x}_i = \mathbf{w}_i^\top \mathbf{X}, \quad (11)$$

$$\mathbf{t}_i = \mathbf{r}_i^\top \mathbf{M}_i. \quad (12)$$

Finally, we can perform a similar template matching as defined before:

$$\hat{y} = \arg \max_i \frac{\mathbf{x}_i^\top \mathbf{t}_i}{\sqrt{\mathbf{x}_i^\top \mathbf{x}_i \cdot \mathbf{t}_i^\top \mathbf{t}_i}}. \quad (13)$$

Intuitively, this classification can be interpreted as maximizing the square root of the explained variance of class-specific models. Hence, the best explaining stimulus sequence is regarded as the attended class.

By definition, the first single trial is classified as semi-supervised without any prior knowledge. We say semi-supervised, because the codebook (i.e. the description of when certain events, e.g. flashes, occur for each class or stimulus) is known *a priori*. However, subsequent single trials use previously classified trials as a kind of training data to facilitate subsequent classification. Here, we assume that previously classified single trials are correctly classified. Then, subsequent classifications are performed on both previous trials and current trial $\mathbf{X} = [\mathbf{X}_{1:t-1}, \mathbf{X}_t]$ and the corresponding structure matrices $\mathbf{M} = [\mathbf{M}_{1:t-1}, \mathbf{M}_t]$.

Instead of computing the CCA for all (stacked) trials over and over, we employ the fact that CCA can be written using covariance matrices:

$$\max_{\mathbf{w}_i, \mathbf{r}_i} \rho(\mathbf{w}_i^\top \mathbf{X}, \mathbf{r}_i^\top \mathbf{M}) = \frac{\mathbf{w}_i^\top \Sigma_{XM} \mathbf{r}_i}{\sqrt{\mathbf{w}_i^\top \Sigma_X \mathbf{w}_i \cdot \mathbf{r}_i^\top \Sigma_M \mathbf{r}_i}}, \quad (14)$$

where Σ_X is the covariance matrix of \mathbf{X} , Σ_M is the covariance matrix of \mathbf{M} and Σ_{XM} is the cross-covariance matrix of \mathbf{X} and \mathbf{M} . We implemented an incremental covariance update to keep track of descriptive statistics rather than full data matrices. Thus, after collection of a new data segment within a single trial, Σ_X , Σ_M and Σ_{XM} are updated accordingly. Each update adds knowledge to the classifier, which enables a better and faster model fit for subsequent trials. At the first data segment, standard averages and covariances are estimated. For subsequent data segments, the update of a covariance matrix was done as follows:

$$m_{\text{new}} = m_{\text{old}} + m_{\text{obs}}, \quad (15)$$

$$\mu_{\text{new}} = \mu_{\text{old}} + \frac{m_{\text{obs}}}{m_{\text{new}}} \sum (X - \mu_{\text{old}}), \quad (16)$$

$$\Sigma_{\text{obs}} = (X - \mu_{\text{old}})^\top (X - \mu_{\text{new}}), \quad (17)$$

$$\Sigma_{\text{new}} = \frac{1}{m_{\text{new}} - 1} \Sigma_{\text{obs}} + \frac{m_{\text{old}} - 1}{m_{\text{new}} - 1} \Sigma_{\text{old}}, \quad (18)$$

where X is the newly observed data segment with which to update the covariance estimate and m_{obs} the number of samples in this data segment. μ_{new} and μ_{old} denote running averages and Σ_{new} and Σ_{old} the running covariances.

This is a critical first single-trial zero-training approach, since it will have a significant influence on the forthcoming single trials. Therefore, to make the approach more robust, the confidence threshold for dynamic stopping was set to 0.99 for the first single trial. In addition, the first single trial was forced to be at least 12 s long in the online experiment and only 2 s in the offline experiment.

2.9. Evaluation offline experiment

The offline data were analyzed using a leave-one-block-out cross-validation. For each fold, a learn-decode landscape (train-time \times test-time \rightarrow accuracy) was estimated without the stopping rule (i.e. static stopping) and learning curves (train-time \rightarrow accuracy) were estimated with the stopping rule (i.e. dynamic stopping). Since the four different training methods have different requirements on the training data, the step size of data increments for the learning curves differs between methods: e-train requires data of full 2.1 s code repetitions for every class, n-train uses equal data from all classes, 1-train is limited to

the data of only a single class, while 0-train has no restriction.

The offline learning curves (train-time \rightarrow accuracy) were estimated using the training data, which contained four single trials of 31.5 s for all n classes. For e-train, the learning steps ranged from $n \times 2.1$ (at least 2.1 per class) to $4 \times n \times 31.5$ in steps of $n \times 2.1$. In n-train, the learning steps ranged from 3×2.1 (at least $3 \times 2.1/n$ per class) to 4×31.5 in steps of 2.1 and also from $n \times 2.1$ to $4 \times n \times 31.5$ in steps of $n \times 2.1$ as in e-train. In 1-train, the learning steps ranged from 2.1 to 4×31.5 in steps of 2.1, because data from only one class are used. This also implies that 1-train learning and decoding curves are estimated for each of the n classes. In 0-train, the learning steps ranged from 2.1 to 4×31.5 in steps of 2.1 and also from $n \times 2.1$ to $4 \times n \times 31.5$ in steps of $n \times 2.1$ as in e-train, which is quite similar to n-train except for two additional initial steps.

The offline decoding curves (test-time \rightarrow accuracy) were estimated on the hold-out validation data, which contained one single trial of 31.5 s for each of the n classes. The decoding curves were estimated for every step in the learning curves and always ranged from 100 ms–31.5 s in steps of 100 ms. Note that in order to measure decoding curves, the classifier was kept static, which means that the adaptivity of 0-train was turned off during validation. This makes the decoding comparable over all four methods at a particular step in the learning curve.

Apart from measuring the decoding curves, we estimated the performance of the dynamic-stopping rule. For this, the classifier was applied to each trial in the validation data until the stopping criterion was met. In this way, classification as well as stopping time were measured.

3. Results

3.1. Offline

We collected a large data set of eight high-quality gel-based EEG electrodes, while participants passively operated a 4×5 calculator grid. We used this data set to offline validate and compare four different training regimes: e-train, n-train, 1-train, 0-train. The main aim of this offline analysis was to show the benefit of an encoding model to significantly reduce the amount of data required without performance loss, ultimately to the point at which no learning data is needed at all.

3.1.1. Inclusion of a faulty code-sequence decreases 1-train performance

We observed that one of the bit-sequences was consistently performing worse than all others in the 1-train procedure. This bit-sequence was one of the two original m-sequences used to generate the set of Gold codes that we added to the set following the literature [37]. However, it turned out that it does

not have similar digital correlation properties that the true members of the code family have.

The 1-train classification accuracy (at 2.1 min learning, 2.1 s static-stopping decoding) of the response to this specific m-sequence (71.6%) was significantly different ($p < 0.05$, Bonferonni corrected, Wilcoxon signed-rank test) than the accuracy of any of the other 19 Gold codes (on average 84.6%). We removed this class from the analysis to prevent any effect on 1-train or 0-train (which starts as a 1-train model). From this point forward, all reported offline analyses deal with an $n = 19$ class problem. Removal of this code did not significantly affect the n-class procedure (85.8% 20-class, 85.7% 19-class, $p = 0.367$, Wilcoxon signed-rank test).

3.1.2. Correction for raster latency improves performance

We observed a consistent decrease in classification accuracy over classes presented in a different row in the calculator grid. This turned out to be caused by the raster latency of the display in use. Moreover, the presentation screen is not updated instantaneously, but written line by line; hence the timing information of the stimuli was systematically biased (i.e. it was synchronized to the right top corner). This effect was also visible as a latency shift in the transient responses estimated with 1-train models and might therefore harm the generalizing capacity of 1-train and 0-train. A model such as e-train is insensitive to this issue because it does not share information between classes, while a method such as n-train can generalize over the shifted responses.

We corrected for the latency correction by time-shifting the stimulus onset times in the n-train, 1-train and 0-train methods with latencies as measured with an opto-sensor from the presentation screen. This approach is similar to the one reported by Nagel and colleagues [40]. The n-train classification accuracy (at 2.1 min learning, 2.1 s static-stopping decoding) significantly improved after correcting for the raster latency (85.0% uncorrected, 85.7% corrected, $p = 0.004$, Wilcoxon signed-rank test). Also, the 1-train classification accuracy (at 2.1 min learning, 2.1 s static-stopping decoding) significantly improved after correction (82.1% uncorrected, 83.4% corrected, $p < 0.001$).

3.1.3. Event definition can be freely chosen

It is important to highlight the flexibility of our encoding model; event types that make up the stimulus can be freely defined. From a mathematical perspective, the choice of event definition used is free and any complete coding can be tested for its ability to model the (non)linear response. From a modeling and neuroscientific perspective, the event coding which maps better onto the actual brain response gives both a model that is easier to interpret (as it

models the neuro-computational mechanisms) and a more learning-efficient model.

As we used double bit-clock modulated (i.e. phase shift keyed) Gold codes, the resulting run lengths for both ones and zeros are restricted to 1 or 2. This bit-stream can be considered as a random concatenation of short (1) and long (11) pulse events, if the short (0) and long (00) inter-stimulus intervals are ignored. This event-type representation (called 'duration' from here on) was used in our previous work [9]. In contrast to the claim of [14], our encoding model is generic, so that any definition of event types can be plugged in. Similar to their work, another more basic event-type representation would model any on bit (1) as an event (called 'on'). Yet another definition (we call 'components') can take for instance, the four patterns that constitute the full sequence (10, 100, 110 and 1100). Alternatively, as the neural pathways are often responding to changes much more than to steady states, a natural alternative event-type definition (we call 'contrast') has two events (01 and 10). The last event-type models responses to any transition (01 or 10) with the same single response (we call 'change'), which might be a better fit with, for instance, color sequences.

Note that not all the event-type representations are complete (e.g. some ignore off periods 0 and 00). This means that these models each perform a different kind of lossy compression, which separates brain activity into relevant and irrelevant aspects, and which, when performing well, can point to interesting representations and processes in the brain, which is why event types are not only relevant in the technical sense.

To prevent an ad hoc selection of these event types, we analyzed the above-mentioned event definitions with the n-train procedure at 39.9 s learning (i.e. 2.1 s for all n classes) and 2.1 s decoding (i.e. one code repetition). For each of the event types, we added an event that marks the onset of trials to fit the large transient response to the onset of stimulation. In total, the models learnt three, two, five, three and two transient responses to the events defined for duration, on, components, contrast and change, respectively. All these transients were modeled as 300 ms responses.

The top three models were the duration event (85.7%), contrast event (85.7%) and component event (85.0%), which did not statistically differ from each other ($p > 0.214$, Wilcoxon signed-rank test). Both the on event (58.1%) and change event model (79.1%) did significantly differ from any other model ($p < 0.001$, Wilcoxon signed-rank test). From this point onwards, all encoding models used the duration event.

3.1.4. Encoding model achieves high classification accuracy in a short learning time

We compared the classification accuracy of the four training regimes by varying the amount of available

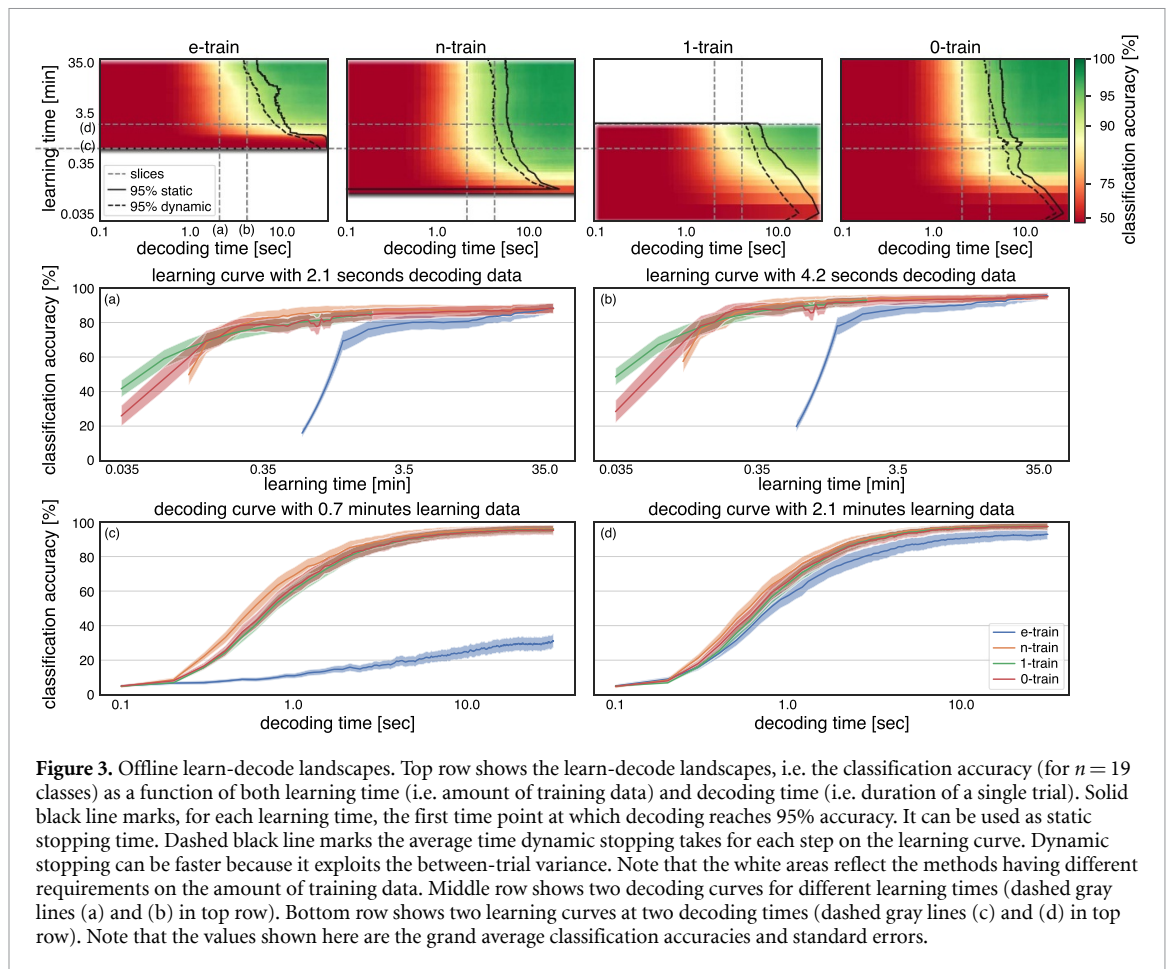


Figure 3. Offline learn-decode landscapes. Top row shows the learn-decode landscapes, i.e. the classification accuracy (for $n = 19$ classes) as a function of both learning time (i.e. amount of training data) and decoding time (i.e. duration of a single trial). Solid black line marks, for each learning time, the first time point at which decoding reaches 95% accuracy. It can be used as static stopping time. Dashed black line marks the average time dynamic stopping takes for each step on the learning curve. Dynamic stopping can be faster because it exploits the between-trial variance. Note that the white areas reflect the methods having different requirements on the amount of training data. Middle row shows two decoding curves for different learning times (dashed gray lines (a) and (b) in top row). Bottom row shows two learning curves at two decoding times (dashed gray lines (c) and (d) in top row). Note that the values shown here are the grand average classification accuracies and standard errors.

training data while also varying the amount of available testing data, estimating so-called learn-decode landscapes (see the first row in figure 3). This representation is a generalization of the learning curve and decoding curve, which, vice versa, are slices through the learn-decode surface (see the second and third row in figure 3). Note that the different methods impose different requirements on the training data, which is why for some methods there are blank areas in these landscapes. Specifically, e-train requires a minimum of 2.1 s (one code repetition) for each of the n classes to measure ERPs, n-train requires some data from all classes, while 1-train can maximally use $1/n$ of the total data set because it is only trained on one class. Note also that the decoding is always performed equally: first, decoding is done on exactly the same hold-out folds of the data set, and second, all classifiers are static during decoding, so the adaptivity of 0-train is turned off at this stage. This makes the learn-decode landscapes comparable between methods. In addition to varying the decoding time (static stopping), we estimated the accuracy with the proposed stopping rule (dynamic stopping, see methods).

As a first comparison between the methods, we zoom in on the learning time-step at which the bare minimum of training data is used (limited by e-train, $2.1 \times 19 = 39.9$ s) and using one code repetition for classification (2.1 s). This is the cross-section of slices

A and C in figure 3. At this point, the classification accuracies are 18.9%, 85.2%, 80.5% and 81.3%, for e-train, n-train, 1-train and 0-train, respectively (see table 1). Individual comparisons revealed that the accuracy of e-train is significantly lower than any of the others ($p < 0.001$, Wilcoxon signed-rank test), and n-train performs significantly better than 1-train ($p = 0.002$, Wilcoxon signed-rank test). None of the other comparisons revealed a significant effect ($p > 0.015$, Bonferroni corrected).

As a second comparison between the methods, we zoom in on the learning time-step at which the maximum amount of training data is used (limited by 1-train, $31.5 \times 4 = 126$ s) and use one code repetition for classification (2.1 s). This is the cross-section of slices A and D in figure 3. At this point, the classification accuracies are 76.4%, 87.3%, 84.9% and 84.7%, for e-train, n-train, 1-train and 0-train, respectively (see table 1). At this point, all comparisons are significantly different ($p < 0.001$, Wilcoxon signed-rank test), except for the comparison between 0-train and 1-train ($p = 0.227$).

As a last point of comparison, we compare e-train, n-train and 0-train at the maximum learning step ($4 \times 19 \times 31.5 = 2394$ s or 39.9 min) to test the convergence of these methods. Note that the 1-train method is left out because there are not enough 1-class data available in the offline data set. At 2.1 s

Table 1. Offline static decoding versus learning time. The grand average classification accuracies (in percentage correctly classified trials) at a decoding time of 2.1 s, for each of the four methods (e-train, n-train, 1-train, 0-train) and for three learning times: 39.9 s (minimum for e-train), 126 s (maximum for 1-train), 39.9 min (maximum data set). In addition to the grand averages, the standard errors are given. Note that for 39.9 min learning there is not enough data available for 1-train; hence the performance is not provided (n.a.).

	Learning time		
	39.9 s	126 s	39.9 min
e-train	18.9 ± 1.8	76.4 ± 4.6	88.3 ± 2.8
n-train	85.2 ± 3.4	87.3 ± 2.8	88.0 ± 2.7
1-train	80.5 ± 3.8	84.9 ± 3.2	88.2 ± 2.7
0-train	81.3 ± 3.5	84.7 ± 3.1	n.a.

decoding time, the classifications are 88.3%, 88.0% and 88.2% for e-train, n-train and 0-train, respectively (see table 1). None of these is significantly different ($p > 0.059$, Wilcoxon signed-rank test).

Finally, the reader can observe a sudden decrease in classification accuracy in e-train (at around 10 min learning) and 0-train (at around 31.5 s learning). In the offline data set, four participants perform poorly compared to the other participants. Participants 10 and 12 show lower classification accuracies in general (which coincides with the decrease of e-train), while participants 24 and 30 show poor classification in only a subset of trials (which coincides with the decrease of 0-train). There is no clear pattern in (loss of) the data quality of these participants compared to other participants. Even though these poor results decrease the grand average classification accuracy, we decided not to remove these participants from the offline data set to provide a complete view of the four methods and how they perform in these situations. For instance, the results show that 0-train can recover from these bad epochs. In addition, note that figure 3 shows results from static-stopping. A dynamic-stopping procedure would ideally not emit a classification until it is certain or otherwise rejects the trial under these circumstances.

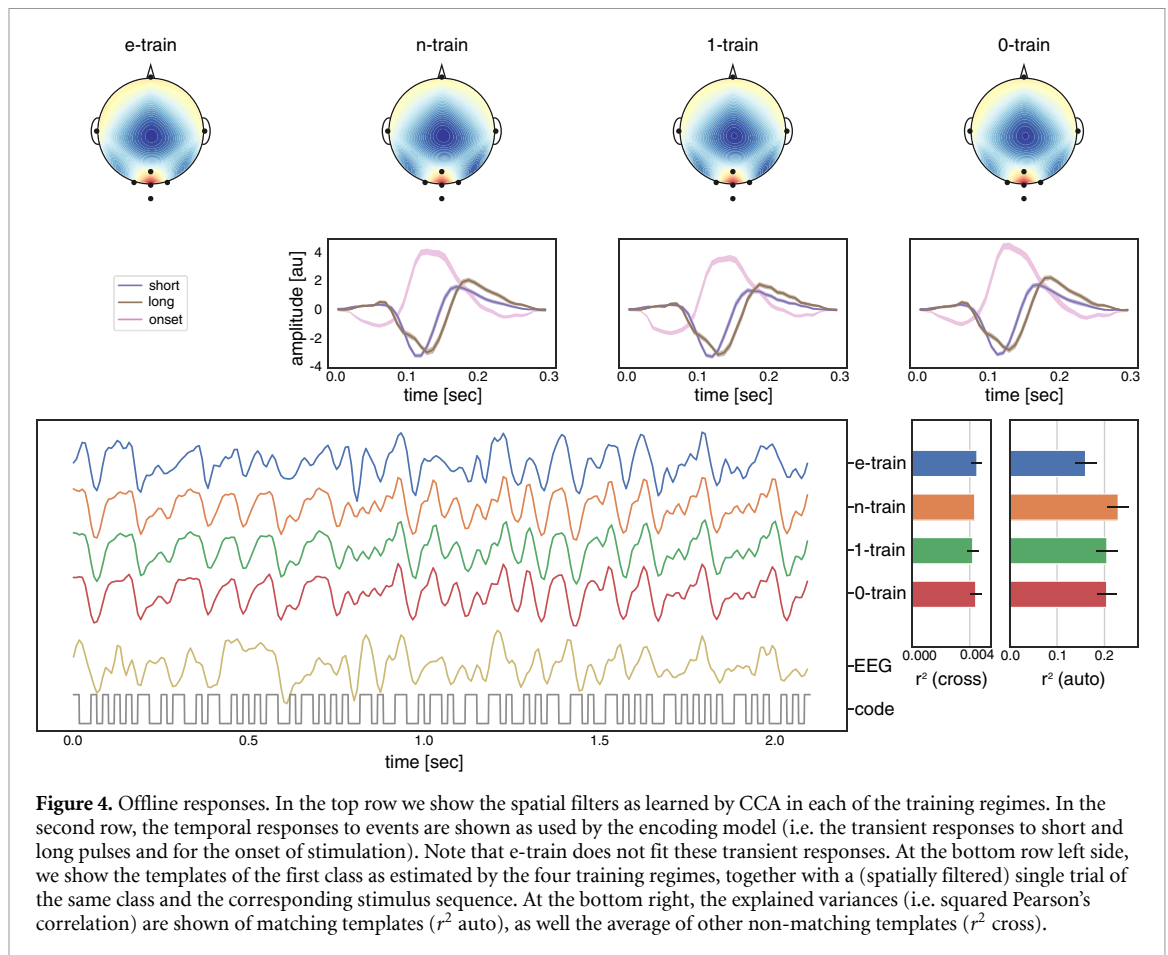
3.1.5. Encoding model learns responses accurately in a short time

We further investigate what the four training regimes learn by inspecting their spatial filters, transient responses and template responses (see figure 4). We do so at a learning time of $19 \times 6.3 = 4 \times 29.925 = 119.7$ s (limited by 1-train) of training data, so that the methods are directly comparable. First, all four training procedures optimized a spatial filter with CCA. For each of the procedures, the grand average spatial filter shows a dominant peak at the occipital pole at electrode Oz (see the top row in figure 4). Second, the n-train, 1-train and 0-train procedures rely on an encoding model that also learns a temporal response, consisting of three transient responses (short flash, long flash, onset of stimulation; see the second row in figure 4). These responses, shared over

the three training methods, show a dominant peak around 100–125 ms, typical for a standard flash VEP. Third, we show the template responses of participant 1 for class 1 for each of the four training methods together with a single-trial EEG time-series and the underlying stimulation bit-sequence (see bottom row left figure in figure 4). Note that each of the methods captures the complicated pattern in the neural activity very well, which is also expressed by the grand average explained variances (see bottom row right two figures in figure 4). First, we estimate how much of the variance of spatially filtered single-trial EEG is explained by the model, as given by the squared Pearson's correlation between templates and single trials that share the same labels. These were $r^2 = 0.160$, $r^2 = 0.229$, $r^2 = 0.205$ and $r^2 = 0.205$, for e-train, n-train, 1-train and 0-train, respectively (see figure 4 right barplot, auto). All comparisons were significantly different ($p < 0.001$, Wilcoxon signed-rank test), except for the difference between 1-train and 0-train ($p = 0.289$). Second, we estimated the grand average explained variance of single trials with competing templates that carry a different label (see figure 4 left barplot, cross). These were $r^2 = 0.004$ for each of the methods, meaning there is minimal confusion with incorrect labels.

3.1.6. Zero-training encoding model allows shorter session times

As a final comparison, we analyzed the performance of the four training regimes in the scope of productivity of a full (train plus test) session. Throughout the BCI literature, the ITR has been one of the main performance measures, which aside from classification accuracy takes the decoding time as well as the number of classes into account. However, it does not project how misclassification triggers the drawback of having to correct the already typed symbol. For this, the BCI field introduced the symbols per minute (SPM) measure, which accounts for the required correction and re-selection of misspelled symbols. Unfortunately, these measures do not incorporate the time it took to train the BCI. Here, we propose a session productivity characterization, based on a number of symbols to transmit, which is known beforehand. We took an arbitrary 50 symbols and estimated the total time it took to perfectly decode, learn and spell that number of symbols. For this, we ran 100 simulations of an online experiment using the offline data, randomly allocating the backspace to a particular symbol and forcing all 50 symbols to be correctly spelled. For the e-train, n-train and 1-train methods, we optimized the training time using figure 3 to minimize the total session time to spell 50 symbols. Note that this might yield an overfitted session time for these methods, but our aim is to compare them to a non-overfitted 0-train. Then, each of the four methods was run using the dynamic stopping procedure to reach the 50 symbols as fast as possible.



The number of emitted symbols evolving over the session time is shown in the top row left figure of figure 5. Note that 0-train starts spelling immediately because it has no learning phase (other than the first trial), while 1-train and n-train need a short learning period. The e-train regime requires a relatively long learning period. These numbers are reflected in the session times, which are 8.3, 4.5, 5.0 and 3.5 min for e-train, n-train, 1-train and 0-train respectively, for which all pairwise comparisons are significantly different ($p < 0.002$, Wilcoxon signed-rank test). During this session time, the SPM is 13.2, 17.2, 15.0 and 17.5 symbols per minute for e-train, n-train, 1-train and 0-train, respectively, which are significantly different for all pairs ($p < 0.001$, Wilcoxon signed-rank test) except for the comparison between n-train and 0-train ($p = 0.339$). In addition, the ITR is 55.8, 72.6, 63.4 and 74.0 for e-train, n-train, 1-train and 0-train, respectively, again for which all pairs are significantly different ($p < 0.001$) except between n-train and 0-train ($p = 0.439$). These results are summarized in table 2.

3.2. Online

We employed a real-time closed-loop zero-training BCI using eight practical water-based EEG electrodes with the aim to validate our proposed calibration-free

method in an online experiment. The BCI presented a 29-key speller grid with which participants were asked to spell sentences. Participants fixated on the target character eliciting a specific kind of cVEP. The BCI used the elicited cVEP to detect the attended cells from the recorded brain activity and provided feedback upon selection, allowing participants to spell sentences with a calibration-free BCI. Participants performed three separate blocks: spelling the alphabet, a predetermined sentence and a sentence of their choice.

The results of the three online blocks are summarized in table 3. Overall, only three participants made a single mistake, which they noticed and corrected immediately. The best participant (sub-01) spelled symbols at a speed of 1.7 s per selection (excluding 1 s inter-trial interval), which makes up for 106.7 bits per minute (22 symbols per minute, including inter-trial time) in block 2. On average, the participants spelled symbols using 3.8 s of EEG data, which corresponds to an average ITR of 66.4 bits per minute and an SPM of 13.4 symbols per minute as averaged over blocks (both including inter-trial time). Note that these results include an initial warm-up period that increases the average decoding time for 0-train, which was also reflected in the offline experiment (see figure 5 top right).

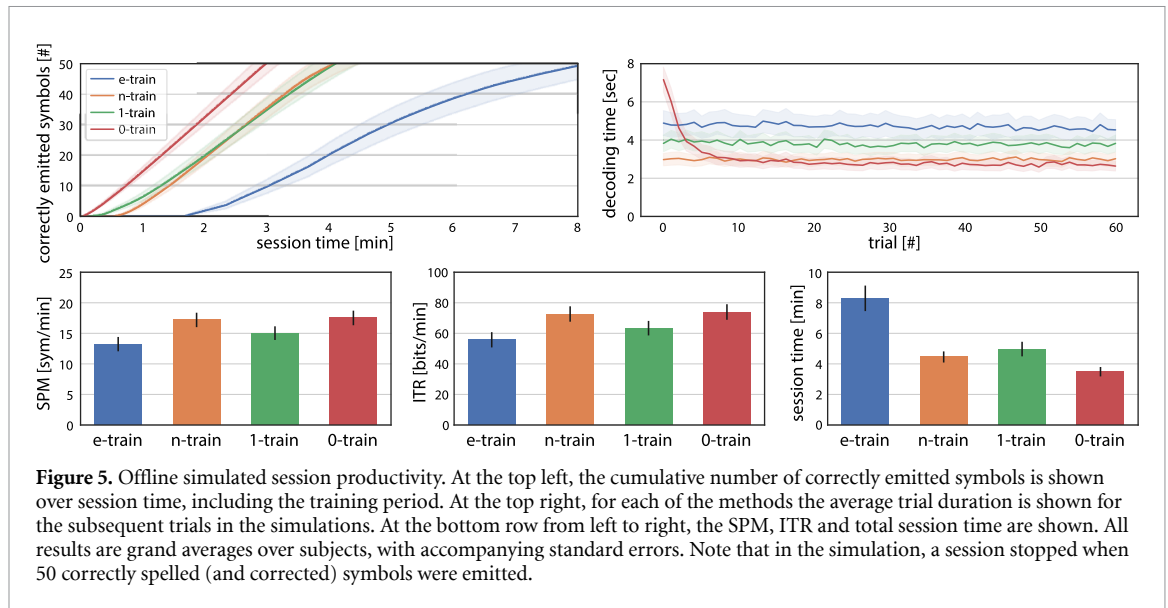


Figure 5. Offline simulated session productivity. At the top left, the cumulative number of correctly emitted symbols is shown over session time, including the training period. At the top right, for each of the methods the average trial duration is shown for the subsequent trials in the simulations. At the bottom row from left to right, the SPM, ITR and total session time are shown. All results are grand averages over subjects, with accompanying standard errors. Note that in the simulation, a session stopped when 50 correctly spelled (and corrected) symbols were emitted.

Table 2. The offline simulated session productivity. The grand average SPM rate in symbols per minute, ITR in bits per minute and session time in s. In addition to the grand averages, the standard errors are given. Note that these numbers are computed for simulated sessions that stopped once 50 correctly spelled (and corrected) symbols were emitted.

	SPM	ITR	Session time
e-train	13.2 ± 1.2	55.8 ± 5.0	8.3 ± 0.8
n-train	17.2 ± 1.2	72.6 ± 5.0	4.5 ± 0.4
1-train	15.0 ± 1.1	63.4 ± 4.8	5.0 ± 0.5
0-train	17.5 ± 1.2	74.0 ± 5.1	3.5 ± 0.3

4. Discussion

4.1. Encoding model reduces training time up to none at all

Using extensive offline analyses on a large data set of 30 participants, we showed that the traditional ERP training (e-train) can be made much more efficient by employing an encoding model that predicts EEG from the stimulus sequence (n-train, 1-train, 0-train). First and foremost, we observed a significantly higher classification accuracy of the encoding model in general compared to the ERP model at multiple points on the learn-decode landscapes (see figure 3). Specifically, n-train achieved a decent classification accuracy of 85% already after only 0.7 min learning and 2.1 s decoding, while e-train achieved a poor 19% at this point. Still, with 39.9 min learning and 2.1 s decoding, both methods performed equally well at 88% for $n = 19$ classes. These results clearly demonstrate the benefit of an encoding model, reducing the need for training data substantially while maintaining classification accuracy. In line, n-train spelled 50 symbols in a session time of only 4.5 min, while e-train required almost twice the time of 8.3 min (see figure 5).

We further reduced the amount of information in the training data set by learning the encoding model

using data from only a single class while generalizing to any other class. Despite the need to remove a faulty code sequence and the fact that the 1-train method was performing significantly lower (about a 3%–5% reduction), these results do show the possibility to calibrate a supervised system on only one stimulus class when an encoding model is available. This can greatly reduce the requirements on the BCI setup. Further research is needed to improve the generalizability of the 1-train model.

Most importantly, we have shown that the encoding model can be employed in a zero-training fashion, fully eliminating the calibration phase. Specifically, we have shown that the 0-train method can be slightly behind the n-train model (i.e. an initial warm-up period), but can achieve similar or even higher performance because of its adaptivity. This is demonstrated by the session performance in which n-train and 0-train achieve similar SPM and ITR of more than 17 symbols per minute and more than 72 bits per minute (see figure 5). Furthermore, 0-train does not require a calibration phase, which gives 0-train a head start of 1 min over n-train, thereby allowing it to have already spelled 15 symbols (in a simulation session of 50 correctly spelled symbols). On top of the benefit in the absolute number of spelled symbols, 0-train can remain adaptive throughout the session because of its semi-supervised learning method. In addition, this calibration-free method allows a plug-and-play BCI, which bypasses the need for a time-consuming passive training period, making the BCI more practical to use.

Finally, we have shown the feasibility of the calibration-free cVEP speller in an online study with nine participants. The participants were capable of spelling symbols on average with an ITR of 66.4 bits per minute and SPM of 13.4 symbols per minute. During an informal debriefing, the participants in general expressed that they

Table 3. The online performances. Participants used the zero-training classifier (0-train) to spell the alphabet (block-01), to copy-spell a given sentence (block-02) and to freely spell a sentence of their choice (block-03). For each, we show the number of trials (K); number of corrections (i.e. backspaces) (C); number of correctly spelled symbols (S); classification accuracy (P); average selection time in seconds excluding inter-trial time (T); ITR in bits per minute including the inter-trial interval of 1 s and SPM. Note that these results include the first trial, which had a forced minimum duration of 12 s.

Subject	Block-01: Alphabet							Block-02: Copy-spell							Block-03: Free-spell						
	K	C	S	P	T	ITR	SPM	K	C	S	P	T	ITR	SPM	K	C	S	P	T	ITR	SPM
Sub-01	27	0	27	100.0	1.9	102.2	21.0	39	0	39	100.0	1.7	106.7	22.0	28	0	28	100.0	1.9	100.1	20.6
Sub-02	27	0	27	100.0	6.3	40.1	8.2	39	0	39	100.0	5.6	44.2	9.1	47	0	47	100.0	8.4	30.9	6.4
Sub-03	27	0	27	100.0	2.8	77.5	16.0	39	0	39	100.0	3.3	67.7	13.9	21	0	21	100.0	2.8	76.5	15.8
Sub-04	27	0	27	100.0	2.5	82.8	17.1	39	0	39	100.0	2.9	75.0	15.4	60	1	58	98.3	3.3	64.3	13.8
Sub-05	27	0	27	100.0	3.3	67.6	13.9	39	0	39	100.0	6.3	40.0	8.2	44	0	44	100.0	4.4	53.5	11.0
Sub-06	27	0	27	100.0	3.8	60.8	12.5	39	0	39	100.0	3.0	73.3	15.1	40	0	40	100.0	3.1	71.3	14.7
Sub-07	27	0	27	100.0	2.6	79.9	16.4	39	0	39	100.0	2.8	77.3	15.9	46	1	44	97.8	4.0	55.3	12.0
Sub-08	27	0	27	100.0	3.5	64.8	13.3	41	1	39	97.6	3.0	69.1	15.1	38	0	38	100.0	2.7	78.8	16.2
Sub-09	27	0	27	100.0	6.4	39.2	8.1	39	0	39	100.0	4.1	57.4	11.8	35	0	35	100.0	6.9	37.1	7.6
Average	27	0	27	100.0	3.7	68.3	14.1	39	0	39	99.7	3.6	67.9	14.1	39	0	39	99.6	4.2	63.1	13.1

liked using the BCI and were amazed by its performance.

We believe our results have been achieved largely because of the encoding model (reconvolution), which estimates responses at the level of the events that make up a sequence rather than the response to full sequences of events (e.g. ERP). First, this reduces the number of parameters substantially and, since there are many repetitions of events within individual sequences, there is also much more data to estimate the responses to such events. Although the definition of events is non-linear, the composition of the overall response out of the event responses is linear. This simple model setup suffices to explain a large proportion of the single-trial variance (see figure 4) and allows for high-speed classification using only limited training data, up to none at all (see figure 5). We defined separate events for short and long flashes, allowing the model to handle sub-additive responses. Of course, other event definitions using more or fewer event types can be handled by the encoding model too and the responses to these events can also have varying response lengths.

4.2. Relation to the literature

Despite having lower ITR and SPM than other supervised cVEP BCIs (see e.g. [7, 12, 14, 41]), to our knowledge, this is the first high-speed zero-training cVEP BCI. So far, only one other study has shown a full zero-training procedure for cVEP BCI [27]. In their work, a language model is employed to post hoc find the most likely spelled word to constrain the model fit. Despite good classification accuracy, their model achieves an ITR of at maximum 35.7 bits per minute, which is almost half of the ITR that our zero-training method achieved on average in the online experiment (66.4 bits per minute). Recently, a transfer-learning cVEP BCI was proposed, though their model still required a small data set to compute the best subset of other participants to train the cross-participant classifier on [28].

It should be noted here that a full zero-training approach eliminating the entire calibration step is not always preferred. First, for some user groups a short calibration step might provide a covariance estimate that can kick-start the zero-training pipeline. In addition, it is an interesting but unexplored question whether transfer learning can provide such a kick-start too. Second, to study learning (or fatigue) effects in the user the availability of a proper baseline as a first supervised calibration would be a useful addition. Third, for subjects for whom the working of the BCI is essential and who are extremely motivated to make it work, a calibration phase that first needs to be passed successfully may be less stressful than repeatedly trying to make a zero-training BCI work. Finally, other paradigms might be useful to boost the system performance, such as neurofeedback

training prior to the actual use, which can substantially improve subsequent BCI performance [42]. In general, it is relevant to mention the (likely) dependence of the performance of a zero-training paradigm on the underlying SNR. If the SNR is too low, then it could be expected that the additional model constraints imposed by (supervised) calibration allow the BCI to find an accurate model whilst a zero-training paradigm might have difficulty in finding the signal in its larger model space.

4.3. From test to train to session performance

Performance measures of different BCIs can be difficult to compare. First, usually performance measures in BCI research are reported for the test session as average speed per selection, without taking into account the (long) training session done beforehand. Second, often the inter-trial interval is ignored, creating unrealistic expectations of real-time BCI performance. Third, the research might be done with over-trained highly focused participants, which is an unrealistic use-case in a health-care setting. For a good overview of the whole process of a full BCI session resulting in a successful communication of a given number of symbols, including training time, accuracy-speed trade-off and inter-trial time, different representations are needed. In figure 5, we show the BCI output in terms of correctly emitted output symbols against time starting at the beginning of the (training) session. Note that even cap fitting, preparation time and briefing may be included in these kinds of overviews. For instance, in this work we used only eight water-based electrodes in the online setup, which altogether can reduce the session time substantially. Most importantly, one can observe that our zero-training approach reaches the same detection speed as a supervised model, while already having emitted several symbols (see figure 5). These session-performance graphs can function both as pragmatic design guidelines for BCI applications with specific requirements, as well as fundamental insight into the character of different training regimes.

4.4. Implications for cognitive neuroscience

Apart from the pragmatic benefit of a faster (i.e. no) calibration method, the comparison between the methods reveals to what extent the underlying model, a decomposition into independent responses to a few event types, holds. We limit ourselves here to evoked responses, time-locked to a stimulus, while in principle there is no reason why there could not be induced responses (oscillations) present too. In this study, we ignored this possibility. An empirically collected ERP template (e-train) of the full response to a sequence can capture regularity of the brain processing that specific sequence, over and above what a model-based synthesized template can capture. This n-train approach is based on the kind

of event in the sequence only, not on their order of appearance. Thus, an ERP-based approach can in principle achieve a higher BCI performance than a modeled approach as it has more degrees of freedom. We found that with the maximum amount of training (39.9 min of data), n-train performs equal to e-train (about 88% classification accuracy at 2.1 s decoding, see figure 3). This means that the underlying encoding model is a good predictor and the EEG response to code-modulated stimuli does not depend on higher-order regularities. For instance, an ERP could capture entrainment (e.g. adaptation or habituation) in the evoked response to an occasional embedding of a repetition of a few similar events in succession (e.g. a local short steady-state response), while this cannot happen in a more constrained model such as reconvolution. Thus, an encoding model that is as successful as the ERP model demonstrates that higher-order patterns (e.g. non-linear ones) in the group of stimuli, such as the ones allowing for entrainment, do not evoke an important component in the EEG and the variance can be explained well already by the much more parsimonious event responses.

Likewise, we found that the 1-train model trained on only one sequence (1-train) was not quite as good as the one trained on all sequences (n-train), regardless of the class chosen to be trained on. Specifically, with the maximum amount of training for 1-train (126 s of data), 1-train achieved a classification accuracy of 84.9% while n-train achieved a significantly higher 87.2%. Therefore, the response to any pattern in the generated family of possible sequences (e.g. modulated Gold codes) generalizes not perfectly but quite well and can predict the response to any other one. This is a stringent requirement as it holds for any stimulus pattern in the set. Still, further research is needed to investigate the small decrease in performance when employing a 1-class model.

Even more so, for the 0-train paradigm it is not only important that the model trains well (i.e. delivers accurate event responses) when the provided stimulus sequence is correct, but that it does not fit well when provided with an incorrect sequence. This can happen when patterns are selected randomly and two classes happen to be very similar. In this case, n-train and 0-train would yield accurately trained event responses, but poor testing performance. In the case of 0-train, the training would also be difficult as it would not become clear which class the training data came from. In the simulated session performance (figure 5), 0-train (17.5 symbols per minute) achieved an SPM equal to that of n-train (17.2 symbols per minute). Thus, this is again a more stringent requirement of the match of the chosen space of sequences with the response model, a match that gained evidence by the 0-train paradigm working so well empirically.

4.5. Implications for cVEP BCI design

Finally, we want to highlight two important issues we addressed during this study. First, we found that one of the stimulation sequences performed significantly lower than the others in the 1-train method. We found that this sequence was one of the original m-sequences used in the generation of the set of Gold codes. In the literature, it is common to add these to the set of Gold codes [37], but it has other correlation properties and might therefore be less suitable for generalization. Second, we found a significant increase in classification accuracy when correcting for the monitor raster latency, which is in line with work from Nagel and colleagues [40]. Both issues complicate models such as 1-train and 0-train, as they are (in the first trial in the case of 0-train) trained on only one class. When not accounting for issues like these, they might learn class-specific statistics that do not generalize well.

5. Conclusion


In conclusion, we have shown the importance of a generative model for limiting the training data for a cVEP BCI up to no calibration at all, without loss of explanatory power or decoding performance. Apart from the direct kick-start, this opens up new directions of practical plug-and-play BCI that might facilitate easier adoption by healthy users as well as patients with short attention spans. The method is generic and allows for many classes to be detected without any prior data.

Acknowledgments

This work was supported by the Netherlands Organization for Scientific Research (NWO/TTW) as Take-off Grant No. 14054 and by the international ALS Association and the Dutch ALS Foundation under Grant Nos. ATC20610 and 2017-57. We thank Jop van Heesch for the development and support of the iOS iPad noise-tagging application. We thank Philip van den Broek and the Technical Support Group of the Faculty of Social Sciences at the Radboud University for the support with technical details of the experiment and the use of the BrainStream software package. We thank Jesse van der Spek for his help with participant recruitment and data acquisition.

ORCID iDs

J Thielen  <https://orcid.org/0000-0002-6264-0367>

J Farquhar  <https://orcid.org/0000-0002-8560-0712>

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] van Gerven M *et al* 2009 The brain–computer interface cycle *J. Neural Eng.* **6** 041001
- [3] Gao S, Wang Y, Gao X and Hong B 2014 Visual and auditory brain–computer interfaces *IEEE Trans. Biomed. Eng.* **61** 1436–47
- [4] Sutter E E 1984 The visual evoked response as a communication channel *Proc. Symp. Biosensors* pp 95–100
- [5] Sutter E E 1992 The brain response interface: communication through visually-induced electrical brain responses *J. Microcomput. Appl.* **15** 31–45
- [6] Bin G, Gao X, Wang Y, Hong B and Gao S 2009 VEP-based brain–computer interfaces: time, frequency and code modulations *IEEE Comput. Intell. Mag.* **4** 22–6
- [7] Bin G, Gao X, Wang Y, Li Y, Hong B and Gao S 2011 A high-speed BCI based on code modulation VEP *J. Neural Eng.* **8** 025015
- [8] Spüler M, Walter A, Rosenstiel W and Bogdan M 2014 Spatial filtering based on canonical correlation analysis for classification of evoked or event-related potentials in EEG data *IEEE Trans. Neural Syst. Rehabil. Eng.* **22** 1097–103
- [9] Thielen J, van den Broek P, Farquhar J and Desain P 2015 Broad-band visually evoked potentials: re(con)volution in brain–computer interfacing *PLoS One* **10** e0133797
- [10] Thielen J, Marsman P, Farquhar J and Desain P 2017 Re(con)volution: accurate response prediction for broad-band evoked potentials-based brain–computer interfaces *Brain–Computer Interface Research* (Berlin: Springer) pp 35–42
- [11] Spüler M, Rosenstiel W and Bogdan M 2012 One class SVM and canonical correlation analysis increase performance in a c-VEP based brain–computer interface (BCI) *ESANN*
- [12] Spüler M, Rosenstiel W and Bogdan M 2012 Online adaptation of a c-VEP brain–computer interface (BCI) based on error-related potentials and unsupervised learning *PLoS One* **7** e51077
- [13] Sato J and Washizawa Y 2016 Neural decoding of code modulated visual evoked potentials by spatio-temporal inverse filtering for brain–computer interfaces 2016 38th Annual Int. Conf. IEEE Engineering in Medicine and Biology Society (EMBC) (IEEE) pp 1484–7
- [14] Nagel S and Spüler M 2019 World’s fastest brain–computer interface: combining EEG2Code with deep learning *PLoS One* **14** e0221909
- [15] Verbaarschot C *et al* (in preparation) Assessing system performance and user experience of a code-modulated visual Brain–Computer Interface speller on patients with Amyotrophic Lateral Sclerosis
- [16] Farwell L A and Donchin E 1988 Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials *Electroencephalogr. Clin. Neurophysiol.* **70** 510–23
- [17] 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
- [18] Nakanishi M, Wang Y, Chen X, Wang Y T, Gao X and Jung T P 2018 Enhancing detection of SSVEPs for a high-speed brain speller using task-related component analysis *IEEE Trans. Biomed. Eng.* **65** 104–12
- [19] Rezeika A, Benda M, Stawicki P, Gembler F, Saboor A and Volosyak I 2018 Brain–computer interface spellers: a review *Brain Sci.* **8** 57
- [20] McFarland D J, Sarnacki W A and Wolpaw J R 2003 Brain–computer interface (BCI) operation: optimizing information transfer rates *Biol. Psychol.* **63** 237–51
- [21] Käthner I, Wriessnegger S C, Müller-Putz G R, Kübler A and Halder S 2014 Effects of mental workload and fatigue on the P300, alpha and theta band power during operation of an ERP (P300) brain–computer interface *Biol. Psychol.* **102** 118–29
- [22] Jayaram V, Alamgir M, Altun Y, Scholkopf B and Grosse-Wentrup M 2016 Transfer learning in brain–computer interfaces *IEEE Comput. Intell. Mag.* **11** 20–31
- [23] Kindermans P J, Tangermann M, Müller K R and Schrauwen B 2014 Integrating dynamic stopping, transfer learning and language models in an adaptive zero-training ERP speller *J. Neural Eng.* **11** 035005
- [24] Kindermans P J, Schreuder M, Schrauwen B, Müller K R and Tangermann M 2014 True zero-training brain–computer interfacing—an online study *PLoS One* **9** e102504
- [25] Hübner D, Verhoeven T, Schmid K, Müller K R, Tangermann M and Kindermans P J 2017 Learning from label proportions in brain–computer interfaces: online unsupervised learning with guarantees *PLoS One* **12** e0175856
- [26] Sato J and Washizawa Y 2015 Reliability-based automatic repeat request for short code modulation visual evoked potentials in brain computer interfaces *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual Int. Conf. IEEE (IEEE)* pp 562–5
- [27] Turi F, Gayraud T and Clerc M 2020 Auto-calibration of c-VEP BCI by word prediction *HAL Open Arch.* **1** 02844024
- [28] Huang Z, Zheng W, Wu Y and Wang Y 2020 Ensemble or pool: a comprehensive study on transfer learning for c-VEP BCI during interpersonal interaction *J. Neurosci. Methods* **343** 108855
- [29] Desain P W M, Thielen J, van den Broek P L C, Farquhar J D R 2019 Brain computer interface using broadband evoked potentials Google Patents US Patent 10,314,508
- [30] Lalor E C, Pearlmutter B A, Reilly R B, McDarby G, Foxe J J 2006 The VESPA: a method for the rapid estimation of a visual evoked potential *NeuroImage* **32** 1549–61
- [31] Nagel S and Spüler M 2018 Modelling the brain response to arbitrary visual stimulation patterns for a flexible high-speed brain–computer interface *PLoS One* **13** e0206107
- [32] Bin G, Gao X, Yan Z, Hong B and Gao S 2009 An online multi-channel SSVEP-based brain–computer interface using a canonical correlation analysis method *J. Neural Eng.* **6** 046002
- [33] Oostenveld R, Fries P, Maris E and Schoffelen J M 2011 FieldTrip: open source software for advanced analysis of MEG, EEG and invasive electrophysiological data *Comput. Intell. Neurosci.* **2011** 156869
- [34] Ahmadi S, Borhanazad M, Tump D, Farquhar J and Desain P 2019 Low channel count montages using sensor tying for VEP-based BCI *J. Neural Eng.* **16** 066038
- [35] Golomb S W 1982 *Shift Register Sequences: Secure and Limited-Access Code Generators, Efficiency Code Generators, Prescribed Property Generators, Mathematical Models* 3rd revised edn. (Singapore: World Scientific) (<https://doi.org/10.1142/9361>)
- [36] Gold R 1967 Optimal binary sequences for spread spectrum multiplexing (corresp.) *IEEE Trans. Inf. Theory* **13** 619–21
- [37] Meel J 1999 Spread spectrum (SS) introduction (De Nayer Instituut, Hogeschool Voor Wetenschap & Kunst, Sint-Katelijne-Waver, Belgium)
- [38] Capilla A, Pazo-Alvarez P, Darriba A, Campo P and Gross J 2011 Steady-state visual evoked potentials can be explained by temporal superposition of transient event-related responses *PLoS One* **6** e14543
- [39] Notbohm A, Kurths J and Herrmann C S 2016 Modification of brain oscillations via rhythmic light stimulation provides evidence for entrainment but not for superposition of event-related responses *Front. Hum. Neurosci.* **10** 10
- [40] Nagel S, Dreher W, Rosenstiel W and Spüler M 2018 The effect of monitor raster latency on VEPs, ERPs and

- brain–computer interface performance *J. Neurosci. Methods* **295** 45–50
- [41] Gemblar F, Stawicki P, Saboor A and Volosyak I 2019 Dynamic time window mechanism for time synchronous VEP-based BCIs-performance evaluation with a dictionary-supported BCI speller employing SSVEP and c-VEP *PLoS One* **14** e0218177
- [42] Wan F, Da Cruz J N, Nan W, Wong C M, Vai M I and Rosa A 2016 Alpha neurofeedback training improves SSVEP-based BCI performance *J. Neural Eng.* **13** 036019