

Foundations of Conditional Diffusion Models for Event-Related Potentials

Author: Guido Klein

Radboud Universiteit



Supervisors: Pierre Guetschel, Gianluigi Silvestri, and
Dr. Michael Tangermann

Master's Thesis

Radboud University

Faculty of Social Sciences

Artificial Intelligence - Intelligent Technology

21-08-2024

S1021586

Abstract

Generative models, specifically diffusion models, can alleviate data scarcity in the brain-computer interface field. Moreover, they could potentially offer a promising solution to calibration sessions. While diffusion models have previously been successfully applied to electroencephalogram (EEG) data, existing models lack flexibility regarding sampling and often require alternative representations of the EEG data. To overcome these limitations, we introduce a novel approach to conditional diffusion models that utilizes classifier-free guidance to directly generate event-related potential (ERP) EEG data that is specific to a combination of labels in the dataset. Moreover, we evaluate the model's ability to generate samples for label combinations excluded during training, demonstrating its potential for transfer learning. In addition to commonly used metrics, domain-specific metrics are introduced to evaluate the specificity and quality of the generated samples. The results indicate that the proposed model can generate ERP EEG data that resembles real data for each combination of labels in each dataset. Furthermore, the model is capable of within-session between-class and between-class transfer learning, while between-session transfer learning remains elusive. The code used for this work is released at: https://gitlab.socsci.ru.nl/neurotech/code/thesis_gui_dokleijn

Contents

1	Introduction	4
1.1	EEG	4
1.2	Diffusion model	5
1.2.1	SMLD	5
1.2.2	DDPM	6
1.2.3	Conditional diffusion models	6
1.3	Related work	7
1.4	Aims and research questions	8
2	Methodology	10
2.1	Data Description	10
2.1.1	Visual ERP	10
2.1.2	Auditory ERP	11
2.1.3	Aphasia	11
2.1.4	Preprocessing	12
2.2	Model	13
2.3	Conditioning	14
2.3.1	Influence of existing models	14
2.3.2	Embedding strategies	15
2.3.3	Label dropout	15
2.4	Forward process	16
2.5	Backward process	17
2.6	Training	17
3	Metrics	17
3.1	Image-domain metrics	18
3.1.1	EEGNet	18
3.1.2	Inception score	19
3.1.3	Fréchet Inception distance	19
3.2	Classifier performance	20
3.3	Domain-invariant metrics	20
3.3.1	Sliced Wasserstein distance	20
3.3.2	Euclidean distance	21
3.3.3	Mean squared error	21
3.4	Domain-specific metrics	21
3.4.1	Evoked metrics	21
3.4.2	Power spectral density metrics	23
3.4.3	Standard deviation metrics	23
4	Visual inspection	23
4.1	Evoked response plot	24
4.2	Spatial covariance matrix plot	24
4.3	Power spectral density plot	24
4.4	Topographic map	24

5	Experiment 1: within-dataset generation	24
5.1	Results: visual ERP dataset	25
5.2	Results: auditory ERP dataset	29
5.3	Results: aphasia dataset	32
6	Experiment 2: outside-dataset generation	36
6.1	Transfer learning scenarios	36
6.1.1	Within-session, between-class transfer learning	36
6.1.2	Between-class transfer learning	37
6.1.3	Between-session transfer learning	37
6.2	Results	37
7	Discussion	43
7.1	Research Question 3	43
7.2	Research Question 4	43
7.3	Limitations	44
7.4	Applications	45
8	Conclusion	45
9	Future work	46
10	References	50
11	Abbreviations	54
A	Appendix	57
A.1	SDE hyperparameters	57
A.2	Mathematical equivalence of DDPM and SMLD	58
A.3	Transfer learning: Auditory ERP dataset	60
A.4	Transfer learning: Aphasia dataset	61
A.5	Effect of label dropout	61
A.5.1	Visual ERP dataset	61
A.5.2	Auditory ERP dataset	64

1 Introduction

1.1 EEG

EEG is one of the most widely used neuroimaging methods in the field brain-computer interfaces (BCIs). Electroencephalogram (EEG) uses electrodes to measure electrical potentials on the scalp. This has a high temporal resolution [1], as the potentials that originated at the dendrites propagate quickly through the brain and the scalp. However, this also has a relatively low spatial resolution because the signal that is measured at each electrode is a mixture of many neuronal and non-neural sources [2]. The signal from non-neural sources can be filtered out by rejecting data that has a biologically implausible amplitude, referred to as peak-to-peak rejection. Furthermore, not all signals that originated at the neurons are task related, which results in a low signal-to-noise ratio (SNR) [2]. Therefore, it is necessary to average over multiple trials of the same task, as this means that the non-task related signals cancel out (assuming this is a Gaussian with a mean of 0) while task related signals remain.

Two of the most prominent EEG paradigms in the field of BCIs are the motor-imagery and the oddball paradigm. The motor-imagery paradigm works by cueing the participant to imagine a certain movement (e.g., imagining moving the right hand). This leads to an event-related (de-)synchronization, which results in a decrease/increase in a specific band power over a specific part of the brain [1]. However, this is an induced change in band power, which means that is not phase-locked. Therefore, simply averaging across trials will cancel out the oscillations. Thus, the data first has to be converted to the frequency domain, after which it can be averaged.

On the other hand, the oddball paradigms work by presenting a relatively infrequent stimulus (target) in a stream of other stimuli (non-target). Presenting the target stimulus evokes a P300 response in the brain, which is a positive deflection in the EEG signal over the parietal cortex approximately 300 ms after stimulus onset [3]. This P300 response is both time and phase-locked, which means that simple averaging is possible [2]. The averaged signal is often referred to as an event-related potential (ERP) response.

Despite its popularity, EEG-based BCI systems also face several challenges. Firstly, there is a shortage of labelled data, which hampers the exploration of more complex classifiers, such as deep neural networks [4]. Secondly, most pipelines that are currently used to classify EEG data require a calibration session in which a classifier is trained. This takes up valuable time and energy of both the experimenter and the participant [4].

To alleviate these challenges, diffusion models, which are a type of generative model, offer a promising solution. These models have been used to generate high-quality synthetic data across a wide variety of domains, including images [5], audio [6] and chemistry [7]. Applying them to EEG data could increase the amount of labelled data. Moreover, it could potentially achieve transfer learning by predicting data of future sessions, which would reduce the need for a calibration session.

1.2 Diffusion model

To apply diffusion models to EEG data, it is vital to first understand the principles of these models. Generative models model the probability distribution of real data based on a collection of data points $(\mathbf{x}_i \sim p_{data}(\mathbf{x}))$. Subsequently, new data points can be generated by sampling from the modelled probability distribution. However, directly modelling the probability distribution is intractable when the data is high dimensional [8].

1.2.1 SMLD

To address this challenge, Song and colleagues proposed to use the gradient of the log probability density function $\nabla_{\mathbf{x}} \log p_{data}(\mathbf{x})$, referred to as the score, which avoids the complexities of modelling the probability distribution directly [9]. The score is equivalent to a vector field, where each hypothetical position in the space is characterized by a vector that has a direction and magnitude, representing the gradient. A model $\mathbf{s}_{\theta}(\mathbf{x})$ parameterized by θ is used to estimate this score using a score-matching loss, which is the L2-norm between the actual score and the estimated score [9]:

$$\frac{1}{2} \mathbb{E}[\|\mathbf{s}_{\theta}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_{data}(\mathbf{x})\|_2^2] \quad (1)$$

Sampling, also known as the backward process, is then done using Langevin dynamics, which iteratively takes a step (with step size ϵ) on the score to gradually change a sample from a prior distribution π (e.g., Gaussian with a fixed mean and variance) into a sample from the real data probability density function [9]. Additionally, a small amount of noise $\boldsymbol{\eta}_t \sim \mathcal{N}(0, \mathbf{I})$ is added to avoid the sampling from collapsing on the mode [9, 10]:

$$\tilde{\mathbf{x}}_t = \tilde{\mathbf{x}}_{t-1} + \frac{\epsilon}{2} \nabla_{\mathbf{x}} \log p_{data}(\tilde{\mathbf{x}}_{t-1}) + \sqrt{\epsilon} \boldsymbol{\eta}_t \quad (2)$$

Thus, if the model is able to accurately predict the score, then we can use the output of the model $\mathbf{s}_{\theta}(\mathbf{x})$ instead of the score $\nabla_{\mathbf{x}} \log p_{data}(\mathbf{x})$ to generate samples from the original data distribution. This type of model is referred to as score matching with Langevin dynamics (SMLD).

However, a model trained using the previously mentioned score-matching loss achieves suboptimal results due to three reasons [9]. Firstly, estimating the score in regions with only a few data points is inherently inaccurate. Secondly, if the data lies on a low dimensional manifold within a high dimension space, all points that are not on the low dimensional manifold have an undefined likelihood. Lastly, if two high-density regions are separated by a low-density region, then the gradient in the low-density region might not be informative about a difference in importance of the two high-density regions.

Fortunately, these issues can be mitigated by injecting various amounts of noise into the data and training the model to estimate the score while taking into account the amount of noise injected. The process of noise injection into

the data is called the forward process. This forward process can be viewed as a stochastic differential equation (SDE), in which the data distribution is gradually transformed into a prior distribution [11]. SDEs are a continuous stochastic process, in which each timestep can be indexed by $t \in [0, T]$. The transition kernel $p_{st}(\mathbf{x}(t)|\mathbf{x}(s))$ from $\mathbf{x}(s)$ to $\mathbf{x}(t)$, under the constraint that $0 \leq s < t \leq T$, is a Gaussian with a mean and variance that can be computed in closed form under certain conditions [11].

However, the model, training, and sampling procedure described earlier cannot deal with this noise injection. Therefore, a few adaptations have to be made to accommodate this change. Firstly, the prediction of the model is made dependent (i.e., conditioned) on the timestep, by adding the timestep as input to the model $\mathbf{s}_\theta(\mathbf{x}_t, t)$. Secondly, the score-matching objective is changed to weight each score based on the timestep of the score using a positive weighting function λ [9, 11]:

$$\mathbb{E}_t \int \lambda(t) \mathbb{E}_{\mathbf{x}(0)} \mathbb{E}_{\mathbf{x}(t)|\mathbf{x}(0)} [\|\mathbf{s}_\theta(\mathbf{x}(t), t) - \nabla_{\mathbf{x}(t)} \log p_{0t}(\mathbf{x}(t)|\mathbf{x}(0))\|_2^2] g \quad (3)$$

Lastly, the Langevin dynamics is no longer necessary for sampling, as the SDE can be rewritten to a reverse-time SDE, which only requires the parameters of the forward SDE and the score at each time point $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ [11]. Thus, only a numerical SDE solver is necessary to perform the sampling. However, first applying the SDE solver as a predictor and then correcting the marginal of the predicted output using a corrector, such as Langevin dynamics, has been shown to improve the sampling quality [11]. This type of sampling is called Predictor-Corrector (PC) sampling.

1.2.2 DDPM

The SMLD framework can be shown to be mathematically equivalent to the denoising-diffusion probabilistic model (DDPM) framework under some constraints. This equivalence is briefly explored in section A.2. However, to understand the rest of this work, it is merely required that one knows that the forward process in the DDPM framework is modelled as a discrete Markov chain, while in the SMLD framework the noise injection is modelled using a continuous SDE (see figure 1).

1.2.3 Conditional diffusion models

In the previous section, we have briefly touched on the notion of conditioning, where the framework requires the model to exploit information about the noise added in the forward process, given by t , to accurately predict the score of the data noised to that timestep.

Essentially, the model performs multitask learning, with the weights being adapted based on timestep t . Beyond conditioning on the noise level, the model can also be conditioned on other variables that influence the score. For example, images labelled as “nature” or “pool” are likely to have completely different probability distributions, so the score that is predicted by the model should

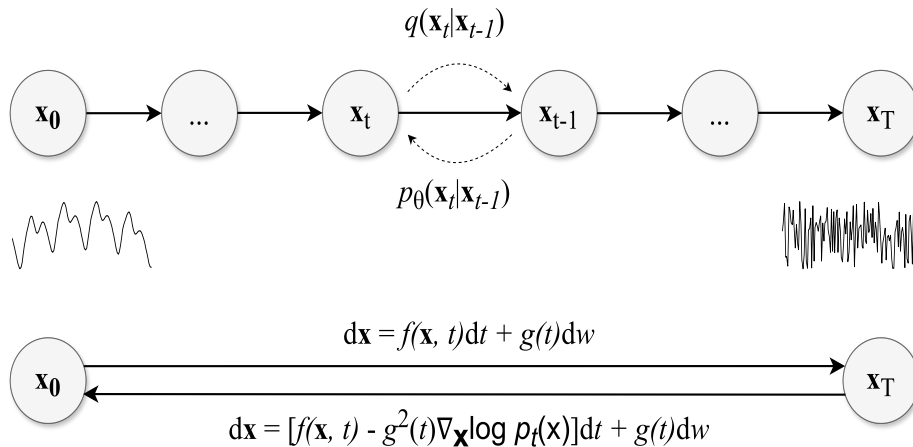


Figure 1: A diagram comparing the Markov chain of a DDPM with the SDE of a SMLD.

vary depending on the image we want to generate. Therefore, if the label \mathbf{c} of the image is given as an additional input, denoted as $\mathbf{s}_\theta(t, \mathbf{c})$, the model can adjust its weights to predict a score that matches the probability distribution of images with that particular label. This allows for generating images that belong to one user-imposed class during sampling. This type of model referred to as a conditional diffusion model, even though technically speaking all diffusion models are conditional.

Conditioning can be achieved through so-called classifier-free guidance (CFG), where the final score is computed as a combination of the estimated score of the unconditioned and conditioned diffusion model [12]:

$$\mathbf{s}_\theta(t, \mathbf{c}) = (1 + w)\mathbf{s}_\theta(t, \mathbf{c}) - w\mathbf{s}_\theta(t) \quad (4)$$

It seems that this would require training both a conditional and unconditional model. However, it is possible to train a single model that acts as both the conditioned and unconditioned model, by replacing some of the labels with a null-token. This null-token can be used during sampling to retrieve the unconditioned score [12].

1.3 Related work

Over the last two years, multiple articles have applied diffusion models to EEG data. In this body of literature, a shift can be observed away from unconditional EEG-domain diffusion models and image-domain inspired conditional diffusion models towards conditional diffusion models that sample EEG data directly. The unconditioned models are often validated using image-domain and domain-invariant metrics [13, 14], while conditioned models are typically validated by

comparing classifier performance with and without dataset augmentation [15–18].

Most initial studies relied on unconditional diffusion models to generate EEG data. For example, Vetter et al. (2023) and Torma et al. (2023) both successfully generated visual ERP EEG data [13, 19]. Additionally, Aristimunha et al. (2023) demonstrated that it was also possible to generate encoded sleep stage EEG data with a power spectral density (PSD) similar to the real data [14]. These studies utilized the Fréchet inception distance (FID) [13, 14], inception score (IS) [13], multi-scale structural similarity index metric (MS-SSIM) [13, 14] and sliced Wasserstein distance (SWD) [13] to validate the quality of the generated EEG data.

In another approach, some studies applied image-domain conditional diffusion models to EEG data, by converting EEG data to a representation that is more similar to images (e.g., spectrograms) [15, 16, 20]. Even though the benefit of conditioning is a significant contribution, this approach does require an additional encoding and decoding step, which could reduce sampling quality due to the lack of end-to-end learning. Validation of these models relied on improvements in classifier performance due to data augmentation [15, 16, 20] and the Jensen-Shannon Divergence (JSD) [15].

Recently, conditional diffusion models that can sample EEG data directly have become the norm. Shu et al. pioneered this approach in a publication in June 2023, by creating a model that can generate spectrogram-specific EEG data [17]. A study published in February 2024 also showed that it is possible to condition on encodings created by a classifier trained on predicting major depressive disorder [18]. These studies solely relied on classifier performance to evaluate the generated samples [17, 18].

1.4 Aims and research questions

Despite these advancements, the field of conditional diffusion models for EEG data remains largely unexplored. Therefore, many design choices have to be made without the guidance of existing literature. Furthermore, there is a noticeable absence of metrics that are designed to assess EEG-specific features, as all previous work either uses classifier performance, domain-invariant metrics or image-domain metrics. Additionally, while image-domain conditional diffusion models can rely on highly sophisticated language embedding models to condition the model on multiple labels (i.e., the words in a sentence) simultaneously [21], these models are unfortunately not available in the BCI domain. Moreover, none of the conditional diffusion models available in the BCI domain are conditioned on more than one label. However, the ability to condition on multiple labels simultaneously makes it possible to train on a wide range of conditions, which should make it possible to 1) flexibly sample from a wide variety of conditions and 2) test transfer learning by removing a specific combination of labels from the training data.

To address these gaps in the literature, the following research questions are answered in this work:

1. Which design choices have to be made to implement a conditional diffusion model for EEG data?

This can be achieved in two ways: by adapting the image-domain conditional diffusion models to the EEG-domain, or by making the EEG-domain unconditioned diffusion models conditioned. We briefly explored both options, but ultimately chose to adapt an unconditioned EEG-domain diffusion model, as the neural networks that are often used in the image-domain conditional diffusion models (i.e., U-Net) proved to work rather poorly with EEG data directly.

2. What metrics can be introduced to assess the quality of the generated ERP EEG data compared to the real data?

In addition to the image-domain, domain-invariant and classifier performance metrics, we also introduce domain-specific metrics that are able to capture EEG-specific features. These metrics, designed to exploit the phase- and time-locked nature of ERP EEG data, should make it more straightforward to assess the capabilities of the model.

3. Given the design choices made as a result of the first research question, it is possible to generate ERP EEG data that is specific to a particular combination of labels in the training set that is similar to the real data of that same combination, as measured by the metrics that are implemented as a result of the second research question?

This is answered by training a conditional diffusion model and sampling ERP EEG data using the labels that are present in the dataset. This sampled ERP EEG data is subsequently evaluated using image-domain, domain-invariant, classifier performance, and domain-specific metrics. The scores obtained by the sampled ERP EEG data are compared to a baseline established on real data, which should give a rough indication of how similar the sampled data is to the real data of that particular combination.

4. To what extent is it possible to generate data ERP EEG data for a specific combination of labels, which is absent from the training set, that is similar to the real data of the same combination, as measured by the metrics that are implemented as a result of the second research question?

To answer this question, the conditional diffusion model is trained on a training dataset, which has specific combinations of data removed (e.g., the target data of the second session of the first participant). This removed combination is subsequently sampled, which entails that the models has to generalize the distribution of the present combinations to this new combination. This transfer learning is tested in a variety of ways, depending on information in the meta-data of the dataset. The scores of the generated data will be compared to the scores of the model trained on the complete dataset (i.e., the model trained for research question three) to evaluate the impact of the data removal.

2 Methodology

The methodology describes all the choices made to answer the first research question, culminating in a diffusion model that can be trained on ERP EEG data directly and is conditioned on multiple labels in parallel.

2.1 Data Description

Three ERP datasets are used to train the diffusion model. All three datasets are based on the P300 oddball paradigm. However, despite the similarity in overall paradigm, there are major design differences between the three datasets. This allows us to evaluate the efficacy of the model to generate ERP EEG under a variety of circumstances. The intricacies of each dataset are briefly discussed in the following section.

2.1.1 Visual ERP

The visual ERP dataset was recorded by Lee and colleagues, who were interested in studying BCI-deficiency across three major paradigms: visual ERP, motor imagery, and steady-state visually evoked potential protocols [22]. In the visual ERP paradigm, the P300 was evoked with a spelling-copy task, which was performed on a 36-symbol visual matrix speller. An example of a matrix speller can be seen in figure 2. Participants were instructed to focus on one symbol, corresponding to a letter in a sentence they were asked to copy. Two adaptations were made to the standard paradigm. Firstly, the P300 response was enhanced by using a familiar face stimulus as intensification [23]. Secondly, random-set presentation was employed to mitigate the adjacency-distraction errors. Adjacency-distraction errors are caused by the participant shifting their attention from the target symbol to a non-target symbol in response to an intensification of a non-target symbol close to the target symbol, resulting in a semi-systematic error [3].

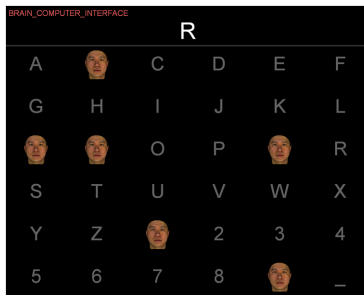


Figure 2: An example of a matrix speller that uses random-set presentations and face stimuli as intensification. Reprinted from Yeom et al. (2014) [24].

In the study, 54 participants underwent two sessions which were held on

different days [22]. During data recording, each visual ERP session was divided into a train and test run. During the train run, trials were not decoded. However, during the test run, the trials were decoded and feedback was given to the participant after each trial [22]. Both runs of the same session are combined to train the diffusion model, as they employ the same copy-spelling task. All runs combined result in 74520 target and 372600 non-target epochs.

The metadata of this dataset contains the subject, session, and class of each trial. These labels all encoded as categorical values. Although sessions could be considered discrete continuous values, we treat them as categorical variables because there are only two sessions, and we do not expect systematic shift between the them.

2.1.2 Auditory ERP

The auditory ERP dataset is focused on modulating the ERP response using a variety of stimulus onset asynchronies (SOAs) [25]. The SOAs were between 60 and 600 ms. Each trial consisted of 90 stimuli presentations, of which 15 are high-pitched target tones and 75 are low-pitched non-target tones. Each of the 13 participants underwent two runs. The first run consisted of 50 to 70 trials, with each trial having a different SOA. The second run consisted of four or five blocks, with five trials per block. The SOA within a block was kept constant. Only the second run is used to train the diffusion model, as the first run is not available through Mother of All BCI Benchmarks (MOABB) at the time of writing [26]. The combined second runs have a total of 15900 target and 79500 non-target epochs.

The metadata of this dataset contains the subject and SOA of each trial. Both subject and SOA are encoded as categorical variables. It could be argued that SOA should be encoded as a continuous variable, however, given that there are often large gaps between two consecutive SOAs, we have chosen to encode it as a categorical variable.

2.1.3 Aphasia

Lastly, the aphasia dataset was recorded using a closed-loop BCI to rehabilitate language in chronic stroke patients that suffer from aphasia [27]. The study used an adapted auditory multi-class spatial ERP (AMUSE) paradigm where each of six bisyllabic words has a different speaker assigned to it [27, 28]. The speakers are evenly spaced on a circle around the participant on ear height. An example of a speaker layout with eight speakers can be seen in figure 3. During a trial, the target word (i.e., one of the six words) was cued by showing a sentence that missed the target word at the end of the sentence [27]. In the study, each word was played 15 times in one trial, unless the early stopping mechanism was triggered. After each trial, both visual and auditory feedback was given on how well the decoder was able to differentiate between target and non-target words. The authors of the study adapted the difficulty of the task by 1) changing the SOA and 2) removing the spatial component by playing

the words via headphones. The data from the headphones condition is not used to train the diffusion model. A total of 10 aphasia patients participated in the study. The amount of sessions and the duration of each session was dependent on the participant, however, the total training time was 30 hours per participant. The combined sessions of all participants equals 159023 target and 795116 non-target epochs.

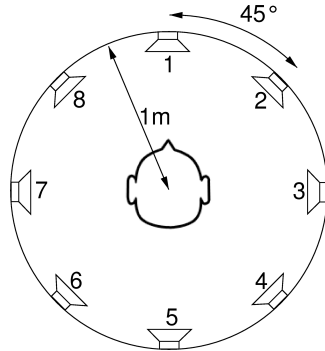


Figure 3: An example of an AMUSE paradigm setup with 8 speakers. Reprinted from Schreuder et al. (2010) [28].

The metadata of this dataset contains a wide variety of variables. However, to reduce the scope of this work, we condition the model on the subject, session, class, and number of feedback trials that the participant has completed so far. The subject, session and class are considered categorical variables, while the number of trials is considered a continuous variable. This encoding scheme is used, as we hope that sessions encoding captures the large non-continuous non-stationarities between sessions, such as electrode position changes, while the number of trials encoding captures the continuous changes in the ERP response as a result of the neurofeedback.

2.1.4 Preprocessing

The visual and auditory ERP datasets are acquired and preprocessed using the MOABB [26], while the aphasia rehabilitation dataset is not available online and is preprocessed using a custom-made script. A relatively simple preprocessing pipeline is used to preprocess the three datasets. Firstly, 19 channels (Fp1, Fp2, F7, F8, F3, F4, Fz, T7, T8, C3, C4, Cz, P7, P8, P3, P4, Pz, O1, and O2) were selected to give full scalp coverage (see figure 4). Secondly, the data was band-pass filtered between 1 and 20 Hz with a 4th-order Butterworth filter. Thirdly, the data was downsampled to 128 Hz. Fourthly, epochs were constructed as 1-second windows, starting from the stimuli onsets. Lastly, peak-to-peak epoch rejection is applied, with the threshold being dependent on the dataset. The thresholds are set to $150\mu\text{V}$, $100\mu\text{V}$, and $150\mu\text{V}$ for the visual ERP, auditory ERP, and aphasia rehabilitation dataset, respectively. This removes 12611 tar-

get and 50074 non-target epochs in the visual ERP dataset, 2153 target and 10444 non-target epochs in the auditory ERP dataset, and 17613 target epochs and 86095 non-target epochs in the aphasia rehabilitation dataset. As a result, 61909 target and 309915 non-target epochs remain in the visual ERP dataset, 13747 target and 66903 non-target epochs in the auditory ERP dataset, and 141410 target and 728213 non-target epochs in the aphasia dataset.

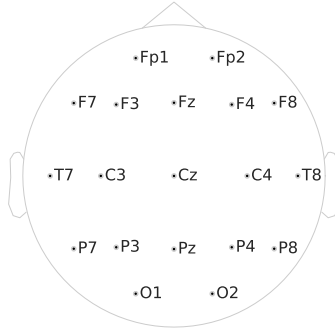


Figure 4: The layout of the selected EEG channels.

2.2 Model

The model to estimate the score is an adaptation of an unconditioned neural network introduced by both Torma et al. and Shu et al. that is able to train on EEG data directly, called EEGWave or diff-EEG, respectively [13, 17]. The layout of this model can be seen in figure 5. This neural network first applies a depthwise convolution with ReLU non-linearity to learn 128 spatial filters. Subsequently, 40 residual layers are applied, with the output of each residual layer being split between 1) a residual connection that is normalized and fed into the next residual layer, and 2) a skip connection that is normalized and put through two pointwise convolutions, with the first one keeping the same number of virtual channels (i.e., 128 given the number of spatial filters) and applying a ReLU activation, and the second reducing the number of virtual channels to the original number of EEG channels.

In each residual layer, the input is summed with the embeddings created for the timestep [13, 17]. Thereafter, a bidirectional convolution with a kernel size of 3 and a dilation cycle of $2^{i \bmod 7}$ with i being the current residual layer is applied. This bidirectional convolution increases the number of virtual channels from 128 to 256, which is subsequently split into two chunks of 128. These chunks are multiplied after each receiving a different activation functions (tahn or sigmoid). Lastly, a pointwise convolution is applied, which doubles the number of virtual channels before the output is split between the skip and residual connection.

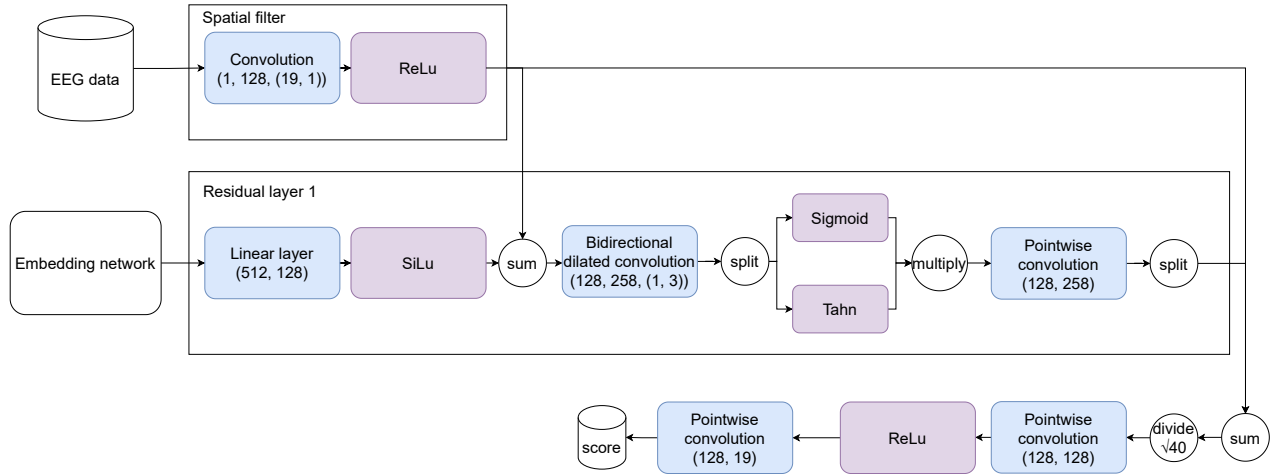


Figure 5: Diagram depicting the first residual layer of the neural network.

2.3 Conditioning

2.3.1 Influence of existing models

There are two differences between EEGWave and diff-EEG that influenced our design choices regarding the embedding architecture. Firstly, EEGWave uses SiLu non-linearity in the timestep embedding neural network, while diff-EEG uses Sigmoid activations [13, 17]. Secondly, both EEGWave and diff-EEG sum the timestep embedding with the data before the bidirectional convolution, however, diff-EEG also adds a spectrogram embedding after bidirectional convolution. Unfortunately, during the implementation of the CFG, we were unaware of the existence of the diff-EEG model, as such, the conditioning in our model done by simply summing the timestep and label embeddings, as prescribed by Luo (2022) [29]. We do not expect a noticeable difference between the two approaches.

Contrary to diff-EEG, which utilizes an embedding based on spectrograms of the EEG data [17], we want to utilize the labels in the metadata to condition the model, which should make it possible to create EEG data that is specific to a certain combination of labels. However, this requires some additional consideration, because certain labels could be drowned out when multiple labels are embedded and subsequently summed. To solve this, we briefly experimented with a concatenation approach, where each label has a dedicated section of the embedding vector, which ensures that the embedding of each label remains intact. However, since no discernible difference was found between the two approach, we decided to use the summing method, as this is the method that is used in the literature [11, 12, 15–18, 29].

2.3.2 Embedding strategies

Now that we have discussed where to add the label and timestep embedding in the model, it is useful to discuss how these label and timestep are embedded. In the EEGWave and diff-EEG architectures, the timesteps are embedded using a continuous positional encoding matrix [13, 17]. This embedding v_{emb}^t of a discrete value t is achieved by a continuous positional encoding matrix:

$$v_{emb}^t = \left[\dots, \sin \left(t \cdot 10^{(i-4)/i_{max}} \right), \cos \left(t \cdot 10^{(i-4)/i_{max}} \right), \dots \right], \quad i \in [0, 63] \quad (5)$$

However, our architecture uses the continuous SMLD framework, rather than the discrete DDPM framework. Therefore, an interpolation is done between the embeddings of the two nearest discrete timesteps. Nonetheless, the original discrete embedding method is still used to embed discrete labels, such as the number of feedback trials in the aphasia dataset.

The categorical variables, such as the class, are encoded using the PyTorch embedding layer, which is a trainable encoding matrix that learns to differentiate between the classes during training. We also briefly experimented with a one-hot encoding vector instead of the PyTorch embedding layer, however, there was no noticeable difference in performance. In hindsight, this is to be expected, as the model can map the one-hot encoding vector to the PyTorch embedding by means of a single linear layer, and similarly, the PyTorch embedding layer can in theory learn a one-hot encoding if that is the optimal encoding strategy.

After these initial embeddings, each label and timestep is put through a small embedding neural network, which can be viewed in figure 6. The output of these embedding neural networks are passed on to each residual layer, where a final linear layer reduces the embeddings from 512-dimensional vectors to 128-dimensional vectors (see figure 5). These vectors are subsequently summed with the residual connection (or the embedded data in the first residual layer) before the bidirectional convolution.

2.3.3 Label dropout

To retrieve the unconditioned score, label dropout is applied in the original implementation of the CFG. However, a preliminary experiment on the visual ERP dataset with $w = 1$ and a label dropout rate of 20% showed that the amplitude of the generated samples were too large. In contrast, only using the conditional score (i.e., $w = 0$) and no label dropout generated samples with a more realistic amplitude. Consequently, the primary experiments are done without label dropout.

Nonetheless, the method of label dropout is described. Additionally, the experiments described in section 5 and section 6 are (partially) repeated with 20% label dropout and $w = 0$ on the visual ERP and auditory ERP datasets to assess the impact of label dropout in isolation. The results are described in section A.5.

In the case of continuous labels, the value is replaced with a random value

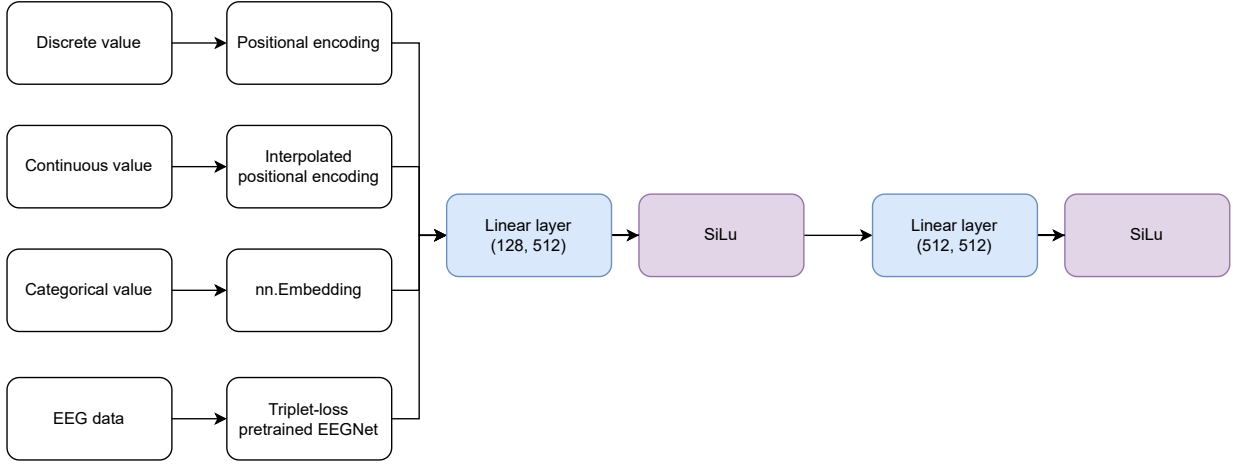


Figure 6: Diagram showing the embedding layers that create the 512-dimensional embedding, which are fed into each residual layer

between the minimum and maximum value within that label. Additionally, the mask of this label is added as input to the conditional model. It should be noted that the timestep is never masked, so this strategy is only used for the number of feedback trials in the aphasia dataset. In the case of categorical labels, one additional class is added, which represents the null-token. Null-tokens are embedded using the same strategy as regular labels.

2.4 Forward process

For all experiments the variance persevering stochastic differential equation (VP SDE) is used as the perturbation kernel. Preliminary results indicated that other SDEs can also be used, however, exploring the impact that different perturbation kernels have on the performance of the model is outside the scope of this work.

The hyperparameters for the VP SDE are set to $\beta_{min} = 0.1$ and $\beta_{max} = 20$ in line with the values used by Ho et al. and Song et al. [11, 30]. However, given that these values are taken from studies conducted in the image domain, where data is usually normalized between $[0,1]$ or $[-1, 1]$, it is not self-evident that these values are sufficient to destroy the non-normalized EEG data. Therefore, a brief experiment was conducted to validate these hyperparameters. The results (found in section A.1) of this experiment indicated that the EEG data is sufficiently destroyed by the VP SDE with the aforementioned parameters.

2.5 Backward process

The backward process is the reverse-time VP SDE discretized in 1000 timesteps, with $t \in [1e^{-5}, 1]$ as $t = 0$ leads to numerical instability [11]. We use PC sampling, with Euler-Maruyama as the predictor and Langevin dynamics as the corrector, with one corrector step per predictor step. Preliminary results indicated that this improved sample quality compared to using the ordinary differential equation (ODE) solver and Predictor-only sampling, which is in line with the findings of Song et al.(2021) [11]. The step size ϵ for the Langevin dynamics is determined dynamically based on the noise, output of the model, and the hyperparameter r [11]. Initial results showed that a r of 0.16 generates samples with a more realistic amplitude, compared to 0.01, and 0.1, hence the experiments in this work are done with $r = 0.16$.

Exponential moving average (EMA) weights are used to estimate the score, which stabilizes the sampling quality by reducing large fluctuations in the weights [31]. The EMA weights θ' are a copy of the weights of the weights that are being trained θ and are updated based on the following update rule:

$$\theta'_{i+1} = \beta\theta'_i + (1 - \beta)\theta_{i+1} \quad (6)$$

with β functioning as a moment parameter, which is set to 0.9999, as is recommended for the VP SDE [11, 31].

2.6 Training

During training, a random 10% of the dataset is used to compute the validation loss, which is used to detect training-related issues and severe overfitting. A gradient clipping threshold of 1.0 is used to prevent excessive gradients, in line with previous research [11, 30]. The batch size is set to 128. The Adam optimizer with a learning rate of $2e^{-4}$ and default betas is used, consistent with the EEGWave implementation [13]. The model is trained for 500k training steps, as previous results indicated that there were no significant improvements when training the model beyond this point on the visual ERP dataset [32]. The trained model¹ is subsequently used to sample ERP EEG data. The sampled EEG data is visualized and metrics are computed to assess the realism.

3 Metrics

Once the model is trained, the samples generated by the model are evaluated using a wide variety of metrics. These metrics can be grouped into four general categories: 1) metrics that come from the application of generative models in the image domain, 2) metrics that are domain invariant, 3) metrics based on classifier performance, 4) metrics that are specific to (ERP) EEG data.

¹Diffusion model checkpoints: <https://huggingface.co/guidance/Diffusion checkpoints/tree/main>

3.1 Image-domain metrics

There are two dominant metrics in the image domain: the IS and the FID. These both rely on activation of a neural network. In the image domain, a standardized pre-trained InceptionV3 network is used [33]. However, so far, the BCI-domain has no standardized neural network to extract these activations. This is likely due to the fact that there is no widespread adoption of one particular dataset, and there is a vast amount of variations possible in the preprocessing pipeline [34]. To address this, we use 1) two datasets that are freely available through MOABB, 2) a straightforward preprocessing pipeline, 3) EEGNet as the feature extractor, as it has very few parameters, making it computationally cheap to extract the activations [35], and 4) Huggingface to share the pre-trained EEGNets. This approach should facilitate a comparison to our findings in future work.

3.1.1 EEGNet

As mentioned previously, a pre-trained EEGNet is used as the feature extractor for the IS and the FID. The EEGNet models are trained to predict the label (target or non-target) of the EEG data. One EEGNet is trained per dataset. The datasets are shuffled, and a random 20% is used as a validation set. They are trained for 1000 epochs with the following hyperparameters: dropout rate of 25%, class-rebalanced weighting of the labels, eight spatial filters, two temporal filters per spatial filter, input window size of 128 (equivalent to sample length of the EEG data), batch size of 128, learning rate of $1e^{-2}$, and the stochastic gradient descent optimizer. The model with the lowest validation loss is used as the final model. The performance of the three pre-trained EEGNet models² can be viewed in table 1.

		Visual ERP	Auditory ERP	Aphasia
	Label ratio	1 : 5.21	1: 5.03	1 : 5.01
	Accuracy	0.862	0.702	0.671
F1-score	Target	0.66	0.42	0.38
	Non-target	0.91	0.80	0.78
Precision	Target	0.55	0.31	0.28
	Non-target	0.96	0.91	0.90
Recall	Target	0.83	0.64	0.61
	Non-target	0.87	0.71	0.68

Table 1: Performance of the trained EEGNet models for each dataset. Label ratio represents the targets to non-target samples after preprocessing, which is used to adjust the loss function.

²EEGNet checkpoints: <https://huggingface.co/gui-do151/EEGNet>

3.1.2 Inception score

Although the IS is one of the most dominant metrics for diffusion models in the image domain, it unfortunately is not applicable to the conditional diffusion models used in this work. The IS is the average Kullback–Leibler (KL) divergence between the conditional class distribution $p(y|x)$ and the marginal class distribution $p(y)$ [36]:

$$\text{IS} = \exp(\mathbb{E}_{\mathbf{x}} KL(p(y|x) || p(y))) \quad (7)$$

This codifies two general principles: the classifier (i.e., EEGNet) should be confident about the content of the generated data, so the entropy of $p(x/y)$ should be low, and there should be a lot of diversity in the labels, so the entropy of $p(y)$ should be high [36]. However, there are a few concerns with using the IS. Firstly, the oddball datasets have an inherent class imbalance, which can bias the classifier in the direction of the majority class, as is the case for the EEGNets. Secondly, the labels of the samples generated by a well-trained conditional diffusion model are user-imposed, which means that the marginal class distribution is user-imposed. Therefore, we do not expect that the IS will be informative about the quality of the samples. Hence, the IS is excluded from the analysis.

3.1.3 Fréchet Inception distance

Fortunately, the other dominant metric in the image domain, the FID, does not suffer from the same problems. The FID computes the Fréchet distance between two Gaussians [33]. One Gaussian is fitted to the mean $\boldsymbol{\mu}_r$ and standard deviation $\boldsymbol{\Sigma}_r$ of the activations in response to real data, while the other is fitted to the mean $\boldsymbol{\mu}_g$ and standard deviation $\boldsymbol{\Sigma}_g$ of the activations in response to generated data [33]. The activations are extracted from the final pooling layer of the trained EEGNet, in line with previous work [13].

$$\text{FID} = \|\boldsymbol{\mu}_g - \boldsymbol{\mu}_r\|_2^2 + \text{Tr}(\boldsymbol{\Sigma}_g + \boldsymbol{\Sigma}_r - 2(\boldsymbol{\Sigma}_g \boldsymbol{\Sigma}_r)^{\frac{1}{2}}) \quad (8)$$

The FID codifies the principle that the trained neural network should not be able to differentiate between real and generated data. However, because EEGNet is trained to be invariant to variables other than label, it might not be able to capture important features in the data, which could artificially lower the FID. Moreover, different input can lead to similar activations, which makes it almost impossible to trace which features have an impact on the FID and which ones are neglected. Furthermore, the FID requires 10k samples to be accurate [33], which makes it impossible to compare within a specific label combination, as there is not enough data available. To address these challenges, domain-specific metrics are introduced, which cover a wide variety of domain-specific features, allowing for a more in-depth evaluation of the diffusion model.

3.2 Classifier performance

Classifier improvements for generated data are often calculated using an augmentation approach, where the difference in accuracy between an augmented and a non-augmented training dataset is reported [16–18, 20, 37]. However, this makes it impossible to disentangle the effects of the increase in training volume and the increase in variety in the dataset. Additionally, the improvement in overall classification accuracy could be detrimental to the classification accuracy of a specific label combination. Therefore, instead of using the data augmentation approach, we decided to train a within-subject within-session/SOA classifier on synthetic data and subsequently test on real data. This approach is similar to Sharma et al. (2023) who also opted to also train on synthetic data and test on real data [15].

Specifically, two classifiers are trained using five-fold stratified cross-validation. The classifiers are a regularized least-squares linear discriminant analysis (LDA), with the shrinkage coefficient determined by the Ledoit-Wolf formula [38]. Both are trained using the average amplitude across channels in non-overlapping time windows [2, 22]. These time windows span between 0.1 and 0.9 seconds, and are each 0.1 second long. For the visual ERP and aphasia dataset, a within-subject within-session classifier is used, while for the auditory ERP dataset, a within-subject within-SOA classifier is used. One LDA is trained on the real training data in the fold and is used as the baseline. The other LDA is trained on an equal amount of data generated with equivalent label combinations. Both models are then tested on the left-out data from the fold. This approach ensures that the test set is the same for both classifiers in each fold, while still allowing within-subject within-session/within-SOA training. The average balanced accuracy (ABA) over the five real data test sets is reported. ABA has a chance level of 0.5, despite the class imbalance.

3.3 Domain-invariant metrics

Domain-invariant metrics is an umbrella term for a wide variety of distance, similarity, and error metrics that can be applied to virtually any domain. A plethora of different domain-invariant metrics are used to compare real and generated EEG data [37]. However, the decision to use one domain-invariant metrics instead of another is rather arbitrary, as they all measure a combination of features in the EEG data. Therefore, we have arbitrarily chosen to use the SWD, the Euclidean distance (ED), and the mean squared error (MSE).

Let us index the real EEG data in trial $i \in I$ at channel $c \in C$ and time point $t \in T$ as \mathbf{x}_r^{ict} and the generated EEG data as \mathbf{x}_g^{ict} . If the data is averaged over trials, such that it becomes evoked EEG data, the trial index is dropped, as the dimensionality of the data is reduced.

3.3.1 Sliced Wasserstein distance

The Wasserstein distance, also known as the Earth Mover’s Distance, is the minimal amount of effort necessary to transform one distribution into another distri-

bution [39]. However, calculating the Wasserstein distance on high-dimensional data is computationally costly [40, 41]. The Sliced Wasserstein distance overcomes this issue by approximating the Wasserstein distance by computing the average Wasserstein distance over multiple random 1-D projections [40]. In our implementation, the SWD is calculated between the flattened real and generated EEG data using 10k projections in the visual and auditory ERP datasets and using 1k projections for the aphasia dataset. The reduction in the number of projections was necessary to reduce the computation burden, considering the large size of the aphasia dataset.

3.3.2 Euclidean distance

The ED, also referred to as the L2-norm, is the length of the shortest line between two points in Euclidean space. The average ED is computed over channels, by summing the ED between the real and generated evoked data at each channel and subsequently dividing by the total number of channels:

$$\text{ED} = \frac{1}{C} \sum_{j=1}^C \|\mathbf{x}_r^c - \mathbf{x}_g^c\|_2 \quad (9)$$

3.3.3 Mean squared error

The MSE, computes the averaged squared differences between the real and predicted values. In this work, the MSE computes the average squared differences between the real and generated evoked EEG data at each time point:

$$\text{MSE} = \frac{1}{C} \sum_{c=1}^C \frac{1}{T} \sum_{t=1}^T (\mathbf{x}_r^{ct} - \mathbf{x}_g^{ct})^2 \quad (10)$$

3.4 Domain-specific metrics

The domain-specific metrics are introduced as each of them is able to capture one specific feature of interest in the EEG data, instead of relying on a combination of features. This makes it possible to select a model that optimizes that particular feature, and also gives a more in-depth overview of the capabilities and pitfalls of the model. The domain-specific metrics are split into three groups. Firstly, evoked metrics exploit the fact that ERP EEG data is both phase- and time-locked, by averaging the responses, which increase the SNR [2]. Secondly, the PSD metrics use the power spectrum of the EEG data. Lastly, the standard deviation metrics use the standard deviation over all the trails, which gives an indication about the diversity in the EEG data.

3.4.1 Evoked metrics

Firstly, the time lag (TL) is used to assess the ability of the model to generate data that has a similar progression to the real data. The cross-correlation between the real and generated evoked data is computed for each channel. This

measures the similarity between the real and generated evoked data as they are time-shifted by τ relative to each other. The TL is the average time in ms across channels that yields the highest cross-correlation:

$$TL = \frac{1}{C} \sum_C \frac{\operatorname{argmax}_{\tau} \left(\sum_{t=1}^T \mathbf{x}_r^{c(t+\tau)} \mathbf{x}_g^{ct} \right)}{f} \quad (11)$$

with f being the sampling frequency, and \mathbf{x}_g and \mathbf{x}_r are zero padded where necessary.

Secondly, the P300 peak position and amplitude are assessed by the peak latency delta (PLD) and peak amplitude delta (PAD) metrics, respectively. These metrics only consider the channel with the most prominent P300 peak when averaged over all the real target data, which is the channel ‘‘O1’’ in the case of the visual ERP dataset and ‘‘Cz’’ in both the auditory ERP and aphasia datasets. The PAD is computed by taking the absolute difference in μV between the highest peak in the real and generated data at the selected channel:

$$PAD = \left| \max_t \mathbf{x}_g^{ct} - \max_t \mathbf{x}_r^{ct} \right|, \quad \text{with } c \text{ being pre-selected} \quad (12)$$

The PLD is the absolute difference in time offset between these peaks, measured in ms:

$$PLD = \left| \operatorname{argmax}_t \mathbf{x}_g^{ct} - \operatorname{argmax}_t \mathbf{x}_r^{ct} \right|, \quad \text{with } c \text{ being pre-selected} \quad (13)$$

Only target trials are included, as a P300 peak is necessary to compute these metrics.

Thirdly, the amplitude of the generated data is assessed by two different metrics. The delta peak-to-peak (D-PTP) measures whether the range of amplitude is correct. This is computed by taking the average absolute difference over channels between the highest point and the lowest point between the real and generated evoked data:

$$D\text{-PTP} = \frac{1}{C} \sum_C \left| \left(\max_t \mathbf{x}_g^{ct} - \min_t \mathbf{x}_g^{ct} \right) - \left(\max_t \mathbf{x}_r^{ct} - \min_t \mathbf{x}_r^{ct} \right) \right| \quad (14)$$

The average power delta (APD) measures whether the average power is similar between the real and generated data. This involves converting the evoked signal into the power, which is done by squaring. Subsequently, the average power per channel is computed. The mean over the absolute difference between the average power of the real and generated data per channel is computed:

$$APD = \frac{1}{C} \sum_C \left| \frac{1}{T} \sum_{t=1}^T (\mathbf{x}_g^{ct})^2 - \frac{1}{T} \sum_{t=1}^T (\mathbf{x}_r^{ct})^2 \right| \quad (15)$$

Lastly, the similarity of the cross-channel interactions, which are represented as a spatial covariance matrix (SCM) [42], is assessed by the spatial covariance

matrix Riemmanian distance (SCM-RD). First, the SCMs of both the real and generated evoked data are computed. Subsequently, the Riemannian distance between the two SCMs is computed:

$$\text{SCM-RD} = \delta_{\text{Riemannian}}\left(\frac{1}{T}\mathbf{x}_r(\mathbf{x}_r)^\top, \frac{1}{T}\mathbf{x}_g(\mathbf{x}_g)^\top\right) \quad (16)$$

Riemannian geometry is used, because it takes into account the curvature of the manifold on which symmetric positive definite matrices, such as SCMs, lie [42].

3.4.2 Power spectral density metrics

The power spectral density metrics are used to assess how well the frequencies are captured by the model. This is split into two metrics. The first metric, power spectral density delta (PSD-D) is used to check whether the generated evoked data has a similar distribution of power across frequency bands as the real evoked data. Firstly, the power-spectral densities of both the real and generated (non-averaged) data are computed. Secondly, only the densities within the bandpass filter are kept, in this case 1 and 20 Hz. Lastly, the Manhattan distance between the mean of both the real and generated densities at each timestep are computed. The other metric, power spectral density standard deviation delta (PSD-STD-D), is described in the next section, as it utilizes the standard deviation of the spectral densities.

3.4.3 Standard deviation metrics

One of the main downsides of using evoked EEG data is that information about the diversity of samples is lost. Nonetheless, it is important that the model is able to generate a variety of EEG data. The final two metrics address this concern by evaluating the difference in standard deviations between the real and generated EEG data. The first metric, the standard deviation Manhattan distance (STD-MD), computes the average Manhattan distance over channels between the standard deviation of the real and generated data:

$$\text{STD-MD} = \frac{1}{C} \sum^C k\sigma_{x_g^c} \quad \sigma_{x_r^c} k_1 \quad (17)$$

The second metric, the PSD-STD-D follows the first two steps of PSD-D, but instead of using the mean, the Manhattan distance between the standard deviation of the spectral-densities of both the real and generated data at each timestep is computed. This gives an indication of the difference in variation in the EEG data at each frequency band.

4 Visual inspection

Besides the quantitative approach to model evaluation described in the previous section, we will also use a qualitative approach in the form of a visual inspection

of the generated EEG data in comparison to the real EEG data. This visualization of the generated data might make it more intuitive to see whether the model can be trusted, which can be valuable in a clinical setting. Although only the evoked response plot and spatial covariance matrix plot are used in the main work, the other plots are also explained for completeness. All generated plots can be found in the GitHub repository ³.

4.1 Evoked response plot

The evoked response plot shows the evoked response at three different channels (Cz, Pz, O1). This can be used to evaluate how well the model is able to recreate the average responses at these channels. Additionally, the standard deviation is plotted, which gives an indication of how diverse the underlying data is.

4.2 Spatial covariance matrix plot

The SCM plot shows the SCMs of the real and generated data, split on the label. This is informative about the extent to which the model is able to generate data that has cross-channel interactions that are similar to the real data.

4.3 Power spectral density plot

The PSD plot shows the average PSD across channels for the target and non-target data. Additionally, it also shows the standard deviation across trials. This can be used to evaluate to what degree the model is able to generate data that has a similar power across all frequencies compared to the real data.

4.4 Topographic map

The topographic maps plot the estimated brain activity, based on the evoked EEG data. This plot is used to evaluate to what extent the model is able to generate data that can have similar underlying brain activity as the real data.

5 Experiment 1: within-dataset generation

The first experiment addresses **research question 3**:

- Given the design choices made as a result of the first research question, it is possible to generate ERP EEG data that is specific to a particular combination of labels in the training set that is similar to the real data of that same combination, as measured by the metrics that are implemented as a result of the second research question?

³GitHub repository: https://gitlab.socsci.ru.nl/neurotech/code/thesis_gui_dokleijn/-/tree/main/figures/Sampled

To this end, the previously described diffusion model is trained separately on each dataset. During training, the model has access to all label combinations present in the dataset. Subsequently, data for each label combination that was present during training is sampled.

Ideally, we would establish a baseline for each metric that gives the performance of the real data in each label combination that is present in the dataset. This is previously referred to as a within-subject within-session/SOA baseline. This would allow us to test whether the model is capable of generating EEG ERP data that is similar to the real data on each combination of labels. Unfortunately, this is impossible for most metrics, as 1) the FID requires more samples than are present in one combination of labels to be accurate [33], and 2) computing the ERP requires many samples to improve the SNR, while computing the within-subject within-session/SOA baseline requires splitting the data in half. The classifier performance metric (ABA) is the only metric for which establishing this baseline is straightforward, as cross-validation can be used to train and test on data from one label combination, as explained in section 3.2.

However, the ABA metric only measures label-relevant features in the data, hence it is vital to use it in conjunction with other metrics. Therefore, we also establish baselines on the other metrics. Firstly, the baseline for the FID is established in two ways:

1. The dataset is split into two random halves and the FID between the two is computed.
2. The subjects of the dataset are randomly split into two equal (or close to equal) groups, and the FID between the groups is computed.

Each split is repeated five times and the average is reported.

Secondly, both the domain-invariant and domain-specific metrics use within-subject between-session/SOA variability as a baseline. This measures the minimum variability between two label combinations within a dataset. The baselines are achieved by computing the metrics between two random sessions or between two random SOAs of one participant. This is repeated for each participant. This process is repeated five times if there are multiple sessions or SOAs that can be selected, and the average over the five splits is reported.

5.1 Results: visual ERP dataset

The sampled EEG data is able to outperform the between-subject baseline for the image domain metric (FID), but is much higher than the random-split baseline (table 2). Furthermore, the sampled data outperforms the within-subject between-session baselines on the domain-invariant metrics and the domain-specific metrics, with the only exception being the PSD-STD-D (table 4 and table 5). Moreover, the sampled EEG data achieves the same performance to the within-session baseline on the classifier performance metric (ABA) (table 3). Splitting the results on subjects and sessions, it is clear that the model is very similar to the performance of the original data across all combinations (figure 7).

The largest decrease in ABA is observed for subject 10 in session 2, with a decrease of 0.030. The evoked response and SCM plots for this combination can be viewed in figure 8.

Overall, these results indicate that the model is able to generate data that is subject, session, and label specific. Nonetheless, there is room for improvement regarding the variability in the frequency bands.

	Baseline - Random split	Baseline - Subject split	Sampled
FID #	$5.713e^{-4}$	$1.231e^{-1}$	$1.239e^{-2}$

Table 2: FID for the visual ERP dataset, comparing sampled EEG data with baselines using random or subject splits.

	Baseline	Sampled
ABA "	0.813	0.819

Table 3: Comparison of the average ABA over all subjects and session when trained on real data (baseline) versus sampled data from the visual ERP dataset.

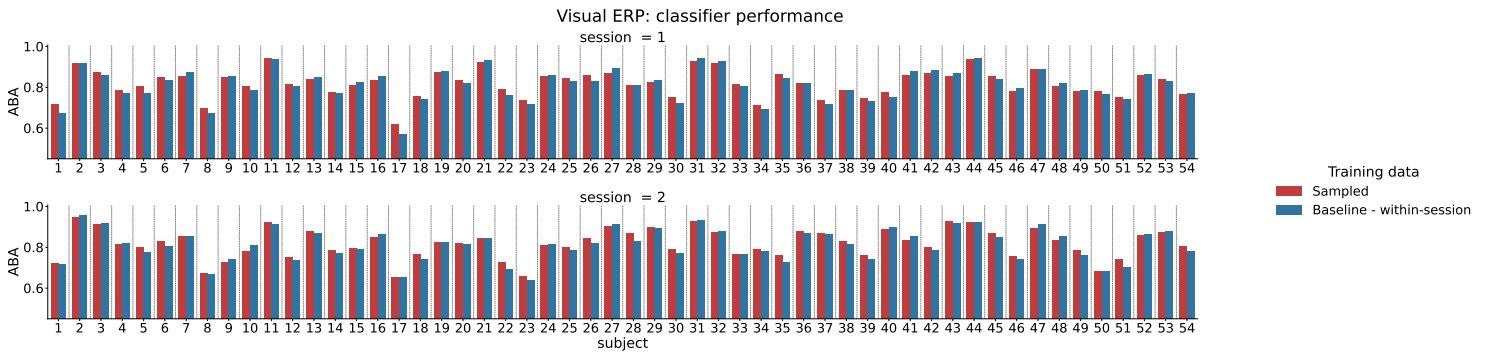


Figure 7: Comparison of the ABA for each subject and session when trained on real data versus sampled data from the visual ERP dataset.

	Target		Non-target	
	Baseline	Sampled	Baseline	Sampled
ED #	9.628	6.516	5.004	3.517
MSE #	1.158	0.416	0.393	0.164
SWD #	1.666	1.371	1.189	1.053

Table 4: Comparison of the average score over subjects and sessions for domain-invariant metrics, comparing the between-sessions baseline and sampled data on the visual ERP dataset.

	Target		Non-target	
	Baseline	Sampled	Baseline	Sampled
PAD #	0.791	0.650	-	-
PLD #	0.046	0.027	-	-
APD #	0.697	0.451	0.333	0.240
D-PTP #	2.106	1.380	1.124	0.885
TL #	0.028	0.009	0.023	0.007
SCM-RD #	8.751	6.113	11.652	8.012
PSD-D #	16.553	11.596	14.681	9.311
STD-MD #	7.994	4.690	2.671	1.813
PSD-STD-D #	1.574	1.648	1.373	1.413

Table 5: Comparison of the average score over subjects and sessions for domain-specific metrics between the between-session baseline and sampled data on the visual ERP dataset.

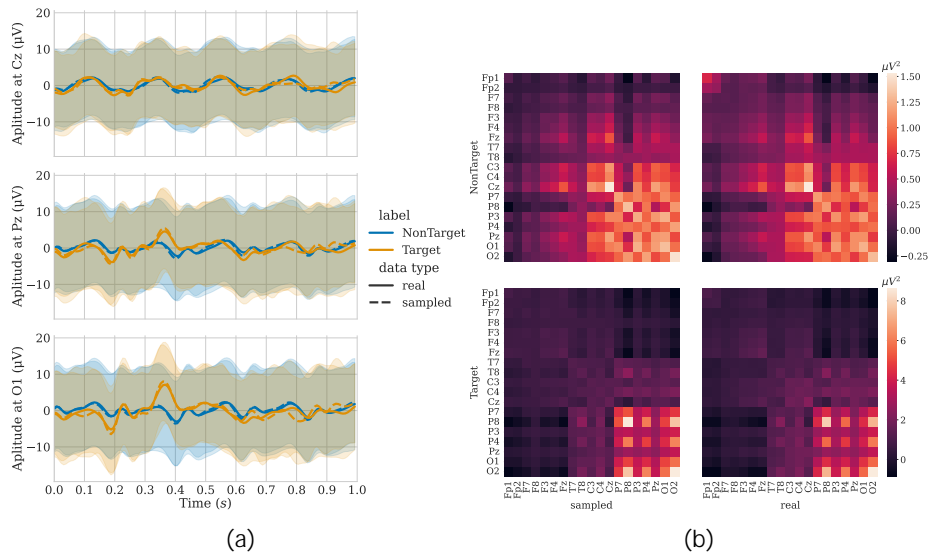


Figure 8: Figures (a) and (b) provide two different comparisons between real and generated data. Figure a shows the real and generated average temporal ERP responses of target and non-target for three selected EEG channels, with the error bands indicating the standard deviations of the data. Figure b shows the SCMs of averaged real (left) and generated (right) responses for both non-target (top) and target (bottom) responses. The figures are based on the averages of EEG data (616 target and 3137 non-target examples), which are sampled for the combination of subject and session combination that resulted in the worst ABA metric (subject 10, session 2) on the visual ERP dataset. The generated target ERP achieved scores of 0.008 on the PLD, 0.900 on the PAD, and 6.150 on the SCM-RD. For the non-target class, the generated ERP achieved a score of 8.477 on the SCM-RD.

5.2 Results: auditory ERP dataset

The sampled EEG data outperforms the between-SOA baseline on the image domain metric (FID), but does not surpass the random-split baseline (table 6). Furthermore, there is a slight increase in average classifier performance (ABA) when training on sampled data compared to training on real data (table 7). Moreover, the results on the ABA are highly similar between the real and sampled data across all subjects and SOAs (figure 9). The highest decrease in ABA is 0.023, which can be observed for subject 1 with a SOA of 60 ms. The evoked response and SCM plots for this combination can be viewed in figure 10. It should be noted that the within-SOA baseline ABA is rather poor at 0.623, thus there is no clear target and non-target ERP in the real data. This also means that the PAD and PLD are unlikely to measure P300-related features, instead they measure the maximum amplitude and the latency of this maximum amplitude in the pre-selected channel (Cz).

The result on the domain-invariant metrics show that sampled EEG data outperforms the within-subject between-SOA baseline on all metrics (table 8). Similarly, the domain-specific metrics indicate that all EEG-related features are generated with relative high accuracy, as the sampled data outperforms the within-subject between-SOA baseline on all metrics with the only exception being the PSD-STD-D (table 9).

	Baseline - Random split	Baseline - SOA split	Sampled
FID #	$1.112e^{-3}$	$7.486e^{-2}$	$2.521e^{-2}$

Table 6: FID for the auditory ERP dataset, comparing sampled EEG data with baselines using random or SOA splits.

	Baseline	Sampled
ABA "	0.623	0.641

Table 7: Comparison of the average ABA over all subjects and SOAs when trained on real data (baseline) versus sampled data from the auditory ERP dataset.

	Target		Non-target	
	Baseline	Sampled	Baseline	Sampled
ED #	13.121	8.107	8.285	4.250
MSE #	1.494	0.667	0.635	0.220
SWD #	1.789	1.512	1.135	0.936

Table 8: Comparison of the average score over subjects and SOAs for domain-invariant metrics between the within-subject between-SOA baseline and sampled data on the auditory ERP dataset.

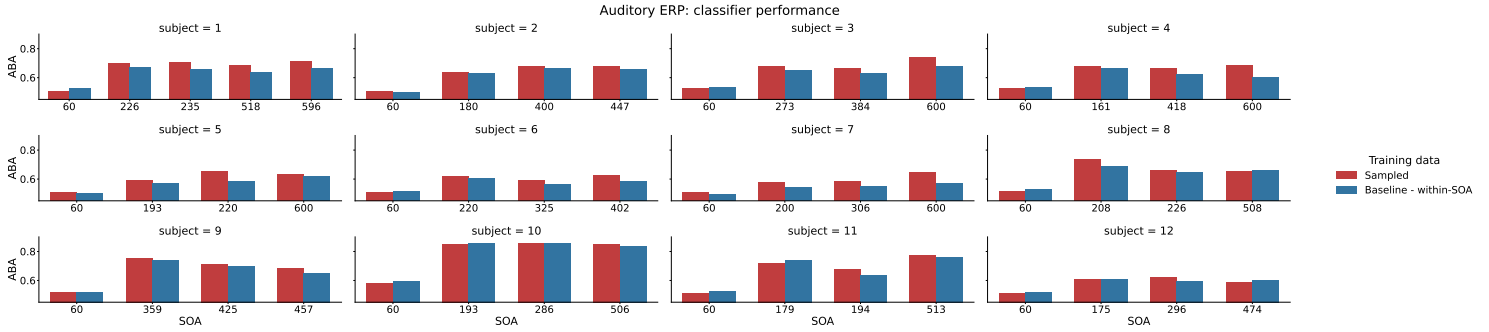


Figure 9: Comparison of the ABA per subject and SOA combination when trained on real data or sampled data of the auditory ERP dataset.

	Target	Non-target		
	Baseline	Sampled	Baseline	Sampled
PAD #	1.502	0.461	-	-
PLD #	0.126	0.109	-	-
APD #	1.068	0.538	0.537	0.284
D-PTP #	3.124	1.069	2.396	0.942
TL #	0.123	0.041	0.156	0.060
SCM-RD #	7.356	6.090	10.833	7.779
PSD-D #	11.675	11.055	9.402	8.351
STD-MD #	7.658	6.752	2.738	2.319
PSD-STD-D #	0.957	1.079	0.763	0.921

Table 9: Comparison of the average score over subjects and SOAs for domain-specific metrics between the between-SOA baseline and sampled data on the auditory ERP dataset.

(a)

(b)

Figure 10: Figures (a) and (b) provide two different comparisons between real and generated data. Figure a shows the real and generated average temporal ERP responses of target and non-target for three selected EEG channels, with the error bands indicating the standard deviations of the data. Figure b shows the SCMs of averaged real (left) and generated (right) responses for both non-target (top) and target (bottom) responses. The figures are based on the averages of EEG data (295 target and 1467 non-target examples), which are sampled for the combination of subject and SOA combination that resulted in the worst ABA metric (subject 1, SOA 60) on the auditory ERP dataset. The generated target ERP achieved scores of 0.289 on the PLD, 0.141 on the PAD, and 5.911 on the SCM-RD. For the non-target class, the generated ERP achieved a score of 8.601 on the SCM-RD.

5.3 Results: aphasia dataset

The sampled EEG data outperforms the subject-split baseline on the image domain metric (FID), but is not able to surpass the random-split baseline (table 10). Furthermore, there is a slight decrease in average classifier performance (ABA) when training on sampled data, compared to training on the real data (table 11). The results of the sampled data are quite similar to the real data across all subjects and sessions, with a few exceptions (see figure 11). The largest decrease is observed for subject 5 in session 10, with a decrease of 0.089. The evoked response and SCM plots for this combination can be viewed in figure 12.

For the domain-invariant metrics, the sampled target EEG data outperforms the between-session baseline on the ED and MSE, but is not able to surpass the baseline on the SWD. The sampled non-target data narrowly outperform the between-session baseline on the MSE but does not exceed the baseline on the ED and SWD (table 12).

Regarding the domain-specific metrics, the amplitude of the sampled data is relatively similar to the real data, as measured by the PAD, APD and D-PTP. The only exception is the non-target sampled data on the D-PTP, which suggests an unusually large difference between the peak and valley of the sampled data compared to the real data. The temporal progression of the sampled data is close to the real data for the target data but relatively dissimilar for the non-target data. Furthermore, the cross channel interactions are well-modelled, with both the target and non-target sampled data beating the between-session baseline on the SCM-RD. However, the PSD and its variability, as measured by the PSD-D and PSD-STD-D metrics, are not well captured, with only the non-target data outperforming the baseline on the PSD-D. Lastly, the diversity of the samples seems to be relatively close to the real data, as sampled data surpasses the between-session baseline on the STD-MD (table 13).

	Baseline - Random split	Baseline - Subject split	Sampled
FID #	$7.972e^{-4}$	7.356	$2.773e^{-1}$

Table 10: FID for the aphasia ERP dataset, comparing sampled EEG data with baselines using random or subject splits.

	Baseline	Sampled
ABA "	0.606	0.595

Table 11: Comparison of the average ABA over all subjects and session when trained on real data (baseline) versus sampled data from the aphasia ERP dataset.

Aphasia: classifier performance

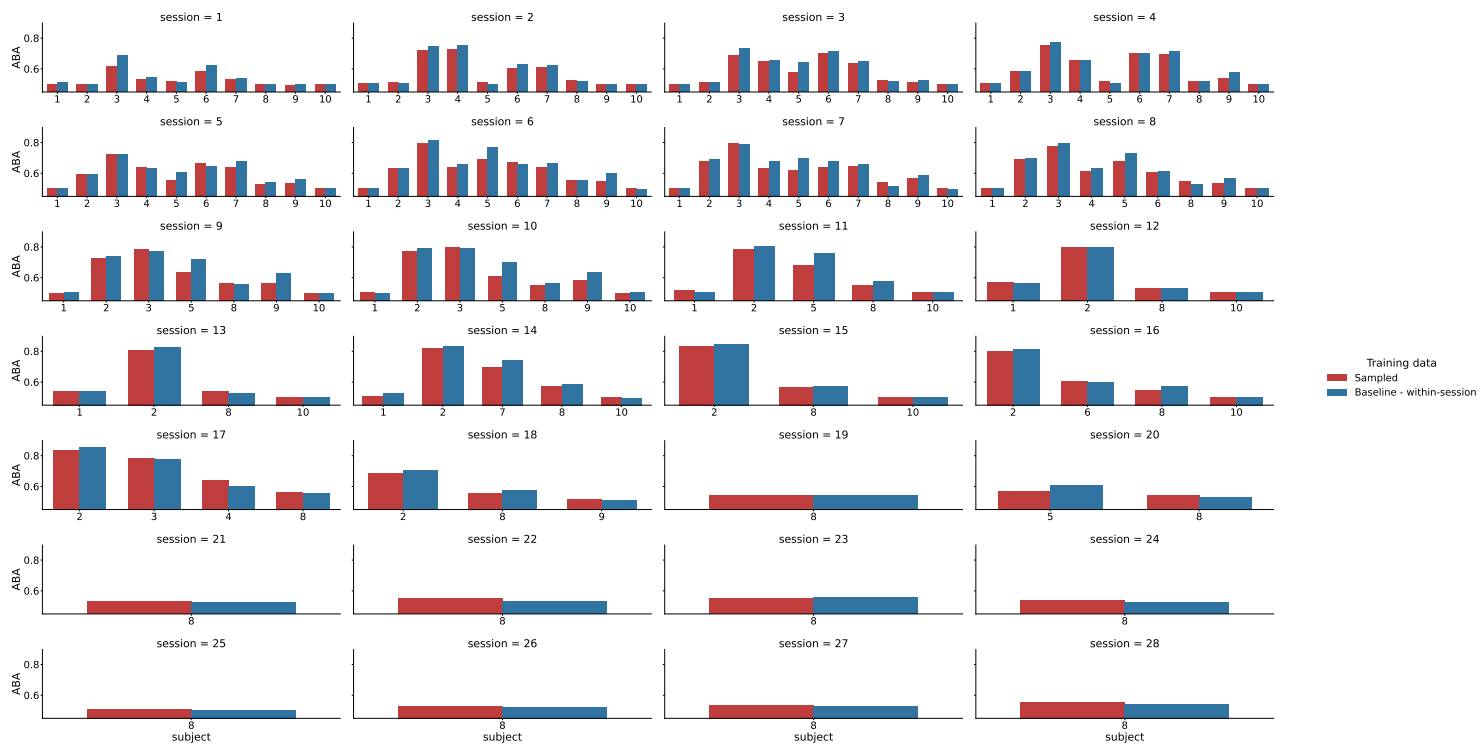


Figure 11: Comparison of the ABA for each subject and session when trained on real data versus sampled data from the aphasia ERP dataset.

	Target		Non-target	
	Baseline	Sampled	Baseline	Sampled
ED #	9.464	8.819	4.818	4.916
MSE #	1.263	0.933	0.277	0.272
SWD #	1.873	2.087	1.256	1.702

Table 12: Comparison of the average score over subjects and sessions for domain-invariant metrics, comparing the between-sessions baseline and sampled data on the aphasia ERP dataset.

	Target		Non-target	
	Baseline	Sampled	Baseline	Sampled
PAD #	0.817	0.704	-	-
PLD #	0.156	0.141	-	-
APD #	1.071	0.986	0.376	0.342
D-PTP #	1.564	1.448	0.367	0.903
TL #	0.080	0.071	0.019	0.050
SCM-RD #	8.972	7.837	10.372	9.452
PSD-D #	20.810	22.606	19.802	18.163
STD-MD #	12.883	6.787	5.044	2.763
PSD-STD-D #	1.882	2.377	1.649	2.269

Table 13: Comparison of the average score over subjects and sessions for domain-specific metrics between the between-session baseline and sampled data on the aphasia ERP dataset.

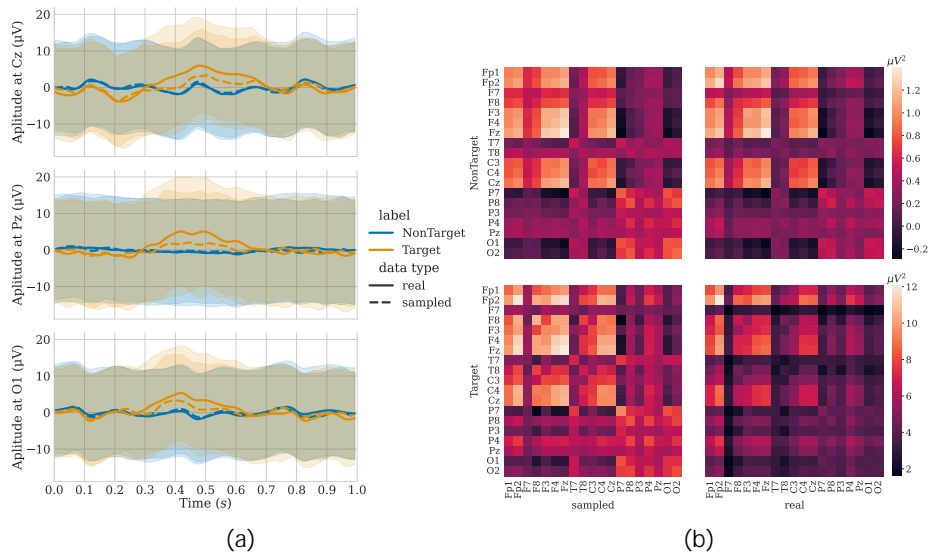


Figure 12: Figures (a) and (b) provide two different comparisons between real and generated data. Figure a shows the real and generated average temporal ERP responses of target and non-target for three selected EEG channels, with the error bands indicating the standard deviations of the data. Figure b shows the SCMs of averaged real (left) and generated (right) responses for both non-target (top) and target (bottom) responses. The figures are based on the averages of EEG data (1570 target and 7909 non-target examples), which are sampled for the combination of subject and session combination that resulted in the worst ABA metric (subject 5, session 10) on the aphasia dataset. The generated target ERP achieved scores of 0.023 on the PLD, 2.695 on the PAD, and 7.695 on the SCM-RD. For the non-target class, the generated ERP achieved a score of 9.982 on the SCM-RD.

6 Experiment 2: outside-dataset generation

The second experiment addresses **research question 4**:

- To what extent is it possible to generate data ERP EEG data for a specific combination of labels, which is absent from the training set, that is similar to the real data of the same combination, as measured by the metrics that are implemented as a result of the second research question?

To answer this question, we test a variety of transfer learning scenarios with differing levels of complexity. The results that are presented in the main text are achieved on the visual ERP dataset, while the results on the auditory ERP and aphasia datasets are relegated to the appendix.

Ideally, we would exclude the label combination for one subject, train the diffusion model and then sample the excluded combination. This process would be repeated for each subject, as is typical in leave-one out validation. However, this is not feasible, because the training of the diffusion model and the subsequent sampling is very computationally costly. Therefore, we have decided to test the transfer learning scenarios by leaving out data of N-subjects per dataset, where N is equal to 25 % of the total number of participants rounded down. The selected participants are kept constant between transfer learning scenarios, such that a change in performance is not due to a change in participants. Unfortunately, this does mean that 1) the results presented here might not generalize to the other subjects in the dataset, and 2) there is less training data available compared to leave-one-out validation, which could decrease the performance of the model. Subjects (2, 4, 6, 7, 8, 16, 24, 26, 33, 39, 40, 41, 53) were randomly selected as the leave N-out subjects for the visual ERP dataset.

6.1 Transfer learning scenarios

Data of later sessions is always predicted based on earlier sessions, as this has a higher ecological validity than predicting earlier sessions based on later sessions. This means that for the visual ERP dataset, the second (i.e., last) session is removed to do the transfer learning scenarios that require removal of data from only one session.

6.1.1 Within-session, between-class transfer learning

The first scenario involves predicting one class of a particular session, based on 1) the other class of the same session of the selected subjects, 2) both classes of previous session(s) of the selected subjects, and 3) all sessions and classes of all other subjects. There is information available about all sources of large non-stationarities (e.g., shifts in electrode placement) and the subject-specific characteristics of the to-be-predicted class in the training data, which makes this a relatively simple scenario.

6.1.2 Between-class transfer learning

The second scenario involves predicting one class for all sessions, based on 1) the other class of each session of the selected subjects, 2) all sessions and classes of all other subjects. This scenario should be slightly more difficult compared to the within-session between-class transfer learning, as the model does not have access to the subject-specific characteristics of the to-be-predicted class.

6.1.3 Between-session transfer learning

The third and final scenario involves predicting a new session based 1) on all previous session(s) of the selected subjects and 2) all sessions and classes of all other subjects. This scenario requires the model to estimate large sources of non-stationarities that happen between sessions. It is unlikely that the model is able to estimate these non-stationarities, as the sources are often session-dependent. For example, it is unlikely that the shift in electrode placement can be estimated based on the shift in electrode placement that occurred between different sessions.

6.2 Results

The results on the ABA metric indicate that the model can predict both target and non-target data in both the within-session between-class transfer scenario and in the between-class transfer learning scenario. Interestingly, in the within-session transfer learning scenario, the ABA of the predicted non-target data in combination with regularly sampled target data outperforms the predicted target data in combination with regularly sampled non-target data, while in the between-class transfer learning scenario, the opposite is true (figure 13).

On the domain-invariant metrics, there is a clear trend that the scores obtained by the predicted class are worse than the scores obtained by the non-predicted class. Moreover, the scores obtained by the non-predicted class in the transfer learning scenarios improve on the results on the non-transfer learning model. The only exception to this is the score obtained on the SWD by the predicted target data, where both the within-session between-class and between-class transfer learning scenarios improve on the result of the non-transfer learning model (figure 15).

In the amplitude-related metrics (PAD, APD, and D-PTP) and the SCM-related metric (SCM-RD), a similar trend to the domain-invariant metrics is observed. Interestingly, the peak latency, as measured by the PLD, of the predicted target data very dissimilar to the real data in the within-session between-class learning scenario, but is rather similar in the between-class transfer learning. The diversity related metrics (STD-MD and PSD-STD-D) indicate that the models where the non-target data is removed generate samples that are closer to the diversity found in the real data compared to all other models.

Regarding the between-session transfer learning scenario, the performance is notably lower on the ABA, with a relatively large decrease in performance

compared to all other scenarios (figure 13). By splitting the data on the multiple subjects and sessions, it can be observed that the between-session transfer learning performs reasonably well for certain subjects, while being close to chance level for other subjects (figure 14). Almost all other metrics corroborate the finding that the between-session transfer learning performs by far the worst out of all transfer learning scenarios.

An example of the change in ERP response across the various transfer learning scenarios on one subject and session can be viewed in figure 18.

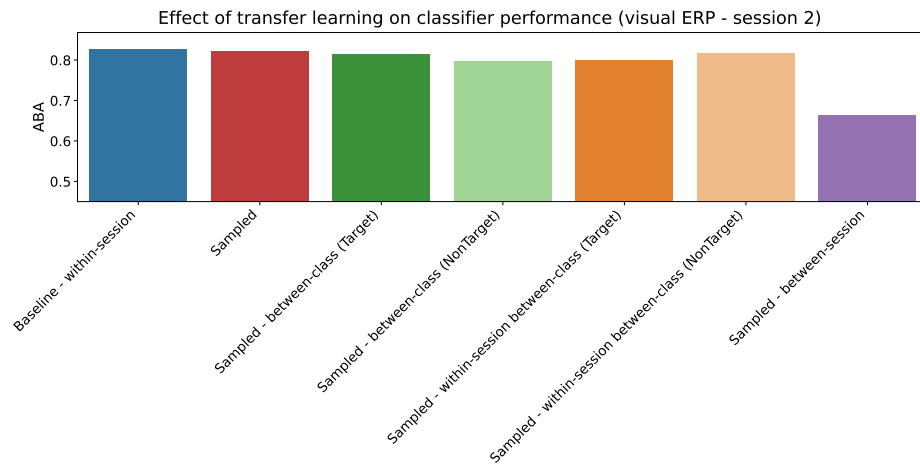


Figure 13: Effect of transfer learning on the average classifier performance across subjects on the second session of the visual ERP dataset. The predicted class that is indicated in brackets, with both classes predicted in the between-session transfer learning scenario.

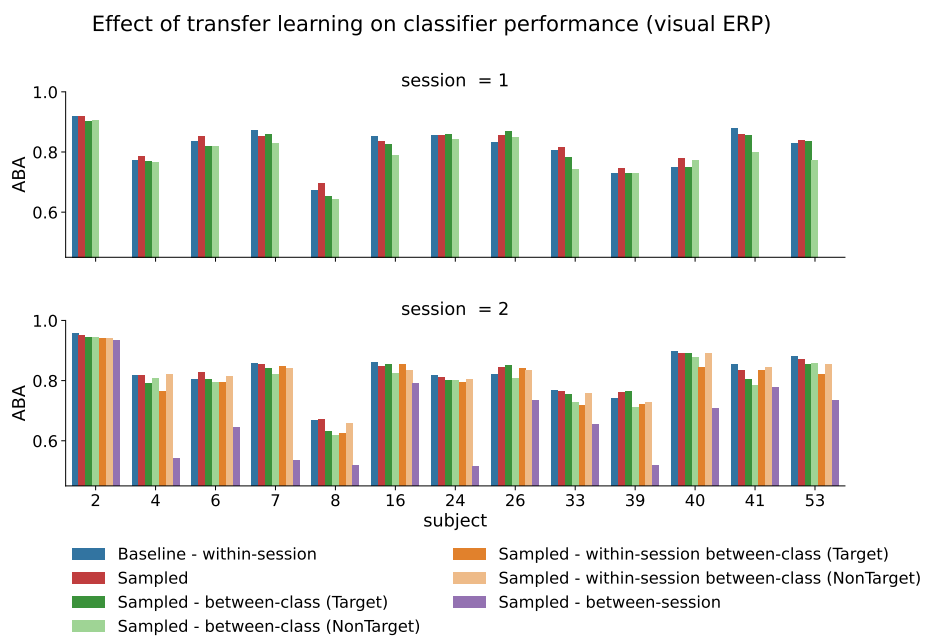


Figure 14: Effect of transfer learning on classifier performance per subject and session of the visual ERP dataset. The predicted class that is indicated in brackets, with both classes predicted in the between-session transfer learning scenario.

Effect of transfer learning on domain-invariant metrics (visual ERP - session 2)

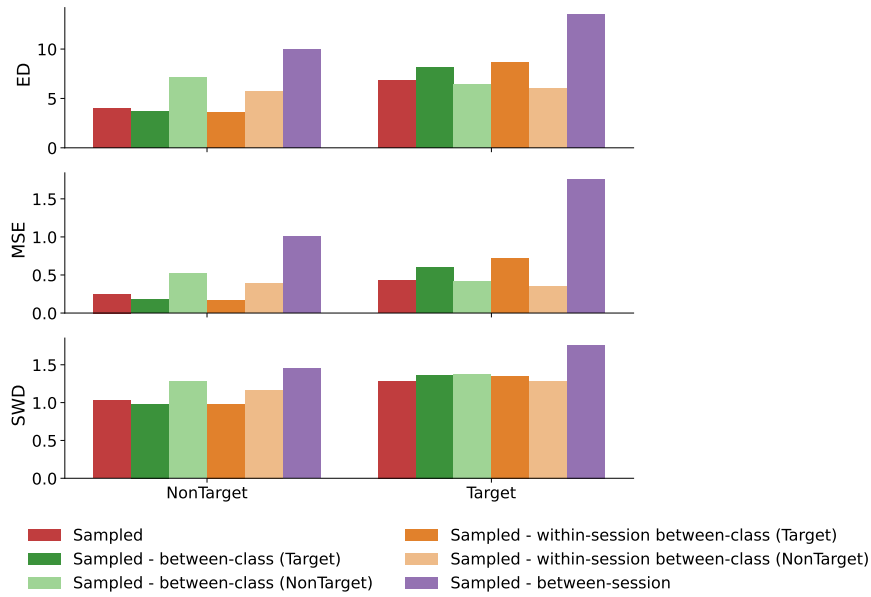


Figure 15: Effect of transfer learning on the average score on the domain-invariant metrics across subjects on the second session of the visual ERP dataset. The predicted class that is indicated in brackets, with both classes predicted in the between-session transfer learning scenario.

Effect of transfer learning on peak metrics (visual ERP - session 2)

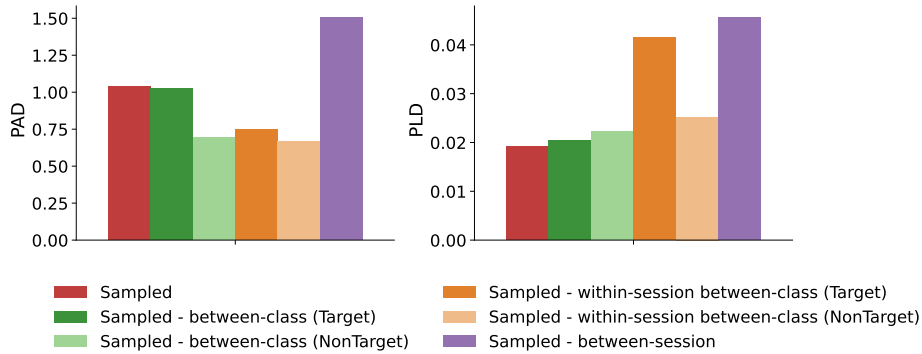


Figure 16: Effect of transfer learning on the average score on the peak metrics across subjects on the second session of the visual ERP dataset. The predicted class that is indicated in brackets, with both classes predicted in the between-session transfer learning scenario.

Effect of transfer learning on domain-specific metrics (visual ERP - session 2)

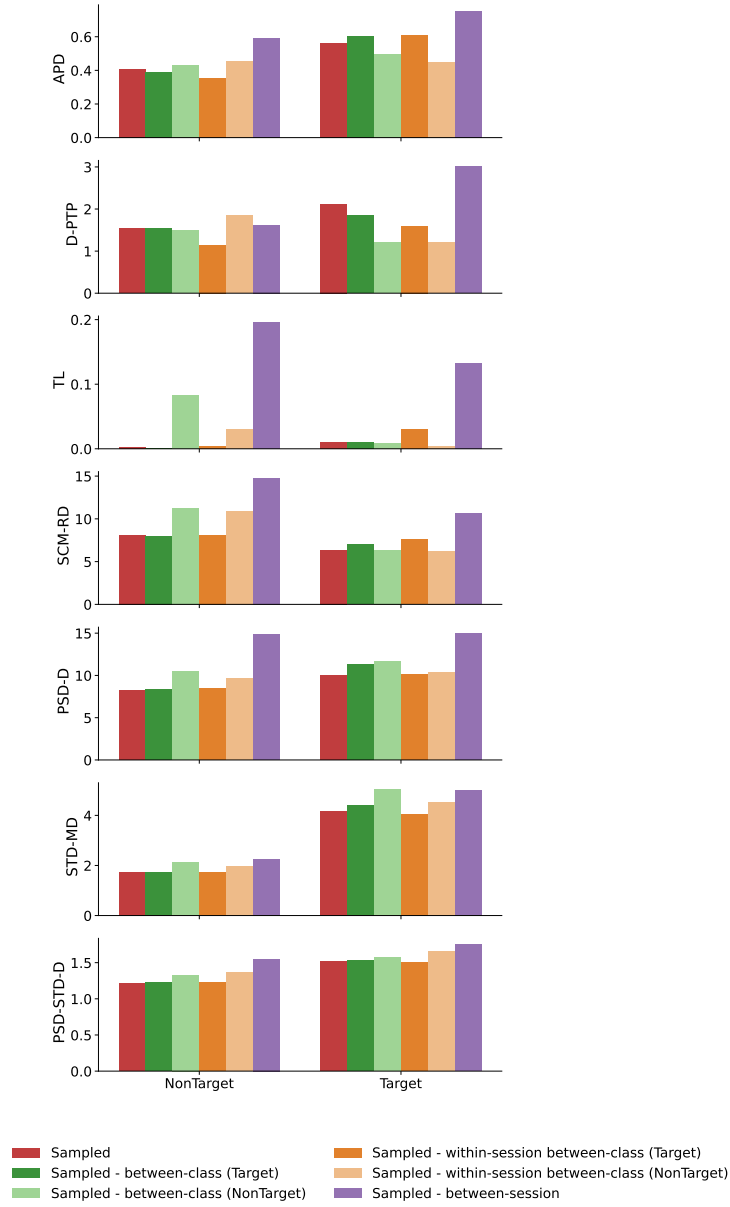


Figure 17: Effect of transfer learning on the average score on the domain-specific metrics across subjects on the second session of the visual ERP dataset. The predicted class that is indicated in brackets, with both classes predicted in the between-session transfer learning scenario.

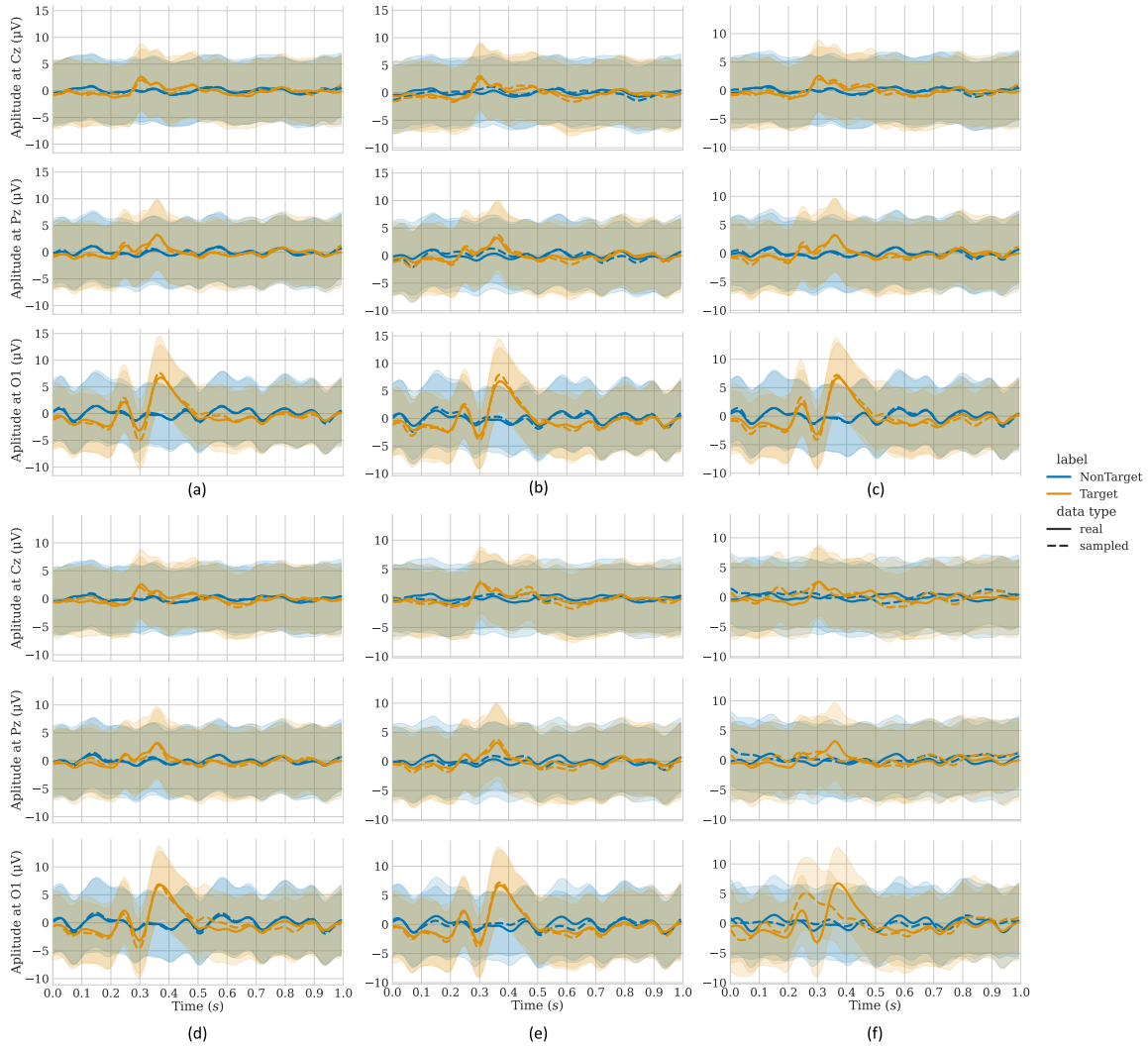


Figure 18: The figure compares sampled ERP response across various transfer learning scenarios using the visual ERP dataset. It illustrates the real and generated average temporal ERP responses of target and non-target stimuli for three selected EEG channels, with error bands indicating standard deviations. The figures are based on the averages of EEG data (658 target and 3331 non-target examples) of subject 24 in session 2. Figure (a) presents regularly sampled data. Figures (b) and (c) depict the within-session, between-class transfer learning scenario, with the target class predicted in (b) and the non-target class predicted in (c). Figures (d) and (e) show the between-class transfer learning scenario, with the target class predicted in (d) and the non-target class predicted in (e). Figure (f) represents the between-session transfer learning scenario, where both classes are predicted.

7 Discussion

7.1 Research Question 3

Visual and auditory ERP datasets The results on the within-training dataset generation indicate that it is possible to generate high-quality EEG data that is specific to each combination of labels in the visual and auditory ERP datasets.

Aphasia dataset In contrast, the results on the aphasia dataset are more ambiguous, as the sampled data is not able to outperform the baseline on multiple metrics. This could be due to a variety of reasons. Firstly, the dataset is much larger than the other two, which could mean that 1) the model was not trained for long enough to achieve optimal results or 2) the model capacity was not sufficient for the complexity of the data. Secondly, the baselines for the domain-specific and domain-invariant metrics are rather dubious, especially for the aphasia and auditory ERP datasets, as will be discussed later. Thirdly, the number of feedback trials could negatively impact the performance due to overfitting, as 1) the number of feedback trials and the session number are highly correlated and 2) the model complexity increases, as the addition of the number of feedback trials requires an additional embedding network. Fourthly, a qualitative inspection shows that the aphasia dataset is quite noisy, which could make it harder for the model to learn the probability distribution underlying the data. Similarly, the low average ABA indicates the absence of a clear ERP response, which could make the metrics that rely on the ERP less reliable. Nonetheless, the high correlation between the ABA of the real and sampled data across the label combinations indicates that the model is capable of generating data that has similar label-related features as the real data in a wide variety of label combinations.

Potential improvements Even though the model is capable of within-dataset generation, there is room for improvement regarding the PSD and the diversity in the data. Tuning the hyperparameters of both the model and the sampling procedure could potentially address these pitfalls. Additionally, including spectrogram information in the conditioning, similar to diff-EEG [17], could improve the PSD of the sample data.

7.2 Research Question 4

Within-session between-class and between-class transfer learning The results on the within-session between-class and between-class transfer learning scenarios indicate that both are feasible with the model. This shows that it is possible to predict the target class for a new subject based on the non-target class and vice versa. This opens the door for exploration of other transfer learning paradigms, such as predict task-related data based on resting state data of the same session.

Between-session transfer learning However, the results on the between-session transfer learning scenario are less promising. The model shows a large drop in performance on almost every metrics. This suggests that 1) the transfer learning scenario is above the capabilities of the model, or 2) there is no consistent trend in the non-stationarities between sessions, making between-session prediction difficult.

Interestingly, some subjects do not have a performance drop in the between-session scenario. We hypothesize that this could be the absence of a change in ERP response between sessions. To test this, we trained the LDA classifier on the first session and used the trained classifier to predict the samples in the second session. The results indicate that while this hypothesis does seem to hold for the 20% label dropout case, it does not explain the results without label dropout (see figure 25). Therefore, it appears that without label dropout the model is adapting the ERP response based on predicted between-session non-stationarities, which suggests that there is no reliable shift between sessions. However, no definitive conclusion about the source for the drop in performance on the between-session transfer learning scenario can be drawn based on these results.

Improvements on the remaining class The results also indicate that removing one class from training data improves results of the model on the remaining class. This improvement suggests that either 1) model capacity may not large enough to represent data of both classes, or 2) class interference might be occurring, where the features present in one class suppress the features of the other class due to parameter sharing. The first issue could be addressed by increasing the number of residual layers. However, in a conditional diffusion model, there is significant overlap in parameters shared across all labels. This makes it difficult to reduce the number of shared parameters without making large changes to the overall diffusion model.

7.3 Limitations

Baseline reliability For the aphasia and auditory ERP datasets, the reliability of the within-subject between-session/SOA is questionable. This is due to the large variety in the number of samples across sessions and SOAs. Consequently, these baselines are computed between datasets of different sizes. This affects 1) the SNR of the ERPs, impacting most domain-invariant and domain-specific metrics, and 2) the standard deviation of the samples, influencing PSD-STD-D and the STD-MD metrics. Despite these concerns, we expect that the baselines do provide a rough estimate of the within-subject variability between sessions and SOAs. Moreover, they should be accurate enough to answer the third research question. It should be noted that this baseline is only necessary in the first application of the metrics, as future models are able to compare directly to the scores obtain by our model.

Domain-specific metrics limitations The domain-specific metrics also have some limitations. Firstly, the PLD is unreliable when there are multiple peaks of similar amplitude in the data. This can be addressed by either 1) only computing the PLD when there is one prominent peak or 2) by computing the peaks of multiple subsets (i.e. 80%) of the real data and only using the lowest PLD. Secondly, the peak-related metrics should be interpreted with caution when the ABA of the real data is relatively low, as this indicates that the absence of a clear P300 response. Thirdly, the sheer amount of metrics makes it difficult to draw an unambiguous conclusion about the model. However, the number of metrics could be reduced, as some measures relatively similar features (e.g., the PLD and TL). Overall, the suite of metrics presented in this work serve as a good starting point for domain-specific evaluation of EEG diffusion models, however, additional improvements are needed.

Sampling speed One of the main downsides of using diffusion models is that the sampling speed is slow, due to the iterative nature of the sampling procedure. However, rapid improvements are being made in the image domain, which are further discussed in section 9.

7.4 Applications

The conditional diffusion model presented in this work can be used for a variety of applications.

Class imbalance The model can be used to alleviate class imbalance by generating samples of the underrepresented class, such as the target class in the case of the oddball paradigms. Class imbalances pose significant problems for accuracy-based classifiers, as the overall accuracy does not give unbiased information about the minority class [43]. Additionally, many classifiers assume that the data contains balanced classes [43]. Therefore, mitigating class-imbalance can 1) enable the application of different classifiers and 2) improve the performance of existing classifiers.

Dataset augmentation The model can be used to augment the existing datasets, which can 1) potentially improve the robustness and accuracy of a classifier, 2) enable the use of more complex classifiers, and 3) be valuable for the benchmarking of novel algorithms.

Fine-tuning The trained weights of the model can be used for fine-tuning on smaller ERP datasets, which is further explored in section 9.

8 Conclusion

In this work, we introduce a diffusion model that is conditioned on multiple labels in parallel using classifier-free guidance, which can generate high-quality

EEG data for each condition in the training data. Moreover, we show that between-class transfer learning is possible with the model. Additionally, we introduce multiple domain-specific metrics that can assist in model evaluation and hyperparameter tuning.

9 Future work

The work presented in this thesis raises many questions, some of which have already been posed in the discussion, that warrant further investigation. To provide some guidance in this endeavour, a comprehensive overview of the potential research directions has been compiled.

Sampling speed Computationally more efficient sampling methods have recently been discovered in the image-domain. One of the most recent innovations is the ability to generate samples using a one-step sampling procedure, similar to generative adversarial networks (GANs) [44]. This enables much quicker sample generation, which makes diffusion models more practical for online applications. Additionally, the reduction in computational cost allows for a more extensive exploration of hyperparameters, potentially improving the performance of the model beyond the result presented in this work.

Motor-imagery paradigm This work exclusively focuses on data obtained from oddball paradigms. However, recent studies have shown that diffusion models can also generate high-quality resting-state and sleep EEG data [14–16, 18], which is more similar to data obtained from motor-imagery paradigms. Consequently, it would be valuable to see whether the results obtained in this work can also translate to motor-imagery paradigms.

At first glance, this adaptation seems relatively simple, as training the diffusion model on motor-imagery data merely requires substituting the dataset. However, it also requires a modification to the metrics, since most metrics used in this work are only suitable for ERP data. To address this, we propose to change the LDA pipeline to use a filter bank common spatial pattern (CSP) for feature extraction [45]. This adjustment should be sufficient to facilitate a preliminary evaluation of the model on motor-imagery EEG data.

Embedding network The embedding neural networks that are currently implemented are not equipped to handle the complete absence of a specific label. One solution that we have briefly experimented with involves pretraining an EEGNet with a 128-dimensional output using triplet loss. This pretrained EEGNet creates embeddings that maximize separability between subjects based on the EEG data, without relying on the subject labels directly. Consequently, novel subjects can still be embedded using the pre-trained EEGNet. The initial results, obtained by removing one subject from the visual ERP dataset when training the EEGNet and the diffusion model, were very promising. Further exploration of this topic could lead to several significant developments:

1. Ability to use new data without retraining: Neither the diffusion model nor the embedding neural network have to be retrained to work with labels not present in the original dataset.
2. Interpolation between labels: The 128-dimensional output of the EEGNet for EEG data that is related to a specific label can be averaged. By subsequently interpolating between the average embeddings of two specific labels, it might be possible to generate EEG that is a mix of the two labels.
3. Between-paradigm transfer learning: By training an EEGNet to distinguish EEG data of different paradigms and combining this with a subject embedding, it might be possible to generate data for a paradigm that a subject has not participated in based on the data that is available from a different paradigm.
4. Qualitative introspection of self-supervised learning models: By conditioning the diffusion model on features extracted from a trained self-supervised learning model, it might be possible to check which features the self-supervised learning model has learned by looking for stationarities in generated samples [46].

Between-session transfer learning Currently there is a congruency between the session and the number of feedback trials in the experiments done on the aphasia dataset. For example, when we try to predict data of session 8 for subject 6 in the between-session transfer learning scenario, the number of feedback trials that is used during data generation is sampled without replacement from the number of feedback trials that subject 6 has experienced in session 8 (e.g., if the subject had seen 6000 trials before session 8 and at the end of the session the subject has seen 7000, then the sampled values are between 6000 and 7000). However, if the hypothesis that between-session non-stationarities are captured by the session embedding and ERP-related learning effects are captured by the number of feedback trials embedding is correct, then it might be possible to estimate changes in the ERP by keeping the session number fixed at a value that is in the training data (e.g., session 7) while changing the number of feedback trials to a value observed in a later session (e.g., session 8). This approach might make it easier for the model to predict a change in the ERP due to rehabilitation, as it does not the prediction of large non-stationarities that happen between sessions.

Another potential method for improving the between-session transfer learning of the aphasia dataset is to condition the diffusion model on 1) an embedding created based on EEG data from an earlier session of the same subject and 2) the number of feedback trials between the input EEG data and the earlier EEG from which the other embedding is created. In this manner, the embedding neural network is forced to learn changes in the ERP response without taking into account session-specific details. This approach utilizes the EEG equivalent of image-to-image diffusion models [47].

Alternatively, instead of predicting cumulative changes in the ERP response based on the number of feedback trials, an alternative approach would be to directly predict the data of the follow-up session using data from the pre-training session. However, given the poor results of between-session transfer learning across datasets, we deem the probability of this approach succeeding rather limited.

Fine-tuning It might be possible to train the diffusion model on a large dataset and subsequently use the pretrained weights for further fine-tuning on a smaller dataset. This could make diffusion models more suitable to smaller datasets. Moreover, it could make it possible to train an unconditioned diffusion model on unlabelled EEG data, of which very large dataset are available, and use these weights to fine-tune a conditional diffusion model on a smaller, labelled dataset.

fMRI-to-EEG Ninthly, by conditioning the diffusion model on functional magnetic resonance imaging (fMRI) images (flattened or embedded using an autoencoder) corresponding to simultaneously recorded EEG data, it could be feasible to convert fMRI images into EEG data. Using a similar approach, converting EEG data into fMRI data might also be possible. This might make it possible to leverage both the high-temporal resolution of EEG data and the high-spatial resolution of fMRI data.

Data efficiency Given that conditional diffusion models can be trained on multiple conditions at once, as opposed to unconditional diffusion models which have to be trained on one specific condition, it seems plausible that conditional diffusion models are more data efficient (i.e., it requires less data of a specific combination to achieve a similar performance). As far as we are aware, this difference in data efficiency has never been properly explored before, even though it would be one of the main reasons for using conditional diffusion models over unconditioned diffusion models in the field of BCIs. This could be tested by training both a conditional and unconditional diffusion model on 25 %, 50 %, 75 %, and 100 % of data of a certain subject, while the conditional diffusion model also has access to all the data of the other subjects. Assessing the generated samples of that particular subject using the metrics presented in this work allows for a comparison between the two models.

Resting-state to task-response transfer learning Lastly, it might be possible to predict task-based responses from resting state data, as there is a high level of functional connectivity between brain areas during rest, as demonstrated by seed-based approaches in fMRI studies [48]. However, it is unclear whether 1) this functional connectivity allows for a prediction of task-related responses, and 2) EEG data is able to adequately capture information about functional connectivity, given the differences in spatial resolution between the two neuroimaging approaches. Preliminary results on the visual ERP dataset were not

very promising. However, each session only has a total of two minutes of resting state data related to the recording of oddball data (one minute recorded before and one minute after the oddball session), which is likely insufficient. Therefore, we would suggest using a dataset that has longer resting-state recordings.

10 References

- [1] R. Abiri, S. Borhani, E. W. Sellers, Y. Jiang, and X. Zhao, “A comprehensive review of EEG-based brain–computer interface paradigms,” *Journal of neural engineering*, vol. 16, no. 1, p. 011 001, 2019.
- [2] B. Blankertz, S. Lemm, M. Treder, S. Haufe, and K.-R. Müller, “Single-trial analysis and classification of ERP components—a tutorial,” *NeuroImage*, vol. 56, no. 2, pp. 814–825, 2011.
- [3] G. Townsend *et al.*, “A novel P300-based brain–computer interface stimulus presentation paradigm: Moving beyond rows and columns,” *Clinical neurophysiology*, vol. 121, no. 7, pp. 1109–1120, 2010.
- [4] F. Lotte *et al.*, “A review of classification algorithms for EEG-based brain–computer interfaces: A 10 year update,” *Journal of neural engineering*, vol. 15, no. 3, p. 031 005, 2018.
- [5] D. Podell *et al.*, “Sdxl: Improving latent diffusion models for high-resolution image synthesis,” *arXiv preprint arXiv:2307.01952*, 2023.
- [6] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, “Diffwave: A versatile diffusion model for audio synthesis,” *arXiv preprint arXiv:2009.09761*, 2020.
- [7] L. Huang, H. Zhang, T. Xu, and K.-C. Wong, “Mdm: Molecular diffusion model for 3d molecule generation,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, 2023, pp. 5105–5112.
- [8] A. Hyvärinen and P. Dayan, “Estimation of non-normalized statistical models by score matching,” *Journal of Machine Learning Research*, vol. 6, no. 4, 2005.
- [9] Y. Song and S. Ermon, “Generative modeling by estimating gradients of the data distribution,” *Advances in neural information processing systems*, vol. 32, 2019.
- [10] M. Welling and Y. W. Teh, “Bayesian learning via stochastic gradient langevin dynamics,” in *Proceedings of the 28th international conference on machine learning (ICML-11)*, Citeseer, 2011, pp. 681–688.
- [11] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-based generative modeling through stochastic differential equations,” *arXiv preprint arXiv:2011.13456*, 2020.
- [12] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” *arXiv preprint arXiv:2207.12598*, 2022.
- [13] S. Torma and L. Szegletes, “Eegwave: A denoising diffusion probabilistic approach for EEG signal generation,” *EasyChair Preprint*, no. 10275, 2023.
- [14] B. Aristimunha *et al.*, “Synthetic sleep EEG signal generation using latent diffusion models,” in *Deep Generative Models for Health Workshop NeurIPS 2023*, 2023.

- [15] G. Sharma, A. Dhall, and R. Subramanian, “Medic: Mitigating EEG data scarcity via class-conditioned diffusion model,” in *Deep Generative Models for Health Workshop NeurIPS 2023*, 2023.
- [16] T. Zhou, X. Chen, Y. Shen, M. Nieuwoudt, C.-M. Pun, and S. Wang, “Generative AI enables EEG data augmentation for alzheimer’s disease detection via diffusion model,” in *2023 IEEE International Symposium on Product Compliance Engineering-Asia (ISPCE-ASIA)*, IEEE, 2023, pp. 1–6.
- [17] K. Shu, Y. Zhao, L. Wu, A. Liu, R. Qian, and X. Chen, “Data augmentation for seizure prediction with generative diffusion model,” *arXiv preprint arXiv:2306.08256*, 2023.
- [18] Y. Wang *et al.*, “Diffmdd: A diffusion-based deep learning framework for MDD diagnosis using EEG,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2024.
- [19] J. Vetter, J. H. Macke, and R. Gao, “Generating realistic neurophysiological time series with denoising diffusion probabilistic models,” *bioRxiv*, pp. 2023–08, 2023.
- [20] G. Tosato, C. M. Dalbagno, and F. Fumagalli, “EEG synthetic data generation using probabilistic diffusion models,” *arXiv preprint arXiv:2303.06068*, 2023.
- [21] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, “Hierarchical text-conditional image generation with clip latents,” *arXiv preprint arXiv:2204.06125*, vol. 1, no. 2, p. 3, 2022.
- [22] M.-H. Lee *et al.*, “EEG dataset and OpenBMI toolbox for three BCI paradigms: An investigation into BCI illiteracy,” *GigaScience*, vol. 8, no. 5, giz002, 2019.
- [23] T. Kaufmann, S. M. Schulz, C. Grünzinger, and A. Kübler, “Flashing characters with famous faces improves ERP-based brain–computer interface performance,” *Journal of neural engineering*, vol. 8, no. 5, p. 056 016, 2011.
- [24] S.-K. Yeom, S. Fazli, K.-R. Müller, and S.-W. Lee, “An efficient ERP-based brain-computer interface using random set presentation and face familiarity,” *PloS one*, vol. 9, no. 11, e111157, 2014.
- [25] J. Sosulski and M. Tangermann, “Spatial filters for auditory evoked potentials transfer between different experimental conditions.,” in *GBCIC*, 2019.
- [26] B. Aristimunha *et al.*, *Mother of all BCI Benchmarks*, version 1.0.0, 2023. [Online]. Available: <https://github.com/NeuroTechX/moabb>.
- [27] M. Musso *et al.*, “Aphasia recovery by language training using a brain–computer interface: A proof-of-concept study,” *Brain communications*, vol. 4, no. 1, fac008, 2022.

- [28] M. Schreuder, B. Blankertz, and M. Tangermann, “A new auditory multi-class brain-computer interface paradigm: Spatial hearing as an informative cue,” *PloS one*, vol. 5, no. 4, e9813, 2010.
- [29] C. Luo, “Understanding diffusion models: A unified perspective,” *arXiv preprint arXiv:2208.11970*, 2022.
- [30] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.
- [31] Y. Song and S. Ermon, “Improved techniques for training score-based generative models,” *Advances in neural information processing systems*, vol. 33, pp. 12 438–12 448, 2020.
- [32] G. Klein, P. Guetschel, G. Silvestri, and M. Tangermann, “Synthesizing EEG signals from event-related potential paradigms with conditional diffusion models,” *arXiv preprint arXiv:2403.18486*, 2024.
- [33] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” *Advances in neural information processing systems*, vol. 30, 2017.
- [34] K. A. Robbins, J. Touryan, T. Mullen, C. Kothe, and N. Bigdely-Shamlo, “How sensitive are EEG results to preprocessing methods: A benchmarking study,” *IEEE transactions on neural systems and rehabilitation engineering*, vol. 28, no. 5, pp. 1081–1090, 2020.
- [35] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, “Eegnet: A compact convolutional neural network for EEG-based brain–computer interfaces,” *Journal of neural engineering*, vol. 15, no. 5, p. 056 013, 2018.
- [36] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” *Advances in neural information processing systems*, vol. 29, 2016.
- [37] A. G. Habashi, A. M. Azab, S. Eldawlatly, and G. M. Aly, “Generative adversarial networks in EEG analysis: An overview,” *Journal of Neuro-Engineering and Rehabilitation*, vol. 20, no. 1, p. 40, 2023.
- [38] O. Ledoit and M. Wolf, “A well-conditioned estimator for large-dimensional covariance matrices,” *Journal of multivariate analysis*, vol. 88, no. 2, pp. 365–411, 2004.
- [39] Y. Rubner, C. Tomasi, and L. J. Guibas, “The earth mover’s distance as a metric for image retrieval,” *International journal of computer vision*, vol. 40, pp. 99–121, 2000.
- [40] J. Rabin, G. Peyré, J. Delon, and M. Bernot, “Wasserstein barycenter and its application to texture mixing,” in *Scale Space and Variational Methods in Computer Vision: Third International Conference, SSVN 2011, Eindhoven, Israel, May 29{June 2, 2011, Revised Selected Papers 3*, Springer, 2012, pp. 435–446.

- [41] N. Bonneel, J. Rabin, G. Peyré, and H. Pfister, “Sliced and radon wasserstein barycenters of measures,” *Journal of Mathematical Imaging and Vision*, vol. 51, pp. 22–45, 2015.
- [42] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, “Classification of covariance matrices using a riemannian-based kernel for BCI applications,” *Neurocomputing*, vol. 112, pp. 172–178, 2013.
- [43] A. Ali, S. M. Shamsuddin, and A. L. Ralescu, “Classification with class imbalance problem,” *Int. J. Advance Soft Compu. Appl*, vol. 5, no. 3, pp. 176–204, 2013.
- [44] T. Yin *et al.*, “One-step diffusion with distribution matching distillation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 6613–6623.
- [45] K. K. Ang, Z. Y. Chin, H. Zhang, and C. Guan, “Filter bank common spatial pattern (FBCSP) in brain-computer interface,” in *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, IEEE, 2008, pp. 2390–2397.
- [46] F. Bordes, R. Balestrieri, and P. Vincent, “High fidelity visualization of what your self-supervised representation knows about,” *arXiv preprint arXiv:2112.09164*, 2021.
- [47] C. Saharia *et al.*, “Palette: Image-to-image diffusion models,” in *ACM SIGGRAPH 2022 conference proceedings*, 2022, pp. 1–10.
- [48] M. P. Van Den Heuvel and H. E. H. Pol, “Exploring the brain network: A review on resting-state fMRI functional connectivity,” *European neuropsychopharmacology*, vol. 20, no. 8, pp. 519–534, 2010.

11 Abbreviations

ABA average balanced accuracy	20
AMUSE auditory multi-class spatial ERP	11
APD average power delta	22
BCI brain-computer interface	4
CFG classifier-free guidance	7
CSP common spatial pattern	46
DDPM denoising-diffusion probabilistic model	6
D-PTP delta peak-to-peak	22
ED Euclidean distance	20
EEG electroencephalogram	1
EMA exponential moving average	17
ERP event-related potential	1
FID Fréchet inception distance	8
fMRI functional magnetic resonance imaging	48
GAN generative adversarial network	46
IS inception score	8
JSD Jensen-Shannon Divergence	8

KL Kullback–Leibler	19
LDA linear discriminant analysis	20
MOABB Mother of All BCI Benchmarks	11
MSE mean squared error	20
MS-SSIM multi-scale structural similarity index metric	8
ODE ordinary differential equation	17
PAD peak amplitude delta	22
PC Predictor-Corrector	6
PLD peak latency delta	22
PSD power spectral density	8
PSD-D power spectral density delta	23
PSD-STD-D power spectral density standard deviation delta	23
SCM spatial covariance matrix	22
SCM-RD spatial covariance matrix Riemmanian distance	22
SDE stochastic differential equation	6
SOA stimulus onset asynchrony	11
SMLD score matching with Langevin dynamics	5

SNR signal-to-noise ratio	4
STD-MD standard deviation Manhattan distance	23
SWD sliced Wasserstein distance	8
TL time lag	21
VP SDE variance persevering stochastic differential equation	16

A Appendix

A.1 SDE hyperparameters

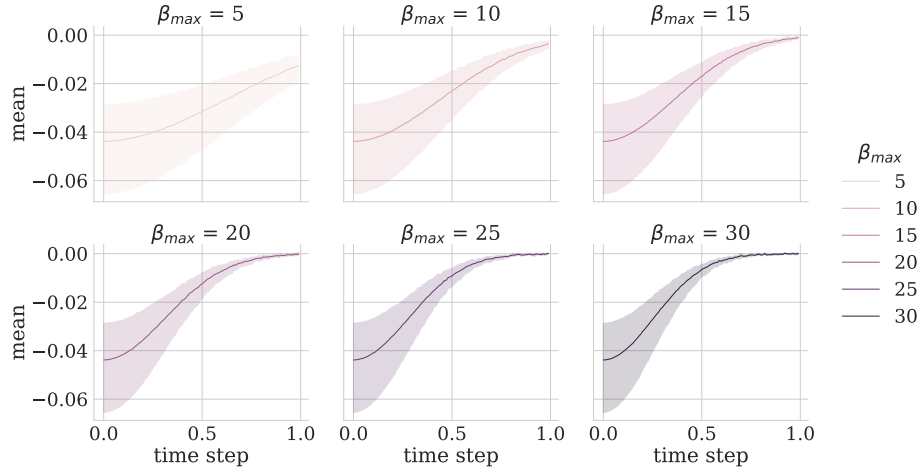


Figure 19: The change in the mean of 5000 random data points of the visual ERP dataset when noised using the VP SDE with a variety of β_{max} , while β_{min} is kept constant at 0.1. This is repeated five times, and the minimum and maximum are displayed as error bands.

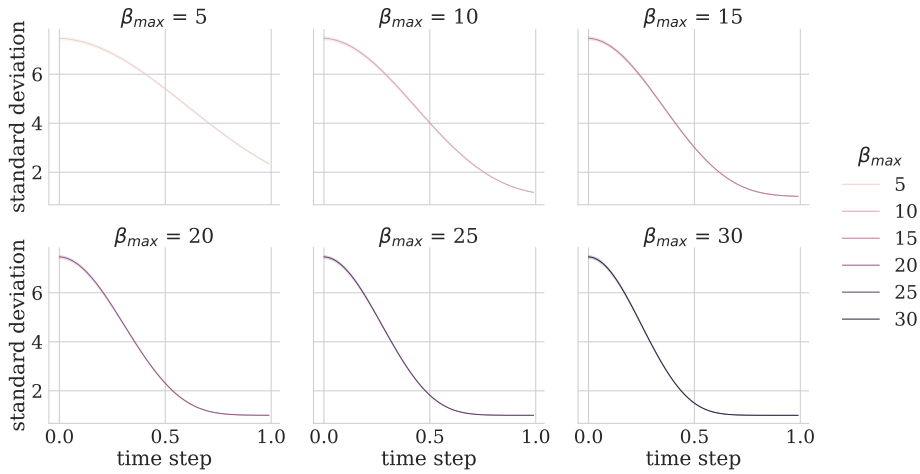


Figure 20: The change in the standard deviation of 5000 random data points of the visual ERP dataset when noised using the VP SDE with a variety of β_{max} , while β_{min} is kept constant at 0.1. This is repeated five times, and the minimum and maximum are displayed as error bands.

A.2 Mathematical equivalence of DDPM and SMLD

The SMLD framework can also be recast into the DDPM framework. However, it is important to note that in the SMLD framework, time points are continuous, while in the DDPM framework, time points are discrete. This explains the difference in indexing: timestep t in the SMLD framework is indexed as $\mathbf{x}(t)$, while in the DDPM framework it is indexed as \mathbf{x}_t .

Firstly, in the DDPM framework, the forward process is described by a discrete Markov chain from $t = 1$ to $t = T$, where each state-transition injects Gaussian noise into the previous state $q(\mathbf{x}_t/\mathbf{x}_{t-1})$ (see figure 1) [11]. The amount of Gaussian noise that is injected at each state-transition is determined by a predetermined variance schedule $(\beta_1, \dots, \beta_T)$ [30]. Thus, the transition from timestep $t-1$ to timestep t is achieved by a mixture between the previous state and the noise [30]:

$$q(\mathbf{x}_t/\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \quad (18)$$

This is equivalent to:

$$\mathbf{x}_t = \sqrt{1 - \beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\eta}_t, \quad \boldsymbol{\eta}_t \sim \mathcal{N}(0, \mathbf{I}) \quad (19)$$

It has been shown that by making the timestep that each state-transitions makes approach 0, this Markov chain can be rewritten to the VP SDE with a continuous timestep $t \in [0, 1]$ [11]:

$$d\mathbf{x} = \frac{1}{2}\beta(t)\mathbf{x} dt + \sqrt{\beta(t)} d\mathbf{w} \quad (20)$$

Additionally, the forward process can also be described in closed form, where \mathbf{x}_t can be computed directly for any timestep t [30]:

$$q(\mathbf{x}_t/\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_0, (1 - \alpha_t)\mathbf{I}), \quad \alpha_t = \prod_{s=1}^t \beta_s \quad (21)$$

Secondly, in the DDPM framework, a model ϵ_θ parameterized by θ is trained to predict the reverse state-transitions:

$$p_\theta(\mathbf{x}_{t-1}/\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)) \quad (22)$$

The $\Sigma_\theta(\mathbf{x}_t, t)$ is set to a constant, for convenience we will use $\beta_t\mathbf{I}$. This means that the only parameter that has to be learned is $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$. By incorporating information about the forward process into the parametrization, it can be shown that $\boldsymbol{\mu}_\theta$ must predict [30]:

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \sqrt{\frac{1 - \beta_t}{1 - \alpha_t}}(\mathbf{x}_t - \sqrt{\frac{\beta_t}{1 - \alpha_t}}\boldsymbol{\epsilon}), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}) \quad (23)$$

Therefore, the model only has to predict the noise $\boldsymbol{\epsilon}$ that was injected into the data point, as the noised data point \mathbf{x}_t is available to the model and the rest

is kept constant. Thus, the loss function becomes the Euclidean norm between the added and predicted noise:

$$\mathbb{E}_t \mathbb{E}_{\mathbf{x}_0} \mathbb{E}_\epsilon [\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t)\|_2^2] \quad (24)$$

This is equivalent to an unweighted score-matching loss in equation (3). Subsequently, the trained model can be applied to reverse the state-transitions:

$$p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) = N(\mathbf{x}_{t-1}; \sqrt{\frac{1}{\beta_t}} (\mathbf{x}_t - \sqrt{\frac{\beta_t}{\alpha_t}} \epsilon_\theta(\mathbf{x}_t, t)), \beta_t \mathbf{I}) \quad (25)$$

This can be rewritten to a Markov chain that is remarkably similar to Langevin dynamics described in equation (2):

$$\mathbf{x}_{t-1} = \sqrt{\frac{1}{\beta_t}} (\mathbf{x}_t - \sqrt{\frac{\beta_t}{\alpha_t}} \epsilon_\theta(\mathbf{x}_t, t)) + \sqrt{\beta_t} \boldsymbol{\eta}_t, \quad \boldsymbol{\eta}_t \sim N(0, \mathbf{I}) \quad (26)$$

Song and colleagues have shown that this is indeed mathematically equivalent to the discretization of the reverse-time VP SDE, which unifies the sampling methods of the DDPM and SMLD framework [11]. Thus, each component of the DDPM framework can be rewritten to an equivalent in the SMLD framework.

A.3 Transfer learning: Auditory ERP dataset

The subjects (1, 3, 11) were randomly selected as the leave N-out subjects for the auditory ERP dataset. The experiments are done without label dropout. The following transfer learning scenarios are tested: 1) between-class transfer learning, and 2) between-SOA transfer learning. In the case of between-SOA transfer learning, the model should predict a specific shift in both the target and non-target response as a result of the change in SOA. This is tested by removing all data for a random SOA for each of the previously selected subjects. As not all subjects have trials with the same SOAs this could not be standardized across subjects. Therefore, the randomly selected SOAs were 235 ms, 273 ms, and 60 ms for subject 1, 3 and 11, respectively.

The results on the classifier performance (ABA) indicate that the prediction of non-target data has the least negative impact, followed by between-SOA transfer learning, and finally followed by the prediction of target data (see figure 21).

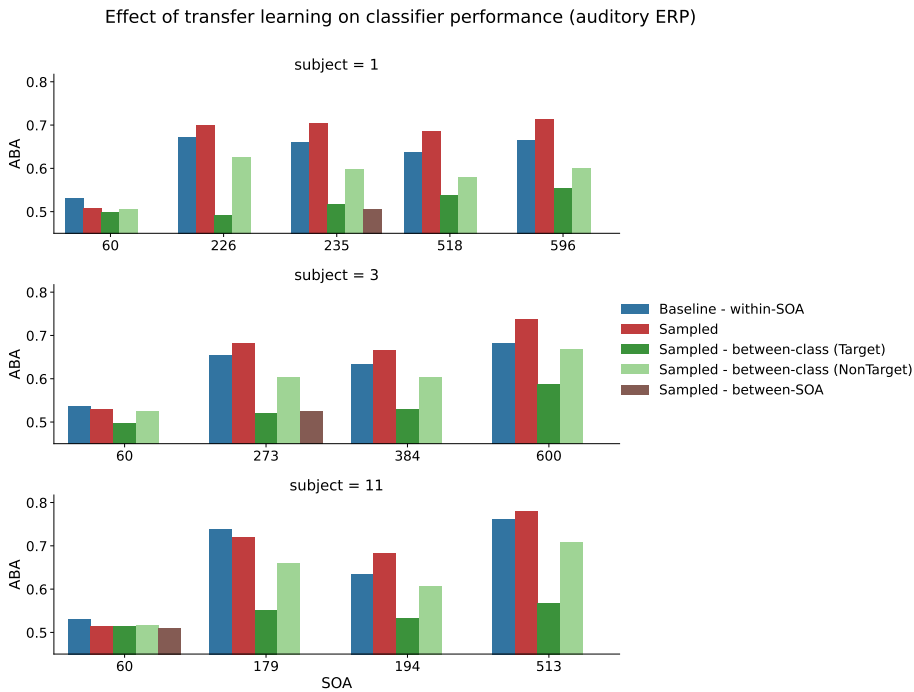


Figure 21: The classifier performance (ABA) of subjects 1, 3, and 11 per SOA in a variety of transfer learning scenarios in the auditory ERP dataset. In the between-SOA transfer learning, only the predicted SOA is added. The predicted class that is indicated in brackets, with both classes predicted in the between-session transfer learning scenario.

A.4 Transfer learning: Aphasia dataset

The subjects (1, 6) were randomly selected as the leave N-out subjects for the aphasia rehabilitation dataset. The experiments are done without label dropout. The following transfer learning scenarios are included: 1) within-session between-class transfer learning, and 2) between-session transfer learning. In both cases, data after the eighth session is removed and data in the eighth session is removed based on the condition (i.e., only target or non-target in case of the between-class transfer learning scenario, and completely removed in the between-session transfer learning scenario), which entails that only data for session eight is predicted. This is done as both subject 1 and 6 have consecutive sessions with the 6D speaker setup for the first eight sessions.

On the one hand, a small decrease in classifier performance can be seen for subject 6 when using data generated by the model trained on the within-session between-class scenario when target data is predicted. On the other hand, there is an increase in classifier performance when the non-target data is predicted. Furthermore, a relatively larger decrease in classifier performance is observed when both classes are predicted, as is the case for the between-session transfer learning scenario (see figure 22).

The classifier performance for the real data of subject 1 is so close to chance level that no statements can be made about a change in performance due to a decrease in data availability during training.

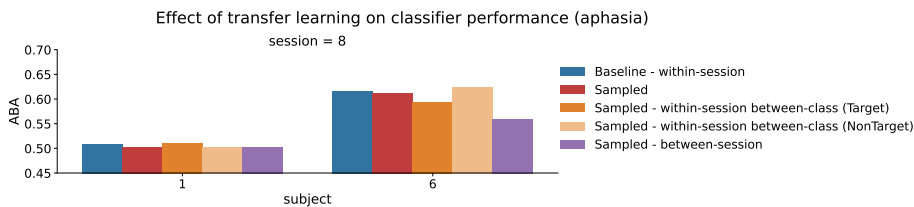


Figure 22: The classifier performance (ABA) in a variety of transfer learning scenarios on subject 1 and 6, session 8 of the aphasia dataset. The predicted class that is indicated in brackets, with both classes predicted in the between-session transfer learning scenario.

A.5 Effect of label dropout

A.5.1 Visual ERP dataset

When training with 20 % label dropout, there is a slight decrease in average ABA across conditions compared to training without label dropout (see figure 23). The highest decrease in ABA for any label combination without label dropout is 0.030, while it is 0.049 with label dropout.

Regarding the within-session between-class transfer learning scenario, label dropout does not seem to impact the ABA when predicting target data (see figure 24). However, performance decreases when predicting non-target data

with label dropout compared to without it. Interestingly, label dropout improves performance in the between-session transfer learning scenario.

The results suggest that the samples generated with label dropout have a similar ERP response to those of the first session, as there seems to be a high correlation between the between-session baseline and the performance of the samples generated with the label dropout model (see figure 25). This correlation is either absent or at least significantly weaker for samples generated without label dropout.

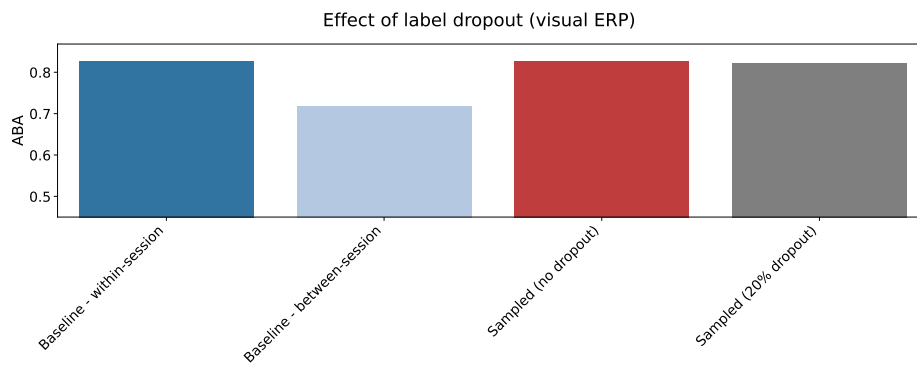


Figure 23: The effect of label dropout on the visual ERP dataset.

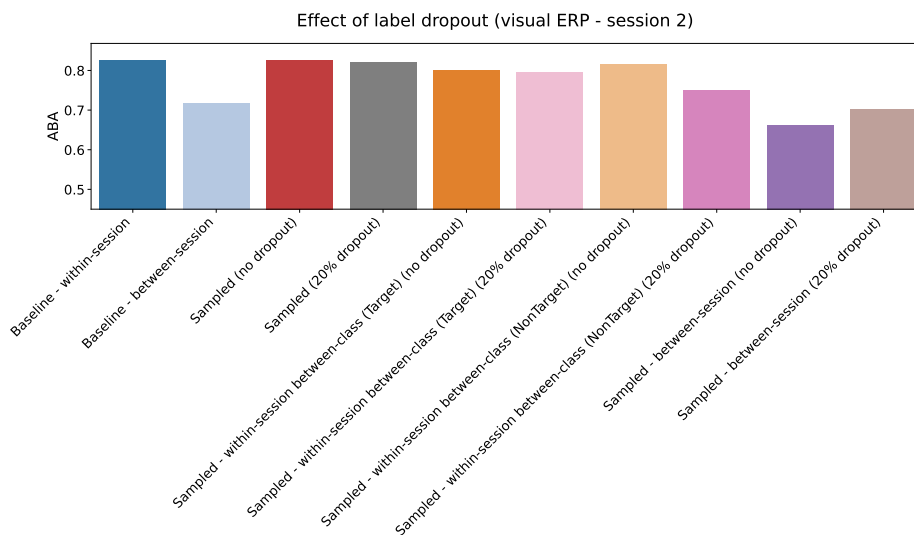


Figure 24: The effect of label dropout on transfer learning in the visual ERP dataset. Results are obtained on the second session of subjects 2, 4, 6, 7, 8, 16, 24, 26, 33, 39, 40, 41, and 53. The predicted class that is indicated in brackets, with both classes predicted in the between-session transfer learning scenario.

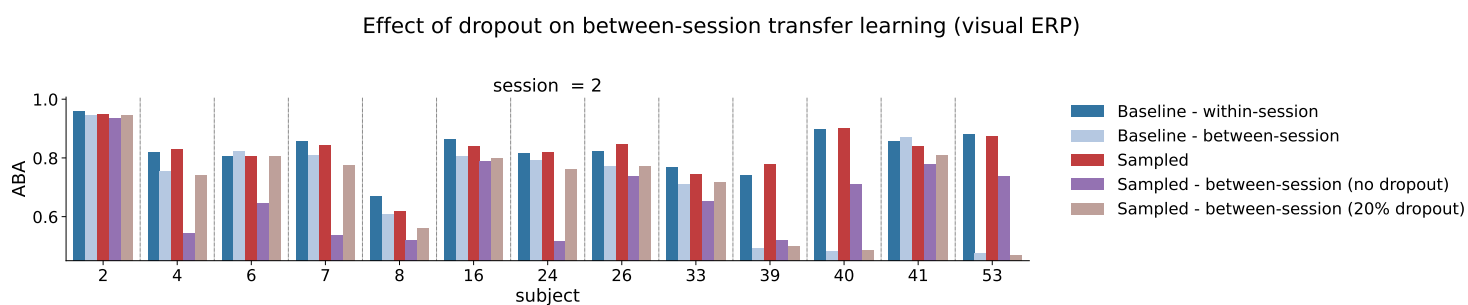


Figure 25: The effect of label dropout on between-session transfer learning in the visual ERP dataset. The predicted class that is indicated in brackets, with both classes predicted in the between-session transfer learning scenario.

A.5.2 Auditory ERP dataset

When training with 20% label dropout, there is a noticeable decrease in average ABA across conditions compared to training without label dropout (see figure 26). The highest decrease in ABA for any label combination without label dropout is 0.023, while it is 0.064 with label dropout.

Regarding the transfer learning scenarios, the inclusion of more labelled training data has a positive effect on both between-class transfer learning scenarios. This is most noticeable when removing non-target data. On the other hand, label dropout does seem to increase ABA performance in the between-SOA transfer learning scenario compared. However, this higher performance might also be attributable to a consistent ERP across SOA, similar to the results obtained on the visual ERP dataset. However, as only three label combinations are tested, this is not further explored.

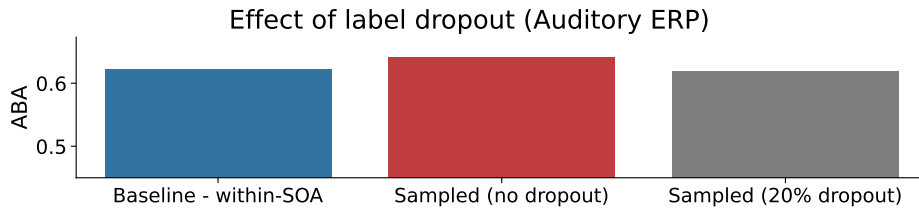


Figure 26: The effect of label dropout on the auditory ERP dataset. The predicted class that is indicated in brackets, with both classes predicted in the between-session transfer learning scenario.

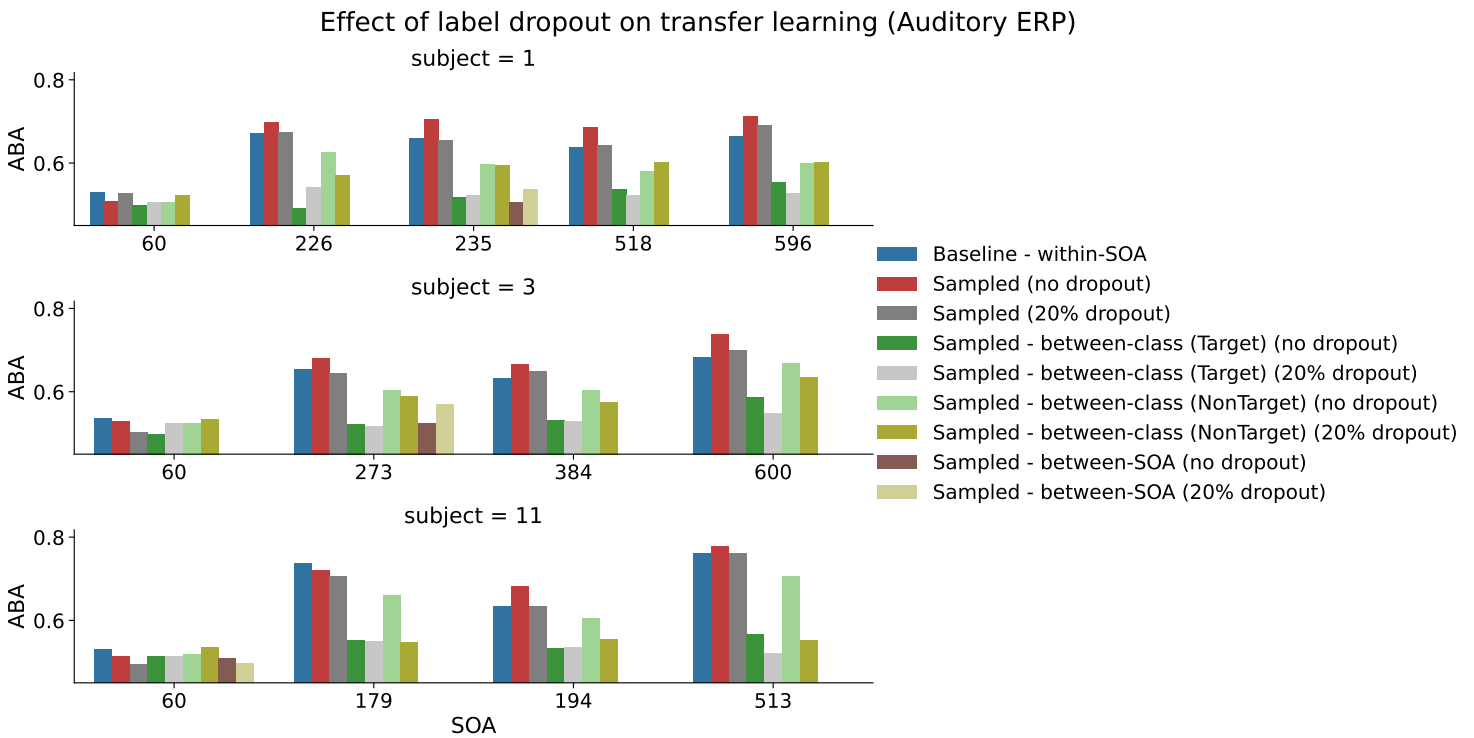


Figure 27: The effect of label dropout on transfer learning in the auditory ERP dataset. The predicted class that is indicated in brackets, with both classes predicted in the between-session transfer learning scenario.