

1 Precise temporal localisation of M/EEG effects with
2 Bayesian generalised additive multilevel models

3 Ladislas Nalborczyk¹ and Paul Bürkner²

4 ¹Aix Marseille Univ, CNRS, LPL

5 ²TU Dortmund University, Department of Statistics

6 Abstract

7 Time-resolved electrophysiological measurements such as those obtained through magneto- or electro-encephalography (M/EEG) offer a unique window onto the neural activity underlying cognitive processes. Researchers are often interested in determining whether and when these signals differ across experimental conditions or participant groups. The conventional approach involves mass-univariate statistical testing across time and space, followed by corrections for multiple comparisons such as cluster-based inference. While effective for controlling error rates at the cluster-level, cluster-based inference comes with a significant limitation: by shifting the focus of inference from individual time points to clusters, it prevents drawing conclusions about the precise onset or offset of observed effects. Here, we introduce a *model-based* approach for analysing M/EEG timeseries such as event-related potentials (ERPs) or decoding performance over time. Our approach leverages Bayesian generalised additive multilevel models, providing posterior probabilities that an effect is above zero (or above chance) at each time point, while naturally accounting for temporal dependencies and between-subject variability. Using both simulated and actual M/EEG datasets, we demonstrate that this approach substantially outperforms conventional methods in estimating the onset and offset of neural effects, yielding more precise and reliable results. We provide an R package implementing the method and describe how it can be integrated into M/EEG analysis pipelines using MNE-Python.

Keywords: EEG, MEG, cluster-based inference, simulation, multiple comparisons, generalised additive models, mixed-effects models, multilevel models, Bayesian statistics, brms

8 **Table of contents**

9 Introduction	3
10 Introduction	3
11 Problem statement	3

12	Statistical errors and cluster-based inference	4
13	Previous work on modelling M/EEG data	5
14	Generalised additive models	6
15	Objectives	8
16	Methods	8
17	M/EEG data simulation	8
18	Model description and model fitting	9
19	Error properties of the proposed approach	12
20	Comparing the onsets/offsets estimates across approaches	13
21	Simulation study	14
22	Application to actual MEG data	14
23	Results	16
24	Simulation study (bias and variance)	16
25	Application to actual MEG data (reliability)	17
26	Discussion	20
27	Summary of the proposed approach	20
28	Limitations and future directions	20
29	Data and code availability	22
30	Packages	22
31	Acknowledgements	22
32	References	23
33	Application to 2D time-resolved decoding results (cross-temporal generalisation)	29
34	Alternative to GAMs: Approximate Gaussian Process regression	32
35	How to choose the GAM basis dimension?	34
36	R package and integration with MNE-Python	35

Ladislas Nalborczyk  <https://orcid.org/0000-0002-7419-9855>

Paul Bürkner  <https://orcid.org/0000-0001-5765-8995>

The authors have no conflicts of interest to disclose.

Correspondence concerning this article should be addressed to Ladislas Nalborczyk, Aix Marseille Univ, CNRS, LPL, 5 avenue Pasteur, 13100 Aix-en-Provence, France, email: ladislas.nalborczyk@cnrs.fr

37 **Precise temporal localisation of M/EEG effects with Bayesian generalised additive
38 multilevel models**

1 **Introduction**

2 **Problem statement**

3 Understanding the temporal dynamics of cognitive processes requires methods that can
4 capture fast-changing neural activity with high temporal resolution. Magnetoencephalography
5 and electroencephalography (M/EEG) are two such methods, widely used in cognitive neuro-
6 science for their ability to track brain activity at the millisecond scale. These techniques provide
7 rich time series data that reflect how neural responses unfold in response to stimuli or tasks. A
8 central goal in many M/EEG studies is to determine whether, when, and where neural responses
9 differ across experimental conditions or groups.

10 The conventional approach involves mass-univariate statistical testing through time
11 and/or space followed by some form of correction for multiple comparisons with the goal of
12 maintaining the familywise error rate (FWER) or the false discovery rate (FDR) at the nomi-
13 nal level (e.g., 5%). Cluster-based inference is the most common way of achieving this sort of
14 error control in the M/EEG literature, being the recommended approach in several software
15 programs (e.g., [EEGLab](#), [Delorme & Makeig, 2004](#); [MNE-Python](#), [Gramfort, 2013](#)). While ef-
16 fective for controlling error rates, cluster-based inference comes with a significant limitation:
17 by shifting the focus of inference from individual datapoints (e.g., timesteps, sensors, voxels) to
18 clusters, it prevents the ability to draw precise conclusions about the spatiotemporal localisation
19 of such effects ([Maris & Oostenveld, 2007](#); [Sassenhagen & Draschkow, 2019](#)). As pointed out
20 by Maris & Oostenveld (2007): “there is a conflict between this interest in localized effects and
21 our choice for a global null hypothesis: by controlling the FA [false alarm] rate under this global
22 null hypothesis, one cannot quantify the uncertainty in the spatiotemporal localization of the
23 effect”. Even worse, Rosenblatt et al. (2018) note that cluster-based inference suffers from low
24 spatial resolution: “Since discovering a cluster means that ‘there exists at least one voxel with
25 an evoked response in the cluster’, and not that ‘all the voxels in the cluster have an evoked
26 response’, it follows that the larger the detected cluster, the less information we have on the
27 location of the activation.” As a consequence, cluster-based inference is expected to perform
28 poorly for localising the onset of M/EEG effects; a property that was later demonstrated in
29 simulation studies (e.g., [Rousselet, 2025](#); [Sassenhagen & Draschkow, 2019](#)).

30 To overcome the limitations of cluster-based inference, we introduce a novel *model-based*
31 approach for precisely localising M/EEG effects in time, space, and other dimensions. The pro-
32 posed approach, based on Bayesian generalised additive multilevel models, allows quantifying
33 the posterior probability of effects being above chance at the level of timesteps, sensors, voxels,
34 etc, while naturally taking into account spatiotemporal dependencies present in M/EEG data.
35 We compare the performance of the proposed approach to well-established alternative meth-
36 ods using both simulated and actual M/EEG data and show that it significantly outperforms
37 alternative methods in estimating the onset and offset of M/EEG effects.

38 **Statistical errors and cluster-based inference**

39 The issues with multiple comparisons represent a common and well-recognised danger in
 40 neuroimaging and M/EEG research, where the collected data allows for a multitude of potential
 41 hypothesis tests and is characterised by complex structures of spatiotemporal dependencies. The
 42 probability of obtaining at least one false positive in an ensemble (family) of m tests (i.e., the
 43 FWER) is computed as $1 - (1 - \alpha)^m$ (for $m = 10$ tests and $\alpha = 0.05$, it is approximately equal
 44 to 0.4). Different methods exist to control the FWER (i.e., bring it back to α). Most methods
 45 apply a simple correction to series of p -values issued from univariate statistical tests (e.g., t-
 46 tests). For instance, the Bonferroni correction (Dunn, 1961) consists in setting the significance
 47 threshold to α/m , or equivalently, multiplying the p -values by m and using the standard α
 48 significance threshold. This method is generally overconservative (i.e., under-powered) as it
 49 assumes statistical independence of the tests, an assumption that is clearly violated in the
 50 context of M/EEG timeseries characterised by massive spatiotemporal dependencies. Some
 51 alternative methods aims at controlling the FDR, defined as the proportion of false positive
 52 *among positive tests* (e.g., Benjamini & Hochberg, 1995; Benjamini & Yekutieli, 2001). However,
 53 a major limitation of both types of corrections is that they do not take into account the spatial
 54 and temporal information contained in M/EEG data.

55 A popular technique to account for spatiotemporal dependencies while controlling the
 56 FWER is cluster-based inference (Bullmore et al., 1999; Maris & Oostenveld, 2007). A typ-
 57 ical cluster-based inference consists of two successive steps (for more details on cluster-based
 58 inference, see for instance Frossard & Renaud, 2022; Maris, 2011; Maris & Oostenveld, 2007;
 59 Sassenhagen & Draschkow, 2019). First, clusters are defined as sets of contiguous timesteps,
 60 sensors, voxels, etc, whose activity, summarised by some test statistic (e.g., a t -value), exceeds
 61 a predefined threshold (e.g., the 95th percentile of the parametric null distribution). Clusters
 62 are then characterised by their height (i.e., maximal value), extent (number of constituent ele-
 63 ments), or some combination of both, for instance by summing the statistics within a cluster, an
 64 approach referred to as “cluster mass” (Maris & Oostenveld, 2007; Pernet et al., 2015). Then,
 65 the null hypothesis is tested by computing a p -value for each identified cluster by comparing
 66 its mass with the null distribution of cluster masses (obtained via permutation). As alluded
 67 previously, a significant cluster is a cluster which contains *at least one* significant time-point.
 68 As such, it would be incorrect to conclude, for instance, that the timestep of a significant cluster
 69 is the first moment at which some conditions differ (Frossard & Renaud, 2022; Sassenhagen &
 70 Draschkow, 2019). In other words, because the inference is performed at the second step (i.e.,
 71 once clusters have been formed), no conclusion can be made about individual datapoints (e.g.,
 72 timesteps, sensors, etc).

73 As different cluster-forming thresholds lead to clusters with different spatial or temporal
 74 extent, this initial threshold modulates the sensitivity of the subsequent permutation test. The
 75 threshold-free cluster enhancement (TFCE) method was introduced by S. Smith & Nichols
 76 (2009) to overcome this choice of an arbitrary threshold. In brief, the TFCE method works
 77 as follows. Instead of picking an arbitrary cluster-forming threshold (e.g., $t = 2$), the methods
 78 consist in trying all (or many) possible thresholds in a given range and checking whether a given
 79 datapoint (e.g., timestep, sensor, voxel) belongs to a significant cluster under any of the set of

80 thresholds. Then, instead of using cluster mass, one uses a weighted average between the cluster
 81 extend (e , how broad is the cluster, that is, how many connected samples it contains) and the
 82 cluster height (h , how high is the cluster, that is, how large is the test statistic). The TFCE
 83 score at each timestep t is given by:

$$\text{TFCE}(t) = \int_{h=h_0}^{h=h_t} e(h)^E h^H dh$$

84 where h_0 is typically 0 and parameters E and H are set a priori (typically to 0.5 and
 85 2, respectively) and control the influence of the extend and height on the TFCE. Note that in
 86 practice, this integral is approximated by a sum over small h increments. Then, a p -value for
 87 each timestep t is computed by comparing its TFCE with the null distribution of TFCE values
 88 (obtained via permutation). For each permuted signal, we keep the maximal value over the whole
 89 signal for the null distribution of the TFCE. The TFCE combined with permutation (assuming
 90 a large enough number of permutations) has been shown to provide accurate FWER control
 91 (e.g., [Pernet et al., 2015](#)). However, further simulation work showed that cluster-based methods
 92 (including TFCE) perform poorly in localising the onset of M/EEG effects (e.g., [Rousselet,](#)
 93 [2025; Sassenhagen & Draschkow, 2019](#)).

94 To sum up, the main limitation of cluster-based inference is that it allows for inference
 95 at the cluster level only, not allowing inference at the level of timesteps, sensors, etc. As a
 96 consequence, it does not allow inferring the precise spatial and temporal localisation of effects.
 97 In the following, we briefly review previous modelling work of M/EEG data. Then, we provide
 98 a short introduction to generalised additive models (GAMs) to illustrate how these models can
 99 be used to precisely estimate the onset and offset of M/EEG effects.

100 Previous work on modelling M/EEG data

101 Scalp-recorded M/EEG signals capture neural activity originating from various brain
 102 regions and are often contaminated by artifacts unrelated to the cognitive processes under
 103 investigation. Consequently, analysing M/EEG data necessitates methods that can disentangle
 104 task-relevant neural signals from extraneous “noise.” A widely adopted technique for this
 105 purpose is the estimation of event-related potentials (ERPs), which are stereotyped electro-
 106 physiological responses time-locked to specific sensory, cognitive, or motor events. Typically,
 107 ERPs are derived by averaging EEG or MEG epochs across multiple trials aligned to the event
 108 of interest (e.g., stimulus onset), thereby enhancing the signal-to-noise ratio by attenuating
 109 non-time-locked activity. However, this averaging approach has notable limitations: it assumes
 110 consistent latency and amplitude across trials and is primarily suited for simple categorical
 111 designs. Such assumptions may not hold in more complex experimental paradigms, potentially
 112 leading to suboptimal ERP estimations.

113 To overcome the limitations of simple averaging, several model-based approaches for
 114 estimating ERPs have been proposed. These methods are motivated by the observation that
 115 traditional ERP averaging is mathematically equivalent to fitting an intercept-only linear re-
 116 gression model in a simple categorical design without overlapping events ([N. J. Smith & Kutas,](#)
 117 [2014a](#)). In contrast to simple averaging, regression-based approaches to ERP estimation of-

fer substantially greater flexibility. Notably, they allow for the modelling of both linear and nonlinear effects of continuous predictors—such as word frequency or age (e.g., N. J. Smith & Kutas, 2014a, 2014b; Tremblay & Newman, 2014)—and enable the disentangling of overlapping cognitive processes (e.g., Ehinger & Dimigen, 2019; Skukies et al., 2024; Skukies & Ehinger, 2021). One widely used implementation of this approach is provided by the LIMO EEG toolbox (Pernet et al., 2011), which follows a multi-stage analysis pipeline. First, a separate regression model is fit for each datapoint (e.g., each time point and electrode) at the individual participant level to estimate ERP responses. This is followed by group-level statistical analyses of the resulting regression coefficients, often accompanied by multiple comparison corrections or cluster-based inference (for recent applied examples, see Dunagan et al., 2025; Wüllhorst et al., 2025).

Although this framework allows for the inclusion of a wide range of predictors—both continuous and categorical, linear and nonlinear—it still has important limitations. First, fitting separate models for each datapoint ignores the spatiotemporal dependencies inherent in M/EEG data, potentially reducing statistical power and interpretability. Second, the subsequent group-level analyses typically do not account for hierarchical dependencies, between-participant or between-stimulus variability, which could otherwise be addressed through multilevel modelling. Finally, because the output of this procedure is summarised by cluster-based inference, its conclusions remain subject to the limitations discussed in the previous section.

Beyond modelling nonlinear effects of continuous predictors on ERP amplitudes, generalised additive models (GAMs) have been employed to capture the temporal dynamics of ERPs themselves, effectively modelling the shape of the waveform over time (Abugaber et al., 2023; Baayen et al., 2018; Meulman et al., 2015, 2023). This approach allows for the estimation of smooth, data-driven functions that characterise how neural responses evolve over time, offering a flexible alternative to traditional linear models. In the following section, we provide a brief introduction to GAMs, highlighting their applicability to M/EEG time series analysis and the advantages they offer over conventional methods.

Generalised additive models

In generalised additive models (GAMs), the functional relationship between the predictors and the response variable is decomposed into a sum of low-dimensional non-parametric functions. A typical GAM has the following form:

$$y_i \sim \text{EF}(\mu_i, \phi)$$

$$g(\mu_i) = \underbrace{\mathbf{A}_i \gamma}_{\text{parametric part}} + \underbrace{\sum_{j=1}^J f_j(x_{ij})}_{\text{non-parametric part}}$$

where $y_i \sim \text{EF}(\mu_i, \phi)$ denotes that the observations y_i are distributed as some member of the exponential family of distributions (e.g., Gaussian, Gamma, Beta, Poisson) with mean μ_i and scale parameter ϕ ; $g(\cdot)$ is the link function, \mathbf{A}_i is the i th row of a known parametric model matrix, γ is a vector of parameters for the parametric terms (to be estimated), f_j is a smooth function of covariate x_j (to be estimated as well). The smooth functions f_j are represented in

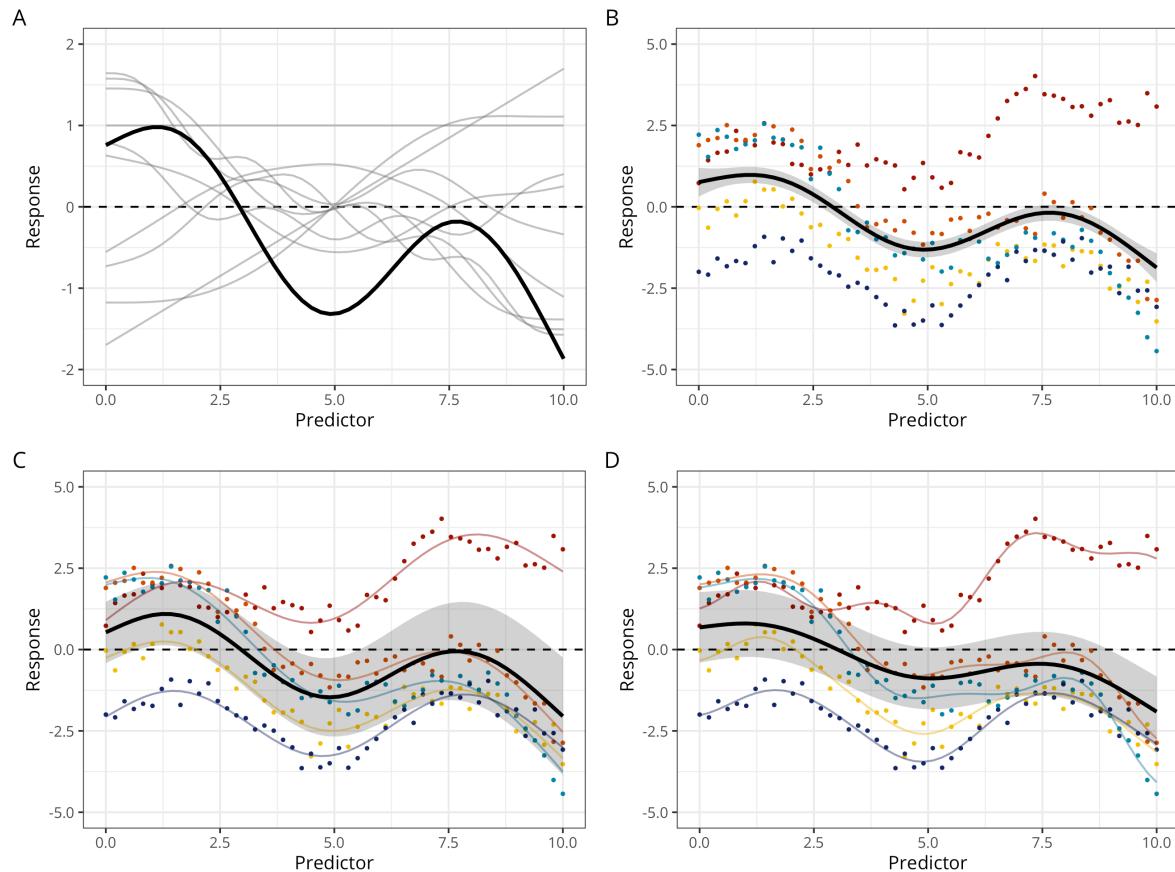
154 the model as a weighted sum of K simpler, basis functions:

$$f_j(x_{ij}) = \sum_{k=1}^K \beta_{jk} b_{jk}(x_{ij})$$

155 where β_{jk} is the weight (coefficient) associated with the k th basis function $b_{jk}()$ evaluated
 156 at the covariate value x_{ij} for the j th smooth function f_j . To clarify the terminology at this point:
 157 *splines* are functions composed of simpler functions. These simpler functions are called *basis*
 158 *functions* (e.g., cubic polynomial, thin-plate) and the set of basis functions is called a *basis*.
 159 Each basis function gets its coefficient and the resultant spline is the sum of these weighted
 160 basis functions (Figure 1A). Splines coefficients are penalised (usually through the squared of
 161 the smooth functions' second derivative) in a way that can be interpreted, in Bayesian terms,
 162 as a prior on the “wiggliness” of the function (Miller, 2025; Wood, 2017a). In other words, more
 163 complex (wiggly) basis functions are automatically penalised.

Figure 1

Different types of GAMs. A: GAMs predictions are computed as the weighted sum (in black) of basis functions (here thin-plate basis functions, in grey). B: Constant-effect GAM, with 5 participants in colours and the group-level prediction in black. C: Varying-intercept + varying-slope GAMM (with constant smoother). D: Varying-intercept + varying-slope + varying-smoother GAMM. In this model, each participant gets its own intercept, slope, and degree of ‘wiggliness’ (smoother).



164 A detailed treatment of the technical underpinnings of GAMs is beyond the scope of

this article (see reference books such as [Hastie & Tibshirani, 2017](#); [Wood, 2017a](#)). However, it is worth emphasising that GAMs have been successfully applied to a wide range of time series data across the cognitive sciences, including pupillometry (e.g., [Rij et al., 2019](#)), articulography (e.g., [Wieling, 2018](#)), speech formant dynamics (e.g., [Sóskuthy, 2021](#)), neuroimaging data (e.g., [Dinga et al., 2021](#)), and event-related potentials (e.g., [Abugaber et al., 2023](#); [Baayen et al., 2018](#); [Meulman et al., 2015, 2023](#)). Their appeal for modelling M/EEG data lies in their ability to flexibly capture the complex shape of ERP waveforms without overfitting, through the use of smooth functions constrained by penalisation. Recent extensions, such as distributional GAMs ([Rigby & Stasinopoulos, 2005](#); [Umlauf et al., 2018a](#)), allow researchers to model not only the mean structure but also the variance (or scale) and other distributional properties as functions of predictors—a feature that has proven useful in modelling neuroimaging data (e.g., [Dinga et al., 2021](#)). Moreover, hierarchical or multilevel GAMs ([E. J. Pedersen et al., 2019](#)) provide a principled way to account for the nested structure of M/EEG data (e.g., trials within participants), enabling the inclusion of varying intercepts, slopes, and smoothers (as illustrated in Figure 1C-D). This approach mitigates the risk of overfitting and reduces the influence of outliers on smooth estimates ([Baayen & Linke, 2020](#); [Meulman et al., 2023](#)).

181 Objectives

182 Cluster-based permutation tests are widely used in M/EEG research to identify statistically significant effects across time and space. However, these methods have notable limitations, particularly in accurately determining the precise onset and offset of neural effects. To address these limitations, we developed a model-based approach relying on Bayesian generalised additive multilevel models implemented in R via the `brms` package ([Bürkner, 2017, 2018](#)). We evaluated the performance of this approach against conventional methods using both simulated and actual M/EEG data. Our findings demonstrate that this method provides more precise and reliable estimates of effect onsets and offsets, offering a valuable tool for researchers seeking to enhance the temporal resolution of their analyses.

191

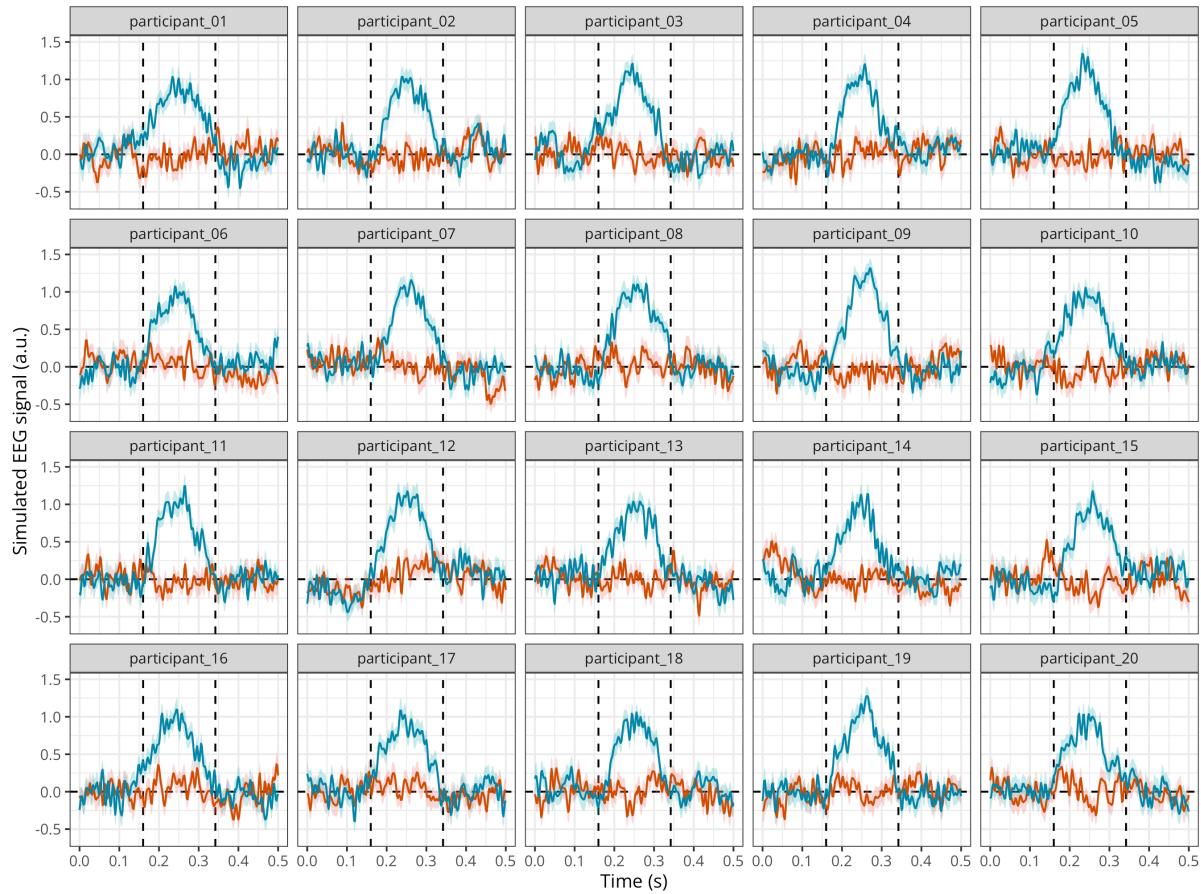
Methods

192 M/EEG data simulation

193 Following the approach of [Sassenhagen & Draschkow \(2019\)](#) and [Rousselet \(2025\)](#), we 194 simulated EEG data stemming from two conditions, one with noise only, and the other with 195 noise + signal. As in previous studies, the noise was generated by superimposing 50 sinusoids 196 at different frequencies, following an EEG-like spectrum (see code in the online supplementary 197 materials and details in [Yeung et al., 2004](#)). As in [Rousselet \(2025\)](#), the signal was generated 198 from a truncated Gaussian distribution with an objective onset at 160 ms, a peak at 250 ms, and 199 an offset at 342 ms. We simulated this signal for 250 timesteps between 0 and 0.5s, akin to a 500 200 Hz sampling rate. We simulated data for a group of 20 participants (with variable true onset) 201 with 50 trials per participant and condition (Figure 2). All figures and simulation results can be 202 reproduced using the R code available on GitHub: https://github.com/lmalborczyk/brms_meeg.

Figure 2

Mean simulated EEG activity in two conditions with 50 trials each, for a group of 20 participants. The error band represents the mean +/- 1 standard error of the mean.



203 Model description and model fitting

204 We then fitted a Bayesian GAM (BGAM) using the `brms` package (Bürkner, 2017, 2018;
 205 Nalborczyk et al., 2019) and default priors (i.e., weakly informative priors). We ran eight
 206 Markov Chain Monte-Carlo (MCMC) to approximate the posterior distribution, including each
 207 5000 iterations and a warmup of 2000 iterations, yielding a total of $8 \times (5000 - 2000) = 24000$
 208 posterior samples to use for inference. Posterior convergence was assessed examining trace plots
 209 as well as the Gelman–Rubin statistic \hat{R} (Gabry et al., 2019; Gelman et al., 2020). The `brms`
 210 package uses the same syntax as the R package `mgcv` v 1.9-3 (Wood, 2017b) for specifying smooth
 211 effects. Figure 3 shows the predictions of this model together with the raw data.

```
# computing the average EEG signals for each participant
ppt_df <- raw_df %>%
  group_by(participant, condition, time) %>%
  summarise(eeg_mean = mean(eeg) ) %>%
  ungroup()

# defining a contrast for condition
```

```

contrasts(ppt_df$condition) <- c(-0.5, 0.5)

# fitting the BGAM
gam <- brm(
  # thin-plate regression splines with k-1 basis functions
  eeg_mean ~ 1 + condition + s(time, bs = "tp", k = 20, by = condition),
  data = ppt_df,
  family = gaussian(),
  warmup = 2000,
  iter = 5000,
  chains = 8,
  cores = 8,
  file = "models/gam.rds"
)

```

212 However, the previous model only included constant (fixed) effects, thus not prop-
 213 erly accounting for between-participant variability. We next fitted a multilevel version of the
 214 BGAM (BGAMM, for an introduction to Bayesian multilevel models in `brms`, see [Nalborczyk](#)
 215 [et al., 2019](#)) including a varying intercept and slope for participant (but only with a constant
 216 smoother). Although it is possible to fit a BGAMM using data at the single-trial level, we
 217 present a computationally lighter version of the model that is fitted directly on by-participant
 218 summary statistics (mean and SD), similar to what is done in meta-analysis.

```

# computing the mean and SD of EEG signals for each participant
summary_df <- raw_df %>%
  group_by(participant, condition, time) %>%
  summarise(eeg_mean = mean(eeg), eeg_sd = sd(eeg) ) %>%
  ungroup()

# defining a contrast for condition
contrasts(summary_df$condition) <- c(-0.5, 0.5)

# fitting the BGAMM
meta_gam <- brm(
  # using the by-participant mean and SD of ERPs
  eeg_mean | se(eeg_sd) ~
    1 + condition + (1 + condition | participant) +
    s(time, bs = "tp", k = 20, by = condition),
  data = summary_df,
  family = gaussian(),
  warmup = 2000,
  iter = 5000,
  chains = 8,

```

```

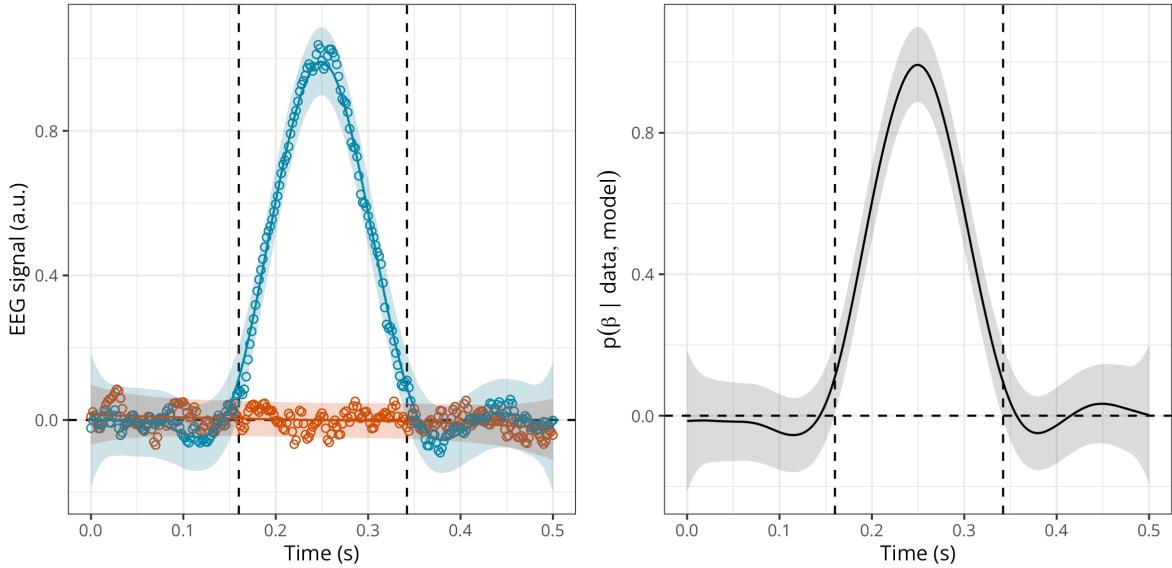
cores = 8,
file = "models/meta_gam.rds"
)

```

219 We depict the posterior predictions together with the posterior estimate of the slope
 220 for **condition** at each timestep (Figure 3). This figure suggests that the BGAMM provides an
 221 adequate description of the simulated data (see further posterior predictive checks in Section C).

Figure 3

Posterior estimate of the EEG activity in each condition (left) and posterior estimate of the difference in EEG activity (right) according to the BGAMM.

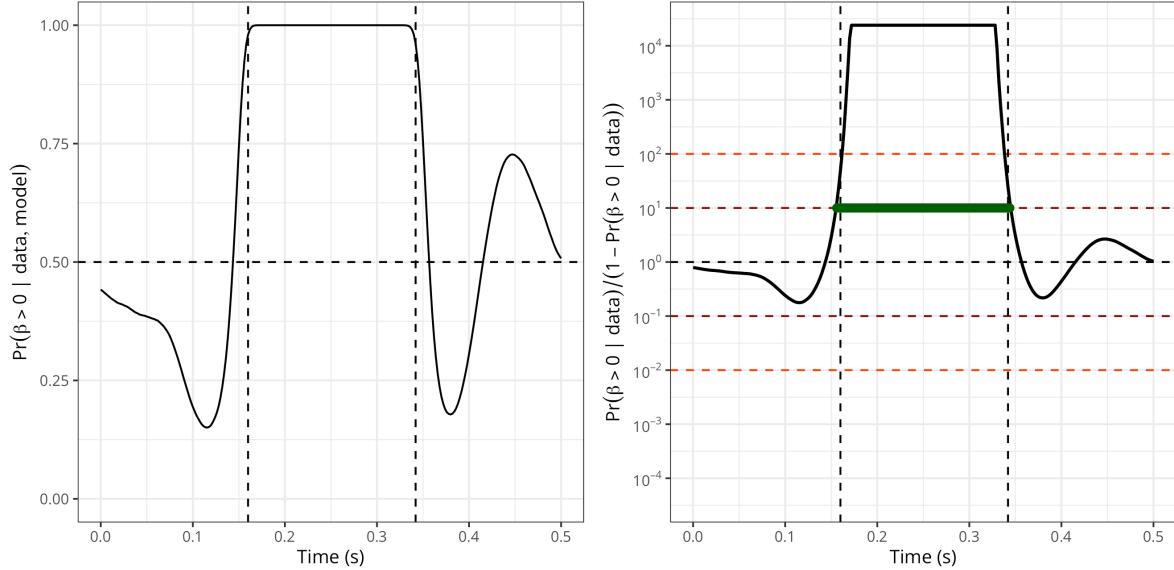


222 We then compute the posterior probability of the slope for **condition** being above 0
 223 (Figure 4, left). From this quantity, we compute the ratio of posterior probabilities (i.e., $p/(1 -$
 224 $p)$) and visualise the timecourse of this ratio superimposed with the conventional thresholds
 225 on evidence ratios (Figure 4, right). Note that a ratio of 10 means that the probability of
 226 the difference being above 0 is 10 times higher than the probability of the difference not being
 227 above 0, given the data, the priors, and other model's assumptions.¹ Thresholding the posterior
 228 probability ratio thus provides a model-based approach for estimating the onset and offset
 229 of M/EEG effects, whose properties will be assessed in the simulation study. An important
 230 advantage is that the proposed approach can be extended to virtually any model structure.

¹This posterior probability ratio, or posterior odds, is equivalent to a Bayes factor, assuming 1:1 prior odds.

Figure 4

Left: Posterior probability of the EEG difference (slope) being above 0 according to the BGAMM. Right: Ratio of posterior probability according to the BGAMM (on a log10 scale). Timesteps above threshold (10) are highlighted in green. NB: the minimum and maximum possible ratio values are determined (bounded) by the number of posterior samples in the model.

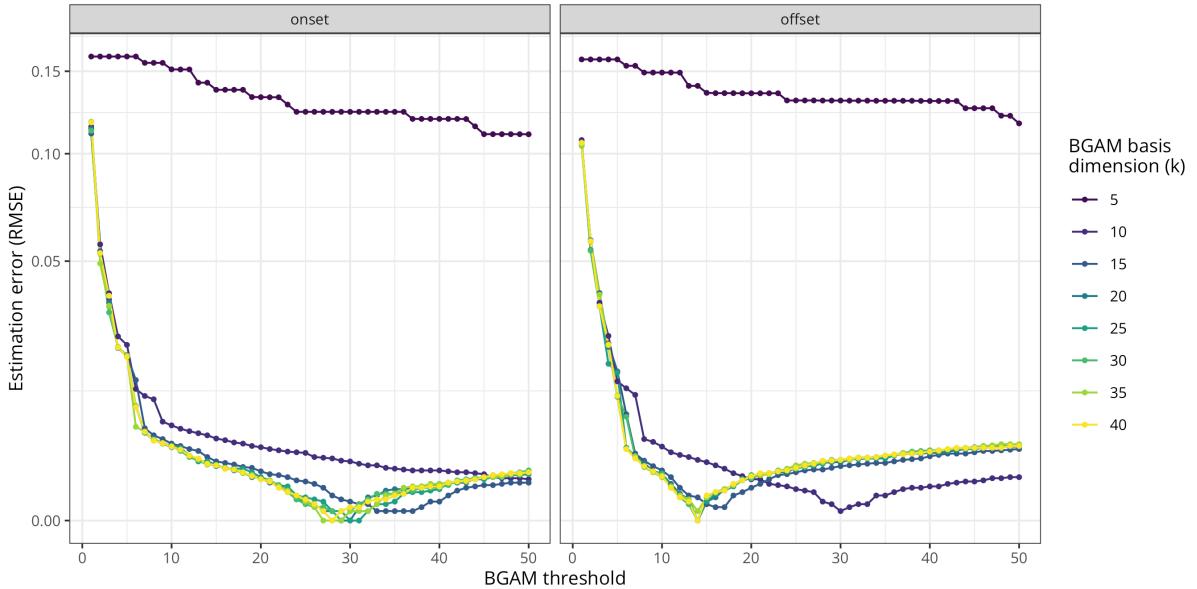


231 Error properties of the proposed approach

232 We then assessed the performance of the proposed approach by computing the differ-
 233 ence between the true and estimated onset/offset of the EEG difference according to various **k**
 234 (basis dimension) and **threshold** values. Remember that the EEG signal was generated from
 235 a truncated Gaussian with an objective onset at 160 ms, a maximum at 250 ms, and an offset
 236 at 342 ms. Figure 5 shows that the multilevel GAM can almost exactly recover the true onset
 237 and offset values, given some reasonable choice of **k** and **threshold** values. We provide more
 238 detailed recommendations on how to set **k** in Section C. This figure further reveals that the op-
 239 timal **k** and **threshold** values may differ for the onset and offset values, and there there seems
 240 to exist a trade-off between these two parameters: lower **k** values lead to poorer estimations, but
 241 these poor estimations can be compensated (only to some extent) by higher **threshold** values
 242 (and reciprocally).

Figure 5

Average estimation error (RMSE) for the onset (left) and offset (right) according to various basis dimension and threshold values for the BGAM (computed from 100 simulated datasets).

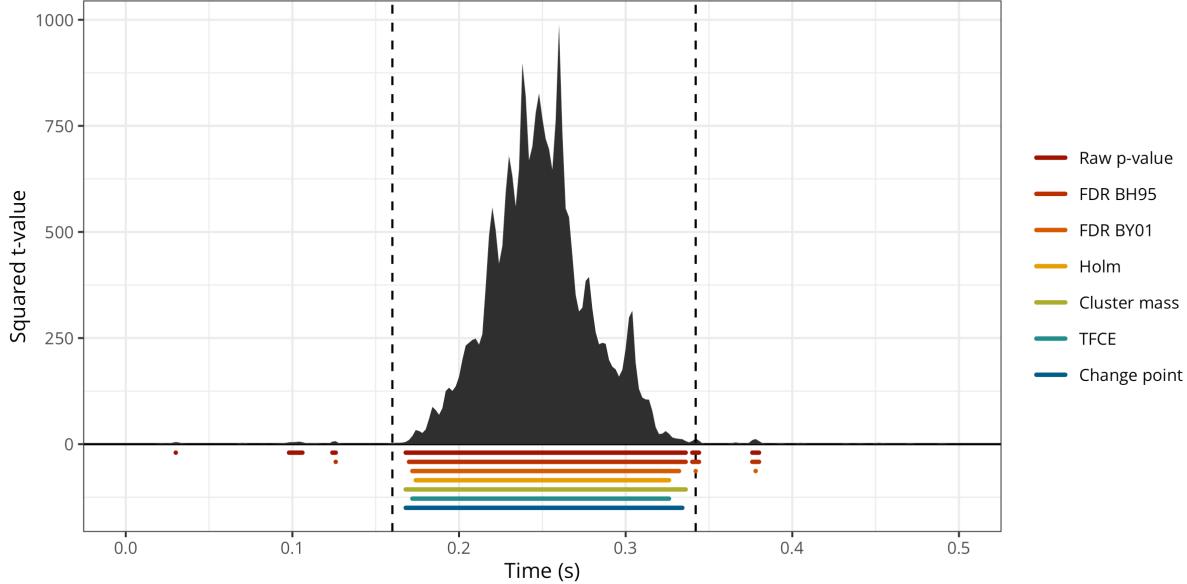


243 Comparing the onsets/offsets estimates across approaches

244 We then compared the ability of the BGAM to accurately estimate the onset and offset
 245 of the ERP difference to other widely-used methods. First, we conducted mass-univariate t-
 246 tests (thus treating each timestep independently) and identified the onset and offset of the ERP
 247 difference as the first and last values crossing an arbitrary significance threshold ($\alpha = 0.05$). We
 248 then followed the same approach but after applying different forms of multiplicity correction
 249 to the p -values. We compared two methods that control the FDR (i.e., BH95, Benjamini &
 250 Hochberg, 1995; and BY01, Benjamini & Yekutieli, 2001), one method that controls the FWER
 251 (i.e., Holm–Bonferroni method, Holm, 1979), and two cluster-based permutation methods (per-
 252 mutation with a single cluster-forming threshold and threshold-free cluster enhancement, TFCE,
 253 S. Smith & Nichols, 2009). The BH95, BY01, and Holm corrections were applied to the p-
 254 values using the `p.adjust()` function in R. The cluster-based inference was implemented using
 255 a cluster-sum statistic of squared t -values, as implemented in MNE-Python (Gramfort, 2013),
 256 called via the R package `reticulate` v 1.42.0 (Ushey et al., 2024). We also compared these
 257 estimates to the onset and offset as estimated using the binary segmentation algorithm, as im-
 258 plemented in the R package `changepoint` v 2.3 (Killick et al., 2022), and applied directly to
 259 the squared t -values (as in Rousselet, 2025). Figure 6 illustrates the onsets and offsets esti-
 260 mated by each method on a single simulated dataset and shows that all methods systematically
 261 overestimate the true onset and underestimate the true offset.

Figure 6

Exemplary timecourse of squared t-values with true onset and offset (vertical black dashed lines) and onsets/offsets identified using the raw p-values, the corrected p-values (BH95, BY01, Holm), the cluster-based methods (Cluster mass, TFCE), or using the binary segmentation method (Change point).



262 Simulation study

263 To assess the accuracy of group-level onset estimation, the various methods were com-
 264 pared using the bias (defined as the median difference between the estimated and true value
 265 of the onset/offset), median absolute error (MAE), root mean square error (RMSE), variance,
 266 and median absolute deviation (MAD) of onset/offset estimates computed on 10,000 simulated
 267 datasets. As in Rousselet (2025), each participant was assigned a random onset between 150
 268 and 170ms. Whereas the focus of the present article focuses is on one-dimensional signals (e.g.,
 269 one M/EEG channel), we provide a 2D application in Section A.

270 Application to actual MEG data

271 To complement the simulation study, we evaluated the performance of all methods on
 272 actual MEG data (decoding results from Nalborczyk et al., in preparation). In this study,
 273 we conducted time-resolved multivariate pattern analysis (MVPA, also known as decoding) of
 274 MEG data during reading tasks. As a result, we obtain a timecourse of decoding performance
 275 (ROC AUC), bounded between 0 and 1, for each participant ($N = 32$). Next, we wanted to test
 276 whether the group-level average decoding accuracy is above chance (i.e., 0.5) at each timestep
 277 (Figure 7). To achieve this, we fitted a BGAM as introduced previously, but we replaced the
 278 Normal likelihood function by a Beta one to account for the bounded nature of AUC values
 279 (between 0 and 1) (for a tutorial on Beta regression, see Coretta & Bürkner, 2025).

280 Note that although we chose a basis dimension of $k = 50$, which seems appropriate for
 281 the present data, this choice should be adapted according to the properties of the modelled
 282 data (e.g., signal-to-noise ratio, prior low-pass filtering, sampling rate) and should be assessed

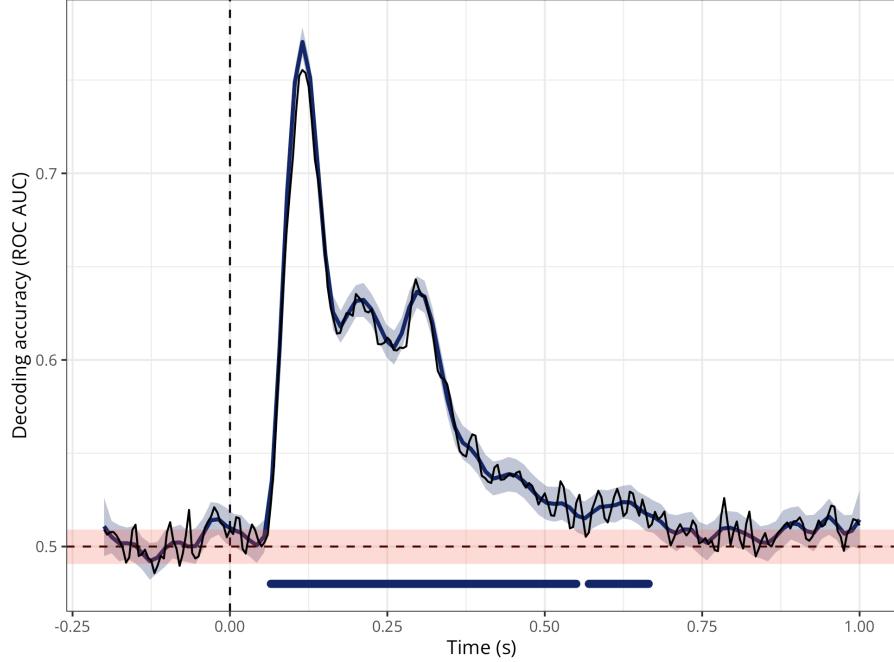
283 by the usual model checking tools (e.g., posterior predictive checks, see also Section C). To
 284 better distinguish signal from noise, we also defined a region of practical equivalence (ROPE,
 285 [Kruschke & Liddell, 2017](#)), defined as the chance level plus the standard deviation of the (group-
 286 level average) decoding performance during the baseline period.

```
# fitting the Beta GAM
meg_decoding_gam <- brm(
  auc ~ s(time, bs = "cr", k = 50),
  data = decoding_df,
  family = Beta(),
  warmup = 2000,
  iter = 5000,
  chains = 4,
  cores = 4
)
```

287 We assessed the reliability of the proposed approach using a form of permutation-based
 288 split-half reliability (as for instance in [Rosenblatt et al., 2018](#)), which consisted of the following
 289 steps. First, we created 1,000 split halves of the data (i.e., with half the participants in the
 290 original data, that is, 16 participants). For each split, we estimated the onset/offset using all
 291 methods described previously. Third, we summarised the distribution of onset/offset estimates
 292 using the median “error” (i.e., difference between the split estimate and the estimate obtained
 293 using the full dataset) and the variance across splits. This approach allows assessing how similar
 294 the estimate of each half split is to the full dataset (thus acting as a proxy for the population)
 295 and how variable the estimates are across split halves.

Figure 7

Group-level average decoding performance ($N=32$) superimposed with the GAM predictions (in blue) and the region of practical equivalence (ROPE, in orange) computed from the baseline period (data from Nalborczyk et al., in preparation). The blue horizontal markers indicate the timesteps at which the posterior probability ratio exceeds 20.



296

Results

297 This section is divided in two parts. First, we present the results from the simulation
 298 study, assessing the bias and variance of each method when applied to simulated data in which
 299 the ground truth is known. Second, we present the results obtained when applying the different
 300 methods to actual MEG data (decoding performance through time), assessing the reliability of
 301 the estimates provided by each method.

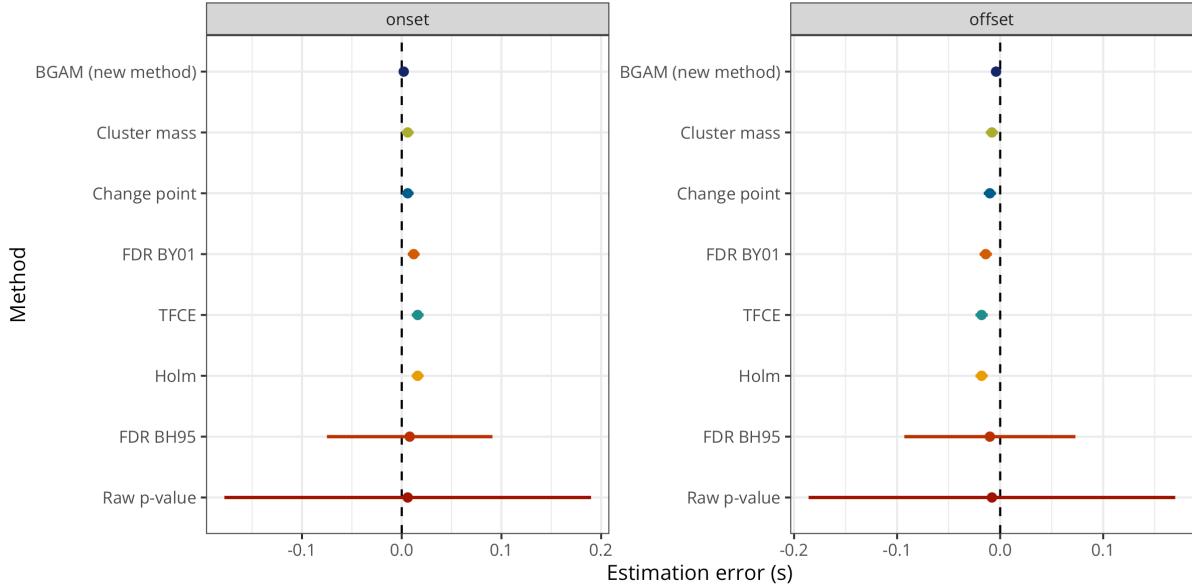
302 **Simulation study (bias and variance)**

303 Figure 8 shows a summary of the simulation results, revealing that the proposed ap-
 304 proach (BGAM) has the lowest median error for both the onset and offset estimates. The Cluster
 305 mass and Change point also have good performance, but surprisingly, the TFCE method has
 306 relatively bad performance for estimating the effect offset (similar performance to the Holm and
 307 FDR BY01 methods). Unsurprisingly, the FDR BH95 and Raw p-value methods show the worst
 308 performance.

309 Results are further summarised in Table 1, which shows that the BGAM is almost perfectly
 310 unbiased (i.e., it has a bias of approximately 2ms for the onset and 4ms for the offset). The
 311 Bias column shows that all methods tend to estimate the onset later than the true onset and
 312 to estimate the offset earlier than the true offset. As can be seen from this table, the BGAM has
 313 the best performance on all included metrics.

Figure 8

Median error and median absolute deviation of the error for the onset (left) and offset (right) estimates according to each method. Methods are ordered from lowest (top) to highest (bottom) median absolute error (separately for the onset and offset estimates).



314 Application to actual MEG data (reliability)

315 Figure 9 shows the group-level average decoding performance through time with onset
 316 and offset estimates from each method. Overall, this figure shows that both the **Raw p-value**
 317 and **FDR BH95** methods are extremely lenient, considering that the decoding performance is
 318 above chance before the onset of the stimulus (false positive) and until the end of the trial. The
 319 **Change point** and **Cluster mass** methods seem the most conservative methods, identifying
 320 a time window from approximately +60ms to +500ms. The **Holm**, **TFCE**, and **BGAM** methods
 321 produce similar estimates of onset and offset, ranging from approximately +60ms to +650ms,
 322 although the **BGAM** method seems to result in fewer clusters.²

323 Figure 10 shows the median difference between the onset and offset estimates from each
 324 data split and the onset and offset estimates from the full dataset (x-axis) along with the variance
 325 of its onset and offset estimates across data splits (error bar). This figure reveals that the **BGAM**
 326 *onset* and *offset* estimates on each split are the closest to the estimates from the full dataset on
 327 average (0ms difference for the onset estimate and 5ms difference for the offset estimate). The
 328 **Raw p-value** method has similar performance for the offset, but given the aberrant estimates
 329 it produces (cf. Figure 9), the fact that it is consistent between data splits and the full dataset
 330 is not convincing on its own. The **Change point** method also has a very good performance
 331 (i.e., very low difference between split estimates and full estimates), but produces too short
 332 cluster of significant decoding performance (cf. Figure 9).³ Overall, the figure reveals that
 333 for all other methods, split datasets produce later onset estimates and earlier offset estimates

²It should be noted that although each method can produce several “clusters” of timesteps, we only considered the first (onset) and last (offset) timesteps identified by each method to compute the estimation error.

³As in Rousselet (2025), we fixed the number of expected change points to two in the binary segmentation algorithm, thus producing always one cluster.

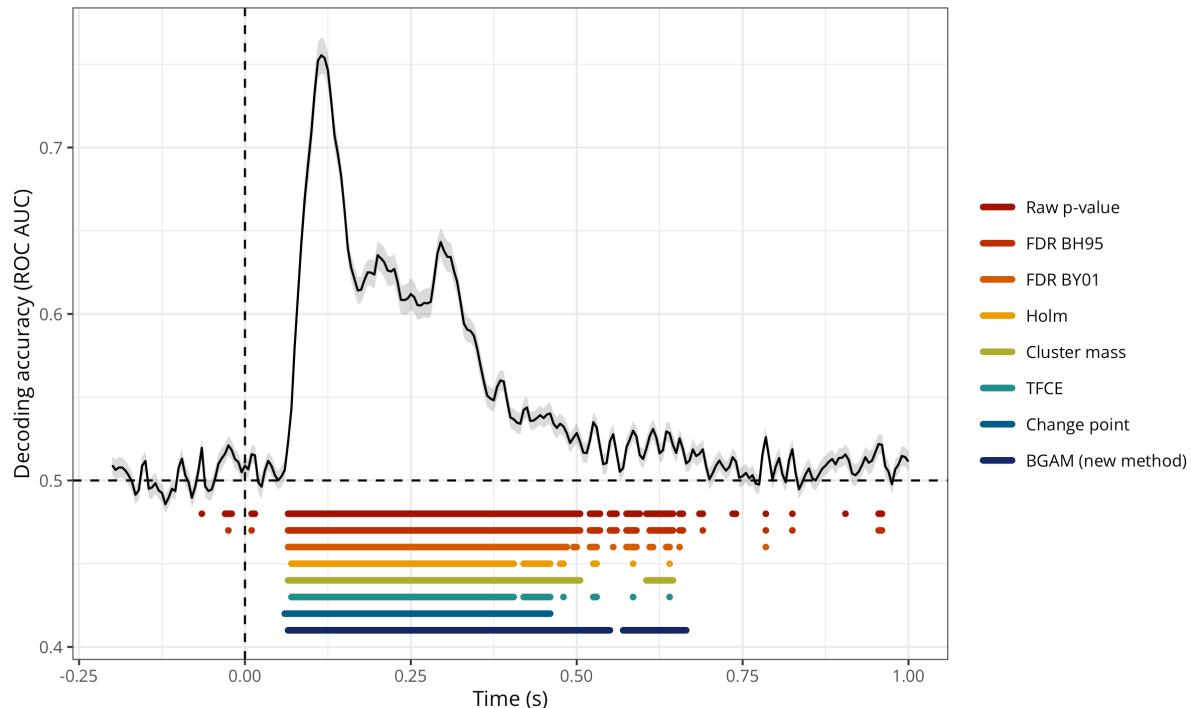
Table 1

Summary statistics of the onset and offset estimates for each method (in ms, ordered by the MAE) according to the main simulation study.

	Bias	MAE	RMSE	Variance	MAD
onset					
BGAM (new method)	2.00	2.00	1.03	43.83	2.97
Cluster mass	6.00	6.00	6.78	228.16	5.93
Change point	6.00	6.00	5.21	34.54	5.93
FDR BY01	12.00	14.00	35.44	7,659.08	5.93
TFCE	16.00	16.00	29.88	2,785.39	5.93
Holm	16.00	16.00	29.91	2,710.69	5.93
FDR BH95	8.00	48.00	57.50	18,986.98	83.03
Raw p-value	6.00	118.00	72.39	25,072.78	183.84
offset					
BGAM (new method)	-4.00	4.00	3.00	45.58	2.97
Cluster mass	-8.00	8.00	8.60	200.72	5.93
Change point	-10.00	10.00	9.22	34.58	5.93
FDR BY01	-14.00	16.00	37.88	7,708.00	5.93
TFCE	-18.00	18.00	32.15	2,822.35	5.93
Holm	-18.00	18.00	31.96	2,720.13	5.93
FDR BH95	-10.00	46.00	60.10	19,025.38	83.03
Raw p-value	-8.00	112.00	71.59	24,969.31	177.91

Figure 9

Group-level average decoding performance through time with onset and offset estimates for each method (data from Nalborczyk et al., in preparation).



334 (compared to the estimates from the model fitted on the full dataset). These results highlight
 335 some desirable properties for a method aiming to precisely and reliably estimate the onset and
 336 offset of M/EEG effects, namely, it should i) have good asymptotic properties on simulated data,

337 ii) provide sensible identified clusters in actual data, and iii) provide reliable/stable estimates
 338 across various partitions of the data. All these desiderata seem to be fulfilled by the BGAM
 339 approach.

Figure 10

Median error and median absolute deviation of the error for the onset (left) and offset (right) estimates according to each method. Methods are ordered from smallest (top) to largest (bottom) median absolute error separately for the onset and offset estimates.

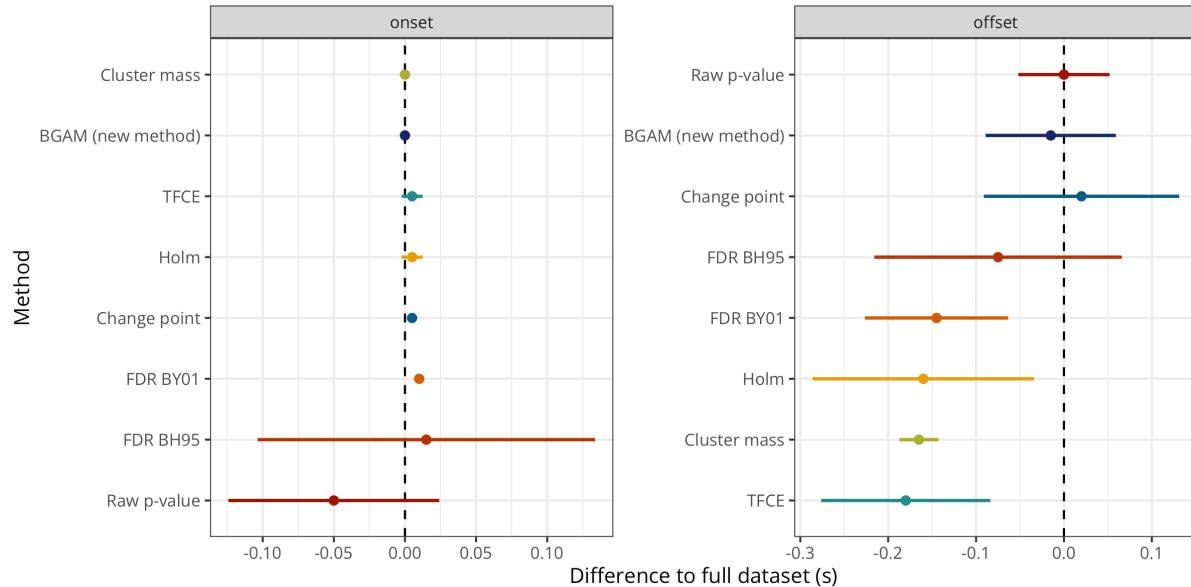


Table 2

Summary statistics of the onset and offset estimates for each method (in ms, ordered by the MAE) according to the reliability simulation study.

	Bias	MAE	RMSE	Variance	MAD
onset					
BGAM (new method)	0.00	0.00	6.91	829.66	0.00
Cluster mass	0.00	0.00	2.07	8.33	0.00
TFCE	5.00	5.00	10.67	101.05	7.41
Holm	5.00	5.00	10.04	96.64	7.41
Change point	5.00	5.00	1.74	96.94	7.41
FDR BY01	10.00	10.00	3.71	1,581.49	0.00
FDR BH95	10.00	90.00	6.30	6,970.82	118.61
Raw p-value	-40.00	60.00	27.50	5,545.80	74.13
offset					
Raw p-value	0.00	35.00	39.65	7,386.26	51.89
BGAM (new method)	-15.00	60.00	38.07	23,511.47	66.72
Change point	20.00	60.00	54.11	8,354.85	81.54
FDR BH95	-75.00	75.00	111.46	15,361.69	140.85
FDR BY01	-145.00	150.00	122.62	17,421.16	88.96
Holm	-160.00	160.00	123.00	13,560.13	126.02
Cluster mass	-165.00	165.00	150.79	4,272.92	29.65
TFCE	-180.00	180.00	131.70	13,720.28	96.37

340

Discussion

341 Summary of the proposed approach

342 Overall, before concluding on the onset/offset of effect based on the model, we need to
 343 ensure that the model provides a faithful description of the data-generating process (e.g., via
 344 posterior predictive checks etc)...

345 The TFCE performs worse than the cluster-sum approach, which was anticipated by
 346 Rousselet (2025) based on the initial results of S. Smith & Nichols (2009)... we did not include
 347 the cluster-depth algorithm (Frossard & Renaud, 2022), as Rousselet (2025) already showed its
 348 performance were worse than the cluster mass algorithm...

349 Limitations and future directions

350 As in previous simulation work (e.g., Rousselet et al., 2008; Sassenhagen & Draschkow,
 351 2019), the present simulation results depend on various choices such as the specific cluster-
 352 forming algorithm and threshold, signal-to-noise ratio, negative impact of preprocessing steps
 353 (e.g., low-pass filter) on temporal resolution... note however, that the same caveats apply to all
 354 methods...

355 Discussion about the nature of the model: we modelled the surface M/EEG signals,
 356 however, the true interests (probably) lie in the brain, that is, in the source space... we could
 357 build a “full” Bayesian model of the generated EEG signal (i.e., including hypotheses about the
 358 source and a forward model), but this model would become computationally heavier...

359 The error properties depend on the threshold parameter, a value of 10 or 20 seems to
 360 be a reasonable default, but the optimal threshold parameter can be adjusted using split-half
 361 reliability assessment... also depends on k...

362 Can be applied to any 1D timeseries (e.g., pupillometry, electromyography)... Extending

363 the approach to spatiotemporal data (i.e., time + sensors) or spatiotemporal time-frequency 4D

364 data...

365 We kept the exemplary models simple, but can be extended by adding varying/random

366 effects (intercept and slope) for item (e.g., word)... but also continuous predictors at the trial

367 level, overlapping factors, etc...

368

Data and code availability

369

The simulation results as well as the R code to reproduce the simulations are available on GitHub: https://github.com/lNALBORCZYK/brms_meeg. The `neurogam` R package is available at <https://github.com/lNALBORCZYK/neurogam>.

372

Packages

373

We used R version 4.4.3 ([R Core Team, 2025](#)) and the following R packages: assertthat v. 0.2.1 ([Wickham, 2019](#)), bamlss v. 1.2.5 ([Muschinski et al., 2024; Umlauf et al., 2018b, 2021](#)), brms v. 2.22.0 ([Bürkner, 2017, 2018, 2021](#)), changepoint v. 2.3 ([Killick et al., 2024; Killick & Eckley, 2014](#)), doParallel v. 1.0.17 ([Corporation & Weston, 2022](#)), easystats v. 0.7.4 ([Lüdecke et al., 2022](#)), foreach v. 1.5.2 ([Microsoft & Weston, 2022](#)), furrr v. 0.3.1 ([Vaughan & Dancho, 2022](#)), future v. 1.49.0 ([Bengtsson, 2021](#)), ggrepel v. 0.9.6 ([Slowikowski, 2024](#)), glue v. 1.8.0 ([Hester & Bryan, 2024](#)), grateful v. 0.2.12 ([Rodriguez-Sanchez & Jackson, 2024](#)), gt v. 1.0.0 ([Iannone et al., 2025](#)), knitr v. 1.50 ([Xie, 2014, 2015, 2025](#)), MetBrewer v. 0.2.0 ([Mills, 2022](#)), mgcv v. 1.9.3 ([Wood, 2003b, 2004, 2011, 2017c; Wood et al., 2016](#)), neurogam v. 0.0.1 ([Nalborczyk, 2025](#)), pakret v. 0.2.2 ([Gallou, 2024](#)), patchwork v. 1.3.0 ([T. L. Pedersen, 2024](#)), rmarkdown v. 2.29 ([Allaire et al., 2024; Xie et al., 2018, 2020](#)), scales v. 1.4.0 ([Wickham et al., 2025](#)), scico v. 1.5.0 ([T. L. Pedersen & Cramer, 2023](#)), tictoc v. 1.2.1 ([Izrailev, 2024](#)), tidybayes v. 3.0.7 ([Kay, 2024](#)), tidytext v. 0.4.2 ([Silge & Robinson, 2016](#)), tidyverse v. 2.0.0 ([Wickham et al., 2019](#)).

387

Acknowledgements

388

Centre de Calcul Intensif d’Aix-Marseille is acknowledged for granting access to its high performance computing resources.

390

References

- 391 Abugaber, D., Finestrat, I., Luque, A., & Morgan-Short, K. (2023). Generalized additive mixed
 392 modeling of EEG supports dual-route accounts of morphosyntax in suggesting no word
 393 frequency effects on processing of regular grammatical forms. *Journal of Neurolinguistics*,
 394 67, 101137. <https://doi.org/10.1016/j.jneuroling.2023.101137>
- 395 Allaire, J., Xie, Y., Dervieux, C., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham,
 396 H., Cheng, J., Chang, W., & Iannone, R. (2024). *rmarkdown: Dynamic documents for r*.
 397 <https://github.com/rstudio/rmarkdown>
- 398 Baayen, R. H., & Linke, M. (2020). *Generalized Additive Mixed Models* (pp. 563–591). Springer
 399 International Publishing. https://doi.org/10.1007/978-3-030-46216-1_23
- 400 Baayen, R. H., Rij, J. van, Cat, C. de, & Wood, S. (2018). *Autocorrelated errors in ex-
 401 perimental data in the language sciences: Some solutions offered by generalized additive
 402 mixed models* (pp. 49–69). Springer International Publishing. [https://doi.org/10.1007/978-3-319-69830-4_4](https://doi.org/10.1007/

 403 978-3-319-69830-4_4)
- 404 Bengtsson, H. (2021). A unifying framework for parallel and distributed processing in r using
 405 futures. *The R Journal*, 13(2), 208–227. <https://doi.org/10.32614/RJ-2021-048>
- 406 Benjamini, Y., & Hochberg, Y. (1995). Controlling the False Discovery Rate: A Practical and
 407 Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society Series B:
 408 Statistical Methodology*, 57(1), 289–300. <https://doi.org/10.1111/j.2517-6161.1995.tb02031.x>
- 409 x
- 410 Benjamini, Y., & Yekutieli, D. (2001). The control of the false discovery rate in multiple
 411 testing under dependency. *The Annals of Statistics*, 29(4). <https://doi.org/10.1214/aos/1013699998>
- 413 Bullmore, E. T., Suckling, J., Overmeyer, S., Rabe-Hesketh, S., Taylor, E., & Brammer, M. J.
 414 (1999). Global, voxel, and cluster tests, by theory and permutation, for a difference between
 415 two groups of structural MR images of the brain. *IEEE Transactions on Medical Imaging*,
 416 18(1), 32–42. <https://doi.org/10.1109/42.750253>
- 417 Bürkner, P.-C. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal
 418 of Statistical Software*, 80(1), 1–28. <https://doi.org/10.18637/jss.v080.i01>
- 419 Bürkner, P.-C. (2018). Advanced Bayesian multilevel modeling with the R package brms. *The
 420 R Journal*, 10(1), 395–411. <https://doi.org/10.32614/RJ-2018-017>
- 421 Bürkner, P.-C. (2021). Bayesian item response modeling in R with brms and Stan. *Journal of
 422 Statistical Software*, 100(5), 1–54. <https://doi.org/10.18637/jss.v100.i05>
- 423 Coretta, S., & Bürkner, P.-. C. (2025). *Bayesian beta regressions with brms in r: A tutorial for
 424 phoneticians*. http://dx.doi.org/10.31219/osf.io/f9rqg_v1
- 425 Corporation, M., & Weston, S. (2022). doParallel: Foreach parallel adaptor for the “parallel”
 426 package. <https://CRAN.R-project.org/package=doParallel>
- 427 Delorme, A., & Makeig, S. (2004). EEGLAB: an open source toolbox for analysis of single-
 428 trial EEG dynamics including independent component analysis. *Journal of Neuroscience
 429 Methods*, 134(1), 9–21. <https://doi.org/10.1016/j.jneumeth.2003.10.009>
- 430 Dinga, R., Fraza, C. J., Bayer, J. M. M., Kia, S. M., Beckmann, C. F., & Marquand, A.
 431 F. (2021). Normative modeling of neuroimaging data using generalized additive models of

- 432 *location scale and shape.* <http://dx.doi.org/10.1101/2021.06.14.448106>
- 433 Dunagan, D., Jordan, T., Hale, J. T., Pylkkänen, L., & Chacón, D. A. (2025). Evaluating the
434 timecourses of morpho-orthographic, lexical, and grammatical processing following rapid
435 parallel visual presentation: An EEG investigation in English. *Cognition*, 257, 106080.
436 <https://doi.org/10.1016/j.cognition.2025.106080>
- 437 Dunn, O. J. (1961). Multiple Comparisons among Means. *Journal of the American Statistical
438 Association*, 56(293), 52–64. <https://doi.org/10.1080/01621459.1961.10482090>
- 439 Ehinger, B. V., & Dimigen, O. (2019). Unfold: An integrated toolbox for overlap correction,
440 non-linear modeling, and regression-based EEG analysis. *PeerJ*, 7, e7838. <https://doi.org/10.7717/peerj.7838>
- 442 Frossard, J., & Renaud, O. (2022). The cluster depth tests: Toward point-wise strong control
443 of the family-wise error rate in massively univariate tests with application to M/EEG.
444 *NeuroImage*, 247, 118824. <https://doi.org/10.1016/j.neuroimage.2021.118824>
- 445 Gabry, J., Simpson, D., Vehtari, A., Betancourt, M., & Gelman, A. (2019). Visualization in
446 Bayesian workflow. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*,
447 182(2), 389–402. <https://doi.org/10.1111/rssa.12378>
- 448 Gallou, A. (2024). *pakret: Cite “R” packages on the fly in “R Markdown” and “Quarto”*. <https://CRAN.R-project.org/package=pakret>
- 450 Gelman, A., Vehtari, A., Simpson, D., Margossian, C. C., Carpenter, B., Yao, Y., Kennedy,
451 L., Gabry, J., Bürkner, P.-C., & Modrák, M. (2020). Bayesian workflow. *arXiv:2011.01808
/Stat*. <http://arxiv.org/abs/2011.01808>
- 453 Gramfort, A. (2013). MEG and EEG data analysis with MNE-python. *Frontiers in Neuro-
454 science*, 7. <https://doi.org/10.3389/fnins.2013.00267>
- 455 Hastie, T. J., & Tibshirani, R. J. (2017). *Generalized Additive Models*. Routledge. <https://doi.org/10.1201/9780203753781>
- 457 Hester, J., & Bryan, J. (2024). *glue: Interpreted string literals*. <https://CRAN.R-project.org/package=glue>
- 459 Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal
460 of Statistics*, 6(2), 65–70. <http://www.jstor.org/stable/4615733>
- 461 Iannone, R., Cheng, J., Schloerke, B., Hughes, E., Lauer, A., Seo, J., Brevoort, K., & Roy, O.
462 (2025). *gt: Easily create presentation-ready display tables*. <https://CRAN.R-project.org/package=gt>
- 464 Izrailev, S. (2024). *tictoc: Functions for timing r scripts, as well as implementations of “Stack”
465 and “StackList” structures*. <https://CRAN.R-project.org/package=tictoc>
- 466 Kay, M. (2024). *tidybayes: Tidy data and geoms for Bayesian models*. <https://doi.org/10.5281/zenodo.1308151>
- 468 Killick, R., & Eckley, I. A. (2014). changepoint: An R package for changepoint analysis. *Journal
469 of Statistical Software*, 58(3), 1–19. <https://www.jstatsoft.org/article/view/v058i03>
- 470 Killick, R., Haynes, K., & Eckley, I. A. (2022). *changepoint: An R package for changepoint
471 analysis*. <https://CRAN.R-project.org/package=changepoint>
- 472 Killick, R., Haynes, K., & Eckley, I. A. (2024). *changepoint: An R package for changepoint
473 analysis*. <https://CRAN.R-project.org/package=changepoint>

- 474 King, J.-R., & Dehaene, S. (2014). Characterizing the dynamics of mental representations:
475 the temporal generalization method. *Trends in Cognitive Sciences*, 18(4), 203–210. <https://doi.org/10.1016/j.tics.2014.01.002>
- 477 Kruschke, J. K., & Liddell, T. M. (2017). The Bayesian New Statistics: Hypothesis testing,
478 estimation, meta-analysis, and power analysis from a Bayesian perspective. *Psychonomic
479 Bulletin & Review*, 25(1), 178–206. <https://doi.org/10.3758/s13423-016-1221-4>
- 480 Lüdecke, D., Ben-Shachar, M. S., Patil, I., Wiernik, B. M., Bacher, E., Thériault, R., &
481 Makowski, D. (2022). easystats: Framework for easy statistical modeling, visualization,
482 and reporting. CRAN. <https://doi.org/10.32614/CRAN.package.easystats>
- 483 Maris, E. (2011). Statistical testing in electrophysiological studies. *Psychophysiology*, 49(4),
484 549–565. <https://doi.org/10.1111/j.1469-8986.2011.01320.x>
- 485 Maris, E., & Oostenveld, R. (2007). Nonparametric statistical testing of EEG- and MEG-data.
486 *Journal of Neuroscience Methods*, 164(1), 177–190. <https://doi.org/10.1016/j.jneumeth.2007.03.024>
- 488 Meulman, N., Sprenger, S. A., Schmid, M. S., & Wieling, M. (2023). GAM-based individual
489 difference measures for L2 ERP studies. *Research Methods in Applied Linguistics*, 2(3),
490 100079. <https://doi.org/10.1016/j.rmal.2023.100079>
- 491 Meulman, N., Wieling, M., Sprenger, S. A., Stowe, L. A., & Schmid, M. S. (2015). Age Effects
492 in L2 Grammar Processing as Revealed by ERPs and How (Not) to Study Them. *PLOS
493 ONE*, 10(12), e0143328. <https://doi.org/10.1371/journal.pone.0143328>
- 494 Microsoft, & Weston, S. (2022). *foreach*: Provides foreach looping construct. <https://CRAN.R-project.org/package=foreach>
- 496 Miller, D. L. (2025). Bayesian views of generalized additive modelling. *Methods in Ecology and
497 Evolution*. <https://doi.org/10.1111/2041-210x.14498>
- 498 Mills, B. R. (2022). *MetBrewer*: Color palettes inspired by works at the metropolitan museum
499 of art. <https://CRAN.R-project.org/package=MetBrewer>
- 500 Muschinski, T., Mayr, G. J., Simon, T., Umlauf, N., & Zeileis, A. (2024). Cholesky-based
501 multivariate Gaussian regression. *Econometrics and Statistics*, 29, 261–281. <https://doi.org/10.1016/j.ecosta.2022.03.001>
- 503 Nalborczyk, L. (2025). *neurogam*: Precise temporal localisation of m/EEG effects with bayesian
504 generalised additive multilevel models. <https://github.com/lnalborczyk/neurogam>
- 505 Nalborczyk, L., Batailler, C., Lœvenbruck, H., Vilain, A., & Bürkner, P.-C. (2019). An In-
506 troduction to Bayesian Multilevel Models Using brms: A Case Study of Gender Effects
507 on Vowel Variability in Standard Indonesian. *Journal of Speech, Language, and Hearing
508 Research*, 62(5), 1225–1242. https://doi.org/10.1044/2018_jslhr-s-18-0006
- 509 Nalborczyk, L., Hauw, F., Torcy, H. de, Dehaene, S., & Cohen, L. (in preparation). *Neural and
510 representational dynamics of tickertape synesthesia*.
- 511 Pedersen, E. J., Miller, D. L., Simpson, G. L., & Ross, N. (2019). Hierarchical generalized
512 additive models in ecology: An introduction with mgcv. *PeerJ*, 7, e6876. <https://doi.org/10.7717/peerj.6876>
- 514 Pedersen, T. L. (2024). *patchwork*: The composer of plots. <https://CRAN.R-project.org/package=patchwork>

- 516 Pedersen, T. L., & Crameri, F. (2023). *scico: Colour palettes based on the scientific colour-maps*.
517 <https://CRAN.R-project.org/package=scico>
- 518 Pernet, C. R., Chauveau, N., Gaspar, C., & Rousselet, G. A. (2011). LIMO EEG: A Tool-
519 box for Hierarchical LInear MOdeling of ElectroEncephaloGraphic Data. *Computational*
520 *Intelligence and Neuroscience*, 2011, 1–11. <https://doi.org/10.1155/2011/831409>
- 521 Pernet, C. R., Latinus, M., Nichols, T. E., & Rousselet, G. A. (2015). Cluster-based com-
522 putational methods for mass univariate analyses of event-related brain potentials/fields: A
523 simulation study. *Journal of Neuroscience Methods*, 250, 85–93. <https://doi.org/10.1016/j.jneumeth.2014.08.003>
- 525 R Core Team. (2025). *R: A language and environment for statistical computing*. R Foundation
526 for Statistical Computing. <https://www.R-project.org/>
- 527 Rasmussen, C. E., & Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning*.
528 <https://doi.org/10.7551/mitpress/3206.001.0001>
- 529 Rigby, R. A., & Stasinopoulos, D. M. (2005). Generalized Additive Models for Location, Scale
530 and Shape. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 54(3),
531 507–554. <https://doi.org/10.1111/j.1467-9876.2005.00510.x>
- 532 Rij, J. van, Hendriks, P., Rijn, H. van, Baayen, R. H., & Wood, S. N. (2019). Analyzing
533 the Time Course of Pupillometric Data. *Trends in Hearing*, 23. <https://doi.org/10.1177/2331216519832483>
- 535 Riutort-Mayol, G., Bürkner, P.-C., Andersen, M. R., Solin, A., & Vehtari, A. (2023). Practical
536 Hilbert space approximate Bayesian Gaussian processes for probabilistic programming.
537 *Statistics and Computing*, 33(1), 17. <https://doi.org/10.1007/s11222-022-10167-2>
- 538 Rodriguez-Sanchez, F., & Jackson, C. P. (2024). *grateful: Facilitate citation of R packages*.
539 <https://pakillo.github.io/grateful/>
- 540 Rosenblatt, J. D., Finos, L., Weeda, W. D., Solari, A., & Goeman, J. J. (2018). All-
541 Resolutions Inference for brain imaging. *NeuroImage*, 181, 786–796. <https://doi.org/10.1016/j.neuroimage.2018.07.060>
- 543 Rousselet, G. A. (2025). Using cluster-based permutation tests to estimate MEG/EEG onsets:
544 How bad is it? *European Journal of Neuroscience*, 61(1), e16618. <https://doi.org/10.1111/ejn.16618>
- 546 Rousselet, G. A., Pernet, C. R., Bennett, P. J., & Sekuler, A. B. (2008). Parametric study
547 of EEG sensitivity to phase noise during face processing. *BMC Neuroscience*, 9(1). <https://doi.org/10.1186/1471-2202-9-98>
- 549 Sassenhagen, J., & Draschkow, D. (2019). Cluster-based permutation tests of MEG/EEG data
550 do not establish significance of effect latency or location. *Psychophysiology*, 56(6). <https://doi.org/10.1111/psyp.13335>
- 552 Silge, J., & Robinson, D. (2016). tidytext: Text mining and analysis using tidy data principles
553 in r. *JOSS*, 1(3). <https://doi.org/10.21105/joss.00037>
- 554 Skukies, R., & Ehinger, B. (2021). Modelling event duration and overlap during EEG analysis.
555 *Journal of Vision*, 21(9), 2037. <https://doi.org/10.1167/jov.21.9.2037>
- 556 Skukies, R., Schepers, J., & Ehinger, B. (2024, December 9). *Brain responses vary in duration*
557 - modeling strategies and challenges. <https://doi.org/10.1101/2024.12.05.626938>

- 558 Slowikowski, K. (2024). *ggrepel: Automatically position non-overlapping text labels with “gg-*
559 *plot2”*. <https://CRAN.R-project.org/package=ggrepel>
- 560 Smith, N. J., & Kutas, M. (2014a). Regression-based estimation of ERP waveforms: I. The
561 rERP framework. *Psychophysiology*, 52(2), 157–168. <https://doi.org/10.1111/psyp.12317>
- 562 Smith, N. J., & Kutas, M. (2014b). Regression-based estimation of ERP waveforms: II. Non-
563 linear effects, overlap correction, and practical considerations. *Psychophysiology*, 52(2),
564 169–181. <https://doi.org/10.1111/psyp.12320>
- 565 Smith, S., & Nichols, T. (2009). Threshold-free cluster enhancement: Addressing problems of
566 smoothing, threshold dependence and localisation in cluster inference. *NeuroImage*, 44(1),
567 83–98. <https://doi.org/10.1016/j.neuroimage.2008.03.061>
- 568 Sóskuthy, M. (2021). Evaluating generalised additive mixed modelling strategies for dynamic
569 speech analysis. *Journal of Phonetics*, 84, 101017. <https://doi.org/10.1016/j.wocn.2020.101017>
- 570 Tremblay, A., & Newman, A. J. (2014). Modeling nonlinear relationships in ERP data using
571 mixed-effects regression with R examples. *Psychophysiology*, 52(1), 124–139. <https://doi.org/10.1111/psyp.12299>
- 572 Umlauf, N., Klein, N., Simon, T., & Zeileis, A. (2021). bamlss: A Lego toolbox for flexible
573 Bayesian regression (and beyond). *Journal of Statistical Software*, 100(4), 1–53. <https://doi.org/10.18637/jss.v100.i04>
- 574 Umlauf, N., Klein, N., & Zeileis, A. (2018a). BAMLSS: Bayesian Additive Models for Location,
575 Scale, and Shape (and Beyond). *Journal of Computational and Graphical Statistics*, 27(3),
576 612–627. <https://doi.org/10.1080/10618600.2017.1407325>
- 577 Umlauf, N., Klein, N., & Zeileis, A. (2018b). BAMLSS: Bayesian additive models for location,
578 scale and shape (and beyond). *Journal of Computational and Graphical Statistics*, 27(3),
579 612–627. <https://doi.org/10.1080/10618600.2017.1407325>
- 580 Ushey, K., Allaire, J., & Tang, Y. (2024). *Reticulate: Interface to ‘python’*. <https://CRAN.R-project.org/package=reticulate>
- 581 Vaughan, D., & Dancho, M. (2022). *furrr: Apply mapping functions in parallel using futures*.
582 <https://CRAN.R-project.org/package=furrr>
- 583 Wickham, H. (2019). *assertthat: Easy pre and post assertions*. <https://CRAN.R-project.org/package=assertthat>
- 584 Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund,
585 G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M.,
586 Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., ... Yutani, H. (2019). Welcome
587 to the tidyverse. *Journal of Open Source Software*, 4(43), 1686. <https://doi.org/10.21105/joss.01686>
- 588 Wickham, H., Pedersen, T. L., & Seidel, D. (2025). *scales: Scale functions for visualization*.
589 <https://CRAN.R-project.org/package=scales>
- 590 Wieling, M. (2018). Analyzing dynamic phonetic data using generalized additive mixed model-
591 ing: A tutorial focusing on articulatory differences between L1 and L2 speakers of English.
592 *Journal of Phonetics*, 70, 86–116. <https://doi.org/10.1016/j.wocn.2018.03.002>
- 593 Wood, S. N. (2003a). Thin Plate Regression Splines. *Journal of the Royal Statistical Society*

- 600 *Series B: Statistical Methodology*, 65(1), 95–114. <https://doi.org/10.1111/1467-9868.00374>
- 601 Wood, S. N. (2003b). Thin-plate regression splines. *Journal of the Royal Statistical Society (B)*,
602 65(1), 95–114. <https://doi.org/10.1111/1467-9868.00374>
- 603 Wood, S. N. (2004). Stable and efficient multiple smoothing parameter estimation for gener-
604 alized additive models. *Journal of the American Statistical Association*, 99(467), 673–686.
605 <https://doi.org/10.1198/016214504000000980>
- 606 Wood, S. N. (2011). Fast stable restricted maximum likelihood and marginal likelihood esti-
607 mation of semiparametric generalized linear models. *Journal of the Royal Statistical Society
(B)*, 73(1), 3–36. <https://doi.org/10.1111/j.1467-9868.2010.00749.x>
- 609 Wood, S. N. (2017a). *Generalized Additive Models*. Chapman; Hall/CRC. <https://doi.org/10.1201/9781315370279>
- 611 Wood, S. N. (2017b). *Generalized additive models: An introduction with r* (2nd ed.). Chapman;
612 Hall/CRC.
- 613 Wood, S. N. (2017c). *Generalized Additive Models: An introduction with R* (2nd ed.). Chapman;
614 Hall/CRC.
- 615 Wood, S. N., Pya, N., & Säfken, B. (2016). Smoothing parameter and model selection for
616 general smooth models (with discussion). *Journal of the American Statistical Association*,
617 111, 1548–1575. <https://doi.org/10.1080/01621459.2016.1180986>
- 618 Wüllhorst, V., Wüllhorst, R., Overmeyer, R., & Endrass, T. (2025). Comprehensive Analysis
619 of Event-Related Potentials of Response Inhibition: The Role of Negative Urgency and
620 Compulsivity. *Psychophysiology*, 62(2). <https://doi.org/10.1111/psyp.70000>
- 621 Xie, Y. (2014). knitr: A comprehensive tool for reproducible research in R. In V. Stodden, F.
622 Leisch, & R. D. Peng (Eds.), *Implementing reproducible computational research*. Chapman;
623 Hall/CRC.
- 624 Xie, Y. (2015). *Dynamic documents with R and knitr* (2nd ed.). Chapman; Hall/CRC. <https://yihui.org/knitr/>
- 626 Xie, Y. (2025). knitr: A general-purpose package for dynamic report generation in R. <https://yihui.org/knitr/>
- 628 Xie, Y., Allaire, J. J., & Grolemund, G. (2018). *R markdown: The definitive guide*. Chapman;
629 Hall/CRC. <https://bookdown.org/yihui/rmarkdown>
- 630 Xie, Y., Dervieux, C., & Riederer, E. (2020). *R markdown cookbook*. Chapman; Hall/CRC.
631 <https://bookdown.org/yihui/rmarkdown-cookbook>
- 632 Yeung, N., Bogacz, R., Holroyd, C. B., & Cohen, J. D. (2004). Detection of synchronized os-
633 cillations in the electroencephalogram: An evaluation of methods. *Psychophysiology*, 41(6),
634 822–832. <https://doi.org/10.1111/j.1469-8986.2004.00239.x>

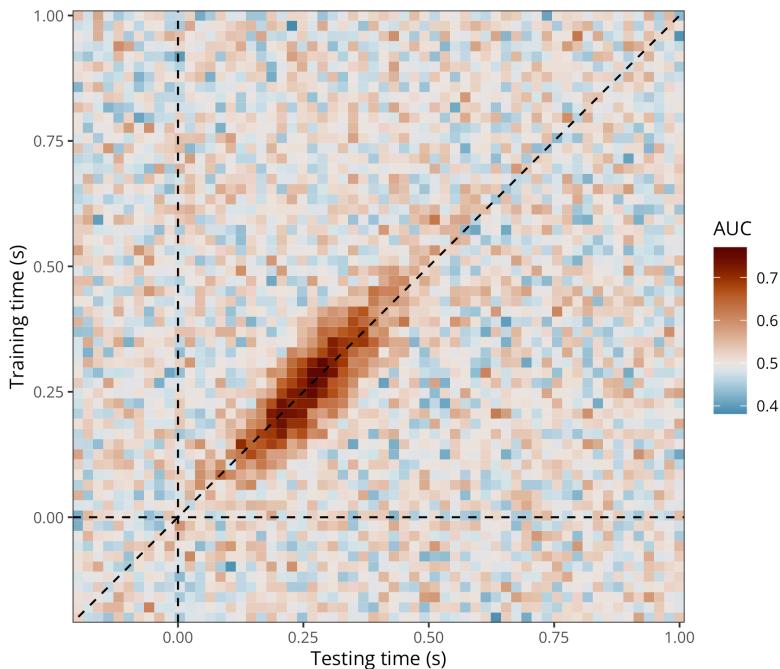
Appendix A

Application to 2D time-resolved decoding results (cross-temporal generalisation)

635 Assume we have M/EEG data and we have conducted cross-temporal generalisation analyses
 636 ([King & Dehaene, 2014](#)). As a result, we have a 2D matrix where each element contains the
 637 decoding accuracy (e.g., ROC AUC) of a classifier trained at timestep training_i and tested at
 638 timestep testing_j (Figure A1).

Figure A1

Exemplary (simulated) group-level average cross-temporal generalisation matrix of decoding performance (ROC AUC).



639 To model cross-temporal generalisation matrices of decoding performance (ROC AUC),
 640 we extended the initial (decoding) GAM to take into account the bivariate temporal distribution
 641 of AUC values, thus producing naturally smoothed estimates (timecourses) of AUC values and
 642 posterior probabilities. This model can be written as follows:

$$\begin{aligned} \text{AUC}_i &\sim \text{Beta}(\mu_i, \phi) \\ g(\mu_i) &= f(\text{train}_i, \text{test}_i) \end{aligned}$$

643 where we assume that AUC values come from a Beta distribution with two parameters
 644 μ and ϕ . We can think of $f(\text{train}_i, \text{test}_i)$ as a surface (a smooth function of two variables) that
 645 we can model using a 2-dimensional splines. Let $\mathbf{s}_i = (\text{train}_i, \text{test}_i)$ be some pair of training and
 646 testing samples, and let $\mathbf{k}_m = (\text{train}_m, \text{test}_m)$ denote the m^{th} knot in the domain of train_i and
 647 test_i . We can then express the smooth function as:

$$f(\text{train}_i, \text{test}_i) = \alpha + \sum_{m=1}^M \beta_m b_m(\tilde{s}_i, \tilde{k}_m)$$

648 Note that $b_m(,)$ is a basis function that maps $R \times R \rightarrow R$. A popular bivariate basis

649 function uses *thin-plate splines* (Wood, 2003a), which extend to $\mathbf{s}_i \in \mathbb{R}^d$ and ∂l_g penalties.
 650 These splines are designed to interpolate and approximate smooth surfaces over two dimensions
 651 (hence the “bivariate” term). For $d = 2$ dimensions and $l = 2$ (smoothness penalty involving
 652 second order derivative):

$$f(\tilde{s}_i) = \alpha + \beta_1 x_i + \beta_2 z_i + \sum_{m=1}^M \beta_{2+m} b_m(\tilde{s}_i, \tilde{k}_m)$$

653 using the the radial basis function given by:

$$b_m(\tilde{s}_i, \tilde{k}_m) = \|\tilde{s}_i - \tilde{k}_m\|^2 \log \|\tilde{s}_i - \tilde{k}_m\|$$

654 where $\|\mathbf{s}_i - \mathbf{k}_m\|$ is the Euclidean distance between the covariate \mathbf{s}_i and the knot location
 655 \mathbf{k}_m . We fitted this model using `brms...` using the `t2()` smooth constructor with full penalties
 656 (E. J. Pedersen et al., 2019)...

```
# fitting a GAM with two temporal dimensions
timegen_gam <- brm(
  # 2D thin-plate spline (tp)
  # auc ~ t2(train_time, test_time, bs = "tp", k = 10),
  auc ~ t2(train_time, test_time, bs = "tp", k = 20),
  data = timegen_data,
  family = Beta(),
  warmup = 2000,
  iter = 5000,
  chains = 8,
  cores = 8,
  # file = "models/timegen_gam_t2.rds" # k = 10
  file = "models/timegen_gam_t2_k20.rds" # k = 20
)

# fitting a GAM with two temporal dimensions to empirical data
timegen_gam <- brm(
  # 2D thin-plate spline (tp)
  # auc ~ t2(train_time, test_time, bs = "tp", k = 30),
  auc ~ t2(train_time, test_time, bs = "tp", k = 20, full = TRUE),
  data = timegen_data,
  family = Beta(),
  warmup = 1000,
  iter = 3000,
  chains = 8,
  cores = 8,
  # file = "models/timegen_gam_meg_t2_k30.rds"
  file = "models/timegen_gam_meg_t2_k20_full.rds"
```

```
)  
  
# retrieving the data  
# timegen_data <- timegen_gam$data  
  
# PPCs  
# pp_check(timegen_gam)
```

657 Could be extended to spatial and temporal dimensions with formulas such as `te(x, y,`
658 `Time, d = c(2, 1) ...`

Appendix B

Alternative to GAMs: Approximate Gaussian Process regression

659 A Gaussian process (GP) is a stochastic process that defines the distribution over a collection
 660 of random variables indexed by a continuous variable, that is $\{f(t) : t \in \mathcal{T}\}$ for some index
 661 set \mathcal{T} (Rasmussen & Williams, 2005; Riutort-Mayol et al., 2023). Whereas Bayesian linear
 662 regression outputs a distribution over the parameters of some predefined parametric model, the
 663 GP approach, in contrast, is a non-parametric approach, in that it finds a distribution over the
 664 possible functions that are consistent with the observed data. However, note that nonparametric
 665 does not mean there aren't parameters, it means that there are infinitely many parameters.

666 From brms documentation: A GP is a stochastic process, which describes the relation
 667 between one or more predictors $x = (x_1, \dots, x_d)$ and a response $f(x)$, where d is the number
 668 of predictors. A GP is the generalization of the multivariate normal distribution to an infinite
 669 number of dimensions. Thus, it can be interpreted as a prior over functions. The values of $f()$
 670 at any finite set of locations are jointly multivariate normal, with a covariance matrix defined
 671 by the covariance kernel $k_p(x_i, x_j)$, where p is the vector of parameters of the GP:

$$(f(x_1), \dots, f(x_n)) \sim \text{MVN}\left(0, (k_p(x_i, x_j))_{i,j=1}^n\right)$$

672 The smoothness and general behaviour of the function f depends only on the choice of
 673 covariance kernel, which ensures that values that are close together in the input space will be
 674 mapped to similar output values...

675 From this perspective, f is a realisation of an infinite dimensional normal distribution:

$$f \sim \text{Normal}(0, C(\lambda))$$

676 where C is a covariance kernel with hyperparameters λ that defines the covariance
 677 between two function values $f(t_1)$ and $f(t_2)$ for two time points t_1 and t_2 (Rasmussen &
 678 Williams, 2005). Similar to the different choices of the basis function for splines, different
 679 choices of the covariance kernel lead to different GPs. In this article, we consider the squared-
 680 exponential (a.k.a. radial basis function) kernel, which computes the squared distance between
 681 points and converts it into a measure of similarity. It is defined as:

$$C(\lambda) := C(t_1, t_2, \sigma, \gamma) := \sigma^2 \exp\left(-\frac{\|t_1 - t_2\|^2}{2\gamma^2}\right)$$

682 with hyperparameters $\lambda = (\sigma, \gamma)$, expressing the overall scale of GP and the length-
 683 scale, respectively (Rasmussen & Williams, 2005). The advantages of this kernel are that it is
 684 computationally efficient and (infinitely) smooth making it a reasonable choice for the purposes
 685 of the present article. Here again, λ hyperparameters are estimated from the data, along with
 686 all other model parameters.

687 Taken from <https://michael-franke.github.io/Bayesian-Regression/practice-sheets/10c-Gaussian-processes.html>: For a given vector \mathbf{x} , we can use the kernel to construct finite
 688 multi-variate normal distribution associated with it like so:

$$\mathbf{x} \mapsto_{GP} \text{MVNormal}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}))$$

690 where m is a function that specifies the mean for the distribution associated with \mathbf{x} . This
691 mapping is essentially the Gaussian process: a systematic association of vectors of arbitrary
692 length with a suitable multi-variate normal distribution.

693 Low-rank approximate Gaussian processes are of main interest in machine learning and

694 statistics due to the high computational demands of exact Gaussian process models ([Riutort-](#)

695 [Mayol et al., 2023](#))...

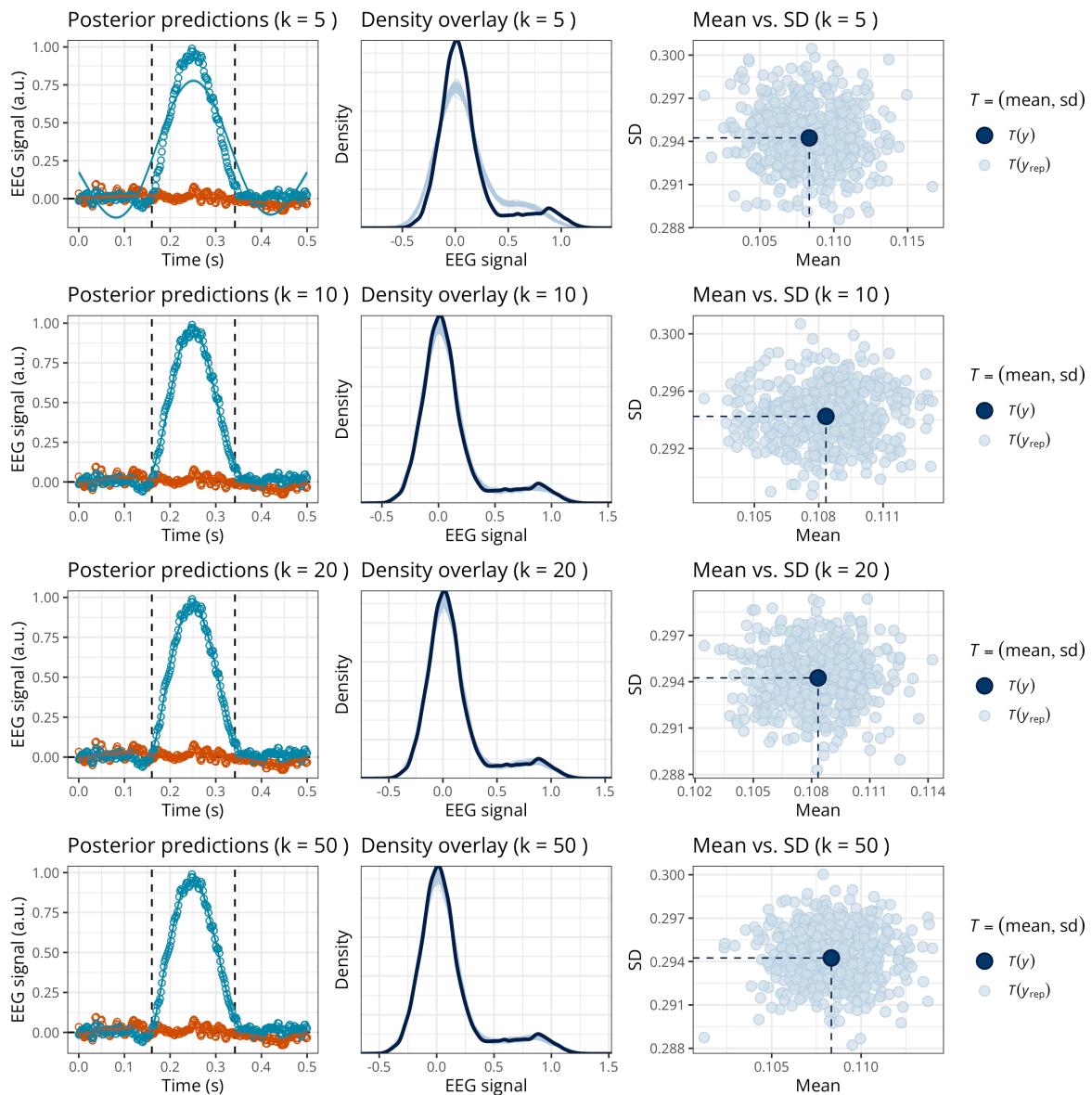
Appendix C

How to choose the GAM basis dimension?

696 Here we should provide recommendation about how to define k ... An option is to vary k and
 697 examine the predictions and posterior predictive checks (PPCs) of each model... In this example
 698 (Figure C1)... However, it is not possible to provide general recommendations, as the optimal
 699 k depends on the sampling rate, the preprocessing steps (e.g., signal-to-noise ratio, low-pass
 700 filtering, etc), and the neural dynamics of the phenomenon under study. Choose k as large a
 701 possible (given limits on computation time etc)? Or use LOOIC/WAIC (seem to select too
 702 simple models)?

Figure C1

Posterior predictions and posterior predictive checks for the GAM with varying k (in rows).



Appendix D

R package and integration with MNE-Python

703 For users who are already familiar with `brms`, the recommended pipeline is to import ERPs or
 704 decoding results in R and analyse these data using the code provided in the main paper. It is
 705 also possible to call functions from the `neurogam` R package (available at <https://github.com/lNALBORCZYK/neurogam>) which come with sensible defaults.
 706

```
# installing (if needed) and loading the neurogam R package
# remotes::install_github("https://github.com/lNALBORCZYK/neurogam")
library(neurogam)

# using the testing_through_time() function from the neurogam package
# this may take a few minutes (or hours depending the machine's
# performance and data size)...
gam_onset_offset <- testing_through_time(
  # dataframe with M/EEG data in long format
  data = raw_df,
  # threshold for defining clusters (20 by default)
  threshold = 20,
  # the *_id arguments are used to specify the relevant columns in data
  participant_id = "participant", meeg_id = "eeg",
  time_id = "time", predictor_id = "condition",
  # number of warmup MCMC iterations
  warmup = 1000,
  # total number of MCMC iterations
  iter = 5000,
  # number of MCMCs
  chains = 4,
  # number of parallel cores to use for running the MCMCs
  cores = 4
)

# displaying the results
gam_onset_offset$clusters
```

707 The `neurogam` package can also be called from Python using the `rpy2` module, and can
 708 easily be integrated into MNE-Python pipelines. For example, we use it below to estimate the
 709 onset and offset of effects for one EEG channel from a MNE evoked object. Note that the code
 710 used to reshape the `sample` MNE dataset is available in the online supplementary materials, and
 711 we refer to the [MNE documentation](#) about converting MNE epochs to Pandas dataframes in long
 712 format (i.e., with one observation per row).

```
# loading the Python modules
import rpy2.robj as robj
from rpy2.robj.packages import importr
from rpy2.robj import pandas2ri
from rpy2.robj.conversion import localconverter

# importing the "neurogam" R package
neurogam = importr("neurogam")

# activating automatic pandas-R conversion
pandas2ri.activate()

# assuming reshaped_df is some M/EEG data reshaped in long format
with localconverter(robj.default_converter + pandas2ri.converter):

    reshaped_df_r = robj.conversion.py2rpy(reshaped_df)

# using the testing_through_time() function from the neurogam R package
gam_onset_offset = neurogam.testing_through_time(
    data=reshaped_df_r,
    threshold=10,
    multilevel=False
)

# displaying the results
print(list(gam_onset_offset) )
```