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

Ladislas Nalborczyk[1] and Paul Bürkner[2]

[1]Aix Marseille Univ, CNRS, LPL

[2]TU Dortmund University, Department of Statistics

Abstract

Time-resolved electrophysiological measurements such as those obtained through magneto- or electro-encephalography (M/EEG) offer a unique window into 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, followed by corrections for multiple comparisons (e.g., FDR, FWER) or cluster-based inference. While effective for controlling error rates, cluster-based methods come with a significant limitation: they shift the focus of inference from individual time points to clusters, making it difficult to draw precise conclusions about the onset or offset of observed effects. Here, we introduce a *model-based* approch for analysing M/EEG timeseries such as event-related potentials (ERPs) or decoding performance over time. Our method leverages Bayesian generalised additive multilevel models (GAMMs), 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 also provide an R package implementing the method and show how it can be seamlessly integrated into M/EEG analysis pipelines using MNE-Python.

Keywords: EEG, MEG, generalised additive models, mixed-effects models, multilevel models, Bayesian statistics, brms

Table of contents

42 **Precise temporal localisation of M/EEG effects with Bayesian generalised additive**
43 **multilevel models**

1 ## Introduction

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

9   Here are some useful references to be discussed (Combrisson & Jerbi, 2015; Ehinger
10 & Dimigen, 2019; Frossard & Renaud, 2021, 2022; Gramfort, 2013; Hayasaka, 2003; Luck &
11 Gaspelin, 2017; Maris & Oostenveld, 2007; E. J. Pedersen et al., 2019; Pernet et al., 2015)... See
12 also Maris (2011) and Rosenblatt et al. (2018) (history of cluster-based approaches)... Clusters
13 failures (Eklund et al., 2016)...

14 ## Problem statement

15   Description of cluster-based approaches (see Sassenhagen & Draschkow, 2019)...

16   However, as pointed by the original authors themselves (Maris & Oostenveld, 2007),
17 "there is a conflict between this interest in localized effects and our choice for a global null
18 hypothesis: by controlling the FA [false alarm] rate under this global null hypothesis, one cannot
19 quantify the uncertainty in the spatiotemporal localization of the effect" (Maris & Oostenveld,
20 2007; Sassenhagen & Draschkow, 2019). In contrast, the proposed approach, based on Bayesian
21 generalised additive multilevel models, allows quantifying the posterior probability of effects
22 being above chance at the level of timesteps, voxels, sensors (etc), while naturally taking into
23 account spatiotemporal dependencies present in M/EEG timeseries.

24 ## Previous work

25   Recent example of GLM for EEG (Fischer & Ullsperger, 2013; Wüllhorst et al., 2025)...
26 See also (Hauk et al., 2006; Rousselet et al., 2008)... Example of two-stage regression analysis
27 (i.e., individual-level then group-level, Dunagan et al., 2024)...

28   See also the rERP framework (N. J. Smith & Kutas, 2014a, 2014b) and Tremblay &
29 Newman (2014)...

30   From Dimigen & Ehinger (2021): Recently, spline regression has been applied to ERPs
31 (e.g., Hendrix et al., 2017; Kryuchkova et al., 2012)... GAMMs for EEG data (Abugaber et al.,
32 2023; Meulman et al., 2015, 2023)...

33   Disentangling overlapping processes (Skukies et al., 2024; Skukies & Ehinger, 2021)...
34 Weighting single trials (Pernet, 2022)... The LIMO toolbox (Pernet et al., 2011)...

35   Recently, Teichmann (2022) provided a detailed tutorial on using Bayes factors (BFs)
36 to analyse the 1D or 2D output from MVPA, that is, for testing, at every timestep, whether
37 decoding performance is above chance level. However, this approach provides timeseries of BFs
38 that ignores temporal dependencies..

39 ## Bayesian regression modelling

40   Short intro/recap about Bayesian (linear and generalised) regression models...

41 ## Generalised additive models

42   See for instance these tutorials (Sóskuthy, 2017; Winter & Wieling, 2016) or application
43 to phonetic data (Sóskuthy, 2021; Wieling, 2018) or this introduction (Baayen & Linke, 2020) or

44 these reference books (Hastie & Tibshirani, 2017; Wood, 2017a)... application to pupillometry
45 (Rij et al., 2019)... GAMLSS for neuroimaging data (Dinga et al., 2021)... Modelling auto-
46 correlation in GAMMs + EEG example (Baayen et al., 2018)...

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

$$y_i \sim \mathrm{EF}\left(\mu_i, \phi\right)$$

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

50 where $y_i \sim \mathrm{EF}\left(\mu_i, \phi\right)$ denotes that the observations y_i are distributed as some member
51 of the exponential family of distributions (e.g., Gaussian, Gamma, Beta, Poisson) with mean μ_i
52 and scale parameter ϕ; $g(\cdot)$ is the link function, $\mathbf{A}_i$ is the ith row of a known parametric model
53 matrix, γ is a vector of parameters for the parametric terms (to be estimated), f_j is a smooth
54 function of covariate x_j (to be estimated as well). The smooth functions f_j are represented in
55 the model via penalised splines basis expansions of the covariates, that are a weighted sum of
56 basis functions:

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

57 where β_{jk} is the weight (coefficient) associated with the kth basis function $b_{jk}()$ evaluated
58 at the covariate value x_{ij} for the jth smooth function f_j. Splines coefficients are penalised
59 (usually through the squared of the smooth functions' second derivative) in a way that can be
60 interpreted, in Bayesian terms, as a prior on the "wiggliness" of the function. In other words,
61 more complex (wiggly) basis function are penalised. To clarify the terminology at this stage:
62 *splines* are functions composed of simpler functions. These simpler functions are basis functions
63 (e.g., a cubic polynomial) and the set of basis functions is a *basis*. Each basis function gets its
64 coefficient and the resultant spline is the sum of these weighted basis functions.

Bayesian generalised additive multilevel models

66 Now describe the Bayesian GAMM (Miller, 2025)... Proper inclusion of varying/random
67 effects in the model specification protects against overly wiggly curves (Baayen & Linke, 2020)...
68 Generalising to scale and shape or "distributional GAMs" (Rigby & Stasinopoulos, 2005; Umlauf
69 et al., 2018) and applied to neuroimaging data (Dinga et al., 2021)...
70 Instead of averaging, obtain the smooth ERP signal from multilevel GAM... less suscep-
71 tible to outliers (Meulman et al., 2023)...

Objectives

73 Focusing on identifying onset and offset of effects (as assessed by ERP differences or
74 decoding performance)... Assessing the performance of a model-based approach (i.e., Bayesian
75 GAMMs) to conventional methods (multiplicity corrections or cluster-based permutation)...

Methods

M/EEG data simulation

78 Following the approach of Sassenhagen & Draschkow (2019) and Rousselet (2025), we
79 simulated EEG data stemming from two conditions, one with noise only, and the other with
80 noise + signal. As in previous studies, the noise was generated by superimposing 50 sinusoids

81 at different frequencies, following an EEG-like spectrum (see details and code in Yeung et al.,
82 2004). As in Rousselet (2025), the signal was generated from truncated Gaussian with an
83 objective onset at 160 ms, a peak at 250 ms, and an offset at 342 ms. We simulated this signal
84 for 250 timesteps between 0 and 0.5s, akin to a 500 Hz sampling rate. We simulated such data
85 for a group of 20 participants with 50 trials per participant and condition (Figure 1).

Figure 1

*Averaged (mean) simulated EEG activity in two conditions with 50 trials each, for a group of
20 participants. The error band represents the mean plus/minus 1 standard error of the mean.*
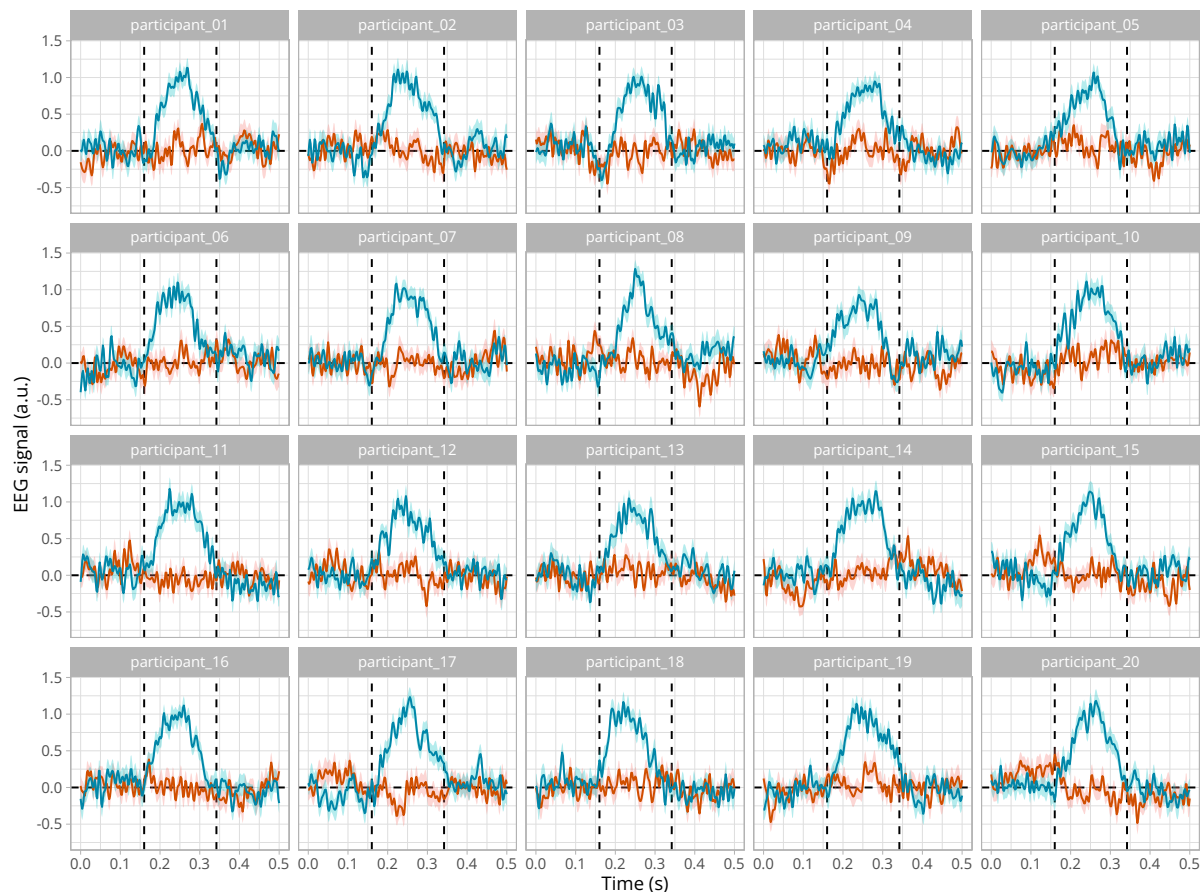
86   We computed the average of the ERP difference (Figure 2)...

Model fitting

88   We then fitted a Bayesian GAM using the `brms` package (Bürkner, 2017, 2018; Nalbor-
89 czyk et al., 2019). We used the default priors in `brms` (i.e., weakly informative priors). We ran
90 eight Markov Chain Monte-Carlo (MCMC) to approximate the posterior distribution, including
91 each 5000 iterations and a warmup of 2000 iterations, yielding a total of $8 \times (5000 - 2000) = 24000$
92 posterior samples to use for inference. Posterior convergence was assessed examining trace plots
93 as well as the Gelman–Rubin statistic $\hat{R}$. The `brms` package uses the same syntax as the R pack-
94 age `mgcv` v 1.9-1 (Wood, 2017b) for specifying smooth effects. Figure 3 shows the predictions
95 of this model together with the raw data.

```
# averaging across participants
ppt_df <- raw_df %>%
    group_by(participant, condition, time) %>%
```

Figure 2

Group-level average difference between conditions (mean +/- standard error of the mean). The 'true' onset and offset are indicated by the vertical dashed lines.
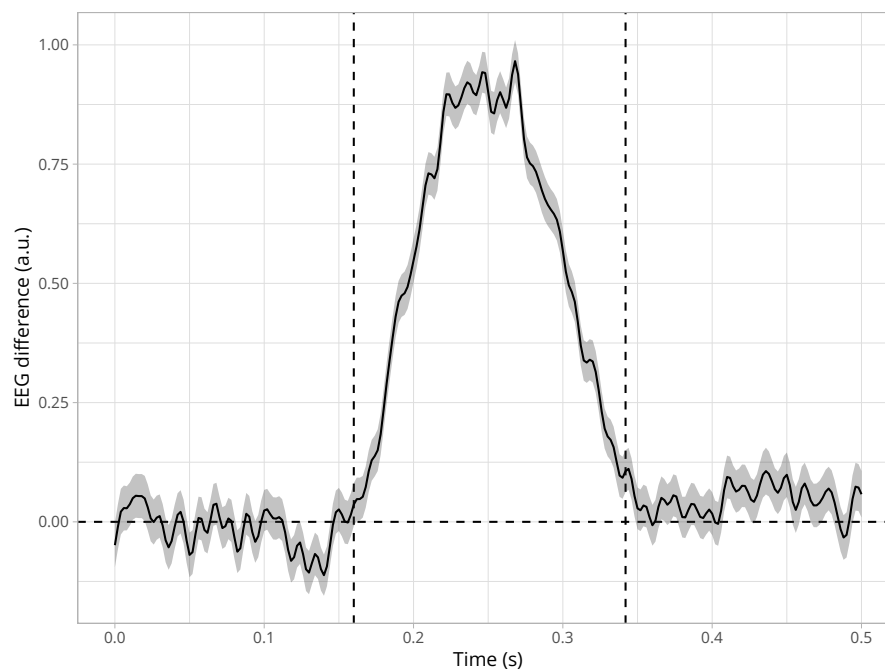

```
    summarise(eeg = mean(eeg) ) %>%
    ungroup()

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

# timing the model fitting
# around 600s for 2000 iterations on 4 chains and 4 cores
# around 190s for 2000 iterations on 4 chains and 4 cores + cmdstanr and 01
# around 440s for 2000 iterations on 4 chains and 4 cores + threading(2), cmdstanr and 01
# library(tictoc)
# tic()

# fitting the GAM
gam <- brm(
    # cubic regression splines with k-1 basis functions
    # eeg ~ condition + s(time, bs = "cr", k = 20, by = condition),
    # thin-plate regression splines with k-1 basis functions
    eeg ~ 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"
    ###############################
```

```
    # speeding up the model fitting #
    ###############################
    # backend = "cmdstanr",
    # threads = threading(2),
    # stan_model_args = list(stanc_options = list("O1") )
    )

# stop timing
# toc()
```

Posterior probability of difference above 0

We then plot the posterior predictions together with the posterior estimate of the slope for `condition` at each timestep (Figure 3).

Figure 3

Posterior estimate of the ERP in each condition (left) or directly for the difference of ERPs (right) according to the GAM.
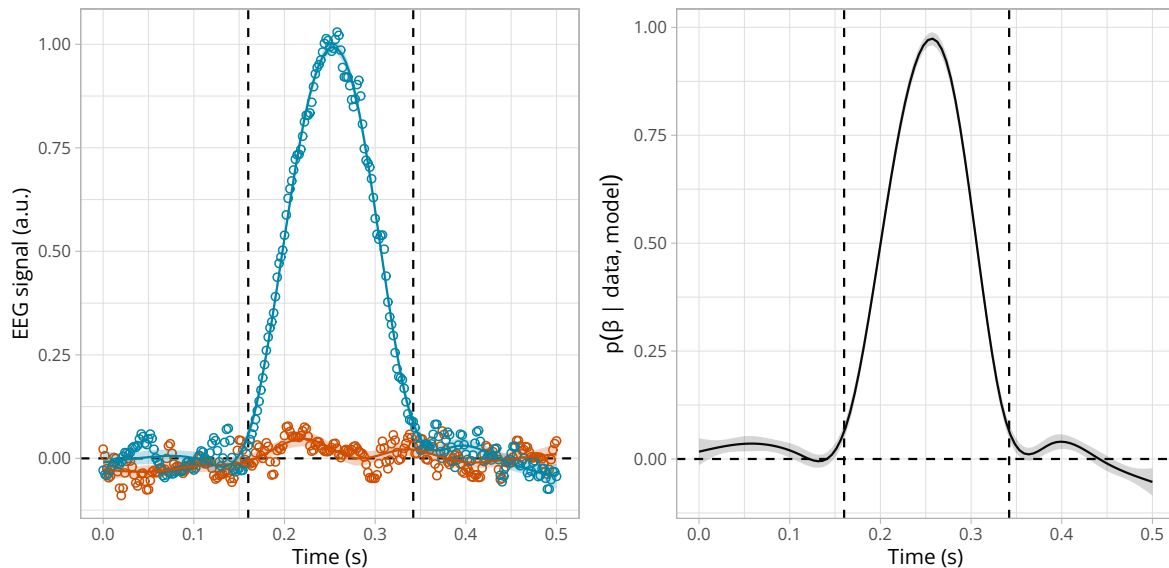
We then compute the posterior probability of the slope for `condition` being above $0 + \epsilon$ (Figure 4, left), with $\epsilon := 0.05$, which can be interpreted as the smallest effect size of interest. We can also express this as the ratio of posterior probabilities (i.e., $p/(1 - p)$) and visualise the timecourse of this ratio superimposed with the conventional thresholds on evidence ratios (Figure 4, right).

Figure 4

Left: Posterior probability of the ERP difference (slope) being above 0 according to the GAM. Right: Ratio of posterior probability according to the GAM (on a log10 scale). Timesteps above threshold (10) are highlighted in green.
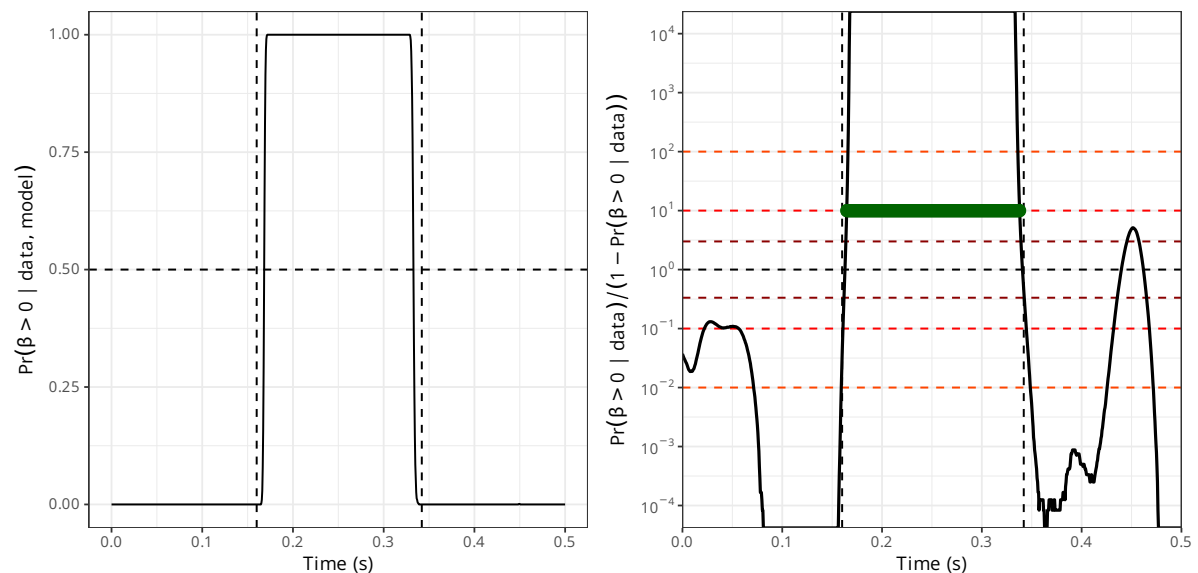
Multilevel modelling using ERP summary statistics

Next we fit a multilevel version of the GAM. Although it is possible to fit a GAMM at the single-trial level, we present a resource-efficient version of the model that is fitted directly on the by-participant summary statistics (mean and SD, similar to what is done in meta-analysis...

```r
# averaging across participants
summary_df <- raw_df %>%
    summarise(
        eeg_mean = mean(eeg),
        eeg_sd = sd(eeg),
        .by = c(participant, condition, time)
        )

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

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

```
    file = "models/meta_gam.rds"
    )
```

Error properties of the proposed approach

108   We then computed the difference between the true and estimated onset/offset of the ERP
110 difference (error $:= |\hat{\theta} - \theta|$), according to various k (GAM basis dimension) and `threshold` val-
111 ues. Remember that the signal is generated from a truncated Gaussian with an objective onset
112 at 160 ms, a maximum at 250 ms, and an offset at 342 ms. Figure 5 shows that the hierarchical
113 GAM can *exactly* recover the true onset and offset values, given some reasonable choice of k
114 and `threshold` values (e.g., a threshold of 20). We provide more detailed recommendations on
115 how to set k in Section C.

Figure 5

*Error function of onset (left) and offset (right) estimation according to various basis dimension
and threshold values (according to the multilevel GAM).*
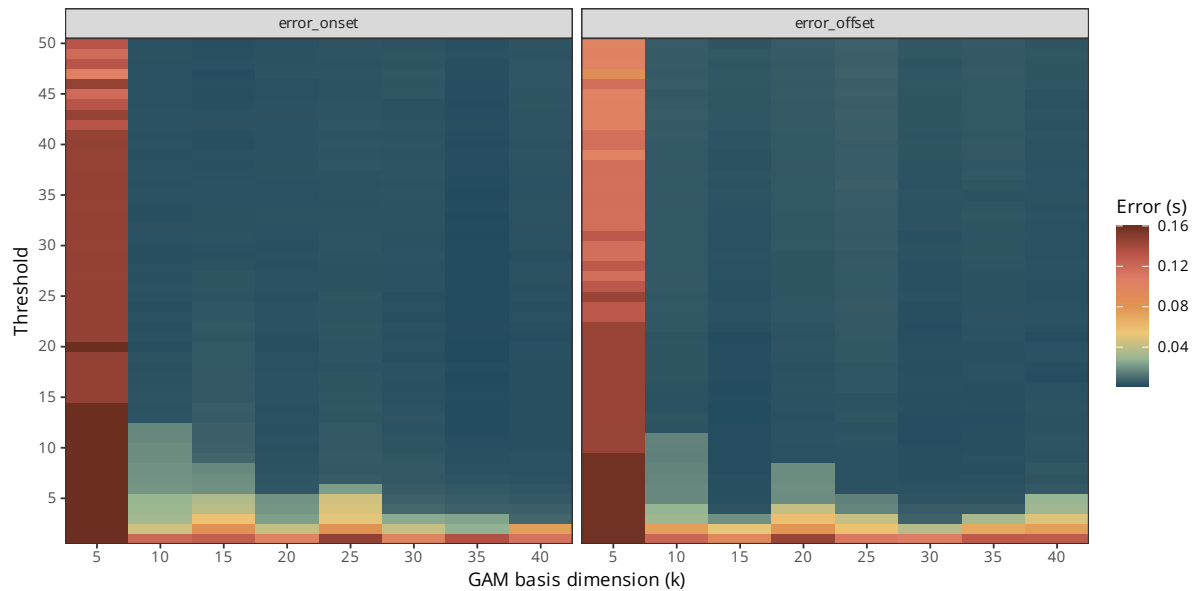
Comparing the identified onsets/offsets to other approaches

To provide a more exhaustive assessment of the performance of the proposed approach, we compared the ability of the GAMM to correctly estimate the onset and offset of the ERP difference to widely-used methods. First, we conducted mass-univariate t-tests (thus treating each timestep independently) and identified the onset and offset of the ERP difference as the first and last values crossing an arbitrary significance threshold ($\alpha = 0.05$). We then followed the same approach but after applying different forms of multiplicity correction to the p-values. We compared two methods that control the false discovery rate (FDR) (i.e., BH95, Benjamini & Hochberg, 1995; and BY01, Benjamini & Yekutieli, 2001), one method that controls the family-wise error rate (FWER) (i.e., Holm–Bonferroni method, Holm, 1979), and two cluster-based permutation methods (permutation with a single cluster-forming threshold and threshold-free cluster enhancement, TFCE, S. Smith & Nichols, 2009). The BH95, BY01, and Holm corrections were applied to the p-values using the `p.adjust()` function in R. The cluster-based inference was implemented using a cluster-sum statistic of squared t-values, as implemented in `MNE-Python` (Gramfort, 2013), called via the R package `reticulate` v 1.42.0 (Ushey et al., 2024). We also compared these estimates to the onset and offset as estimated using the binary segmentation algorithm, as implemented in the R package `changepoint` v 2.3 (Killick et al., 2022a), and applied directly to the squared t-values (as in Rousselet, 2025). Figure 6 illustrates the onsets and offsets estimated by each method on a single simulated dataset and shows that all methods over-estimate the true onset and under-estimate 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).
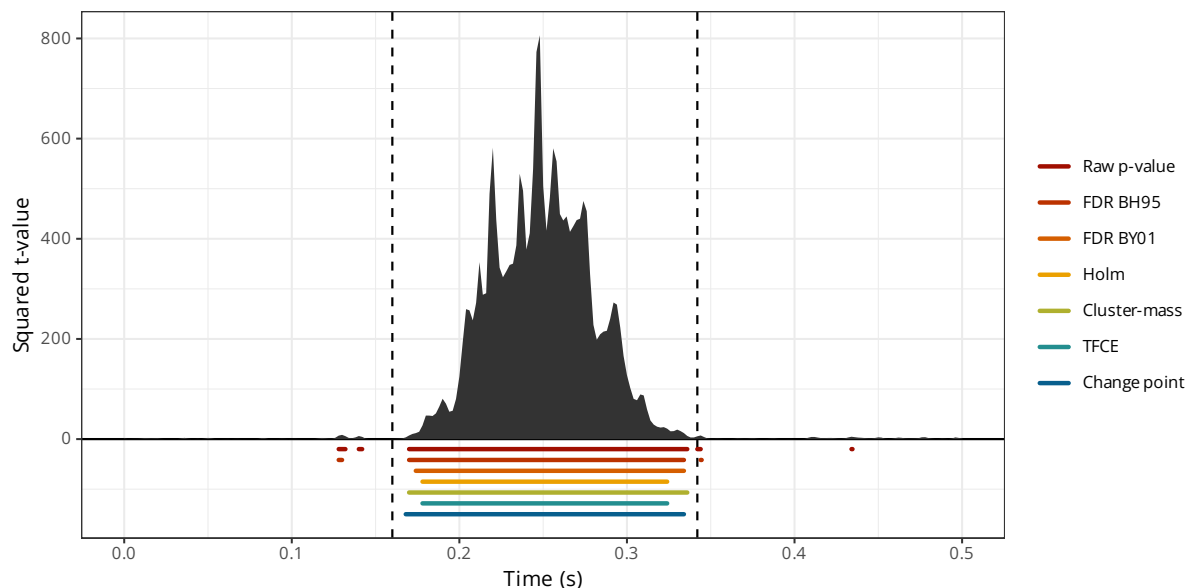
Simulation study

The various methods were assessed using the median error and variance of onset/offset estimates computed on 10.000 simulated datasets...

Application to actual MEG data

Next, we assessed the performance of the proposed approach on actual MEG data (decoding results from Nalborczyk et al., in preparation). We conducted time-resolved multivariate pattern analysis (MVPA), also known as decoding... As a result, we have a timecourse of decoding performance (ROC AUC), bounded between 0 and 1, for each participant (for a total of 32 participants). Now, we want to *test* whether the group-level average decoding accuracy is above chance (i.e., 0.5) at each timestep (Figure 7). We fitted a similar GAM as discussed previously, but we replaced the Normal likelihood function by a Beta one to account for the bounded nature of AUC values (between 0 and 1) (for a tutorial on Beta regression, see Coretta & Bürkner, 2025). Note that although we chose a basis dimension of $k = 50$, which seems appropriate for the present data, this choice should be adapted according to the properties of the modelled data (e.g., signal-to-noise ratio, prior low-pass filtering, sampling rate, etc) and should be assessed by the usual model checking tools (e.g., posterior predictive checks, see also Section C). To distinguish signal from noise, we also defined a region of practical equivalence (ROPE, Kruschke & Liddell, 2017), defined as the chance level plus the standard deviation of the (group-level average) decoding performance during the baseline period.

```r
# 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
    )
```

We assessed the reliability of the proposed approach using a form of permutation-based split-half reliability (as for instance in Rosenblatt et al., 2018), which consisted of the following steps. First, we created 1000 split halves of the data (i.e., with half the participants in the original data, that is, 16 participants). For each split, we estimated the onset/offset using all methods described previously. Third, we summarised the distribution of onset/offset estimates using the median "error" (i.e., difference between the split estimate and the estimate obtained using the full dataset) and the variance across splits.

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. The blue horizontal markers indicate the timesteps at which the posterior probability ratio exceeds 20.
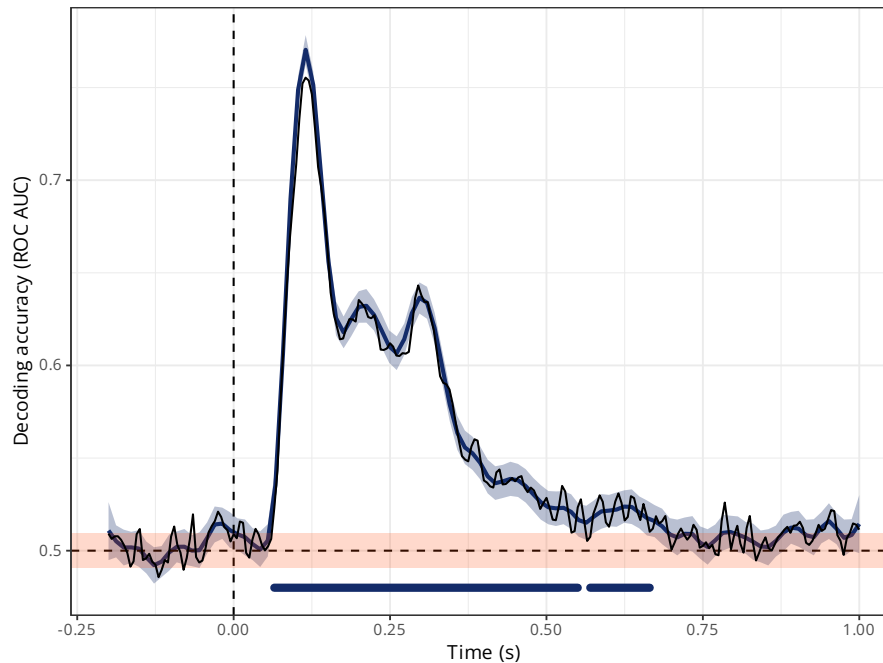
Results

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

Simulation results (bias and variance)

Figure 8 shows a summary of the simulation results, revealing that the proposed approach (BGAM) has the lowest median error and variance for both the onset and offset estimates.

Figure 8

Median error and variance of onset and offset estimates for each method. Variance is plotted on a log10 scale for visual purposes.
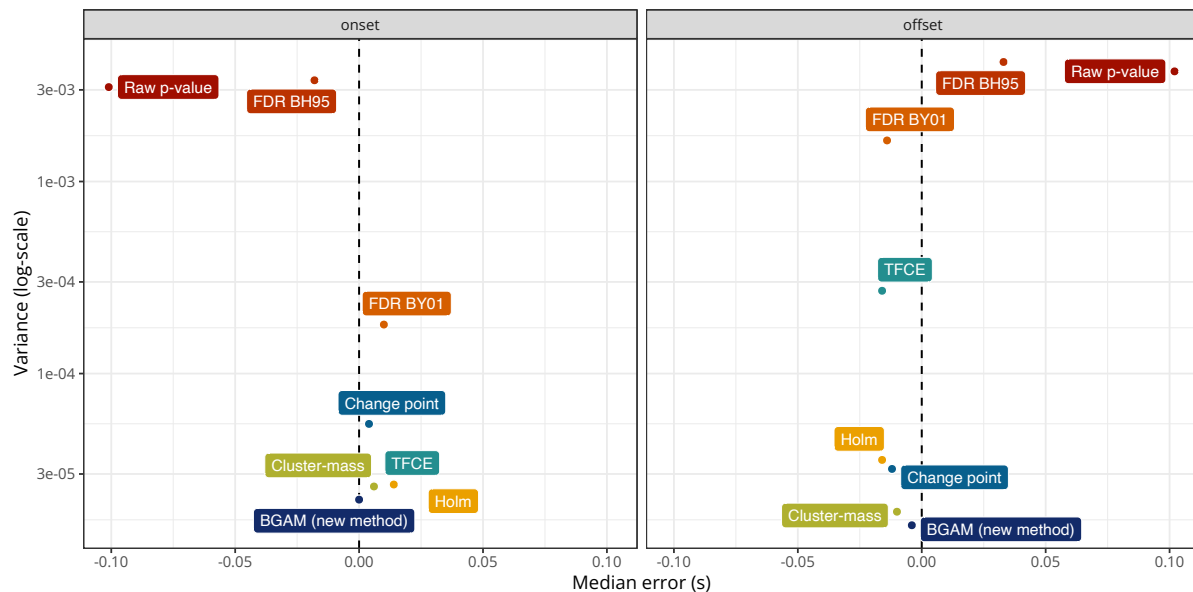
Application to actual MEG data (reliability)

Figure 9 shows the group-level average decoding performance through time with onset and offset estimates for each method. Overall, this figure shows that both the `Raw p-value` and `FDR BH95` methods were extremely lenient, considering that the decoding performance was above chance before the onset of the stimulus (false positive) and until the end of the trial. The `Change point` and `Cluster mass` methods were the most conservative methods, identifying a time window from approximately +60ms to +500ms. The `Holm`, `TFCE`, and `BGAM` methods produced similar estimates of onset and offset, ranging from approximately +60ms to +650ms.[1]

Figure 10 shows the median difference between the onset and offset estimates from each data split and the onset and offset estimates from the full dataset (x-axis) along with the variance of its onset and offset estimates across data splits (error bar). This figure reveals that the `BGAM` *onset and offset* estimates on each split are the closest to the estimates from the full dataset on average (0ms difference for the onset estimate and 5ms difference for the offset estimate). The `Raw p-value` method has similar performance, but given the aberrant estimates it produces (cf. Figure 9), the fact that it is consistent between data splits and the full dataset is not convincing on its own. The `Change point` method also has a very good performance (i.e., very low difference between split estimates and full estimates), but produces too short cluster of significant decoding performance (cf. Figure 9). Overall, the figure reveals that for all other methods, split datasets produce later onset estimates and earlier offset estimates (as compared to the estimates from the model fitted on the full dataset). These results highlight that the results of any method should be assessed using both i) good asymptotic properties on simulated data, ii) sensible identified clusters in actual data, and iii) reliable/stable estimates on actual data.

---

[1]It should be noted that although each method can produce several "clusters" of "significant" timesteps, we only considered the first (onset) and last (offset) timesteps identified by each method to compute the error (difference).

Figure 9

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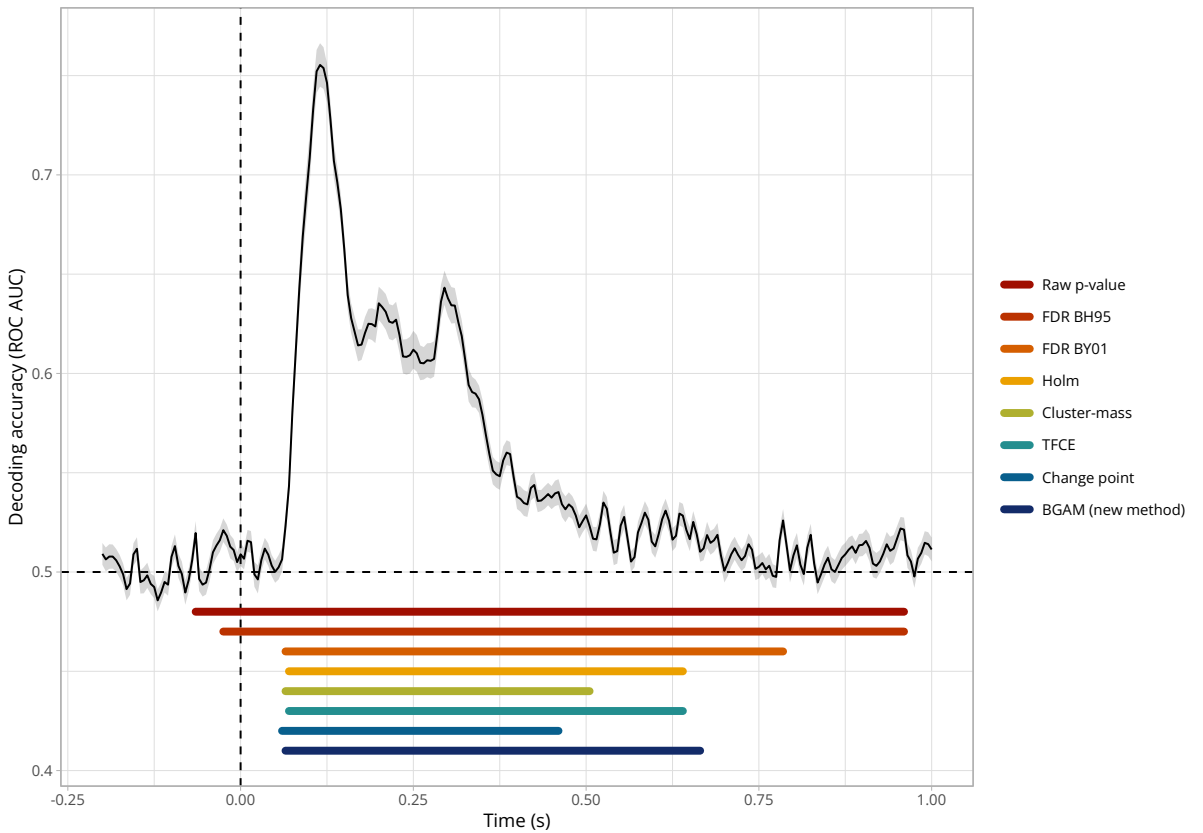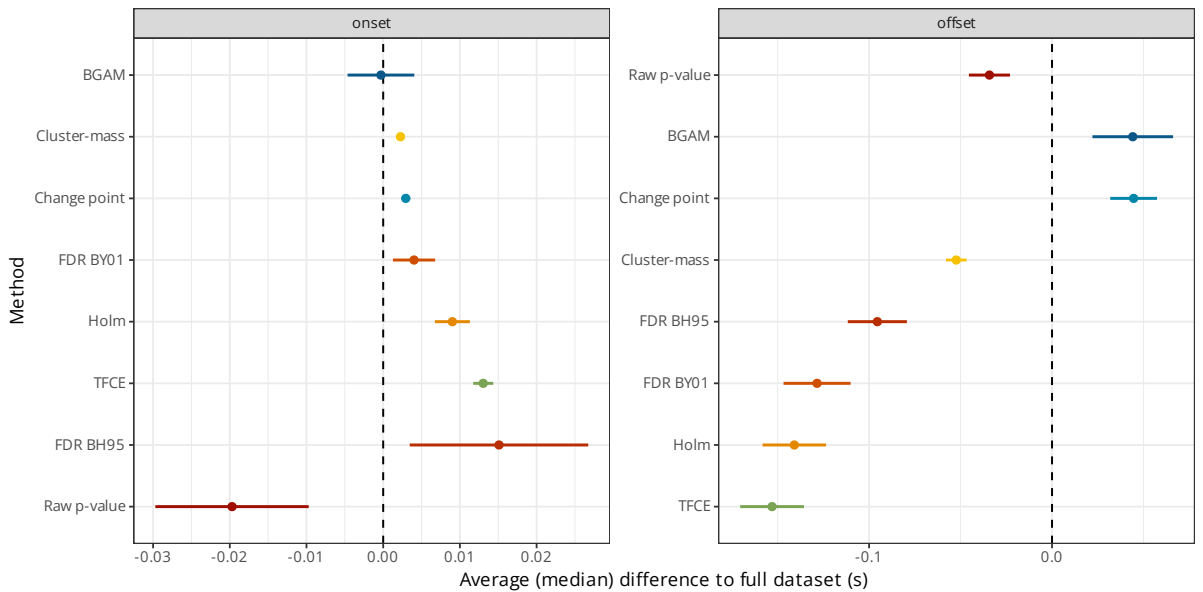

Figure 10

Median absolute error and variance of onset (left) and offset (right) estimates for each method.

Discussion

Summary of the proposed approach

Overall, before concluding on the onset/offset of effect based on the model, we need to ensure that the model provides a faithful description of the data-generating process (e.g., via

198 posterior predictive checks etc)...

Increasing potential usage

200 Prepare a wrapper `R` package and show how to call it in `Python` and integrate it with
201 `MNE-Python` (Gramfort, 2013) pipelines...

Limitations and future directions

203 As in previous simulation work (e.g., Rousselet et al., 2008; Sassenhagen & Draschkow,
204 2019), the present simulation results depend on various choices such as the specific cluster-
205 forming algorithm and threshold, signal-to-noise ratio, negative impact of preprocessing steps
206 (e.g., low-pass filter) on temporal resolution... note however, that the same caveats apply to all
207 methods...
208 The error properties depend on the threshold parameter, a value of 10 or 20 seems to
209 be a reasonable default, but the optimal threshold parameter can be adjusted using split-half
210 reliability assessment...
211 Can be applied to any 1D timeseries (e.g., pupillometry, electromyography)... Extending
212 the approach to spatiotemporal data (i.e., time + sensors)...
213 We kept the exemplary models simple, but can be extended by adding varying/random
214 effects (intercept and slope) for item (e.g., word)... but also continuous predictors at the trial
215 level?

Conclusions

217 ...

Data and code availability

The simulation results as well as the `R` code to reproduce the simulations are available on GitHub: https://github.com/lnalborczyk/brms_meeg.

Packages

We used R version 4.2.3 (R Core Team, 2023) and the following R packages: brms v. 2.22.0 (Bürkner, 2017, 2018, 2021), changepoint v. 2.2.4 (Killick et al., 2022b; Killick & Eckley, 2014), grateful v. 0.2.10 (Rodriguez-Sanchez & Jackson, 2023), knitr v. 1.45 (Xie, 2014, 2015, 2023), MetBrewer v. 0.2.0 (Mills, 2022), pakret v. 0.2.2 (Gallou, 2024), patchwork v. 1.2.0 (T. L. Pedersen, 2024), rmarkdown v. 2.29 (Allaire et al., 2024; Xie et al., 2018, 2020), scales v. 1.3.0 (Wickham et al., 2023), scico v. 1.5.0 (T. L. Pedersen & Crameri, 2023), tidybayes v. 3.0.6 (Kay, 2023), tidyverse v. 2.0.0 (Wickham et al., 2019).

Acknolwedgements

References

Abugaber, D., Finestrat, I., Luque, A., & Morgan-Short, K. (2023). Generalized additive mixed modeling of EEG supports dual-route accounts of morphosyntax in suggesting no word frequency effects on processing of regular grammatical forms. *Journal of Neurolinguistics*, *67*, 101137. https://doi.org/10.1016/j.jneuroling.2023.101137

Allaire, J., Xie, Y., Dervieux, C., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., Chang, W., & Iannone, R. (2024). *rmarkdown: Dynamic documents for r.* https://github.com/rstudio/rmarkdown

Baayen, R. H., & Linke, M. (2020). *Generalized Additive Mixed Models* (pp. 563–591). Springer International Publishing. https://doi.org/10.1007/978-3-030-46216-1_23

Baayen, R. H., Rij, J. van, Cat, C. de, & Wood, S. (2018). *Autocorrelated errors in experimental data in the language sciences: Some solutions offered by generalized additive mixed models* (pp. 49–69). Springer International Publishing. https://doi.org/10.1007/978-3-319-69830-4_4

Benjamini, Y., & Hochberg, Y. (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *57*(1), 289–300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x

Benjamini, Y., & Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, *29*(4). https://doi.org/10.1214/aos/1013699998

Bürkner, P.-C. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, *80*(1), 1–28. https://doi.org/10.18637/jss.v080.i01

Bürkner, P.-C. (2018). Advanced Bayesian multilevel modeling with the R package brms. *The R Journal*, *10*(1), 395–411. https://doi.org/10.32614/RJ-2018-017

Bürkner, P.-C. (2021). Bayesian item response modeling in R with brms and Stan. *Journal of Statistical Software*, *100*(5), 1–54. https://doi.org/10.18637/jss.v100.i05

Combrisson, E., & Jerbi, K. (2015). Exceeding chance level by chance: The caveat of theoretical chance levels in brain signal classification and statistical assessment of decoding accuracy. *Journal of Neuroscience Methods*, *250*, 126–136. https://doi.org/10.1016/j.jneumeth.2015.01.010

Coretta, S., & Bürkner, P.-. C. (2025). *Bayesian beta regressions with brms in r: A tutorial for phoneticians.* http://dx.doi.org/10.31219/osf.io/f9rqg_v1

Dimigen, O., & Ehinger, B. V. (2021). Regression-based analysis of combined EEG and eye-tracking data: Theory and applications. *Journal of Vision*, *21*(1), 3. https://doi.org/10.1167/jov.21.1.3

Dinga, R., Fraza, C. J., Bayer, J. M. M., Kia, S. M., Beckmann, C. F., & Marquand, A. F. (2021). *Normative modeling of neuroimaging data using generalized additive models of location scale and shape.* http://dx.doi.org/10.1101/2021.06.14.448106

Dunagan, D., Jordan, T., Hale, J. T., Pylkkänen, L., & Chacón, D. A. (2024). *Evaluating the timecourses of morpho-orthographic, lexical, and grammatical processing following rapid parallel visual presentation: An EEG investigation in english.* http://dx.doi.org/10.1101/2024.04.10.588861

Ehinger, B. V., & Dimigen, O. (2019). Unfold: An integrated toolbox for overlap correction, non-linear modeling, and regression-based EEG analysis. *PeerJ*, *7*, e7838. https://doi.org/10.7717/peerj.7838

Eklund, A., Nichols, T. E., & Knutsson, H. (2016). Cluster failure: Why fMRI inferences for spatial extent have inflated false-positive rates. *Proceedings of the National Academy of Sciences*, *113*(28), 7900–7905. https://doi.org/10.1073/pnas.1602413113

Fischer, Adrian G., & Ullsperger, M. (2013). Real and Fictive Outcomes Are Processed Differently but Converge on a Common Adaptive Mechanism. *Neuron*, *79*(6), 1243–1255.

283   https://doi.org/10.1016/j.neuron.2013.07.006

284 Frossard, J., & Renaud, O. (2021). Permutation Tests for Regression, ANOVA, and Comparison
285   of Signals: The **permuco** Package. *Journal of Statistical Software*, *99*(15). https://doi.org/
286   10.18637/jss.v099.i15

287 Frossard, J., & Renaud, O. (2022). The cluster depth tests: Toward point-wise strong con-
288   trol of the family-wise error rate in massively univariate tests with application to M/EEG.
289   *NeuroImage*, *247*, 118824. https://doi.org/10.1016/j.neuroimage.2021.118824

290 Gallou, A. (2024). *pakret: Cite "R" packages on the fly in "R Markdown" and "Quarto"*. https:
291   //CRAN.R-project.org/package=pakret

292 Gramfort, A. (2013). MEG and EEG data analysis with MNE-python. *Frontiers in Neuro-
293   science*, *7*. https://doi.org/10.3389/fnins.2013.00267

294 Hastie, T. J., & Tibshirani, R. J. (2017). *Generalized Additive Models*. Routledge. https:
295   //doi.org/10.1201/9780203753781

296 Hauk, O., Davis, M. H., Ford, M., Pulvermüller, F., & Marslen-Wilson, W. D. (2006). The
297   time course of visual word recognition as revealed by linear regression analysis of ERP data.
298   *NeuroImage*, *30*(4), 1383–1400. https://doi.org/10.1016/j.neuroimage.2005.11.048

299 Hayasaka, S. (2003). Validating cluster size inference: Random field and permutation methods.
300   *NeuroImage*, *20*(4), 2343–2356. https://doi.org/10.1016/j.neuroimage.2003.08.003

301 Hendrix, P., Bolger, P., & Baayen, H. (2017). Distinct ERP signatures of word frequency, phrase
302   frequency, and prototypicality in speech production. *Journal of Experimental Psychology:
303   Learning, Memory, and Cognition*, *43*(1), 128–149. https://doi.org/10.1037/a0040332

304 Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal
305   of Statistics*, *6*(2), 65–70. http://www.jstor.org/stable/4615733

306 Kay, M. (2023). *tidybayes: Tidy data and geoms for Bayesian models*. https://doi.org/10.5281/
307   zenodo.1308151

308 Killick, R., & Eckley, I. A. (2014). changepoint: An R package for changepoint analysis. *Journal
309   of Statistical Software*, *58*(3), 1–19. https://www.jstatsoft.org/article/view/v058i03

310 Killick, R., Haynes, K., & Eckley, I. A. (2022a). *changepoint: An R package for changepoint
311   analysis*. https://CRAN.R-project.org/package=changepoint

312 Killick, R., Haynes, K., & Eckley, I. A. (2022b). *changepoint: An R package for changepoint
313   analysis*. https://CRAN.R-project.org/package=changepoint

314 King, J.-R., & Dehaene, S. (2014). Characterizing the dynamics of mental representations:
315   the temporal generalization method. *Trends in Cognitive Sciences*, *18*(4), 203–210. https:
316   //doi.org/10.1016/j.tics.2014.01.002

317 Kruschke, J. K., & Liddell, T. M. (2017). The Bayesian New Statistics: Hypothesis testing,
318   estimation, meta-analysis, and power analysis from a Bayesian perspective. *Psychonomic
319   Bulletin & Review*, *25*(1), 178–206. https://doi.org/10.3758/s13423-016-1221-4

320 Kryuchkova, T., Tucker, B. V., Wurm, L. H., & Baayen, R. H. (2012). Danger and usefulness
321   are detected early in auditory lexical processing: Evidence from electroencephalography.
322   *Brain and Language*, *122*(2), 81–91. https://doi.org/10.1016/j.bandl.2012.05.005

323 Luck, S. J., & Gaspelin, N. (2017). How to get statistically significant effects in any ERP
324   experiment (and why you shouldn't). *Psychophysiology*, *54*(1), 146–157. https://doi.org/10.
325   1111/psyp.12639

326 Maris, E. (2011). Statistical testing in electrophysiological studies. *Psychophysiology*, *49*(4),
327   549–565. https://doi.org/10.1111/j.1469-8986.2011.01320.x

328 Maris, E., & Oostenveld, R. (2007). Nonparametric statistical testing of EEG- and MEG-data.
329   *Journal of Neuroscience Methods*, *164*(1), 177–190. https://doi.org/10.1016/j.jneumeth.
330   2007.03.024

331 Meulman, N., Sprenger, S. A., Schmid, M. S., & Wieling, M. (2023). GAM-based individual
332   difference measures for L2 ERP studies. *Research Methods in Applied Linguistics*, *2*(3),
333   100079. https://doi.org/10.1016/j.rmal.2023.100079

Meulman, N., Wieling, M., Sprenger, S. A., Stowe, L. A., & Schmid, M. S. (2015). Age Effects in L2 Grammar Processing as Revealed by ERPs and How (Not) to Study Them. *PLOS ONE*, *10*(12), e0143328. https://doi.org/10.1371/journal.pone.0143328

Miller, D. L. (2025). Bayesian views of generalized additive modelling. *Methods in Ecology and Evolution*. https://doi.org/10.1111/2041-210x.14498

Mills, B. R. (2022). *MetBrewer: Color palettes inspired by works at the metropolitan museum of art*. https://CRAN.R-project.org/package=MetBrewer

Nalborczyk, L., Batailler, C., Lœvenbruck, H., Vilain, A., & Bürkner, P.-C. (2019). An Introduction to Bayesian Multilevel Models Using brms: A Case Study of Gender Effects on Vowel Variability in Standard Indonesian. *Journal of Speech, Language, and Hearing Research*, *62*(5), 1225–1242. https://doi.org/10.1044/2018_jslhr-s-18-0006

Nalborczyk, L., Hauw, F., Torcy, H. de, Dehaene, S., & Cohen, L. (in preparation). *Neural and representational dynamics of tickertape synesthesia.*

Pedersen, E. J., Miller, D. L., Simpson, G. L., & Ross, N. (2019). Hierarchical generalized additive models in ecology: An introduction with mgcv. *PeerJ*, *7*, e6876. https://doi.org/10.7717/peerj.6876

Pedersen, T. L. (2024). *patchwork: The composer of plots*. https://CRAN.R-project.org/package=patchwork

Pedersen, T. L., & Crameri, F. (2023). *scico: Colour palettes based on the scientific colour-maps*. https://CRAN.R-project.org/package=scico

Pernet, C. R. (2022). Electroencephalography robust statistical linear modelling using a single weight per trial. *Aperture Neuro*, *2*, 1–22. https://doi.org/10.52294/apertureneuro.2022.2.seoo9435

Pernet, C. R., Chauveau, N., Gaspar, C., & Rousselet, G. A. (2011). LIMO EEG: A Toolbox for Hierarchical LInear MOdeling of ElectroEncephaloGraphic Data. *Computational Intelligence and Neuroscience*, *2011*, 1–11. https://doi.org/10.1155/2011/831409

Pernet, C. R., Latinus, M., Nichols, T. E., & Rousselet, G. A. (2015). Cluster-based computational methods for mass univariate analyses of event-related brain potentials/fields: A simulation study. *Journal of Neuroscience Methods*, *250*, 85–93. https://doi.org/10.1016/j.jneumeth.2014.08.003

R Core Team. (2023). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Rasmussen, C. E., & Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning*. https://doi.org/10.7551/mitpress/3206.001.0001

Rigby, R. A., & Stasinopoulos, D. M. (2005). Generalized Additive Models for Location, Scale and Shape. *Journal of the Royal Statistical Society Series C: Applied Statistics*, *54*(3), 507–554. https://doi.org/10.1111/j.1467-9876.2005.00510.x

Rij, J. van, Hendriks, P., Rijn, H. van, Baayen, R. H., & Wood, S. N. (2019). Analyzing the Time Course of Pupillometric Data. *Trends in Hearing*, *23*. https://doi.org/10.1177/2331216519832483

Riutort-Mayol, G., Bürkner, P.-C., Andersen, M. R., Solin, A., & Vehtari, A. (2023). Practical Hilbert space approximate Bayesian Gaussian processes for probabilistic programming. *Statistics and Computing*, *33*(1), 17. https://doi.org/10.1007/s11222-022-10167-2

Rodriguez-Sanchez, F., & Jackson, C. P. (2023). *grateful: Facilitate citation of r packages*. https://pakillo.github.io/grateful/

Rosenblatt, J. D., Finos, L., Weeda, W. D., Solari, A., & Goeman, J. J. (2018). All-Resolutions Inference for brain imaging. *NeuroImage*, *181*, 786–796. https://doi.org/10.1016/j.neuroimage.2018.07.060

Rousselet, G. A. (2025). Using cluster-based permutation tests to estimate MEG/EEG onsets: How bad is it? *European Journal of Neuroscience*, *61*(1), e16618. https://doi.org/10.1111/ejn.16618

385 Rousselet, G. A., Pernet, C. R., Bennett, P. J., & Sekuler, A. B. (2008). Parametric study
386   of EEG sensitivity to phase noise during face processing. *BMC Neuroscience*, *9*(1). https:
387   //doi.org/10.1186/1471-2202-9-98

388 Sassenhagen, J., & Draschkow, D. (2019). Cluster-based permutation tests of MEG/EEG data
389   do not establish significance of effect latency or location. *Psychophysiology*, *56*(6). https:
390   //doi.org/10.1111/psyp.13335

391 Skukies, R., & Ehinger, B. (2021). Modelling event duration and overlap during EEG analysis.
392   *Journal of Vision*, *21*(9), 2037. https://doi.org/10.1167/jov.21.9.2037

393 Skukies, R., Schepers, J., & Ehinger, B. (2024, December 9). *Brain responses vary in duration
394   - modeling strategies and challenges*. https://doi.org/10.1101/2024.12.05.626938

395 Smith, N. J., & Kutas, M. (2014a). Regression-based estimation of ERP waveforms: I. The
396   rERP framework. *Psychophysiology*, *52*(2), 157–168. https://doi.org/10.1111/psyp.12317

397 Smith, N. J., & Kutas, M. (2014b). Regression-based estimation of ERP waveforms: II. Non-
398   linear effects, overlap correction, and practical considerations. *Psychophysiology*, *52*(2),
399   169–181. https://doi.org/10.1111/psyp.12320

400 Smith, S., & Nichols, T. (2009). Threshold-free cluster enhancement: Addressing problems of
401   smoothing, threshold dependence and localisation in cluster inference. *NeuroImage*, *44*(1),
402   83–98. https://doi.org/10.1016/j.neuroimage.2008.03.061

403 Sóskuthy, M. (2017). *Generalised additive mixed models for dynamic analysis in linguistics: A
404   practical introduction*. https://doi.org/10.48550/ARXIV.1703.05339

405 Sóskuthy, M. (2021). Evaluating generalised additive mixed modelling strategies for dynamic
406   speech analysis. *Journal of Phonetics*, *84*, 101017. https://doi.org/10.1016/j.wocn.2020.
407   101017

408 Teichmann, L. (2022). An empirically driven guide on using bayes factors for m/EEG decoding.
409   *Aperture Neuro*, *2*, 1–10. https://doi.org/10.52294/apertureneuro.2022.2.maoc6465

410 Tremblay, A., & Newman, A. J. (2014). Modeling nonlinear relationships in ERP data using
411   mixed-effects regression with R examples. *Psychophysiology*, *52*(1), 124–139. https://doi.
412   org/10.1111/psyp.12299

413 Umlauf, N., Klein, N., & Zeileis, A. (2018). BAMLSS: Bayesian Additive Models for Location,
414   Scale, and Shape (and Beyond). *Journal of Computational and Graphical Statistics*, *27*(3),
415   612–627. https://doi.org/10.1080/10618600.2017.1407325

416 Ushey, K., Allaire, J., & Tang, Y. (2024). *Reticulate: Interface to 'python'*. https://CRAN.
417   R-project.org/package=reticulate

418 Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund,
419   G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M.,
420   Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., ... Yutani, H. (2019). Welcome
421   to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/
422   joss.01686

423 Wickham, H., Pedersen, T. L., & Seidel, D. (2023). *scales: Scale functions for visualization*.
424   https://CRAN.R-project.org/package=scales

425 Wieling, M. (2018). Analyzing dynamic phonetic data using generalized additive mixed model-
426   ing: A tutorial focusing on articulatory differences between L1 and L2 speakers of English.
427   *Journal of Phonetics*, *70*, 86–116. https://doi.org/10.1016/j.wocn.2018.03.002

428 Winter, B., & Wieling, M. (2016). How to analyze linguistic change using mixed models, Growth
429   Curve Analysis and Generalized Additive Modeling. *Journal of Language Evolution*, *1*(1),
430   7–18. https://doi.org/10.1093/jole/lzv003

431 Wood, S. N. (2003). Thin Plate Regression Splines. *Journal of the Royal Statistical Society
432   Series B: Statistical Methodology*, *65*(1), 95–114. https://doi.org/10.1111/1467-9868.00374

433 Wood, S. N. (2017a). *Generalized Additive Models*. Chapman; Hall/CRC. https://doi.org/10.
434   1201/9781315370279

435 Wood, S. N. (2017b). *Generalized additive models: An introduction with r* (2nd ed.). Chapman;

436   Hall/CRC.

437   Wüllhorst, V., Wüllhorst, R., Overmeyer, R., & Endrass, T. (2025). Comprehensive Analysis
438   of Event-Related Potentials of Response Inhibition: The Role of Negative Urgency and
439   Compulsivity. *Psychophysiology*, *62*(2). https://doi.org/10.1111/psyp.70000

440   Xie, Y. (2014). knitr: A comprehensive tool for reproducible research in R. In V. Stodden, F.
441   Leisch, & R. D. Peng (Eds.), *Implementing reproducible computational research*. Chapman;
442   Hall/CRC.

443   Xie, Y. (2015). *Dynamic documents with R and knitr* (2nd ed.). Chapman; Hall/CRC. https:
444   //yihui.org/knitr/

445   Xie, Y. (2023). *knitr: A general-purpose package for dynamic report generation in r.* https:
446   //yihui.org/knitr/

447   Xie, Y., Allaire, J. J., & Grolemund, G. (2018). *R markdown: The definitive guide.* Chapman;
448   Hall/CRC. https://bookdown.org/yihui/rmarkdown

449   Xie, Y., Dervieux, C., & Riederer, E. (2020). *R markdown cookbook.* Chapman; Hall/CRC.
450   https://bookdown.org/yihui/rmarkdown-cookbook

451   Yeung, N., Bogacz, R., Holroyd, C. B., & Cohen, J. D. (2004). Detection of synchronized os-
452   cillations in the electroencephalogram: An evaluation of methods. *Psychophysiology*, *41*(6),
453   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)

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

Figure A1

Exemplary (simulated) group-level average cross-temporal generalisation matrix of decoding performance (ROC AUC).
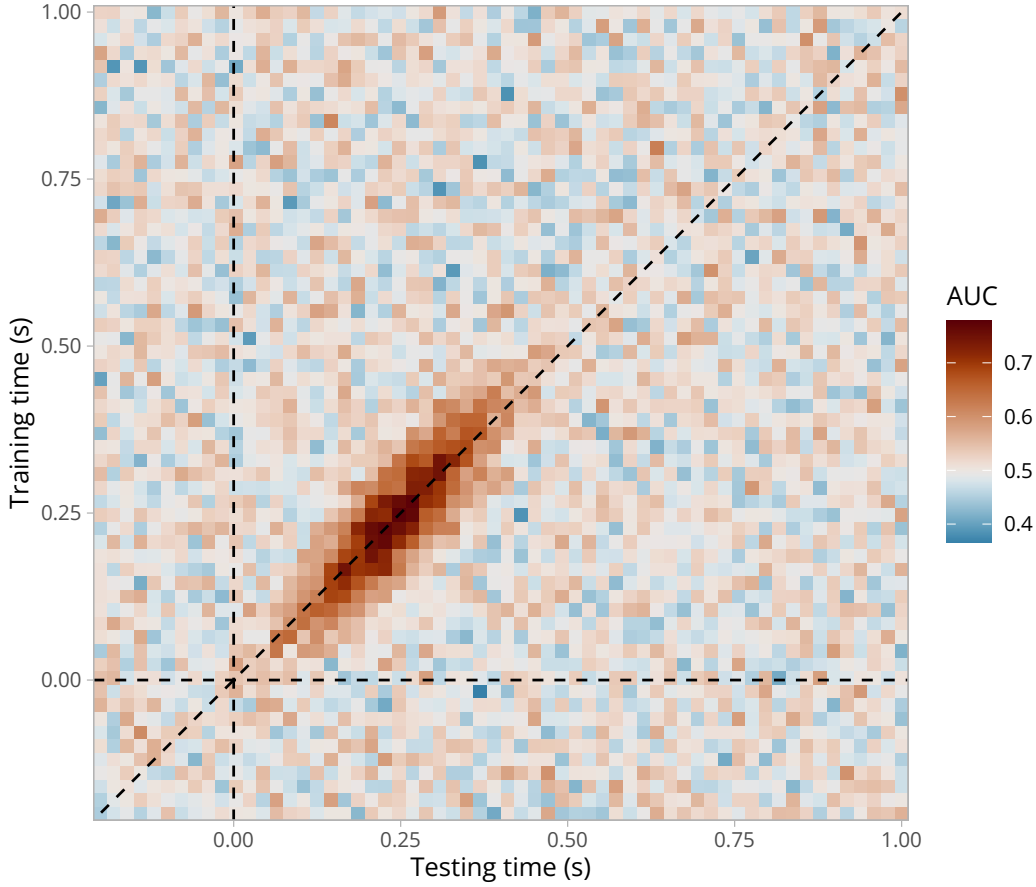
458 To model cross-temporal generalisation matrices of decoding performance (ROC AUC),
459 we extended the initial (decoding) GAM to take into account the bivariate temporal distribution
460 of AUC values, thus producing naturally smoothed estimates (timecourses) of AUC values and
461 posterior probabilities. This model can be written as follows:

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

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

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

467   Note that $b_m(,)$ is a basis function that maps $R \times R \to R$. A popular bivariate basis
468   function uses *thin-plate splines* (Wood, 2003), which extend to $\mathbf{s}_i \in \mathbb{R}^d$ and ∂l_g penalties. These
469   splines are designed to interpolate and approximate smooth surfaces over two dimensions (hence
470   the "bivariate" term). For $d = 2$ dimensions and $l = 2$ (smoothness penalty involving second
471   order derivative):

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

472   using the the radial basis function given by:

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

473   where $\|\mathbf{s}_i - \mathbf{k}_m\|$ is the Euclidean distance between the covariate $\mathbf{s}_i$ and the knot location
474   $\mathbf{k}_m$. We fitted this model using `brms`...

```
# fitting a GAM with two temporal dimensions
timegen_gam <- brm(
    # 2D thin-plate spline (tp)
    auc ~ t2(train_time, test_time, bs = "tp", k = 20),
    data = timegen_data,
    family = Beta(),
    warmup = 1000,
    iter = 2000,
    chains = 8,
    cores = 8,
    file = "models/timegen_gam_t2.rds"
    )


# fitting a GP with two temporal dimensions
# timegen_gp <- brm(
#     auc ~ gp(train_time, test_time, k = 20),
#     data = timegen_data,
#     family = Beta(),
#     control = list(adapt_delta = 0.95),
#     iter = 2000,
#     chains = 4,
#     cores = 4,
#     file = "models/timegen_gp.rds"
#     )
```

Figure A2

Predicted AUC values (left) and posterior probabilities of decoding accuracy being above chance level (right) according to the bivariate GAM.
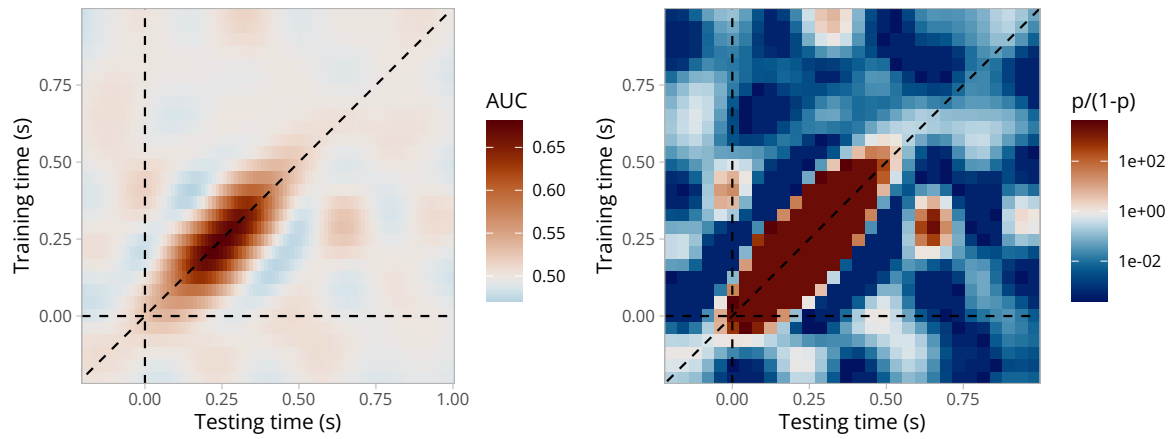
475   Could be extended to spatial and temporal dimensions with formulas such as `te(x, y,`
476   `Time, d = c(2, 1) )`...

Appendix B
Alternative to GAMs: Approximate Gaussian Process regression

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

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

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

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

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

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

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

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

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

505 Taken from https://michael-franke.github.io/Bayesian-Regression/practice-sheets/
506 10c-Gaussian-processes.html: For a given vector $\mathbf{x}$, we can use the kernel to construct finite
507 multi-variate normal distribution associated with it like so:

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

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

511 Low-rank approximate Gaussian processes are of main interest in machine learning and
512 statistics due to the high computational demands of exact Gaussian process models (Riutort-
513 Mayol et al., 2023)...

Appendix C
How to choose the GAM basis dimension?

514 Here provide recommendation about how to define k. An option is to vary k and examine the
515 predictions and posterior predictive checks (PPCs) of each model... In this example (Figure C1)...
516 However, it is not possible to provide general recommendations, as the optimal k depends on
517 the sampling rate, the preprocessing steps (e.g., signal-to-noise ratio, low-pass filtering, etc),
518 and the neural dynamics of the phenomenon under study.

Figure C1

Posterior predictions and posterior predictive checks for the GAM with varying k (in rows).
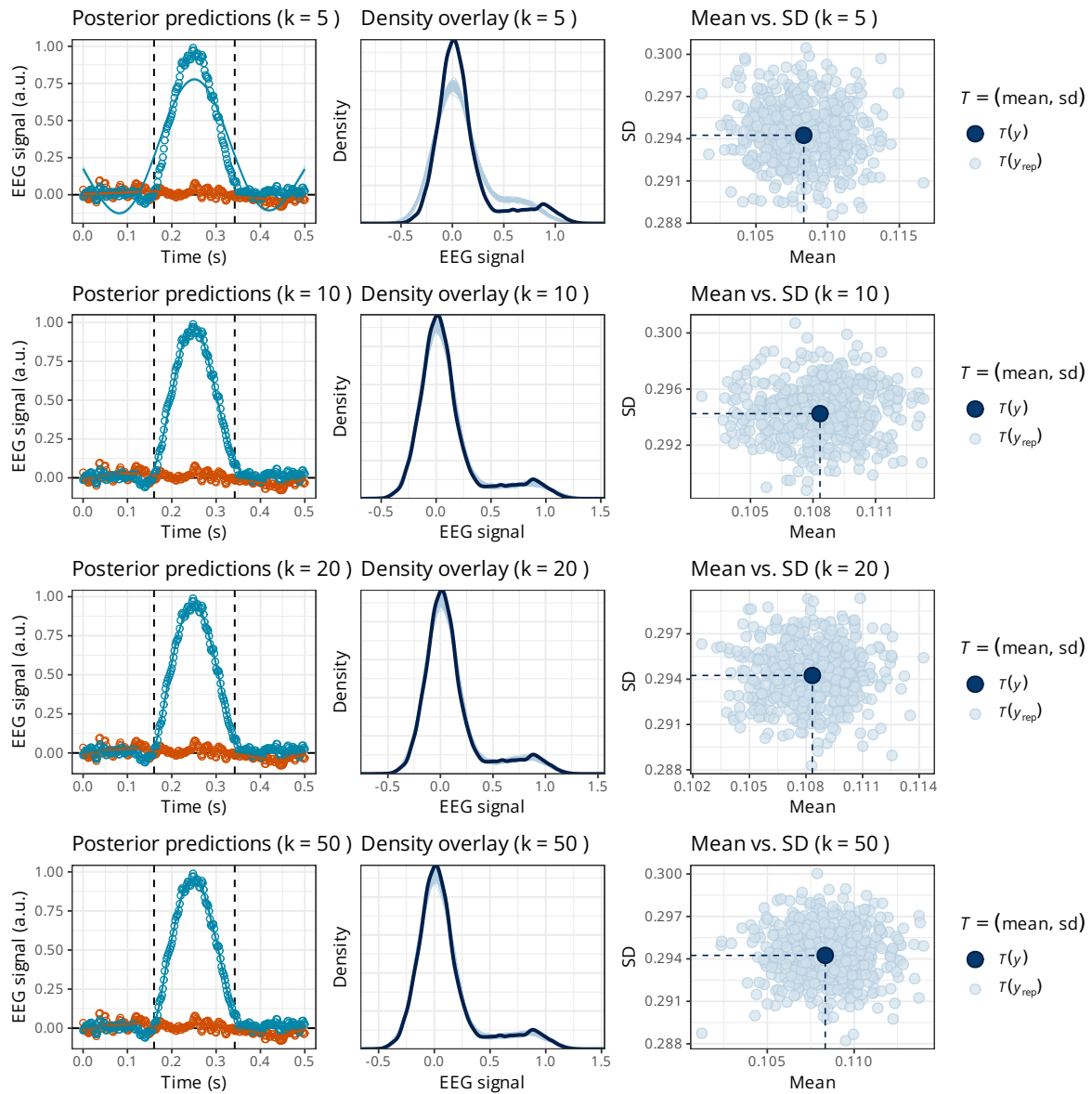
Appendix D
Integration with `MNE-Python`

519 For users who are already familiar with `brms`, the recommended pipeline is to import ERPs
520 or decoding results in `R` and analyse these data using the code provided in the main paper.
521 However, it is also possible to simply call functions from the `neurogam R` package (available at
522 https://github.com/lnalborczyk/neurogam), which come with sensible defaults.

```
# 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
gam_onset_offset <- testing_through_time(
    data = raw_df,
    participant_id = "participant", meeg_id = "eeg",
    time_id = "time", predictor_id = "condition",
    warmup = 1000, iter = 5000, chains = 4, cores = 4
    )


# displaying the results
gam_onset_offset$clusters
```

523 The `neurogam` package can also be called from `Python` using the `rpy2` module, and can
524 easily be integrated into `MNE-Python` pipelines. For example, we use it below to estimate the
525 onset and offset of effects for one channel from a MNE evoked object. Note that the code used
526 to reshape the `sample MNE` dataset is provided later.

```
# loading the Python modules
import rpy2.robjects as robjects
from rpy2.robjects.packages import importr
from rpy2.robjects import pandas2ri
from rpy2.robjects.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(robjects.default_converter + pandas2ri.converter):

    reshaped_df_r = robjects.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,
    post_prob_ratio_threshold=3,
    multilevel=False
    )
```

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

```python
# loading the Python modules
import mne
from mne.datasets import sample
import numpy as np
import pandas as pd

# defining the path to the "sample" dataset
path = sample.data_path()

# loading the evoked data
evokeds = mne.read_evokeds(path / "MEG" / "sample" / "sample_audvis-ave.fif")

# defining a function to reshape the data
def reshape_eeg_channels_as_participants(evokeds, condition_labels):

    all_dfs = []

    for evoked, condition in zip(evokeds, condition_labels):

        # selecting only EEG channels
        picks = mne.pick_types(evoked.info, meg=False, eeg=True)
        data = evoked.data[picks, :]
        channel_names = np.array(evoked.ch_names)[picks]

        n_channels, n_times = data.shape
        # repeating time for each channel
        times = np.tile(evoked.times, n_channels)
        meeg_values = data.flatten()
        pseudo_participant_ids = np.repeat(channel_names, n_times)

        # converting to dataframe
        df = pd.DataFrame({
            "participant": pseudo_participant_ids,
            "time": times,
            "eeg": meeg_values,
            "condition": condition
            })

        all_dfs.append(df)

    return pd.concat(all_dfs, ignore_index=True)

# picking two evoked conditions
faces = evokeds[0]
scrambled = evokeds[1]

# reshaping data by pretending each EEG channel is a participant
reshaped_df = reshape_eeg_channels_as_participants(
```

```
    evokeds=[faces, scrambled],
    condition_labels=["faces", "scrambled"]
    )
```