Blocked Cross-Validation for Generalized Additive Extreme Value Models of Dependent Data

Benjamin D. Youngman
Department of Mathematics and Statistics, University of Exeter, UK
b.youngman@exeter.ac.uk

November 2026

Abstract

Generalized additive model forms let the parameters of extreme value distributions vary with covariates, and their smoothing parameters are typically estimated by restricted maximum likelihood (REML). Environmental extremes are rarely independent, and REML then smooths too little. Smoothing parameters are estimated here by blocked cross-validation, in which folds comprise approximately independent clusters, such as years. As the support of extreme value distributions depends on their parameters, a training fit can exclude validation data, whose likelihood is then zero. This is overcome by giving validation data a small weight in each training fit, so that the criterion is finite and smooth, and its gradient requires only first and second derivatives of the log-likelihood. The same clusters give a one-step jackknife covariance matrix with an allowance for smoothing bias, so no model for dependence is required. In simulations of spatially dependent annual maxima, blocked cross-validation reduced the root mean squared error of 100-year return level estimates by 7% to 9% relative to REML. Where the support constrained training fits, alternatives either gave no estimates or were less accurate, whereas small validation weights were reliable. The approach is implemented in the R package evgam and applied to annual maximum precipitation over Colorado.

Keywords: composite likelihood; jackknife; restricted maximum likelihood; sandwich estimator; smoothing parameter

1 Introduction

This work intends to provide a simple and robust approach to estimating the smoothness of generalized additive model (GAM) forms for extreme value distribution parameters when data are dependent. GAM forms are now routinely used to capture variation in the generalized extreme value (GEV) and generalized Pareto distribution (GPD) parameters with covariates; see, e.g., Chavez-Demoulin & Davison (2005) and Youngman (2019). By coupling GAM forms with the restricted maximum likelihood (REML) approach to inference of Wood (2011) and Wood et al. (2016), Youngman (2019) and Youngman (2022) offer objective and relatively fast estimation of such forms for extreme-value model, which is implemented in the package evgam (Youngman 2026*a*) for R (R Core Team 2026).

Environmental extremes are rarely independent. Exceedances of a high threshold by daily rainfall or wind gusts tend to occur in clusters, and annual maxima at nearby locations are often caused by the same weather systems. Modeling such dependence is typically difficult, and so a common pragmatic

choice is to base inference on the *independence likelihood*, i.e., the product of marginal densities, and then to adjust uncertainty estimates (Lindsay 1988, Chandler & Bate 2007, Varin et al. 2011). Estimates are then consistent but, without adjustment, the information in the data is overstated, so that standard errors are too small and nominal 95% confidence intervals can have coverage far below 95%. When using GAM forms, smoother parameter estimation can also be affected, as REML estimation typically assumes independence. Undersmoothing is a common consequence of this model mis-specification (Opsomer et al. 2001).

Cross-validation can help combat undersmoothing as it relies on out-of-sample predictive performance, which is often what matters in practice. However, to perform reliably it requires that validation data are independent of training data. Wood (2026) shows how a broad class of such criteria, termed neighborhood cross-validation (NCV), can be computed and optimized efficiently for quadratically penalized models, by approximating each fit to data with a neighborhood omitted by a single Newton step from the fit to all data, and shows that omitting a neighborhood of each datum avoids the undersmoothing that short-range autocorrelation otherwise causes. This work considers the case, common for environmental extremes, in which data partition into approximately independent clusters, such as years, but in which dependence within clusters may be strong and long-range, such as between all locations of a network in the same year. Smoothing parameters are then estimated by blocked cross-validation, in which folds comprise entire clusters (Roberts et al. 2017), which is the special case of NCV in which the omitted and predicted are the same and define clusters.

Here, where interest lies in fitting extreme value distributions, a robust cross-validation approach must also allow that distributions can have a finite upper endpoint. If ignored, cross-validation can become unreliable because validation data could exceed an upper endpoint estimated from training data. This is evident from the simulation study of Section 6.

This work proposes a simple and intuitive approach to reliable estimation of smoothing parameters when estimating extreme value distributions that uses all data to estimate distributions, but gives validation a small weight, ϵ. This ensures that they are below the estimated distribution's upper end-point, and in turn that their predictive likelihood is non-zero. Using a small weight, as opposed to a zero weight, avoids parameter estimates lying on boundaries, and gives a differentiable cross-validation objective which can additionally be analytically calculated. Similarly to NCV, and unlike REML, this gradient requires only first and second derivatives of the log-likelihood. Small validation weights are compared with zero weights, i.e. $\epsilon = 0$, and also with one-step and few-step approximations to training fits, including that of Wood (2026), which replaces the validation loss by a quadratic approximation. In this work, the same clusters used for cross-validation are also used to give reliable uncertainty estimates, through a cluster-based sandwich covariance matrix, a one-step jackknife version of it, and the latter with an allowance for smoothing bias. The blocked cross-validation approach described in Section 3 is implemented in the R package evgam, and so is available for all of its extreme value distributions, and is assessed for extremes through simulation, including against NCV, and an application. The small-weight adjustment of Section 3.2 is available for the GEV distribution and GPD.

The next section outlines GAM forms for extreme value distribution parameters and their estimation by REML. Section 3 then introduces blocked cross-validation for smoothing parameter estimation, with small validation weights, and Section 4 uncertainty quantification under dependence. Implementation with evgam is described in Section 5. Section 6 gives simulation studies that focus on performance in extreme value analyses. Section 7 presents an application to return-level estimation for annual maximum precipitation in Colorado. The work is summarized in Section 8. Computational details are given in Appendix A, further details of the implementation in evgam in

Appendix B, further simulation results in Appendix C, and simulation studies of threshold excesses in Appendix D.

2 Generalized additive extreme value models

Let $Y(\boldsymbol{x})$ denote a random variable indexed by covariate $\boldsymbol{x}$. For annual maxima, for example, assume that $Y(\boldsymbol{x}) \sim GEV(\mu(\boldsymbol{x}), \psi(\boldsymbol{x}), \xi(\boldsymbol{x}))$, and for excesses of a high threshold $u(\boldsymbol{x})$ that $Y(\boldsymbol{x}) - u(\boldsymbol{x}) \mid Y(\boldsymbol{x}) > u(\boldsymbol{x}) \sim GPD(\psi_u(\boldsymbol{x}), \xi(\boldsymbol{x}))$. Parameters are related via fixed link functions to $\eta_*(\boldsymbol{x})$, e.g., $\mu(\boldsymbol{x}) = \eta_\mu(\boldsymbol{x})$, $\log \psi(\boldsymbol{x}) = \eta_\psi(\boldsymbol{x})$ and $\xi(\boldsymbol{x}) = \eta_\xi(\boldsymbol{x})$, where $\eta_*(\boldsymbol{x})$ has basis representation

$$\eta_*(\boldsymbol{x}) = \beta_0 + \sum_{k=1}^{K} \sum_{d=1}^{D_k} \beta_{kd} b_{kd}(\boldsymbol{x}) = \mathbf{x}^\top \boldsymbol{\beta},$$

with β_{kd} and b_{kd} basis coefficients and functions, respectively, and $\mathbf{x}^\top$ a row of a design matrix $\mathbf{X}$. Let

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} \log f_*(y_i; \boldsymbol{\beta})$$

be the log-likelihood for data $\boldsymbol{y} = \{y_1, \ldots, y_n\}$ with corresponding covariates $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, where $\boldsymbol{\beta}$ stacks the coefficients for all parameters and f_* denotes the model's density function, e.g., f_{GEV} evaluated at $\mu(\boldsymbol{x}_i)$, $\psi(\boldsymbol{x}_i)$ and $\xi(\boldsymbol{x}_i)$.

Smoothness is controlled through penalties on $\boldsymbol{\beta}$. Let

$$\ell(\boldsymbol{\beta}, \boldsymbol{\lambda}) = \ell(\boldsymbol{\beta}) - \frac{1}{2} \boldsymbol{\beta}^\top \mathbf{S}_{\boldsymbol{\lambda}} \boldsymbol{\beta}, \qquad \mathbf{S}_{\boldsymbol{\lambda}} = \sum_{k=1}^{K} \lambda_k \mathbf{S}_k, \tag{1}$$

be the penalized log-likelihood, where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_K)$ are smoothing parameters and $\mathbf{S}_k$ are penalty matrices, which are determined by the chosen basis functions. Write $\rho_k = \log \lambda_k$ and let $\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}$ maximize $\ell(\boldsymbol{\beta}, \boldsymbol{\lambda})$ for given $\boldsymbol{\lambda}$. Then $\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}$ satisfies

$$\nabla \ell(\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}) - \mathbf{S}_{\boldsymbol{\lambda}} \hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}} = \mathbf{0}. \tag{2}$$

Let $\mathbf{H}(\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}) = -\nabla \nabla^\top \ell(\boldsymbol{\beta}, \boldsymbol{\lambda})|_{\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}}$, henceforth $\mathbf{H}$, and $\mathbf{H}_0 = -\nabla \nabla^\top \ell(\boldsymbol{\beta})|_{\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}}$, so that $\mathbf{H} = \mathbf{H}_0 + \mathbf{S}_{\boldsymbol{\lambda}}$.

If the penalty is viewed as an improper Gaussian prior for $\boldsymbol{\beta}$, then $\mathbf{V}_p = \mathbf{H}^{-1}$ is the Bayesian covariance matrix of $\boldsymbol{\beta}$ (Wahba 1990, Wood et al. 2016). Let

$$\ell(\boldsymbol{\lambda}) = \ell(\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}, \boldsymbol{\lambda}) + \frac{1}{2} \log |\mathbf{S}_{\boldsymbol{\lambda}}|_+ - \frac{1}{2} \log |\mathbf{H}(\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}})| + \text{constant}, \tag{3}$$

be the restricted log-likelihood, which, following Wood (2011), results from integrating $\boldsymbol{\beta}$ out using Laplace's method (Tierney & Kadane 1986), and where $|\mathbf{S}_{\boldsymbol{\lambda}}|_+$ denotes the product of positive eigenvalues of $\mathbf{S}_{\boldsymbol{\lambda}}$. Smoothing parameter estimates, $\hat{\boldsymbol{\lambda}}$, maximize $\ell(\boldsymbol{\lambda})$. Fitting a model therefore involves inner iterations, which give $\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}$ for given $\boldsymbol{\lambda}$, and outer iterations, which give $\hat{\boldsymbol{\lambda}}$.

Often data are dependent, which results in $\ell(\boldsymbol{\beta})$ being mis-specified. Instead, it is an independence likelihood and its curvature can overstate the information in the data. If it does, REML tends to give $\hat{\boldsymbol{\lambda}}$ that are too small.

3 Blocked cross-validation

3.1 Cross-validation log-likelihood

Suppose that data can be partitioned into C clusters, such that data within clusters may be dependent, but that data in different clusters are approximately independent. Clusters might, for example, be calendar years of daily data, or years of annual maxima observed at a network of locations. Clusters are then assigned to folds; let $\mathcal{V}_1, \ldots, \mathcal{V}_M$ denote the indices of data in each fold, so that each fold comprises entire clusters. For fold m, let

$$\ell_{(m)}(\boldsymbol{\beta}) = \sum_{i \in \mathcal{V}_m} \log f_*(y_i; \boldsymbol{\beta}) \quad \text{and} \quad \ell_{(-m)}(\boldsymbol{\beta}) = \sum_{i \notin \mathcal{V}_m} \log f_*(y_i; \boldsymbol{\beta})$$

denote validation and training log-likelihoods, respectively, and let $\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}^{(-m)}$ denote a training fit, which is defined in Section 3.2. Then let

$$\ell_{\mathrm{CV}}(\boldsymbol{\lambda}) = \sum_{m=1}^{M} \ell_{(m)}(\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}^{(-m)}), \tag{4}$$

be the blocked cross-validated log-likelihood, and let $\hat{\boldsymbol{\lambda}}$ denote the values of $\boldsymbol{\lambda}$ at which it is maximized. Log-likelihood $\ell_{\mathrm{CV}}(\boldsymbol{\lambda})$ is the logarithmic score (Gneiting & Raftery 2007) of each training fit's predictive distribution for held-out clusters, summed over folds. Folds may comprise single clusters, i.e., $M = C$, which gives leave-one-cluster-out cross-validation, or, following Arlot & Celisse (2010), may be formed by specifying a ratio of training to validation data and splitting clusters at random accordingly.

Log-likelihood $\ell_{\mathrm{CV}}(\boldsymbol{\lambda})$ is a special case of NCV (Wood 2026), in which, the data omitted from the training fit and the data predicted are set to be the same, and equivalent to the clusters defined here. Holding out entire clusters is key to the use of $\ell_{\mathrm{CV}}(\boldsymbol{\lambda})$ for dependent data. Otherise, if dependent data are split between training and validation sets, the same dependencies in the training data can exist in the validation data, resulting in the same undersmoothing of that can happen with REML (Opsomer et al. 2001, Roberts et al. 2017, Wood 2026). Blocking eliminates this as training data carry information about validation data only through the smooth structure being estimated. As $\ell_{\mathrm{CV}}(\boldsymbol{\lambda})$ is based on the independence likelihood for both training and validation data, no model for dependence is required.

3.2 Ensuring non-zero likelihoods for validation data

The usual training fit maximizes the penalized training log-likelihood, $\ell_{(-m)}(\boldsymbol{\beta}) - \frac{1}{2}\boldsymbol{\beta}^\top \mathbf{S}_{\boldsymbol{\lambda}} \boldsymbol{\beta}$, and so ignores validation data. However, both the GEV and GPD have a finite upper endpoints if $\xi < 0$. A training fit can place validation data above such upper endpoints, resulting in $\ell_{(m)}(\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}^{(-m)}) = -\infty$. Instead, for given $\epsilon \geq 0$, let $\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}^{(-m)}$ maximize

$$\ell_{(-m)}^{\epsilon}(\boldsymbol{\beta}) = \ell_{(-m)}(\boldsymbol{\beta}) + \epsilon\, \ell_{(m)}(\boldsymbol{\beta}) - \tfrac{1}{2}\boldsymbol{\beta}^\top \mathbf{S}_{\boldsymbol{\lambda}} \boldsymbol{\beta}, \tag{5}$$

so that validation data are given weight ϵ in each training fit, relative to the unit weight of training data. The case $\epsilon = 0$ corresponds to omitting validation data, except that $\ell_{(-m)}^{\epsilon}(\boldsymbol{\beta})$ is $-\infty$ unless all data, including validation data, are within the support.

Here is proposed to use a small, positive value for ϵ. Validation data then have negligible influence on training fits, as is the aim of cross-validation. Function $\ell_{(-m)}^{\epsilon}(\boldsymbol{\beta})$ is also expected to be differentiable

at each $\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}^{(-m)}$. As a result $\ell_{\mathrm{CV}}(\boldsymbol{\lambda})$ is differentiable and its gradient has closed form; for underlying mathematical details see Wood et al. (2016). The aim is to avoid training fits on that give validation data on the boundary of an extreme value distribution's support, so that optimization of $\boldsymbol{\lambda}$ can benefit from the gradient of $\ell_{\mathrm{CV}}(\boldsymbol{\lambda})$ with respect to $\boldsymbol{\lambda}$.

–>

The choice of ϵ requires a trade-off as small ϵ can give sharp changes in $\ell_{\mathrm{CV}}(\boldsymbol{\lambda})$ as data move towards or away from the boundary, which make training fits harder to compute, whereas larger ϵ let validation data contribute more to training fits. Sensitivity to choice of ϵ is discussed in Section 6, the result of which is ro recommend $\epsilon = 10^{-2}$, which balanced these effects well.

3.3 Optimization for training data

By default, each training fit is computed exactly, through inner Newton iterations started from its previous estimates (Appendix A.1). Wood (2026) instead approximates each training fit by a single Newton step from the fit to all data, which, when each of many small neighborhoods is omitted in turn, reduces the cost of evaluating NCV to about that of a single fit. Its accuracy is assessed in Section 6 for folds that comprise single clusters, and for folds of about $1/M$ of the data, together with that of two Newton steps. As the fit to all data includes validation data, its support includes them, and so a Newton step can be halved until validation data are within the support. Alternatively, Wood (2026) replaces each validation log-likelihood by its quadratic approximation about the fit to all data, which is finite for all $\boldsymbol{\beta}$. Small validation weights retain the exact validation log-likelihood, and apply equally to exact and approximate training fits.

A BFGS quasi-Newton method, as implemented in evgam, is used to maximize $\ell_{\mathrm{CV}}(\boldsymbol{\lambda})$ for outer iterations. For exact training fits, its gradient with respect to ρ_k follows from implicit differentiation of the score equation of $\ell^{\epsilon}_{(-m)}(\boldsymbol{\beta})$ and, unlike $\ell(\boldsymbol{\lambda})$, requires only first and second derivatives of the log-likelihood, which are already calculated during the inner iterations. For approximate training fits, the gradient is found by finite differences. Details are given in Appendix A.

4 Uncertainty quantification under dependence

4.1 Sandwich covariance matrix

For independent data, uncertainty in $\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}$ can be quantified through $\mathbf{V}_p = \mathbf{H}^{-1}$. However, this give unreliable uncertainty estimates for dependent data because the curvature of the independence log-likelihood, $\mathbf{H}_0$, overstates the information in the data (Chandler & Bate 2007, Varin et al. 2011).

Instead, uncertainty estimates can be based on the sandwich estimator,

$$\mathbf{V}_s = \mathbf{H}^{-1}\hat{\mathbf{J}}\mathbf{H}^{-1}, \tag{6}$$

where $\hat{\mathbf{J}}$ is an estimate of the variance of the score, $\nabla\ell(\boldsymbol{\beta})$ (Godambe 1960, White 1982). Choosing clusters to be approximately independent mans that $\mathbf{J}$ can be estimated from cluster-based scores. Let $\nabla\ell_c(\boldsymbol{\beta})$ denote the gradient of $\ell(\boldsymbol{\beta})$ summed over the data in cluster c, and let

$$\mathbf{u}_c = \nabla\ell_c(\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}) - \frac{1}{C}\mathbf{S}_{\boldsymbol{\lambda}}\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}, \qquad \hat{\mathbf{J}} = \sum_{c=1}^{C}\mathbf{u}_c\mathbf{u}_c^{\top},$$

so that $\mathbf{u}_c$ is cluster c's contribution to the penalized score, and the penalty is split across clusters. Inclsuding the penalty term ensures that $\sum_c \mathbf{u}_c = \mathbf{0}$. Note that if there are fewer clusters than

coefficients then $\hat{\mathbf{J}}$ is rank-deficient with rank at most $C-1$. Also, $\mathbf{V}_s$ characterizes the variation of $\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}$ about its expectation and therefore does not account for smoothing bias. By considering independent data, as $\hat{\mathbf{J}}$ estimates $\mathbf{H}_0$, then $\mathbf{V}_s \approx \mathbf{V}_f = \mathbf{H}^{-1}\mathbf{H}_0\mathbf{H}^{-1}$, which is smaller than $\mathbf{V}_p$ by $\mathbf{H}^{-1}\mathbf{S}_{\boldsymbol{\lambda}}\mathbf{H}^{-1}$. This difference must be allowed for in uncertainty estimates in order to achieve good coverage (Nychka 1988, Marra & Wood 2012). The balance of these effects for extreme value GAMs is assessed in Section 6.

4.2 Adjusting for relatively few clusters

Cluster-based sandwich covariance matrices tend to underestimate variance when there are few clusters, or when some clusters have high leverage (Bell & McCaffrey 2002). A jackknife alternative is as follows. For cluster c, let $\mathbf{H}_{0,c} = -\nabla\nabla^\top \ell_c(\boldsymbol{\beta})|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}}$ be its contribution to $\mathbf{H}_0$, and let $\boldsymbol{\delta}_c = (\mathbf{H} - \mathbf{H}_{0,c})^{-1}\nabla\ell_c(\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}})$, so that $\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}} - \boldsymbol{\delta}_c$ is one Newton step from $\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}$ towards the estimate with cluster c removed and $\boldsymbol{\lambda}$ fixed. Then let

$$\mathbf{V}_j = \frac{C-1}{C}\sum_{c=1}^{C}(\boldsymbol{\delta}_c - \bar{\boldsymbol{\delta}})(\boldsymbol{\delta}_c - \bar{\boldsymbol{\delta}})^\top, \tag{7}$$

where $\bar{\boldsymbol{\delta}} = C^{-1}\sum_c \boldsymbol{\delta}_c$, be the one-step jackknife covariance matrix. Replacing $\mathbf{H}$ by $\mathbf{H} - \mathbf{H}_{0,c}$ inflates each cluster's contribution to $\mathbf{V}_s$ according to its leverage, and for linear regression $\mathbf{V}_j$ is the CR3 covariance matrix of Bell & McCaffrey (2002). It requires only second derivatives of the log-likelihood, and one further Cholesky decomposition for each cluster. The step $\boldsymbol{\delta}_c$ is the one-step approximation to the change in estimates on omitting data used by NCV (Wood 2026), who notes that jackknife estimates of variance are too small when omitted data are correlated with retained data, and corrects for this using cross-validated residuals. Here, as clusters are approximately independent, no such correlation is present by construction. A further version evaluates each cluster's score at $\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}} - \boldsymbol{\delta}_c$, i.e., uses cross-validated scores, and is consistent with $\mathbf{V}_j$ to first order. It gave intervals within about 2% of the width of those from $\mathbf{V}_j$ in simulations, except for distributions with bounded support, for which $\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}} - \boldsymbol{\delta}_c$ can place cluster c's data close to the boundary of the support, and this version then gave very wide intervals. It is therefore not considered further.

4.3 Smoothing bias

As noted in Section 4.1, $\mathbf{V}_s$ and $\mathbf{V}_j$ omit $\mathbf{H}^{-1}\mathbf{S}_{\boldsymbol{\lambda}}\mathbf{H}^{-1}$, and so an allowance for smoothing bias adds it back, giving, for example,

$$\mathbf{V}_b = \mathbf{V}_j + \alpha\,\mathbf{H}^{-1}\mathbf{S}_{\boldsymbol{\lambda}}\mathbf{H}^{-1}, \tag{8}$$

for some $\alpha \geq 1$. As for Bayesian intervals, intervals based on $\mathbf{V}_b$ target coverage averaged across covariate values, rather than pointwise coverage, and so may still under-cover where the quantity of interest varies sharply, and over-cover elsewhere. Taking $\alpha = 1$ adds the allowance appropriate for independent data. Wood (2026) proposes instead to divide the allowance by a learning rate, which, in the form used here, gives

$$\alpha = \max\{\mathrm{tr}(\mathbf{V}_j)/\mathrm{tr}(\mathbf{V}_f), 1\},$$

the ratio of the scale of a covariance matrix that accounts for dependence to that of one that does not. Under positive dependence $\alpha > 1$, and so this inflates the allowance in line with the information lost to dependence. Both are assessed in Section 6, in which α was typically between 7 and 12, and are termed the *unscaled* and *scaled* allowances, respectively.

4.4 Smoothing parameter uncertainty

Uncertainty estimates based on $\mathbf{V}_s$, $\mathbf{V}_j$ or $\mathbf{V}_b$ assume that $\hat{\boldsymbol{\lambda}}$ is known. Let $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_K)$ and let $\mathbf{V}_\rho$ be a covariance matrix for $\hat{\boldsymbol{\rho}}$. Following Wood et al. (2016), uncertainty in $\hat{\boldsymbol{\rho}}$ can be propagated through the first-order correction

$$\mathbf{V}_c = \mathbf{V} + \mathbf{D}\mathbf{V}_\rho\mathbf{D}^\top, \qquad \mathbf{D} = \partial\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}/\partial\boldsymbol{\rho}^\top,$$

with $\mathbf{V}$ given by $\mathbf{V}_p$, $\mathbf{V}_s$, $\mathbf{V}_j$ or $\mathbf{V}_b$, where $\mathbf{D}$ follows from implicit differentiation of (2). For REML, $\mathbf{V}_\rho$ is given by Wood et al. (2016). For cross-validation with exact training fits, $\hat{\boldsymbol{\rho}}$ can also be considered an M-estimator: the gradient of $\ell_{\mathrm{CV}}(\boldsymbol{\lambda})$ is a sum of contributions, $\boldsymbol{\psi}_c$, from each held-out cluster, which gives $\mathbf{V}_\rho \approx \mathbf{G}^{-1}(\sum_c \boldsymbol{\psi}_c\boldsymbol{\psi}_c^\top)\mathbf{G}^{-1}$, where $\mathbf{G}$ is the negative Hessian of $\ell_{\mathrm{CV}}(\boldsymbol{\lambda})$ with respect to $\boldsymbol{\rho}$, which is found by differencing that gradient. This approximation ignores dependence between contributions from clusters held out together, which share a training fit, requires the optimization of $\ell_{\mathrm{CV}}(\boldsymbol{\lambda})$ to have converged, and so should be considered only a rough guide. It is not used for the results of Sections 6 and 7, whose intervals all condition on $\hat{\boldsymbol{\lambda}}$. Where smoothing parameter uncertainty is important, a cluster bootstrap of the entire procedure is recommended.

5 Implementation in evgam

Blocked cross-validation and cluster-based covariance matrices are implemented in evgam (Youngman 2022) through functions `cvevgam()` and `evgam_sandwich()`, respectively, and are therefore available for all of evgam's distributions. Both functions are included from version 1.0.4 of evgam, which is available from the Comprehensive R Archive Network (CRAN) (Youngman 2026*a*), and the latest development version is available from GitHub at https://github.com/byoungman/evgam. An example call is

```
fit <- cvevgam(list(prcp ~ s(x, y) + s(elev), ~ s(x, y), ~ 1),
               data = prcp_df, family = "gev2",
               folds = 5, group = "year", seed = 1, rho0.offset = 2)
fit_jk <- evgam_sandwich(fit, id = "year", type = "jackknife",
                         bias = "scaled")
```

which gives five-fold blocked cross-validation with folds of entire years, small validation weights, $\epsilon = 10^{-2}$, which are the default for families `"gev2"` and `"gpd2"`, and then $\mathbf{V}_b$ with the scaled allowance. By default, $\boldsymbol{\rho}$ starts from its REML estimates plus `rho0.offset`, which is zero by default. Positive values, such as `rho0.offset = 2` used in Sections 6 and 7, start from smoother fits, which, for distributions with bounded support, makes training fits less likely to exclude validation data at the start. Both functions return objects of class `evgam`, and so `predict()`, `simulate()`, `plot()` and `summary()` can be used as usual. Further arguments, e.g., for other folds and for approximate training fits, and the use of mgcv for NCV, are described in Appendix B.

6 Simulation studies

6.1 Design

These studies mimic the application of Section 7, in which return levels are estimated from annual maxima at a network of dependent locations; studies of threshold excesses are given in Appendix D. Each study comprises 100 simulations, each of which comprises the following steps.

1. Draw $J = 200$ locations, $x_1 < \cdots < x_J$, uniformly on $[0, 1]$.
2. Let $W(x)$ be a Gaussian process with zero mean, unit variance and correlation function $\exp\{-(|x - x'|/\phi_1)^{\delta_1}\}$, where $\phi_1 = 0.2$ and $\delta_1 \approx 2$, and let $\psi(x) = \psi_0 \exp\{0.25W(x)\}$, which varies smoothly with x, with a few peaks and troughs on $[0, 1]$.
3. For each of 50 independent years, which form $C = 50$ clusters, simulate $Y(x_1), \ldots, Y(x_J)$ from a Gaussian copula with correlation function $\exp(-|x - x'|/\phi_2)$, where $\phi_2 = 0.045$, and margins $\Pr\{Y(x) \leq y\} = F_{\text{GPD}}(y; \psi(x), \xi)^{365}$, i.e., the maximum of 365 independent daily values, which is very close to $GEV(\mu(x), \sigma(x), \xi)$.
4. Fit $Y(x) \sim GEV(\mu(x), \sigma(x), \xi)$, with $\mu(x) = g_\mu(x)$, $\log \sigma(x) = g_\sigma(x)$ and ξ constant, where g_μ and g_σ are cubic regression splines of rank 30, with smoothing parameters estimated by each approach below. If REML fails, return to step 1.
5. For each fit and covariance matrix $\mathbf{V}$, estimate $z_p(x_j)$, which satisfies $\Pr\{Y(x_j) \leq z_p(x_j)\} = 1 - p$, with $p = 0.01$, and give a 95% confidence interval by the 2.5% and 97.5% quantiles of $z_p(x_j)$ for 5000 samples from $N(\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}, \mathbf{V})$, which conditions on $\hat{\boldsymbol{\lambda}}$.

For the GEV distribution, shapre parameters $\xi = 0.15$, and $\xi = -0.4$ are used, for which annual maxima have an unbounded and bounded upper tail, respectively. Constant ψ_0 is chosen so that the GEV location parameter is about 33.5 on average, e.g., $\psi_0 = 3.53$ for $\xi = 0.15$. Smoothing parameters are estimated by

- REML;
- blocked cross-validation, with years as clusters, whose versions differ in their folds: five groups of ten years assigned at random ("five-fold"), single years ("leave-one-year-out"), or ten random splits of years with ratio 4:1 ("random splits"); in their training fits: exact, or approximated by one or two Newton steps or by the quadratic approximation of Wood (2026); and in their treatment of validation data: omitted, or given weight $\epsilon = 0$, 10^{-4}, 10^{-3} or 10^{-2} (Section 3.2), with $\epsilon = 10^{-2}$ unless stated otherwise. Each starts from REML estimates of $\boldsymbol{\rho}$ plus 2, i.e., `rho0.offset = 2` (Section 5);
- NCV, using `gam()` of the R package mgcv (version 1.9-5) with family `gevlss`, the same model and neighborhoods comprising entire years, and, for $\xi = -0.4$, also QNCV, through `method = "QNCV"`.

Uncertainty is quantified through $\mathbf{V}_p$, $\mathbf{V}_s$, $\mathbf{V}_j$ and $\mathbf{V}_b$ with the unscaled and scaled allowances, with years as clusters, and, for NCV, through the covariance matrix of Wood (2026). To separate NCV's smoothing parameter estimates from its covariance matrix, evgam is also fitted with smoothing parameters fixed at those of NCV. Accuracy is summarized by root mean squared error (RMSE), $\{J^{-1} \sum_j [\hat{z}_p(x_j) - z_p(x_j)]^2\}^{1/2}$, and two approaches by the geometric mean over simulations of the ratio of their RMSEs, termed the *paired RMSE ratio*. Coverage of 95% confidence intervals is averaged over locations and then over simulations. Monte Carlo uncertainty is quantified through the variability of each summary over simulations.

6.2 Annual maxima with unbounded upper tail

Blocked cross-validation smoothed more than REML, and was more accurate. In the median, log smoothing parameters from leave-one-year-out cross-validation exceeded those of REML by 2.7 for g_μ and 3.5 for g_σ, which about halved the effective degrees of freedom, and every version reduced the RMSE of REML by about 9%, as did NCV (Table 1). The additional smoothing flattens peaks (Figure 1b): for the highest 10% of $z_{0.01}(x)$ in each simulation, bias was -3.1 for leave-one-year-out cross-validation and -3.6 for NCV, against -0.5 for REML.

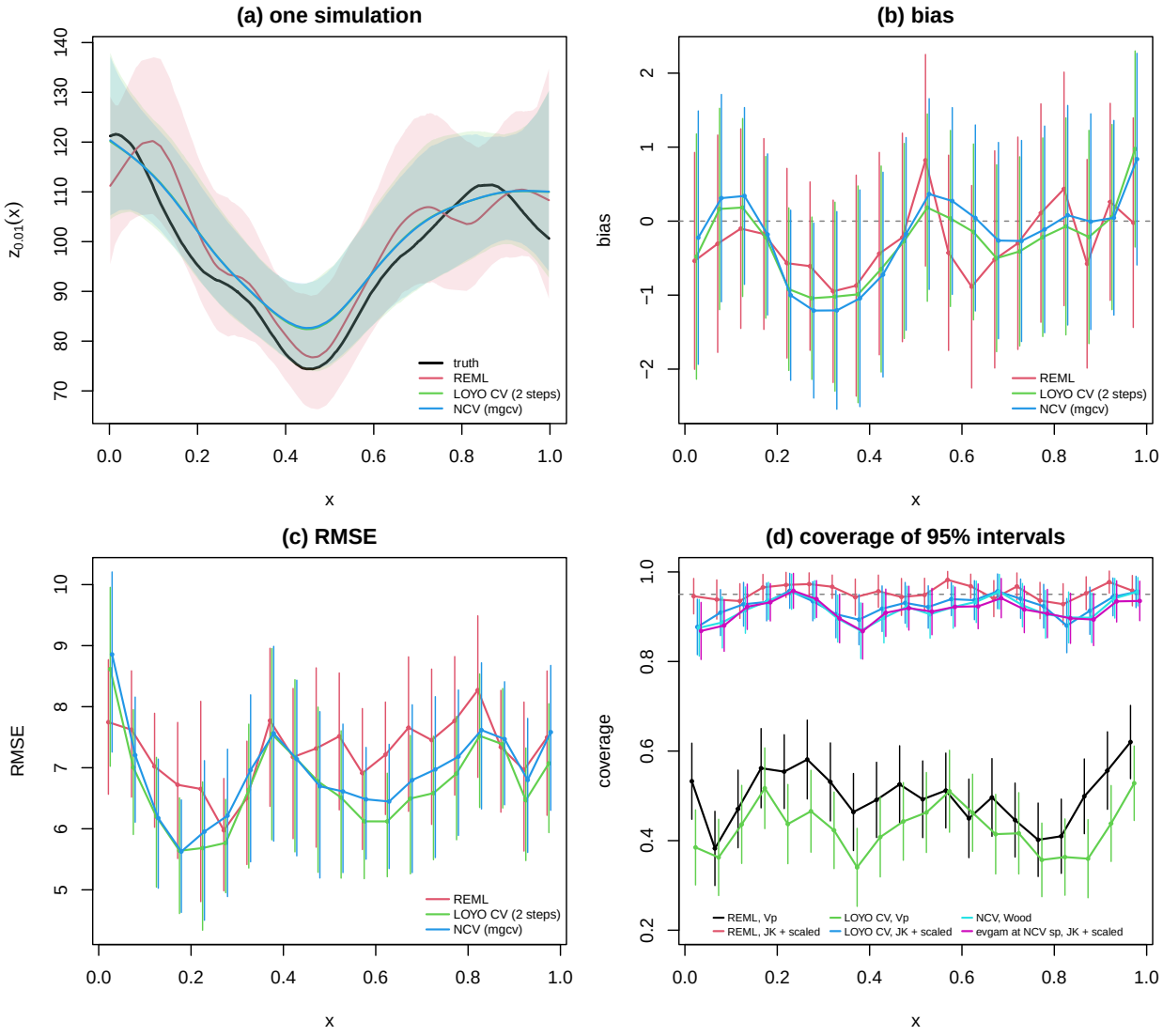


Figure 1: Simulation study of annual maxima, $\xi = 0.15$: (a) true $z_{0.01}(x)$ for a single simulation with its estimates and 95% confidence intervals from REML and leave-one-year-out cross-validation with two Newton steps (LOYO CV), both based on $\mathbf{V}_b$ with the scaled allowance, and from NCV, based on the covariance matrix of Wood (2026); (b) bias and (c) RMSE of estimates of $z_{0.01}(x)$ against x, in bins of width 0.05; and (d) coverage probability of 95% confidence intervals against x, for each approach, where JK denotes $\mathbf{V}_b$ with the scaled allowance, and Wood the covariance matrix of Wood (2026). Vertical bars give approximate 95% Monte Carlo intervals.

Table 1: Simulation study of annual maxima, $\xi = 0.15$: paired RMSE ratio of $z_{0.01}(x)$ estimates relative to REML, with 95% Monte Carlo intervals; coverage probability of 95% confidence intervals based on $\mathbf{V}_p$, $\mathbf{V}_j$ and $\mathbf{V}_b$ with unscaled and scaled allowances; and mean time per fit. †For NCV, coverage is based on the covariance matrix of Wood (2026). Monte Carlo standard errors of coverage probabilities are about 0.01.

Smoothing parameters	RMSE ratio	$\mathbf{V}_p$	$\mathbf{V}_j$	Unscaled	Scaled	Time (s)
REML	1	0.49	0.93	0.93	0.95	2
Leave-one-year-out, two steps	0.906 (0.881, 0.931)	0.42	0.90	0.90	0.92	43
Leave-one-year-out, quadratic	0.906 (0.881, 0.933)	0.41	–	0.90	0.92	14
Five-fold, exact	0.911 (0.888, 0.935)	0.42	–	0.90	0.92	11
NCV	0.912 (0.881, 0.945)	–	–	–	$0.92^\dagger$	61
evgam at NCV's $\hat{\boldsymbol{\lambda}}$	0.912 (0.881, 0.945)	0.41	–	0.89	0.91	–

The versions of blocked cross-validation differed little (Appendix C). Two Newton steps reproduce exact leave-one-year-out cross-validation in about half the time, and one step and the quadratic approximation changed estimates of $z_{0.01}(x)$ by less than 1%. As the support did not constrain training fits, the treatment of validation data made no difference. Splines of rank 50 changed RMSEs by at most 0.3%.

Coverage was determined mainly by the covariance matrix. Intervals based on $\mathbf{V}_p$, which ignores dependence within years, had coverage below 0.5 (Table 1). $\mathbf{V}_j$ increased this to 0.93 for REML and 0.90 for leave-one-year-out cross-validation, and $\mathbf{V}_b$ with the scaled allowance to 0.95 and 0.92, whereas the unscaled allowance added little. The covariance matrix of NCV gave similar coverage. For cross-validation, the shortfall is due to bias at peaks, where coverage was 0.89, rather than to narrow intervals, as the median interval width was about 3.92×RMSE. Cross-validation therefore gave more accurate point estimates, and REML better-calibrated intervals.

6.3 Annual maxima with bound upper tail

Where the support constrained training fits, small validation weights were needed. With $\xi = -0.4$, the largest data were close to the upper endpoint: for the REML fit, the smallest value of $1 + \xi\{y - \mu(x)\}/\sigma(x)$, which is zero at the endpoint, was 0.026 in the median simulation. Omitting validation data then gave a criterion that was not finite for 98% of simulations (Table 2). With $\epsilon = 0$, training fits were on the boundary of the support, where the gradient of $\ell^\epsilon_{(-m)}(\boldsymbol{\beta})$ was about 200 rather than zero, and so the optimization converged for only 3% of simulations, and estimates were about 6% less accurate than with $\epsilon = 10^{-2}$, for which it always converged, in 23 seconds per fit, against 4 seconds for REML. Smaller ϵ also always converged but, as anticipated in Section 3.2, its training fits were harder to compute: at the selected smoothing parameters, some had not converged to the tolerance used for inner iterations in 74% and 91% of simulations for $\epsilon = 10^{-3}$ and 10^{-4}, against 50% for 10^{-2}, and estimates were about 1% less accurate.

Approximate training fits were less reliable or less accurate. With one Newton step, step halving makes the criterion change sharply with $\boldsymbol{\rho}$, and the optimization converged for only 47% of simulations. The quadratic approximation of Wood (2026), as implemented in evgam, converged for 99%, but was about 4% less accurate than $\epsilon = 10^{-2}$, as it replaces, rather than approximates, the validation log-likelihood where the support constrains training fits.

NCV and QNCV implemented in mgcv performed poorly where the support constrained training fits. NCV gave no estimates for 98 of 100 simulations, because its estimates put validatoin data

Table 2: Simulation study of annual maxima with $\xi = -0.4$: for five-fold cross-validation with each treatment of validation data, and leave-one-year-out cross-validation with approximate training fits, the percentage of simulations for which the criterion was not finite at its starting values (Fail) or its optimization converged (Converged), the paired RMSE ratio of $z_{0.01}(x)$ estimates relative to REML, with 95% Monte Carlo intervals, and the coverage probability of 95% confidence intervals based on $\mathbf{V}_b$ with the scaled allowance, over all locations (All) and over the 10% of locations with the highest $z_{0.01}(x)$ in each simulation (Top 10%). For REML, coverage is 0.93 and 0.92. †NCV of Wood (2026), using mgcv, leaving out one year at a time; Fail gives the percentage of simulations for which `gam()` gave no estimates as its criterion was not finite. ‡Coverage is based on the covariance matrix of Wood (2026).

				Coverage	
Approach	Fail	Converged	RMSE ratio	All	Top 10%
Validation data omitted	98	–	–	–	–
$\epsilon = 0$	1	3	0.981 (0.938, 1.026)	0.93	0.88
$\epsilon = 10^{-4}$	0	100	0.939 (0.912, 0.966)	0.94	0.89
$\epsilon = 10^{-3}$	0	99	0.937 (0.910, 0.965)	0.94	0.89
$\epsilon = 10^{-2}$	0	100	0.928 (0.903, 0.955)	0.94	0.89
Leave-one-year-out, one step	0	47	0.927 (0.905, 0.948)	0.94	0.90
Leave-one-year-out, quadratic	0	99	0.966 (0.938, 0.995)	0.93	0.86
NCV†	98	1	–	–	–
QNCV†	0	11	1.854 (1.483, 2.318)	0.79‡	–

outside of the fitted distribution's support, resulting in and undefined log-likelihood. QNCV always gave a finite criterion, even though exact training fits at its smoothing parameters put validation data outside the support for 95% of simulations, but its optimization converged for only 11%, and its RMSE was almost twice that of REML. The quadratic approximation as implemented in evgam has the same criterion but converged for 99% of simulations and was far more accurate, and so the difference lies in their computation, probably mainly the optimization of QNCV.

Coverage with the scaled allowance was close to 0.95 for all approaches other than QNCV, but again lowest at peaks: for the highest 10% of $z_{0.01}(x)$, it was 0.89 with $\epsilon = 10^{-2}$, against 0.92 for REML. Simulations of threshold excesses (Appendix D) gave the same findings about validation weights.

7 Application: Colorado annual maximum precipitation

The approach is now used to estimate return levels for annual maximum daily precipitation over Colorado, US, from the NA-CORDEX ensemble of regional climate model simulations (Mearns et al. 2017). Annual maxima are of daily precipitation simulated by WRF driven by GFDL-ESM2M for the historical period, on a regular grid of spacing 0.22°, and are available in R package evgmrf (Youngman 2026*b*) as dataset `COorder`. They comprise 56 years of annual maxima (1950-2005) for 448 grid cells, which is a 28 by 16 longitude/latitude grid covering Colorado, and gives 25088 annual maxima. Annual maxima at nearby grid cells in a given year are often caused by the same storms, and are strongly dependent: Spearman's correlation between annual maxima at adjacent grid cells averages 0.76, and at grid cells four apart 0.3, and it is still 0.11 at grid cells ten apart. Different years are, however, approximately independent, and so years are treated as independent clusters, and grid cells within years as dependent, i.e., $C = 56$.

Assume that $Y(\boldsymbol{x}) \sim GEV(\mu(\boldsymbol{x}), \psi(\boldsymbol{x}), \xi)$, where $\boldsymbol{x}$ comprises longitude, x, latitude, y, and stan-

dardized elevation, e, with $\mu(\boldsymbol{x}) = g_{\mu,1}(x, y) + g_{\mu,2}(e)$, $\log \psi(\boldsymbol{x}) = g_\psi(x, y)$ and ξ constant, where $g_{\mu,1}$ and g_ψ are thin plate regression splines of rank 30, and $g_{\mu,2}$ is a thin plate regression spline of rank 10. The GEV is parametrized so that $-1 < \xi < 0.5$. Smoothing parameters are estimated by REML and by blocked cross-validation with five folds of entire years, with small validation weights, $\epsilon = 10^{-2}$. REML is used as it would be if dependence were ignored, i.e., with uncertainty quantified through $\mathbf{V}_p$. For blocked cross-validation, uncertainty is quantified through $\mathbf{V}_b$ with the scaled allowance, i.e., the jackknife covariance matrix $\mathbf{V}_j$ with the scaled allowance for smoothing bias. Intervals condition on $\hat{\boldsymbol{\lambda}}$.

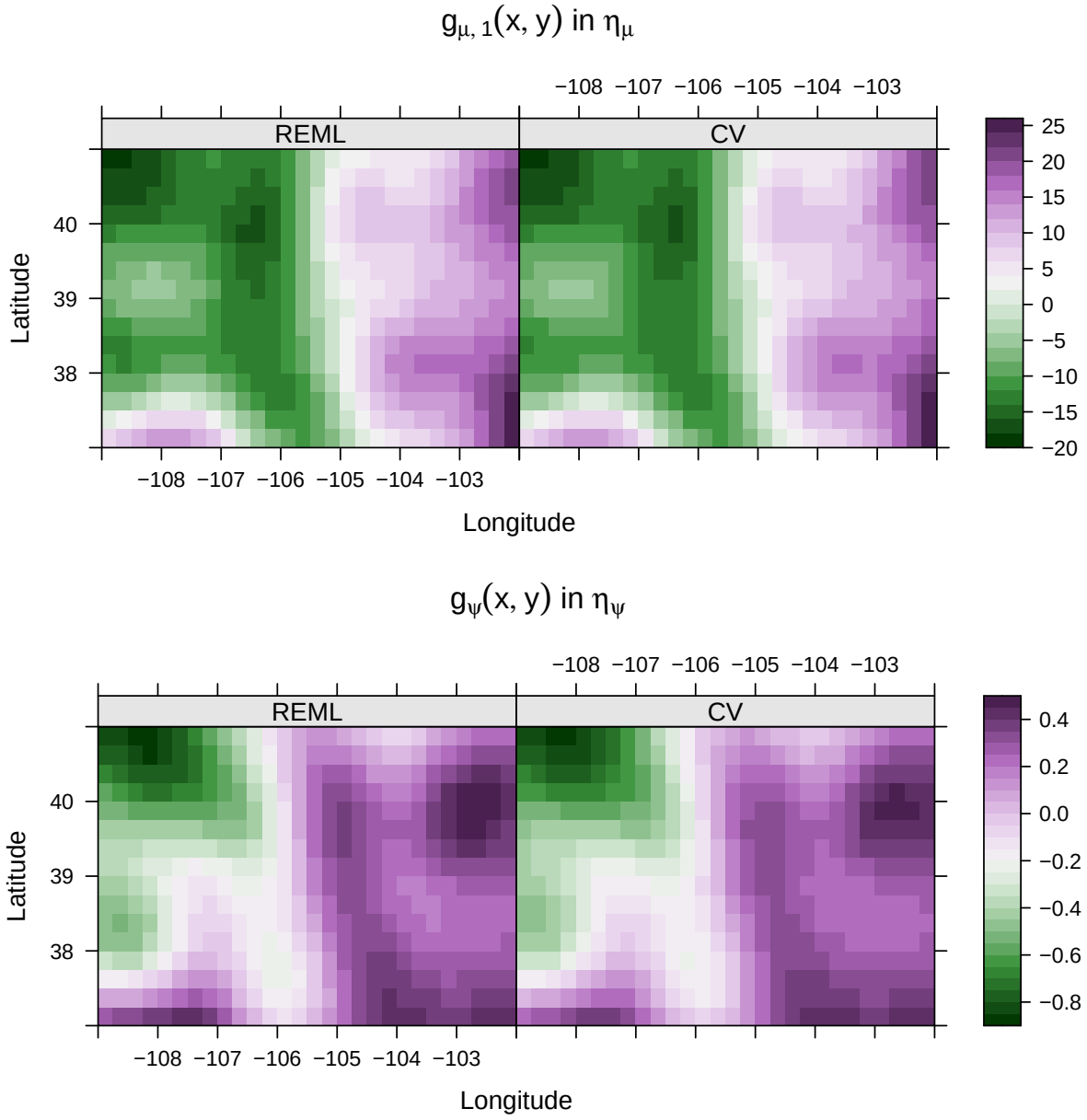


Figure 2: Estimated spatial smooths, $g_{\mu,1}(x, y)$ in η_μ (top) and $g_\psi(x, y)$ in η_ψ (bottom), from REML (left) and blocked cross-validation (right), each centered to have mean zero over the grid. Each row shares a color scale.

Figures 2 and 3 show the estimated spatial smooths and elevation effect from the two fits. No

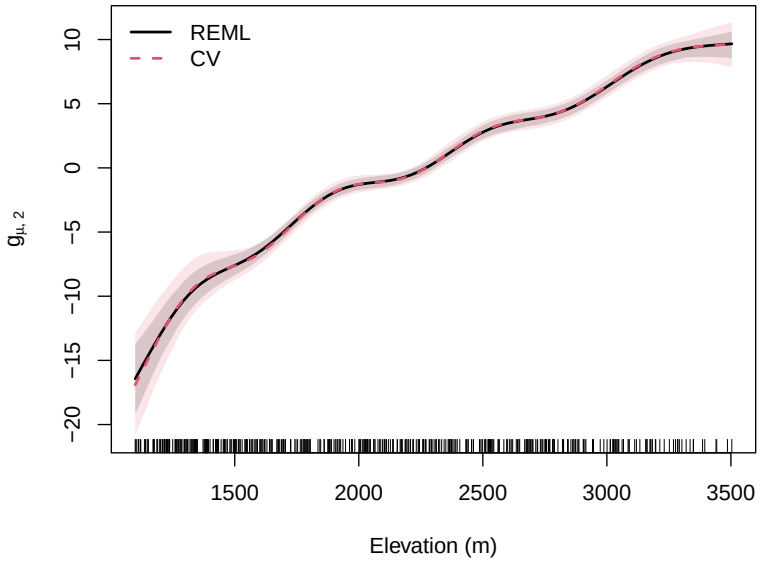


Figure 3: Estimated elevation effect, $g_{\mu,2}$, in η_μ, from REML (solid) and blocked cross-validation (dashed), with pointwise 95% confidence intervals based on $\mathbf{V}_p$ for REML and on the jackknife covariance matrix with the scaled allowance for smoothing bias for blocked cross-validation. Each curve is centered to have mean zero over the range of elevation shown. Ticks show grid cells' elevations.

meaningful difference is evident between the estimated smooths, but a clear increase in uncerainty estimates can be seen in the trend with elevation.

The model has 70 coefficients and there are 56 clusters, and so $\hat{\mathbf{J}}$, and hence $\mathbf{V}_s$ and $\mathbf{V}_j$, have rank at most 55. The cross-validated fit has 65.6 effective degrees of freedom, and so in some directions in which it varies, $\mathbf{V}_s$ and $\mathbf{V}_j$ give no variance. The allowance for smoothing bias in $\mathbf{V}_b$ adds variance in every penalized direction, so that $\mathbf{V}_b$ has rank 70, but in those directions it is the only source of variance, and so standard errors of quantities that depend strongly on them may be too small. Fewer effective degrees of freedom than clusters, e.g., through a smaller basis, or more clusters, would avoid this.

Figure 4 maps 100-year return level estimates, i.e., z_p for $p = 0.01$ (Youngman 2022, Section 2.2), from REML and blocked cross-validation, together with their standard errors, the latter of which account for dependence between grid cells within years. Return level estimates from blocked cross-validation differ from those from REML by between -4.6% and 3.8%. For blocked cross-validation, the median ratio of standard errors based on $\mathbf{V}_s$ to those based on $\mathbf{V}_p$ is 2.9; $\mathbf{V}_j$ increases standard errors by a further factor of 1.02, and the scaled allowance of $\mathbf{V}_b$ by 0.984. Overall, standard errors from blocked cross-validation are 2.8 times those from REML in the median, which indicates how much REML, used as if data were independent, understates uncertainty. Although return level estimates look very similar, standard errors from blocked cross-validation are up to seven times greater than those of REML, where return level estimates are largest.

8 Summary

Blocked cross-validation, the special case of neighborhood cross-validation (Wood 2026) in which folds comprise entire, approximately independent clusters, offers a relatively simple approach to

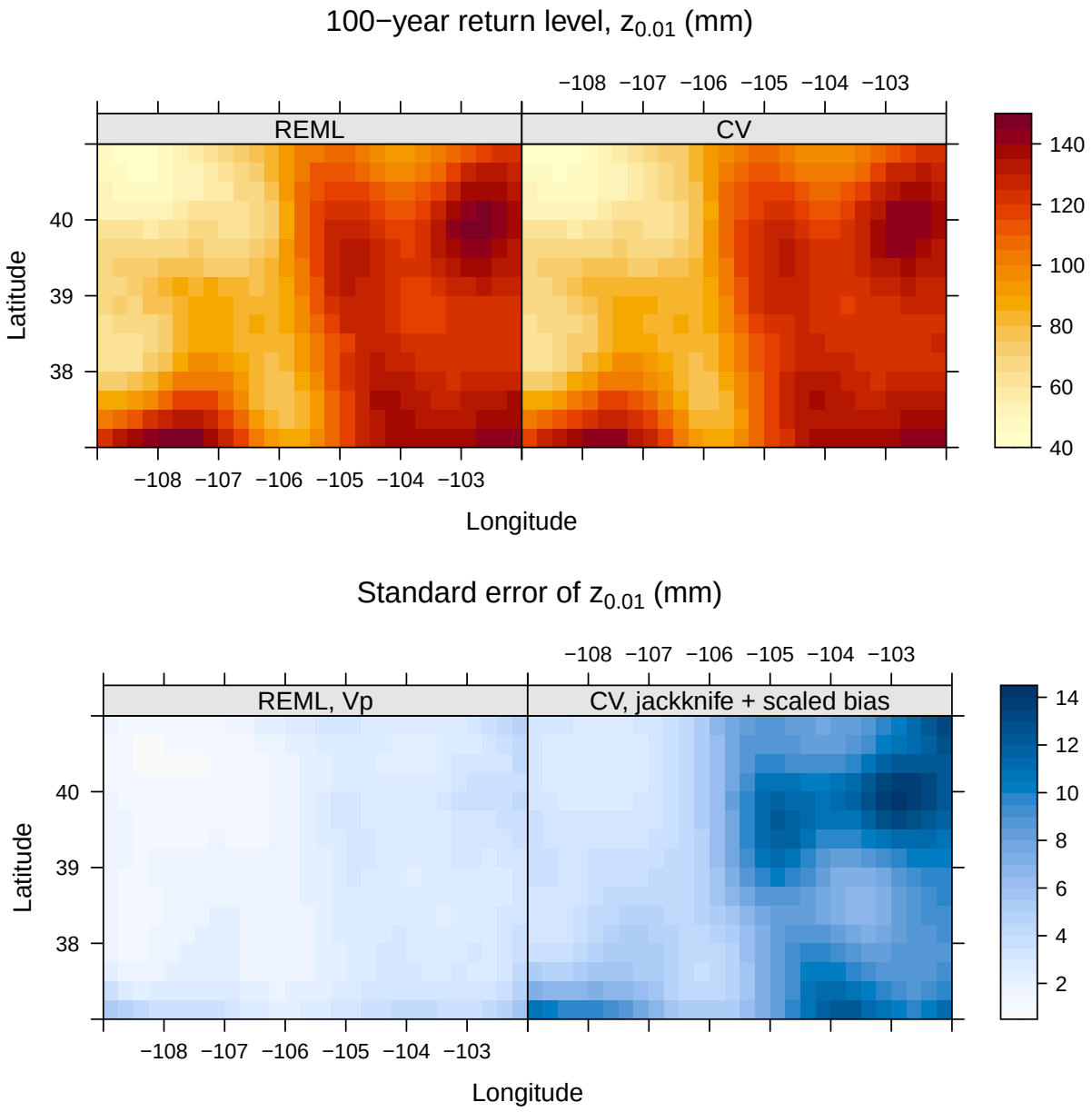


Figure 4: Maps of 100-year return level estimates, $z_{0.01}$ (mm; top), from REML (left) and blocked cross-validation (right), and of their standard errors (mm; bottom), based on $\mathbf{V}_p$ for REML and on the jackknife covariance matrix with the scaled allowance for smoothing bias for blocked cross-validation. Each row shares a color scale.

estimating smoothing parameters of GAM forms for extreme value distribution parameters when data are dependent. Its main novelty is to give validation data a small weight, ϵ, in each training fit, which ensures that they are within the support of each training fit, so that the criterion is finite and smooth, and its gradient is valid. Coupled with cluster-based covariance matrices, neither estimation nor uncertainty quantification requires a model for dependence, only a choice of clusters.

In simulations of annual maxima at dependent locations, blocked cross-validation smoothed more than REML and reduced the RMSE of 100-year return level estimates by 7% to 9%, as did NCV where the support did not constrain training fits. Where it did, omitting validation data, and NCV as implemented in mgcv, gave a criterion that was not finite for 98% of data sets, $\epsilon = 0$ rarely converged, and QNCV rarely converged and had almost twice the RMSE of REML, whereas $\epsilon = 10^{-2}$ converged for every data set and was as accurate as any alternative. Newton-step approximations to training fits were accurate only where the support did not constrain them, and the quadratic approximation, as implemented in evgam, converged reliably but was then less accurate. Intervals based on $\mathbf{V}_p$ had coverage below 0.5, whereas $\mathbf{V}_b$ with the scaled allowance gave 0.92 to 0.95, with the shortfall due to bias at peaks rather than to intervals that were too narrow. Simulations of threshold excesses gave the same findings.

The approach has some drawbacks. Each evaluation of $\ell_{\mathrm{CV}}(\boldsymbol{\lambda})$ requires M training fits, each to all data, although warm starts limit this cost. Estimates depend on the assignment of clusters to folds, and clusters must be sufficiently long, or separated, to be approximately independent, yet sufficiently numerous for both cross-validation and $\hat{\mathbf{J}}$ to be stable. The additional smoothing increases smoothing bias, which $\mathbf{V}_b$ allows for only averaged across covariate values, and so smoothing parameter estimates from cross-validation, and intervals from REML with $\mathbf{V}_b$, may be preferred for point estimates and for intervals, respectively. Intervals here also condition on $\hat{\boldsymbol{\lambda}}$.

Small validation weights apply to any distribution whose log-likelihood tends to $-\infty$ at the boundary of its support, to other forms of cross-validation, such as NCV with small neighborhoods, and to penalized likelihood models outside extreme value analysis. Other proper scoring rules, such as threshold-weighted scores (Gneiting & Raftery 2007), could better target upper tails, and a cluster bootstrap of the entire procedure would better quantify smoothing parameter uncertainty.

Software and code

Functions `cvevgam()` and `evgam_sandwich()` are available in version 1.0.4 of the R package evgam, available from CRAN (Youngman 2026*a*), and its latest version at https://github.com/byoungman/evgam. R scripts to reproduce the simulation studies of Section 6 and the application of Section 7 are available from the author.

Acknowledgements

The author is grateful to Ioannis Papastathopoulos, Phil Sansom, Sophia Groninger, Patrik Bohlinger, Theo Economou, Simon Brown, Emma Simpson, Haya Kamel and Paul Northrop for conversations that motivated this work. No funding was obtained for this work.

Disclosure statement

The author reports there are no competing interests to declare.

Declaration of generative AI use

Generative AI (Claude Sonnet 4.6, 5 and 5.5, Anthropic) was used to assist with drafting and checking some R code, and with formulating some skeleton manuscript text and proof-reading. The author reviewed all such output and takes full responsibility for the content.

Data availability statement

The annual maxima used in Section 7 are from the NA-CORDEX ensemble (Mearns et al. 2017) and are available as dataset `COorder` in the R package evgmrf (Youngman 2026*b*), available from CRAN.

A Computational details of blocked cross-validation

A.1 Computational considerations

Log-likelihood $\ell_{\mathrm{CV}}(\boldsymbol{\lambda})$ is maximized using the BFGS quasi-Newton method that evgam uses for outer iterations. Three considerations improve efficiency. First, each fold's basis functions, penalty matrices and design matrices are created once, so that evaluating $\ell_{\mathrm{CV}}(\boldsymbol{\lambda})$ only requires re-running each fold's inner iterations. Secondly, each fold's inner iterations for a new $\boldsymbol{\lambda}$ start from its coefficient estimates for the last accepted $\boldsymbol{\lambda}$; as $\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}^{(-m)}$ varies smoothly with $\boldsymbol{\lambda}$, few Newton steps are then typically needed. Thirdly, the gradient of $\ell_{\mathrm{CV}}(\boldsymbol{\lambda})$ is only needed for accepted steps, for which fold coefficient estimates are already available, and not for every evaluation of $\ell_{\mathrm{CV}}(\boldsymbol{\lambda})$ during line searches.

As the gradient of $\ell_{\mathrm{CV}}(\boldsymbol{\lambda})$ assumes that each training fit is a stationary point, inner iterations are judged to have converged only if the largest absolute gradient of $\ell_{(-m)}^{\epsilon}(\boldsymbol{\beta})$ is below a tolerance, which by default is smaller than that used for REML. A training fit that has not converged is continued from its estimates, and then started again from the average of other folds' estimates and from evgam's initial values, keeping whichever attains the largest value of $\ell_{(-m)}^{\epsilon}(\boldsymbol{\beta})$. If the BFGS update fails to give a direction of ascent, it is skipped, or the approximation to the inverse Hessian is reset, rather than the outer iterations stopping, and, if they stop with gradient above a tolerance, they are restarted from where they stopped.

Once $\hat{\boldsymbol{\lambda}}$ has been found, the model is refitted to all data. As each training set comprises a proportion $(M-1)/M$ of the data, $\hat{\boldsymbol{\lambda}}$ could be rescaled before this final fit. Asymptotically, the optimal λ_k in $\ell(\boldsymbol{\beta}, \boldsymbol{\lambda})$ grows like $n^{1/(2s+1)}$ for penalties on derivatives of order s, which suggests multiplying by $\{M/(M-1)\}^{1/(2s+1)}$. This is approximately 1.05 for $M = 5$ and $s = 2$, and so is negligible compared with the sampling variability of $\hat{\boldsymbol{\rho}}$; multiplying by $M/(M-1)$, which holds λ_k/n fixed, would over-correct. No rescaling is used by default.

For training fits approximated by Newton steps, let $\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}$ maximize $\ell(\boldsymbol{\beta}, \boldsymbol{\lambda})$. A first step is $\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}} - \mathbf{H}_{(-m)}^{-1}\mathbf{g}_{(-m)}$, where $\mathbf{g}_{(-m)}$ is the gradient of the penalized training log-likelihood at $\hat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}$, and $\mathbf{H}_{(-m)}$ is its negative Hessian there, and further steps are taken from the result. Each step is halved until all data are within the support and the training objective does not decrease, beyond rounding error, and steps stop once the gradient is below the tolerance used for inner iterations. The gradient of $\ell_{\mathrm{CV}}(\boldsymbol{\lambda})$ is then found by central differences.

A.2 Distributions with bounded support

The GPD has a finite upper endpoint if $\xi < 0$, and the GEV distribution has a finite upper endpoint if $\xi < 0$ and a finite lower endpoint if $\xi > 0$. If a held-out observation lies outside the support of a training fit, then that observation has zero likelihood and $\ell_{\mathrm{CV}}(\boldsymbol{\lambda}) = -\infty$. Starting from smoother fits, such as from REML estimates of $\boldsymbol{\rho}$ plus a constant, reduces the chance of this, but, in the simulations of Section 6 with $\xi = -0.4$, did not avoid it.

Small validation weights avoid it as follows. Consider the GPD, for which $\log f_{\mathrm{GPD}}(y; \psi, \xi) = -\log\psi - (1 + 1/\xi)\log z$, where $z = 1 + \xi y/\psi$, and suppose that $-1 < \xi < 0$, so that $1 + 1/\xi < 0$ and $\log f_{\mathrm{GPD}} \to -\infty$ as $z \to 0$, i.e., as y approaches the upper endpoint. Suppose that, without validation data, the training fit would place validation datum y_i outside the support, so that, in $\ell^{\epsilon}_{(-m)}(\boldsymbol{\beta})$, the gradient of the training log-likelihood pushes z_i towards zero, with magnitude, g say, that does not depend on ϵ. The validation datum's contribution to the gradient of $\ell^{\epsilon}_{(-m)}(\boldsymbol{\beta})$ pushes z_i away from zero, with magnitude approximately $\epsilon\,|1 + 1/\xi|/z_i$. At the training fit these balance, which gives

$$z_i \approx \epsilon\,|1 + 1/\xi|/g,$$

i.e., z_i is approximately proportional to ϵ. The validation datum's contribution to $\ell_{\mathrm{CV}}(\boldsymbol{\lambda})$ is then approximately $-\log\psi - (1 + 1/\xi)\log z_i \approx |1 + 1/\xi|\log\epsilon + \text{constant}$, and so is bounded below, and the curvature of $\epsilon \log f_{\mathrm{GPD}}$ in z_i is $\epsilon\,|1 + 1/\xi|/z_i^2$, which grows like $1/\epsilon$. The GEV distribution's upper endpoint behaves in the same way. Its lower endpoint for $\xi > 0$ behaves differently, as then $\log f_{\mathrm{GEV}}$ includes $-z^{-1/\xi}$, which tends to $-\infty$ faster, but again ensures that validation data are within the support.

Small validation weights therefore cap, in effect, the loss of each validation datum that a training fit would exclude, while leaving the loss of other validation data unchanged. Wood (2026) instead replaces the loss of every validation datum by its quadratic approximation about the fit to all data, which is finite for all $\boldsymbol{\beta}$. A proper scoring rule that remains finite outside the support, such as the continuous ranked probability score (Gneiting & Raftery 2007), could also be used, although its gradient would no longer take a simple form.

B Further details of the implementation in evgam

This section gives further details of functions `cvevgam()` and `evgam_sandwich()` of the R package evgam, whose use is illustrated in Section 5. Arguments of `cvevgam()` that control cross-validation are as follows.

- `group`: the variable that defines clusters, e.g., `group = "year"`, so that folds comprise entire clusters.
- `folds`: the number of folds, e.g., `folds = 5` for five-fold cross-validation; `folds` equal to the number of clusters gives leave-one-cluster-out cross-validation.
- `ratio`: a ratio of training to validation data, which gives repeated random splits of clusters instead of folds.
- `holdout` and `holdout.weight`: `holdout = "weights"` gives validation data weight `holdout.weight`, i.e., ϵ, and `holdout = "subset"` omits them. For families whose likelihood functions take weights, currently `"gev2"` and `"gpd2"`, small validation weights are the default, with `holdout.weight = 0.01`, i.e., $\epsilon = 10^{-2}$.
- `approx` and `nsteps`: `approx = "onestep"` approximates training fits by `nsteps` Newton steps from the fit to all data, and `approx = "quadratic"` uses the quadratic approximation

of Wood (2026); the default, `approx = "exact"`, gives exact training fits.
- `rho0` and `rho0.offset`: starting values for $\boldsymbol{\rho}$. By default, these are the REML estimates from the fit to all data plus `rho0.offset`, which is zero by default.

The returned object is of class `evgam`, and so `predict()`, `simulate()`, `plot()` and `summary()` can be used as usual, with additional element `cv` giving $\hat{\boldsymbol{\rho}}$, the maximized criterion, its gradient, whether its optimization converged, the convergence of training fits and the folds used. Function `evgam_sandwich()` replaces a fit's covariance matrices, through the following arguments.

- `type`: `type = "full"` gives $\mathbf{V}_s$, and `type = "jackknife"` gives $\mathbf{V}_j$.
- `bias`: `bias = TRUE` or `bias = "scaled"` adds the unscaled or scaled allowance of $\mathbf{V}_b$.

It can also be applied to `evgam` objects for which smoothing parameters were estimated by REML.

For models fitted by the R package mgcv, NCV is available through `gam()`'s argument `method = "NCV"`, with neighborhoods given by its argument `nei` (Wood 2026), and `method = "QNCV"` replaces each predicted datum's log-likelihood by its quadratic approximation about the fit to all data. For the covariance matrix of Wood (2026) to be computed, each datum must be given its own neighborhood, e.g., all data in its year, rather than each year forming a single neighborhood.

C Versions of blocked cross-validation for annual maxima

Table 3 compares versions of blocked cross-validation in the simulation study of annual maxima with $\xi = 0.15$ of Section 6.

Table 3: Simulation study of annual maxima, $\xi = 0.15$: versions of blocked cross-validation, compared with exact leave-one-year-out cross-validation through the paired RMSE ratio of $z_{0.01}(x)$ estimates, with 95% Monte Carlo intervals, and the median absolute difference in $\hat{\rho}$ for g_μ and g_σ; and mean time per fit; exact training fits use $\epsilon = 10^{-2}$ unless stated otherwise.

Folds	Training fits	RMSE ratio	$\lvert\Delta\hat{\rho}_\mu\rvert$	$\lvert\Delta\hat{\rho}_\sigma\rvert$	Time (s)
Leave-one-year-out	exact	1	0	0	79
Leave-one-year-out	two steps	1.000 (1.000, 1.001)	0.019	0.024	41
Leave-one-year-out	one step	0.997 (0.987, 1.007)	0.111	0.134	17
Leave-one-year-out	quadratic	1.001 (0.998, 1.005)	0.103	0.119	13
Five-fold	exact	1.007 (0.997, 1.017)	0.221	0.305	11
Five-fold	one step	1.012 (0.997, 1.028)	0.239	0.429	10
Five-fold	exact, validation data omitted	1.007 (0.997, 1.017)	0.207	0.297	9
Five-fold	exact, $\epsilon = 0$	1.007 (0.997, 1.017)	0.207	0.297	11
Random splits	exact	1.014 (0.991, 1.038)	0.373	0.607	20

D Simulations of threshold excesses

These simulations follow Section 6, except for steps 3 and 4.

3. For each of 50 independent years, simulate two independent events, each of which gives an excess at every location, from the Gaussian copula, with margins $GPD(\psi(x), \xi)$.
4. Fit $Y(x) \sim GPD(\psi(x), \xi)$, with $\log \psi(x) = g_\psi(x)$ and ξ constant, where g_ψ is a cubic regression spline of rank 30.

Here $z_p(x)$ is the excess exceeded on average once every $1/p$ years, i.e., with exceedance probability $p/2$ for each event. Two shapes are used: $\xi = -0.15$, for which the upper endpoint is far beyond the

data, and $\xi = -0.4$, for which it is close to the largest data; ψ_0 is chosen so that $z_{0.01}(x)$ is about 33.5 on average. NCV is not considered, as mgcv has no GPD family.

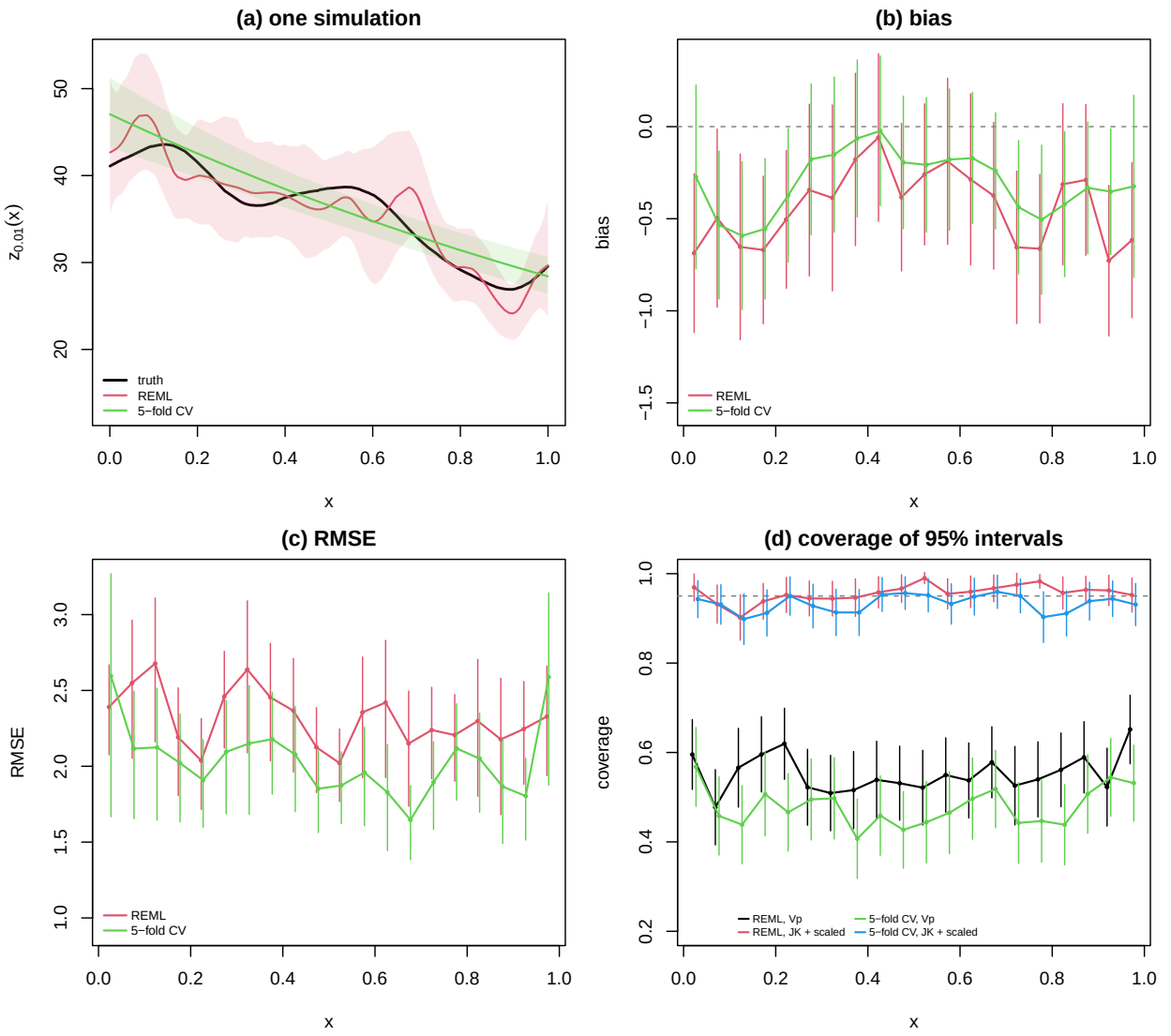


Figure 5: Simulation study of threshold excesses, $\xi = -0.15$: (a) true $z_{0.01}(x)$ for a single simulation with its estimates and 95% confidence intervals, based on $\mathbf{V}_b$ with the scaled allowance, from REML and five-fold cross-validation with $\epsilon = 10^{-2}$; (b) bias and (c) RMSE of estimates of $z_{0.01}(x)$ against x, in bins of width 0.05; and (d) coverage probability of 95% confidence intervals against x, where JK denotes $\mathbf{V}_b$ with the scaled allowance. Vertical bars give approximate 95% Monte Carlo intervals.

For $\xi = -0.15$, the support rarely constrained training fits, and every treatment of validation data gave the same estimates; omitting them gave a criterion that was not finite for only 2 of 100 simulations. Five-fold cross-validation reduced the RMSE of REML by about 14%, and coverage with the scaled allowance was 0.95 for REML and 0.93 for cross-validation (Figure 5).

For $\xi = -0.4$, the findings were the same as for annual maxima (Table 4 and Figure 6). Omitting validation data gave a criterion that was not finite for 91% of simulations. With $\epsilon = 0$, the optimization converged for only 24% of simulations, and estimates were about 6% less accurate

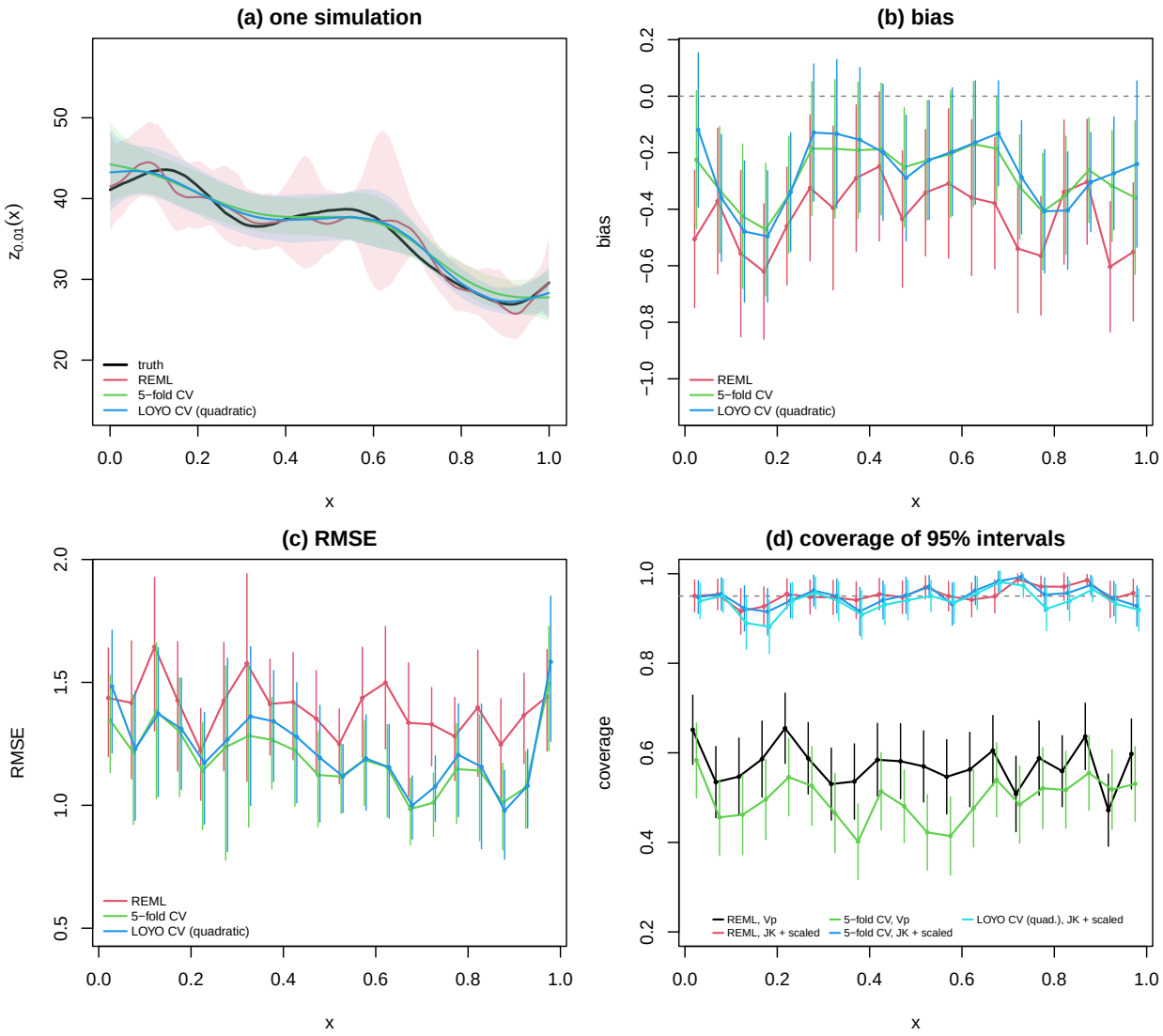


Figure 6: Simulation study of threshold excesses, $\xi = -0.4$: (a) true $z_{0.01}(x)$ for a single simulation with its estimates and 95% confidence intervals, based on $\mathbf{V}_b$ with the scaled allowance, from REML and five-fold cross-validation with $\epsilon = 10^{-2}$ and leave-one-year-out cross-validation with the quadratic approximation; (b) bias and (c) RMSE of estimates of $z_{0.01}(x)$ against x, in bins of width 0.05; and (d) coverage probability of 95% confidence intervals against x, where JK denotes $\mathbf{V}_b$ with the scaled allowance. Vertical bars give approximate 95% Monte Carlo intervals.

Table 4: Simulation study of threshold excesses with $\xi = -0.4$: as Table 2. For REML, coverage is 0.95.

Approach	Fail	Converged	RMSE ratio	Coverage
Validation data omitted	91	–	–	–
$\epsilon = 0$	0	24	0.881 (0.846, 0.917)	0.93
$\epsilon = 10^{-4}$	0	99	0.842 (0.813, 0.873)	0.95
$\epsilon = 10^{-3}$	0	100	0.835 (0.808, 0.864)	0.95
$\epsilon = 10^{-2}$	0	100	0.834 (0.808, 0.860)	0.95
Leave-one-year-out, one step	0	62	0.846 (0.822, 0.871)	0.95
Leave-one-year-out, two steps	0	52	0.858 (0.833, 0.885)	0.95
Leave-one-year-out, quadratic	0	99	0.843 (0.811, 0.876)	0.94

than with $\epsilon = 10^{-2}$, which always converged, reduced the RMSE of REML by about 17%, and took 14 seconds per fit, against 2 seconds for REML. Newton steps converged for only 52% to 62% of simulations, whereas, unlike for annual maxima, the quadratic approximation was as accurate as exact training fits. Coverage with the scaled allowance was 0.93 to 0.95 for all approaches.

The behavior of ϵ is illustrated by refitting, with each value of ϵ, the one simulation for which $\epsilon = 10^{-4}$ did not converge. In three of its five folds, training fits put validation data close to the boundary of the support, and the smallest value of $1 + \xi y/\psi(x)$ among validation data was almost exactly proportional to ϵ, as anticipated in Section A.2, e.g., 7.8×10^{-7}, 7.8×10^{-6} and 7.9×10^{-5} for $\epsilon = 10^{-4}$, 10^{-3} and 10^{-2} in one fold. With $\epsilon = 0$, these validation data were on the boundary, where the largest absolute gradient of $\ell^{\epsilon}_{(-m)}(\boldsymbol{\beta})$ was between 56 and 407, rather than zero. The gradient of $\ell_{\mathrm{CV}}(\boldsymbol{\lambda})$ from implicit differentiation was then of order 10^{10}, and central differences with steps 0.1, 0.01 and 0.001 were about 2×10^2, 2×10^3 and 2.5×10^4 in absolute value, as $\ell_{\mathrm{CV}}(\boldsymbol{\lambda})$ changed sharply with $\boldsymbol{\rho}$. With $\epsilon = 10^{-2}$, central differences were consistent with the gradient from implicit differentiation, which was close to zero, for smaller steps.

References

Arlot, S. & Celisse, A. (2010), 'A survey of cross-validation procedures for model selection', *Statistics Surveys* **4**, 40–79.

Bell, R. M. & McCaffrey, D. F. (2002), 'Bias reduction in standard errors for linear regression with multi-stage samples', *Survey Methodology* **28**(2), 169–181.

Chandler, R. E. & Bate, S. (2007), 'Inference for clustered data using the independence loglikelihood', *Biometrika* **94**(1), 167–183.

Chavez-Demoulin, V. & Davison, A. C. (2005), 'Generalized additive modelling of sample extremes', *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **54**(1), 207–222.

Gneiting, T. & Raftery, A. E. (2007), 'Strictly proper scoring rules, prediction, and estimation', *Journal of the American Statistical Association* **102**(477), 359–378.

Godambe, V. P. (1960), 'An optimum property of regular maximum likelihood estimation', *The Annals of Mathematical Statistics* **31**(4), 1208–1211.

Lindsay, B. G. (1988), 'Composite likelihood methods', *Contemporary Mathematics* **80**(1), 221–239.

Marra, G. & Wood, S. N. (2012), 'Coverage properties of confidence intervals for generalized additive model components', *Scandinavian Journal of Statistics* **39**(1), 53–74.

Mearns, L. O. et al. (2017), 'The NA-CORDEX dataset, version 1.0', NCAR Climate Data Gateway, Boulder, CO.

Nychka, D. (1988), 'Bayesian confidence intervals for smoothing splines', *Journal of the American Statistical Association* **83**(404), 1134–1143.

Opsomer, J., Wang, Y. & Yang, Y. (2001), 'Nonparametric regression with correlated errors', *Statistical Science* **16**(2), 134–153.

R Core Team (2026), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
URL: *https://www.R-project.org/*

Roberts, D. R., Bahn, V., Ciuti, S., Boyce, M. S., Elith, J., Guillera-Arroita, G., Hauenstein, S., Lahoz-Monfort, J. J., Schröder, B., Thuiller, W., Warton, D. I., Wintle, B. A., Hartig, F. & Dormann, C. F. (2017), 'Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure', *Ecography* **40**(8), 913–929.

Tierney, L. & Kadane, J. B. (1986), 'Accurate approximations for posterior moments and marginal densities', *Journal of the American Statistical Association* **81**(393), 82–86.

Varin, C., Reid, N. & Firth, D. (2011), 'An overview of composite likelihood methods', *Statistica Sinica* **21**(1), 5–42.

Wahba, G. (1990), *Spline Models for Observational Data*, SIAM, Philadelphia.

White, H. (1982), 'Maximum likelihood estimation of misspecified models', *Econometrica* **50**(1), 1–25.

Wood, S. N. (2011), 'Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **73**(1), 3–36.

Wood, S. N. (2026), 'On neighbourhood cross validation', *Statistical Science* **41**(4), 827–846.

Wood, S. N., Pya, N. & Säfken, B. (2016), 'Smoothing parameter and model selection for general smooth models', *Journal of the American Statistical Association* **111**(516), 1548–1563.

Youngman, B. D. (2019), 'Generalized additive models for exceedances of high thresholds with an application to return level estimation for U.S. wind gusts', *Journal of the American Statistical Association* **114**(528), 1865–1879.

Youngman, B. D. (2022), 'evgam: An R package for generalized additive extreme value models', *Journal of Statistical Software* **103**(3).

Youngman, B. D. (2026*a*), *evgam: Generalised Additive Extreme Value Models*. R package version 1.0.3.
URL: *https://CRAN.R-project.org/package=evgam*

Youngman, B. D. (2026*b*), *evgmrf: Extreme Value Models with Gaussian Markov Random Fields*. R package version 0.0.1.