Variational inference for Bayesian bridge regression

Carlos Tadeu Pagani Zanini1 · Helio S. Migon1 · Ronaldo Dias2

Abstract
The bridge approach for regularization of coefficients in regression models uses $\ell_\alpha$ norm, with $\alpha \in (0, +\infty)$, to define a penalization on large values of the regression coefficients. Particular cases include the Lasso and Ridge penalizations. In Bayesian models, the penalization is enforced by a prior distribution on the coefficients. Although MCMC approaches are available for Bayesian bridge regression, they can be very slow for large datasets, specially in high dimensions. This paper develops an implementation of Automatic Differentiation Variational Inference for Bayesian inference on semi-parametric regression models with bridge penalization. The non-parametric effects of covariates are modeled by B-splines. The proposed inference procedure allows the use of small batches of data at each iteration (due to stochastic gradient based updates), therefore drastically reducing computational time in comparison with MCMC. Full Bayesian inference is preserved so joint uncertainty estimates for all model parameters are available. A simulation study shows the main properties of the proposed method and an application to a large real dataset is presented.

Keywords
Variational Inference · Bridge penalization · Bayesian inference · Splines

1 Introduction and related works
It is not uncommon for an experimenter to be interested in understanding how covariates might explain a response variable. For this, one can assume a general non-parametric regression model:

$$y_i = g(x_{i1}, \ldots, x_{ij}) + \epsilon_i,$$

where $\epsilon_i \sim N(0, \phi^{-1})$ for $i = 1, \ldots, n$.

A usual approach to estimate the surface $g$ is to consider the well-known (General) Additive Model, which briefly means that $g$ can be decomposed as

$$g(x_1, \ldots, x_J) = g_0 + \sum_{j=1}^{J} g_j(x_j),$$

with covariates $x_j = (x_{1j}, \ldots, x_{nj})^\top$, for $j = 1, \ldots, J$ and $i = 1, \ldots, n$. Each univariate function $g_j$ can be written as a linear combination of basis functions $B_{j1}, \ldots, B_{kJ_j}$, such as B-splines, wavelets, Fourier basis, etc. That is, $g_j(x_j) = \sum_{k=1}^{K_j} B_{jk}(x_j)\beta_{jk}$. The term $g_0 \in \mathbb{R}$ represents either an intercept, or parametric covariate effects (not modelled by basis functions).

Throughout this work, the basis functions chosen to represent the univariate function $g_j$ are the well-known B-splines. Thus, the surface regression model is:

$$y_i = g_0 + \sum_{j=1}^{J} \sum_{k=1}^{K_j} \beta_{jk} B_{jk}(x_{ij}) + \epsilon_i.$$  \hspace{1cm} (1)

In regression models that include non-parametric effects, it is crucial to assign some form of penalization in order to calibrate the level of complexity of the relationships between the covariates and the response to avoid overfitting. Generalized additive models (GAM) from Hastie et al. (1986) use one

References

Carlos Tadeu Pagani Zanini
carlozanini@dme.ufrj.br

Helio S. Migon
migon@im.ufrj.br

Ronaldo Dias
dias@unicamp.br

1 Departamento de Métodos Estatísticos, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brazil

2 Instituto de Matemática, Estatística e Ciência da Computação, Universidade Estadual de Campinas, Campinas, SP, Brazil
spline basis for each covariate to model the non-parametric effects. For common choices of spline basis functions such as B-spline, see Eilers and Marx (1996). The level of smoothness can be controlled by a penalization acting on the splines coefficients, see Currie and Durban (2002). Some Bayesian approaches in this context include Lang and Brezger (2004) and Hastie and Tibshirani (2000).

In particular, after writing (1) in matrix form, the semi-parametric regression model can be written as:

\[ y = X_0 \beta_0 + \sum_{j=1}^{J} X_j \beta_j + \epsilon, \]

where \( \epsilon \sim N(0, \phi^{-1} I_n) \), with \( \sum_{k=1}^{K_j} B_{jk}(x_j) \beta_{jk} = X_j \beta_j \), where \( X_j \) is the \( n \times K_j \) matrix \( X_j = [B_{j1}(x_j), \ldots, B_{jK_j}(x_j)] \) and \( \beta_j = (\beta_{j1}, \ldots, \beta_{jK_j})^{\top} \). Note that the \( n \times K_0 \) matrix \( X_0 \) comprises possible parametric covariates (with coefficients \( \beta_0 \in \mathbb{R}^{K_0} \)) and hence can also be viewed as Partial Splines (Silverman and Green 1994). In particular, if \( X_0 = 1 \) then \( X_0 \beta_0 \) plays the role of an intercept as \( g_0 \) from Eq. (1) according to an Additive Model formulation.

In general, penalizing over-complex models often reduces the risk of overfitting and produces better generalization to hold out data. For this, under the frequentist point of view, one have to solve the following optimization problem:

\[
\arg \min_{\beta} \left\{ \|y - g_0 - \sum_{j=1}^{J} K_j \beta_{jk} B_{jk}(x_j)\|^2 + \right. \\
+ \sum_{j=1}^{J} \lambda_j \mathcal{P}(\beta_j, \alpha_j) \right\},
\]

where \( \lambda_j \) is the smoothing parameter and \( \mathcal{P}(\beta_j, \alpha_j) \) is the roughness penalty term which depends (possibly) on a parameter \( \alpha_j \).

Different types of penalties \( \mathcal{P}(\beta_j, \alpha_j) \) can be applied. For instance, \( \mathcal{P}(\beta_j, \alpha_j = 1) = \|\beta_j\|_1 \) represents a penalization in terms of the \( \ell_1 \)-norm, \( \mathcal{P}(\beta_j, \alpha_j = 2) = \|\beta_j\|_2^2 \) represents the \( \ell_2 \)-norm squared, the smoothing splines penalty is represented as \( \mathcal{P}(\beta_j, \alpha_j = 2) = \int (\sum_{j} \beta_j \beta_j^\top) \), and P-spline as \( \mathcal{P}(\beta_j, \alpha_j = 2) = \sum_{j=1}^{K} (\Delta^k \beta_j)^2 \), where \( \Delta^k \) is the \( k \)-th order difference operator.

Penalties defined in terms of \( \ell_0 \) norm in the context of Bayesian regression models include the Ridge (Hoerl and Kennard 1970a, b), the Lasso (Park and Casella 2008), elastic net (Li and Lin 2010) (convex combination of both \( \ell_1 \) and \( \ell_2 \) norms) and bridge (Polson et al. 2014; Mallick and Yi 2018) (\( \alpha > 0 \)), first presented by Frank and Friedman (1993). The different choices of \( \ell_0 \) norm for the penalization imply different forms of shrinking the regression coefficients towards zero. While Ridge regression does not zero out coefficients, the Lasso penalization is capable of producing sparse solutions to the corresponding maximum a posteriori objective, therefore working as a variable selection procedure (Tibshirani 1996, 1997).

In the context of bridge penalized linear regression with Gaussian response, a possible bridge model for semi-parametric regression is a regularization problem defined as,

\[
\arg \min_{\beta, \alpha} \frac{1}{2} \| y - X_0 \beta_0 - \sum_{j=1}^{J} X_j \beta_j \|^2 + \sum_{j=1}^{J} \lambda_j \| \beta_j \|_1^{\alpha_j}/\alpha_j \tag{2}
\]

where \( \lambda_j > 0 \) is the penalty term that controls the strength of shrinkage over \( \beta_j \) and \( \alpha_j > 0 \) is the concavity parameter of the penalty function. From a Bayesian perspective, Hastie et al. (2015), Casella et al. (2010) and Leng et al. (2014), highlight that one can interpret the term \( \| \beta_j \|_1^{\alpha_j}/\alpha_j \) proportional to the negative log-prior density of \( \beta_j \) with the contours illustrated in Fig. 1 representing the contours of the prior distribution. The case \( \alpha_j < 1 \) implies a non-convex prior that concentrates more mass along the coordinates’ axis, producing solutions with fewer nonzero coefficients and less shrinkage.

Polson et al. (2014) and Mallick and Yi (2018) develop full Bayesian inference in Bayesian bridge regression through fixed dimension MCMC based on variable augmentation schemes. Denison et al. (1998) and Dias and Gamerman (2002) describe reversible jumps MCMC algorithms for selecting the number of knots in B-splines and P-splines. These MCMC approaches, however, are computationally very expensive when applied to large datasets.

Other forms of penalization are also relevant. The elastic net provides sparsity of representation similarly to the Lasso while also encouraging a grouping effect where strongly correlated predictors tend to be in or out of the model together (Zou and Hastie 2005). In Bayesian regression models, the horseshoe prior implies a form of variable selection that acts strongly on coefficients of small magnitude while having little influence on coefficients of large magnitude (Carvalho et al. 2009).

This work proposes an alternative to MCMC for approximate inference on Bayesian bridge semi-parametric regression with B-splines based on the Automatic Differentiation Variational Inference (ADVI) by Kucukelbir et al. (2015) and Kucukelbir et al. (2017). The proposed inference algorithm jointly estimates all parameters in the model (including \( \alpha_j \) and \( \lambda_j \)) and drastically increases computational speed for large datasets in comparison with MCMC implementations since stochastic gradient updates involved in the underlying optimization process small batches of data at each iteration, rather than the entire dataset as required by MCMC schemes. A key point in variational inference concerns the choice of the variational family of distributions,
which needs to be tractable while at the same time flexible enough to approximate the posterior distribution well. Armagan (2009) describe a mean-field variational inference method for Bayesian bridge regression model with approximate inference for the bridge parameter $\alpha$. Alves et al. (2021) also proposes a variational approach based on mean-field assumption for Bayesian inference in regression models with splines, however it is restricted to Lasso penalization. In contrast, the ADVI does not require the oversimplifying mean-field assumption for the variational family, as required by coordinate ascending variational inference (Blei et al. 2017), producing very close approximations to the full posterior distribution as demonstrated in the simulation study. Previous gradient based variational procedures, such as Ranganath et al. (2014) or Kingma and Welling (2013) could be used, but they imply more restricted forms of dependence in the variational family and produce noisier estimates of the gradients when compared to ADVI. Although other approaches, such as the semi-implicit variational inference by Yin and Zhou (2018), define more flexible variational families, we found the simpler ADVI approach to produce a good balance between high computational speed and accurate posterior approximations under the semi-parametric Bayesian bridge model.

To summarize, the main contribution of the paper is the development of a full Bayesian inference procedure based on variational inference for semi-parametric regression with bridge penalization. As strengths of the proposed methodology, we can list the following:

1. The proposed inference method is flexible enough to capture the dependence structure in the target posterior distribution since the variational family does not require the mean-field assumption. More specifically, for the mean-field approach, the joint variational distribution of the parameters is the product of its marginals. On the other hand, ADVI admits dependence structure given by a transformation of a multivariate normal distribution with full covariance matrix.

2. It enables full Bayesian inference on semi-parametric regression for large datasets at drastically lower computing times when compared with more traditional MCMC implementations due to batch processing of data by the stochastic gradient updates specially for large data.

The remainder of the paper is structured as follows. The Bayesian bridge semi-parametric model is described in Sect. 2. Section 3 covers the MCMC scheme by Mallick and Yi (2018), used for comparison with the proposed ADVI for Bayesian bridge semi-parametric regression defined in Sect. 4. Section 5 concerns a simulation study with focus on large data. Section 6 contains an application to real data on energy charges in Brazil and, finally, Sect. 7 presents the conclusions and future works.

2 The Bayesian bridge model for semi-parametric regression

A Bayesian bridge model for multiple covariates can be written in the form of a hierarchical model as follows:

$$(y \mid \beta_0, \beta_1, \ldots, \beta_J, \phi) 
\sim N \left( X_0 \beta_0 + \sum_{j=1}^{J} X_j \beta_j, \phi^{-1} I_n \right),$$

$$(\beta_{jk} \mid \lambda_j, \phi, \alpha_j) \sim GG \left( 0, \lambda_j^{-1}, \phi^{-1}, \alpha_j \right),$$

$$j = 1, \ldots, J, \ k = 1, \ldots, K_j. \quad (3)$$

where $y \in \mathbb{R}^n$, $N(\mu, \Sigma)$ denotes the multivariate Gaussian distribution with mean $\mu$ and covariance matrix $\Sigma$, $GG(\mu, \sigma, \alpha)$ denotes the generalized Gaussian distribution (Subbotin 1923) with mean $\mu$, scale $\sigma$ and shape $\alpha$. The generalized Gaussian density evaluated at $x \in \mathbb{R}$ is denoted by

$$GG(x \mid \mu, \sigma, \alpha) = \frac{\alpha}{2\sigma \Gamma(\alpha^{-1})} \exp \left\{ -\left( \frac{|x - \mu|}{\sigma} \right)^\alpha \right\}.$$ 

Equation (3) therefore implies the prior probability density function

$$p(\beta_{jk} \mid \lambda_j, \phi, \alpha_j)$$
so that maximizing the posterior density \( p(\mathbf{\beta} | \mathbf{y}, \lambda, \phi, \alpha, \beta_0) \) over \( \mathbf{\beta} \) and \( \beta_0 \) for fixed \( (\lambda, \phi, \alpha) \) is equivalent to solving (2), where \( \mathbf{\beta} = (\beta_1, \ldots, \beta_J) \), \( \lambda = (\lambda_1, \ldots, \lambda_J) \), \( \alpha = (\alpha_1, \ldots, \alpha_J) \). The particular form of dependency of \( \beta_{jk} \) on \( \lambda_j \) and \( \phi \) expressed by Eq. (4) avoids undesired multimodality in the joint posterior distribution of \( \beta_{jk} \) and \( \phi \) (marginally on the other model parameters) as pointed out by Park and Casella (2008).

The proposed semi-parametric model splits the regressors into matrices \( \mathbf{X}_j \), \( j = 1, \ldots, J \) of covariates with their own penalized effects and a matrix \( \mathbf{X}_0 \) of covariates with unpenalized effects. In the proposed formulation, \( \mathbf{X}_j \) represents a spline basis function that implies non-parametric effects to the corresponding covariates. B-splines were chosen as the set of spline basis functions to build \( \mathbf{X}_j \) with the bridge penalization acting on the \( B \)-spline coefficients. Naturally, other types of spline basis functions could also be used.

Within the Bayesian framework, it is possible to provide joint (approximate) posterior estimates for all parameters, including \( \alpha \) and \( \lambda \) by specifying a prior distribution on them, as will be shown further in Sects. 3 and 4. Unlike the usual Lasso method, the proposed Bayesian model does not zero the coefficients but shrink them instead. If the posterior of \( \alpha_j \) concentrates mass around 1 or 2, then the results tend to be similar to those produced by Lasso and Ridge, respectively. Notice, however, that \( \alpha_j \) will never be exactly 1 or 2 due to the continuous nature of its posterior distribution.

### 3 MCMC on Bayesian bridge regression

Both Polson et al. (2014) and Mallick and Yi (2018) propose MCMC schemes for Bayesian bridge regression for fixed \( \alpha_j \) based on different data augmentation schemes. The algorithm by Mallick and Yi (2018) was chosen for comparison with the proposed ADVI for Bayesian bridge, mostly because of the simplicity of its gamma-uniform variable augmentation in comparison with the one proposed by Polson et al. (2014). Accordingly, one can recover (3) by specifying

\[
(u_{jk} | \alpha_j, \lambda_j) \sim \text{Ga} \left( \frac{1}{\alpha_j} + 1, \lambda_j \right), \quad k = 1, \ldots, K_j,
\]

(\( u_{jk} \) is not marginalized). It was shown that best approximates the posterior of \( \beta_{jk} \) on \( \mathbf{\beta}_j \) fixed. To include \( \alpha_j \) in the sampling scheme, one can carry out a Metropolis–Hastings step with a transformed random walk proposal as described in Sect. A.2.1 (when the auxiliary variables \( u_{jk} \) are marginalized) or in Sect. A.2.2 (when \( u_{jk} \) is not marginalized). It was found that the marginalized random walk proposal produce better mixing Markov chains without much effort to tune the proposal variance. Naturally, other Metropolis–Hastings proposals could be used.

### 4 Variational inference

The variational approach searches among a predefined variational family \( \mathcal{Q} = \{ q_{\psi}(\theta) : \psi \in \mathcal{V} \} \) (where its members are densities indexed by the variational parameter \( \psi \)) for the density \( q_{\psi^*}(\theta) \) that best approximates the posterior \( p(\theta | \mathbf{y}) \) in terms of Kullback–Leibler divergence. For the proposed Bayesian semi-parametric bridge regression, \( \theta = (\beta_0, \mathbf{\beta}, \phi, \lambda, \alpha)^T \in \mathbb{R}^{k_0} \times \mathbb{R}^{k_1 + \ldots + k_J} \times \mathbb{R}^+ \times (0, +\infty)^J \times \prod_{j=1}^{J} (0, C_j) \).

Therefore, the variational objective is to obtain

\[
\arg\min_{\psi \in \mathcal{V}} KL \left( q_{\psi}(\theta) \parallel p(\theta | \mathbf{y}) \right).
\]
which is equivalent to maximizing the evidence lower bound (ELBO), i.e., \( \arg \max_{\theta} ELBO(y, \psi) \), where

\[
ELBO(y, \psi) := E_{q(\theta)} [\log p(y, \theta) - \log q(\theta)] = \int [\log p(y, \theta) - \log q(\theta)] q(\theta) d\theta.
\]

One way to maximize the ELBO is to calculate its gradient \( \nabla_{\psi} ELBO(y, \psi) \) and use it in a stochastic gradient ascend based algorithm, such as Adam (Kingma and Ba (2014)), Adagrad (Duchi et al. 2011), AdaDelta (Zeiler 2012) and others. The issue with such gradient ascending methods is evaluating the expectation in the ELBO, which is often intractable. In this case, a common solution involves rewriting the ELBO in a way that the gradient operator can switch order with the expectation so the resulting \( \nabla_{\psi} ELBO(y, \psi) \) is an expected value with respect to the variational distribution. The resulting expectation can then be estimated via Monte Carlo methods.

This work implements ADVI (Sect. 4.3) to the semi-parametric Bayesian bridge regression described in Sect. 2. The variational approaches by Ranganath et al. (2014) and by Kingma and Welling (2013) are briefly presented for comparison and contextualization. A reader interested in variational inference for semi-parametric regression models may see Luts and Wand (2015), Menictas and Wand (2015), Ong et al. (2017) and Wand (2017) to mention a few.

### 4.1 Score method

Ranganath et al. (2014) introduced the black box variational inference (BBVI) in which the score method (also known as “log trick”) allows the gradient of the ELBO to be written as an expectation with respect to the variational distribution. The resulting expectation can be estimated by Monte Carlo as follows

\[
\nabla_{\psi} ELBO(y, \psi) = \int \nabla_{\psi} \{[\log p(y \mid \theta) + \log p(\theta)] + \log q(\theta)\} d\theta
\approx E_{q(\theta)}[\nabla_{\theta} \{[\log p(y \mid \theta) + \log p(\theta) - \log q(\theta)]
\times \nabla_{\theta} \log q(\theta)]
\approx \frac{1}{M} \sum_{m=1}^{M} \left[ \log p(y \mid \theta^{(m)}) + \log p(\theta^{(m)}) + - \log q(\theta^{(m)}) \right] \nabla_{\psi} \log q(\theta^{(m)}),
\]

(6)

where \( \theta^{(m)} \overset{iid}{\sim} q(\theta), m = 1 \ldots M \).

Evaluating (6) requires (i) the prior, likelihood and variational densities to be analytically available and (ii) the ability to draw samples from the variational distribution \( q(\theta) \). However, in many applications, the Monte Carlo estimate from (6) has high variance (Paisley et al. 2012), even when control variates and Rao-Blackwellization are used to reduce variance. It is simple to mitigate the mean field hypothesis hypothesis in the BBVI method to some extent by grouping blocks of components within \( \theta \) and specifying a dependent multivariate variational distribution within each block. However specifying a single dependent distribution for the entire vector \( \theta \) is typically hard, specially when its components lie in different subspaces of \( \mathbb{R} \) (so a multivariate Gaussian or Student-t would be inappropriate choices for \( q(\theta) \)). We found it crucial to account for full dependence structure in the variational family for the case of semi-parametric Bayesian bridge model. More details are provided in Appendix B.

### 4.2 Reparameterization gradient

An alternative way to derive Monte Carlo estimates for the gradient of the ELBO is described in Kingma and Welling (2013). The authors present the reparameterization trick, which assumes that the parameter vector \( \theta \) can be analytically written as \( \theta = T(\epsilon) \) where \( T(\epsilon) \) is a differentiable deterministic transformation involving the variational parameters \( \psi \) and \( \epsilon \) is a random noise required to have a closed form distribution \( q^*(\epsilon) \) that does not depend on \( \psi \) and is easy to sample from. For example, if \( \theta \sim N(m, S) \), then we can write \( \theta = T_m S(\epsilon) = m + L \epsilon \), with \( L \) being the Cholesky decomposition of the covariance matrix \( S \) (i.e., \( L L^\top = S \) and \( \epsilon \sim N(0, I) \)). Another example of reparameterization is the log-Normal distribution: if \( \theta \sim \log N(m, s^2) \), one can write \( \theta = T_{m,s}(\epsilon) = \exp[m + s \times \epsilon] \) with \( \epsilon \sim N(0, 1) \).

The reparameterization of \( \theta \) allows one to switch the order of the gradient with the expectation when deriving the updating equations for optimizing the ELBO. The gradient of the ELBO under reparameterization becomes

\[
\nabla_{\psi} ELBO(\psi) = E_{q^*(\epsilon)}[\nabla_{\psi} \log p(y, T(\psi(\epsilon)))]
- E_{q^*(\epsilon)}[\nabla_{\psi} \log q^*(T(\psi(\epsilon)))]
\]

If a random sample \( \epsilon^{(1)}, \ldots, \epsilon^{(M)} \) is drawn from \( q^*(\epsilon) \), the Monte Carlo estimate for the gradient is obtained as

\[
\nabla_{\psi} ELBO(\psi) = \frac{1}{M} \sum_{m=1}^{M} \nabla_{\psi} \log p(y, T(\psi(\epsilon^{(m)})))
- \frac{1}{M} \sum_{m=1}^{M} \nabla_{\psi} \log q^*(T(\psi(\epsilon^{(m)}))),
\]

(7)
The variance of the Monte Carlo estimates for the reparameterization gradients tend to exhibit lower variance than the BBVI Monte Carlo estimates, but the difficulty of assuming a full dependence structure on the variational family for different entries of $\theta$ persists. In Appendix C an implementation of the reparameterization method for variational inference on the proposed semi-parametric Bayesian bridge regression model is briefly described.

4.3 Automatic differentiation variational inference

The implementation of ADVI method from Kucukelbir et al. (2015) and Kucukelbir et al. (2017) to the proposed semi-parametric Bayesian bridge model is briefly described in this section. The ADVI shares similarities with the reparameterization method of Kingma and Welling (2013) since it also makes use of reparameterization in order to write the gradient of the ELBO as an expectation to be approximated by Monte Carlo. The key distinction is that the method works in a transformed parameter space that is (ideally) suitable to be modeled as a multivariate normal distribution.

Suppose the original parameters $\theta$ vary on a subset $\Theta \subset \mathbb{R}^d$ with $\Theta \neq \mathbb{R}^d$. This happens for example when one or more entries of $\theta$ lie in constrained subsets of $\mathbb{R}$, say $\mathbb{R}^+$ or the interval $(0, 1)$ for instance. We consider the transformed parameter vector $\xi = T(\theta)$ where $T: \Theta \rightarrow \mathbb{R}^d$ is a diffeomorphism map (differentiable and invertible transformation) such that $\xi$ lies in $\mathbb{R}^d$, with no restrictions. For the mean-field approach, the joint variational distribution of the parameters is the product of its marginals: $q_\phi(\theta) = \prod_{j=1}^d q_{\phi_j}(\theta_j)$ for $\theta = (\theta_1, \ldots, \theta_d)$. On the other hand, ADVI admits dependence structure given by $\theta = T^{-1}(\xi)$ where $\xi \sim N(\mathbf{m}, \Sigma)$ with full covariance matrix $\Sigma$.

In the case of the Bayesian bridge model, we have the original parameters $\theta = (\beta_0, \beta_1, \ldots, \beta_J, \phi, \lambda_1, \ldots, \lambda_J, \alpha_1, \ldots, \alpha_J)^T \in \mathbb{R}^{K_0} \times \mathbb{R}^{K_1+\ldots+K_J} \times (0, +\infty)^{J+1} \times \prod_{j=1}^J (0, C_j)$. A possible choice for $T$ is

$$T(\theta) = (\beta_0, \beta_1, \ldots, \beta_J, \phi, \log \lambda_1, \ldots, \log \lambda_J, \log \frac{\alpha_1}{C_1-\alpha_1}, \ldots, \log \frac{\alpha_J}{C_J-\alpha_J})^T.$$

In the remainder of this section, we will denote $\xi = (\xi_{\beta_0}, \xi_{\beta_1}, \ldots, \xi_{\beta_J}, \xi_{\phi}, \xi_{\lambda_1}, \ldots, \xi_{\lambda_J}, \xi_{\alpha_1}, \ldots, \xi_{\alpha_J})^T$, where $\xi_{\beta_0} = \beta_0$, $\xi_{\beta_j} = \beta_j$, $\xi_{\phi} = \log \phi$, $\xi_{\lambda_j} = \log \lambda_j$, $\xi_{\alpha_j} = \log \frac{\alpha_j}{C_j-\alpha_j}$ for $j = 1, \ldots, J$. The joint distribution $p(y, \theta)$ is defined as in the right-hand side of Eq. (A1) with likelihood $p(y|\beta_0, \beta_1, \ldots, \beta_J, \phi) \sim N(\sum_{j=0}^J X_j \beta_j, \phi^{-1} \mathbf{I})$ and prior $p(\theta) = p(\beta_0)p(\phi)\prod_{j} p(\beta_j|\lambda_j, \phi, \alpha_j) p(\lambda_j) p(\alpha_j)$.

For the ADVI method, we redefine the joint model density in terms of the joint distribution $p(y, \xi)$, here denoted as $\hat{p}(y, \xi)$ to distinguish it from the original joint density $p(y, \theta)$. It follows that

$$\hat{p}(y, \xi) = p(y, \theta) \bigg|_{\theta=T^{-1}(\xi)} \propto |J T^{-1}(\xi)|,$$

where $J T^{-1}(\xi)$ represents the Jacobian of the inverse transformation $T^{-1}: \mathbb{R}^d \rightarrow \Theta$,

$$T^{-1}(\xi) = \left( \xi_{\beta_0}, \xi_{\beta_1}, \ldots, \xi_{\beta_J}, e^{\xi_{\phi}}, e^{\xi_{\lambda_1}}, \ldots, e^{\xi_{\lambda_J}}, \frac{C_j}{1+e^{-\xi_{\alpha_j}}}, \ldots, \frac{C_j}{1+e^{-\xi_{\alpha_J}}} \right)^T$$

which in this case is $|J T^{-1}(\xi)| = e^{\xi_{\phi}} \prod_{j=1}^J C_j e^{\xi_{\lambda_j}}/(1+e^{-\xi_{\alpha_j}})^2$. A multivariate Gaussian variational distribution $q(\xi, m, LL^T) = N(\xi, \mathbf{m}, \mathbf{LL}^T)$ is specified for $\xi$ and the variational parameters are $\psi = (\mathbf{m}, \mathbf{L})$. By reparameterizing $\xi = m + L \epsilon$, the ADVI method enables calculation of the gradient of the ELBO as an expectation, which can be approximated via Monte Carlo:

$$\nabla_\psi \text{ELBO}(y, \psi) = \nabla_\psi \mathbb{E}_{q(\xi)} \left[ \log \hat{p}(y, \xi) - \log q(\xi) \right]$$

$$= \nabla_\psi \mathbb{E}_{\epsilon \sim N(0, I)} \left[ \log p(y, T^{-1}(\epsilon)) + \log |J T^{-1}(\epsilon)| - \log N(\xi, \mathbf{m}, \mathbf{LL}^T) \right]_{\xi=m+L\epsilon}$$

$$= \mathbb{E}_{\epsilon \sim N(0, I)} \left\{ \nabla_\psi \left[ \log p(y, T^{-1}(\epsilon)) + \log |J T^{-1}(\epsilon)| - \log N(\xi, \mathbf{m}, \mathbf{LL}^T) \right]_{\xi=m+L\epsilon} \right\}$$

$$\approx \frac{1}{M} \sum_{\ell=1}^M \nabla_\psi \left[ \log p(y, T^{-1}(\epsilon)) + \log |J T^{-1}(\epsilon)| - \log N(\xi, \mathbf{m}, \mathbf{LL}^T) \right]_{\xi=m+L\epsilon(\ell)}$$

(9)

where $\epsilon^{(\ell)} \overset{iid}{\sim} N(0, I)$, $\ell = 1, \ldots, M$.

One can also easily compute the stochastic gradient approximation of (9) for a random minibatch $\tilde{y}$ of size $K < n$ from the full response data vector $y = (y_1, \ldots, y_n)^T$. There are many ways in which one can draw random batches $\tilde{y}$. The only requirement is that the Monte Carlo estimate for the gradient based on $\tilde{y}$ is unbiased for the full gradient based on $y$. 

\[ \text{Springer} \]
In this work, we randomly permute the indexes of $y$ at the beginning of each epoch to form $y^* = (y_{i_1}, \ldots, y_{i_n})^T$ where $(i_1, \ldots, i_n)$ is a permutation of $(1, \ldots, n)$ and pick blocks of $K$ consecutive observations from $y^*$ to form the random batches $y$ at each iteration. See Algorithm 1 for a more in depth description of how the batches are drawn. The Monte Carlo estimate for stochastic gradient of the ELBO becomes

$$\tilde{\nabla}_\psi \text{ELBO}(\tilde{y}, \psi) = \frac{n}{KM} \sum_{\ell=1}^M \nabla_\psi \log p(\tilde{y} \mid T^{-1}(\xi)) \bigg|_{\xi=m+L^{(\ell)}} + \frac{1}{M} \sum_{\ell=1}^M \nabla_\psi \left[ \log p(T^{-1}(\xi)) + \log \mathcal{J}_{T^{-1}(\xi)} \right] - \log N(\xi; m, LL^\top) \bigg|_{\xi=m+L^{(\ell)}}.$$  

(10)

In Eq. (10), the terms $p(\tilde{y} \mid T^{-1}(\xi))$ and $p(T^{-1}(\xi))$ denote the densities $p(\tilde{y} \mid \theta)$ and $p(\theta)$ evaluated at $\theta = T^{-1}(\xi)$. The multiplicative factor $\frac{n}{KM}$ rescales the batch gradient to make it an unbiased estimate for the full posterior $p(\theta \mid y)$ based on $n$ observations rather than the batch size posterior, which is based on $M$ data points only. Algorithm 1 summarizes the steps of the proposed ADVI algorithm for semi-parametric Bayesian bridge regression. Evaluating Eq. (10) is central to Algorithm 1 so the details of its implementation via automatic differentiation are discussed Appendix D. The full expression of the ELBO in Eq. (10) (step 8 of Algorithm 1) is derived in Eqs. (D1) and (D2). In summary, automatic differentiation calculates the gradients with respect to $\psi$ numerically in an automatic fashion without the need of writing down the analytic expression of the gradients (Baydin et al. 2018). The update Eq. in step 8 corresponds to the basic stochastic gradient ascent, although any gradient based method such as Adam, Adagrad and others could be used.

The ADVI method was found to suit well to the Bayesian bridge semi-parametric proposed model, both in terms of computational speed and also in terms of quality of posterior approximation, due to the full dependence structure incorporated in the variational family. These findings are documented in the simulation study and in the application to real data presented in Sects. 5 and 6.

5 Numerical experiments

In this section, posterior estimates under the MCMC approach based on Mallick and Yi (2018) as described in Sect. 3 and the proposed ADVI for semi-parametric Bayesian bridge, as described in Sect. 4.3 are compared under different scenarios, focusing on large datasets. We also compare results with the

Algorithm 1: ADVI algorithm for semi parametric Bayesian bridge regression

Input:
Data: $y \in \mathbb{R}^n$, $X_j \in \mathbb{R}^{n \times \mathbb{R}^{K_j}}$, $j = 0, \ldots, J$, Learning rate: $\delta \in (0, 1)$, Batch size: $K \in \mathbb{N}$, Number of Monte Carlo samples: $M \in \mathbb{N}$, Number of epochs $N_{epochs} \in \mathbb{N}$, Number of samples from $q_\theta(\theta)$: $S \in \mathbb{N}$, Initial values for variational parameters: $\psi = (m, L)$.

1. Calculate number of batches per epoch as $N_{batches} = \lfloor n/K \rfloor$, where $n$ is the number of observations.

2. for epoch in $\{1, \ldots, N_{epochs}\}$ do

3. Randomly permute the response data $y = (y_1, \ldots, y_n)$ and permutate $\tilde{y} = (y_{b1}, \ldots, y_{bK})$ accordingly.

4. for $b \in \{1, \ldots, N_{batches}\}$ do

5. Select the $b$-th batch of data from $y^*$ as $\tilde{y} = (y_{b1}, \ldots, y_{bK})$.

6. Sample $\epsilon^{(t)} \sim N(0, I), t \in \{1, \ldots, M\}$.

7. Evaluate $\tilde{\nabla}_\psi \text{ELBO}(\tilde{y}, \psi)$ from equation (10) via automatic differentiation using equations (D1) and (D2) (see Appendix D).

8. Gradient ascent iteration:

   $\psi \leftarrow \psi + \delta \times \tilde{\nabla}_\psi \text{ELBO}(\tilde{y}, \psi)$.

end

9. end of epoch (whole pass through data).

end

10. Sample $\epsilon^{(s)} \sim N(0, I), s = 1, \ldots, S$.

11. Extract $m$ and $L$ from updated $\hat{\psi} = (m, L)$ (see Appendix D).

12. Generate samples $\theta^{(s)} \sim q_\psi(\theta)$ by taking $\theta^{(s)} = T^{-1}(m + L^{(s)})$, $s = 1, \ldots, S$, where $T^{-1}$ is defined in equation (8).

Output:
Updated variational parameters $\hat{\psi} = (m, L)$ and i.i.d. samples from variational distribution $\theta^{(s)} = T^{-1}(m + L^{(s)})$, $s = 1, \ldots, S$.

Bayesian formulations for generalized additive models provided by the R packages: mgcv (Wood 2012, 2017), brms (Bürkner 2017), and INLA (Rue et al. 2009). Finally, we exemplify the proposed procedure in a synthetic dataset with multiple covariates.

5.1 Scenario 1: small datasets

In this scenario, multiple small datasets are simulated and used to draw comparisons between the proposed ADVI for semi-parametric regression as described in Sect. 4.3 and the MCMC scheme described in Sect. 3. One hundred datasets were simulated, with $n = 100$ observations each, according
to the model defined by $y_i \overset{ind}{\sim} N(f_\beta(x_i), \sigma^2)$, $i = 1, \ldots, n$, where $f_\beta(x)$ represents the non-parametric effects of the covariate $x$, modeled by a B-spline with coefficients $\beta$. The observational variance parameter $\sigma^2$ was fixed at 1 when simulating the data and the nodes for the B-spline basis were regularly spaced from $-0.066$ to 1.066 with 0.033 units of consecutive distance, which implied 34 B-spline coefficients. We also fixed the covariates $x_i$, $i = 1, \ldots, 100$ on a regular grid over the interval (0, 1). The B-spline coefficients $\beta_k$, $k = 1, \ldots, 34$ were sampled as follows: $\beta_1, \ldots, \beta_{10} \overset{iid}{\sim} N(5, 2)$, $\beta_{16}, \ldots, \beta_{25} \overset{iid}{\sim} N(10, 2)$, $\beta_{31}, \ldots, \beta_{34} \overset{iid}{\sim} N(4, 0.25)$ and $\beta_k = 0 \forall k \in \{11, \ldots, 15\} \cup \{26, \ldots, 30\}$ with simulated values of the coefficients truncated to the nearest integers. The true simulated values for $\beta_k$, $k \in \{1, \ldots, 34\}$ can be seen in Fig. 15 and are shared for all 100 simulated datasets, which implies only one underlying B-splines curve, along which all the replicas for $y = (y_1, \ldots, y_n)$ are simulated. Since the B-splines do not include intercept, the parameter $\beta_0$ does not need to be included in the model.

The prior hyperparameters specified for Bayesian inference under both MCMC and Variational Bayes (VB) approach via ADVI were fixed as $\alpha_0 = a_0 = a_\phi = b_0 = b_\lambda = b_\phi = 1$, therefore representing vague prior knowledge about $\alpha$, $\lambda$ and $\phi$. Figure 2 shows the posterior inference for $\mu(x) = f_\beta(x)$, $x \in (0, 1)$ based on the first replica of the simulated data. MCMC and VB produce very similar posterior estimates for $\mu(x)$, with noticeable differences only at the beginning and end of the series. The same holds for the chosen alternative inference methods: Integrated Nested Laplace Approximation (INLA) by Rue et al. (2009) from the inla R package, the brms R package by Bürkner (2017) (which implements MCMC within Stan), mgcv R package (Wood 2012) and the smooth.spline R function (default options were used for all packages).

Next, Fig. 3 compares the ADVI with bridge penalization on the spline basis coefficients as in Eq. (2), to its analogous Bayesian p-splines formulation by Lang and Brezger (2004) (MCMC) and to the off-the-shelf smooth.spline R function.

The objective is to compare the 3 distinct forms of penalization: i) directly on the coefficients (proposed ADVI); ii) on the second order differences on consecutive coefficients (P-splines) and iii) integral of second derivative (smooth.spline). The fitted curves are very similar, with the proposed ADVI for Bayesian bridge penalization being slightly less smooth than the other two approaches. Furthermore, the observed mean absolute prediction error when estimating $\mu = f_\beta(x)$ was lower when using the ADVI (0.4263) when compared to Bayesian P-splines (0.5648) and smoothing splines (0.5296).

Figure 4 shows that the posterior marginal distributions for the spline coefficients obtained by the proposed variational procedure are more similar to MCMC than INLA and brms and that the marginals capture the true values of the coefficients. Figure 5 shows MCMC and ADVI posterior marginals for $\lambda, \phi$ and $\alpha$. The marginal posterior distributions for $\phi$ and $\lambda$ under VB and MCMC are similar, while $\alpha$ posterior estimates via VB are less dispersed in comparison with MCMC marginal posterior densities. Finally, joint posterior uncertainties for the B-spline coefficients are also very similar under both approaches, as exemplified in Fig. 6. Figure 6 also shows how important it is to account for full dependence structure on the variational family as allowed by ADVI as opposed to the simpler and more restrictive mean-field assumption. Notice, for example, that coefficients associated with first and second-order neighbor B-spline knots exhibit very strong negative and positive correlations, respectively (main diagonal and first secondary diagonal panels, respectively), which could not be captured if the mean-field assumption was used. Further examples comparing ADVI and mean-field assumption regarding estimation of posterior dependency in other models can be found in Kucukelbir et al. (2017). Figure 7 shows that dependency structures involving the remaining parameters are also well recovered by ADVI, except for the bridge penalization parameter $\alpha$. Despite the discrepancies regarding estimation of $\alpha$ by the VB and MCMC methods, there is very little difference regarding the goodness of fit to the simulated data (see Fig. 2 for example).
Fig. 4 Marginal posterior distributions for $\beta_k$, $k = 1, \ldots, 7$ based on the first replicate of the simulated data. Red dots denote the true values of the parameters. The marginal densities for the remaining coefficients can be seen in Fig. 15.

Fig. 5 Marginal posterior distributions for $\phi$, $\lambda$ and $\alpha$ based on the first replicate of the simulated data. Red dashed curves represent VB and black curves represent MCMC marginal posterior approximations, which might indicate that the data brings little information about $\alpha$.

Furthermore, similar conclusions can be drawn from the other 99 replicates. Posterior point estimates produced by the proposed ADVI and by the baseline MCMC on all 100 replicates are compared in Figs. 8 and 16 in the appendix. Point estimates for $\beta_k$, $k = 1, \ldots, 34$ are very similar under both methods, except for $\beta_{34}$. However, a closer look reveals that the 34-th column of the B-spline regression matrix $X$ has all entries equal to 0, except the last one, which makes $\beta_{34}$ very hard to estimate as it has very little impact on the mean function $f_\beta(x)$. The level of agreement between the point estimates produced by MCMC and the proposed ADVI for $\lambda$ and $\alpha$ are not as high as for the regression coefficients $\beta_k$, $k = 1, \ldots, 34$.

5.2 Scenario 2: varying sample sizes

In this section, the effect of sample sizes in computational times of the MCMC and ADVI algorithms in the context of estimation of the semi-parametric Bayesian bridge is investigated, focusing on large datasets. The specifications for the simulated data are the same as described in Sect. 5.1, except that now the sample sizes of the simulated data vary as $n \in \{10^3, 10^4, 5 \times 10^4, 10^5, 5 \times 10^5, 10^6\}$. For all six cases, the MCMC was ran for 5000 iterations (except for $n=1000$, which required 50000 iterations until we could get reasonable evidence of convergence), initializing $\beta$ and $\phi$ at their OLS estimates under no regularization. For the VB implementation, we always used learning rate 0.01 for Adam and $M = 100$ Monte Carlo samples to approximate expectations in the calculation of the ELBO. Distinct batch sizes and number of epochs were used for each sample size according to Table 1, which implies 1000 to 10000 iterations in total for ADVI depending on the value of $n$ to be compared with the results obtained after 5000 iterations of MCMC. For comparison, implementation of brms, mgcv and INLA are also included, however, it is important to highlight that these 3 procedures are not directly comparable in terms of computational times since the models are not the same as the
Fig. 7 ADVI and MCMC bivariate joint posteriors for $\beta_1$, $\beta_2$, $\phi$, $\lambda$, $\alpha$ semi-parametric Bayesian bridge regression implemented in ADVI and MCMC.

Regarding computational time, the proposed ADVI algorithm for semi-parametric Bayesian bridge regression is orders of magnitude faster than MCMC, as shown in Table 2.

ADVI remains faster even when running more iterations than MCMC and also when VB times are adjusted to account for the 5000 iterations ran under MCMC.

It is important to address the quality of the variational approximations to the true posterior distribution. In this sense, Fig. 17 shows the p-values obtained when comparing (via Kolmogorov-Smirnov tests) marginal posterior predictive distributions estimated by VB and MCMC. The shapes of the histograms are approximately uniform with low prevalence of small p-values, as expected under the null hypothesis that the distributions under MCMC and VB are the same. For example, the empirical proportions of p-values below 0.05 are close to 0.05, which corresponds to the expected proportion of false discoveries when $H_0$ is true and a type 1 error of 0.05 is fixed. In summary, the proposed VB approach is capable of reaching convergence faster than MCMC, using less epochs and providing an accurate approximation of the posterior. The brms is the slowest procedure with computational times much higher than the others. Computation under mgcv is faster but relatively simpler than the other methods and it is a frequentist inference procedure. Notice that INLA is the fastest Bayesian method, however the Newton Raphson procedure within INLA did not converge in case of $n \geq 50,000$.

Fig. 8 Point estimates for all parameters under MCMC and VB for each one of the 100 replicas. The black lines represent the identity function. The scatterplots for the remaining coefficients can be seen in Fig. 16 in the Appendix.

Fig. 9 Simulated mean of $y$ as a function of covariates $x_1$ and $x_2$ according to Sect. 5.3. Posterior point estimate for the average of $y$ under the proposed MCMC approach.

 Springer
Fig. 10 Comparison between posterior inference on non-linear effects of covariates under VB, MCMC and GAM when fitting the simulated data from Sect. 5.3

Fig. 11 Comparison between posterior inference on $\alpha_1, \alpha_2, \lambda_1, \lambda_2, \phi$ under VB and MCMC when fitting the simulated data from Sect. 5.3

Fig. 12 Full Energy Charge data (measured hourly)

Fig. 13 VB 95% credibility bands and posterior mean for the average response. Only the first 1000 observations of Energy Charge data are shown.
Fig. 14  VB 95% credibility bands and posterior mean for the average response. Only the last 1000 observations of Energy Charge data are shown.

Fig. 15  Marginal posterior distributions for $\beta_k$, $k = 1, \ldots, 34$ based on the first replicate of the simulated data. Red dots denote the true values of the parameters.
In this scenario, a dataset with \( n = 1000 \) observations \((y_i, x_{i1}, x_{i2}), i = 1, \ldots, n\) were simulated from the model \( y_i = \beta_0 + f_1(x_{i1}; \tau_1) + f_2(x_{i2}; \tau_2) + \epsilon_i, \epsilon_i \sim N(0, \sigma^2)\), where \( f_i(\cdot; \tau_i) \) denotes a realization of a 1-dimensional Gaussian process with mean 0 and covariance function \( \text{Cov} : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}, \text{Cov}(z, z') = \exp[-\tau_i^{-1}(z - z')^2]\). In the sim-
### Table 2: Computational times (in seconds) for MCMC and ADVI implementations according to the size of the simulated data ($n$)

| $n$   | MCMC  | ADVI  | ADVI (5000 iterations) | brms | mgcv | INLA |
|-------|-------|-------|------------------------|------|------|------|
| 1,000 | 219s* | 6s    | 15s                    | 72s  | 0.02s | 0.6s |
| 10,000| 64s   | 6s    | 15s                    | 462s | 0.3s  | 1.2s |
| 50,000| 294s  | 15s   | 15s                    | 3,500s | 1s   | 3s   |
| 100,000| 618s  | 13s   | 65s                    | 8,071s | 1.5s | 6s   |
| 500,000| 2,744s| 76s   | 76s                    | 45,233s | 6.5s | NA   |
| 1,000,000| 4,570s| 143s  | 72s                    | 103,680s | 12s | NA   |

*In this case we ran 50000 mcmc iterations instead of 5000 because of lack of evidence for convergence under MCMC.

---

**Figure 17:** $p$-values for the Kolmogorov-Smirnov test for comparison of each marginal posterior predictive distributions under ADVI and MCMC. In red, we highlight $P$-type 1 error) = 5%. The expression $P(\text{Rej. } H_0)$ represents the proportion of times we would reject the null hypothesis of equal distributions under ADVI and MCMC if we were to reject whenever $p$-value < 0.05.

---

ulation, we used $\tau_1 = \tau_2 = 1$. The objective is to investigate how the proposed semi-parametric bridge regression model performs when there are more than one covariate affecting the response in a non-parametric way. The results presented here assume $n_1 = n_2 = 100$ knots for each B-spline non-parametric effect. MCMC was run for 50000 iterations with 30000 iterations being excluded as burn-in with further spacing the sampled values from 200 to 200 in order to produce approximately independent Monte Carlo chain (see Fig. 18 in Appendix E). The proposed VB approach was run with batch size 1000 (therefore using full gradient updates) with Monte Carlo sample size $M = 100$ and 5000 iterations in total. MCMC via brms uses 10000 iterations with the first 2500 being excluded as burn-in and spacing of 20 to reach approximate independence (see Fig. 18).

Figure 9 shows that the underlying simulated surface for the mean of the response variable $y$ as a function of $x_1$ and $x_2$ is accurately estimated by the MCMC and VB approaches. When comparing the model formulation under MCMC, VB, and the point estimates from GAM, Figure 10 shows that they yield virtually indistinguishable estimates for the non-parametric effects. While GAM penalizes the second-order differences of the basis functions’ coefficients, the bridge penalization used in the MCMC and VB leads to the same fit. Also, Fig. 11 shows VB and MCMC produce similar approximations to the posterior marginal distributions of $\alpha_1$, $\alpha_2$ and $\phi$, while $\lambda_1$ and $\lambda_2$ (standardized) uncertainties seem to be underestimated by VB when compared with MCMC. We highlight that the proposed VB approach was able to identify that both $\lambda_1$ and $\lambda_2$ are very small with high probability,
Table 3 Computational times (in seconds) and out of sample mean absolute error (MAE) with respect to the true underlying mean surface for MCMC, ADVI, mgcv, INLA and brms implementations on simulated dataset of Sect. 5.3.

| Method | MCMC  | ADVI  | brms  | mgcv  | INLA  |
|--------|-------|-------|-------|-------|-------|
| Time   | 1899s*| 134s  | 97s** | 0.34s | 3.81s |
| MAE    | 0.032 | 0.032 | 0.033 | 0.032 | 0.083 |

*Computational times of MCMC refer to 50000 iterations. **Computational times of brms refers to 10000 iterations.

as well as in MCMC. In addition, the proposed procedure needed only a fraction of MCMC computational time to fit the covariate effects, which are virtually indistinguishable from those obtained via MCMC (see Fig. 10). Such useful results were obtained besides the slight discrepancies between VB and MCMC on $\lambda_1$ and $\lambda_2$. This could be due to the choice of a log normal distribution on the marginals for $\lambda_1$ and $\lambda_2$ by our proposed variational family. More flexible variational families can be found in Yin and Zhou (2018) however at the expense of a higher computational cost. Nevertheless, a deep study of flexible variational families will not be addressed in this work.

Table 3 compares the computational times and the performance of the methods in terms of mean absolute error. MCMC takes 1899 s to run 50000 iterations while ADVI takes 134 s requiring 5000 iterations to converge and produce iid approximate samples from the posterior. Notice that VB is faster than MCMC even if we would use the same number of iterations for MCMC and VB as comparison, ignoring the fact that MCMC needed large spacing among its sampled values to approximately achieve independence (see Fig. 18). In terms of the performance measured in absolute mean error with respect to the underlying simulated mean surface, MCMC, ADVI and mgcv produce the smallest errors. Despite being the fastest Bayesian inference algorithm, INLA produced the biggest mean absolute error. We also measured computational times for brms, mgcv and INLA, resulting in 97 s, 0.34s and 3.81s respectively. Notice, however, that the computational times obtained by brms, mgcv and INLA are not directly comparable with the MCMC and ADVI because they fit different (less flexible) models: INLA assumes second order $\ell_2$-based penalization on consecutive coefficients of the spline basis functions (no estimation of $\alpha_1$ and $\alpha_2$), mgcv does classical GAM fit and brms assumes no penalization.
6 Real data application

This section illustrates the use of the proposed ADVI inference procedure for semi-parametric Bayesian bridge regression on a large real dataset. The data consists of hourly measured Energy Charges starting from 2014-04-06 (10pm) to 2022-03-31 (11pm) averaged over stations in the Northern region of Brazil (see Fig. 12). In total, there are \( n = 69717 \) observations. The dataset is maintained by Operador Nacional do Sistema Elétrico (ONS) and can be obtained at https://dados.ons.org.br/dataset/carga-energia.

The data exhibits strong seasonal patterns with multiple frequencies due to periodicities in energy consumption according with time of the day, day of the week, season of the year and possibly more. We model the seasonal harmonics as the cosine and sine Fourier basis representation for weekly periodicity (period = 24 \( \times \) 7 = 168) following West and Harrison (1997). To capture overall level changes in the series, we included a cubic B-spline with one knot at every 100 h for a total of 700 knots. The B-spline coefficients are subjected to bridge penalization while the coefficients of the Fourier harmonics basis functions are not penalized. In total, the resulting covariate matrix has \( p = 868 \) columns.

The proposed ADVI inference approach took approximately 831 s (less than 14 min) to complete 2000 epochs using 100 Monte Carlo samples to estimate the gradient of the ELBO. On the other hand, MCMC takes 69239 s (19.23 h) to run the same 2000 iterations (epochs). Posterior estimates for the beginning and end of the series are shown in Figs. 13 and 14.

7 Conclusions and future work

The present work developed a variational inference procedure based on ADVI for Bayesian inference in semi-parametric bridge regression models. The use of small batches of data at each iteration of the training algorithm reduces computational time in comparison with a more traditional MCMC approach. Full Bayesian inference is preserved so joint uncertainty estimates for all model parameters are available. It was verified in the simulation study that the joint posterior is well approximated by the proposed variational family. In this context, investigation of the coverage of confidence bands is left as a venue of future research.

Some directions for future work include (i) the extension to non-Gaussian distributions for the response variable; (ii) expanding flexibility of the variational family with other approaches such as the semi implicit variational inference from Yin and Zhou (2018); (iii) considering different spline basis functions to represent non-parametric effects of covariates including GAM models with tensor splines avoiding MCMC and RJMCMC. See details in Denison et al. (1998), Dias and Gamerman (2002), Li and Villani (2013) and references therein.

Acknowledgements This paper was partially supported by Fapesp Grants (R. Dias) 2018/04654-9, (R. Dias and H. S. Migon) 2019/10800-0 and (R. Dias) 2019/00787-7.

Author Contributions All authors contributed with the statistical analysis and the writing/revision of the manuscript.

Appendix A Details for the MCMC algorithm

A.1 Full conditionals

The proposed ADVI inference approach took approximately 831 s (less than 14 min) to complete 2000 epochs using 100 Monte Carlo samples to estimate the gradient of the ELBO. On the other hand, MCMC takes 69239 s (19.23 h) to run the same 2000 iterations (epochs). Posterior estimates for the beginning and end of the series are shown in Figs. 13 and 14.

\[
p(\beta, \beta_0, \lambda, \phi, \alpha | y) \propto p(y | \beta, \beta_0, \phi) p(\beta_0) \times p(\beta | \lambda, \phi, \alpha) p(\alpha | \lambda, \phi) p(\alpha),
\]

where \( \lambda = (\lambda_1, \ldots, \lambda_J) \) and \( \alpha = (\alpha_1, \ldots, \alpha_J) \).

The joint posterior distribution with the uniform-gamma variable augmentation is expressed as

\[
p(\beta, \beta_0, \lambda, \phi, \alpha | y) \propto p(y | \beta, \beta_0, \phi) p(\beta_0) \times p(\beta | \alpha, \phi) p(\alpha | \lambda, \phi) p(\alpha).
\]

With \( \alpha \) fixed, all full conditional distributions are analytically available, therefore a straightforward Gibbs sampler scheme can be implemented. The procedure from Damien and Walker (2001) is used to get a numerically stable sampler for the truncated Gaussian and truncated exponential distributions that appear in the Gibbs sampler scheme that follows.

**Posterior full conditional distribution of** \( \beta_j \)

\[
(\beta_j | y, \beta_0, \phi, \alpha) \sim N \left( (X_j^\top X_j)^{-1} X_j^\top (y - X_0 \beta_0 - \sum_{k \neq j} X_k \beta_k), (X_j^\top X_j)^{-1} \phi^{-1} \right) I_{B_j},
\]

where

\[
B_j = \left\{ (\beta_{j1}, \ldots, \beta_{jK_j}) \in \mathbb{R}^{K_j} : |\beta_{jk}| < u_j^k \phi^{-\frac{1}{2}}, \forall k \in \{1, \ldots, K_j\} \right\}
\]
and \( N(\mathbf{\mu}, \Sigma)_{\mathcal{A}} \) denotes the multivariate Gaussian distribution with mean \( \mathbf{\mu} \) and covariance matrix \( \Sigma \), truncated in the set \( \mathcal{A} \).

**Posterior full conditional distribution of \( \beta_0 \)**

\[
(\beta_0 | \mathbf{y}, \phi, \beta) \sim N \left( \left( \mathbf{X}_0^T \mathbf{X}_0 \phi + \Sigma_0^{-1} \right)^{-1} \times \mathbf{X}_0^T \left( \mathbf{y} - \sum_{j=1}^{J} \mathbf{X}_j \beta_j \right) + \Sigma_0^{-1} \mathbf{\mu}_0 \right, \left( \mathbf{X}_0^T \mathbf{X}_0 \phi + \Sigma_0^{-1} \right)^{-1}
\]

**Posterior full conditional distribution of \( \phi \)**

\[
(\phi | \mathbf{y}, \beta, \beta_0, \mathbf{u}, \alpha) \sim Ga \left( \frac{n}{2} + \frac{1}{2} \sum_{j=1}^{J} K_j + a_{\phi}, \frac{1}{2} RSS(\beta, \beta_0) + b_{\phi} \right)
\]

where \( RSS(\beta, \beta_0) = (\mathbf{y} - \sum_{j=1}^{J} \mathbf{X}_j \beta_j - \mathbf{X}_0 \beta_0)^T(\mathbf{y} - \sum_{j=1}^{J} \mathbf{X}_j \beta_j - \mathbf{X}_0 \beta_0) \) denotes the residual sum of squares and \( Ga(a, b)_{\mathcal{A}} \) denotes the Gamma distribution with mean \( a/b \) and variance \( a/b^2 \) truncated in the set \( \mathcal{A} \).

**Posterior full conditional distribution of \( u_{j\ell} \)**

\[
(u_{j\ell} | \mathbf{y}, \lambda_j, \beta, \phi, \alpha_j) \sim Exp(\lambda_j) \mathbb{1}_{\{0, \min_{\ell} \lambda_j | \beta_{j\ell} | - 2\}}
\]

**A.2 Estimation of \( \alpha_j \)**

There is no possible choice of prior for \( \alpha_j \) that leads to an analytically available full conditional distribution on \( \alpha_j \). Therefore a Metropolis-Hastings (MH) step is proposed for \( \alpha_j \). The MH algorithm requires specification of a proposal distribution \( q(\alpha_j^* | \alpha_j^{(i)}) \) where \( \alpha_j^{(i)} \) denotes the value of \( \alpha_j \) in the current iteration \( i \) and \( \alpha_j^* \) denotes the proposed value for \( \alpha_j \) according to the proposal density \( q(\cdot | \alpha_j^{(i)}) \).

**A.2.1 Marginalized MH proposal**

When sampling \( \alpha_j \), we consider the reduced parameter vector \( \theta = (\alpha, \beta, \beta_0, \phi, \lambda) \) marginalizing out the auxiliary variables \( \mathbf{u} \). Alternatively, appendix A.2.2 describes a proposal \( q(\cdot | \alpha_j^{(i)} \) based on the full parameter vector \( \theta = (\alpha, \beta, \beta_0, \phi, \lambda, \mathbf{u}) \) including the augmented variables \( \mathbf{u} \). It was found crucial to marginalize \( \mathbf{u} \) out of the model when sampling \( \alpha_j \) in the MCMC.

We denote by \( \theta^* = (\alpha^*, \alpha_{-j}, \beta, \beta_0, \phi, \lambda) \) the non-augmented parameter vector at the proposed \( \alpha_j = \alpha_j^* \), with \( \alpha_{-j} \) denoting the vector \( \alpha \) excluding its \( j \)-th entry and by \( \theta = (\alpha, \beta, \beta_0, \phi, \lambda) \) the parameter vector with \( \alpha_j \) representing the current iteration. Notice that the auxiliary variables \( \mathbf{u} \) are marginalized out. The proposed \( \alpha_j^* \) is accepted with probability \( \rho(\alpha_j^* | \alpha_j) = \min(1, r(\alpha_j^* | \alpha_j)) \) where

\[
r(\alpha_j^* | \alpha_j) = \frac{p(y | \theta^*) p(\theta^*) q(\alpha_j | \alpha_j^*)}{p(y | \theta) p(\theta) q(\alpha_j | \alpha_j)}
\]

We chose a prior on \( \alpha_j \) given by \( \alpha_j = C_j \eta_j \) where \( \eta_j \sim Beta(\alpha_j, b_n) \). The MH proposal \( q(\alpha_j^* | \alpha_j) \) is derived from a Gaussian random walk on \( v = \log \frac{\alpha_j^*}{\alpha_j} \), i.e., \( q(v^* | v) = N(v^*; v, W) \) where \( N(x; \mu, \sigma^2) \) denotes the Gaussian density with mean \( \mu \) and variance \( \sigma^2 \) evaluated at \( x \). It is easy to show that the Gaussian random walk \( q(v^* | v) \) on \( v \) implies

\[
q(\alpha_j^* | \alpha_j) = N \left( \log \frac{\alpha_j^*}{\alpha_j}; \log \frac{\alpha_j}{\alpha_j}, W_j \right) \times \frac{C_j}{\alpha_j (\alpha_j - C_j)}.
\]

After simplifications,

\[
r(\alpha_j^* | \alpha_j) = \frac{\alpha_j^* K_j}{\alpha_j K_j} \times \frac{\Gamma(\alpha_j^{-1} K_j)}{\Gamma(\alpha_j^{-1} K_j + \alpha_j)} \times \frac{K_j}{\alpha_j - C_j} \times \exp \left\{ -\frac{\sum_{\ell=1}^{K_j} \left( \beta_{j\ell} \mid \alpha_j^* \right)^2 \phi_j^2 - \left( \beta_{j\ell} \mid \alpha_j \right)^2 \phi_j^2}{\alpha_j^* (\alpha_j^* - C_j)^b_n} \right\}
\]

\[
\times \frac{\alpha_j^{a_n} (C_j - \alpha_j)^{b_n}}{\alpha_j^* (\alpha_j - C_j)^{b_n}}.
\]

**A.2.2 Non-marginalized MH proposal**

Section A.2.1 described a MH scheme for \( \alpha_j \) taking the advantage of the marginalization of the auxiliary variables \( \mathbf{u} \). This section describes a simpler alternative that does not marginalize \( \mathbf{u} \). However, we could not get well mixing Markov chains by using the simpler proposal described here. We chose a prior on \( \alpha_j \) given by \( \alpha_j = C_j \eta_j \) where \( \eta_j \sim Beta(\alpha_j, b_n) \). The MH proposal \( q(\alpha_j^* | \alpha_j) \) is derived
from a Gaussian random walk on \( v = \log \frac{\alpha_j}{c_j - \alpha_j} \), i.e., \( q(v^* | v) = N(v^*; v, W) \) where \( N(x; \mu, \sigma^2) \) denotes the Gaussian density with mean \( \mu \) and variance \( \sigma^2 \) evaluated at \( x \). It is easy to show that the Gaussian random walk \( q(v^* | v) \) on \( v \) implies
\[
q(\alpha_j^* | \alpha_j) = N \left( \log \frac{\alpha_j^*}{c_j - \alpha_j^*}; \log \frac{\alpha_j}{c_j - \alpha_j}, W \right) \times \frac{c_j}{\alpha_j^*(c_j - \alpha_j^*)}.
\]

After simplifications,
\[
r(\alpha_j^* | \alpha_j) = \lambda_j \left( \frac{\alpha_j^*}{c_j - \alpha_j^*} \right) \times \frac{\Gamma \left( \frac{1}{\alpha_j} + 1 \right)^{K_j}}{\Gamma \left( \frac{1}{\alpha_j^*} + 1 \right)^{K_j}} \times \frac{\alpha_j^{\alpha_j}(C_j - \alpha_j)^{b_0}}{\alpha_j^{\alpha_j^*}(C_j - \alpha_j^*)^{b_0}} 1(m \leq \alpha_j^* \leq M),
\]

where
\[
m = \max \left\{ 0 \cup \left\{ \log(u_{jk}) : k \in S_- \right\} \right\},
M = \min \left\{ \left\{ \log(u_{jk}) : k \in S_+ \right\} \right\},
S_- := \{ j = 1, \ldots, K_j : 0 < |\beta_j k | < \phi^{-1/2} \},
S_+ := \{ j = 1, \ldots, K_j : |\beta_j k | > \phi^{-1/2} \}.
\]

It is important to notice that \( S_- \) does not include values of \( k \) such that \( \beta_j k = 0 \).

Appendix B Details of BBVI for semi-parametric Bayesian bridge

The details for implementation of BBVI for the proposed semi-parametric Bayesian bridge are described in this section. Only the case with \( J = 1 \) is shown, although the calculations could be easily extended to the multivariate case where \( J \in \mathbb{N} \).

Under mean-field, the proposed marginal variational distributions are \( q(\beta | m^*_\beta, s^*_\beta) = N(\beta | m^*_\beta, s^*_\beta), q(\beta_0 | m^*_\beta_0, s^*_\beta_0) = N(\beta_0 | m^*_\beta_0, s^*_\beta_0), q(\phi | a^*_\phi, b^*_\phi) = \text{Ga}(\phi | a^*_\phi, b^*_\phi), q(\lambda | a^*_\lambda, b^*_\lambda) = \text{Ga}(\lambda | a^*_\lambda, b^*_\lambda), q(\alpha | a^*_\alpha, b^*_\alpha) = C_1 \times \text{Beta}(\alpha | a^*_\alpha, b^*_\alpha). \)

We now describe the analytical expressions for the gradient of \( \log q(\theta) \) with respect to each variational parameter. In the equations below, \( \text{dig}(\cdot) \) denotes the digamma function, i.e., the derivative of the log gamma function.
\[
\nabla_{\phi} \log q(\beta | m^*_\beta, S^*_\beta) = \frac{1}{2} (\beta - m^*_\beta)(\beta - m^*_\beta)^T + \frac{1}{2} S^*_\beta
\]
\[
\nabla_{m^*_\beta} \log q(\beta | m^*_\beta, S^*_\beta) = S^{-1}_\beta (\beta - m^*_\beta)
\]
\[
\nabla_{s^*_\beta} \log q(\beta | m^*_\beta, S^*_\beta) = \frac{1}{2} S^{-1}_\beta - \frac{1}{2} (\beta_0 - m^*_\beta_0) \times (\beta_0 - m^*_\beta_0)^T
\]
\[
\nabla_{m^*_\beta_0} \log q(\beta | m^*_\beta_0, S^*_\beta_0) = S^{-1}_\beta (\beta_m^*_0 - \beta^*_\beta)
\]
\[
\nabla_{s^*_\beta_0} \log q(\beta | m^*_\beta_0, S^*_\beta_0) = S^{-1}_\beta (\beta_0 - m^*_\beta_0)
\]
\[
\nabla_{a^*_\phi} \log q(\phi | a^*_\phi, b^*_\phi) = \frac{a^*_\phi}{b^*_\phi} - \phi
\]
\[
\nabla_{b^*_\phi} \log q(\phi | a^*_\phi, b^*_\phi) = \frac{a^*_\phi}{b^*_\phi} - \phi
\]
\[
\nabla_{a^*_\lambda} \log q(\lambda | a^*_\lambda, b^*_\lambda) = \frac{a^*_\lambda}{b^*_\lambda} - \lambda
\]
\[
\nabla_{b^*_\lambda} \log q(\lambda | a^*_\lambda, b^*_\lambda) = \frac{a^*_\lambda}{b^*_\lambda} - \lambda
\]
\[
\nabla_{a^*_\alpha} \log q(\alpha | a^*_\alpha, b^*_\alpha) = -\log C_1 + \text{dig}(a^*_\alpha + b^*_\alpha)
\]
\[
\nabla_{b^*_\alpha} \log q(\alpha | a^*_\alpha, b^*_\alpha) = -\log C_1 + \text{dig}(a^*_\alpha + b^*_\alpha)
\]

Appendix C Variational family in Bayesian bridge reparameterization method

The details for implementation of the reparameterization method for the proposed semi-parametric Bayesian bridge are described in this section. Only the case with \( J = 1 \) is shown, although the calculations could be easily extended to the multivariate case where \( J \in \mathbb{N} \).

For the Bayesian bridge semi-parametric regression model, we have \( \theta = (\beta^*_0, \beta, \lambda, \phi, \alpha)^T \) as the parameter vector. The proposed marginal variational distributions are \( q(\beta | m^*_\beta, S^*_\beta) = N(\beta | m^*_\beta, S^*_\beta), q(\beta_0 | m^*_\beta_0, S^*_\beta_0) = N(\beta_0 | m^*_\beta_0, S^*_\beta_0), q(\phi | a^*_\phi, b^*_\phi) = \text{Ga}(\phi | a^*_\phi, b^*_\phi), q(\lambda | a^*_\lambda, b^*_\lambda) = \text{Ga}(\lambda | a^*_\lambda, b^*_\lambda), q(\alpha | a^*_\alpha, b^*_\alpha) = C_1 \times \text{Beta}(\alpha | a^*_\alpha, b^*_\alpha). \)

The entropy of multivariate normal distributions and log-normal distributions are available in closed form. If \( q(\theta) = N_d(\theta; \mu, S) \), where \( N_d(\theta; \mu, S) \) denotes the density of a \( d \)-dimensional multivariate normal distribution with mean vector \( \mu \) and covariance matrix \( S \) evaluated at \( \theta \), the entropy of \( q(\theta) \) is \( H(q(\theta); \mu, S) = \frac{1}{2} \log |2\pi eS| + \sum_{i=1}^d \ell_i, \) where \( \ell_i \) denotes the \( i \)-th entry of the diagonal of \( L \), the Cholesky decomposition of \( S \).
For the log-Normal distribution, if \( q(\theta) = \log N(\theta; m, s) \), meaning \( \log \theta \sim N(m, s^2) \), then the corresponding entropy is \( H(\theta; m, s) = \log_2 s \sqrt{2\pi e} \). The entropy for \( \alpha \) can be evaluated approximately via Monte Carlo using the derivatives obtained at the end of Appendix B.

### Appendix D Details on evaluating gradient of the ELBO under ADVI

This section derives in detail the approximation of the gradient of the ELBO in Eq. (10) to facilitate implementation of Algorithm 1.

Let \( \epsilon^{(\ell)} \sim N(0_d, I_d) \), \( \ell = 1, \ldots, M \) where \( d = K_0 + K_1 + \ldots + K_J + 2 + 1 \) is the dimension of the parameter vector \( \theta \). Then we compute \( \xi^{(\ell)} = m + Le^{(\ell)} \) where \( \xi^{(\ell)} = (\xi_{0}^{(\ell)}, \xi_{\beta_1}^{(\ell)}, \ldots, \xi_{\beta_J}^{(\ell)}, \xi_{\phi}^{(\ell)}, \xi_{\lambda_1}^{(\ell)}, \ldots, \xi_{\lambda_J}^{(\ell)}, \xi_{\alpha_1}^{(\ell)}, \ldots, \xi_{\alpha_J}^{(\ell)})^T \), hence \( \xi^{(\ell)} \sim N(m, LL^T) \), and, using the transformation \( T^{-1} \) defined in Eq. (8), we compute the implied Monte Carlo samples in the original parameter vector \( \theta^{(\ell)} = T^{-1}(\xi^{(\ell)}) = (\beta_0^{(\ell)}, \beta_1^{(\ell)}, \ldots, \beta_J^{(\ell)}, \phi^{(\ell)}, \lambda_1^{(\ell)}, \ldots, \lambda_J^{(\ell)}, \alpha_1^{(\ell)}, \ldots, \alpha_J^{(\ell)})^T \) as

\[
\begin{align*}
\beta_0^{(\ell)} &= \xi_{0}^{(\ell)}, \\
\beta_j^{(\ell)} &= \xi_{\beta_j}^{(\ell)}, \quad j = 1, \ldots, J \\
\phi^{(\ell)} &= e^{\xi_{\phi}^{(\ell)}}, \\
\lambda_j^{(\ell)} &= e^{\xi_{\lambda_j}^{(\ell)}}, \quad j = 1, \ldots, J \\
\alpha_j^{(\ell)} &= \frac{C_j}{1 + e^{-\xi_{\alpha_j}^{(\ell)}}}, \quad j = 1, \ldots, J.
\end{align*}
\]

The parameter vectors \( \theta^{(\ell)}, \ell = 1, \ldots, M \) are then used to compute \( \text{ELBO}(\tilde{\theta}, \psi) \).

Recall that ADVI does not require the user to analytically derive the full expression for the gradient \( \tilde{\psi} \text{ELBO}(\tilde{\theta}, \psi) \) in Eq. (10). Instead, the gradient for ADVI is easily calculated via automatic differentiation, which numerically (automatically) computes the gradient using the chain rule

\[
\tilde{\psi} \text{ELBO}(\tilde{\theta}, \psi) = \sum_{\ell=1}^{M} \frac{\partial \text{ELBO}(\tilde{\theta}, \psi)}{\partial \theta^{(\ell)}} \frac{\partial \theta^{(\ell)}}{\partial \xi^{(\ell)}} \frac{\partial \xi^{(\ell)}}{\partial \psi}
\]

(D1)

without the need of analytic derivation of each partial derivative by the practitioner. Many software are available to implement automatic differentiation, such as Pytorch, Tensorflow, Theano, Stan, and others. For example, Kucukelbir et al. (2015) uses Stan in the context of ADVI to implement automatic differentiation.

Automatic differentiation only requires evaluation of \( \text{ELBO}(\tilde{\theta}, \psi) \). Since \( \theta^{(\ell)} = T^{-1}(\xi^{(\ell)}) \) and \( \xi^{(\ell)} = m + Le^{(\ell)} \), the only term in Eq. (D1), that needs further investigation is the first one, which requires to express \( \text{ELBO}(\tilde{\theta}, \psi) \) as a function of \( \theta^{(\ell)} \). Starting from Eq. (10), we get

\[
\text{ELBO}(\tilde{\theta}, \psi) = -\frac{n}{KM} \sum_{\ell=1}^{M} \log p(\tilde{\theta} | \theta^{(\ell)})
\]

\[
+ \frac{1}{M} \sum_{\ell=1}^{M} \left[ \log p(\theta^{(\ell)}) + \log |J^{-1}(\xi^{(\ell)})| \right]
\]

\[
+ \frac{1}{M} \sum_{\ell=1}^{M} \log p(|\lambda^{(\ell)}|, |\phi^{(\ell)}|, |\alpha^{(\ell)}|)
\]

\[
+ \frac{1}{M} \sum_{\ell=1}^{M} \sum_{j=1}^{J} \log p(\lambda_j^{(\ell)}) + \sum_{j=1}^{J} \log p(\alpha_j^{(\ell)})
\]

\[
+ \sum_{i=1}^{d} \sum_{j=1}^{J} \log \lambda_j^{(\ell)} + \log \alpha_j^{(\ell)} + \log \left( C_j - \alpha_j^{(\ell)} \right)
\]

\[
+ \sum_{i=1}^{d} L_{ii} + \text{const}
\]

\[
= \frac{n}{KM} \sum_{\ell=1}^{M} \log N \left( \tilde{\theta}; \sum_{j=0}^{J} \bar{\xi}_j \beta_j^{(\ell)}, \phi^{(\ell)} \right)
\]

\[
+ \frac{1}{M} \sum_{\ell=1}^{M} \log N(\beta_0^{(\ell)}; \mu_0, \Sigma_0)
\]

\[
+ \frac{1}{M} \sum_{\ell=1}^{M} \log Ga(\phi^{(\ell)}; a_{\phi}, b_{\phi})
\]

\[
+ \frac{1}{M} \sum_{\ell=1}^{M} \sum_{j=1}^{K} \log \text{GG} \left( \beta_j^{(\ell)}; 0, \frac{\xi_j^{(\ell)} - \alpha_j^{(\ell)}}{\sqrt{\phi^{(\ell)}}, \alpha_j^{(\ell)}} \right)
\]

\[
+ \frac{1}{M} \sum_{\ell=1}^{M} \sum_{j=1}^{K} \log \text{GG} \left( \lambda_j^{(\ell)}; a_{\lambda}, b_{\lambda} \right)
\]
+ \frac{1}{M} \sum_{\ell=1}^{M} \sum_{j=1}^{J} \log \left[ \alpha_j^{(\ell)} a_{h-1}^{(\ell)} (C_j - \alpha_j^{(\ell)}) b_{h-1}^{(\ell)} \right] \\
+ \frac{1}{M} \sum_{\ell=1}^{M} \log \phi^{(\ell)} \\
+ \frac{1}{M} \sum_{\ell=1}^{M} \sum_{j=1}^{J} \left[ \log \lambda_j^{(\ell)} + \log \alpha_j^{(\ell)} + \log (C_j - \alpha_j^{(\ell)}) \right] \\
+ \sum_{i=1}^{d} L_{ai} + \text{const}, \tag{D2}

where the terms const represent additive constants that do not depend on the variational parameters $\psi = (m, l)$ and the term $\sum_{i=1}^{d} L_{ai}$ comes from the fact that the entropy of a multivariate Gaussian random vector $x \sim N(\mu, \Sigma)$ is $\mathbb{H}[x] = \mathbb{E}_{x \sim N(\mu, \Sigma)} [-\log N(x|\mu, \Sigma)] = \log (2\pi e \Sigma)$.

Notice that the gradient $\nabla_{\psi} ELBO(\tilde{y}, \psi)$ will be calculated with respect to the column vector $m$ and the lower triangular matrix $L$, since $\psi = (m, L)$. In the algorithm, first, $m$ and $L$ are initialized. Then, at each iteration, the automatic differentiation software computes the derivatives with respect to each entry of $m$ and the non-zero entries of $L$, updates each of them, and rebuilds $m$ and $L$.

### Appendix E Further results and comparisons

This section presents more results on the simulation described in Sect. 5.1.

Figure 15 contains the posterior marginal distributions for all parameters of the model described in Sect. 5.1. The marginals for $\beta_1, \ldots, \beta_3$ are also shown in Fig. 4. As in Fig. 4, Fig. 15 shows high level of agreement between marginals obtained via MCMC and ADVI, with brms having higher discrepancy with respect to the MCMC. INLA also approximates well the marginal distributions except for a small number of coefficients (e.g., $\beta_1, \beta_2, \beta_3$).

Figure 16 complements Figure 8 from Sect. 5.1 with the point estimates for all parameters obtained via ADVI and MCMC for each of the 100 simulated datasets.

To further address the quality of the variational approximations to the true posterior distribution, Fig. 17 shows the p-values obtained when comparing (via Kolmogorov-Smirnov tests) marginal posterior predictive distributions estimated by VB and MCMC. The shapes of the histograms are approximately uniform with a low prevalence of small p-values, as expected under the null hypothesis that the distributions under MCMC and VB are the same. For example, the empirical proportions of p-values below 0.05 are close to 0.05, which corresponds to the expected proportion of false discoveries when $H_0$ is true and a type 1 error of 0.05 is fixed.

Regarding simulation scenario 3 described in Sect. 5.3, Figure 18 shows that the resulting MCMC chain (for $\lambda_1$ specifically) is highly correlated and justifies the spacing of 200 by 200 iterations to get an approximately independent approximation to the posterior. Marginals for other parameters also exhibit similarly strong dependence.

### References

Alves, L., Dias, R., Migon, H.S.: Variational full bayes lasso: Knots selection in regression splines (2021). arXiv preprint arXiv:2102.13548

Armagan, A.: Variational bridge regression. In: Artificial Intelligence and Statistics, pp. 17–24 (2009)

Baydin, A.G., Pearlmutter, B.A., Radul, A.A., Siskind, J.M.: Automatic differentiation in machine learning: a survey. J. Mach. Learn. Res. 18, 1–43 (2018)

Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: a review for statisticians. J. Am. Stat. Assoc. 112(518), 859–877 (2017)

Bürkner, P.C.: brms: An R package for Bayesian multilevel models using Stan. J. Stat. Softw. 80(1), 1–28 (2017)

Carvalho, C.M., Polson, N.G., Scott, J.G.: Handling sparsity via the horseshoe. In: Artificial Intelligence and Statistics, pp. 73–80 (2009)

Casella, G., Ghosh, M., Gill, J., Kyung, M.: Penalized regression, standard errors, and Bayesian lassos. Bayesian Anal. 5(2), 369–411 (2010). https://doi.org/10.1214/10-BA607

Currie, I.D., Durban, M.: Flexible smoothing with p-splines: a unified approach. Stat. Model. 2(4), 333–349 (2002)

Damien, P., Walker, S.G.: Sampling truncated normal, beta, and gamma densities. J. Comput. Graph. Stat. 10(2), 206–215 (2001)

Denison, D., Mallick, B., Smith, A.: Automatic Bayesian curve fitting. J. R. Stat. Soc. Ser. B (Stat. Methodol.) 60(2), 333–350 (1998)

Dias, R., Gnermaner, D.: A Bayesian approach to hybrid splines non-parametric regression. J. Stat. Comput. Simul. 72(4), 85–297 (2002)

Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. 12(7), 2121–2159 (2011)

Eilers, P.H., Marx, B.D.: Flexible smoothing with b-splines and penalties. Stat. Sci. 11(2), 89–121 (1996)

Frank, L.E., Friedman, J.H.: A statistical view of some chemometrics regression tools. Technometrics 35(2), 120–135 (1993)

Hastie, T., Tibshirani, R.: Generalized additive models. Stat. Sci. 1(3), 297–310 (1986)

Hastie, T., Tibshirani, R.: Bayesian backfitting. Stat. Sci. 15(3), 196–213 (2000)

Hastie, T., Tibshirani, R., Wainwright, M.: Statistical learning with sparsity. Monogr. Stat. Appl. Probab. 143, 143 (2015)

Hoerl, A.E., Kennard, R.W.: Ridge regression: applications to nonorthogonal problems. Technometrics 12(1), 69–82 (1970)

Hoerl, A.E., Kennard, R.W.: Ridge regression: biased estimation for nonorthogonal problems. Technometrics 12(1), 55–67 (1970)

Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization (2014). arXiv preprint arXiv:1412.6980

Kingma, D.P., Welling, M.: Auto-encoding variational Bayes (2013). arXiv preprint arXiv:1312.6114

Kucukelbir, A., Ranganath, R., Gelman, A., Blei, D.: Automatic variational inference in Stan. In: Advances in neural information processing systems 28 (2015)
Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., Blei, D.M.: Automatic differentiation variational inference. J. Mach. Learn. Res. 18(1), 430–474 (2017)

Lang, S., Brezger, A.: Bayesian p-splines. J. Comput. Graph. Stat. 13(1), 183–212 (2004)

Leng, C., Tran, M.N., Nott, D.: Bayesian adaptive lasso. Ann. Inst. Stat. Math. 66, 221–244 (2014)

Li, F., Villani, M.: Efficient Bayesian multivariate surface regression. Scand. J. Stat. 40(4), 706–723 (2013)

Li, Q., Lin, N.: The Bayesian elastic net. Bayesian Anal. 5(1), 151–170 (2010)

Luts, J., Wand, M.P.: Variational inference for count response semiparametric regression. Bayesian Anal. 10(4), 991–1023 (2015)

Mallick, H., Yi, N.: Bayesian bridge regression. J. Appl. Stat. 45(6), 988–1008 (2018)

Menictas, M., Wand, M.P.: Variational inference for heteroscedastic semiparametric regression. Aust. N. Z. J. Stat. 57(1), 119–138 (2015)

Ong, V.M.H., Mensah, D.K., Nott, D.J., Jo, S., Park, B., Choi, T.: A variational Bayes approach to a semiparametric regression using Gaussian process priors. Electron. J. Stat. 11(2), 4258–4296 (2017)

Paisley, J.W., Blei, D.M., Jordan, M.I.: Variational Bayesian inference with stochastic search. In: Proceedings of the 29th International Conference on Machine Learning (icml-12), pp. 1367–1374 (2012)

Park, T., Casella, G.: The Bayesian lasso. J. Am. Stat. Assoc. 103(482), 681–686 (2008)

Polson, N.G., Scott, J.G., Windle, J.: The Bayesian bridge. J. R. Stat. Soc. Ser. B (Stat. Methodol.) 76(4), 713–733 (2014)

Ranganath, R., Gerrish, S., Blei, D.M.: Black box variational inference. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics (2014)

Rue, H., Martino, S., Chopin, N.: Approximate Bayesian inference for latent gaussian models by using integrated nested Laplace approximations. J. R. Stat. Soc. Ser. B (Stat. Methodol.) 71(2), 319–392 (2009)

Silverman, B.W., Green, P.J.: Nonparametric Regression and Generalized Linear Models. Chapman and Hall, London (1994)

Subbotin, M.T.: On the law of frequency of error. Sbornik: Math. 31(2), 296–301 (1923)

Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Soc.: Ser. B (Methodol.) 58(1), 267–288 (1996)

Tibshirani, R.: The lasso method for variable selection in the cox model. Stat. Med. 16(4), 385–395 (1997)

Wand, M.P.: Fast approximate inference for arbitrarily large semiparametric regression models via message passing. J. Am. Stat. Assoc. 112(517), 137–168 (2017)

West, M., Harrison, J.: Bayesian Forecasting and Dynamic Models. Springer Series in Statistics, Springer-Verlag, New York (1997)

Wood, S.: mgcv: Mixed gam computation vehicle with GCV/AIC/REML smoothness estimation. R package version 1.8-1 (2012)

Wood, S.N.: Generalized Additive Models: An Introduction with R, 2nd edn. CRC Press, Boca Raton (2017)

Yin, M., Zhou, M.: Semi-implicit variational inference. In: International conference on machine learning, pp. 5660–5669 (2018)

Zeiler, M.D.: Adadelta: an adaptive learning rate method (2012). arXiv preprint arXiv:1212.5701

Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. J. R. Stat. Soc. Ser. B (Stat. Methodol.) 67(2), 301–320 (2005)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.