Erasmus University Rotterdam
Erasmus School of Economics
Master Thesis Econometrics and Management Science - Econometrics

Time Varying Bayesian Additive Regression Trees

A.G. (Daniëlle) Lam (575651)

Supervisor:	Dr. E.P. (Eoghan) O'Neill
Second assessor:	A. (Aishameriane) Venes Schmidt
Date final version:	June 22, 2024

Time Varying Bayesian Additive Regression Trees

Daniëlle Lam[a]

Abstract

Driven by the widespread evidence of parameter instability in macroeconomic forecasting models, numerous time-varying parameter models have been proposed. This paper proposes an extension to Bayesian additive regression trees by specifying a time-varying parameter (TVP-BART) in every terminal node of the tree ensemble. The TVP-BART attempts to capture potential changes in the economy's underlying structure in a flexible way. To address the potential absence of time observations in terminal nodes, SoftBART is similarly extended to contain time-varying terminal node parameters. Simulation exercises demonstrate that the addition of time variation is especially beneficial when time variation includes both a random walk and structural break. Additionally, employing US macroeconomic data to forecast inflation serves as an empirical application of the proposed methodology.

Keywords: Bayesian additive regression trees; Time-varying parameters; Inflation forecasting

[a] Erasmus School of Economics Master student. Contact: 575651al@eur.nl

Contents

1 Introduction

Ensembles of decision trees are a powerful tool for getting flexible estimates of regression functions. Methods like gradient-boosted decision trees, random forests, and Bayesian regression trees exemplify this approach. In particular, Bayesian tree-based models, such as the Bayesian additive regression trees (BART) model (Chipman et al. 2010), are of special interest due to their computational efficiency, scalability to large datasets, minimal tuning requirements, and natural uncertainty quantification. Each terminal node of a regression tree in BART produces a scalar parameter and the scalars from all trees are summed to create a model forecast. However, there has been a growing awareness of the empirical need to allow for parameter change. This study extends BART to incorporate time-varying parameters in terminal nodes (TVP-BART), employing the efficient method proposed by Hauzenberger et al. (2022b). Additionally, to avoid estimation problems in the terminal node parameters due to the deterministic allocation used in BART, SoftBART is similarly extended to include time-varying terminal node parameters (TVP-SoftBART). In SoftBART all observations are included in each terminal node albeit with weights that are specific to both the observations and nodes (Linero & Yang 2018).

In macroeconomics and finance, models commonly used for forecasting are fully parametric, of which the vector autoregressive (VAR) model is one of the most prominent examples (Sims 1980). In such traditional econometric models one generally assumes that the first and second moments of the target variables are stable representing the assumption of model stability. However, macroeconomic relations seldom satisfy model stability, and models that allow these moments to vary over time have been shown to improve macroeconomic forecasting. This relaxation does not only result in more accurate point forecasts but also improves density forecasts (Pettenuzzo & Timmermann 2017). Although numerous studies suggest parameters change over time, the optimal method to integrate this instability into model specifications is difficult to find. The evolution of parameters can take many forms. For example one assumes that they vary with the behaviour of observable economic variables, or the way it changes is assumed to be unobservable. In addition, the development can either be discrete and abrupt or continuous and smooth. Examples of model specifications that allow parameters to vary include Markov switching models, threshold and smooth transition models. Since Primiceri (2005) introduced time variation and stochastic volatility into the VAR model, various extensions have appeared, improving both univariate and multivariate models in the way they capture time-varying dynamics (Groen et al. 2013, Koop & Korobilis 2013, Belmonte et al. 2014, Huber et al. 2021). In this literature, parameters are assumed to develop according to a random walk.

Another stream of literature proposes Bayesian non-parametric time series models to relax the assumption of linearity. During normal periods, when macroeconomic relations remain stable, linearity may adequately fit the data. However, during turbulent periods, important changes often occur in key relations, making models that assume linearity too rigid. Non-parametric models, such as BART, allow for greater flexibility by making minimal assumptions about the functional form. A main advantage is that uncertainty about the functional form and the parameters is included in the posterior predictive distribution. Recently, non-parametric VARs have been proposed where the parameters are modelled using BART, the BAVART model (Huber & Rossini 2022, Huber et al. 2023). Additionally, Hauzenberger et al. (2022a) propose a non-parametric time-varying parameter VAR (TVP-VAR) model where BART is utilised to model the coefficients. This approach accommodates time-varying parameters, although not within the regression tree functions.

Furthermore, incorporating time-varying parameters into a model specification can worsen the problem of overparametrization. This change to the model specification drastically increases the number of parameters that need to be estimated. This can lead to the issue of overfitting, where the model captures noise in the data rather than the exact relation between the inputs and the target. In this case, the model performs well on training data but poorly on new testing data caused by their excessive flexibil-

ity. To mitigate this problem in a TVP-VAR, D'Agostino et al. (2013) for example uses only a limited information set. Another solution is the use of Bayesian methods since prior information can be essential to prevent the curse of dimensionality. Although computationally intensive, one often utilizes Markov Chain Monte Carlo (MCMC) techniques, such as a Gibbs sampler. For instance, Hauzenberger et al. (2022b) introduces a fast and flexible framework to estimate time-varying parameter (TVP) regressions. This framework allows for a flexible patterns of time variation in the parameters rather than restricting their evolution only to a random walk or autoregressive process. In this approach, it is essential to write a regression model with time-varying parameters and exogenous variables as a high-dimensional static regression problem. Additionally, to speed up the computation a singular-value decomposition is used and in combination with a conditionally conjugate priors, this results in a fast and scalable algorithm. Furthermore, unlike most computationally efficient algorithms for TVP regressions, this approach distinguishes itself by avoiding any approximations.

Most studies studying the inclusion of time-varying parameters into a model specification focus on the empirical application of forecasting US inflation. While traditionally linear and parametric models are used, a recent study by Clark et al. (2023) displays the ability of non-parametric BART-based VARs to more accurately forecast various quarterly US macroeconomic indicators, including inflation. Instead of evaluating only point forecasts, the uncertainty surrounding these forecasts is also included in the evaluation by comparing density and tail-risk forecasts. This is mainly motivated by two recent events, the financial crisis and the COVID-19 pandemic. Based on tail forecasts, they find that flexible models improve upon VAR models with stochastic volatility. In addition, they show that when the mean is modelled using BART, including a heteroskedastic error specification only leads to a marginal improvement in forecasting accuracy. Additionally, Clark et al. (2024) uses Bayesian techniques and BART to forecast quarterly US inflation. These non-parametric models excel in both point and density forecasts, particularly during volatile periods such as the pandemic.

The strong empirical performance of BART for forecasting and inference gives rise to the main contribution of this paper. This paper aims to bridge the literature on BART with the literature on TVP models. In particular, a non-parametric BART is proposed that has several key features that are important in macroeconomic and financial forecasting. First, the existing BART literature is extended by adjusting the methodology to time series data whereas the current literature focuses on cross-sectional data. Second, the leaf node parameters are allowed to be time-varying. To avoid under-identification of the time-varying parameters, SoftBART is similarly extended since it includes all time observations in each terminal node albeit with weights. The estimation of time-varying parameters in each leaf node is performed by recasting the time-varying parameter model as a static regression. The TVP-BART and TVP-SoftBART have some similarities to MOTR-BART introduced by Prado et al. (2021a) where each leaf node is modelled by a linear part. However, the covariates that are used in the linear regression in each of the leaf nodes in TVP-BART and TVP-SoftBART are different to MOTR-BART, which uses only the split variables. Motivated by the increased accuracy of the addition of TVP in VAR models, it is similarly expected that the new proposed BART methodology generates better forecasting performance than its time-invariant counterpart.

To illustrate the use of the models, the macroeconomic application of forecasting quarterly US inflation is revisited. In this application, US inflation is forecasted using a restricted set of macroeconomic variables from the FRED-QD database that have previously been found to be important indicators of inflation. Quarterly data is frequently used in studying the time-varying behaviour of US inflation, see for example Groen et al. (2013) and Clark et al. (2024). In addition, using the monthly macroeconomic database would result in a substantially larger number of observations, making estimation time considerably larger and hence infeasible. This setup is similar to Clark et al. (2024), and similarly, the forecasts are evaluated using point, density and tail risk forecasts. In addition, the proposed BART methods are compared during the COVID-19 pandemic. Results show that the BART-based models that allow for time variation in the

conditional mean result in accurate inflation point and density forecasts for the medium-term horizon, namely four-quarters-ahead. In particular, TVP-SoftBART excels in both point forecast as well as in density forecast accuracy. In addition, the performance of the models is compared using synthetic data. In this simulation, several DGPs are employed ranging from a simple non-linear model without time variation towards more complex DGPs with time-varying parameters. The time-varying BART models perform the best when time variation includes both random walk variation as well as a structural break. Especially TVP-BART excels in forecasting accuracy. In addition, it obtains the best in-sample fits, in both the simulation and empirical example, while it fails in some cases to maintain this out-of-sample. Therefore this model may be prone to overfitting. It should also be noted that adding time variation to BART is costly in terms of computation time. Therefore it may be questionable whether the additional accuracy these models offer when time variation is more pronounced weighs up against the considerable additional computation time.

The remainder of this paper is structured as follows. Section 2 discusses related literature. Section 3 introduces the econometric framework. This Section includes a discussion of BART, MOTR-BART, TVP-BART, SoftBART, Soft MOTR-BART and TVP-SoftBART, and a forecast evaluation methodology. Section 4 includes the simulation framework and results. Next, Section 5 contains the description of the empirical application of forecasting US inflation, as well as a discussion of the results. Section 6 concludes.

2 Background

Ensemble methods are popular and flexible methods part of Machine Learning that combine a set of trees each in a different way. These include boosting (Friedman 2001), bagging (Breiman 1996), and random forests (Breiman 2001), with a comprehensive overview available in Masini et al. (2023). Boosting uses a series of individual trees, by iteratively fitting them such that each catches some part of the variation that is not captured by the remaining trees. This is in contrast to bagging and random forests. These techniques generate multiple independent trees and try to reduce the variance by averaging the predictions across the trees. Bayesian Additive Regression Trees (BART) also employs a sum of trees approach, but in contrast to boosting a regularization prior is used to keep each tree small. BART has some similarities to boosting since BART also employs an iterative procedure to fit successive residuals. However, the variable to be estimated is approximated by summing the response estimate of each single tree. This is similar to bagging where the trees are averaged to obtain an estimate for the target.

Several extensions of BART exist to address its limitations. For instance, SoftBART, introduced by Linero & Yang (2018), generates an estimated function that is smoother and can handle sparsity better than the original BART model. In SoftBART observations are assigned to terminal nodes with a certain probability instead of deterministically as in BART. Another extension is MOTR-BART, proposed by Prado et al. (2021a), which uses a linear predictor instead of a scalar to make predictions in each terminal node. In addition, varying coefficient BART (VC-BART) is introduced by Deshpande et al. (2020) where the functional form is modelled to be linear but where each coefficient is estimated by BART using a set of effect modifiers as input variables. Using a linear functional form allows for easy interpretation and in this way, each variable effect is modelled using a separate BART model. Another work that is related to VC-BART is a combined semi-parametric BART (Prado et al. 2021b), where the response is approximated by a linear predictor and a BART model, where the main effect is estimated using the linear part and any remaining unspecified interactions and non-linearities are captured by BART. This work forms an extension to the semi-parametric BART proposed by Zeldow et al. (2019) where the covariates used as inputs for the linear model and BART are mutually exclusive.

Inspired by the forecasting advancements in VAR models reached through the inclusion of time-varying parameters, several non-parametric models have been adjusted to include time-variation. For instance, Goulet Coulombe (2020) introduces the Macroeconomic Random Forest (MRF) by modifying

the Local Linear Forest from Friedberg et al. (2020), which uses a linear predictor instead of a scalar in its terminal nodes. Drawing inspiration from time-varying parameter (TVP) methods, the covariates are transformed and weighted according to the time structure. A drawback of the MRF is its use of the same functional form, a random forest, to estimate the coefficients for all variables. Another example is the nonparametric TVP-VAR model proposed by Hauzenberger et al. (2022b), where each TVP is modelled using BART. This work extends the non-parametric VARs and mixed frequency VARS developed by Huber & Rossini (2022) and Huber et al. (2023), respectively, which also utilize BART and demonstrate strong forecasting performance. However, the BART model employed does not itself include time-varying parameters. The linear combination approach in this model is similar to VC-BART by Deshpande et al. (2020).

Many applications of TVP regressions specifically focus on inflation forecasting. This focus is due to the well-documented time-varying nature of inflation and the importance of accurate inflation forecasts for policymakers, economic agents and academic researchers. Traditionally, VARs and factor models have been considered time-invariant. However, substantial evidence in inflation forecasting indicates time dependency in both the conditional mean and variance. Although there is little evidence of sudden shifts in inflation, models that permit gradual changes tend to perform better (Groen et al. 2013, Pettenuzzo & Timmermann 2017). In these forecasting exercises, the random walk introduced by Atkeson et al. (2001) is often used as simple benchmark since it is generally recognized to be a difficult benchmark to beat in out-of-sample forecasting.

3 Methods

In this Section, the BART framework is discussed together with the extensions to allow for time-varying parameters. BART uses a sum of regression trees where each regression tree is a stepwise function. When the regression trees are summed together, the piecewise constant functions are summed and a more complex stepwise function is created which can approximate non-linear functions. The idea of BART is that many trees together can explain a more complex function by each explaining only a small part of the response. Therefore, each tree is considered a weak learner. However, this piecewise constant function that is learned by BART is not smooth; thus, SoftBART was introduced by Linero & Yang (2018). In addition, each piece of function that is estimated is time-invariant, which in this Section is extended to include a time-varying function which in a way also allows for a smoother estimated function. This Section also discusses how estimating time-varying parameters in each terminal node of a tree can be seen as a static linear regression of a large dimension. This Section concludes with a discussion of the forecast evaluation methodology.

3.1 BART

Bayesian Additive Regression Trees (BART) introduced by Chipman et al. (2010) is a non-parametric Bayesian algorithm that produces a group of trees by selecting the covariates and split points at random. Each tree can be changed using four moves: growing, pruning, changing or swapping, and is compared to the previous version via a Metropolis-Hastings step on the part of the response variable that is not explained by the remaining trees.

BART considers a univariate response variable that is approximated by a sum of predicted values from a set of trees as

$$\begin{aligned} y_t &= f(\boldsymbol{x}_t) + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \sigma^2) \\ f(\boldsymbol{x}_t) &= \sum_{j=1}^{m} g(\boldsymbol{x}_t; \mathcal{T}_j, \mathcal{M}_j), \end{aligned} \tag{1}$$

where y_t denotes the response variable, $\boldsymbol{x}_t = (x_{t1}, ..., x_{tk})$ represents the t-th row of the design matrix X

with k covariates, m the number of trees, and $\mathcal{T}_j$ contains the set of splitting rules that defines tree j and $\mathcal{M}_j = \{\mu_{j1}, \dots \mu_{jb_j}\}$ the set of predicted values with b_j denoting the number of terminal nodes of tree j. Each prediction is given by

$$g(\boldsymbol{x}_t; \mathcal{T}_j, \mathcal{M}_j) = \sum_{\ell=1}^{b_j} \mathbb{1}\{\boldsymbol{x}_t \in \mathcal{P}_{j\ell}\}\mu_{j\ell},$$

where the splitting rules that define the terminal nodes for tree j result in partitions of the observations, such that $\mathcal{P}_{j\ell}$ denotes the partition of observations that reach terminal node ℓ of tree j.

To obtain posterior draws of the trees, associated terminal node parameters and error variance, a prior distribution needs to be specified. The joint prior distribution of these quantities, $P(\mathcal{T}_1, \mathcal{M}_1, \dots, \mathcal{T}_m, \mathcal{M}_m, \sigma)$, can easily be split into smaller parts using the fact that $\{\mathcal{T}_1, \mathcal{M}_1, \dots, \mathcal{T}_m, \mathcal{M}_m\}$ and σ^2 are independent and that $\mathcal{T}_1, \mathcal{M}_1, \dots, \mathcal{T}_m, \mathcal{M}_m$ are independent of each other. The joint prior specification can be written as

$$\begin{aligned}
P(\mathcal{T}_1, \mathcal{M}_1, \dots, \mathcal{T}_m, \mathcal{M}_m, \sigma) &= P(\mathcal{T}_1, \mathcal{M}_1, \dots, \mathcal{T}_m, \mathcal{M}_m)P(\sigma) \\
&= \left[\prod_{j=1}^{m} P(\mathcal{T}_j, \mathcal{M}_j)\right] P(\sigma) \\
&= \left[\prod_{j=1}^{m} P(\mathcal{M}_j|\mathcal{T}_j)P(\mathcal{T}_j)\right] P(\sigma) \\
&= \left[\prod_{j=1}^{m} \left\{\prod_{\ell=1}^{b_j} P(\mu_{j\ell}|\mathcal{T}_j)\right\} P(\mathcal{T}_j)\right] P(\sigma).
\end{aligned}$$

Therefore, the prior of the BART model consists of three components (1) the tree structure itself ($P(\mathcal{T}_j)$) (2) the terminal node parameters given the tree structure ($P(\mu_{j\ell}|\mathcal{T}_j)$) and (3) the error variance σ^2 which is independent of the tree structure and the terminal node parameters.

To manage the depth of each tree, the prior on the tree is given by

$$p(\mathcal{T}_j) = \prod_{\ell \in \mathcal{L}_1} [\alpha(1 + d_{j\ell})^{-\beta}] \times \prod_{\ell \in \mathcal{L}_j} [1 - \alpha(1 + d_{j\ell})^{-\beta}]$$

where $\mathcal{L}_1$ and $\mathcal{L}_j$ represent the sets of indices in the internal and terminal nodes respectively, $d_{j\ell}$ is the depth of node ℓ in tree j, $\alpha \in (0, 1)$ and $\beta \geq 0$. α and β are the prior parameters on the tree structure, such that $\alpha(1 + d_{j\ell})^{-\beta}$ is the probability of node ℓ being at internal depth $d_{j\ell}$. Chipman et al. (2010) recommend $\alpha = 0.95$ and $\beta = 2$. The uniform distribution is the default distribution to select the covariate to split upon in an internal node. Once the covariate is selected, the split value is selected using again the uniform distribution.

Given a tree with a set of terminal nodes, each terminal node has a parameter representing the estimate of the response in this partition of the predictor space. This parameter is the fitted value assigned to any observation in this node. The prior on each of the leaf parameters $\mu_{j\ell}$ is given by

$$\mu_{j\ell}|\mathcal{T}_j \sim \mathcal{N}(0, \sigma_\mu^2),$$

σ_μ^2 is the prior variance of the leaf node parameters, where Chipman et al. (2010) recommend $\sigma_\mu = 0.5/k\sqrt{m}$ with $k = 2$ after scaling $\boldsymbol{y}$. However, instead of estimating only a scalar in each terminal node, the parameters in the terminal nodes are changed to be time-dependent which is discussed in Section 3.3.

The final prior is on the error variance σ^2 and is chosen to be

$$\sigma^2 \sim IG(\nu/2, \nu\lambda/2)$$

where $IG(\alpha, \beta)$ is the inverse gamma distribution with shape parameter α and rate parameter β.

Posterior Simulation

A Gibbs sampler is employed to generate draws from the posterior distribution of

$$p((\mathcal{T}_1, \mathcal{M}_1), ..., (\mathcal{T}_m, \mathcal{M}_m), \sigma^2 | \boldsymbol{y}, X).$$

The Gibbs sampler employed in BART uses a form of Bayesian backfitting. Each tree is fit iteratively while holding all other trees constant and fitting only the part of the response that is not explained by the remaining trees. Denote the set of all trees except the j-th tree by $\mathcal{T}_{(j)}$ and similarly let $\mathcal{M}_{(j)}$ denote the set of terminal node parameters of all tree except the j-th tree. Then, $\mathcal{T}_{(j)}$ will be a set of $m-1$ trees. The Gibbs sampler consists of m successive draws of $(\mathcal{T}_j, \mathcal{M}_j)$ conditionally on $(\mathcal{T}_{(j)}, \mathcal{M}_{(j)}, \sigma, \boldsymbol{y}, X)$. The conditional distribution $p(\mathcal{T}_j, \mathcal{M}_j | \mathcal{T}_{(j)}, \mathcal{M}_{(j)}, \sigma, \boldsymbol{y}, X)$ depends on $(\mathcal{T}_{(j)}, \mathcal{M}_{(j)}, \sigma, \boldsymbol{y}, X)$ through the partial residuals

$$\boldsymbol{r}^{(j)} \equiv \boldsymbol{y} - \sum_{k \neq j} g(X; \mathcal{T}_k, \mathcal{M}_k)$$

Thus the m draws of $(\mathcal{T}_j, \mathcal{M}_j)$ given $(\mathcal{T}_{(j)}, \mathcal{M}_{(j)}, \sigma, \boldsymbol{y}, X)$ are equivalent to m draws from $(\mathcal{T}_j, \mathcal{M}_j | \boldsymbol{r}^{(j)}, \sigma^2, X)$. This is equivalent to the posterior of a single tree model $r_t^{(j)} = g(\boldsymbol{x}_t; \mathcal{T}_j, \mathcal{M}_j) + \varepsilon_t$, where the residual response forms the target. Next, the Gibbs sampler uses a Metropolis-Hastings (MH) step to either accept or reject a proposed change to the first tree's structure. If the proposed tree is accepted, the terminal node parameters are updated and the Gibbs sampler moves on to the next tree. A tree can be changed by small modifications to its structure: growing a terminal node by adding two child nodes (Grow), pruning two child nodes (Prune), or changing a split rule (Change).[1] Below is the MH ratio where the parameter sampled is the tree, and the data is the responses unexplained by the remaining trees, $\boldsymbol{r}^{(j)}$. The new, proposed tree is denoted with an asterisk and the original tree is without the asterisk. For further elaboration see Kapelner & Bleich (2013).

$$\alpha(\mathcal{T}_j, \mathcal{T}_j^*) = \min \left\{ 1, \frac{p(\boldsymbol{r}^{(j)} | \mathcal{T}_j^*, \sigma^2) p(\mathcal{T}_j^*) p(\mathcal{T}_j^* \to \mathcal{T}_j)}{p(\boldsymbol{r}^{(j)} | \mathcal{T}_j, \sigma^2) p(\mathcal{T}_j) p(\mathcal{T}_j \to \mathcal{T}_j^*)} \right\}, \tag{2}$$

where $\frac{p(\boldsymbol{r}^{(j)} | \mathcal{T}_j^*, \sigma^2)}{p(\boldsymbol{r}^{(j)} | \mathcal{T}_j, \sigma^2)}$ denote the likelihood ratio where the tree structure of the original tree and proposed tree determine which responses fall into which of the terminal nodes such that the associated response in each of the node can be calculated. Note that the terminal nodes solely determine the likelihoods. $\frac{p(\mathcal{T}_j^* \to \mathcal{T}_j)}{p(\mathcal{T}_j \to \mathcal{T}_j^*)}$ is the transition ratio and $\frac{p(\mathcal{T}_j^*)}{p(\mathcal{T}_j)}$ is ratio of the tree structures. The newly proposed tree is accepted if a draw from the standard uniform distribution is less than or equal to the value of $\alpha(\mathcal{T}_j, \mathcal{T}_j^*)$, otherwise the original tree $\mathcal{T}_j$ is kept.

After drawing a new tree, $\mathcal{T}_j$, the associated terminal parameters need to be drawn which are stored in $\mathcal{M}_j$. These terminal node parameters are necessary for the subsequent residual $\boldsymbol{r}^{(j+1)}$. Within a given terminal node, since both the prior and the likelihood are normally distributed, the posterior of each of the leaf parameters is conjugate normal with the mean being a weighted combination. The posterior

[1] Chipman et al. (2010) also considered the perturbation Swap in the original BART formulation. Kapelner & Bleich (2013) excludes this modification to the tree structure due to the complexity of this change.

distribution of the time-invariant terminal node parameters is given by

$$\mu_{j\ell}|\mathcal{T}_j, r^{(j)}, \sigma^2 \sim \mathcal{N}\left(\frac{\sigma^{-2}\sum_{t\in\mathcal{P}_{j\ell}} r_t^{(j)}}{T_{j\ell}/\sigma^2 + \sigma_\mu^{-2}}, \frac{1}{T_{j\ell}/\sigma^2 + \sigma_\mu^{-2}}\right),$$

with $r_t^{(j)}$ the t-th partial residual of tree j and $T_{j\ell}$ the number of observations in leaf node ℓ of tree j.

When all m trees and terminal node parameters are generated, the posterior variance can be drawn from the following full conditional distribution

$$\sigma^2|\mathcal{T}_1, \mathcal{M}_1, ..., \mathcal{T}_m, \mathcal{M}_m, X, \boldsymbol{y} \sim IG\left(\frac{T+\nu}{2}, \frac{S+\nu\lambda}{2}\right), \quad S = \sum_{t=1}^{T}(y_t - \hat{y}_t)^2,$$

where T denotes the total number of observations, and $\hat{y}_t$ the fitted response, with $\hat{y}_t = \sum_{j=1}^{m} g(X_t; \mathcal{T}_j, \mathcal{M}_j)$.

3.2 MOTR-BART

The first BART extension discussed is Model Trees Bayesian Additive Regression Trees (MOTR-BART) introduced by Prado et al. (2021a). The main difference compared to BART is that in each terminal node, a linear regression is estimated instead of only estimating a scalar. The covariates in the linear predictor are the variables that are used as splitting variables in creating the tree. This approach forms the basis for TVP-BART and therefore is discussed first.

The MOTR-BART specification is given by

$$\begin{aligned} y_t &= \sum_{j=1}^{m} g(\boldsymbol{x}_t; \mathcal{T}_j, \mathcal{B}_j) + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \sigma^2) \\ g(\boldsymbol{x}_t; \mathcal{T}_j, \mathcal{B}_j) &= \sum_{\ell=1}^{b_j} \mathbb{1}\{\boldsymbol{x}_t \in \mathcal{P}_{j\ell}\}\boldsymbol{x}_{tj\ell}\boldsymbol{\beta}_{j\ell}, \end{aligned} \tag{3}$$

where $\mathcal{B}_j$ is the set of parameters of all linear predictors of tree j and $\boldsymbol{x}_{tj\ell}$ is a vector of splitting variables observed at time t that lead to terminal node ℓ of tree j. The prior on the tree and error variance is the same as for BART, only the prior for $\boldsymbol{\beta}_{j\ell}$ is different. The prior is given by

$$\boldsymbol{\beta}_{j\ell}|\mathcal{T}_j \sim \mathcal{N}_{q_{j\ell}}(0, \sigma^2 V), \quad V = \sigma_\beta^2 I_{q_{j\ell}}$$

where $I_{q_{j\ell}}$ denotes the identity matrix of dimension $q_{j\ell}$, which represents the number of regressor variables of terminal node ℓ of tree j plus one for the intercept. Similarly to what Prado et al. (2021a) suggests, the intercept and slopes have a different prior distribution. The conjugate priors are $\sigma_{\beta_0}^2 \sim IG(a_0, b_0)$ and $\sigma_\beta^2 \sim IG(a_1, b_1)$ respectively.

Posterior Simulation

Similar to BART, a Gibbs sampler generates draws from the posterior distributions. The conditional distribution of tree j depends on the other $m-1$ trees through the partial residuals

$$r_t^{(j)} = y_t - \sum_{k\neq j} g(\boldsymbol{x}_t; \mathcal{T}_k, \mathcal{B}_k) \tag{4}$$

$$= \sum_{\ell=1}^{b_j} \mathbb{1}\{\boldsymbol{x}_t \in \mathcal{P}_{j\ell}\}\boldsymbol{x}_{tj\ell}\boldsymbol{\beta}_{j\ell} + \varepsilon_t. \tag{5}$$

These partial residuals are necessary to calculate the acceptance probability in the MH step to accept or reject a newly proposed tree, see Equation (2). It should be noted that in the MOTR-BART specification of Prado et al. (2021a) tree transition probabilities ($\frac{p(\mathcal{T}_j^* \to \mathcal{T}_j)}{p(\mathcal{T}_j \to \mathcal{T}_j^*)}$ in Equation (2)) were originally not included in the MH-step of the trees. However, the MOTR-BART specification used in this paper includes transition probabilities of each tree in the MH-step similar to BART and therefore is slightly different than the original specification as in Prado et al. (2021a). Finally, the marginal likelihood of the partial residuals, which is necessary to calculate the acceptance probability in the MH step, is given by

$$p(\boldsymbol{r}^{(j)}|X, \sigma^2, \mathcal{T}_j) = (\sigma^2)^{-(T/2)} \prod_{\ell=1}^{b_j} \left[|V|^{-1/2} |\Lambda_{j\ell}|^{1/2} \exp\left(-\frac{1}{2\sigma^2} [-\tilde{\boldsymbol{\beta}}_{j\ell}' \Lambda_{j\ell}^{-1} \tilde{\boldsymbol{\beta}}_{j\ell} + \boldsymbol{r}_\ell^{(j)'} \boldsymbol{r}_\ell^{(j)}] \right) \right],$$

where $\tilde{\boldsymbol{\beta}}_{j\ell} = \Lambda_{j\ell}(X_{j\ell}' r_\ell^{(j)})$, $X_{j\ell}$ is the node-specific regressor matrix of tree j and $\Lambda_{j\ell} = (X_{j\ell}' X_{j\ell} + V^{-1})^{-1}$.

After drawing a new tree, the terminal node parameters are drawn from the posterior distribution given by

$$\boldsymbol{\beta}_{j\ell}|\boldsymbol{r}^{(j)}, X_{j\ell}, \sigma^2, \sigma_{\beta_0}^2, \sigma_\beta^2, \mathcal{T}_j \sim \mathcal{N}_{q_{j\ell}}(\tilde{\boldsymbol{\beta}}_{j\ell}, \sigma^2 \Lambda_{j\ell}).$$

When all m trees and terminal node parameters are drawn, the posterior variance can be drawn, using the same posterior distribution as for BART and therefore is given by

$$\sigma^2|\mathcal{T}_1, \mathcal{B}_1, ..., \mathcal{T}_m, \mathcal{B}_m, X, \boldsymbol{y} \sim IG\left(\frac{T+\nu}{2}, \frac{S+\nu\lambda}{2}\right), \quad S = \sum_{t=1}^{T} (y_t - \hat{y}_t)^2,$$

where T denotes the total number of observations, and $\hat{y}_t$ the fitted response, with $\hat{y}_t = \sum_{j=1}^m g(\boldsymbol{x}_t; \mathcal{T}_j, \mathcal{B}_j)$. The only difference is because of the fitted response, which now includes a linear predictor.

The variance of the terminal node parameters, $\sigma_{\beta_0}^2$ and σ_β^2 are simulated from the following posterior distributions

$$\sigma_{\beta_0}^2|- \sim IG\left(a_0 + \frac{\sum_{j=1}^m b_j}{2}, b_0 + \frac{\boldsymbol{\beta}_0' \boldsymbol{\beta}_0}{2\sigma^2}\right)$$
$$\sigma_\beta^2|- \sim IG\left(a_1 + \frac{\sum_{j=1}^m \sum_{\ell=1}^{b_j} p_{j\ell}}{2}, b_1 + \frac{\boldsymbol{\beta}' \boldsymbol{\beta}}{2\sigma^2}\right)$$

where $\boldsymbol{\beta}_0$ is a vector consisting of the intercepts from all terminal nodes of all trees, and $\boldsymbol{\beta}$ is a vector with the slopes from all linear predictors of all trees. In addition, $p_{j\ell}$ denotes the number of covariates in the linear predictor of terminal node ℓ of tree j, and is thus equivalent to $q_{j\ell} - 1$.

3.3 TVP-BART

In both BART and MOTR-BART, the terminal node parameters, given by $\mu_{j\ell}$ and $\beta_{j\ell}$ respectively, are time-invariant. To allow for time variation in the terminal node of each tree, the conditional mean is modelled by time-varying parameters, rather than a time-invariant scalar or a linear predictor. Therefore, BART is changed into

$$y_t = \sum_{j=1}^{m} g_{j,t}(\boldsymbol{x}_t; \mathcal{T}_j, \mathcal{B}_j) + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \sigma^2), \tag{6}$$

where $\mathcal{B}_j = \{\boldsymbol{\beta}_{j\ell}\}_{\ell=1}^{b_j}$ with $\boldsymbol{\beta}_{j\ell} = (\beta_{j\ell 1}, \beta_{j\ell 2}, \ldots, \beta_{j\ell T})'$ being a vector containing the time-varying parameters for tree j and leaf node ℓ. Similar to BART, the relation of the current tree j with the other $m-1$

trees is via the partial residual which in this model is defined by

$$\begin{aligned} r_t^{(j)} &= y_t - \sum_{s \neq j}^{m} g_{s,t}(\boldsymbol{x}_t; \mathcal{T}_s, \mathcal{B}_s) \\ &= g_{j,t}(\boldsymbol{x}_t; \mathcal{T}_j, \mathcal{B}_j) + \varepsilon_t, \end{aligned} \tag{7}$$

such that the response variable that is fitted in j-th tree is the unexplained part of the target y_t.

Let the terminal node variables be denoted by $\boldsymbol{z}_t$. In the case of BART, the terminal node prediction is given by the mean of the response of the observations that fall into that leaf. Therefore, the terminal node variables only include a constant, meaning that $\boldsymbol{z}_t = 1$ for $t = 1, \ldots, T$. Following the approach of Hauzenberger et al. (2022b), estimating the time-varying coefficients can also be viewed as a node-specific static linear regression with terminal node covariates $\boldsymbol{z}_1, \ldots \boldsymbol{z}_T$, and response $\boldsymbol{r}^{(j)}$. Since only scalars are estimated in the terminal nodes, the covariates $\boldsymbol{z}_1$, $\boldsymbol{z}_2$,..., $\boldsymbol{z}_T$ are all of length one and equal to one. In addition, to impose random walk regularization, the design matrix L is a lower triangular matrix. Similarly, each $\beta_{j\ell t}$ is of dimension one and it should be noted that the time-varying parameter at time s is given by $\sum_{t=1}^{s} \beta_{j\ell t}$. This results in the following model to estimate the part of the response observations that fall into terminal node l that is explained by tree j

$$\underbrace{\begin{pmatrix} r_1^{(j)} \\ r_2^{(j)} \\ \vdots \\ r_T^{(j)} \end{pmatrix}}_{\boldsymbol{r}^{(j)}} = \underbrace{\begin{pmatrix} 1 & 0 & \ldots & 0 \\ 1 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \ldots & 1 \end{pmatrix}}_{L} \underbrace{\begin{pmatrix} \beta_{j\ell 1} \\ \beta_{j\ell 2} \\ \vdots \\ \beta_{j\ell T} \end{pmatrix}}_{\boldsymbol{\beta}_{j\ell}} + \sigma \underbrace{\begin{pmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_T \end{pmatrix}}_{\boldsymbol{\eta}}, \tag{8}$$

where $\eta_t \sim \mathcal{N}(0, 1)$ for $t = 1, \ldots, T$.

However, it should be noted that not all time observations may be present in a terminal node (i.e. $\mathbb{1}\{\boldsymbol{x}_t \in \mathcal{P}_{j\ell}\} = 0$ if observation t is not included in terminal node ℓ of the tree j). Note that when an observation is not present in terminal node ℓ, the corresponding row of the covariate matrix is removed. However, the relevant column is not removed. The terminal node prediction in the TVP-BART model is therefore given by $L_{[\mathcal{P}_{j\ell},.]}\boldsymbol{\beta}_{j\ell}$, where $L_{[\mathcal{P}_{j\ell},.]}$ denotes the matrix L but with the rows restricted to the time observations that are contained in terminal node ℓ. Therefore, Equation (6) can be rewritten as

$$y_t = \sum_{j=1}^{m} \sum_{\ell=1}^{b_j} \mathbb{1}\{\boldsymbol{x}_t \in \mathcal{P}_{j\ell}\} L_{[t,.]} \boldsymbol{\beta}_{j\ell} + \varepsilon_t \tag{9}$$

and similarly, Equation (7) can now also be written as

$$r_t^{(j)} = y_t - \sum_{s \neq j}^{m} \sum_{\ell=1}^{b_s} \mathbb{1}\{\boldsymbol{x}_t \in \mathcal{P}_{s\ell}\} L_{[t,.]} \boldsymbol{\beta}_{s\ell} \tag{10}$$

$$= \sum_{\ell=1}^{b_j} \mathbb{1}\{\boldsymbol{x}_t \in \mathcal{P}_{j\ell}\} L_{[t,.]} \boldsymbol{\beta}_{j\ell} + \varepsilon_t \tag{11}$$

Or in vector notation

$$\begin{aligned} \boldsymbol{r}^{(j)} &= \boldsymbol{y} - \sum_{s \neq j}^{m} \sum_{\ell=1}^{b_s} L_{[\mathcal{P}_{s\ell},.]} \boldsymbol{\beta}_{s\ell} \\ &= \sum_{\ell=1}^{b_j} L_{[\mathcal{P}_{j\ell},.]} \boldsymbol{\beta}_{j\ell} + \boldsymbol{\varepsilon} \end{aligned}$$

Comparing this vector notation to the partial residuals of MOTR-BART in Equation (5), it can be noted that TVP-BART is equivalent to MOTR-BART with fixed terminal node variable matrix $X_{j\ell} = L_{[\mathcal{P}_{j\ell},.]}$. Therefore, sampling proceeds as in standard MOTR-BART, where $X_{j\ell}$ is replaced by $L_{[\mathcal{P}_{j\ell},.]}$. Because some time observations are missing from a particular terminal node ℓ, it might be difficult to identify $\boldsymbol{\beta}_{j\ell}$. In addition, TVP-BART does not contain an intercept in contrast to MOTR-BART and therefore the variance of the terminal node parameters σ^2_β is simulated using all estimated parameters. The posterior variance of the terminal node parameters is simulated from

$$\sigma^2_\beta|- \sim IG\left(a_0 + \frac{T\sum_{j=1}^m b_j}{2}, b_0 + \frac{\sum_{j=1}^m \sum_{\ell=1}^{b_j} \boldsymbol{\beta}'_{j\ell}\boldsymbol{\beta}_{j\ell}}{2\sigma^2}\right).$$

3.4 SoftBART

Soft Bayesian Additive Regression Trees (SoftBART) by Linero & Yang (2018) is an extension of BART that allows for a smoother estimated function instead of the piece-wise constant function that BART learns. The previous Section discussed that the terminal nodes of a tree in BART may not include all observations. However, SoftBART includes all observations in the terminal nodes, albeit with weights specific to the terminal node and the time observation. This leads to the following expression

$$\begin{aligned} y_t &= \sum_{j=1}^m g(\boldsymbol{x}_t; \mathcal{T}_j, \mathcal{M}_j) + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \sigma^2) \\ g(\boldsymbol{x}_t; \mathcal{T}_j, \mathcal{M}_j) &= \sum_{\ell=1}^{b_j} \phi_{jt\ell}\mu_{j\ell} = \boldsymbol{\phi}'_{jt}\boldsymbol{\mu}_j, \end{aligned} \tag{12}$$

where $\boldsymbol{\phi}_{jt} = (\phi_{jt1}, \phi_{jt2}, \ldots, \phi_{jtb_j})'$ and $\boldsymbol{\mu}_j = (\mu_{j1}, \mu_{j2}, \ldots, \mu_{jb_j})$. Rather than $\boldsymbol{x}_t$ following a deterministic path down the tree, $\boldsymbol{x}_t$ instead follows a probabilistic path, with $\boldsymbol{x}_t$ going left at branch b with probability

$$\psi(\boldsymbol{x}_t; \mathcal{T}_j, \ell) = \psi(\frac{x_{ti} - C_b}{\tau_b}),$$

where τ_b is a bandwidth parameter associated with branch b which manages how sharp a decision is that is made (with a sharp decision model when $\tau_b \to 0$) and $\psi(.)$ is the logistic function. Given the tree structure, each branch node b is given a decision rule of the form $[x_{ti} \leq C_b]$, where $\boldsymbol{x}_t$ is going left down the tree if the condition is met and right down the tree if otherwise. Averaging over all potential courses, the probability of observation t going to leaf ℓ of tree j is

$$\phi_{jt\ell} = \phi(\boldsymbol{x}_t; \mathcal{T}_j, \ell) = \prod_{b \in A(\ell)} \psi(\boldsymbol{x}_t; \mathcal{T}_j, b)^{1-R_b}\{1 - \psi(\boldsymbol{x}_t; \mathcal{T}_j, b)^{R_b}\},$$

where $A(\ell)$ is the set of ancestor nodes of leaf ℓ and $R_b = 1$ if the path to ℓ goes right at b.

In SoftBART, each regression tree is developed through an MCMC backfitting algorithm and the Metropolis-Hastings step defines whether a tree structure is changed by either growing, pruning, changing or swapping. In general, this procedure is similar to BART. The following priors are frequently assumed

$$\mu_{j\ell}|\mathcal{T}_j \sim \mathcal{N}(0, \sigma^2_\mu/m), \quad \sigma^2 \sim IG(\nu/2, \nu\lambda/2).$$

Posterior Simulation

Similar to BART, a Gibbs sampler is used to generate draws from the posterior distribution of

$$p((\mathcal{T}_1, \mathcal{M}_1), ..., (\mathcal{T}_m, \mathcal{M}_m), \tau, \sigma^2|\boldsymbol{y}, X).$$

Note that in this case also the bandwidth parameter τ is contained in the joint posterior distribution. Similar to the description of BART, the conditional distribution of the j-th tree depends on the other $m-1$ trees through the partial residuals

$$\boldsymbol{r}^{(j)} \equiv \boldsymbol{y} - \sum_{k \neq j} g(X; \mathcal{T}_k, \mathcal{M}_k)$$

The Gibbs sampler starts by proposing a change to the first tree's structure which is then accepted or rejected via an MH step with the following probability

$$\alpha(\mathcal{T}_j, \mathcal{T}_j^*) = \min\left\{1, \frac{p(\boldsymbol{r}^{(j)}|\mathcal{T}_j^*, \tau, \sigma^2)p(\mathcal{T}_j^*)p(\mathcal{T}_j^* \to \mathcal{T}_j)}{p(\boldsymbol{r}^{(j)}|\mathcal{T}_j, \tau, \sigma^2)p(\mathcal{T}_j)p(\mathcal{T}_j \to \mathcal{T}_j^*)}\right\}$$

Using the MH algorithm, the posterior distribution of the bandwidth is generated using a random walk sample, where the proposed bandwidth τ^* is generated from $\log(\tau_j^*) = \log(\tau_j) + u_j$, with $u_j \sim \mathcal{U}(-1, 1)$ (Linero & Yang 2018). The transition density is given by $p(\tau^* \to \tau) = 0.5\tau^{-1}$ such that the bandwidths are accepted with probability

$$\begin{aligned}\alpha(\tau_j, \tau_j^*) &= \min\left\{1, \frac{p(\boldsymbol{r}^{(j)}|\tau_j^*, \mathcal{T}_j, \sigma^2)p(\tau_j^*)p(\tau_j^* \to \tau_j)}{p(\boldsymbol{r}^{(j)}|\tau_j, \mathcal{T}_j, \sigma^2)p(\tau_j)p(\tau_j \to \tau_j^*)}\right\} \\ &= \min\left\{1, \frac{p(\boldsymbol{r}^{(j)}|\tau_j^*, \mathcal{T}_j, \sigma^2)p(\tau_j^*)\tau_j^*}{p(\boldsymbol{r}^{(j)}|\tau_j, \mathcal{T}_j, \sigma^2)p(\tau_j)\tau_j}\right\}\end{aligned}$$

The conditional distribution of the partial residuals, which is necessary to calculate the acceptance probability in the MH steps, is given by

$$p(\boldsymbol{r}^{(j)}|\tau_j, \mathcal{T}_j, \sigma^2) = \frac{|2\pi\Omega|^{1/2}}{(2\pi\sigma^2)^{T/2}|2\pi\sigma_\mu^2 I|^{1/2}} \exp\left(-\frac{1}{2}\left[-\hat{\boldsymbol{\mu}}'\Omega^{-1}\hat{\boldsymbol{\mu}} + \frac{\boldsymbol{r}^{(j)\prime}\boldsymbol{r}^{(j)}}{\sigma^2}\right]\right)$$

where the vertical lines represent the determinant and

$$\Omega = (\Phi + \frac{\sigma_\mu^2}{m} I_{b_j})^{-1}, \quad \Phi = \sum_{t=1}^{T} \boldsymbol{\phi}_{jt}\boldsymbol{\phi}_{jt}'/\sigma^2, \quad \hat{\boldsymbol{\mu}} = \Omega \sum_{t=1}^{T} r_t^{(j)} \boldsymbol{\phi}_{jt}/\sigma^2$$

and $\boldsymbol{\phi}_{jt} = (\phi_{jt1}, .., \phi_{jtb_j})'$. In addition, the full conditional of the terminal node parameters of tree j contained in $\mathcal{M}_j$ is $\mathcal{N}(\hat{\mu}, \Omega)$.

3.5 SMOTR-BART

Soft Model Trees Bayesian Additive Regression Trees (SMOTR-BART) is an extension of MOTR-BART that combines SoftBART with the linear terminal-node model of MOTR-BART, proposed in a Master's Thesis by El Yaakoubi (2022). This methodology forms the foundation for TVP-SoftBART which is introduced in Section 3.6. Similar to SoftBART, each terminal node contains all time observation albeit with weights specific to the terminal node and time observation. The SMOTR-BART specification is given by

$$\begin{aligned} y_t &= \sum_{j=1}^{m} g(\boldsymbol{x}_t; \mathcal{T}_j, \mathcal{B}_j) + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \sigma^2) \\ g(\boldsymbol{x}_t; \mathcal{T}_j, \mathcal{B}_j) &= \sum_{\ell=1}^{b_j} \phi_{jt\ell}\boldsymbol{x}_{tj\ell}\boldsymbol{\beta}_{j\ell} = \tilde{\boldsymbol{x}}_{tj}\boldsymbol{\beta}_j, \end{aligned} \tag{13}$$

where $\tilde{\boldsymbol{x}}_{tj} = (\phi_{jt1}\boldsymbol{x}_{tj1}, \ldots, \phi_{jtb_j}\boldsymbol{x}_{tjb_j})$ represents the weighted covariates and $\boldsymbol{\beta}_j = (\boldsymbol{\beta}'_{j1}, \ldots, \boldsymbol{\beta}'_{jb_j})'$ is a large vector that is stacked with vectors each containing the parameters of a linear predictor in a single terminal node of tree j. This can also be written in vector notation as

$$\boldsymbol{y} = \sum_{j=1}^{m} \tilde{X}_j \boldsymbol{\beta}_j + \boldsymbol{\varepsilon}, \tag{14}$$

where

$$\tilde{X}_j = \begin{bmatrix} \phi_{j11}\boldsymbol{x}_{1j1} & \phi_{j12}\boldsymbol{x}_{1j2} & \cdots & \phi_{j1b_j}\boldsymbol{x}_{1jb_j} \\ \phi_{j21}\boldsymbol{x}_{2j1} & \phi_{j22}\boldsymbol{x}_{2j2} & \cdots & \phi_{j2b_j}\boldsymbol{x}_{2jb_j} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{jT1}\boldsymbol{x}_{Tj1} & \phi_{jT2}\boldsymbol{x}_{Tj2} & \cdots & \phi_{jTb_j}\boldsymbol{x}_{Tjb_j} \end{bmatrix}$$

The prior on the terminal node parameters is given by

$$\boldsymbol{\beta}_{j\ell}|\mathcal{T}_j \sim \mathcal{N}_{q_{j\ell}}(0, \sigma^2 V), \quad V = \frac{\sigma_\beta^2}{m} I_{q_{j\ell}}$$

where $I_{q_{j\ell}}$ denotes the identity matrix with $q_{j\ell}$ as the number of covariates in the linear predictor of terminal node ℓ of tree j including a constant regressor for the intercept. This prior is similar to SoftBART and MOTR-BART. The first because of the division by the number of trees (m), and the second because of prior variance structure $\sigma^2 V$. The prior in terms of the larger vector $\boldsymbol{\beta}_j$ is

$$\boldsymbol{\beta}_j|\mathcal{T}_j \sim \mathcal{N}_{\tilde{q}_j}(0, \sigma^2 V_j), \quad V_j = \frac{\sigma_\beta^2}{m} I_{\tilde{q}_j}$$

where $\tilde{q}_j = \sum_{\ell=1}^{b_j} q_{j\ell}$. Similarly to what Prado et al. (2021a) suggests, the intercept and slopes are penalized differently. The conjugate priors are $\sigma^2_{\beta_0} \sim IG(a_0, b_0)$ and $\sigma^2_\beta \sim IG(a_1, b_1)$ respectively.

Posterior Simulation

Given that the SMOTR-BART specification given in Equation (14) is equivalent to a reweighted MOTR-BART specification, posterior simulation is straightforward. The marginal likelihood of the partial residuals is given by

$$p(\boldsymbol{r}^{(j)}|\tilde{X}_j, \sigma^2, \mathcal{T}_j) = (\sigma^2)^{-(T/2)} \left[|V_j|^{-1/2} |\Lambda_j|^{1/2} \exp\left(-\frac{1}{2\sigma^2} [-\tilde{\beta}'_j \Lambda_j^{-1} \tilde{\beta}_j + \boldsymbol{r}^{(j)'} \boldsymbol{r}^{(j)}] \right) \right],$$

where $\tilde{\beta}_j = \Lambda_j(\tilde{X}'_j r^{(j)})$ and $\Lambda_j = (\tilde{X}'_j \tilde{X}_j + V_j^{-1})^{-1}$. The posterior distribution of the terminal node parameters is

$$\boldsymbol{\beta}_j|\boldsymbol{r}^{(j)}, \tilde{X}_j, \sigma^2, \sigma^2_{\beta_0}, \sigma^2_\beta, \mathcal{T}_j \sim \mathcal{N}_{\tilde{q}_j}(\tilde{\beta}_j, \sigma^2 \Lambda_j).$$

Given the prior on $\sigma^2_{\beta_0} \sim IG(a_0, b_0)$ and $\sigma^2_\beta \sim IG(a_1, b_1)$, the full conditional posterior distribution is given by

$$\sigma^2_{\beta_0}|- \sim IG\left(a_0 + \frac{\sum_{j=1}^m b_j}{2}, b_0 + \frac{m}{2\sigma^2} \boldsymbol{\beta}'_0 \boldsymbol{\beta}_0 \right)$$
$$\sigma^2_\beta|- \sim IG\left(a_1 + \frac{\sum_{j=1}^m \sum_{\ell=1}^{b_j} p_{j\ell}}{2}, b_1 + \frac{m}{2\sigma^2} \boldsymbol{\beta}' \boldsymbol{\beta} \right),$$

which is very similar to the posterior distribution of the terminal node variance of MOTR-BART, only

the number of trees, m, enter the distribution due to the different prior specification of $\boldsymbol{\beta}_{j\ell}$.

3.6 TVP-SoftBART

Similarly to TVP-BART, SoftBART is changed to allow for time-varying parameters in the leaf nodes of the trees used in SoftBART. Therefore, TVP-SoftBART is similarly given by

$$y_t = \sum_{j=1}^{m} g_{j,t}(\boldsymbol{x}_t; \mathcal{T}_j, \mathcal{B}_j) + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \sigma^2) \tag{15}$$

In TVP-SoftBART, however, the terminal node model is changed to include the probability $\phi_{jt\ell}$ that observation t ends in terminal node ℓ of tree j. In this way, all time observations are included in each terminal node, but with weights specific to the terminal node and observation. The TVP-SoftBART specification is given by

$$\begin{aligned} y_t &= \sum_{j=1}^{m} g_{j,t}(\boldsymbol{x}_t; \mathcal{T}_j, \mathcal{B}_j) + \varepsilon_t \\ g_{j,t}(\boldsymbol{x}_t; \mathcal{T}_j, \mathcal{B}_j) &= \sum_{\ell=1}^{b_j} \phi_{jt\ell} L_{[t,\cdot]} \boldsymbol{\beta}_{j\ell} \end{aligned} \tag{16}$$

This can also be written in vector notation as

$$\boldsymbol{y} = \sum_{j=1}^{m} \tilde{L}_j \boldsymbol{\beta}_j + \boldsymbol{\varepsilon},$$

where

$$\tilde{L}_j = \begin{bmatrix} \phi_{j11} L_{1,\cdot} & \phi_{j12} L_{1,\cdot} & \dots & \phi_{j1b_j} L_{1,\cdot} \\ \phi_{j21} L_{2,\cdot} & \phi_{j22} L_{2,\cdot} & \dots & \phi_{j2b_j} L_{2,\cdot} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{jT1} L_{T,\cdot} & \phi_{jT2} L_{T,\cdot} & \dots & \phi_{jTb_j} L_{T,\cdot} \end{bmatrix} = \begin{bmatrix} Diag(\boldsymbol{\phi}_{j,1:T,1}) L & Diag(\boldsymbol{\phi}_{j,1:T,2}) L & \dots & Diag(\boldsymbol{\phi}_{j,1:T,b_j}) L \end{bmatrix}$$

and

$$\boldsymbol{\beta}_j = \begin{bmatrix} \boldsymbol{\beta}_{j1} \\ \vdots \\ \boldsymbol{\beta}_{jb_j} \end{bmatrix}$$

with $\boldsymbol{\phi}_{j,1:T,\ell} = (\phi_{j1\ell}, \phi_{j2\ell}, \dots, \phi_{jT\ell})'$, $\boldsymbol{\beta}_{j\ell} = (\beta_{j\ell 1}, \beta_{j\ell 2}, \dots, \beta_{j\ell T})'$ for $\ell = 1, \dots, b_j$ and L being a lower triangular matrix as in Equation (8). In this way, $\boldsymbol{\beta}_j$ includes all time-varying node parameters of tree j.

Similarly to the previous BART algorithms, each regression tree is generated through an MCMC backfitting algorithm and a Metropolis-Hastings step to determine whether a tree is modified. In general, this follows the BART principle in the sense that the posterior results depend on regularization priors such that each tree is a weak learner. It should be noted that in TVP-SoftBART each prediction in the leaf node is constructed by a linear regression using a fixed covariate matrix $\tilde{L}_j$. This is similar to MOTR-BART, introduced by Prado et al. (2021a), where the prediction in each leaf node is created by using only the variables that were used as splitting conditions as regressor variables. However, this methodology does not allow for soft splitting but was later also introduced by El Yaakoubi (2022), called SMOTR-BART, which is necessary in this case to construct $\tilde{L}_j$.

The prior on the time-varying terminal node parameters that is used is

$$\boldsymbol{\beta}_{j\ell} | \mathcal{T}_j \sim \mathcal{N}_T(0, \sigma^2 V), \quad V = \frac{\sigma_\beta^2}{m} I_T$$

where I_T denotes the identity matrix. The prior in terms of the larger vector $\boldsymbol{\beta}_j$ is then given by

$$\boldsymbol{\beta}_j|\mathcal{T}_j \sim \mathcal{N}_{b_j \times T}(0, \sigma^2 V_j), \quad V_j = \frac{\sigma_\beta^2}{m} I_{b_j \times T},$$

where $I_{b_j \times T}$ denotes the identity matrix of dimension $(b_j \times T) \times (b_j \times T)$. In addition, an optional prior proposed in Prado et al. (2021a) is used where $\sigma_\beta^2 \sim IG(a_0, b_0)$. Similar to before, the prior on σ^2 is $IG(\nu/2, \nu\lambda/2)$.

Posterior Simulation

To determine the posterior distribution of the trees, leaf node parameters, bandwidth and variance given the data the Gibbs sampler is employed. To draw the structure of the trees a Metropolis-Hastings step is used, similar to before. The posterior only depends on the other trees via the partial residual. The partial residuals are given by

$$r_t^{(j)} = y_t - \sum_{s \neq j} \sum_{\ell=1}^{b_s} \phi_{st\ell} L_{[t,.]} \boldsymbol{\beta}_{s\ell} \tag{17}$$

$$= \sum_{j=1}^{b_j} \phi_{jt\ell} L_{[t,.]} \boldsymbol{\beta}_{j\ell} + \varepsilon_t \tag{18}$$

which can also be represented in the vector notation in the following way

$$\boldsymbol{r}^{(j)} = \tilde{L}_j \boldsymbol{\beta}_j + \boldsymbol{\varepsilon}$$

The Gibbs sampler starts by proposing a new tree structure in each step which is accepted or rejected using an MH step, with the following probabilities

$$\alpha(\mathcal{T}_j, \mathcal{T}_j^*) = \min\left\{1, \frac{p(\boldsymbol{r}^{(j)}|\mathcal{T}_j^*, \tau, \sigma^2) p(\mathcal{T}_j^*) p(\mathcal{T}_j^* \to \mathcal{T}_j)}{p(\boldsymbol{r}^{(j)}|\mathcal{T}_j, \tau, \sigma^2) p(\mathcal{T}_j) p(\mathcal{T}_j \to \mathcal{T}_j^*)}\right\}$$

$$\alpha(\tau_j, \tau_j^*) = \min\left\{1, \frac{p(\boldsymbol{r}^{(j)}|\tau_j^*, \mathcal{T}_j, \sigma^2) p(\tau_j^*) \tau_j^*}{p(\boldsymbol{r}^{(j)}|\tau_j, \mathcal{T}_j, \sigma^2) p(\tau_j) \tau_j}\right\}.$$

While most of the ratios are the same as before, the marginal likelihood of $\boldsymbol{r}^{(j)}$ does change. The full conditional of $\boldsymbol{r}^{(j)}$ is given by

$$p(\boldsymbol{r}^{(j)}|\tilde{L}_j, \sigma^2, \mathcal{T}_j) = (\sigma^2)^{-T/2} \left[|V_j|^{-1/2} |\Lambda_j|^{1/2} \times \exp\left(-\frac{1}{2\sigma^2}[-\tilde{\boldsymbol{\beta}}_j' \Lambda_j^{-1} \tilde{\boldsymbol{\beta}}_j + \boldsymbol{r}^{(j)'} \boldsymbol{r}^{(j)}]\right)\right] \tag{19}$$

where $\tilde{\boldsymbol{\beta}}_j = \Lambda_j(\tilde{L}_j' r^{(j)})$ and $\Lambda_j = (\tilde{L}_j' \tilde{L}_j + V_j^{-1})^{-1}$.

The posterior distribution of the leaf node parameter vector $\boldsymbol{\beta}_j$ using the MOTR-BART specification is given by

$$\boldsymbol{\beta}_j|\boldsymbol{r}^{(j)}, \tilde{L}_j, \sigma^2, \sigma_\beta^2, \mathcal{T}_j \sim \mathcal{N}_{b_j \times T}(\tilde{\boldsymbol{\beta}}_j, \sigma^2 \Lambda_j).$$

It should be noted however that this may be computationally expensive, due to the inversion necessary to calculate the matrix Λ_j which is of dimension $[(b_j \times T) \times (b_j \times T)]$.

To decrease the computational burden, a singular value decomposition (SVD) of the design matrix $\tilde{L}_j$ is used following Hauzenberger et al. (2022b). The SVD of the matrix $\tilde{L}_j$ is

$$\underbrace{\tilde{L}_j}_{T \times (T \times b_j)} = \underbrace{U}_{T \times T} \underbrace{\Lambda}_{T \times T} \underbrace{W'}_{T \times (T \times b_j)}, \tag{20}$$

where U and W are orthogonal matrices such that $W^{'}W = I_T$ and Λ denotes a diagonal matrix with singular values, denoted by $\boldsymbol{\lambda}$, of $\tilde{L}_j$ as diagonal elements. It should be noted that $rank(\tilde{L}_j) = \min\{T, (T \times b_j)\} = T$. Therefore, the approach of Hauzenberger et al. (2022b) of using the SVD results in an exact low-rank structure implying no loss of information through the use of the SVD. This is in contrast to Trippe et al. (2019) which introduces the SVD approach in a Bayesian context to identify a lower-dimensional subspace such that posterior computation can be performed at a lower computational expense.

The reason to use the SVD instead of $\tilde{L}_j$ is that several convenient properties of the SVD speed up computation. For instance, using a Gaussian prior leads to a computationally convenient expression of the posterior distribution of $\boldsymbol{\beta}$ which avoids complex matrix manipulations such as inversion and Cholesky decomposition of high-dimensional matrices. Section 4.2 contains a discussion on how much the SVD speeds up computation in comparison to a Cholesky decomposition of the matrix Λ_j^{-1}.

With this SVD, the posterior of $\boldsymbol{\beta}_j$ takes the following form

$$\begin{aligned}
\boldsymbol{\beta}_j|\boldsymbol{r}^{(j)}, \tilde{L}_j, \sigma^2, \sigma_\beta^2, \mathcal{T}_j &\sim \mathcal{N}(\boldsymbol{\mu}_\beta, \sigma^2 \Lambda_j^{SVD}) \\
\boldsymbol{\mu}_\beta &= \Lambda_j^{SVD}(\tilde{L}_j^{'}\boldsymbol{r}^{(j)}) \\
&= \left[W \text{diag}\left(\frac{\boldsymbol{\lambda}}{\frac{m}{\sigma_\beta^2}\boldsymbol{\iota}_T + \boldsymbol{\lambda}^2}\right)\right] U^{'}\boldsymbol{r}^{(j)} \\
\Lambda_j^{SVD} &= (W \text{diag}(\boldsymbol{\lambda} \odot \boldsymbol{\lambda}) W^{'} + V_j^{-1})^{-1} \\
&= V_j - V_j W(\text{diag}(\boldsymbol{\lambda} \odot \boldsymbol{\lambda})^{-1} + W^{'}V_j W)^{-1} W^{'} V_j
\end{aligned} \tag{21}$$

where $\odot$ denotes the dot product. The computational burden is now given by the matrix $\Xi = (\text{diag}(\lambda \odot \lambda)^{-1} + W^{'}V_j W)^{-1}$. However, given the prior variance that is used for $\boldsymbol{\beta}_j$ consists of a scalar times the identity matrix, also known as a ridge prior, the matrix Ξ reduces to a diagonal matrix.

Given the prior on $\sigma_\beta^2 \sim IG(a_0, b_0)$, the full conditional posterior distribution can be derived in the following way.

$$\begin{aligned}
p(\sigma_\beta^2|.) &\propto \left(\frac{1}{\sigma \frac{\sigma_\beta}{\sqrt{m}}\sqrt{2\pi}}\right)^{T\sum_{j=1}^m b_j} \exp\left\{-\frac{m}{2\sigma^2\sigma_\beta^2}\sum_{j=1}^m \boldsymbol{\beta}_j^{'}\boldsymbol{\beta}_j\right\} \times (\sigma_\beta^2)^{-(a_0+1)} \exp\left\{-\frac{b_0}{\sigma_\beta^2}\right\} \\
&\propto (\sigma_\beta^2)^{-(a_0 + \frac{T\sum_{j=1}^m b_j}{2} + 1)} \exp\left\{-\frac{1}{\sigma_\beta^2}\left(\frac{m}{2\sigma^2}\sum_{j=1}^m \boldsymbol{\beta}_j^{'}\boldsymbol{\beta}_j + b_0\right)\right\}
\end{aligned}$$

Therefore the conditional posterior density of σ_β^2 is $IG(a_0 + \frac{T\sum_{j=1}^m b_j}{2}, b_0 + \frac{m}{2\sigma^2}\sum_{j=1}^m \boldsymbol{\beta}_j^{'}\boldsymbol{\beta}_j)$.

3.7 Posterior Predictive Density

To generate predictions within a Bayesian framework, it is common to use the posterior predictive density. This is essentially a density forecast that not only reflects the uncertainty in the model parameters but also the noise estimate at each Gibbs sampler. Posterior draws of this density can be used to obtain point forecasts, by either taking the mean or median. In addition, this density allows us to easily obtain other quantities, for example, predictions of certain quantiles.

Given the past data $\boldsymbol{y}$ and X up to time T, the posterior predictive density of y_{T+1} at time $T+1$ is given by

$$p(y_{T+1}|\boldsymbol{y}, X, \boldsymbol{x}_{T+1}) = \int p(y_{T+1}|\boldsymbol{x}_{T+1}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\boldsymbol{y}, X) d\boldsymbol{\theta}, \tag{22}$$

where $\boldsymbol{x}_{T+1} = (x_{1,T+1}, \dots, x_{k,T+1})^{'}$ is a k-dimensional vector containing the covariates at time $T+1$, and where $\boldsymbol{\theta}$ is generic notation that refers to all parameters and latent states, such as the time-varying

parameters, in the model.

More specifically in the BART framework, the posterior predictive distribution is given by

$$y_{T+1}|\boldsymbol{y}, X, \boldsymbol{x}_{T+1} \sim \mathcal{N}(\hat{f}(\boldsymbol{x}_{T+1}), \hat{\sigma}^2)$$

where $\hat{f}(\boldsymbol{x}_{T+1})$ is the posterior conditional mean estimate and $\hat{\sigma}^2$ is the posterior estimate of the noise variance.

In each Gibbs step, samples of y_{T+1} are generated using the model as DGP, where the parameters and latent states are replaced by drawing from the posterior distributions. For the time-invariant models, these posterior draws follow naturally from the Gibbs sampler. For TVP-BART and TVP-SoftBART simulating from the posterior predictive distribution is slightly more involved. Because of the time-variation a new parameter $\beta_{j\ell,T+1}$ needs to be simulated from the prior distribution since there exists no posterior distribution. This is in contrast to the time-varying parameters up to time T which are drawn from their posterior distributions. The posterior conditional mean estimate of TVP-BART is given by

$$\begin{aligned}\hat{f}(\boldsymbol{x}_{T+1}) &= \sum_{j=1}^{m}\sum_{\ell=1}^{b_j} \mathbb{1}\{\boldsymbol{x}_{T+1} \in \mathcal{P}_{j\ell}\} L_{[T,.]}\boldsymbol{\beta}_{j\ell} + \beta_{j\ell,T+1} \\ &= \sum_{j=1}^{m}\left[\sum_{\ell=1}^{b_j} \mathbb{1}\{\boldsymbol{x}_{T+1} \in \mathcal{P}_{j\ell}\} L_{[T,.]}\boldsymbol{\beta}_{j\ell}\right] + \beta_{j,T+1},\end{aligned}$$

where the last equation follows from the fact that observation $T+1$ can only end up in one of the terminal nodes of tree j and it does not matter in which of the terminal nodes, since $\beta_{j\ell,T+1}$ is drawn from the prior model which is the same for each terminal node. The prior model from which $\beta_{j,T+1}$ is drawn is given by

$$\beta_{j,T+1} \sim \mathcal{N}(0, \sigma^2 V_{T+1}), \quad V_{T+1} = \sigma_\beta^2.$$

It is important to note that $\boldsymbol{\beta}_{j\ell}$, σ^2 and σ_β^2 are posterior draws from a single Gibbs iteration. These steps need to be repeated for each Gibbs iteration such that each Gibbs iteration results in a new posterior conditional mean estimate and posterior variance. In addition, $\sum_{j=1}^{m}\beta_{j,T+1}$ can be simulated from $\mathcal{N}(0, m\sigma^2\sigma_\beta^2)$ instead of simulating each $\beta_{j,T+1}$ separately.

Similarly, the posterior conditional mean estimate of TVP-SoftBART is given by

$$\begin{aligned}\hat{f}(\boldsymbol{x}_{T+1}) &= \sum_{j=1}^{m}\sum_{\ell=1}^{b_j} \phi_{j,T+1,\ell}\left[L_{[T,.]}\boldsymbol{\beta}_{j\ell} + \beta_{j\ell,T+1}\right] \\ &= \sum_{j=1}^{m}\left[\sum_{\ell=1}^{b_j} \phi_{j,T+1,\ell} L_{[T,.]}\boldsymbol{\beta}_{j\ell}\right] + \beta_{j,T+1}\end{aligned}$$

where the last equation follows from $\sum_{\ell=1}^{b_j}\phi_{j,T+1,\ell} = 1$ for all j and the fact that the prior distribution of $\beta_{j\ell,T+1}$ is the same for each terminal node ℓ of tree j. The prior distribution of $\beta_{j,T+1}$ given by

$$\beta_{j,T+1} \sim \mathcal{N}(0, \sigma^2 V_{T+1}), \quad V_{T+1} = \frac{\sigma_\beta^2}{m}.$$

Lastly, it can be noted that $\sum_{j=1}^{m}\beta_{j,T+1}$ can be sampled from $\mathcal{N}(0, \sigma^2\sigma_\beta^2)$ instead of sampling each future parameter $\beta_{j,T+1}$ for each tree separately. Since $\beta_{j,T+1}$ for each terminal node is simulated from its prior distribution in both TVP-BART and TVP-SoftBART, it has zero mean. Therefore it will only affect the confidence interval.

For each Gibbs sample, *normal_samples_per_gibbs_sample* number of samples are taken from the predictive distribution estimate. Often, only one sample is taken, but more can be taken to better approxi-

mate the posterior predictive density distribution. This results in a total of a *normal_samples_per_gibbs_sample* times the number of post-burn-in iterations. After repeating these steps for each Gibbs sampler, the mean can be used as a point forecast. For each iteration in the MCMC sampler, the following steps are repeated to obtain posterior draws from the predictive distribution.

1) Draw $\boldsymbol{\beta}_{j\ell}$ for each tree j, σ^2 and σ_β^2 using the Gibbs sampling algorithm as described earlier.

2) Draw y_{T+1} *normal_samples_per_gibbs_sample* times from
$\mathcal{N}(\sum_{j=1}^{m}\sum_{\ell=1}^{b_j}\mathbb{1}\{\boldsymbol{x}_{T+1}\in\mathcal{P}_{j\ell}\}L_{[T,.]}\boldsymbol{\beta}_{j\ell},\sigma^2(1+m\sigma_\beta^2))$ for TVP-BART
$\mathcal{N}(\sum_{j=1}^{m}\sum_{\ell=1}^{b_j}\phi_{j,T+1,\ell}\sum_{t=1}^{T}\beta_{j\ell t},\sigma^2(1+\sigma_\beta^2))$ for TVP-SoftBART

3.8 Posterior Sampling Procedure

Algorithm 1 gives a full structure of the TVP-SoftBART algorithm.[2] The trees, hyper-parameters, partial residuals, the number of MCMC iterations and the number of samples from the posterior predictive distribution have to be initialised. Within each MCMC iteration, candidate trees $\mathcal{T}_j^*$ are generated which may be accepted or rejected as the current tree with probability $\alpha(\mathcal{T}_j,\mathcal{T}_j^*)$. Similarly a new bandwidth τ_j^* is proposed and either accepted or rejected. After this, the terminal node parameters $\boldsymbol{\beta}_j$ are generated. After drawing the trees and terminal node parameters, the final fits $\hat{\boldsymbol{y}}$ are generated, as well as the error variance of the terminal node parameters σ_β^2 and the error variance σ^2. Finally, *normal_samples_per_gibbs_sample* numbers of samples from the posterior predictive distribution are obtained.

TVP-BART would result in a very similar algorithm only some steps are omitted, for example, in the MH-step for the bandwidth τ, there is no soft splitting and the regressor matrix of the terminal node model is $L_{[\mathcal{P}_{j\ell},.]}$. Therefore, it is omitted for brevity.

For the remainder of this paper, 1000 burn-in and 2500 post burn-in iterations are used, 10 trees are used ($m=10$) and 100 samples are drawn per Gibbs sample from the posterior predictive distribution (*normal_samples_per_gibbs_sample* $=100$).[3] This results in a total of $250,000$ draws from the posterior predictive density. Further, $\alpha=0.5$, $\beta=1$, $\nu=3$, $\lambda=0.1$, $a_0=1$ and $b_0=1$.

[2]MOTR-BART, SMOTR-BART, TVP-BART and TVP-SoftBART are available as an R package on https://github.com/EoghanONeill.

[3]As an experiment, the third DGP of the simulation exercise discussed in Section 4 is also run with *normal_samples_per_gibbs_sample* set to 1 instead of 100. Results are reported in Appendix A and can be compared to Table III. Generally, all models perform worse with this number of samples, however, it does require less storage.

Algorithm 1 TVP-SoftBART Algorithm

Input: $\boldsymbol{y}$ (response variable) and $\boldsymbol{X}$ (set of independent variables) which are standardized to mean zero and unit variance

Output: The posterior distribution of trees and terminal node parameters

Initialization: $\mathcal{T} = (\mathcal{T}_1, \ldots, \mathcal{T}_m)$ to stumps, $\beta_{j\ell t} = 0$, $R_1 = \boldsymbol{y}$, number of trees (m), the number of MCMC iterations (burn-in and post-burn-in)$(nIter)$,$normal\_samples\_per\_gibbs\_sample$

Initialize prior hyperparameters: α, β, ν, λ, a_0 and b_0.

for k in $1 : nIter$ **do**

  for j in $1 : m$ **do**

    Update the current partial residual $r_j^k = \boldsymbol{y} - \sum_{i \neq j} g(\boldsymbol{X}; \mathcal{T}_i, \mathcal{M}_i)$

    Propose a new tree $\mathcal{T}_j^*$ by growing, pruning, changing or swapping

    Metropolis-Hastings step

      Compute $\alpha(\mathcal{T}_j, \mathcal{T}_j^*)$

      Sample $u \sim \mathcal{U}(0, 1)$

    if $u \leq \alpha(\mathcal{T}_j, \mathcal{T}_j^*)$ **then**

      $\mathcal{T}_j = \mathcal{T}_j^*$

    else

      $\mathcal{T}_j = \mathcal{T}_j$

    end if

      Follow a similar procedure for τ by computing $\alpha(\tau_j, \tau_j^*)$

    Terminal Node Model

    for $\ell = 1 : b_j$ **do**

      Compute $\phi_{jt\ell}$ $\forall t = 1, \ldots, T$ to construct $\tilde{L}_j$

      Compute the SVD of $\tilde{L}_j = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{W}'$

        a) Compute $\boldsymbol{\mu}_\beta$ using Equation (21)

        b) Simulate $\boldsymbol{a} \sim \mathcal{N}(\boldsymbol{0}_{T \times b_j}, V_j)$ and $\boldsymbol{b} \sim \mathcal{N}(\boldsymbol{0}_T, \text{diag}(\boldsymbol{\lambda} \odot \boldsymbol{\lambda})^{-1})$

        c) Update $\boldsymbol{\beta}_j^k = \boldsymbol{\mu}_\beta + \sigma(\boldsymbol{a} - V_j \boldsymbol{W}\boldsymbol{\Xi}(\boldsymbol{W}'\boldsymbol{a} + \boldsymbol{b}))$

    end for

  end for

  Update $\hat{y}_t^k = \sum_{j=1}^m g_{j,t}(\boldsymbol{x}_t; \mathcal{T}_j, \mathcal{M}_j) = \sum_{j=1}^m \sum_{\ell=1}^{b_j} \phi_{jt\ell} L_{[t,.]} \beta_{j\ell t}^k$ $\forall t = 1, \ldots, T$

  or $\hat{\boldsymbol{y}}^k = \sum_{j=1}^m \tilde{L}_j \boldsymbol{\beta}_j^k$

  Update $(\sigma^2)^k$ by sampling from $p(\sigma^2 | \mathcal{T}_1, \mathcal{B}_1, \ldots, \mathcal{T}_m, \mathcal{B}_m, \boldsymbol{X}, \boldsymbol{y})$

  Update $(\sigma_\beta^2)^k$ by sampling from $p(\sigma_\beta^2 | \mathcal{T}_1, \mathcal{B}_1, \ldots, \mathcal{T}_m, \mathcal{B}_m, \boldsymbol{X}, \boldsymbol{y})$

  Out-of-Sample Forecasting

  Draw $normal\_samples\_per\_gibbs\_sample$ times from $\mathcal{N}(\sum_{j=1}^m \sum_{\ell=1}^{b_j} \phi_{j,T+1,\ell} \sum_{t=1}^T \beta_{j\ell t}^k, (\sigma^2)^k(1 + (\sigma_\beta^2)^k))$

end for

3.9 Forecast Evaluation Methodology

This Section discusses the error metrics that are used to evaluate the forecasts generated by the models. Instead of only focusing on point forecasts, density forecasts are also evaluated. In this case, the uncertainty surrounding the model forecasts is also included in the evaluation. Rather than focusing on the entire density, two additional density evaluation metrics are also considered. However, given that the models considered in this paper do not include a heteroskedastic error variance, evaluating the models on these additional error metrics may not drastically change conclusions that are based on the error metric which considers the entire density.

First, the point forecasts are evaluated using the root mean squared error (RMSE). This is one of the most commonly used error metrics to evaluate point forecasts. The RMSE is given by

$$RMSE = \sqrt{\frac{1}{T_{oos}} \sum_{t=1}^{T_{OOS}} (\hat{y}_t - y_t)^2}, \tag{23}$$

where T_{oos} is the number of observations over which the forecast is evaluated, for example, the out-of-sample period. $\hat{y}_t$ is the point forecast which is the mean of the posterior predictive distribution resulting from the Gibbs sampler and y_t is the actual realization.

Second, the density forecasts are evaluated using the *continuous ranked probability score* (CRPS). Following Groen et al. (2013) this CRPS is used to prevent drawbacks of the usually employed log score the logarithm of the predictive density evaluated in y_t). These include its vulnerability to outliers and that it may not deem observations that are close but not equal to realizations important, also noted in Gneiting & Raftery (2007). Therefore, CRPS is used as a baseline evaluation of the overall predictive density.

The CRPS is given by

$$\begin{aligned} CRPS_t(y_t) &= \int_{-\infty}^{\infty} (F(z) - \mathbb{1}(y_t \leq z))^2 dz \\ &= E|\hat{y}_t - y_t| - 0.5E|\hat{y}_t - \hat{y}_t^{'}|, \end{aligned} \tag{24}$$

where $F(.)$ is the cumulative distribution function belonging to the predictive density, y_t is the realization of the forecasted variable and $\hat{y}_t$ and $\hat{y}_t^{'}$ are independent random draws from the posterior predictive density. The lower the value the more accurate the density forecast. The CRPS can be easily computed using the posterior draws from the MCMC sampler and random resampling. The average across the out-of-sample period is used and denoted by avCRPS.

Third, two implementations of the *quantile-weighted CRPS* (qwCRPS) are used to focus on tail forecasts (Gneiting & Ranjan 2011). The quantile score is computed as

$$QS_{\tau,t} = (y_t - \mathcal{Q}_{\tau,t})(\tau - \mathbb{1}(y_t \leq \mathcal{Q}_{\tau,t})),$$

where $\mathcal{Q}_{\tau,t}$ is the forecast of quantile τ. The qwCRPS is computed as a weighted sum of quantile scores at a range of J quantiles

$$qwCRPS_t = \frac{2}{J-1} \sum_{j=1}^{J-1} \omega(\tau_j) QSS_{\tau_j,t} \tag{25}$$

with $\tau_j = j/J$. Similar to Clark et al. (2023), 19 quantiles are used such that $\tau \in \{0.05, 0.10, \ldots, 0.90, 0.95\}$ where $J = 20$ to compute the weighted scores. The first implementation of qwCRPS targets both tails of the predictive distribution using the weight function $\omega(\tau_j) = (2\tau_j - 1)^2$. The second implementation targets only the left tail, and with this the downside risk, by using the weight function $\omega(\tau_j) = (1 - \tau_j)^2$. To report these scores, the averages across the out-of-sample period are used, where *avCRPS-T* is used to refer to the first and *avCRPS-L* to the second implementation of qwCRPS.

4 Simulation Study

In this Section, the use of the proposed methodology is illustrated and whether it successfully covers different features of the data-generating processes (DGPs). To assess the performance of new methodologies in this context, it is common to use synthetic data, see for example Groen et al. (2013) and Hauzenberger et al. (2024).

The simulation consists of different DGPs moving from a simple non-linear specification to a framework including time-varying parameters and a structural break. This simulation setup is meant to compactly encapsulate the usual nonlinearities and time-variation in parameters often considered in empirical studies.[4] The precise form of the non-linear DGP that includes TVPs is given by

$$y_{t+1} = \beta_0 + \beta_{1t} y_t + \beta_{2t} x_{1t} + \beta_{3t} x_{2t}^2 + \beta_{4t} \sin(x_{3t} y_t) + \beta_{5t} x_{1,t-1} + \sigma \varepsilon_{t+1}, \quad \text{for} \quad t = 1, \dots, T-1 \tag{26}$$

with $\varepsilon_t \sim \mathcal{N}(0,1)$, $\sigma = 0.1$ and $x_{j,t} \sim \mathcal{N}(0,1)$ for $j = 1, \dots, 5$. The regressors x_{4t} and x_{5t} are not included in the model, such that they have a zero coefficient. They are generated to create some noise around the choice of covariate for the models. Each time-varying parameter follows a random walk with standard deviation 0.015 ($\beta_{jt} \sim \mathcal{N}(\beta_{j,t-1}, (0.015)^2)$) for $j = 1, \dots, 4$, but β_{5t} contains a structural break such that $\beta_{5t} = -0.4$ for $t \leq T/2$ and $\beta_{5t} = 0.75$ for $t > T/2$. The initial coefficients are given by $\beta_0 = 0.5$, $\beta_{10} = 0.6$, $\beta_{20} = -0.2$, $\beta_{30} = 0.4$, $\beta_{40} = 0.2$, and $y_1 = 0$ and $x_{10} = 0$.

In this simulation study, three different specifications are considered. The first DGP is a non-linear specification without time variation in the parameters. In this case, each parameter is set to its initial value and is not allowed to change. The structural break parameter β_{5t} is set to 0, so no structural break is present in this DGP. The second DGP consists of random walk variation in the parameters β_{jt} for $j = 1, \dots, 4$. The last DGP allows for both random walk time-variation as well as a structural break. Given that machine learning based algorithms might overfit, the time-invariant DGP is included as a natural check. In addition, these DGPs offer the possibility to show the performance of the approaches and under which circumstances the benefits are especially pronounced. The methods that are investigated BART, SoftBART, MOTR-BART, SMOTR-BART, TVP-BART and TVP-SoftBART.[5] All methods are estimated using 1000 iterations as the burn-in period and 2500 as the post-burn-in period.

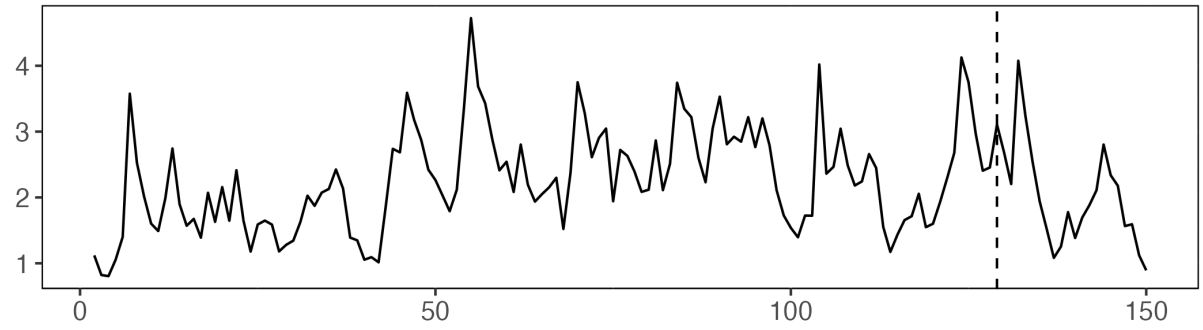
The design of this simulation exercise is recursive using an expanding window of data, where T is set to 150 and the last 20 observations are considered as the hold-out sample for evaluation. Using only a single draw may lead to questions about whether the favourable performance is particularly due to the specific realization of the DGP. Therefore, instead of only simulating one realization, the proposed DGP is simulated five times. For each out-of-sample time observation, a new model needs to be estimated which makes the procedure computationally expensive. In this case, each model is re-estimated 20 times for three different DGPs. To keep the computation feasible, only five different realizations of the DGPs are considered.[6] Figure I illustrates a single realization of each DGP for $T = 150$. Other realizations typically look very similar and are thus omitted for brevity.

[4] Inspired by the Friedman Model in Friedman (1991) but adjusted to a time series context by including also lags of the target and inputs. In this way, the DGP is constructed to mirror the dynamics properties observed for actual macroeconomic aggregates.

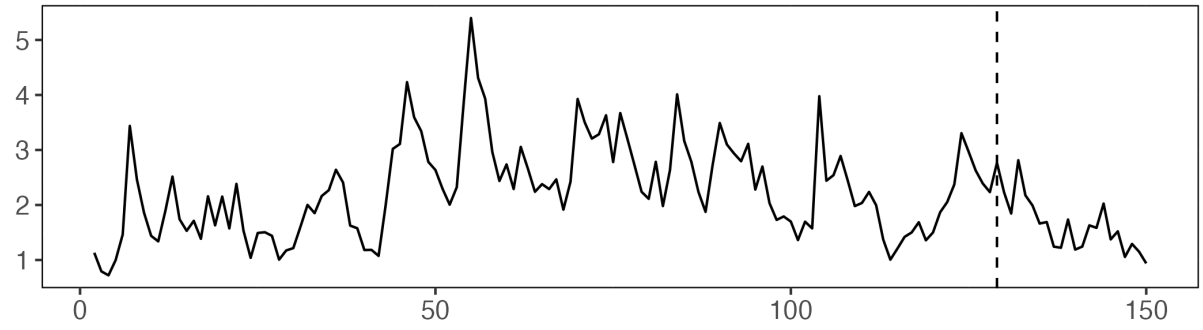
[5] BART is estimated using bartMachine and SoftBART using SoftBART in R. The default settings are used, only the burn-in and post-burn-in iterations are changed.

[6] Estimating TVP-BART ranges from 3.065 to 3.859 minutes and TVP-SoftBART ranges from 4.687 to 6.148 minutes on an Apple M3. The increase in estimation time is due to the addition of more time observations in the expanding window. Simulating the 3 DGPs 5 times for 20 out-of-sample observations takes roughly 17.5 hours for TVP-BART and 27.5 hours for TVP-SoftBART.

(a) DGP 1 (no time-variation)



(b) DGP 2 (random-walk time-variation)



(c) DGP 3 (random-walk time-variation and structural break)

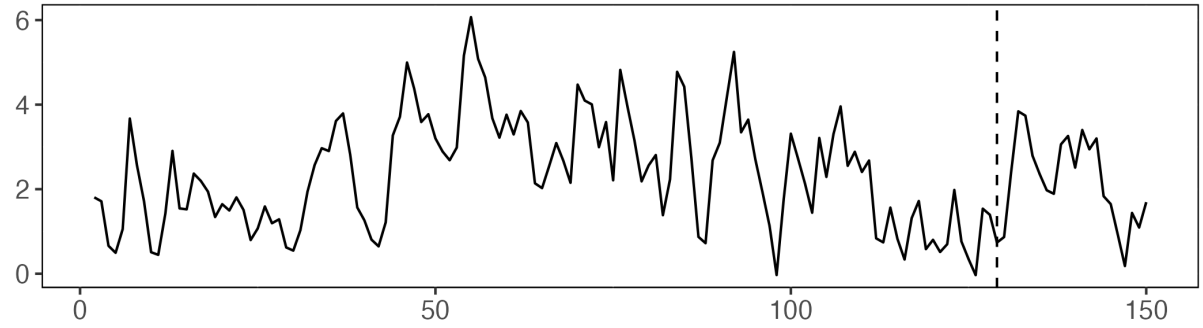


FIGURE I. Single realization from the DGPs for $T = 150$. The dashed black line marks the beginning of the hold-out period.

4.1 Simulation Results

Tables I, II and III present the results for the three different Data Generating Processes (DGPs). Table I illustrates the (absolute) error metrics for the DGP characterized solely by non-linearity, without any time variation. Table II introduces time variation in the parameters via a random walk, and Table III further complicates this with a structural break, adding a more rigorous time variation. Each model's performance is represented by the mean values over five simulations, calculated over the in-sample period (from t=1 to t=129) and the out-of-sample period (from t=130 to t=149), where standard deviations are represented in parentheses.

In Table I, TVP-BART demonstrates the lowest RMSE, indicating its superior accuracy for in-sample fits, and is closely followed by BART. However, the average RMSE of the time-invariant BART-based models closely follow TVP-BART and obtain lower standard deviations. TVP-SoftBART obtains the

TABLE I. Forecast Evaluation for Simulations - DGP 1 (no time-variation)

	In-Sample Performance				Out-of-Sample Performance			
	RMSE	avCRPS	avCRPS-T	avCRPS-L	RMSE	avCRPS	avCRPS-T	avCRPS-L
BART	0.097	**0.052**	**0.014**	**0.018**	0.271	0.141	0.033	0.044
	(0.035)	(0.009)	(0.002)	(0.003)	(0.096)	(0.037)	(0.009)	(0.007)
SoftBART	0.119	0.091	0.027	0.033	0.200	0.118	0.031	0.041
	(0.006)	(0.002)	(0.001)	(0.001)	(0.079)	(0.028)	(0.006)	(0.007)
MOTR-BART	0.107	0.062	0.014	0.020	0.185	**0.095**	**0.022**	**0.030**
	(0.004)	(0.002)	(0.000)	(0.001)	(0.056)	(0.020)	(0.005)	(0.005)
SMOTR-BART	0.148	0.084	0.019	0.027	**0.180**	0.104	0.023	0.033
	(0.005)	(0.003)	(0.001)	(0.001)	(0.013)	(0.009)	(0.001)	(0.003)
TVP-BART	**0.094**	0.079	0.024	0.029	0.413	0.231	0.054	0.076
	(0.028)	(0.015)	(0.004)	(0.005)	(0.109)	(0.048)	(0.012)	(0.017)
TVP-SoftBART	0.202	0.167	0.051	0.060	0.564	0.311	0.076	0.093
	(0.053)	(0.020)	(0.006)	(0.007)	(0.185)	(0.082)	(0.022)	(0.018)

Note: Simulation results for DGP 1, see Equation (26). For each error metric, the (absolute) means over 5 simulations are shown calculated over the in-sample period ($t = 1, \ldots, 129$) and out-of-sample period ($t = 130, \ldots, 149$). Bold numbers indicate the best performance (lowest error metric). The standard deviation is reported in parentheses. See Section 3.9 for error metrics definition.

largest RMSE and standard deviation and therefore generates the worst in-sample fits. Similar conclusions apply to the density forecasts based on the remaining error metrics. Despite TVP-BART's strong in-sample performance, it struggles with out-of-sample forecasts. Conversely, SMOTR-BART achieves the most accurate point forecasts, and MOTR-BART excels in density forecasts.

Table II displays the estimation results for the DGP including random walk time variation in the parameters. Again, TVP-BART obtains the lowest average RMSE in-sample but is closely followed by the time-invariant BART models. In addition, TVP-SoftBART performs the worst, which can be noted from the highest average RMSE and standard deviation. Also for the density forecasts, similar conclusions as for the DGP without time-variation apply. However, the error metrics are generally higher indicating that the models have more difficulty accurately capturing the time-variation.

Table III presents the outcomes of the simulation for the DGP that includes both random walk time variation and structural break. BART, SoftBART, MOTR-BART and SMOTR-BART struggle substantially more in capturing this time series compared to the previous DGPs. This is evident from for example the average RMSE that is almost three times larger than the ones from the second DGP shown in Table II and approximately four times larger than the ones from the first DGP shown in Table

TABLE II. Forecast Evaluation for Simulations - DGP 2 (random-walk time-variation)

	In-Sample Performance				Out-of-Sample Performance			
	RMSE	avCRPS	avCRPS-T	avCRPS-L	RMSE	avCRPS	avCRPS-T	avCRPS-L
BART	0.135	**0.073**	**0.019**	**0.024**	0.374	0.199	0.046	0.063
	(0.032)	(0.010)	(0.003)	(0.003)	(0.125)	(0.051)	(0.013)	(0.019)
SoftBART	0.193	0.129	0.035	0.045	0.330	0.184	0.045	0.056
	(0.027)	(0.014)	(0.003)	(0.004)	(0.141)	(0.039)	(0.010)	(0.012)
MOTR-BART	0.162	0.093	0.021	0.031	0.331	0.182	0.042	0.056
	(0.029)	(0.016)	(0.004)	(0.006)	(0.116)	(0.052)	(0.013)	(0.018)
SMOTR-BART	0.206	0.116	0.027	0.038	**0.303**	**0.167**	**0.038**	**0.056**
	(0.033)	(0.019)	(0.005)	(0.006)	(0.065)	(0.037)	(0.009)	(0.013)
TVP-BART	**0.101**	0.085	0.026	0.031	0.423	0.243	0.059	0.082
	(0.042)	(0.024)	(0.007)	(0.009)	(0.191)	(0.077)	(0.020)	(0.032)
TVP-SoftBART	0.226	0.186	0.056	0.067	0.630	0.347	0.085	0.103
	(0.031)	(0.024)	(0.008)	(0.010)	(0.268)	(0.120)	(0.029)	(0.031)

Note: Simulation results for DGP 2, see Equation (26). This DGP includes random walk time variation in the parameters. For each error metric, the (absolute) means over 5 simulations are shown calculated over the in-sample period ($t = 1, \ldots, 129$) and out-of-sample period ($t = 130, \ldots, 149$). Bold numbers indicate the best performance (lowest error metric). The standard deviation is reported in parentheses. See Section 3.9 for error metrics definition.

TABLE III. Forecast Evaluation for Simulations - DGP 3 (random-walk time-variation and structural break)

	In-Sample Performance				Out-of-Sample Performance			
	RMSE	avCRPS	avCRPS-T	avCRPS-L	RMSE	avCRPS	avCRPS-T	avCRPS-L
BART	0.342	0.197	0.048	0.065	0.896	0.500	0.115	0.165
	(0.082)	(0.045)	(0.009)	(0.014)	(0.113)	(0.075)	(0.020)	(0.033)
SoftBART	0.542	0.308	0.073	0.103	0.831	0.462	0.103	0.148
	(0.083)	(0.043)	(0.009)	(0.012)	(0.097)	(0.069)	(0.014)	(0.024)
MOTR-BART	0.356	0.202	0.047	0.067	0.874	0.466	0.112	0.151
	(0.060)	(0.031)	(0.007)	(0.010)	(0.120)	(0.075)	(0.019)	(0.029)
SMOTR-BART	0.571	0.317	0.072	0.102	0.911	0.512	0.114	0.169
	(0.103)	(0.061)	(0.013)	(0.017)	(0.067)	(0.045)	(0.009)	(0.019)
TVP-BART	**0.118**	**0.107**	**0.033**	**0.039**	**0.604**	**0.348**	**0.083**	**0.113**
	(0.042)	(0.029)	(0.008)	(0.010)	(0.106)	(0.037)	(0.008)	(0.010)
TVP-SoftBART	0.214	0.228	0.074	0.086	0.775	0.433	0.105	0.137
	(0.044)	(0.032)	(0.011)	(0.012)	(0.222)	(0.112)	(0.024)	(0.032)

Note: Simulation results for DGP 3, see Equation (26). This DGP includes random walk time variation in the parameters and a structural break. For each error metric, the (absolute) means over 5 simulations are shown calculated over the in-sample period ($t = 1, \ldots, 129$) and out-of-sample period ($t = 130, \ldots, 149$). Bold numbers indicate the best performance (lowest error metric). The standard deviation is reported in parentheses. See Section 3.9 for error metrics definition.

I. Only in this DGP, TVP-BART can maintain its superior performance in-sample to the out-of-sample period. This is the case for both the point and density forecasts. TVP-SoftBART captures the time series in-sample better compared to the previous DGPs, shown by the relatively low mean and standard deviation in all error metrics. This model also performs relatively well in the out-of-sample period since it obtains average error metrics that closely follow the best-performing model, TVP-BART. However, it should be noted that the uncertainty surrounding the forecasts is the largest of the models considered.

Given the absence of time variation in the first DGP, it is expected that the time-invariant models other than TVP-BART and TVP-SoftBART perform better. However, the bad performance of the latter models shows their susceptibility to overfitting. The second DGP does include some time variation and therefore it is expected that the time-varying models would perform better than the time-invariant models. However, this is not the case. In the third DGP, which extends the second DGP by adding a structural break, the time-varying models obtain the best forecasting accuracy. It is expected that these models would capture the time series the best, even though they are constructed especially for random walk variation. The reason why these models still can capture structural break time variation may be explained by the fact that a structural break parameter can be created from a sum of random walk parameters.

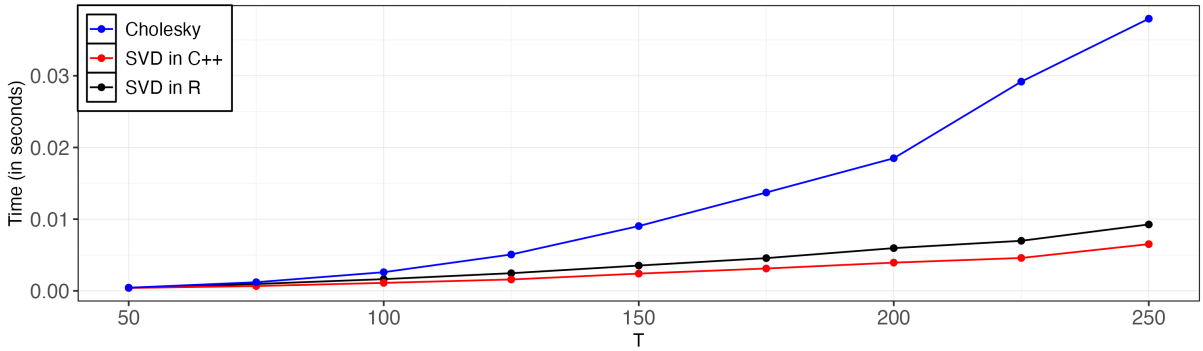
4.2 Runtime Comparison

To illustrate how the computation time changes with T, Figure II shows the computation as a function of T which ranges from 50 to 250. The lines refer to the actual time (based on Apple M3) necessary to simulate from the full conditional of the time-varying parameters in panel (a). Panel (b) displays the time it takes to estimate TVP-SoftBART once on data simulated using the second DGP of size T. The used approaches include a Cholesky decomposition of the matrix Λ_j^{-1} and a singular value decomposition (SVD) of the regressor matrix $\tilde{L}_j$ discussed in Section 3.6. To speed up the computation time even more, the parts of the code that require SVD are also programmed in C++ using Rcpp and RcppArmadillo. This includes drawing the time-varying parameters as well as the computation of the full conditional distribution of the partial residuals in the Metropolis-Hastings step which requires the posterior mean and variance of the time-varying parameters.

Figure IIa demonstrates that employing SVD accelerates computation, reducing the computation time by roughly 75% compared to using the Cholesky decomposition in R. Looking at the estimation of the time-varying parameters, the difference between the R code and C++ code does not seem that large.

However, Figure IIb displays the actual time it takes to estimate TVP-SoftBART once, so including estimating time-varying parameters for each regression tree and each MCMC iteration. It shows that SVD is again faster than Cholesky decomposition, but when the sample size increases one benefits the most from using the C++ code. This difference arises because time-varying parameters are estimated for each tree and every MCMC run, resulting in a small difference being accumulated numerous times. Additionally, the computation of the conditional likelihood of the partial residuals is also an important driver of the total time to estimate TVP-SoftBART. This is in particular caused by the computation of the posterior mean and variance of the time-varying parameters.

(a) Time to estimate time-varying parameters in a single tree



(b) Total Time to estimate TVP-SoftBART

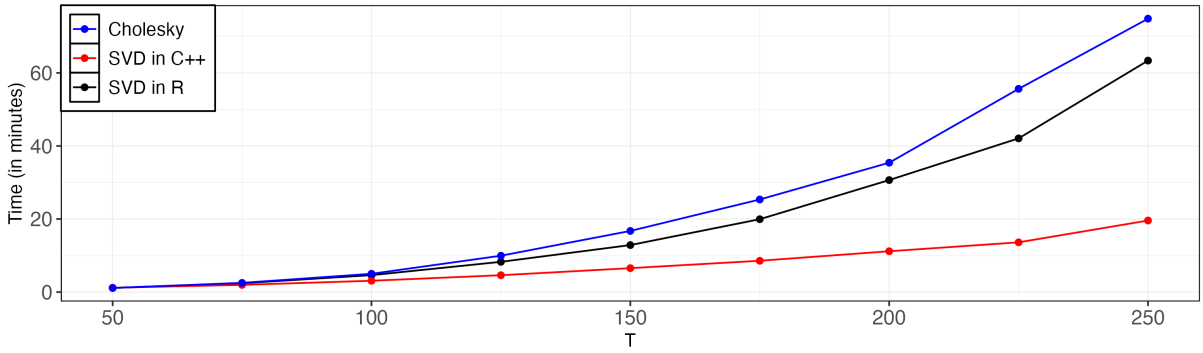


FIGURE II. The top figure shows the average time it takes to estimate the time-varying parameters in TVP-SoftBART, and the bottom figure shows the time it takes to estimate the whole model using SVD in R, C++, and using a Cholesky decomposition.

4.3 Robustness Check

In Section 4.1, it is shown that the time-varying BART framework provides the best out-of-sample forecasting accuracy among the BART models considered when time-variation not only consists of random walks but also of a structural break. While these models are constructed to work well in case of random walk time variation, it is not shown by their forecasting accuracy in the second DGP which includes this type of variation. There are a few potential reasons why this could be the case. First, the variation in the time-varying parameters may be too small such that a simple constant parameter works better since it avoids uncertainty in the estimation of the additional time-varying parameters. The structural break parameter increases the variance of the target and makes the time variation more pronounced. Therefore, in this Section the second DGP is again studied but with increased variance ($\sigma = 0.4$ instead of $\sigma = 0.1$) and parameter variance (0.1 instead of 0.015).

TABLE IV. Forecast Evaluation for Simulations - DGP 2 (random-walk time-variation + increased variances)

	In-Sample Performance				Out-of-Sample Performance			
	RMSE	avCRPS	avCRPS-T	avCRPS-L	RMSE	avCRPS	avCRPS-T	avCRPS-L
BART	0.640	0.367	0.089	0.121	1.656	0.902	0.212	0.283
	(0.089)	(0.052)	(0.014)	(0.019)	(0.236)	(0.138)	(0.034)	(0.053)
SoftBART	0.984	0.536	0.123	0.176	**1.535**	0.854	0.200	0.272
	(0.254)	(0.122)	(0.031)	(0.046)	(0.371)	(0.217)	(0.051)	(0.072)
MOTR-BART	0.684	0.391	0.089	0.127	1.562	**0.843**	**0.200**	**0.266**
	(0.131)	(0.071)	(0.015)	(0.023)	(0.366)	(0.207)	(0.050)	(0.073)
SMOTR-BART	1.165	0.630	0.146	0.201	1.664	0.919	0.213	0.290
	(0.187)	(0.098)	(0.027)	(0.040)	(0.312)	(0.201)	(0.042)	(0.076)
TVP-BART	**0.205**	**0.182**	**0.055**	**0.066**	1.593	0.901	0.211	0.280
	(0.085)	(0.068)	(0.020)	(0.024)	(0.462)	(0.226)	(0.066)	(0.072)
TVP-SoftBART	0.608	0.428	0.122	0.150	1.881	1.032	0.240	0.315
	(0.229)	(0.131)	(0.038)	(0.046)	(0.210)	(0.126)	(0.028)	(0.037)

Note: Simulation results for DGP 2, see Equation (26), but with increased error variance $\sigma = 0.4$ instead of $\sigma = 0.1$ and parameter variance (0.1 instead of 0.015). See Table II for a comparison. This DGP includes random walk time variation in the parameters. For each error metric, the (absolute) means over 5 simulations are shown calculated over the in-sample period ($t = 1, \ldots, 129$) and out-of-sample period ($t = 130, \ldots, 149$). Bold numbers indicate the best performance (lowest error metric). The standard deviation is reported in parentheses. See Section 3.9 for error metrics definition.

Table IV displays the estimation results for the DGP including random walk time variation in the parameters, but with increased error and parameter variance. The in-sample results show that TVP-BART produces the most accurate point and density forecasts among the models considered. This is not the same as in the low variance case shown in Table II. In that case, TVP-BART does not produce the most accurate density fits in-sample. Additionally, SMOTR-BART obtains the most accurate point and density forecasts out-of-sample. In this DGP, with higher variance, SoftBART obtains the lowest RMSE and thus creates the most accurate out-of-sample point forecasts. In terms of density forecasts, MOTR-BART is the most accurate. In comparison to Table II, TVP-BART is considerably closer to the best model in terms of error metrics. In the large variance case, TVP-BART obtains an average RMSE that is only 3.788% higher than the best model among the considered models. To compare, in the low variance case, TVP-BART obtained an average RMSE that was approximately 40% higher than the best model. This can also be observed for the remaining error metrics. Similarly for TVP-SoftBART, the average RMSE is 108% higher than the best model in the low variance case. In the high variance case, this is only 22.544%. Increasing the variance, and especially the time variation is indeed beneficial for the time-varying BART models in terms of forecasting accuracy.

Another potential explanation for the fact that the time-varying BART models are not the best forecasting model in the case of a DGP that includes random walk variation may be the short sample size. The relatively short sample of 149 observations means that random walk time variation does not substantially alter the parameters out-of-sample compared to their in-sample means. In addition, the time-varying models may benefit from a larger sample, since it has more observations to learn the time-variation. The second DGP is again considered, which includes random walk time variation in the parameters and is extended to include a larger error and parameter variance as in the previous extension. In contrast to the previous extension, the sample size is increased from $T = 150$ to $T = 300$. However, increasing the sample size is costly, as shown in Figure II. Therefore, only one of the five simulations in the previous DGP is considered. This simulation is the one with the largest observed out-of-sample error metrics for TVP-BART.

Table V displays the (absolute) error metrics of one simulation of the second DGP. The top panel shows the results of one of the simulations considered in Table IV, and the bottom panel contains the results when the sample size increased to $T = 300$. Similar to the average results displayed in Table IV TVP-BART obtains the lowest in-sample error metrics when considering only this particular simulation realization as shown in the top panel. TVP-BART maintains the lowest in-sample error metrics when

TABLE V. Forecast Evaluation for Simulations - DGP 2 (random-walk time-variation + increased variances + more time observations)

	In-Sample Performance				Out-of-Sample Performance			
	RMSE	avCRPS	avCRPS-T	avCRPS-L	RMSE	avCRPS	avCRPS-T	avCRPS-L
T = 150								
BART	0.514	0.294	0.071	0.097	1.956	1.036	0.250	0.295
SoftBART	0.842	0.451	0.105	0.145	2.070	1.124	0.266	0.333
MOTR-BART	0.540	0.310	0.072	0.102	2.044	1.036	0.254	0.291
SMOTR-BART	1.083	0.548	0.127	0.171	1.997	1.042	0.253	0.284
TVP-BART	**0.153**	**0.139**	**0.043**	**0.051**	2.413	1.301	0.328	0.407
TVP-SoftBART	0.643	0.358	0.094	0.120	**1.938**	**0.989**	**0.237**	**0.287**
T = 300								
BART	0.814	0.352	0.083	0.120	2.253	1.261	0.311	0.487
SoftBART	0.853	0.479	0.107	0.155	2.190	1.232	0.300	0.466
MOTR-BART	0.776	0.436	0.100	0.141	1.964	1.135	0.265	0.415
SMOTR-BART	1.147	0.593	0.138	0.194	2.195	1.245	0.296	0.461
TVP-BART	**0.215**	**0.171**	**0.050**	**0.061**	**1.476**	**0.823**	**0.182**	**0.263**
TVP-SoftBART	0.957	0.478	0.127	0.169	2.283	1.213	0.305	0.393

Note: Simulation results for DGP 2, see Equation (26), but with increased error variance $\sigma = 0.4$ instead of $\sigma = 0.1$ and parameter variance (0.1 instead of 0.015) and more time observations ($T = 300$ instead of $T = 150$). See Table IV for a comparison. This DGP includes random walk time variation in the parameters. For each error metric, the (absolute) value is shown which is calculated over the in-sample period ($t = 1, \ldots, 279$) and out-of-sample period ($t = 279, \ldots, 299$). This simulation is one of the five simulations considered in Table IV which resulted in the worst out-of-sample performance for TVP-BART. (All simulations would be computationally infeasible). Bold numbers indicate the best performance (lowest error metric). See Section 3.9 for error metrics definition.

the sample size is increased. The out-of-sample performance is the most interesting in this case. In this simulation, TVP-BART performed the worst in the small sample case. However, when the sample size is increased TVP-BART performs by far the best. TVP-SoftBART performed worse when the sample size was increased relative to the other models. Therefore, it would be interesting to investigate also the other simulation runs to see whether this holds more generally.

To conclude this Section, if computational resources were less constrained, it would be worthwhile to explore whether the high standard deviations observed in TVP-SoftBART are specific to these five simulations or represent a general trend. Additionally, given unlimited computation time, it would be interesting to examine whether the time-varying models would better capture out-of-sample time variation with more post-burn-in iterations. In addition, the relatively short sample period means that random walk time variation does not greatly alter the parameters out-of-sample compared to their in-sample means, which allows models without time variation to perform particularly well. Increasing the sample size in one simulation run showed that the performance of TVP-BART improved, while TVP-SoftBART performed worse. It would be interesting to see if this holds more generally if more simulations are performed. In addition, increasing the error variance and parameter variance improved the forecasting performance of the time-varying BART models.

Despite the limitations in the simulation set-up due to computational feasibility, it provides valuable insights. Specifically, the time-varying models, TVP-BART and TVP-SoftBART, exhibit exceptional performance when the time-variation is more pronounced, such as in the case of a structural break in a parameter or larger error variances. Furthermore, while TVP-BART achieves the lowest mean RMSE across all DGPs in-sample, it does not maintain this performance in the out-of-sample period, indicating a potential susceptibility to overfitting.

5 Empirical Application: Forecasting Inflation

In this Section, the proposed BART methodology is applied to forecast US inflation and, by extending the illustrative simulation example, it is shown how the time-varying BART methodology may improve upon existing BART-based methods. First, the data is introduced, and then the focus is on evaluating both the point and density forecasts using the error metrics given in Section 3.9.

5.1 Data

The data is from FRED-QD (McCracken & Ng 2020) from 1969:Q1 to 2022:Q4 (216 observations).[7] Similarly to previous work, see for example Giannone et al. (2015), a moderately sized dataset is used which includes an array of key macro indicators that are commonly viewed as potential predictors of inflation. In particular, the dataset that is used is the moderately sized dataset that is used in Clark et al. (2024). Appendix B provides a complete overview of the dataset and associated transformations. The focus is on forecasting quarterly CPI inflation expressed in an annual rate, commonly measured as $(400/h)\ln(P_{t+h}/P_t)$ at forecasting horizon $h = 1, 4$.[8] Other studies assume that inflation is an $I(1)$ process and the first difference of inflation is often taken to make it stationary. However, following Groen et al. (2013), inflation is not transformed since we want the models to capture any time-variation in the mean which may disappear when taking the first difference. The time series of CPI inflation are shown in Figure III, including a dashed line indicating the start of the out-of-sample period at 2017Q1. Due to the transformation, the series for $h = 4$ is smoother and more persistent than for $h = 1$. There is a large literature that documents the time-varying properties of inflation, which are also visible in Figure III. For example, inflation peaked in 1974-1975 and around 1980 and the high levels of inflation declined slowly indicating an increased persistence, which disappeared after 1982-19833 (Cogley & Sargent 2005). Additionally, there is widespread evidence that the variability of inflation decreased from the late 1980s and early 1990s onwards as a result of exogenous variance breaks and breaks in the mean and/or persistence (Sims & Zha 2006). Also, more recent events, like the financial crisis of 2008-2009 and the COVID-19 pandemic of 2020-2021, resulted in exogenous breaks in the mean and/or persistence.

The methods that are used to forecast inflation for both forecasting horizons are the random walk model (RW), BART, SoftBART, MOTR-BART, SMOTR-BART, TVP-BART and TVP-SoftBART. The first among these models is traditionally seen as a hard-to-beat model when it comes to out-of-sample inflation forecasts (Atkeson et al. 2001). This model is also frequently referred to as a naive model, which predicts that inflation over the next h quarters is expected to be equal to inflation over the previous h quarters. Similar to the simulation set-up, an expanding window is used to forecast the period 2017:Q1-2022:Q4 (24 observations) where each model is trained using ten trees, a burn-in sample of 1000 and a post-burn-in of 2500. Forecasts are again evaluated using the root mean squared error (RMSE), the average continuous ranked probability score (avCRPS), and the quantile weighted continuous ranked probability score, with weights on the tails (avCRPS-T) and on the left tail only (avCRPS-L). Section 3.9 contains a concise overview of these error metrics. Ratios of the RMSE, avCRPS, avCRPS-T and avCRPS-L measures are reported relative to those of the RW model. A ratio smaller than 1 indicates that a model generates a more precise point forecast for the RMSE, and more precise forecasts of certain parts of the (unknown) distribution of future inflation rates for the remaining measures compared to the RW.

[7] Similar to a large stream of literature investigating the time-varying properties of inflation, quarterly data is studied, for example Primiceri (2005), Groen et al. (2013), Clark et al. (2024). In addition, using monthly data would substantially increase the number of observations. This likely benefits the forecasting performance of the time-varying BART models, but it would drastically increase the computation time.

[8] Note that we follow Clark et al. (2024) in modelling annual inflation, $y_{t+4} = (100)\ln(P_{t+4}/P_t)$ and not annualized quarter-on-quarter growth rates, $y_{t+4} = (400)\ln(P_{t+4}/P_{t+3})$. Therefore error metrics can be lower for higher-order forecasts. Usually, this cannot be the case, because it involves more uncertainty.

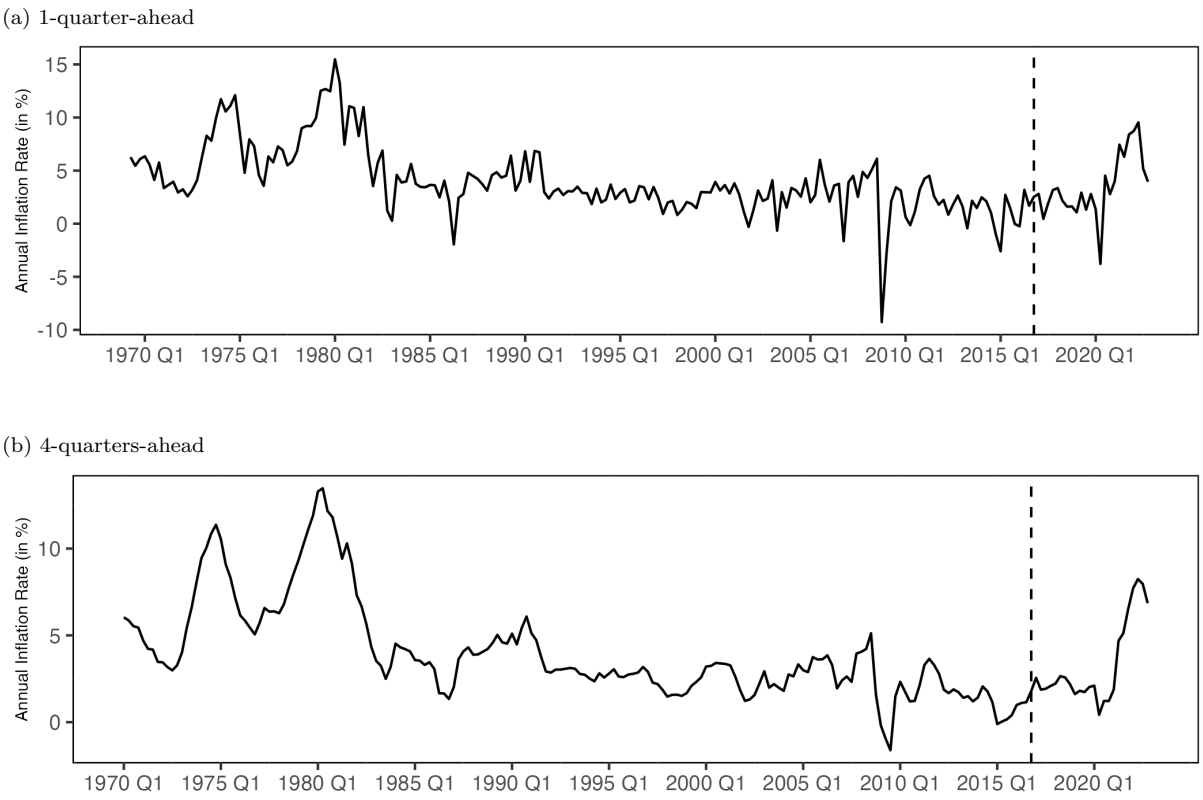


FIGURE III. CPI inflation over the entire sample, where the dotted line indicates the start of the hold-out period. (a) 1-quarter-ahead inflation and (b) 4-quarters ahead inflation.

A number of the employed models are nested and an expanding window is used which will impact the distribution of the Diebold-Mariano statistic as in Diebold & Mariano (2002) for both point and density forecasts. Instead, the Harvey et al. (1997) small sample correction of the Diebold & Mariano (2002) statistic with standard normal critical values is used, which is shown to result in good sized test of the null hypothesis of equal finite-sample forecast accuracy for both nested and nonnested models, including cases with expanded window-based model updating; see, for example Clark & McCracken (2013). A more elaborate investigation of the size of this test, and a comparison against more advanced tests are beyond the scope of this thesis, but instead, the existing literature is followed, for example Groen et al. (2013) and Clark et al. (2024). The null of equal finite-sample forecast precision based on either the RMSE, avCRPS, avCRPS-T or avCRPS-L measure is tested against the alternative that a model outperforms the RW benchmark.

5.2 In-Sample Results

Although the main focus of the analysis of the empirical example is on assessing the out-of-sample forecasting performance of the BART methodology, we first briefly discuss in this subsection the in-sample properties of the BART-based models when they are estimated on the in-sample period, running from 1969:Q2 to 2016:Q4 for $h = 1$ and from 1970:Q1 to 2016:Q4 for $h = 4$. Table VI reports ratios of the evaluation measures for the BART-based models relative to measures produced by the RW for forecasting inflation one quarter and four quarters ahead. The RW error metrics are absolute numbers.

First, we consider the point forecast evaluation for $h = 1$ displayed in the left panel of Table VI. It appears that all models outperform the RW and are statistically significant. This is not a surprising result since the RW is known to perform well out-of-sample and not necessarily in-sample. In contrast to

the BART-based models, the RW uses lagged inflation as a forecast and cannot adjust the fit. Although all models perform favourably compared to the RW, there are some differences between the models with TVP-BART performing the best by obtaining an RMSE that is five times smaller than the RMSE of the RW, and SMOTR-BART the worst. For the density forecast similar conclusions apply, since all models significantly outperform the RW specification and TVP-BART performs the best closely followed by BART and SMOTR-BART does the worst in capturing the density.

The left panel of Table VI shows the relative error metrics for forecasting horizon $h = 4$. Furthermore, all models outperform the RW, and TVP-BART generates the most accurate point and density fits in this in-sample period. However, in this case, TVP-SoftBART closely follows TVP-BART in creating accurate fits and it performs very similar to BART when focusing on the density forecast evaluation. Again, all models seem to overfit relative to SMOTR-BART. Based on all error metrics, allowing for time variation is favourable in modelling inflation both one and four quarters ahead. Although TVP-BART is the best-performing model in-sample, it takes drastically more computation time than the standard BART-based methodology.[9]

TABLE VI. In-sample evaluation for forecasting US inflation

	1-quarter-ahead forecast (h=1)				4-quarters-ahead forecast (h=4)			
	RMSE	avCRPS	avCRPS-T	avCRPS-L	RMSE	avCRPS	avCRPS-T	avCRPS-L
RW	2.250	1.209	0.278	0.384	1.944	1.119	0.251	0.349
BART	0.314***	0.345***	0.381***	0.371***	0.185***	0.203***	0.242***	0.213***
SoftBART	0.523***	0.533***	0.533***	0.544***	0.384***	0.380***	0.398***	0.393***
MOTR-BART	0.492***	0.510***	0.523***	0.528***	0.272***	0.274***	0.303***	0.286***
SMOTR-BART	0.714***	0.697***	0.717***	0.739***	0.629***	0.607***	0.625***	0.600***
TVP-BART	0.202***	0.276***	0.339***	0.310***	0.064***	0.143***	0.211***	0.174***
TVP-SoftBART	0.517***	0.547***	0.614***	0.611***	0.075***	0.192***	0.293***	0.238***

Note: The error metrics are computed after estimating the model once over the sample period 1969:Q2-2016:Q4 for $h = 1$ and 1970:Q1-2016:Q4 for $h = 4$. The numbers are ratios of error metrics relative to the random walk (RW) for which absolute numbers are displayed. Asterixs that are shown correspond to Diebold & Mariano (2002) test with Harvey et al. (1997) correction for the null hypothesis of equal finite-sample forecasting accuracy versus the alternative hypothesis that a model outperforms RW for either of these measure, where *,** and *** indicate rejection of this null at the 10%, 5% and 1% levels, respectively, based on one-sided standard normal critical values.

5.3 Out-of-Sample Results

Table VII contains error metrics relative to the error metrics of the RW computed over the out-of-sample period, computed both over the entire period from 2017:Q1 to 2022:Q4 as well as over the pandemic and post-pandemic subperiod (2020:Q1-2022:Q4). The RW error metrics are shown in absolute numbers, and numbers in bold indicate a better performance than the RW based on the error metric indicated in the column. The last evaluation samples span the pandemic and post-pandemic period which could have caused time variation in the dynamics of inflation rates. Panel A displays the estimation results for forecasting horizon $h = 1$, and panel B for forecasting horizon $h = 4$.

First, we consider the point forecasts for 1-quarter-ahead inflation forecasts as displayed in Panel A of Table VII for the out-of-sample period 2017:Q1-2022:Q4. BART and SoftBART perform relatively well compared to RW although not significantly better. TVP-BART and TVP-SoftBART follow in performance but perform slightly worse than the RW over this sample period. Lastly, MOTR-BART and SMOTR-BART generate the worst point forecast, especially SMOTR-BART which attains almost twice as high RMSE compared to the RW. This may be because MOTR-BART and SMOTR-BART use a linear predictor in each terminal node, while inflation may be more accurately described by fewer and possibly

[9]Estimating TVP-BART only once ranges from 6.599 to 8.080 minutes and TVP-SoftBART takes approximately 10.388 to 13.395 minutes on an Apple M3. Due to the expanding window of 20 out-of-sample observations, total computation time is approximately 2.5 hours for TVP-BART and 4 hours for TVP-SoftBART.

only lagged inflation. In addition, the regressor variables attain unusual values during the pandemic period, and in combination with coefficients estimated in a more stable time before, this may result in extreme forecasts. Next, when considering the forecast of the entire density by the avCRPS measure, the Table shows that only SoftBART can generate a more accurate forecast than the RW but again not significant. It is closely followed by BART, TVP-BART and TVP-SoftBART which all perform slightly worse than the RW. MOTR-BART and SMOTR-BART again do the worst job in terms of avCRPS. Focusing only on the tails of the density forecast leads to similar conclusions. Evaluating only the left tail forecast, as is done by avCRPS-L, shows that SoftBART is statistically more accurate than the RW, and TVP-BART generates a lower error metric than the RW although not significant.

The right side of Panel A in Table VII reports the relative error metrics over the sub-sample 2020:Q1-2022:Q4. In general, the same conclusions apply for this subsample period as for the complete hold-out period. It should be noted that the error metrics of the RW in this subsample period are generally higher than in the larger out-of-sample period as reported in the left side of Panel A. Therefore the RW has generally more difficulty capturing inflation in this subsample period. This is due to the high variability of inflation in this subsample period, while the random walk uses the inflation realizations of the period starting before the pandemic period as inflation forecasts.

TABLE VII. Forecast evaluation for forecasting US inflation

	2017:Q1-2022:Q4				2020:Q1-2022:Q4			
	RMSE	avCRPS	avCRPS-T	avCRPS-L	RMSE	avCRPS	avCRPS-T	avCRPS-L
Panel A: 1-quarter-ahead forecast (h=1)								
RW	2.590	1.413	0.327	0.422	3.433	1.969	0.461	0.579
BART	**0.993**	1.022	1.015	1.095	**0.997**	1.096	1.110	1.154
SoftBART	**0.803**	**0.829**	**0.829**	**0.812***	**0.778**	**0.828**	**0.847**	**0.782**
MOTR-BART	1.355	1.134	1.148	1.265	1.388	1.216	1.254	1.391
SMOTR-BART	2.046	1.234	1.140	1.347	2.148	1.387	1.257	1.518
TVP-BART	1.074	1.010	1.015	**0.998**	1.088	1.085	1.097	1.029
TVP-SoftBART	1.089	1.051	1.041	1.121	1.094	1.107	1.093	1.165
Panel B: 4-quarters-ahead forecast (h=4)								
RW	2.317	1.285	0.311	0.358	3.188	1.948	0.464	0.512
BART	**0.656**	**0.612***	**0.616****	**0.673**	**0.638**	**0.600***	**0.630***	**0.655**
SoftBART	**0.749**	**0.708***	**0.688****	**0.748**	**0.696***	**0.628****	**0.653****	**0.632***
MOTR-BART	1.117	**0.938**	**0.905**	**0.942**	1.108	**0.943**	**0.954**	**0.937**
SMOTR-BART	1.066	**0.886**	**0.870**	1.092	1.046	**0.881**	**0.877**	1.149
TVP-BART	**0.647***	**0.550****	**0.523****	**0.615****	**0.645**	**0.581***	**0.549***	**0.692**
TVP-SoftBART	**0.465***	**0.424****	**0.431****	**0.485****	**0.467**	**0.431***	**0.417***	**0.494**

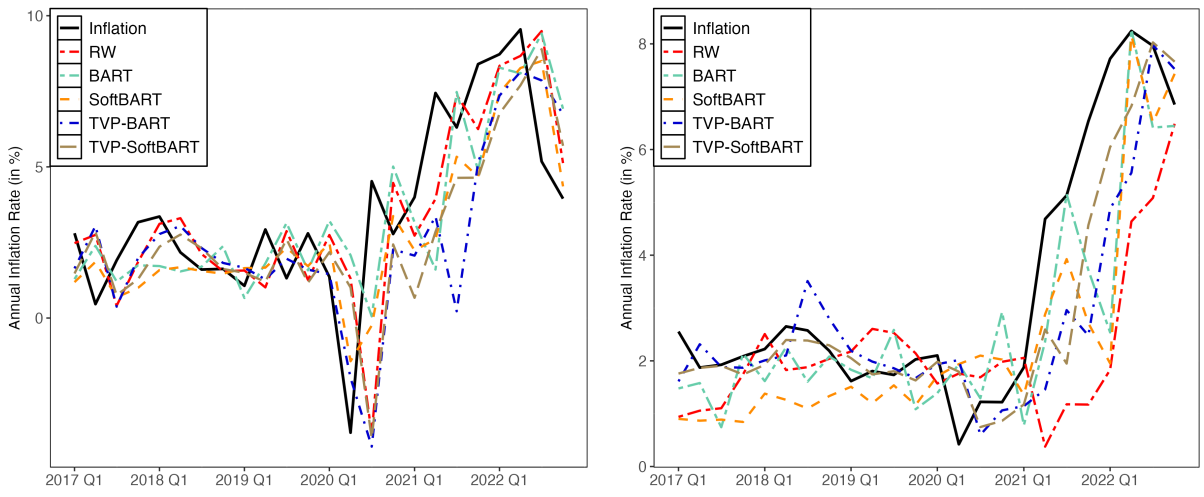
Note: The numbers are ratios of error metrics relative to the random walk (RW) for which absolute numbers are displayed. Bold numbers indicate a better performance than the RW. Asterixs that are shown correspond to Diebold & Mariano (2002) test with Harvey et al. (1997) correction for the null hypothesis of equal finite-sample forecasting accuracy versus the alternative hypothesis that a model outperforms RW for either of these measure, where *,** and *** indicate rejection of this null at the 10%, 5% and 1% levels, respectively, based on one-sided standard normal critical values. 1-quarter ahead forecast in Panel A, and 4-quarters ahead in Panel B. The first four columns of error metrics are computed over the complete hold-out period (2017Q1-2022Q4), while the last four columns only cover the pandemic and post-pandemic period (2020Q1-2022Q4).

Next, Panel B contains the relative evaluation metrics for the 4-quarters-ahead inflation forecast. First, we consider the point forecasts. TVP-SoftBART attains the lowest RMSE and is statistically different from the RMSE of the RW. It is followed by TVP-BART which also generates an RMSE statistically lower than the RMSE of the RW. The RMSE of TVP-BART is closely followed by the RMSE of BART and SoftBART, although lower than the RMSE of the benchmark, not significant. MOTR-BART and SMOTR-BART perform the worst. However, in contrast to $h = 1$, MOTR-BART generates the worst point forecast according to the RMSE and not SMOTR-BART, which performs almost the same as the RW. In terms of density forecasts, there is evidence based on the avCRPS measure that all BART-based models are producing more precise density forecasts than the RW model. Especially TVP-

BART and TVP-SoftBART generate accurate density forecasts shown by their low avCRPS value and statistical significance. This superior performance is also apparent when considering only the tails of the density forecast, but also when considering only the left tail. BART and SoftBART also generate accurate point and density forecasts, but based on the error metrics, they are slightly worse than TVP-BART and TVP-SoftBART. When focusing only on the left tail of predicted density BART and SoftBART obtain a low value of avCRPS-L, but in contrast to TVP-BART and TVP-SoftBART they are not able to beat the RW statistically.

The right-hand side of Panel B in Table VII reports on the forecast evaluation for 4-quarters-ahead inflation over the 2020:Q1-2022:Q4. Over this period, the error metrics are generally higher than over the complete hold-out period. Due to the large variability, the inflation in this period is harder to forecast. While the RW error metrics are higher, the performance of the BART-based models relative to the RW is similar to the period before. Again, TVP-BART and TVP-SoftBART outperform the RW in terms of both point forecast accuracy and density forecast accuracy.

To understand the performance of the models out-of-sample better, Figure IV contains the realization of inflation and the forecasts by the RW, BART, SoftBART, TVP-BART and TVP-SoftBART for $h = 1$ in the left figure and $h = 4$ in the right figure (MOTR-BART and SMOTR-BART are left out due to their worse performance). First, we consider the left panel of Figure IV which shows that the BART-based model forecasts roughly follow the RW forecast. It can be noted that the forecasts produced by BART and SoftBART are more flat, especially apparent in 2020:Q3. In addition, the path that the TVP-SoftBART follows is very similar to the RW forecast only its level is lower than the RW forecast. TVP-BART, while similar to TVP-SoftBART, makes a big miss in the middle of 2021. The right panel shows the inflation realization and forecasts using the forecast horizon $h = 4$. While TVP-SoftBART seems again smoothed out, BART is more spiky in this case. The forecast by TVP-SoftBART tracks the inflation realization closely and appears to be the most accurate of all model forecasts. The TVP-BART forecast seems to be very similar to the forecast by TVP-SoftBART but makes a big miss in the middle of 2018. This may be due to the underidentification of the parameter estimates for example. Appendix C contains additional figures containing the inflation realization, the median, and a prediction interval for BART, SoftBART, TVP-BART and TVP-SoftBART.



(a) 1-quarter-ahead

(b) 4-quarters-ahead

FIGURE IV. Inflation forecasts

In general, the BART-based methodology benefits from time-variation in-sample, but for small forecast horizons, $h = 1$, fails to maintain this out-of-sample. This may indicate that this model has overfitted the in-sample period and therefore it may be useful to focus more on tuning this model to obtain a better

in-sample and out-of-sample balance. However, it should be noted that most of the BART extensions fail to beat the RW forecast for this forecasting horizon, including the time-varying BART models. When considering a larger forecasting horizon, $h = 4$, TVP-BART and TVP-SoftBART succeed in producing the most accurate point and density forecast and based on various error metrics statistically outperform the RW. The better forecasting performance of the models for this horizon may be due to the smoother nature of the variable to be forecasted. This variable is more persistent, more closely related to its previous value, but has a lower variance than for $h = 1$. It should be noted that the time-invariant terminal node models, BART and SoftBART, especially the last one, are very close in terms of error metrics to the time-varying models. Therefore it may be questionable whether the additional accuracy of the time-varying models weighs up to the computational burden.

6 Conclusion

In conclusion, this paper addresses the limitations of traditional BART models in the context of macroeconomic forecasting by introducing time-varying parameter BART models, namely TVP-BART and TVP-SoftBART. These models extend the capabilities of existing BART methodologies by incorporating time variation in the parameters of the terminal nodes, thus improving the model's ability to capture dynamic changes in economic relationships over time. In addition, this paper provides an elaborate overview of BART and its extension tailored to a time series context. BART models the conditional mean as a constant by fitting multiple trees that explain only a small part of the univariate response. The extensions include SoftBART, MOTR-BART and a combination of these, SMOTR-BART. These models form the foundation for the newly proposed models, TVP-BART and TVP-SoftBART. To allow for time-variation in the terminal node of each tree, the conditional mean is modelled by a time-varying parameter, rather than a time-invariant parameter as in BART or a linear part as in MOTR-BART. The proposed methods maintain computational efficiency and scalability by recasting the time-varying parameter model as a static regression problem, avoiding the common pitfalls of overparametrization. These static terminal node regression models resemble MOTR-BART where the conditional mean is fitted using a linear predictor. In addition, soft splitting is introduced in this model to avoid under-identification of parameters because some time observations may be missing in a terminal node.

The performance of the BART-based methodology is assessed in the context of synthetic data, which includes three different nested data-generating processes (DGPs). The three DGPs include parameters with (i) no time-variation, (ii) random walk time variation and (iii) random walk time variation and a structural break. Allowing for time variation in the conditional mean of BART is especially beneficial in terms of forecasting accuracy in the third DGP. When time-variation is non-existent, or parameters change only gradually, as in the first and second DGP respectively, TVP-BART is susceptible to overfitting the training data and fails to maintain this accuracy on new testing data. Furthermore, the conditional mean in the time-varying BART models is modelled using random walk variation. A structural break parameter can still be captured by these models, since the sum of random walk parameters may adequately model a structural break parameter. It is, however, expected that these models generate the most accurate forecasts in the second DGP. Results show that this is not the case and therefore this DGP is further investigated. This includes increasing the error and parameter variances as well as increasing the sample size. The time-varying BART models benefit from increasing these variances in terms of out-of-sample forecasting accuracy. Since increasing the sample size is computationally expensive, only one simulation realization of the second DGP is investigated. This simulation run shows that TVP-BART performs better when the sample size is doubled.

In addition, the BART-based methodology is applied to forecasting quarterly US inflation by evaluating again both the point and density forecasts. The results show that for the small forecast horizon, namely one-quarter ahead forecasts, all BART-based models fail to outperform the random walk bench-

mark significantly. Only SoftBART attains lower error metrics than the random walk over the complete out-of-sample period, from 2017:Q1 to 2022:Q4, as well as in the pandemic and post-pandemic period, 2020:Q1-2022:Q4. For the larger forecast horizon, namely the four-quarters-ahead forecasts, BART, SoftBART, TVP-BART and TVP-SoftBART outperform the random walk. In particular, TVP-SoftBART excels in both point and density forecasting accuracy in both sub-samples. Similarly to the simulation exercise, TVP-BART obtains low error metrics in-sample but fails to maintain this out-of-sample for the small forecasting horizon. This is again an indication that this model may be susceptible to overfitting.

In general, adding time-variation in BART is beneficial in terms of forecasting accuracy if time-variation is more pronounced. This is evident from the simulation exercise where the performance of TVP-BART and TVP-SoftBART was the best when the DGP allowed for random walk time variation and a structural break in the parameters. For the empirical example, it can be noted that the TVP-BART and TVP-SoftBART particularly excel on longer forecasting horizons, when the inflation realizations to be forecasted are more persistent. However, it should be noted that the time-varying BART models are closely followed by the time-invariant BART and SoftBART in both the simulation and the empirical exercise in terms of error metrics. However, TVP-BART and TVP-SoftBART take considerably more computation time than BART and SoftBART and therefore there exists a trade-off between forecasting accuracy and computation time.

Various attempts have been made in the literature to introduce time-variation in BART. However, TVP-BART and TVP-SoftBART form the first among them to include time-variation in the terminal nodes of each tree. Yet, the main limitation is the computation time. The expanding window estimation that is employed requires estimating each model again when a new observation enters and therefore is a major driver of the computation time. Therefore, a potential research direction could be in the direction of Sequential Markov Chains to limit computation time. For example, the sequential tree model of Taddy et al. (2011) where the model state changes in time with the accumulation of new data may be particularly useful.

In addition, an interesting research direction would be the influence of the sample size. The time-varying BART models does not perform the best in the simulation exercise where random walk variation was introduced in the parameters. Increasing the sample size may however alter this finding. A first attempt in this direction shows that increasing the sample size benefited TVP-BART in out-of-sample forecasting accuracy. However, more simulations are necessary to generalize this finding. It may also be interesting to investigate the effect of the persistence in a time series on the performance of the time-varying BART models. In addition, changing the prior specifications and investigating the influence on the forecasting accuracy may be an interesting extension. Another future research direction is the addition of heteroskedasticity into the time-varying BART models. In the proposed framework, the error variance is assumed to be time-invariant. However, real-world data do not always follow the simple constant-variance process modelled by BART. Complementing the proposed BART-based methodology with stochastic volatility on the errors can prevent the detection of spurious variations in the time-varying coefficients by capturing some of the variability in the error term. Stochastic volatility processes have attained considerable popularity, with one of the most commonly used being the state space model introduced by Kim et al. (1998), which assumes an autoregressive process for log volatility. Despite the improved forecasting accuracy that the inclusion of stochastic volatility brings to econometric models, its estimation poses challenges due to the intractability of the likelihood function. Various efforts have been made to enhance the efficiency of the Markov Chain Monte Carlo (MCMC) sampler. While it may be expected that the addition of this feature increases the forecasting accuracy, it is also an additional computationally expensive aspect.

References

Atkeson, A., Ohanian, L. E. et al. (2001), 'Are phillips curves useful for forecasting inflation?', *Federal Reserve bank of Minneapolis quarterly review* **25**(1), 2–11.

Belmonte, M. A., Koop, G. & Korobilis, D. (2014), 'Hierarchical shrinkage in time-varying parameter models', *Journal of Forecasting* **33**(1), 80–94.

Breiman, L. (1996), 'Bagging predictors', *Machine learning* **24**, 123–140.

Breiman, L. (2001), 'Random forests', *Machine learning* **45**, 5–32.

Chipman, H. A., George, E. I. & McCulloch, R. E. (2010), 'Bart: Bayesian additive regression trees', *The Annals of Applied Statistics* **4**(1), 266–298.

Clark, T. E., Huber, F., Koop, G. & Marcellino, M. (2024), 'Forecasting us inflation using bayesian nonparametric models', *The Annals of Applied Statistics* **18**(2), 1421–1444.

Clark, T. E., Huber, F., Koop, G., Marcellino, M. & Pfarrhofer, M. (2023), 'Tail forecasting with multivariate bayesian additive regression trees', *International Economic Review* **64**(3), 979–1022.

Clark, T. & McCracken, M. (2013), 'Advances in forecast evaluation', *Handbook of economic forecasting* **2**, 1107–1201.

Cogley, T. & Sargent, T. J. (2005), 'Drifts and volatilities: monetary policies and outcomes in the post wwii us', *Review of Economic dynamics* **8**(2), 262–302.

D'Agostino, A., Gambetti, L. & Giannone, D. (2013), 'Macroeconomic forecasting and structural change', *Journal of applied econometrics* **28**(1), 82–101.

Deshpande, S. K., Bai, R., Balocchi, C., Starling, J. E. & Weiss, J. (2020), 'Vcbart: Bayesian trees for varying coefficients', *arXiv preprint arXiv:2003.06416* .

Diebold, F. X. & Mariano, R. S. (2002), 'Comparing predictive accuracy', *Journal of Business & economic statistics* **20**(1), 134–144.

El Yaakoubi, A. (2022), Smoothed and local-linear extensions to bayesian additive regression trees for econometric forecasting, Master's thesis, Erasmus University Rotterdam.

Friedberg, R., Tibshirani, J., Athey, S. & Wager, S. (2020), 'Local linear forests', *Journal of Computational and Graphical Statistics* **30**(2), 503–517.

Friedman, J. H. (1991), 'Multivariate adaptive regression splines', *The annals of statistics* **19**(1), 1–67.

Friedman, J. H. (2001), 'Greedy function approximation: a gradient boosting machine', *Annals of statistics* pp. 1189–1232.

Giannone, D., Lenza, M. & Primiceri, G. E. (2015), 'Prior selection for vector autoregressions', *Review of Economics and Statistics* **97**(2), 436–451.

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

Gneiting, T. & Ranjan, R. (2011), 'Comparing density forecasts using threshold-and quantile-weighted scoring rules', *Journal of Business & Economic Statistics* **29**(3), 411–422.

Goulet Coulombe, P. (2020), 'The macroeconomy as a random forest', *Available at SSRN 3633110* .

Groen, J. J., Paap, R. & Ravazzolo, F. (2013), 'Real-time inflation forecasting in a changing world', *Journal of Business & Economic Statistics* **31**(1), 29–44.

Harvey, D., Leybourne, S. & Newbold, P. (1997), 'Testing the equality of prediction mean squared errors', *International Journal of forecasting* **13**(2), 281–291.

Hauzenberger, N., Huber, F., Koop, G. & Mitchell, J. (2022a), 'Bayesian modeling of time-varying parameters using regression trees', *arXiv preprint arXiv:2209.11970* .

Hauzenberger, N., Huber, F., Koop, G. & Onorante, L. (2022b), 'Fast and flexible bayesian inference in time-varying parameter regression models', *Journal of Business & Economic Statistics* **40**(4), 1904–1918.

Hauzenberger, N., Huber, F., Marcellino, M. & Petz, N. (2024), 'Gaussian process vector autoregressions and macroeconomic uncertainty', *Journal of Business & Economic Statistics* (just-accepted), 1–27.

Huber, F., Koop, G. & Onorante, L. (2021), 'Inducing sparsity and shrinkage in time-varying parameter models', *Journal of Business & Economic Statistics* **39**(3), 669–683.

Huber, F., Koop, G., Onorante, L., Pfarrhofer, M. & Schreiner, J. (2023), 'Nowcasting in a pandemic using non-parametric mixed frequency vars', *Journal of Econometrics* **232**(1), 52–69.

Huber, F. & Rossini, L. (2022), 'Inference in bayesian additive vector autoregressive tree models', *The Annals of Applied Statistics* **16**(1), 104–123.

Kapelner, A. & Bleich, J. (2013), 'Bartmachine: Machine learning with bayesian additive regression trees', *arXiv preprint arXiv:1312.2171* .

Kim, S., Shephard, N. & Chib, S. (1998), 'Stochastic volatility: likelihood inference and comparison with arch models', *The review of economic studies* **65**(3), 361–393.

Koop, G. & Korobilis, D. (2013), 'Large time-varying parameter vars', *Journal of Econometrics* **177**(2), 185–198.

Linero, A. R. & Yang, Y. (2018), 'Bayesian regression tree ensembles that adapt to smoothness and sparsity', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **80**(5), 1087–1110.

Masini, R. P., Medeiros, M. C. & Mendes, E. F. (2023), 'Machine learning advances for time series forecasting', *Journal of economic surveys* **37**(1), 76–111.

McCracken, M. & Ng, S. (2020), Fred-qd: A quarterly database for macroeconomic research, Technical report, National Bureau of Economic Research.

Pettenuzzo, D. & Timmermann, A. (2017), 'Forecasting macroeconomic variables under model instability', *Journal of business & economic statistics* **35**(2), 183–201.

Prado, E. B., Moral, R. A. & Parnell, A. C. (2021a), 'Bayesian additive regression trees with model trees', *Statistics and Computing* **31**(3), 1–13.

Prado, E. B., Parnell, A. C., Murphy, K., McJames, N., O'Shea, A. & Moral, R. A. (2021b), 'Accounting for shared covariates in semi-parametric bayesian additive regression trees', *arXiv preprint arXiv:2108.07636* .

Primiceri, G. E. (2005), 'Time varying structural vector autoregressions and monetary policy', *The Review of Economic Studies* **72**(3), 821–852.

Sims, C. A. (1980), 'Macroeconomics and reality', *Econometrica: journal of the Econometric Society* pp. 1–48.

Sims, C. A. & Zha, T. (2006), 'Were there regime switches in us monetary policy?', *American Economic Review* **96**(1), 54–81.

Taddy, M. A., Gramacy, R. B. & Polson, N. G. (2011), 'Dynamic trees for learning and design', *Journal of the American Statistical Association* **106**(493), 109–123.

Trippe, B., Huggins, J., Agrawal, R. & Broderick, T. (2019), Lr-glm: High-dimensional bayesian inference using low-rank data approximations, *in* 'International Conference on Machine Learning', PMLR, pp. 6315–6324.

Zeldow, B., Re III, V. L. & Roy, J. (2019), 'A semiparametric modelling approach using bayesian additive regression trees with an application to evaluate heterogeneous treatment effects', *The annals of applied statistics* **13**(3), 1989.

A Extra Results

TABLE VIII. Forecast Evaluation for Simulations - DGP 3 (*normal_samples_per_gibbs_sampler* set to 1)

	In-Sample Performance				Out-of-Sample Performance			
	RMSE	avCRPS	avCRPS-T	avCRPS-L	RMSE	avCRPS	avCRPS-T	avCRPS-L
Panel C: DGP 3 (random-walk time-variation and structural break)								
BART	0.355	0.193	0.044	0.061	0.906	0.523	0.127	0.182
	(0.082)	(0.051)	(0.011)	(0.018)	(0.154)	(0.114)	(0.034)	(0.054)
SoftBART	0.561	0.333	0.081	0.117	0.853	0.534	0.134	0.188
	(0.082)	(0.063)	(0.018)	(0.023)	(0.152)	(0.135)	(0.040)	(0.065)
MOTR-BART	0.367	0.206	0.047	0.069	0.884	0.501	0.126	0.171
	(0.061)	(0.034)	(0.009)	(0.013)	(0.147)	(0.104)	(0.029)	(0.046)
SMOTR-BART	0.582	0.364	0.093	0.128	0.932	0.606	0.158	0.226
	(0.105)	(0.091)	(0.027)	(0.034)	(0.119)	(0.113)	(0.037)	(0.061)
TVP-BART	**0.141**	**0.084**	**0.021**	**0.023**	**0.611**	**0.345**	**0.080**	**0.109**
	(0.048)	(0.024)	(0.006)	(0.007)	(0.095)	(0.038)	(0.006)	(0.011)
TVP-SoftBART	0.294	0.169	0.036	0.048	0.786	0.465	0.114	0.154
	(0.044)	(0.023)	(0.005)	(0.006)	(0.215)	(0.139)	(0.037)	(0.046)

Note: *normal_samples_per_gibbs_sample* is set to 1 instead of 100 which is used throughout the thesis. Simulation results for DGP 3, see Equation (26). This DGP includes random walk time variation in the parameters and a structural break. For each error metric, the means over 5 simulations are shown calculated over the in-sample period ($t = 1, \ldots, 129$) and out-of-sample period ($t = 130, \ldots, 149$). Bold numbers indicate the best performance (lowest error metric). The standard deviation is reported in parentheses. See Section 3.9 for error metrics definition.

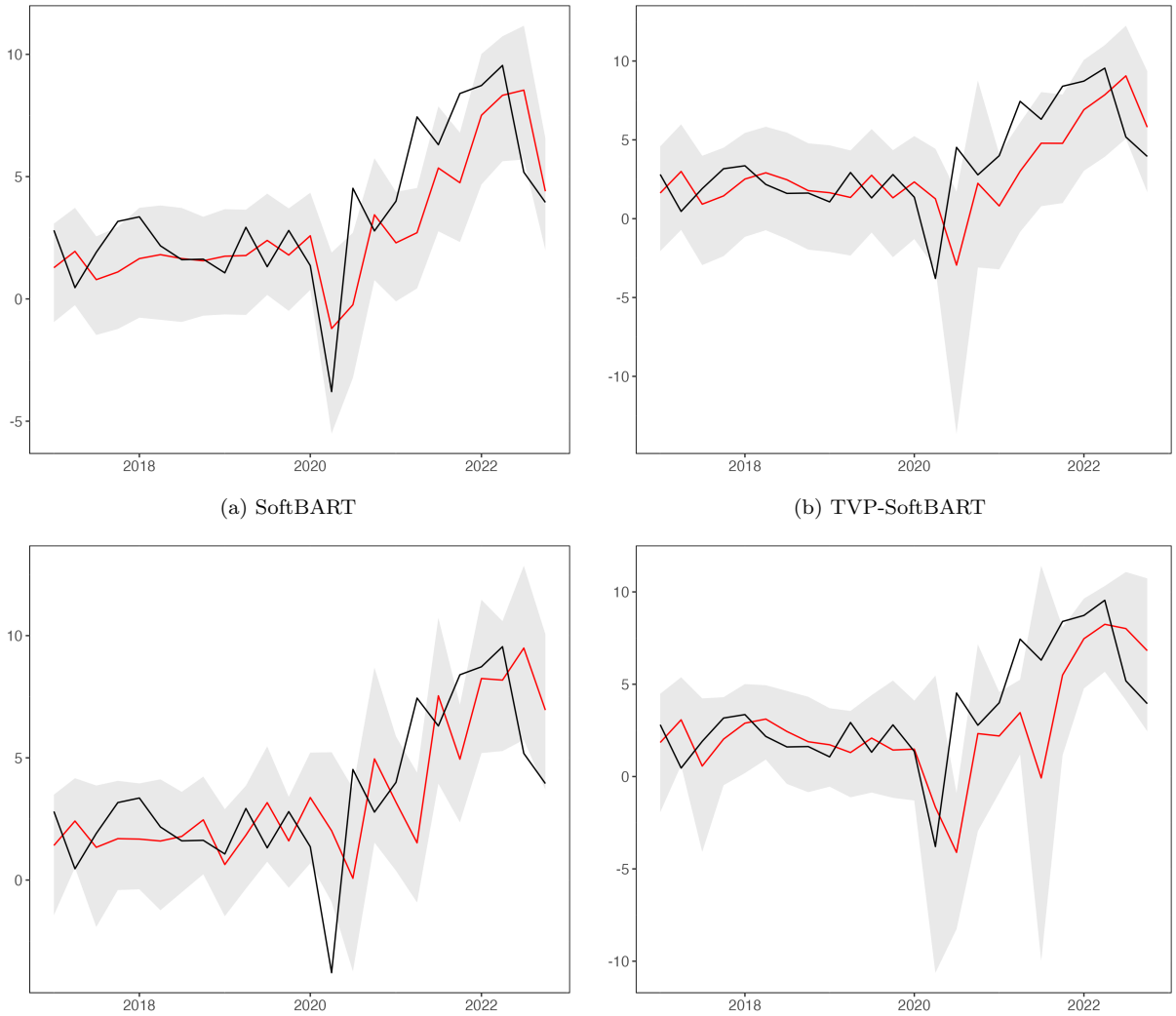
B Data

TABLE IX. Data Description

FRED-Code	Series	Trans.
GDPC1	Real Gross Domestic Product	Δln
PCECC96	Real Personal Consumption Expenditures	Δln
FPIx	Real private fixed investment	Δln
GCEC1	Real Government Consumption Expenditures and Gross Investment	Δln
INDPRO	IP:Total index Industrial Production Index	Δln
CUMFNS	Capacity Utilization: Manufacturing (SIC)	level
PAYEMS	Emp:Nonform All Employees: Total nonfarm	Δln
CE16OV	Civilian Employment	Δln
UNRATE	Civilian Unemployment Rate	Δ
AWHMAN	Average Weekly Hours of Production and Nonsupervisory Employees: Manufacturing (Hours)	level
CES0600000007	Average Weekly Hours of Production and Nonsupervisory Employees:Goods-Producing	Δ
CLAIMSx	Initial Claims	Δln
GDPCTPI	Gross Domestic Product: Chain-type Price Index	Δ^2ln
CPIAUCSL	Consumer Price Index for All Urban Consumers: All Items	Δ^2ln
PPIACO	Producer Price Index for All Commodities	Δ^2ln
WPSID61	Producer Price Index by Comodity Intermediate Materials: Supplies & Components	Δ^2ln
WPSID62	Producer Price Index: Crude Materials for Further Processing	Δ^2ln
COMPRNFB	Nonfarm Business Sector: Real Compensation per Hour (Index 2012=100)	Δln
ULCNFB	Nonfarm Business Sector: Unit Labor Cost (Index 2012=100)	Δln
CES0600000008	Average Hourly Earnings of Production and Nonsupervisory Employees	Δ^2ln
FEDFUNDS	Effective Federal Funds Rate (Percent)	Δ
BAA10YM	Moody's Seasoned Baa Corporate Bond Yield Relative to Yield on 10-Year Treasury	level
GS10TB3Mx	10-Year Treasury Constant Maturity Minus 3-Month Treasyrt Bill, secondary market	level
CPF3MTB3Mx	3-Month Commercial Paper Minus 3-Month Treasury Bill, secondary market	level
M2REAL	Real M2 Money Stock	Δln
BUSLOANSx	Real Commercial and Industrial Loans, All Commercial Banks	Δln
CONSUMERx	Real Consumer Loans at All Commercial Banks	Δln
SP500	S&P's Common Stock Price Index: Composite	Δln

Note: 'FRED-Code' refers to the code of the series at fred.stlouisfed.org. Transformations ('Trans.'): Δ indicates first differences, Δ^2 second differences and ln is the natural logarithm.

C Extra Figures



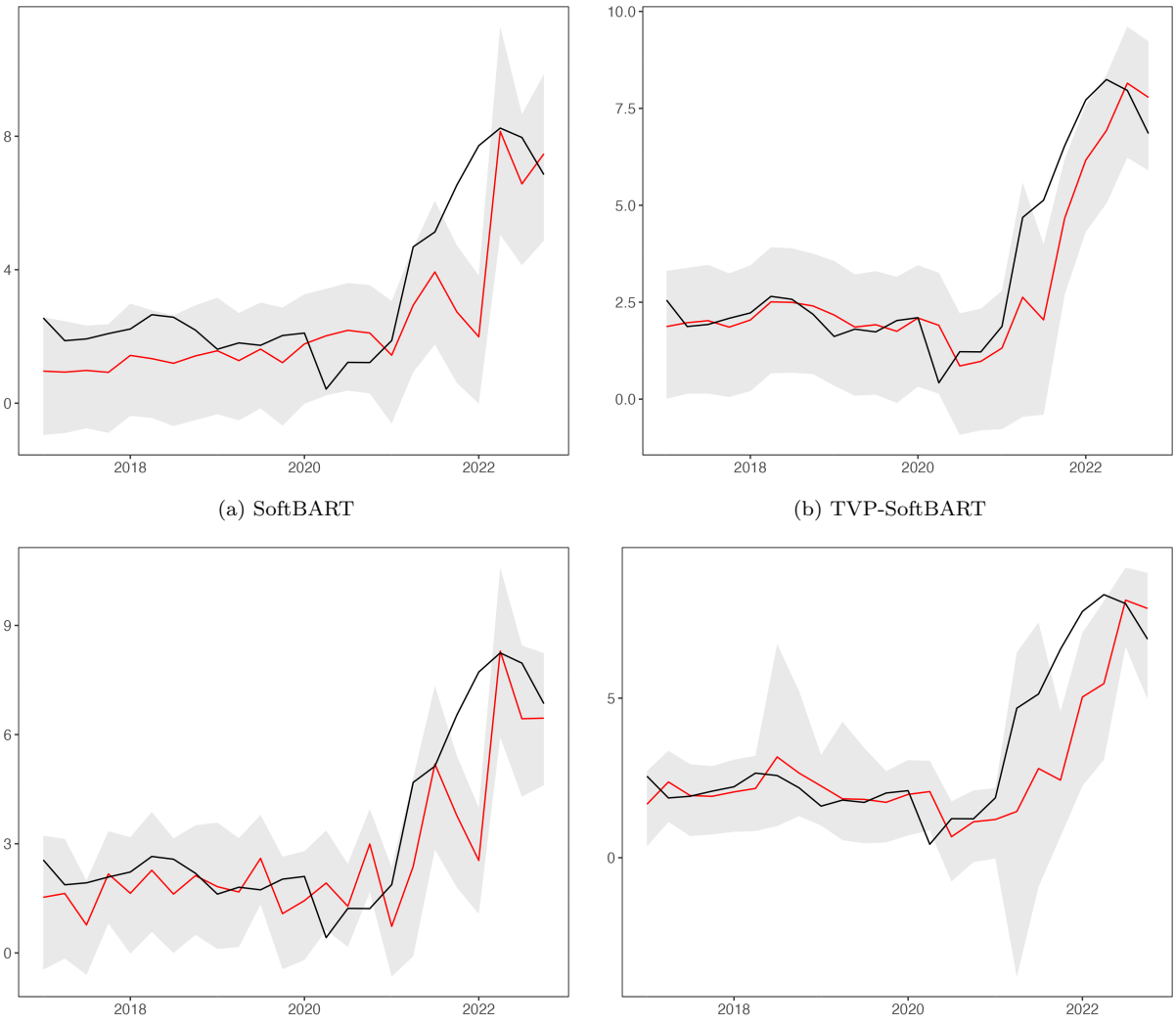
(a) SoftBART

(b) TVP-SoftBART

(c) BART

(d) TVP-BART

FIGURE V. 1-quarter-ahead inflation forecast (h=1). The grey shaded area refers to the 5th and 95th prediction intervals, the middle solid black line denotes the quarterly inflation and the red line is the posterior median.



(a) SoftBART

(b) TVP-SoftBART

(c) BART

(d) TVP-BART

FIGURE VI. 4-quarters-ahead inflation forecast (h=4). The grey shaded area refers to the 5th and 95th prediction intervals, the middle solid black line denotes the quarterly inflation and the red line is the posterior median.