Real-time Inflation Forecasting Using Non-linear Dimension Reduction Techniques

NIKO HAUZENBERGER[1, 2], FLORIAN HUBER[1], and KARIN KLIEBER*[1]

[1] *University of Salzburg*
[2] *Vienna University of Economics and Business*

December 16, 2020

In this paper, we assess whether using non-linear dimension reduction techniques pays off for forecasting inflation in real-time. Several recent methods from the machine learning literature are adopted to map a large dimensional dataset into a lower dimensional set of latent factors. We model the relationship between inflation and these latent factors using state-of-the-art time-varying parameter (TVP) regressions with shrinkage priors. Using monthly real-time data for the US, our results suggest that adding such non-linearities yields forecasts that are on average highly competitive to ones obtained from methods using linear dimension reduction techniques. Zooming into model performance over time moreover reveals that controlling for non-linear relations in the data is of particular importance during recessionary episodes of the business cycle.

JEL: C11, C32, C40, C53, E31
Keywords: Non-linear principal components, machine learning, time-varying parameter regression, density forecasting, real-time data

*Corresponding author: Karin Klieber. Salzburg Centre of European Union Studies, University of Salzburg. *Address*: Mönchsberg 2a, 5020 Salzburg, Austria. *Email*: karin.klieber@sbg.ac.at. We thank Michael Pfarrhofer and Anna Stelzer for valuable comments and suggestions. The authors gratefully acknowledge financial support from the Austrian Science Fund (FWF, grant no. ZK 35) and the Oesterreichische Nationalbank (OeNB, Anniversary Fund, project no. 18127).

1 Introduction

Inflation expectations are used as crucial inputs for economic decision making in central banks such as the European Central Bank (ECB) and the US Federal Reserve (Fed). Given current and expected inflation, economic agents decide on how much to consume, save and invest. In addition, measures of inflation expectations are often employed to estimate the slope of the Phillips curve, infer the output gap or the natural rate of interest. Hence, being able to accurately predict inflation is key for designing and implementing appropriate monetary policies in a forward looking manner.

Although the literature on modeling inflation is voluminous and the efforts invested considerable, predicting inflation remains a difficult task (Stock and Watson, 2007) and simple univariate models are still difficult to beat. The recent literature, however, has shown that using large datasets (Stock and Watson, 2002) and/or sophisticated models (see Koop and Potter, 2007; Koop and Korobilis, 2012; D'Agostino et al., 2013; Koop and Korobilis, 2013; Clark and Ravazzolo, 2015; Chan et al., 2018; Jarocinski and Lenza, 2018) has the potential to improve upon simpler benchmarks.

These studies often extract information from huge datasets. This is commonly achieved by extracting a relatively small number of principal components (PCs) and including them in a second stage regression model. While this approach performs well empirically, it fails to capture non-linear relations in the dataset. In the presence of non-linearities, using simple PCs potentially reduces predictive accuracy by ignoring important features of the data. Moreover, the regression model that links the PCs with inflation is often assumed to feature constant parameters and homoscedastic errors. In the presence of structural breaks and/or heteroscedasticity, this may adversely affect forecasting accuracy.

Investigating whether allowing for non-linearities in the compression stage pays off for inflation forecasting is the key objective of the present paper. Building on recent advances in machine learning (see Gallant and White, 1992; McAdam and McNelis, 2005; Exterkate et al., 2016; Chakraborty and Joseph, 2017; Heaton et al., 2017; Mullainathan and Spiess, 2017; Feng et al., 2018; Coulombe et al., 2019; Kelly et al., 2019; Medeiros et al., 2019), we adopt several non-linear dimension reduction techniques. The resulting latent factors are then linked to inflation in a second stage regression. In this second stage regression we allow for substantial flexibility. Specifically, we consider dynamic regression models that allow for time-varying parameters (TVPs) and stochastic volatility (SV). Since the inclusion of a relatively large number of latent factors can still imply a considerable number of parameters (and this problem is even more severe in the TVP regression case), we rely on state-of-the-art shrinkage techniques.

From an empirical standpoint it is necessary to investigate how these dimension reduction techniques perform over time and during different business cycle phases. We show this using a thorough real-time forecasting experiment for the US. Our forecasting application uses monthly real-time datasets (i.e., the FRED-MD database proposed in McCracken and Ng (2016)) and includes a battery of well established models commonly used in central banks and other policy institutions to forecast inflation.

Our results show that dimension reduction techniques yield forecasts that are highly competitive to the ones obtained from using linear methods based on PCs. At a first glance, this shows that existing models already perform well and using more sophisticated methods yields only modest gains in predictive accuracy. However, zooming into model performance over time reveals that controlling for non-linear relations in the data is of particular importance during recessionary episodes of the business cycle.

This finding gives rise to the second contribution of our paper. Since we find that more sophisticated non-linear dimension reduction methods outperform simpler techniques during recessions, we combine the considered models using dynamic model averaging (see Raftery et al., 2010; Koop and Korobilis, 2013). We show that combining our proposed set of models with a variety of standard forecasting models yields predictive densities which are superior to the single best performing model in overall terms. These effects are even more pronounced when interest centers on multi-step ahead forecasting.

The remainder of this paper is structured as follows. Section 2 discusses a set of dimension reduction techniques. Section 3 introduces the econometric modeling environment that we use to forecast inflation. Section 4 provides the results of the forecasting horse race and introduces weighted combinations of the competing models including the results of the forecast combinations. The last section summarizes and concludes the paper.

2 Linear and Non-linear Dimension Reduction Techniques

Suppose that we are interested in predicting inflation using a large number of K regressors that we store in a $T \times K$ matrix $\boldsymbol{X} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_T)'$, where $\boldsymbol{x}_t$ denotes a K-dimensional vector of observations at time t. If K is large relative to T, estimation of an unrestricted model that uses all columns in $\boldsymbol{X}$ quickly becomes cumbersome and overfitting issues arise. As a solution, dimension reduction techniques are commonly employed (see, e.g., Stock and Watson, 2002; Bernanke et al., 2005). These methods strike a balance between model fit and parsimony. At a very general level, the key idea is to introduce a function f that takes the matrix $\boldsymbol{X}$ as input and yields a lower dimensional representation $\boldsymbol{Z} = f(\boldsymbol{X}) = (\boldsymbol{z}_1, \dots, \boldsymbol{z}_T)'$, which is of dimension $T \times q$, as output. The critical assumption to achieve parsimony is that $K \gg q$. The latent factors in $\boldsymbol{Z}$ are then linked to inflation through a dynamic regression model (see Section 3).

The function $f : \mathbb{R}^{T\times K} \rightarrow \mathbb{R}^{T\times q}$ is typically assumed to be linear with the most prominent example being PCs. In this paper, we will consider several choices of f that range from linear to highly non-linear (such as manifold learning as well as deep learning algorithms) specifications. We subsequently analyze how these different specifications impact inflation forecasting accuracy. In the following subsections, we briefly discuss the different techniques and refer to the original papers for additional information.

2.1 Principal Component Analysis (PCA)

Minor alterations of the main PCA algorithm allow for introducing non-linearities in two ways. First, we can introduce a non-linear function g that maps the covariates onto a matrix $\boldsymbol{W} = g(\boldsymbol{X})$. Second, we could alter the sample covariance matrix (the kernel) with a function h: $\boldsymbol{\kappa} = h(\boldsymbol{W}'\boldsymbol{W})$. Both $\boldsymbol{W}$ and $\boldsymbol{\kappa}$ form the two main ingredients of a general PCA reducing the dimension to q, as outlined below (for details, see Schölkopf et al., 1998).

Independent of the functional form of g and h, we obtain PCs by performing a truncated singular value decomposition (SVD) of the transformed sample covariance matrix $\boldsymbol{\kappa}$. Conditional on the first q eigenvalues, the resulting factor matrix $\boldsymbol{Z}$ is of dimension $T \times q$. These PCs, for appropriate q, explain the vast majority of variation in $\boldsymbol{X}$. In the following, the relationship between the PCs and $\boldsymbol{X}$ is:

$$\boldsymbol{Z} = f(\boldsymbol{X}) = g(\boldsymbol{X})\boldsymbol{\Lambda}(\boldsymbol{\kappa}) = \boldsymbol{W}\boldsymbol{\Lambda}(\boldsymbol{\kappa}), \tag{1}$$

with $\boldsymbol{\Lambda}(\boldsymbol{\kappa})$ being the truncated $K \times q$ eigenvector matrix of $\boldsymbol{\kappa}$ (Stock and Watson, 2002). Notice that this is always conditional on deciding on a suitable number q of PCs. The number of factors is a crucial parameter that strongly influences predictive accuracy and inference (Bai and Ng, 2002). In our empirical work, we consider a small ($q = 5$), moderate ($q = 15$), and large ($q = 30$) number of PCs. In the case of a large number of PCs, we use shrinkage to solve overparameterization concerns.

By varying the functional form of g and h we are now able to discuss the first set of linear- and non-linear dimension reduction techniques belonging to the class of PCA:

1. **Linear PCs**

   The simplest way is to define both g and h as the unity function, resulting in $\boldsymbol{W} = \boldsymbol{X}$ and $\boldsymbol{\kappa} = \boldsymbol{X}'\boldsymbol{X}$. Due to the linear link between the PCs and the data, PCA is very easy to implement and yields consistent estimators for the latent factors if K and T go to infinity (Stock and Watson, 2002; Bai and Ng, 2008). Even if there is some time-variation in the factor loadings, Stock and Watson (1999) show that principal components remain a consistent estimator for the factors if K is large.

2. **Squared PCs**

   The literature suggests several ways to overcome the linearity restriction of PCs. Bai and Ng (2008), for example, apply a quadratic link function between the latent factors and the regressors, yielding a more flexible factor structure. This method considers squaring the elements of $\boldsymbol{X}$ resulting in

   $$\boldsymbol{W} = \boldsymbol{X}^2 \quad \text{and} \quad \boldsymbol{\kappa} = (\boldsymbol{X}^2)'(\boldsymbol{X}^2), \tag{2}$$

   with $\boldsymbol{X}^2 = (\boldsymbol{X} \odot \boldsymbol{X})$ and $\odot$ denoting element-wise multiplication.

   Squared PCs focus on the second moments of the covariate matrix and allow for a non-linear relationship between the principal components and the predictors. Bai and Ng (2008) show that quadratic variables can have substantial predictive power as they provide additional information on the underlying time series. Intuitively speaking, given that we transform our data to stationarity in the empirical work, this transformation strongly overweights situations characterized by sharp movements in the columns of $\boldsymbol{X}$ (such as during a recession). By contrast, periods characterized by little variation in our macroeconomic panel are transformed to mildly fluctuate around zero (and thus carry little predictive content for inflation). In our empirical model, our regressions always feature lagged inflation and this transformation thus effectively implies that in tranquil periods, the model is close to an autoregressive model whereas in crisis periods, more information is introduced.

3. **Kernel PCs**

   Another approach for non-linear PCs is the kernel principal component analysis (KPCA). KPCA dates back to Schölkopf et al. (1998), who proposed using integral operator kernel functions to compute PCs in a non-linear manner. In essence, this amounts to implicitly applying a non-linear transformation of the data through a kernel function and then applying PCA on this transformed dataset. Such an approach has been used for forecasting in Giovannelli (2012) and Exterkate et al. (2016).

   We allow for non-linearities in the kernel function between the data and the factors by defining h to be a Gaussian or a polynomial kernel $\boldsymbol{\kappa}$ (which is $K \times K$) with the (i, j)th element given by

$$\kappa_{ij} = \exp\left(-\frac{||\boldsymbol{x}_{\bullet i} - \boldsymbol{x}_{\bullet j}||}{2c_1^2}\right) \tag{3}$$

   for a Gaussian kernel and

$$\kappa_{ij} = \left(\frac{\boldsymbol{x}'_{\bullet i}\boldsymbol{x}_{\bullet j}}{c_0^2} + 1\right)^2 \tag{4}$$

   for a polynomial kernel.

   Here, $\boldsymbol{W} = \boldsymbol{X}$ (i.e., g is the unity function), $\boldsymbol{x}_{\bullet i}$ and $\boldsymbol{x}_{\bullet j}$ $(i, j = 1, \ldots, K)$ denote two columns of $\boldsymbol{X}$ while c_0 and c_1 are scaling parameters. As suggested by Exterkate et al. (2016) we set $c_0 = \sqrt{(K+2)/2}$ and $c_1 = \sqrt{c_K}/\pi$ with c_K being the 95th percentile of the χ^2 distribution with K degrees of freedom.

2.2 Diffusion Maps

Diffusion maps, originally proposed in Coifman et al. (2005) and Coifman and Lafon (2006), are another set of non-linear dimension reduction techniques that retain local interactions between data points in the presence of substantial non-linearities in the data.[1] The local interactions are preserved by introducing a random walk process.

The random walk captures the notion that moving between similar data points is more likely than moving to points which are less similar. We assume that the weight function which determines the strength of the relationship between $\boldsymbol{x}_{\bullet i}$ to $\boldsymbol{x}_{\bullet j}$ is given by

$$w(\boldsymbol{x}_{\bullet i}, \boldsymbol{x}_{\bullet j}) = \exp\left(\frac{||\boldsymbol{x}_{\bullet i} - \boldsymbol{x}_{\bullet j}||^2}{c_2}\right), \tag{5}$$

where $||\boldsymbol{x}_{\bullet i} - \boldsymbol{x}_{\bullet j}||$ denotes the Euclidean distance between $\boldsymbol{x}_{\bullet i}$ and $\boldsymbol{x}_{\bullet j}$ and c_2 is a tuning parameter set such that $w(\boldsymbol{x}_{\bullet i}, \boldsymbol{x}_{\bullet j})$ is close to zero except for $\boldsymbol{x}_{\bullet i} \approx \boldsymbol{x}_{\bullet j}$. Here, c_2 is determined by the median distance of k-nearest neighbors of $\boldsymbol{x}_{\bullet i}$ as suggested by Zelnik-Manor and Perona

[1] For an application to astronomical spectra, see Richards et al. (2009).

(2004). The number of k is chosen by taking a small percentage of K (i.e., 1 %) such that it scales with the size of the dataset.

The probability of moving from $\boldsymbol{x}_{\bullet i}$ to $\boldsymbol{x}_{\bullet j}$ is then simply obtained by normalizing:

$$p_{i \to j} = \text{Prob}(\boldsymbol{x}_{\bullet i} \to \boldsymbol{x}_{\bullet j}) = \frac{w(\boldsymbol{x}_{\bullet i}, \boldsymbol{x}_{\bullet j})}{\sum_j w(\boldsymbol{x}_{\bullet i}, \boldsymbol{x}_{\bullet j})}. \tag{6}$$

This probability tends to be small except for the situation where $\boldsymbol{x}_{\bullet i}$ and $\boldsymbol{x}_{\bullet j}$ are similar to each other. As a result, the probability that the random walk moves from $\boldsymbol{x}_{\bullet i}$ to $\boldsymbol{x}_{\bullet j}$ will be large if they are equal but rather small if both covariates differ strongly. Let $\boldsymbol{P}$ denote a transition matrix of dimension $K \times K$ with (i,j)th element given by $p_{i \to j}$. The probability of moving from $\boldsymbol{x}_{\bullet i}$ to $\boldsymbol{x}_{\bullet j}$ in $n = 1, 2, \ldots$ steps is then simply the matrix power of $\boldsymbol{P}^n$, with typical element denoted by $p^n_{i \to j}$. Using a biorthogonal spectral decomposition of $\boldsymbol{P}^n$ yields:

$$p^n_{i \to j} = \sum_{s \geq 0} \lambda^n_s \psi_s(\boldsymbol{x}_{\bullet i}) \phi_s(\boldsymbol{x}_{\bullet j}), \tag{7}$$

with ψ_s and ϕ_s denoting left and right eigenvectors of $\boldsymbol{P}$, respectively. The corresponding eigenvalues are given by λ_s.

We then proceed by computing the so-called diffusion distance as follows:

$$\xi^2_n(\boldsymbol{x}_{\bullet i}, \boldsymbol{x}_{\bullet j}) = \sum_j \frac{(p^n_{i \to j} - p^n_{s \to j})^2}{p_0(\boldsymbol{x}_{\bullet j})}, \tag{8}$$

with p_0 being a normalizing factor that measures the proportion the random walk spends at $\boldsymbol{x}_{\bullet j}$. This measure turns out to be robust with respect to noise and outliers. Coifman and Lafon (2006) show that

$$\xi^2_n(\boldsymbol{x}_{\bullet i}, \boldsymbol{x}_{\bullet j}) = \sum_{s=1}^{\infty} \lambda^{2n}_s (\psi_s(\boldsymbol{x}_{\bullet i}) - \psi_s(\boldsymbol{x}_{\bullet j}))^2. \tag{9}$$

This allows us to introduce the family of diffusion maps from $\mathbb{R}^K \to \mathbb{R}^q$ given by:

$$\boldsymbol{\Xi}_n(\boldsymbol{x}_{\bullet i}) = [\lambda^n_1 \psi_1(\boldsymbol{x}_{\bullet i}), \ldots, \lambda^n_q \psi_q(\boldsymbol{x}_{\bullet i})]. \tag{10}$$

The distance matrix can then be approximated as:

$$\xi^2_n(\boldsymbol{x}_{\bullet i}, \boldsymbol{x}_{\bullet j}) \approx \sum_{s=1}^{q} \lambda^{2n}_s (\psi_s(\boldsymbol{x}_{\bullet i}) - \psi_s(\boldsymbol{x}_{\bullet j}))^2 = ||\boldsymbol{\Xi}_n(\boldsymbol{x}_{\bullet i}) - \boldsymbol{\Xi}_n(\boldsymbol{x}_{\bullet j})||^2. \tag{11}$$

Intuitively, this equation states that we now approximate diffusion distances in $\mathbb{R}^K$ through the Euclidian distance between $\boldsymbol{\Xi}_n(\boldsymbol{x}_{\bullet i})$ and $\boldsymbol{\Xi}_n(\boldsymbol{x}_{\bullet j})$. This discussion implies that we have to choose n and q and we do this by setting $q = \{5, 15, 30\}$ according to our approach with either a small, moderate or large number of factors and $n = T$, the number of time periods. The algorithm in our application is implemented using the R-package `diffusionMap` (Richards and Cannoodt, 2019).

2.3 Local Linear Embedding

Locally linear embeddings (LLE) have been introduced by Roweis and Saul (2000). Intuitively, the LLE algorithm maps a high dimensional input dataset $\boldsymbol{X}$ into a lower dimensional space while being neighborhood-preserving. This implies that points which are close to each other in the original space are also close to each other in the transformed space.

The LLE algorithm is based on the assumption that each $\boldsymbol{x}_{\bullet i}$ is sampled from some underlying manifold. If this manifold is well defined, each $\boldsymbol{x}_{\bullet i}$ and its neighbors $\boldsymbol{x}_{\bullet j}$ are located close to a locally linear patch of this manifold. One consequence is that each $\boldsymbol{x}_{\bullet i}$ can, conditional on suitably chosen linear coefficients, be reconstructed from its neighbors $\boldsymbol{x}_{\bullet j}$ $j \neq i$. This reconstruction, however, will be corrupted by measurement errors. Roweis and Saul (2000) introduce a cost function to quantify these errors:

$$C(\boldsymbol{\Omega}) = \sum_i (\boldsymbol{x}_{\bullet i} - \sum_j \omega_{ij} \boldsymbol{x}_{\bullet j})^2, \tag{12}$$

with $\boldsymbol{\Omega}$ denoting a weight matrix with the (i, j)th element given by ω_{ij}. This cost function is then minimized subject to the constraint that each $\boldsymbol{x}_{\bullet i}$ is reconstructed only from its neighbors. This implies that $\omega_{ij} = 0$ if $\boldsymbol{x}_{\bullet j}$ is not a neighbor of $\boldsymbol{x}_{\bullet i}$. The second constraint is that the matrix $\boldsymbol{\Omega}$ is row-stochastic, i.e., the rows sum to one. Conditional on these two restrictions, the cost function can be minimized by solving a least squares problem.

To make this algorithm operational we need to define our notion of neighbors. In the following, we will use the k-nearest neighbors in terms of the Euclidean distance. We choose the number of neighbors by applying the algorithm proposed by Kayo (2006), which automatically determines the optimal number for k. The q latent factors in $\boldsymbol{Z}$, with typical ith column $\boldsymbol{z}_{\bullet i}$, are then obtained by minimizing:

$$\Phi(\boldsymbol{Z}) = \sum_i |\boldsymbol{z}_{\bullet i} - \sum_j \Omega_{ij} \boldsymbol{z}_{\bullet j}|^2, \tag{13}$$

which implies a quadratic form in $\boldsymbol{z}_t$. Subject to suitable constraints, this problem can be easily solved by computing:

$$\boldsymbol{M} = (\boldsymbol{I}_T - \boldsymbol{\Omega})'(\boldsymbol{I}_T - \boldsymbol{\Omega}), \tag{14}$$

and finding the $q + 1$ eigenvectors of $\boldsymbol{M}$ associated with the $q + 1$ smallest eigenvalues. The bottom eigenvector is then discarded to arrive at q factors. For our application, we use the R-package `lle` (Diedrich and Abel, 2012).

2.4 Isometric Feature Mapping

Isometric Feature Mapping (ISOMAP) is one of the earliest methods developed in the category of manifold learning algorithms. Introduced by Tenenbaum et al. (2000), the ISOMAP algorithm determines the geodesic distance on the manifold and uses multidimensional scaling to come up with a low number of factors describing the underlying dataset. Originally, ISOMAP was constructed for applications in visual perception and image recognition. In economics and finance, some recent papers highlight its usefulness (see, e.g., Ribeiro et al., 2008; Lin et al., 2011; Orsenigo and Vercellis, 2013; Zime, 2014).

The algorithm consists of three steps. In the first step, a dissimilarity index that measures the distance between data points is computed. These distances are then used to identify neighboring points on the manifold. In the second step, the algorithm estimates the geodesic distance between the data points as shortest path distances. In the third step, metric scaling is performed by applying classical multidimensional scaling (MDS) to the matrix of distances. For the dissimilarity transformation, we determine the distance between point i and j by the Manhattan index $d_{ij} = \sum_k |x_{ki} - x_{kj}|$ and collect those points where i is one of the k-nearest neighbors of j in a dissimilarity matrix. For our empirical application, we again choose the number of neighbors by applying the algorithm proposed by Kayo (2006) and use the algorithm implemented in the R-package `vegan` (Oksanen et al., 2019).

The described non-linear transformation of the dataset enables the identification of a non-linear structure hidden in a high-dimensional dataset and maps it to a lower dimension. Instead of pairwise Euclidean distances, ISOMAP uses the geodesic distances on the manifold and compresses information under consideration of the global structure.

2.5 Non-linear Compression with Deep Learning

Deep learning algorithms are characterized by not only non-linearly converting input to output but also representing the input itself in a transformed way. This is called representation learning in the sense that representations of the data are expressed in terms of other, simpler, representations before mapping the data input to output values.

One tool which performs representation of itself as well as representation to output is the Autoencoder (AE). The first step is accomplished by the encoder function, which maps an input to an internal representation. The second part, which maps the representation to the output, is called the decoder function. Their ability to extract factors, which largely explain variability of the observed data, in a non-linear manner makes deep learners a powerful tool complementing the range of commonly used dimension reduction techniques (Goodfellow et al., 2016). In empirical finance, Heaton et al. (2017), Feng et al. (2018) and Kelly et al. (2019) show that the application of these methods is beneficial to predict asset returns.

Based on deep learning techniques, we propose obtaining hierarchical predictors $\boldsymbol{Z}$ by applying a number of $l \in \{1, \ldots, L\}$ non-linear transformations to $\boldsymbol{X}$. The non-linear transformations are also called hidden layers with L giving the depth of our architecture and $f_1, \ldots, f_L$ denoting univariate activation functions for each layer. More specifically, activation functions (non-linearly) transform data in each layer, taking the output of the previous layer. A common choice is the hyperbolic tangent (tanh) given by $\frac{\exp(\boldsymbol{X}) - \exp(-\boldsymbol{X})}{\exp(\boldsymbol{X}) + \exp(-\boldsymbol{X})}$, justified by several findings in recent studies such as Saxe et al. (2019) or Andreini et al. (2020).

The structure of our deep learning algorithm can be represented in form of a composition of univariate semi-affine functions given by

$$f_l^{W^{(l)}, b_l} = f_l \left(\sum_{i=1}^{N_l} \boldsymbol{W}_{\bullet i}^{(l)} \hat{\boldsymbol{x}}_{\bullet i}^{(l)} + b_l \right), \quad 1 \leq l \leq L, \tag{15}$$

with $\boldsymbol{W}^{(l)}$ denoting a weighting matrix associated with layer l (with $\boldsymbol{W}_{\bullet i}^{(l)}$ denoting the ith column of $\boldsymbol{W}^{(l)}$), $\hat{\boldsymbol{x}}_{\bullet i}^{(l)}$ denotes the ith column of an input matrix $\hat{\boldsymbol{X}}^{(l)}$ to layer l, b_l is the corresponding bias term and N_l denotes the number of neurons that determine the width of the network. Notice that if $l = 1$, $\hat{\boldsymbol{X}}^{(1)} = \boldsymbol{X}$ and the input matrix is obtained recursively by using the activation functions.

The lower dimensional representation of our covariate matrix is then obtained by computing the composite map:

$$\boldsymbol{Z} = f(\boldsymbol{X}) = (f_1^{W^{(1)},b_1} \circ \cdots \circ f_L^{W^{(L)},b_L})(\boldsymbol{X}). \tag{16}$$

The optimal sets of $\hat{\boldsymbol{W}} = (\hat{\boldsymbol{W}}^{(0)}, \ldots, \hat{\boldsymbol{W}}^{(L)})$ and $\hat{b} = (\hat{b}_0, \ldots, \hat{b}_L)$ are obtained by computing a loss function, most commonly the mean squared error of the in-sample fit. The complexity of the neural network is determined by choosing the number of hidden layers L and the number of neurons in each layer N_l. We create five hidden layers with the number of neurons evenly downsizing to the desired number of factors. Corresponding to the standard literature (see, e.g., Huang, 2003; Heaton, 2008), a huge number of covariates requires a more complex structure (i.e., a higher number of hidden layers). Furthermore, it is recommended to set the number of neurons between the size of the input and the output layer where N_l is high in the first hidden layer and smaller in the following layers. We employ the R interface to `keras` (Allaire and Chollet, 2019), a high-level neural networks API and widely used package for implementing deep learning models.

3 A TVP Regression for Forecasting Inflation

In the following, we introduce the predictive regression that links our target variable (US inflation) to $\boldsymbol{Z}$ and p lags of inflation. Following Stock and Watson (1999), inflation is specified such that:

$$y_{t+h} = \ln\left(\frac{\text{CPI}_{t+h}}{\text{CPI}_t}\right) - \ln\left(\frac{\text{CPI}_t}{\text{CPI}_{t-1}}\right), \tag{17}$$

with CPI_{t+h} denoting the consumer price index in period $t + h$.

In the empirical application we set $h \in \{1, 3, 12\}$. y_{t+h} is then modeled using a dynamic regression model:

$$y_{t+h} = \boldsymbol{d}_t' \boldsymbol{\beta}_{t+h} + \epsilon_{t+h}, \quad \epsilon_{t+h} \sim \mathcal{N}(0, \sigma_{t+h}^2), \tag{18}$$

where $\boldsymbol{\beta}_{t+h}$ is a vector of TVPs associated with $M(= q + p)$ covariates denoted by $\boldsymbol{d}_t$ and σ_{t+h}^2 is a time-varying error variance. $\boldsymbol{d}_t$ might include the latent factors extracted from the various methods discussed in the previous subsection, lags of inflation, an intercept term or other covariates which are not compressed.

Following much of the literature (Taylor, 1982; Belmonte et al., 2014; Kalli and Griffin, 2014; Kastner and Frühwirth-Schnatter, 2014; Stock and Watson, 2016; Chan, 2017; Huber et al., 2020) we assume that the TVPs and the error variances evolve according to independent stochastic processes:

$$\begin{pmatrix} \boldsymbol{\beta}_{t+h} \\ \log \sigma^2_{t+h} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\beta}_{t+h-1} \\ \mu_h + \rho_h \log \sigma^2_{t+h-1} \end{pmatrix}, \begin{pmatrix} \boldsymbol{V} & 0 \\ 0 & \vartheta^2_h \end{pmatrix} \right), \tag{19}$$

with μ_h denoting the conditional mean of the log-volatility, ρ_h its persistence parameter and ϑ^2_h the error variance of $\log \sigma^2_{t+h}$. The matrix $\boldsymbol{V}$ is a $M \times M$-dimensional variance-covariance matrix with $\boldsymbol{V} = \text{diag}(v^2_1, \dots, v^2_M)$ and v^2_j being the process innovation variance that determines the amount of time-variation in $\boldsymbol{\beta}_{t+h}$. This setup implies that the TVPs are assumed to follow a random walk process while the log-volatilities evolve according to an AR(1) process.

The model described by Eq. 18 and Eq. 19 is a flexible state space model that encompasses a wide range of models commonly used for forecasting inflation. For instance, if we set $\boldsymbol{V} = \boldsymbol{0}_M$ and $\vartheta^2 = 0$, we obtain a constant parameter model. If $\boldsymbol{d}_t$ includes the lags of inflation and (lagged) PCs, we obtain a model closely related to the one used in Stock and Watson (2002). If we set $d_t = 1$ and allow for TVPs, we obtain a model very closely related to the unobserved components stochastic volatility (UC-SV) successfully adopted in Stock and Watson (1999). A plethora of other models can be identified by appropriately choosing $\boldsymbol{d}_t$, $\boldsymbol{V}$ and ϑ^2. This flexibility, however, calls for model selection. We select appropriate submodels by using Bayesian methods for estimation and forecasting. These techniques are further discussed in Appendix A and allow for data-based shrinkage towards simpler nested alternatives.

4 Forecasting US Inflation

4.1 Data Overview, Design of the Forecasting Exercise and Competitors

For the empirical application, we consider the popular FRED-MD database. This dataset is publicly accessible and available in real-time. The monthly data vintages ensure that we only use information that would have been available at the time a given forecast is being produced. A detailed description of the databases can be found in McCracken and Ng (2016). To achieve approximate stationarity we transform the dataset as given in Appendix B. Furthermore, each time series is standardized to have sample mean zero and unit sample variance prior to using the non-linear dimension reduction techniques.

Our US dataset includes 105 monthly variables that span the period from 1963:01 to 2019:06. The forecasting design relies on a rolling window, as justified by Clark (2011), that initially ranges from 1980:01 to 1999:12. For each month of the hold-out sample, which starts in 2000:01 and ends in 2018:12, we compute the h-step ahead predictive distribution for each model (for $h \in \{1, 3, 12\}$), keeping the length of the estimation sample fixed at 240 observations (i.e., a rolling window of 20 years).

One key limitation is that all methods are specified conditionally on $\boldsymbol{d}_t$ and thus implicitly on the specific function f used to move from $\boldsymbol{X}$ to $\boldsymbol{Z}$. Another key object of this paper is to control for uncertainty with respect to f by using dynamic model averaging techniques. For obtaining predictive combinations, we use the first 24 observations of our hold-out sample. The remaining periods (i.e., ranging from 2002:01 to 2018:12) then constitute our evaluation sample. For these periods we contrast each forecast (including the combined ones) with the realization of inflation in the final vintage of 2019:06. With such a strategy we aim at minimizing the risk that realized inflation especially at the end of the evaluation sample is still subject to revisions itself.[2]

In terms of competing models we can classify the specifications along two dimensions:

1. **How $\boldsymbol{d}_t$ is constructed.** First and importantly, let $\boldsymbol{s}_t$ denote an K_0-dimensional vector of covariates except for y_t. $\boldsymbol{x}_t = (\boldsymbol{s}'_t, \dots, \boldsymbol{s}'_{t-p+1})'$ is then composed of p lags of s_t with $K = pK_0$. In our empirical work we set $p = 12$ and include all variables in the dataset (except for the CPI series, i.e., $K_0 = 104$). We then use the different dimension reduction techniques outlined in Section 2 to estimate $\boldsymbol{z}_t$. Moreover, we add p lags of y_t to $\boldsymbol{z}_t$. This serves to investigate how different dimension reduction techniques perform when interest centers on predicting inflation. Moreover, we also consider simple AR(12) models as well as extended Phillips curve models (see, e.g., De Mol et al., 2008; Stock and Watson, 2008; Koop and Korobilis, 2012; Hauzenberger et al., 2019) as additional competitors. For the estimation of the extended Phillips curve model we select 20 covariates such that various economic sectors are covered.[3] Details can be found in Appendix B.

2. **The relationship between $\boldsymbol{d}_t$ and y_{t+h}.** The second dimension along which our models differ is the specific relationship described by Eq. 18. To investigate whether non-linear dimension reduction techniques are sufficient to control for unknown forms of non-linearities, we benchmark all our models that feature TVPs with their respective constant parameter counterparts. To perform model selection we consider two priors. The first one is the Horseshoe (HS) prior (Carvalho et al., 2010) and the second one is the stochastic search variable selection (SSVS) prior outlined in George and McCulloch (1993).

4.2 Full-sample Results across Dimension Reduction Techniques

In this subsection we briefly discuss how the factors obtained from using different dimension reduction techniques look like. For exposition, we choose $q = 5$ factors. Panels (a) to (h) in Figure 1 show the different factors and reveal remarkable differences across methods used to compress the data. Considering the different variants of the PCs suggests that the factors behave quite similar and exhibit a rather persistent behavior. This, however, does not hold for the case of squared PCA. In this case, the factors show sharp spikes during the global financial crisis. This is not surprising since squaring the input dataset, which has been transformed for

[2] In general, the literature argues that most of the data revisions take place in the first quarter while afterwards the vintages remain relatively unchanged (see Croushore, 2011; Pfarrhofer, 2020). Therefore a gap of six months between the final observation of inflation in the evaluation sample (2018:12) and our final vintage (2019:06) is considered as enough to render evaluation valid.

[3] We consider 20 covariates spanning different economic sectors, e.g.,

- **real activity:** industrial production (INDPRO), real personal income (W875RX1), housing (HOUST, PERMIT), capacity utilization (CUMFNS), etc.
- **labor market:** unemployment rates (UNRATE, CLAIMSx), employment (PAYEMS), avg. weekly hours of production (CES0600000007), etc.
- **price indices:** producer price index (PPICMM)
- **others:** Federal Funds Rate (FEDFUNDS), money supply (M2REAL), 3-M (TB3MS) and 10-y (GS10) treasuries, etc.

approximate stationarity, strongly overweights large absolute changes and squared PCA picks this up.

The other non-linear techniques tend to conserve more high frequency movements and yield factors that seem to be more noisy. This is especially pronounced in the case of the Autoencoder (see panel (a)) which yields factors that are heavily characterized by noise without displaying clear trends or persistent behavior. By contrast, when we consider diffusion maps (see panel (b)) the first impression is that the factors seem to be a mixture between squared PCA and one of the remaining PCA-based approaches. The changes during the global financial crisis and in the beginning of the 1980s are more pronounced and a slightly higher degree of noise is transferred from $\boldsymbol{x}_t$ to $\boldsymbol{z}_t$.

A similar pattern arises for LLE (see panel (d)). In this case, some of the factors behave similar to a regime-switching process with a moderate number of regimes. For instance, the dark gray line behaves similar to the PCs during the first few years of the sample. It then strongly decreases in the midst of the 1980s before returning to values observed in the beginning of the sample. Then, in the first half of the 1990s, we observe a strong increase (reaching a peak of around 5) before the factor quickly reverts back to the previous regime. This regime stays in place from 1996 to around 2003. Then we again find that dynamics change and the corresponding factor increases in the run-up to the global financial crisis. Similar patterns can be found for the other factors obtained from using LLE to compress the input data.

Considering ISOMAP shows that the first few factors appear to be highly persistent. These factors look very smooth for some periods but seem to exhibit oscillating behavior during other time periods. The intensity of these cycles, however, is small. The final few factors are fully characterized by these oscillating dynamics.

This brief discussion shows that the non-linear dimension reduction techniques yield very similar results with distinct dynamics. Some of them (especially the Autoencoder) pick up a lot of high frequency movements. These movements might be irrelevant for modeling inflation dynamics but could nevertheless carry relevant information during certain periods in time. A similar argument applies to the other techniques which also yield factors that change their behavior over time.

4.3 Density and Point Forecast Performance

We now consider point and density forecasting performance of the different models and dimension reduction techniques. The forecast performance is evaluated through averaged log predictive likelihoods (LPLs) for density forecasts and root mean squared errors (RMSEs) for point forecasts. Superior models are those with high scores in terms of LPL and low values in terms of RMSE. Formal descriptions of the evaluation metrics are provided in Appendix A. We benchmark all models relative to the autoregressive (AR) model with constant parameters and the HS prior. The first entry in the tables gives the actual LPL score (in averages) with actual RMSEs in parentheses for our benchmark model. The remaining entries are relative LPLs with relative RMSEs in parentheses.

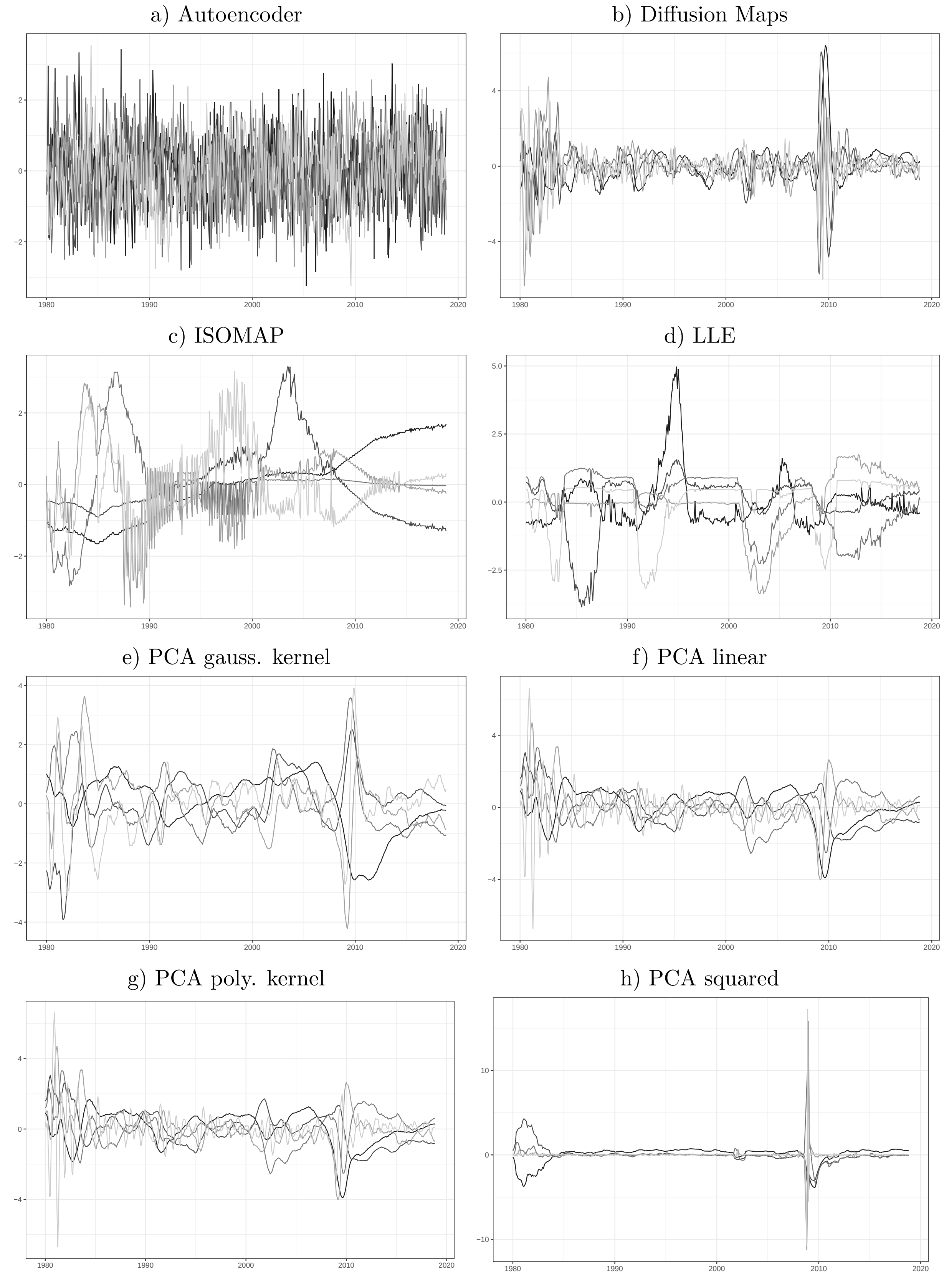
Starting with the one-step ahead horizon, Table 1 shows the relative LPLs and RMSEs (in parentheses) for inflation forecasts. This table suggests that, in terms of density forecasts, using dimension reduction techniques (both linear and non-linear) and allowing for non-linearities between the factors and inflation improves density forecasts substantially. This does not carry over to point forecasts. When we consider relative RMSEs, only small improvements are obtained by using more sophisticated modeling techniques.

Comparing linear to non-linear dimension reduction methods suggests that forecasts can be further improved. In particular, we observe that along the different reduction techniques, squared PCA performs well. One explanation for this might relate to the fact that simple models such as a random walk or other univariate benchmarks are hard to beat in a real-time forecasting exercise (see Atkeson et al., 2001; Stock and Watson, 2008; Stella and Stock, 2013). When taking a closer look on Figure 1 (h) we see that the factors are close to zero in tranquil periods, while at the same time, show substantial movements in times of turmoil. Conditional on relatively small regression coefficients in Eq. 18, this pattern suggests that the forecast densities are close to the ones obtained from a random walk model. But in recessionary episodes, the factors convey information on the level and volatility of inflation that might be useful for predicting during crises periods (see, e.g., Chan, 2017; Huber and Pfarrhofer, 2020).

When we consider the different specifications for the observation equation we find that allowing for time-variation in the parameters improves one-step ahead predictive densities. These improvements appear to be substantial for all specifications except the model using squared PCA. For squared PCA, we find only limited differences between constant and TVP regressions (conditional on the specific prior). The single best performing model for the one-step ahead inflation forecasts is the TVP model with a Horseshoe prior and five factors obtained by using squared PCA.

Again, the strong differences in predictive accuracy between constant and TVP specifications arise from the necessity to discriminate between different stages of the business cycle. The somewhat smaller differences in the case of squared PCA are driven by the specific shape of the latent factors and the reason outlined in the previous paragraph.

Next, we inspect the longer forecast horizons in greater detail. Table 2 depicts the forecast performance of all competitors for one-quarter and one-year ahead. The table indicates that non-linear dimension reduction techniques clearly outperform the autoregressive benchmark and perform similarly to the linear PCAs. Results reveal that diffusion maps, isometric feature mapping as well as squared PCA in combination with time variation in the coefficients yield high LPLs. Here, again, the best performing model is squared PCA, which beats all other dimension reduction techniques irrespective of the prior structure or whether constant or time-varying parameters are considered. For point forecasts, we again find little differences relative to the univariate benchmark model.



a) Autoencoder

b) Diffusion Maps

c) ISOMAP

d) LLE

e) PCA gauss. kernel

f) PCA linear

g) PCA poly. kernel

h) PCA squared

Figure 1: Illustration of linear and non-linear dimension reduction techniques applied to our US dataset with $K = 104$ based on the last vintage (end of year 2018). By focussing on $q = 5$ we depict normalized factors with mean zero and variance one ranging from January 1980 to December 2018.

Table 1: One-month ahead forecast performance.

Specification	**One-month ahead**			
	const. (HS)	const. (SSVS)	TVP (HS)	TVP (SSVS)
AR	-336.98	0.40	15.57	19.69
	(1.18)	(1.01)	(1.00)	(1.01)
Autoencoder (q = 5)	1.67	4.64	13.71	22.51
	(1.00)	(**1.00**)	(1.00)	(**1.00**)
Autoencoder (q = 15)	1.00	2.88	10.79	14.00
	(1.00)	(1.01)	(1.01)	(1.05)
Autoencoder (q = 30)	2.32	0.31	12.93	12.97
	(1.00)	(1.01)	(1.00)	(1.06)
Diffusion Maps (q = 5)	2.57	1.14	13.81	15.59
	(1.00)	(1.01)	(1.01)	(1.12)
Diffusion Maps (q = 15)	0.71	2.92	13.54	17.26
	(1.00)	(1.01)	(1.00)	(1.06)
Diffusion Maps (q = 30)	2.28	3.14	14.44	-0.36
	(1.00)	(1.02)	(1.00)	(1.15)
Extended PC	11.25	15.73		
	(**0.99**)	(1.07)		
ISOMAP (q = 5)	0.99	-0.58	10.80	19.21
	(1.00)	(1.01)	(1.00)	(1.01)
ISOMAP (q = 15)	0.06	1.30	9.71	18.86
	(1.00)	(1.01)	(1.01)	(1.02)
ISOMAP (q = 30)	-1.18	2.38	9.73	20.37
	(1.00)	(1.01)	(1.02)	(1.03)
LLE (q = 5)	0.18	-1.83	13.81	19.75
	(1.00)	(1.01)	(**1.00**)	(1.01)
LLE (q = 15)	-2.02	0.05	11.64	19.06
	(1.01)	(1.01)	(1.00)	(1.01)
LLE (q = 30)	-1.11	-3.63	6.71	19.68
	(1.00)	(1.01)	(1.01)	(1.01)
PCA gauss. kernel (q = 5)	-0.74	0.67	13.69	15.85
	(1.00)	(1.01)	(1.00)	(1.05)
PCA gauss. kernel (q = 15)	-0.20	2.65	14.49	11.27
	(1.00)	(1.01)	(1.01)	(1.17)
PCA gauss. kernel (q = 30)	0.28	6.86	15.78	-5.34
	(1.00)	(1.01)	(1.01)	(1.30)
PCA linear (q = 5)	-0.80	0.51	11.48	18.95
	(1.00)	(1.01)	(1.01)	(1.03)
PCA linear (q = 15)	-0.51	2.32	12.56	18.95
	(1.01)	(1.01)	(1.02)	(1.04)
PCA linear (q = 30)	0.27	7.05	16.46	**25.51**
	(1.01)	(1.00)	(1.02)	(1.03)
PCA poly. kernel (q = 5)	1.86	-0.39	12.52	15.02
	(1.00)	(1.01)	(1.00)	(1.05)
PCA poly. kernel (q = 15)	-0.11	2.78	15.56	11.82
	(1.00)	(1.01)	(1.00)	(1.18)
PCA poly. kernel (q = 30)	0.64	4.44	16.10	0.59
	(1.00)	(1.01)	(1.01)	(1.22)
PCA squared (q = 5)	16.79	**26.48**	**30.15**	17.53
	(1.02)	(1.01)	(1.02)	(1.19)
PCA squared (q = 15)	**21.26**	23.80	25.65	10.10
	(1.02)	(1.03)	(1.02)	(1.61)
PCA squared (q = 30)	19.01	21.96	23.46	2.94
	(1.01)	(1.04)	(1.02)	(1.86)

Note: The first (red shaded) entry gives the actual LPL score in averages with actual RMSEs in parentheses of our benchmark model, which is the autoregressive (AR) model with constant parameters and the HS prior. All other entries are relative LPLs with relative RMSEs in parentheses.

Table 2: One-quarter and one-year ahead forecast performance.

Specification	One-quarter ahead				One-year ahead			
	const. (HS)	const. (SSVS)	TVP (HS)	TVP (SSVS)	const. (HS)	const. (SSVS)	TVP (HS)	TVP (SSVS)
AR	-383.12 (1.31)	13.10 (0.99)	23.66 (1.00)	31.64 (1.03)	-408.15 (1.41)	8.87 (1.01)	16.52 (1.00)	26.25 (1.01)
Autoencoder (q = 5)	1.02 (1.00)	17.96 (1.00)	26.39 (0.99)	36.60 (**1.03**)	0.51 (1.01)	11.24 (1.00)	15.55 (**1.00**)	34.21 (**1.01**)
Autoencoder (q = 15)	0.34 (1.00)	10.34 (1.00)	21.68 (1.00)	34.66 (1.06)	3.44 (1.00)	12.95 (1.01)	15.60 (1.00)	39.35 (1.02)
Autoencoder (q = 30)	0.00 (1.00)	19.77 (1.00)	19.29 (1.00)	33.63 (1.09)	-1.54 (1.00)	12.97 (1.00)	12.55 (1.00)	37.53 (1.04)
Diffusion Maps (q = 5)	1.09 (1.00)	17.18 (0.99)	25.75 (0.99)	40.24 (1.13)	-0.43 (1.00)	17.39 (1.00)	16.16 (1.00)	29.34 (1.48)
Diffusion Maps (q = 15)	-1.56 (1.00)	18.56 (0.99)	25.54 (0.99)	**48.55** (1.07)	1.16 (1.00)	16.37 (1.01)	17.51 (1.00)	43.38 (1.48)
Diffusion Maps (q = 30)	2.47 (1.00)	21.93 (0.99)	26.93 (0.99)	44.25 (1.07)	-1.32 (1.00)	16.78 (1.02)	17.72 (1.00)	36.94 (1.52)
Extended PC	10.38 (1.00)	44.29 (1.05)			4.70 (1.01)	46.71 (1.06)		
ISOMAP (q = 5)	2.63 (1.00)	14.00 (0.99)	21.90 (1.00)	34.08 (1.03)	-1.28 (1.00)	9.26 (1.01)	13.52 (1.01)	26.90 (1.03)
ISOMAP (q = 15)	1.32 (1.00)	13.67 (1.00)	19.14 (1.01)	33.98 (1.04)	1.95 (1.00)	12.66 (1.00)	11.08 (1.00)	34.82 (1.02)
ISOMAP (q = 30)	6.39 (**0.99**)	20.23 (**0.98**)	16.20 (1.00)	40.61 (1.03)	-10.28 (1.01)	5.01 (1.01)	8.05 (1.01)	27.53 (1.04)
LLE (q = 5)	-2.94 (1.00)	8.09 (1.00)	24.14 (1.00)	33.60 (1.03)	-1.86 (1.00)	5.70 (1.01)	11.68 (1.00)	27.44 (1.01)
LLE (q = 15)	-7.05 (1.00)	10.62 (1.00)	16.88 (1.00)	33.67 (1.03)	1.02 (1.00)	4.45 (1.01)	9.70 (1.00)	26.66 (1.01)
LLE (q = 30)	-6.21 (1.00)	8.25 (1.00)	15.55 (1.00)	31.47 (1.04)	-4.56 (1.00)	4.14 (1.00)	8.30 (1.00)	29.24 (1.01)
PCA gauss. kernel (q = 5)	2.83 (1.00)	14.65 (0.99)	24.91 (1.00)	32.19 (1.08)	1.53 (1.00)	10.61 (1.00)	14.64 (1.00)	31.25 (1.11)
PCA gauss. kernel (q = 15)	0.85 (1.00)	18.40 (1.00)	21.89 (1.00)	27.12 (1.34)	-2.78 (1.01)	15.55 (1.01)	15.88 (1.05)	27.95 (1.32)
PCA gauss. kernel (q = 30)	4.74 (1.00)	18.56 (1.00)	27.43 (1.00)	9.82 (1.55)	-2.27 (1.01)	19.62 (1.02)	11.78 (1.05)	16.79 (1.51)
PCA linear (q = 5)	1.06 (1.00)	12.45 (1.00)	20.24 (1.00)	34.72 (1.04)	0.61 (1.01)	10.96 (1.01)	18.44 (1.00)	30.69 (1.02)
PCA linear (q = 15)	4.74 (1.00)	16.12 (1.00)	22.90 (1.01)	37.77 (1.05)	-0.79 (1.01)	16.83 (1.01)	18.19 (1.03)	36.97 (1.09)
PCA linear (q = 30)	7.76 (0.99)	21.16 (0.99)	22.94 (1.01)	45.40 (1.04)	2.52 (1.00)	22.03 (**1.00**)	16.49 (1.03)	42.25 (1.10)
PCA poly. kernel (q = 5)	2.32 (1.00)	10.80 (1.00)	21.68 (1.00)	34.27 (1.09)	4.74 (1.00)	13.20 (1.01)	15.21 (1.00)	34.53 (1.08)
PCA poly. kernel (q = 15)	-1.33 (1.00)	14.52 (1.00)	23.42 (1.00)	31.02 (1.24)	2.37 (1.00)	11.31 (1.01)	16.09 (1.01)	32.05 (1.25)
PCA poly. kernel (q = 30)	1.68 (1.00)	19.78 (0.99)	23.15 (**0.99**)	23.30 (1.36)	-0.07 (1.00)	15.35 (1.00)	15.37 (1.02)	8.87 (1.48)
PCA squared (q = 5)	**54.76** (1.03)	**54.26** (1.06)	**65.09** (1.01)	44.01 (2.60)	**69.24** (**0.95**)	**80.56** (1.02)	**74.74** (1.03)	**51.17** (3.36)
PCA squared (q = 15)	51.10 (1.04)	54.01 (1.03)	57.09 (1.03)	32.56 (3.18)	55.54 (1.00)	68.09 (1.25)	67.21 (1.05)	28.52 (4.32)
PCA squared (q = 30)	48.84 (1.04)	52.21 (1.03)	60.12 (1.04)	23.10 (3.35)	62.97 (0.99)	69.93 (1.25)	70.63 (1.05)	19.67 (4.62)

Note: The first (red shaded) entry gives the actual LPL score in averages with actual RMSEs in parentheses of our benchmark model, which is the autoregressive (AR) model with constant parameters and the HS prior. All other entries are relative LPLs with relative RMSEs in parentheses.

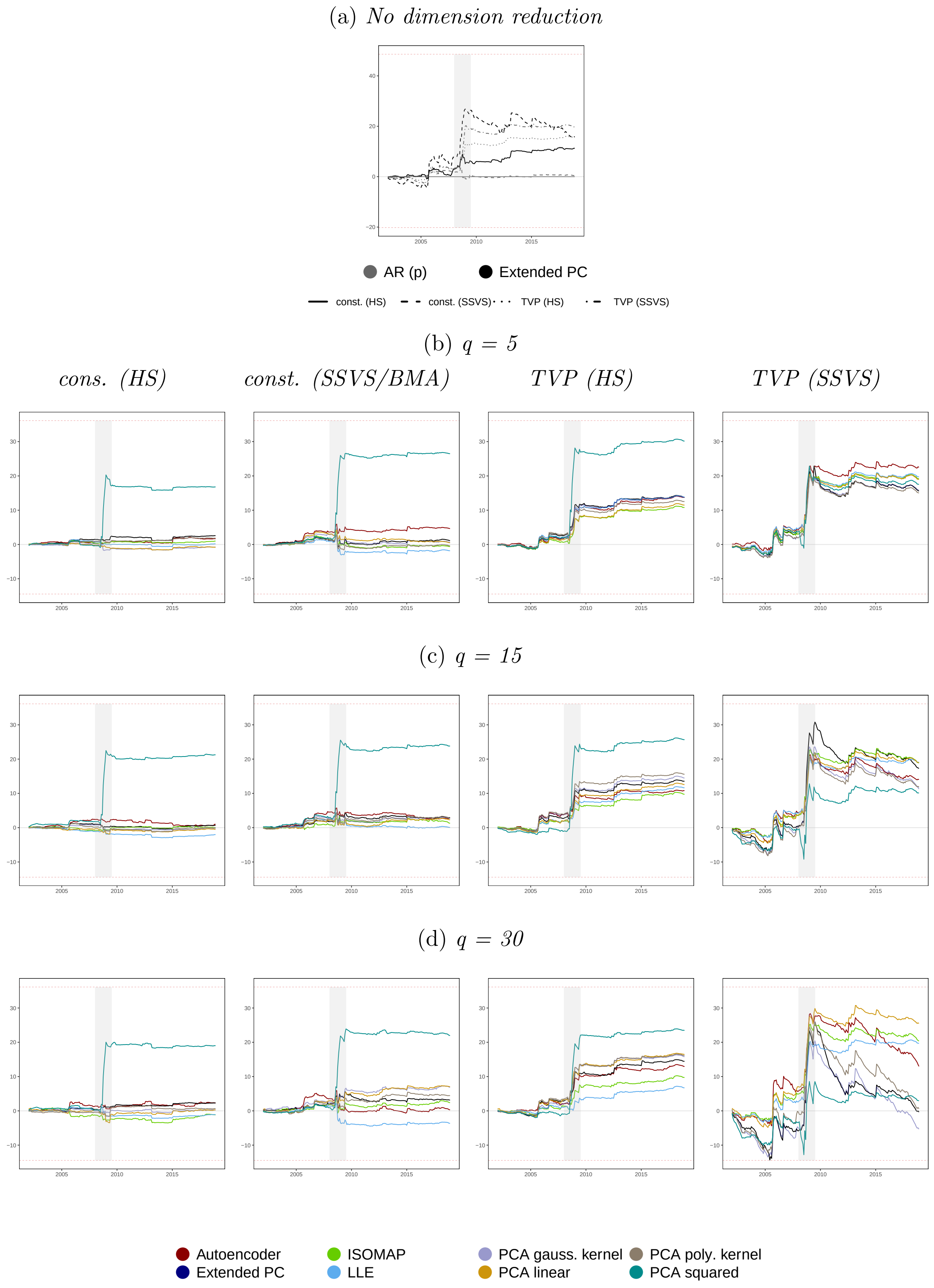
So far, the LPLs are averaged over the full evaluation sample and thus only measure model quality over the full hold-out period (Geweke and Amisano, 2010). However, this might mask important differences in forecast performance of the different models and compression techniques over time. Figure 2 depicts the average LPLs along the hold-out sample for the short run forecasting exercise. The figure suggests a great deal of performance variation over time. Regardless of the model specification and the number of factors included in the models, accounting for instabilities in the relationship between the factors and inflation through time-varying parameters improves the forecasting performance. Especially during the global financial crisis (the gray shaded area), more flexible model specifications yield greater improvements relative to the univariate benchmark and compared to constant specifications.

4.4 Dynamic Model Learning Based on Density Forecast Performance

The final paragraph in the previous subsection showed that model performance varies considerably over time. The key implication is that non-linear compression techniques are useful during turbulent times whereas forecast evidence is less pronounced in normal times. In this subsection, we ask whether combining models in a dynamic manner further improves predictive accuracy.

After having obtained the predictive densities of y_{t+h} for the different dimensionality reduction techniques and model specifications, the goal is to exploit the advantages of both linear and non-linear approaches. This is achieved by combining models in a model pool such that better performing models over certain periods receive larger weights while inferior models are subsequently down-weighted. The literature on forecast combinations suggests several different weighting schemes, ranging from simply averaging over all models (see, e.g., Hendry and Clements, 2004; Hall and Mitchell, 2007; Clark and McCracken, 2010; Berg and Henzel, 2015) to estimating weights based on the models' performances according to the minimization of an objective or loss function (see, e.g., Timmermann, 2006; Hall and Mitchell, 2007; Geweke and Amisano, 2011; Conflitti et al., 2015; Pettenuzzo and Ravazzolo, 2016) or according to the posterior probabilities of the predictive densities (see, e.g., Raftery et al., 2010; Koop and Korobilis, 2012; Beckmann et al., 2020). Since the weights might change over time, we aim to compute them in a dynamic manner.

Combining the different predictive densities according to their posterior probabilities is referred to as Bayesian model averaging (BMA). The resulting weights are capable of reflecting the predictive power of each model for the respective periods. Dynamic model averaging (DMA), as specified by Raftery et al. (2010), extends the approach by adding a discount (or *forgetting*) factor to control for a model's forecasting performance in the recent past. The 'recent past' is determined by the discount factor, with higher values attaching greater importance to past forecasting performances of the model and lower values gradually ignoring results of past predictive densities. Similar to Beckmann et al. (2020), Koop and Korobilis (2012) and Raftery et al. (2010), we apply DMA to combine the predictive densities of our various models.



(a) *No dimension reduction*

(b) q = *5*

cons. (HS) *const. (SSVS/BMA)* *TVP (HS)* *TVP (SSVS)*

(c) q = *15*

(d) q = *30*

Figure 2: Evolution of one-month ahead cumulative LPBFs relative to the benchmark. The red dashed lines refer to the maximum/minimum Bayes factor over the full hold-out sample. The light gray shaded areas indicate the NBER recessions in the US.

DMA works as follows. Let $\boldsymbol{\varrho}_{t+h|t+h} = (\varrho_{t+h|t+h,1}, \dots, \varrho_{t+h|t+h,J})'$ denote a set of weights for J competing models. These (horizon-specific) weights vary over time and depend on the recent predictive performance of the model according to:

$$\varrho_{t+h|t,j} = \frac{\varrho_{t|t,j}^{\delta}}{\sum_{l=1}^{J} \varrho_{t|t,l}^{\delta}}, \tag{20}$$

$$\varrho_{t+h|t+h,j} = \frac{\varrho_{t+h|t,j}\ p_j(y_{t+h}|y_{1:t})}{\sum_{l=1}^{J} \varrho_{t+h|t,l}\ p_l(y_{t+h}|y_{1:t})} \tag{21}$$

where $p_j(y_{t+h}|y_{1:t})$ denotes the h-step ahead predictive distribution of model j and $\delta \in (0, 1]$ denotes a forgetting factor close to one. In our empirical work we set $\delta = 0.9$. Notice that if $\delta = 1$, we obtain standard BMA weights while $\delta = 0$ would imply that the weights depend exclusively on the forecasting performance in the last period.

4.5 Forecasting Performance of Predictive Combinations from Dynamic Model Learning

Weights obtained by combining models according to their predictive power convey useful information about the adequacy of each model over time. In order to get a comprehensive picture of the effects of different model modifications, we combine our models and model specifications in various ways.

Table 3 presents the forecasting results when we use DMA to combine models. Again, all models are benchmarked to the AR model with constant parameters and the HS prior. The first row depicts the relative performance of the best performing single model for the chosen time horizon.

The table can be understood as follows. Each entry includes *all* dimension reduction techniques. The rows define whether the model space includes all factors $q \in \{5, 15, 30\}$ or whether we combine models with a fixed number of factors exclusively. The columns refer to model spaces which include only constant parameter, time-varying parameter or both specifications in the respective model pool. Since we also discriminate between two competing priors we also consider model weights if we condition on either the HS or the SSVS prior or average across both prior specifications (the first upper part of the table with {HS, SSVS}).

Across all three forecast horizons considered, we again find only limited accuracy improvements for point forecasts relative to the AR model. This, however, does not carry over to LPLs. For density forecasts, we find that DMA-based combinations improve upon the single best performing model for all forecast horizons. Hence, allowing models to change over the hold-out period leads to superior predictive accuracy.

Table 3: Forecast performance of predictive combinations.

Specification		**One-month ahead**			**One-quarter ahead**			**One-year ahead**		
Single best performing model			30.15			65.09			80.56	
			(0.99)			(0.98)			(0.95)	
Prior	Combination	Const.	TVP	{const., TVP}	Const.	TVP	{const., TVP}	Const.	TVP	{const., TVP}
{HS, SSVS}	q = {5, 15, 30}	23.73	32.14	29.79	53.32	66.46	65.11	85.74	83.03	83.82
		(1.01)	(1.04)	(1.04)	(1.02)	(0.98)	(0.99)	(1.03)	(1.01)	(1.02)
	q = 5	24.13	27.57	26.67	52.68	65.02	62.83	83.21	82.55	83.29
		(1.01)	(1.03)	(1.02)	(1.02)	(1.02)	(1.03)	(0.99)	(1.01)	(0.99)
	q = 15	21.37	31.07	28.94	52.28	62.00	60.89	79.67	80.24	79.62
		(1.02)	(1.02)	(1.02)	(1.02)	(1.02)	(1.02)	(1.20)	(1.04)	(1.05)
	q = 30	22.70	**34.14**	32.30	54.29	**67.58**	**66.96**	83.94	80.91	81.22
		(1.01)	(1.04)	(1.04)	(1.02)	(**0.97**)	(**0.97**)	(1.17)	(1.04)	(1.09)
HS	q = {5, 15, 30}	18.83	29.73	26.87	50.86	62.41	59.96	74.17	**85.39**	83.87
		(1.00)	(1.00)	(1.01)	(1.03)	(1.01)	(1.02)	(**0.92**)	(**1.00**)	(**0.99**)
	q = 5	17.97	29.95	26.85	52.31	63.74	61.57	72.00	82.70	81.20
		(1.00)	(1.01)	(1.01)	(1.03)	(1.01)	(1.01)	(0.93)	(1.01)	(1.00)
	q = 15	18.82	29.44	26.37	50.34	59.23	56.77	66.54	82.79	81.50
		(1.01)	(**1.00**)	(1.00)	(1.03)	(1.02)	(1.02)	(0.96)	(1.01)	(1.01)
	q = 30	18.32	26.66	23.81	48.94	61.53	58.95	70.54	83.68	82.02
		(**0.99**)	(1.00)	(**1.00**)	(1.03)	(1.03)	(1.02)	(0.96)	(1.01)	(1.00)
SSVS	q = {5, 15, 30}	25.99	30.36	31.01	55.21	60.56	64.46	**86.68**	67.03	**84.16**
		(1.00)	(1.04)	(1.04)	(1.02)	(1.00)	(0.98)	(1.03)	(1.22)	(1.06)
	q = 5	**26.47**	19.21	26.40	53.96	48.94	60.31	83.82	66.32	83.95
		(1.00)	(1.16)	(1.02)	(1.04)	(2.35)	(1.06)	(0.99)	(2.56)	(0.99)
	q = 15	22.65	28.16	29.18	53.63	54.81	60.25	80.07	63.57	76.81
		(1.02)	(1.02)	(1.02)	(1.02)	(1.04)	(1.02)	(1.20)	(3.17)	(1.23)
	q = 30	24.93	30.24	**33.09**	**55.77**	60.75	65.60	85.66	64.72	81.69
		(1.02)	(1.05)	(1.04)	(**1.01**)	(1.00)	(0.98)	(1.16)	(1.24)	(1.18)

Note: The first (grey shaded) row states the results of the single best performing model as presented in the previous chapter for each forecast horizon benchmarked to the AR model with constant parameters and the HS prior. All other rows show the relative results for the combinations of the different dimension reduction techniques according to the specification stated in the rows and columns headers. For example, the entry in row $\{HS, SSVS\}$, $q = \{5, 15, 30\}$ and column *Const.* combines all models estimated with constant parameters, the HS prior, the SSVS prior, 5, 15 and 30 factors. Entries denote the relative LPL with relative RMSE in parantheses benchmarked against the AR model with constant parameters and the HS prior.

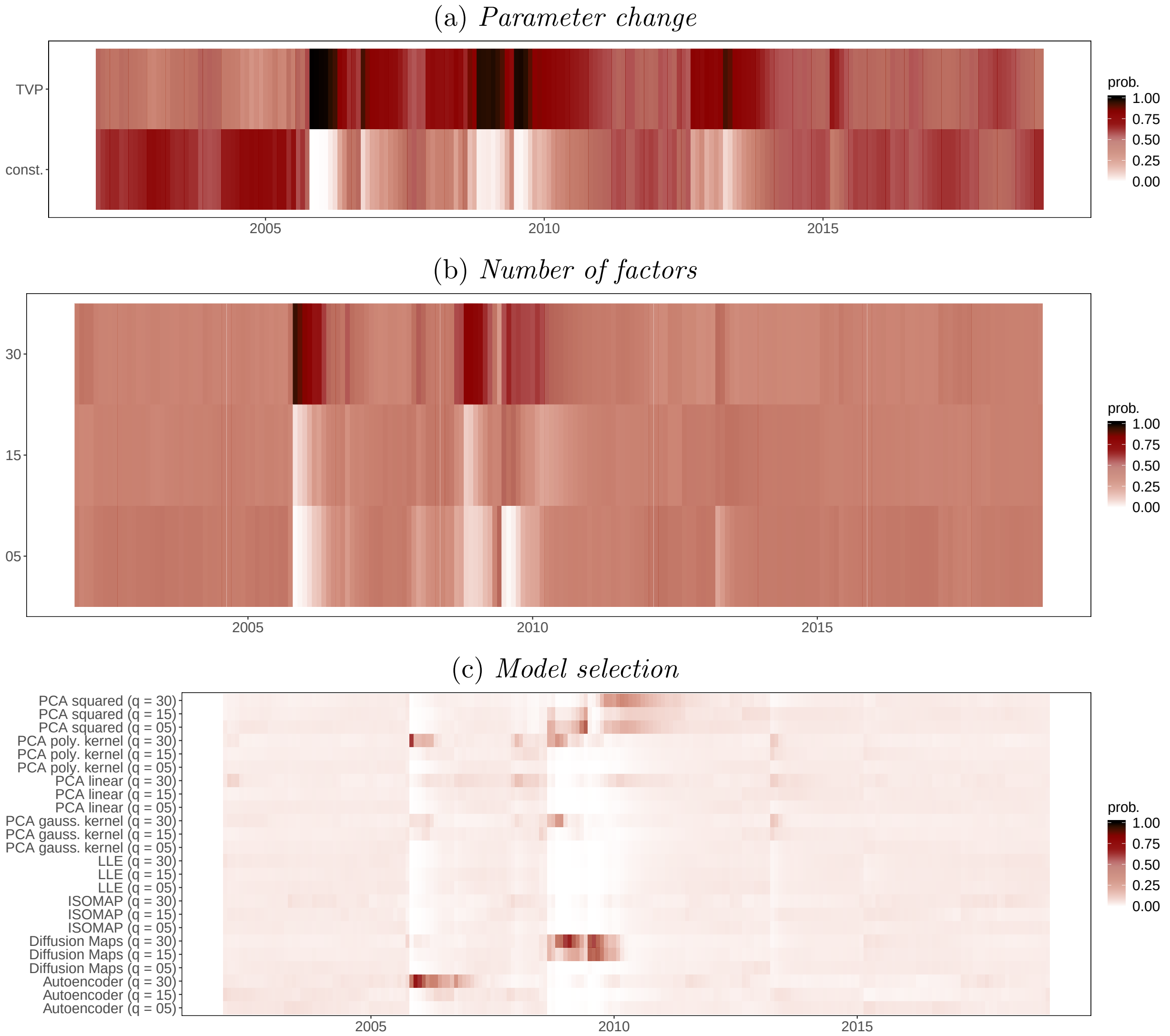
Comparing whether restricting the model a priori improves predictions yields mixed insights. For the one-month and one-quarter ahead predictions we find that a combination scheme that uses only TVP models but both priors and $q = 30$ factors yields the most precise forecasts. In the case of one-year ahead forecasts, we find that pooling across different q's and exclusively including constant parameter models translates into highest LPLs. In general, the differences in predictive performance across the DMA-based averaging schemes are small. Hence, as a general suggestion we can recommend applying DMA and using the most exhaustive model space available (i.e., including both priors, the different number of factors and TVP and constant parameter regressions).

To investigate which model receives substantial posterior weight over time, Figure 3 depicts the weights associated with the one-step ahead LPLs over the hold-out period. Panel (a) displays the weight placed on models that allow for TVP, panel (b) shows the weight attached to the different number of factors and panel (c) shows the weight attached to each model. These weights are obtained by using the full model space (i.e., that includes both priors, TVP and constant parameter regressions and all number of factors). The weight placed on TVP specifications, for instance, is then simply obtained by summing up the weights associated with the different models that feature TVPs.

Starting with the top panel of the figure, we observe that during the beginning of the sample, appreciable model weight is placed on constant parameter models. In the mid of 2006, this changes and DMA places increasing posterior mass on models that allow for time-variation in the parameters. In the period from the beginning of 2007 to the onset of the financial crisis, we see that the weight on TVP models somewhat decreases. During the financial crisis, we again experience a pronounced increase in posterior weight towards TVP regression. In that period, constant parameter models only play a limited role in forming inflation forecasts. With some few exceptions, the remainder of the hold-out period is characterized by evenly distributed posterior mass across constant and TVP regressions.

The middle panel of Figure 3 shows that DMA places increasing posterior mass on models with a large number of factors during recessions (and, similar to panel (a), in 2006). This indicates that in turbulent times it seems to pay off to include many factors. Since our previous analysis reveals that point forecasts are very similar to the ones obtained from simpler univariate models, this finding is most likely driven by a superior density forecasting performance. Hence, we conjecture that the main driving force behind the strong performance of a model with many factors is that this increases posterior uncertainty (through the inclusion of a large number of covariates), which ultimately leads to slightly wider credible sets, implying a higher probability of observing outlying observations.

The bottom panel (panel (c)) of Figure 3 provides information on how much weight is allocated to models that exploit non-linear dimension reduction techniques. Again, we observe that non-linear dimension reduction techniques obtain considerable posterior mass during 2006 and the financial crisis of 2007/2008. In 2006, the Autoencoder with $q = 15$ receives substantial posterior weight. During the financial crisis, we find that diffusion maps and squared PCA feature large weights. Apart from these two periods, weights allocated to non-linear dimension reduction techniques are generally close to zero.


(a) *Parameter change*

(b) *Number of factors*

(c) *Model selection*

Figure 3: Evolution of the weights determined by DMA for one-month ahead cumulative LPBFs.

This discussion highlights that the strong performance of DMA relative to the single best performing model can be, at least partly, attributed to changes in model weights across business cycles. In expansionary periods with stable inflation rates and macroeconomic fundamentals, linear and simple models dominate the model pool. By contrast, adding more sophisticated models and dimension reduction techniques pays off during recessions. A dynamic combination of different approaches thus improves real-time inflation forecasts.

5 Closing Remarks

In macroeconomics, the vast majority of researchers compresses information using linear methods such as principal components to efficiently summarize huge datasets in forecasting applications. Machine learning techniques describing large datasets with relatively few latent factors have gained relevance in the last years in various areas. In this paper, we have shown that using such approaches potentially improves real-time inflation forecasts for a wide range of competing model specifications. Our findings indicate that point forecasts of simpler models are hard to beat. But when interest centers on predictive distributions, we find that more sophisticated modeling techniques that rely on non-linear dimension reduction yield favorable inflation predictions. These predictions can be further improved by using DMA to dynamically weight different models, dimension reduction methods and priors. Doing so further improves density forecasts. Weights obtained from dynamic model averaging reveal that using TVP models in combination with non-linear approaches to dimension reduction is preferred in turbulent times.

References

Allaire, J., and F. Chollet (2019): *keras: R Interface to 'Keras'*, R package version 2.2.5.0.

Andreini, P., C. Izzo, and G. Ricco (2020): "Deep Dynamic Factor Models," *arXiv preprint arXiv:2007.11887*.

Atkeson, A., L. E. Ohanian, et al. (2001): "Are Phillips curves useful for forecasting inflation?," *Federal Reserve Bank of Minneapolis Quarterly Review*, 25(1), 2–11.

Bai, J., and S. Ng (2002): "Determining the number of factors in approximate factor models," *Econometrica*, 70(1), 191–221.

——— (2008): "Forecasting economic time series using targeted predictors," *Journal of Econometrics*, 146(2), 304–317.

Beckmann, J., G. Koop, D. Korobilis, and R. A. Schüssler (2020): "Exchange rate predictability and dynamic Bayesian learning," *Journal of Applied Econometrics*, 35(4), 410–421.

Belmonte, M., G. Koop, and D. Korobilis (2014): "Hierarchical shrinkage in time-varying coefficient models," *Journal of Forecasting*, 33(1), 80–94.

Berg, T. O., and S. R. Henzel (2015): "Point and density forecasts for the euro area using Bayesian VARs," *International Journal of Forecasting*, 31(4), 1067–1095.

Bernanke, B. S., J. Boivin, and P. Eliasz (2005): "Measuring the effects of monetary policy: A factor-augmented vector autoregressive (FAVAR) approach," *The Quarterly Journal of Economics*, 120(1), 387–422.

Carter, C., and R. Kohn (1994): "On Gibbs sampling for state space models," *Biometrika*, 81(3), 541–553.

Carvalho, C. M., N. G. Polson, and J. G. Scott (2010): "The horseshoe estimator for sparse signals," *Biometrika*, 97(2), 465–480.

Chakraborty, C., and A. Joseph (2017): "Machine learning at central banks," Bank of England Working Papers 674, Bank of England.

Chan, J. C. (2017): "The stochastic volatility in mean model with time-varying parameters: An application to inflation modeling," *Journal of Business & Economic Statistics*, 35(1), 17–28.

Chan, J. C., T. E. Clark, and G. Koop (2018): "A new model of inflation, trend inflation, and long-run inflation expectations," *Journal of Money, Credit and Banking*, 50(1), 5–53.

Clark, T. E. (2011): "Real-time density forecasts from Bayesian vector autoregressions with stochastic volatility," *Journal of Business & Economic Statistics*, 29(3), 327–341.

Clark, T. E., and M. W. McCracken (2010): "Averaging forecasts from VARs with uncertain instabilities," *Journal of Applied Econometrics*, 25(1), 5–29.

Clark, T. E., and F. Ravazzolo (2015): "Macroeconomic forecasting performance under alternative specifications of time-varying volatility," *Journal of Applied Econometrics*, 30(4), 551–575.

Coifman, R. R., and S. Lafon (2006): "Diffusion maps," *Applied and Computational Harmonic Analysis*, 21(1), 5–30.

Coifman, R. R., S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker (2005): "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps," *Proceedings of the National Academy of Sciences*, 102(21), 7426–7431.

Conflitti, C., C. De Mol, and D. Giannone (2015): "Optimal combination of survey forecasts," *International Journal of Forecasting*, 31(4), 1096–1103.

Coulombe, P. G., M. Leroux, D. Stevanovic, and S. Surprenant (2019): "How is machine learning useful for macroeconomic forecasting?," CIRANO Working Papers 2019s-22, CIRANO.

Croushore, D. (2011): "Frontiers of real-time data analysis," *Journal of Economic Literature*, 49(1), 72–100.

D'Agostino, A., L. Gambetti, and D. Giannone (2013): "Macroeconomic forecasting and structural change," *Journal of Applied Econometrics*, 28(1), 82–101.

De Mol, C., D. Giannone, and L. Reichlin (2008): "Forecasting using a large number of predictors: Is Bayesian shrinkage a valid alternative to principal components?," *Journal of Econometrics*, 146(2), 318–328.

Diedrich, H., and D. M. Abel (2012): *lle: Locally linear embedding*, R package version 1.1.

Exterkate, P., P. J. Groenen, C. Heij, and D. van Dijk (2016): "Nonlinear forecasting with many predictors using kernel ridge regression," *International Journal of Forecasting*, 32(3), 736–753.

Feng, G., J. He, and N. G. Polson (2018): "Deep learning for predicting asset returns," *arXiv preprint arXiv:1804.09314*.

Frühwirth-Schnatter, S. (1994): "Data augmentation and dynamic linear models," *Journal of Time Series Analysis*, 15(2), 183–202.

Frühwirth-Schnatter, S., and H. Wagner (2010): "Stochastic model specification search for Gaussian and partial non-Gaussian state space models," *Journal of Econometrics*, 154(1), 85–100.

Gallant, A. R., and H. White (1992): "On learning the derivatives of an unknown mapping with multilayer feedforward networks," *Neural Networks*, 5(1), 129–138.

George, E. I., and R. E. McCulloch (1993): "Variable selection via Gibbs sampling," *Journal of the American Statistical Association*, 88(423), 881–889.

George, E. I., D. Sun, and S. Ni (2008): "Bayesian stochastic search for VAR model restrictions," *Journal of Econometrics*, 142(1), 553–580.

Geweke, J., and G. Amisano (2010): "Comparing and evaluating Bayesian predictive distributions of asset returns," *International Journal of Forecasting*, 26(2), 216–230.

——— (2011): "Optimal prediction pools," *Journal of Econometrics*, 164(1), 130–141.

Giovannelli, A. (2012): "Nonlinear forecasting using large datasets: Evidences on US and Euro area economies," CEIS Research Paper 255, Tor Vergata University, CEIS.

Goodfellow, I., Y. Bengio, and A. Courville (2016): *Deep Learning*. MIT Press.

Hall, S. G., and J. Mitchell (2007): "Combining density forecasts," *International Journal of Forecasting*, 23(1), 1–13.

Hauzenberger, N., F. Huber, G. Koop, and L. Onorante (2019): "Fast and Flexible Bayesian Inference in Time-varying Parameter Regression Models," *arXiv preprint arXiv:1910.10779*.

Heaton, J. (2008): *Introduction to neural networks with Java*. Heaton Research, Inc.

Heaton, J. B., N. G. Polson, and J. H. Witte (2017): "Deep learning for finance: deep portfolios," *Applied Stochastic Models in Business and Industry*, 33(1), 3–12.

Hendry, D. F., and M. P. Clements (2004): "Pooling of forecasts," *The Econometrics Journal*, 7(1), 1–31.

Huang, G.-B. (2003): "Learning capability and storage capacity of two-hidden-layer feedforward networks," *IEEE Transactions on Neural Networks*, 14(2), 274–281.

Huber, F., G. Koop, and L. Onorante (2020): "Inducing sparsity and shrinkage in time-varying parameter models," *Journal of Business & Economic Statistics*, (forthcoming).

Huber, F., and M. Pfarrhofer (2020): "Dynamic shrinkage in time-varying parameter stochastic volatility in mean models," *Journal of Applied Econometrics*, (forthcoming).

Jarocinski, M., and M. Lenza (2018): "An inflation-predicting measure of the output gap in the Euro area," *Journal of Money, Credit and Banking*, 50(6), 1189–1224.

Kalli, M., and J. E. Griffin (2014): "Time-varying sparsity in dynamic regression models," *Journal of Econometrics*, 178(2), 779 – 793.

Kastner, G. (2016): "Dealing with stochastic volatility in time series using the R package stochvol," *Journal of Statistical Software*, 69(5), 1–30.

Kastner, G., and S. Frühwirth-Schnatter (2014): "Ancillarity-sufficiency interweaving strategy (ASIS) for boosting MCMC estimation of stochastic volatility models," *Computational Statistics & Data Analysis*, 76, 408–423.

Kayo, O. (2006): "LOCALLY LINEAR EMBEDDING ALGORITHM–Extensions and applications," .

Kelly, B. T., S. Pruitt, and Y. Su (2019): "Characteristics are covariances: A unified model of risk and return," *Journal of Financial Economics*, 134(3), 501–524.

Koop, G., and D. Korobilis (2012): "Forecating inflation using dynamic model averaging," *International Economic Review*, 53(3), 867–886.

——— (2013): "Large time-varying parameter VARs," *Journal of Econometrics*, 177(2), 185–198.

Koop, G., and S. M. Potter (2007): "Estimation and forecasting in models with multiple breaks," *The Review of Economic Studies*, 74(3), 763–789.

Lin, F., C.-C. Yeh, and M.-Y. Lee (2011): "The use of hybrid manifold learning and support vector machines in the prediction of business failure," *Knowledge-Based Systems*, 24(1), 95–101.

Makalic, E., and D. F. Schmidt (2015): "A simple sampler for the horseshoe estimator," *IEEE Signal Processing Letters*, 23(1), 179–182.

McAdam, P., and P. McNelis (2005): "Forecasting inflation with thick models and neural networks," *Economic Modelling*, 22(5), 848–867.

McCracken, M. W., and S. Ng (2016): "FRED-MD: A monthly database for macroeconomic research," *Journal of Business & Economic Statistics*, 34(4), 574–589.

Medeiros, M. C., G. F. Vasconcelos, Á. Veiga, and E. Zilberman (2019): "Forecasting inflation in a data-rich environment: the benefits of machine learning methods," *Journal of Business & Economic Statistics*, (forthcoming).

Mullainathan, S., and J. Spiess (2017): "Machine learning: An applied econometric approach," *Journal of Economic Perspectives*, 31(2), 87–106.

Oksanen, J., F. G. Blanchet, M. Friendly, R. Kindt, P. Legendre, D. McGlinn, P. R. Minchin, R. B. O'Hara, G. L. Simpson, P. Solymos, M. H. H. Stevens, E. Szoecs, and H. Wagner (2019): *vegan: Community Ecology Package*, R package version 2.5-6.

Orsenigo, C., and C. Vercellis (2013): "Linear versus nonlinear dimensionality reduction for banks' credit rating prediction," *Knowledge-Based Systems*, 47, 14–22.

Pettenuzzo, D., and F. Ravazzolo (2016): "Optimal portfolio choice under decision-based model combinations," *Journal of Applied Econometrics*, 31(7), 1312–1332.

Pfarrhofer, M. (2020): "Forecasts with Bayesian vector autoregressions under real time conditions," *arXiv preprint arXiv:2004.04984*.

Polson, N. G., and J. G. Scott (2010): "Shrink globally, act locally: Sparse Bayesian regularization and prediction," *Bayesian Statistics*, 9, 501–538.

Raftery, A., M. Kárný, and P. Ettler (2010): "Online prediction under model uncertainty via Dynamic Model Averaging: Application to a cold rolling mill," *Technometrics*, 52(1), 52–66.

Ribeiro, B., A. Vieira, and J. C. das Neves (2008): "Supervised Isomap with dissimilarity measures in embedding learning," in *Iberoamerican Congress on Pattern Recognition*, pp. 389–396. Springer.

Richards, J., and R. Cannoodt (2019): *diffusionMap: Diffusion Map*, R package version 1.2.0.

Richards, J. W., P. E. Freeman, A. B. Lee, and C. M. Schafer (2009): "Exploiting low-dimensional structure in astronomical spectra," *The Astrophysical Journal*, 691(1), 32–42.

Roweis, S. T., and L. K. Saul (2000): “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, 290(5500), 2323–2326.

Saxe, A. M., Y. Bansal, J. Dapello, M. Advani, A. Kolchinsky, B. D. Tracey, and D. D. Cox (2019): “On the information bottleneck theory of deep learning,” *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12), 124020.

Schölkopf, B., A. Smola, and K.-R. Müller (1998): “Nonlinear component analysis as a kernel eigenvalue problem,” *Neural Computation*, 10(5), 1299–1319.

Stella, A., and J. H. Stock (2013): “A state-dependent model for inflation forecasting,” *FRB International Finance Discussion Paper*, (1062).

Stock, J., and M. Watson (1999): “Forecasting inflation,” *Journal of Monetary Economics*, 44(2), 293–335.

——— (2002): “Macroeconomic forecasting using diffusion indexes,” *Journal of Business & Economic Statistics*, 20(2), 147–162.

——— (2008): “Phillips curve inflation forecasts,” NBER Working Papers 14322, National Bureau of Economic Research, Inc.

Stock, J. H., and M. W. Watson (2007): “Why has U.S. inflation become harder to forecast?,” *Journal of Money, Credit and Banking*, 39(s1), 3–33.

Stock, J. H., and M. W. Watson (2016): “Core inflation and trend inflation,” *Review of Economics and Statistics*, 98(4), 770–784.

Taylor, S. J. (1982): “Financial returns modelled by the product of two stochastic processes-a study of the daily sugar prices 1961-75,” *Time Series Analysis: Theory and Practice*, 1, 203–226.

Tenenbaum, J. B., V. De Silva, and J. C. Langford (2000): “A global geometric framework for nonlinear dimensionality reduction,” *Science*, 290(5500), 2319–2323.

Timmermann, A. (2006): “Forecast combinations,” *Handbook of Economic Forecasting*, 1, 135–196.

Zelnik-Manor, L., and P. Perona (2004): “Self-tuning spectral clustering,” *Advances in neural information processing systems*, 17, 1601–1608.

Zime, S. (2014): “Economic performance evaluation and classification using hybrid manifold learning and support vector machine model,” in *2014 11th International Computer Conference on Wavelet Actiev Media Technology and Information Processing (ICCWAMTIP)*. IEEE, 184–191.

Appendices

A Technical Appendix

A.1 Non-centered Parameterization

To implement the Bayesian priors to achieve shrinkage in the TVP regression defined by Eq. 18 and Eq. 19, we use the non-centered parameterization proposed in Frühwirth-Schnatter and Wagner (2010). Intuitively speaking, this allows us to move the process innovation variances into the observation equation and discriminate between a time-invariant and a time-varying part of the model. The non-centered parameterization of the model is given by:

$$y_{t+h} = \boldsymbol{d}'_{t+h}\boldsymbol{\beta}_0 + \boldsymbol{d}'_{t+h}\sqrt{\boldsymbol{V}}\tilde{\boldsymbol{\beta}}_{t+h} + \epsilon_{t+h}, \quad \epsilon_{t+h} \sim \mathcal{N}(0, \sigma^2_{t+h}) \tag{B.1}$$

$$\tilde{\boldsymbol{\beta}}_{t+h} = \tilde{\boldsymbol{\beta}}_{t+h-1} + \varepsilon_{t+h}, \quad \varepsilon_{t+h} \sim \mathcal{N}(0, \boldsymbol{I}_M), \quad \tilde{\boldsymbol{\beta}}_0 = \boldsymbol{0}_M, \tag{B.2}$$

where the jth element in $\tilde{\boldsymbol{\beta}}_{t+h}$ is given by $\tilde{\boldsymbol{\beta}}_{jt+h} = \frac{\beta_{jt+h} - \beta_{j0}}{\sqrt{v_j}}$ for $j = 1, \dots, M$.

Conditional on the normalized states $\tilde{\boldsymbol{\beta}}$, Eq. B.1 can be written as a linear regression model as follows:

$$y_{t+h} = \boldsymbol{D}'_{t+h}\boldsymbol{\alpha} + \epsilon_{t+h}, \tag{B.3}$$

with $\boldsymbol{D}_{t+h} = [\boldsymbol{d}'_{t+h}, (\tilde{\boldsymbol{\beta}}_{t+h} \odot \boldsymbol{d}_{t+h})']'$ denoting a $2M$-dimensional vector of regressors and $\boldsymbol{\alpha} = (\boldsymbol{\beta}'_0, v_1, \dots, v'_M)$ is a $2M$-dimensional coefficient vector. This parameterization implies that the state innovation variances (or more precisely the square roots) are moved into the observation equation and we can estimate them alongside $\boldsymbol{\beta}_0$ (conditional on the states $\tilde{\boldsymbol{\beta}}_{t+h}$).

A.2 Prior Setup

A.2.1 Priors on the Regression Coefficients

We use a zero-mean multivariate Gaussian prior on $\boldsymbol{\alpha}$:

$$\boldsymbol{\alpha}|\underline{\boldsymbol{V}} \sim \mathcal{N}(\boldsymbol{0}, \underline{\boldsymbol{V}}), \tag{B.4}$$

with $\underline{\boldsymbol{V}}$ denoting a $2M$-dimensional prior variance-covariance matrix $\underline{\boldsymbol{V}} = \text{diag}\left(\tau_1^2, \dots, \tau_{2M}^2\right)$. This matrix collects the prior shrinkage parameters τ_j associated with the time-invariant regression coefficients and the process innovation standard deviations.

In the empirical work, the priors we consider differ in the specification of $\underline{\boldsymbol{V}}$. The first is the stochastic search variable selection (SSVS) prior of George and McCulloch (1993) and the second the Horseshoe (HS) prior of Carvalho et al. (2010).

1. **SSVS Prior:**
   The SSVS prior pushes coefficients associated with irrelevant variables towards zero by using a mixture of Gaussians. A specific mixture component is selected by introducing an auxiliary binary indicator variable γ_j. More formally, the SSVS prior specifies τ_j^2 $(j = 1, \dots, 2M)$ such that

   $$\tau_j^2 = (1 - \gamma_j)\underline{\tau}_{0j}^2 + \gamma_j \underline{\tau}_{1j}^2, \tag{B.5}$$

   with $\underline{\tau}_{0j} \ll \underline{\tau}_{1j}$ being fixed prior variances. If $\gamma_j = 1$, the prior variance is $\underline{\tau}_{1j}$ which is set to a large value. Hence, little shrinkage is introduced. By contrast, if $\gamma_j = 0$, the prior variance $\underline{\tau}_{0j}$ is close to zero and the corresponding prior weight will be large, leading to a posterior distribution that is tightly centered on zero.

The prior probability that $\gamma_j = 1$ is set equal to:

$$\text{Prob}(\gamma_j = 1) = 1 - \text{Prob}(\gamma_j = 0) = p_m, \quad p_m = \frac{1}{2}. \tag{B.6}$$

This choice of the prior inclusion probability implies that every quantity is equally likely to enter the model.

To control for scaling differences, we adopt the semi-automatic approach proposed in George et al. (2008) and choose $\overline{\tau}^2_{0j} = 0.01\ \hat{\sigma}_j$ and $\overline{\tau}^2_{0j} = 100\ \hat{\sigma}_j$ for $j = 1, \ldots, M$. Here, $\hat{\sigma}^2_j$ denotes the OLS variance of a standard regression model with constant parameters.

2. **Horseshoe Prior:**

   The horseshoe prior of Carvalho et al. (2010) achieves shrinkage by introducing local and global shrinkage parameters (see Polson and Scott, 2010). These follow a standard half-Cauchy distribution restricted to the positive real numbers. That is:

$$\tau^2_j = \zeta^2_j \varsigma^2, \quad \zeta_m \sim \mathcal{C}^+(1, 0), \quad \varsigma \sim \mathcal{C}^+(1, 0) \tag{B.7}$$

   While the global component ς strongly pushes all coefficients in $\boldsymbol{\alpha}$ towards the prior mean (i.e., zero), the local scalings $\{\zeta_j\}_{j=1}^{2M}$ allow for variable-specific departures from zero in light of a global scaling parameter close to zero. This flexibility leads to heavy tails in the marginal prior (obtained after integrating out ζ_j) which turns out to be useful for forecasting.

A.3 Full Conditional Posterior Simulation

We carry out posterior inference by using a Markov chain Monte Carlo (MCMC) algorithm to simulate from the joint posterior of the parameters, the log-volatilities and the TVPs. This MCMC algorithm consists of the following steps:

1. Conditional on the time-varying part of the coefficients and the stochastic volatilities, we draw $(\boldsymbol{\beta}_0, v_1, \ldots, v_M)'$ from $\mathcal{N}(\overline{\boldsymbol{\beta}}, \overline{\boldsymbol{V}})$ with $\overline{\boldsymbol{V}} = (\tilde{\boldsymbol{D}}'\tilde{\boldsymbol{D}} + \underline{\boldsymbol{V}}^{-1})^{-1}$ and $\underline{\boldsymbol{\beta}} = \overline{\boldsymbol{V}}(\tilde{\boldsymbol{D}}\tilde{\boldsymbol{y}})$. $\tilde{\boldsymbol{y}}$ is a $T-$dimensional vector with typical element y_t/σ_t and $\tilde{\boldsymbol{D}}$ is a $T \times (2M)$ matrix with typical row $\boldsymbol{D}_t/\sigma_t$.

2. Controlling for all other model parameters, the full history of $\tilde{\boldsymbol{\beta}}_{t+h}$ is sampled using the forward-filtering backward-sampling (FFBS) algorithm proposed by Carter and Kohn (1994); Frühwirth-Schnatter (1994). For constant parameter models this step is skipped.

3. The stochastic volatilities $\log \sigma^2_{t+h}$ are drawn by employing the algorithm of Kastner and Frühwirth-Schnatter (2014) implemented in the `stochvol` R-package of Kastner (2016).

4. Sampling the diagonal elements of $\boldsymbol{V}$ depends on the specific prior setup chosen.

   - If the SSVS prior is used, we simulate the indicators γ_j from a Bernoulli distribution with the probability that $\gamma_j = 1$ given by

$$\text{Prob}(\gamma_j = 1|\beta_j) = \frac{\overline{u}_{1j}}{\overline{u}_{0j} + \overline{u}_{1j}}$$
$$\overline{u}_{1j} = \tau^{-1}_{1m} \exp\left\{-\frac{\beta^2_j}{2\tau^2_{1j}}\right\} \times p_m$$
$$\overline{u}_{0j} = \tau^{-1}_{0m} \exp\left\{-\frac{\beta^2_j}{2\tau^2_{0j}}\right\} \times (1 - p_m).$$

- In case we adopt the HS prior, we rely on the hierarchical representation of Makalic and Schmidt (2015). Introducing auxiliary random quantities which follow an inverse Gamma distribution we can draw ζ_j and ς as follows:

$$
\begin{aligned}
\zeta_j|\beta_j, \varsigma, \eta &\sim \mathcal{G}^{-1}\left(1, \eta_j^{-1} + \frac{\beta_j^2}{2\varsigma}\right)\\
\varsigma|\beta_j, \zeta_j, \varphi &\sim \mathcal{G}^{-1}\left(\frac{2M+1}{2}, \varphi^{-1} + \frac{1}{2}\sum_{j=1}^{2M} \beta_j^2 \zeta_j^{-1}\right)\\
\eta_j|\zeta_j &\sim \mathcal{G}^{-1}\left(1, 1+\zeta_j^{-1}\right),\\
\varphi|\varsigma &\sim \mathcal{G}^{-1}\left(1, 1+\varphi^{-1}\right)
\end{aligned}
$$

We sample from the relevant full conditional posterior distributions iteratively. This is repeated $10,000$ times and the first $2,000$ draws are discarded as burn-in.

B Data Appendix

The Federal Reserve Economic Data (FRED) contains monthly observations of macroeconomic variables for the US and is available for download at https://research.stlouisfed.org. Details on the dataset can be found in McCracken and Ng (2016) . For each data vintage (available from 1999:08), the time series start from January 1959. Due to missing values in some of the series, we preselect 105 variables and transform them according to Table C.1. We select all variables for our models except for the extended Phillips curve, where we choose the variables indicated by column *PART*.

Table C.1: Data description

FRED.Mnemonic	Description	Trans I(0)	PART	FULL
RPI	Real personal income	5		x
W875RX1	Real personal income ex transfer receipts	5	x	x
INDPRO	IP Index	5	x	x
IPFPNSS	IP: Final Products	5		x
IPFINAL	IP: Final Products (Market Group)	5		x
IPCONGD	IP: Consumer Goods	5		x
IPMAT	IP: Materials	5		x
IPMANSICS	IP: Manufacturing (SIC)	5		x
CUMFNS	Capacity Utilization: Manufacturing	2	x	x
CLF16OV	Civilian Labor Force	5		x
CE16OV	Civilian Employment	5		x
UNRATE	Civilian Unemployment Rate	2	x	x
UEMPMEAN	Average Duration of Unemployment (Weeks)	2		x
UEMPLT5	Civilians Unemployed : Less Than 5 Weeks	5		x
UEMP5TO14	Civilians Unemployed for 5-14 Weeks	5		x
UEMP15OV	Civilians Unemployed : 15 Weeks & Over	5		x
UEMP15T26	Civilians Unemployed for 15-26 Weeks	5		x
UEMP27OV	Civilians Unemployed for 27 Weeks and Over	5		x
CLAIMSx	Initial Claims	5	x	x
PAYEMS	All Employees: Total nonfarm	5	x	x
USGOOD	All Employees: Goods-Producing Industries	5		x
CES1021000001	All Employees: Mining and Logging: Mining	5		x
USCONS	All Employees: Construction	5		x
MANEMP	All Employees: Manufacturing	5		x
DMANEMP	All Employees: Durable goods	5		x
NDMANEMP	All Employees: Nondurable goods	5		x
SRVPRD	All Employees: Service-Providing Industries	5		x
USWTRADE	All Employees: Wholesale Trade	5		x
USTRADE	All Employees: Retail Trade	5		x
USFIRE	All Employees: Financial Activities	5		x
USGOVT	All Employees: Government	5		x
CES0600000007	Avg Weekly Hours: Goods-Producing	1	x	x
AWOTMAN	Avg Weekly Overtime Hourse: Manufacturing	2		x
AWHMAN	Avg Weekly Hours: Manufacturing	1		x
CES0600000008	Avg Hourly Earnings: Goods-Producing	6	x	x
CES2000000008	Avg Hourly Earnings: Construction	6		x
CES3000000008	Avg Hourly Earnings: Manufacturing	6		x
HOUST	Housing Starts: Total New Privately Owned	4	x	x
HOUSTNE	Housing Starts, Northeast	4		x
HOUSTMW	Housing Starts, Midwest	4		x
HOUSTS	Housing Starts, South	4		x
HOUSTW	Housing Starts, West	4		x
PERMIT	New Private Housing Permits (SAAR)	4		x
PERMITNE	New Private Housing Permits, Northeast (SAAR)	4		x
PERMITMW	New Private Housing Permits, Midwest (SAAR)	4		x
PERMITS	New Private Housing Permits, South (SAAR)	4		x
PERMITW	New Private Housing Permits, West (SAAR	4		x
CMRMTSPLx	Real Manu. and TradeIndustries Sales	5	x	x
RETAILx	Retail and Food Services Sales	5		x
AMDMNOx	New Orders for Durable goods	5		x
ANDENOx	New Orders for Nondefense Capital goods	5		x
AMDMUOx	Unfilled Orders for Durable goods	5		x
BUSINVx	Total Business Inventories	5	x	x
ISRATIOx	Total Business: Inventories to Sales Ratio	2		x
UMCSENTx	Consumer Sentiment Index	2		x
OILPRICEx	Crude Oil, , spliced WTI and Cushing	6		x
PPICMM	PPI: Metals and metal products	6	x	x
CPIAUCSL	CPI : All Items	6		x
CPIAPPSL	CPI : Apparel	6		x
CPITRNSL	CPI : Transportation	6		x

Data description (cont.)

FRED.Mnemonic	Description	Trans I(0)	PART	FULL
CPIMEDSL	CPI : Medical Care	6		x
CUSR0000SAC	CPI : Commodities	6		x
CUSR0000SAS	CPI : Services	6		x
CPIULFSL	CPI : All Items Less Food	6		x
CUSR0000SA0L5	CPI : All Items Less Medical Care	6		x
FEDFUNDS	Effective Federal Funds Rate	2	x	x
M1SL	M1 Money Stock	6		x
M2SL	M2 Money Stock	6		x
M2REAL	Real M2 Money Stock	5	x	x
AMBSL	St. Louis Adjusted Monetary Base	6		x
TOTRESNS	Total Reserves of Depository Institutions	6		x
NONBORRES	Reserves of Depository Institutions	7		x
BUSLOANS	Commercial and Industrial Loans	6	x	x
REALLN	Real Estate Loans at All Commerical Banks	6	x	x
NONREVSL	Total Nonrevolving Credit	6		x
CONSPI	Nonrevolving consumer credit to Personal Income	2		x
MZMSL	MZM Money Stock	6		x
DTCOLNVHFNM	Consumer Motor Vehicle Loans Outstanding	6		x
DTCTHFNM	Total Consumer Total Consumer Loans and Leases Outstanding	6		x
INVEST	Securities in Bank Credit at All Commercial Banks	6		x
CP3Mx	3-Month AA Financial Commercial Paper Rate	2		x
TB3MS	3-Month Treasury Bill	2	x	x
TB6MS	6-Month Treasury Bill	2		x
GS1	1-Year Treasury Rate	2		x
GS5	5-Year Treasury Rate	2		x
GS10	10-Year Treasury Rate	2	x	x
AAA	Moody's Seasoned Aaa Corporate Bond Yield	2		x
BAA	Moody's Seasoned Baa Corporate Bond Yield	2		x
COMPAPFFx	3-Month Commercial Paper Minus FEDFUNDS	1		x
TB3SMFFM	3-Month Treasury C Minus FEDFUNDS	1		x
TB6SMFFM	6-Month Treasury C Minus FEDFUNDS	1		x
T1YFFM	1-Year Treasury C Minus FEDFUNDS	1		x
T5YFFM	5-Year Treasury C Minus FEDFUNDS	1		x
T10YFFM	10-Year Treasury C Minus FEDFUNDS	1		x
AAAFFM	Moody's Aaa Corporate Bond Minus FEDFUNDS	1		x
BAAFFM	Moody's Baa Corporate Bond Minus FEDFUNDS	1		x
TWEXMMTH	Trade Weighted Trade Weighted U.S. Dollar Index: Major Currencies	5		x
EXSZUSx	Switzerland / U.S. Foreign Exchange Rate	5	x	x
EXJPUSx	Japan / U.S. Foreign Exchange Rate	5		x
EXUSUKx	U.S. / UK Foreign Exchange Rate	5		x
EXCAUSx	Canada / U.S. Foreign Exchange Rate	5		x
S.P.500	S&Ps Common Stock Price Index: Composite	5	x	x
S.P..indust	S&Ps Common Stock Price Index: Industrials	5		x
S.P.div.yield	S&Ps Composite Common Stock: Dividend Yield	2		x
S.P.PE.ratio	S&Ps Composite Common Stock: Price-Earnings Ratio	5		x

Note: Column **Trans I(0)** denotes the transformation of each time series to achieve approximate stationarity: (1) no transformation, (2) Δx_t, (4) $log(x_t)$, (5) $\Delta log(x_t)$, (6) $\Delta^2 log(x_t)$, (7) $\Delta(x_t/x_{t-1} - 1.0)$