Implied Factors: The Linear Skeleton of Machine Learning Forecasts*

Philippe Goulet Coulombe†
Université du Québec à Montréal

Karin Klieber‡
European Central Bank, OeNB

Gabriel Rodriguez-Rondon§
Bank of Canada

February 28, 2026

Preliminary and incomplete.
Please do not circulate without permission.

Abstract

For interpretability, sparsity is key and linearity is preferable. Yet neither is common in machine-learning-based macroeconomic forecasting, where large predictor sets are fed into nonlinear, nonparametric models. We show that any supervised learning algorithm whose predictions are linear in the target variable—regardless of nonlinearities in the features—admits an exact linear factor representation. We call the resulting components the model's implied factors. For example, the forecasts of a random forest can be replicated exactly by a linear regression on its implied factors. Studying these factors provides a new route to uncover low-dimensional structure, even when the model was not built with such structure in mind. Empirically, a small number of implied factors suffices to reproduce out-of-sample forecasts of U.S. GDP, inflation, and unemployment. Across models, these factors cluster into a few common signals, suggesting that differences are largely quantitative rather than conceptual.

*The views expressed in this paper do not necessarily reflect those of the European Central Bank, the Oesterreichische Nationalbank, the Eurosystem, nor the Bank of Canada. This draft: March 1, 2026.

†Département des Sciences Économiques, goulet_coulombe.philippe@uqam.ca.
‡Monetary Policy Section, karin.klieber@oenb.at.
§Canadian Economic Analysis Department, grodriguezrondon@bankofcanada.ca.

1 Introduction

Many machine learning models—including regularized linear regressions, tree-based ensembles, and neural networks—yield out-of-sample forecasts that are *linear* in the training outcomes (Goulet Coulombe et al., 2024). That is, each model's predictions can be written as a matrix (a cross-sample hat matrix) applied to $\boldsymbol{y}_{\text{train}}$. As noted in Goulet Coulombe (2025b), the ordinary least squares hat matrix admits a factorization into orthonormal components that reveals OLS as a proximity-based estimator computing inner products in a transformed regressor space. We extend this observation to general machine learning models: the eigendecomposition of any such hat matrix yields a set of *implied factors*—latent predictors implied by the model's weighting of observations. The pipeline is straightforward: (i) recover the forecast weights following the approach of Goulet Coulombe et al. (2024), (ii) stack them into a matrix, and (iii) run the eigendecomposition. The key property is that, when these implied factors are used as generated regressors in OLS, they *exactly* reproduce the forecast of the original machine learning model. In other words, every such model admits a linear regression representation.

Even more interestingly, in practice only a small number of implied factors is needed, sometimes as few as two or three. So, the machine learning forecast can often be reproduced by a low-dimensional linear regression with a sparse set of predictors. When these factors can be named through time series visualization—and many of them can—we have at our disposal more compelling stories to tell than those offered by overcrowded variable importance plots. This is particularly appealing in high-dimensional settings, where individual variable contributions are microscopic, but where meaningful structure can emerge through aggregation into interpretable latent components.

The framework also opens several additional avenues. Standard information criteria or Lasso can select a parsimonious factor subset, further trimming the linear representation while preserving (or even improving) out-of-sample performance. The post-regression can be realigned to recent data to account for structural change, and implied factors from different models can be clustered into common signals, revealing the degree of agreement in the latent information exploited by competing methods.

Empirical Results. We apply the framework to quarterly forecasting of three U.S. macroeconomic aggregates—real GDP growth, CPI inflation, and the unemployment rate—using ridge regression, kernel ridge regression, random forests, gradient-boosted trees, and neural networks. The machine learning models generally outperform an AR(4) benchmark, often substantially so, and implied-factor specifications retaining as few as two or three factors reproduce the full models' out-of-sample accuracy. Implied factors also compare favorably to the leading supervised dimension reduction methods (PCR, PLS, sPCA, SPCA, SsPCA), with mixed-factor

specifications that pool across models appearing particularly promising.

Clustering implied factors across models reveals that, despite differences in architecture and estimation, the extracted signals organize into a small number of common economic themes. For real GDP, the leading factor of tree-based models resembles a Markov-switching intercept (Hamilton, 1989)—positive during expansions and dropping sharply during recessions—while the neural network produces a smoother rendition evoking an output gap. For CPI inflation, the dominant signal tracks low-frequency, regime-level movements in prices, resembling deviations from anchored inflation expectations; an additional factor captures the real-activity channel. For the unemployment rate, a single dominant business-cycle factor accounts for the bulk of predictive content across models. In all three cases, cross-model differences are largely quantitative—in amplitude and persistence—rather than conceptual.

Finally, implied factors benchmark well against supervised factor methods. The K=3 implied-factor specifications frequently match or outperform PCR, PLS, sPCA, and their supervised variants, and mixed-factor specifications that pool implied factors across all five models rank among the top performers.

Related Literature. Our starting observation—that machine learning forecasts are linear in the training outcomes—is shared with Goulet Coulombe et al. (2024), who exploit this property to decompose individual predictions into weighted combinations of historical episodes. The two papers tackle the same underlying problem from complementary angles: when the predictor set is large, individual variable importance is diffuse and often meaningless. Goulet Coulombe et al. (2024) respond by shifting attention to *observations*—asking which historical episodes matter most for a given forecast. The present paper instead moves to a low-dimensional *latent variable space*, extracting a small number of implied factors from the cross-sample hat matrix that summarize the model's entire predictive representation. In fact, the two perspectives are tightly intertwined: the historical weighting studied in Goulet Coulombe et al. (2024) is, by construction, an inner-product-based proximity metric computed in the space of implied factors. The observation weights *are* the Gram matrix of the factors introduced here.

Interpretability of machine learning. The growing use of machine learning in economics has spurred demand for interpretability tools, broadly organized along two axes (Goulet Coulombe et al., 2024). Feature-based methods such as SHAP (Lundberg and Lee, 2017) and LIME (Ribeiro et al., 2016) attribute predictions to individual covariates; however, in high-dimensional settings with correlated predictors, the resulting attributions are often unstable and hard to interpret economically. Instance-based methods take a complementary approach: data Shapley (Ghorbani and Zou, 2019; Jia et al., 2019), influence functions (Koh and Liang, 2017), and TracIn (Pruthi et al., 2020) assess the contribution of individual training observations to a given prediction.

Implied factors offer a third route: a low-dimensional latent space that is neither about single variables nor about single observations, but about the predictive directions that collectively drive the model's forecast.

Factor models and dimension reduction. A large literature uses latent factors for forecasting and inference. Stock and Watson (2002) introduce principal component regression (PCR), while Bai and Ng (2002) develop information criteria for selecting the number of factors. Because PCR factors are unsupervised, they need not align with the target; several extensions incorporate target information directly. Bair et al. (2006) propose supervised PCA, Kelly and Pruitt (2013) develop partial least squares (PLS) through the three-pass regression filter, and Huang et al. (2022) introduce scaled PCA (sPCA), which reweights predictors by their marginal correlation with the target. Hounyo and Li (2025) extend this to supervised scaled PCA (SsPCA), and Gao and Tsay (2024) propose supervised dynamic PCA, which scales predictors and their lags by forecasting significance before extracting factors.

On the nonlinear side, autoencoders learn compressed representations by minimizing reconstruction error (Hinton and Salakhutdinov, 2006). Gu et al. (2021) apply this framework to asset pricing, while Hauzenberger et al. (2023) use autoencoders for nonlinear factor extraction in macroeconomic forecasting, finding gains especially during recessions. Because autoencoders are trained to reconstruct $\boldsymbol{X}$, their factors capture feature-space variation, not necessarily variation relevant for predicting y. Implied factors differ from all of these approaches in a fundamental way: rather than constructing factors from the predictor covariance matrix, possibly after reweighting, screening, or nonlinear encoding, they are extracted from the eigendecomposition of a model's cross-sample hat matrix, and thus inherit whatever structure the underlying model has learned. We treat supervised PCA-type procedures as our closest benchmarking family, while emphasizing that implied factors are model-implied rather than a new member of the supervised PCA class.

Inherently interpretable models. A related strand of research designs machine learning models that produce interpretable latent variables by construction. Goulet Coulombe (2025a) designs a neural network whose architecture mirrors the Phillips Curve, so that each output neuron maps to an interpretable economic component. Bianchi et al. (2021) adopt a similar strategy for bond return prediction, routing distinct groups of macro variables through separate subnetworks that interact only at the output layer. Goulet Coulombe (2024) use random forests to estimate time-varying parameters in a linear macro equation, nesting threshold and smooth transition models. Farrell et al. (2021) embed deep networks inside structural micro models to recover heterogeneous treatment effects with valid inference. Fan et al. (2022) embed deep neural networks within a conditional factor model for asset pricing. All of these approaches achieve interpretability by imposing structure at the model design stage. Our framework is

complementary: implied factors are a post-hoc decomposition that applies to *any* model whose predictions are linear in the training outcomes, without requiring architectural choices tailored to a specific interpretation. In other words, it is a post-hoc interpretation technique that is fully compatible with traditional, off-the-shelf machine learning algorithms.

Outline. Section 2 introduces the cross-sample hat matrix and implied factors. Section 3 describes the forecasting design, assesses out-of-sample performance, examines how many implied factors are needed to reproduce full-model forecasts, clusters implied signals across models, and compares to traditional factor methods. Section 4 concludes. Additional methodological details, testing procedures, and supplementary empirical results are reported in the Appendix.

2 Implied Factors

In this section, we begin by reviewing the equivalence between ordinary least squares (OLS) predictions and similarity-based estimators defined through inner products in an orthonormal regressor space, as presented in Goulet Coulombe (2025b). We then show how this perspective extends naturally to any prediction engine that is linear in the outcome variables. Finally, we discuss the implications of this framework for empirical machine learning practice, particularly in applied macroeconomics and forecasting.

2.1 The OLS Case

We begin with OLS because it provides a transparent and self-contained illustration of the implied-factor decomposition; the general case follows in Section 2.2. Let $\boldsymbol{X}_{\text{train}} \in \mathbb{R}^{N \times P}$ denote the training design matrix, $\boldsymbol{y} \in \mathbb{R}^{N}$ the response vector, and $\boldsymbol{X}_{\text{test}} \in \mathbb{R}^{J \times P}$ the out-of-sample features. The standard OLS prediction is

$$\hat{\boldsymbol{y}}_{\text{test}} = \boldsymbol{X}_{\text{test}}(\boldsymbol{X}'_{\text{train}}\boldsymbol{X}_{\text{train}})^{-1}\boldsymbol{X}'_{\text{train}}\boldsymbol{y}. \tag{1}$$

As noted in Goulet Coulombe (2025b), this familiar expression can be reinterpreted as a proximity-based estimator. Using the eigendecomposition of $(\boldsymbol{X}'_{\text{train}}\boldsymbol{X}_{\text{train}})^{-1}$, we rewrite (1) as

$$\hat{\boldsymbol{y}}_{\text{test}} = \boldsymbol{X}_{\text{test}}\boldsymbol{U}\Lambda^{-1}\boldsymbol{U}'\boldsymbol{X}_{\text{train}}\boldsymbol{y}_{\text{train}}, \tag{2}$$

where $\boldsymbol{U}\Lambda^{-1}\boldsymbol{U}'$ is the eigendecomposition of $(\boldsymbol{X}'_{\text{train}}\boldsymbol{X}_{\text{train}})^{-1}$, with Λ being the diagonal matrix of eigenvalues of $\boldsymbol{X}'_{\text{train}}\boldsymbol{X}_{\text{train}}$, and $\boldsymbol{U}$ its eigenvectors. Decomposing the Λ into $\Lambda^{-\frac{1}{2}}\Lambda^{-\frac{1}{2}}$, we

obtain:

$$\hat{\boldsymbol{y}}_{\text{test}} = \underbrace{\boldsymbol{X}_{\text{test}}\boldsymbol{U}\Lambda^{-\frac{1}{2}}}_{\boldsymbol{F}_{\text{test}}} \underbrace{\Lambda^{-\frac{1}{2}}\boldsymbol{U}'\boldsymbol{X}'_{\text{train}}}_{\boldsymbol{F}'_{\text{train}}} \boldsymbol{y}_{\text{train}}. \tag{3}$$

This formulation highlights a natural similarity-based interpretation: both $\boldsymbol{F}_{\text{test}} = \boldsymbol{X}_{\text{test}}\boldsymbol{U}\Lambda^{-\frac{1}{2}}$ and $\boldsymbol{F}_{\text{train}} = \boldsymbol{X}_{\text{train}}\boldsymbol{U}\Lambda^{-\frac{1}{2}}$ are standardized factor scores providing an orthonormal representation of $\boldsymbol{X}_{\text{test}}$ and $\boldsymbol{X}_{\text{train}}$, respectively. The representation preserves the original input dimension, so $\boldsymbol{F}_{\text{train}} \in \mathbb{R}^{N\times P}$ and $\boldsymbol{F}_{\text{test}} \in \mathbb{R}^{J\times P}$. For $\boldsymbol{F}_{\text{train}}$, we have that

$$\boldsymbol{F}'_{\text{train}}\boldsymbol{F}_{\text{train}} = I_P. \tag{4}$$

However, since Λ and $\boldsymbol{U}$ are estimated from the training data only, the identity $\boldsymbol{F}'_{\text{test}}\boldsymbol{F}_{\text{test}} = I_P$ does not hold exactly—just as $\hat{\boldsymbol{\beta}}$ minimizes squared residuals by construction on the training data, but not necessarily on the test sample.

Thus, OLS out-of-sample predictions can be written as:

$$\underbrace{\hat{\boldsymbol{y}}_{\text{test}}}_{1\times J} = \underbrace{\boldsymbol{F}_{\text{test}}\boldsymbol{F}'_{\text{train}}}_{J\times N} \underbrace{\boldsymbol{y}_{\text{train}}}_{N\times 1}, \tag{5}$$

where $\boldsymbol{F}_{\text{test}}\boldsymbol{F}'_{\text{train}}$ contains inner products between test observations (represented in $\mathbb{R}^P$ by $\boldsymbol{F}_{\text{test}}$) and training observations (represented by $\boldsymbol{F}_{\text{train}}$). This reveals OLS as a similarity-based method, where the prediction for a test point j is:

$$\hat{y}_j = \sum_{i=1}^{N} \underbrace{\langle F_j, F_i \rangle}_{\omega_{ji}} y_i, \tag{6}$$

with $\langle \cdot, \cdot \rangle$ the Euclidean inner product. Each weight ω_{ji} measures similarity between factors and decomposes as:

$$\omega_{ji} = \|F_j\|\|F_i\| \cos(\theta_{ji}), \tag{7}$$

where $\cos(\theta_{ji}) = \frac{\langle F_j, F_i \rangle}{\|F_j\|\|F_i\|}$, θ_{ji} is the angle between F_j and F_i, and $\|\cdot\|$ denotes the vector norm. Large weights arise from either strong alignment ($\cos(\theta_{ji})$) or large norms. Since $\|F_j\|$ is common across i, it does not affect relative weights for predicting $\hat{y}_j$.

Implied Factors. We refer to $\boldsymbol{F}_{\text{train}}$ as the *implied factors* of OLS: orthonormal latent predictors that reconstruct OLS predictions when used as regressors. The factorization is purely alge-

braic; it holds for any dataset regardless of its statistical properties, requires no distributional assumptions, and does not imply dimensionality reduction. These factors are intrinsic to the OLS solution and exist independently of any underlying "true" factor structure.

It is worth noting that the equivalence $\hat{\boldsymbol{y}}_{\text{test}} = \boldsymbol{F}_{\text{test}}\boldsymbol{F}'_{\text{train}}\boldsymbol{y}_{\text{train}} = \boldsymbol{F}_{\text{test}}\hat{\theta}$, where $\hat{\theta} = \boldsymbol{F}'_{\text{train}}\boldsymbol{y}_{\text{train}}$, is a trivial consequence of linear algebra in the OLS case: since $\boldsymbol{F}'_{\text{train}}\boldsymbol{F}_{\text{train}} = I_P$, regressing $\boldsymbol{y}_{\text{train}}$ on $\boldsymbol{F}_{\text{train}}$ yields $\hat{\theta} = (\boldsymbol{F}'_{\text{train}}\boldsymbol{F}_{\text{train}})^{-1}\boldsymbol{F}'_{\text{train}}\boldsymbol{y}_{\text{train}} = \boldsymbol{F}'_{\text{train}}\boldsymbol{y}_{\text{train}}$, and the prediction is identical to the original. There is no dimension reduction here—$\boldsymbol{F}_{\text{train}}$ has P columns, the same as $\boldsymbol{X}_{\text{train}}$—and the idempotency of the hat matrix ensures perfect replication. In the OLS setting, where coefficients $\hat{\boldsymbol{\beta}}$ are directly available and interpretable, there is little reason to prefer this alternative representation. However, the picture will change fairly dramatically for machine learning models with no obvious $\hat{\boldsymbol{\beta}}$ to look at.

In the OLS case, the implied factors $\boldsymbol{F}_{\text{train}} = \boldsymbol{X}_{\text{train}}\boldsymbol{U}\Lambda^{-1/2}$ are exactly the principal components of $\boldsymbol{X}_{\text{train}}$, rescaled by the inverse square root of their eigenvalues. Since $\boldsymbol{U}$ contains the eigenvectors of $\boldsymbol{X}'_{\text{train}}\boldsymbol{X}_{\text{train}}$ and P factors are retained, keeping all OLS implied factors is equivalent to running principal component regression with all P components—which itself reproduces the OLS fit. This equivalence breaks down for machine learning models, whose cross-sample hat matrices are shaped by nonlinear feature transformations, regularization, and data-adaptive basis expansions.

Representations of the form (5) have long appeared in textbooks, usually in the context of fitted values (i.e., when $\boldsymbol{X}_{\text{test}} = \boldsymbol{X}_{\text{train}}$), as they follow directly from the singular value decomposition (SVD) of the hat matrix (Hastie et al., 2009). The key step in what follows is to extend this decomposition to general machine learning models whose hat matrices arise through means other than least squares.

2.2 General Case

Conveniently, this proximity-based interpretation in a transformed space extends beyond linear models, as many machine learning algorithms produce predictions that remain linear in the training outcomes. That is, they either are, or can be reformulated as, linear mappings from realized in-sample responses to both in-sample and out-of-sample predictions. In decreasing order of triviality, this property holds for ridge regression, kernel ridge regression, random forests, boosting, and neural networks with a linear output layer (Goulet Coulombe et al., 2024). More precisely, their predictions can be expressed as

$$\underbrace{\hat{\boldsymbol{y}}_{\text{train}}}_{1\times N} = \underbrace{\boldsymbol{H}_{\text{train}}}_{N\times N}\ \underbrace{\boldsymbol{y}_{\text{train}}}_{N\times 1} \qquad \text{and} \qquad \underbrace{\hat{\boldsymbol{y}}_{\text{test}}}_{1\times J} = \underbrace{\boldsymbol{H}_{\text{test}}}_{J\times N}\ \underbrace{\boldsymbol{y}_{\text{train}}}_{N\times 1} \tag{8}$$

where $\boldsymbol{H}$ is the so-called "hat" matrix Hastie et al. (2009). While $\boldsymbol{H}_{\text{train}}$ has an explicit formulation in terms of input matrices for least squares or kernelized versions, it must be retrieved through other means for tree-based ensembles and neural networks (Goulet Coulombe et al., 2024).

The key insight is that, *regardless of its origin*, $\boldsymbol{H}_{\text{train}}$ can be factorized such that

$$\boldsymbol{H}_{\text{train}} = \boldsymbol{F}_{\text{train}}\boldsymbol{F}'_{\text{train}} \quad \text{and} \quad \boldsymbol{H}_{\text{test}} = \boldsymbol{F}_{\text{test}}\boldsymbol{F}'_{\text{train}}. \tag{9}$$

We refer to the resulting factors as the implied factors of the algorithm that produced $\boldsymbol{H}_{\text{train}}$ and $\boldsymbol{H}_{\text{test}}$ from the dataset $[\boldsymbol{y}_{\text{train}} \ \boldsymbol{X}_{\text{train}}]$. This has several implications. First, any machine learning model that fits the bill of (8) can be viewed as a similarity-based estimator based on inner products in the space of implied factors. In this view, algorithms differ not in their proximity metric but in the features used to compute it. The second implication follows directly from the first and our discussion for OLS in Section 2.1. This similarity-based estimator using implied factors is also equivalently a correlation-based one using those same factors as features in OLS. Putting all this together, we have the following equivalence for the prediction of an arbitrary observation j:

$$\underbrace{\hat{y}_j}_{\text{Prediction}} = \underbrace{\hat{g}(X_j)}_{\text{Black Box}} = \underbrace{F_j\boldsymbol{F}'_{\text{train}}\boldsymbol{y}_{\text{train}}}_{\text{Similarity-based Estimator}} = \underbrace{F_j\hat{\theta}}_{\text{Linear Regression}}. \tag{10}$$

where $\hat{\theta}$ is the vector coefficients obtained from regressing $\boldsymbol{y}_{\text{train}}$ on $\boldsymbol{F}_{\text{train}}$. As a result, forecasts generated by $\hat{g}(\cdot)$ can be interpreted through the lens of factor-augmented models in the tradition of Stock and Watson (2002), allowing us to visualize implied factors as time series and, when possible, give them names.

Recovering $\boldsymbol{H}_{\text{train}}$ and $\boldsymbol{H}_{\text{test}}$. Obtaining $\boldsymbol{H}_{\text{train}}$ and $\boldsymbol{H}_{\text{test}}$ is mostly straightforward and algorithm-specific. For many regularized estimators and least-squares-based methods, the hat matrix follows directly from the formula of the estimator itself. For kernel-based regression, it is likewise directly available through duality (the representer theorem). For tree-based ensembles, the corresponding accounting operation surveys which observations land in which leaves across trees. Finally, for neural networks with a linear output layer, the activations of the last hidden layer can be used as generated regressors in a ridge regression that perfectly replicates the network's predictions; this can also be motivated from the first-order conditions of the neural network itself, and yields the observation weights straightforwardly. The explicit formulas for each model class are provided in Appendix A.1; see also Goulet Coulombe et al. (2024) for a comprehensive treatment.

Recovering In- and Out-of-Sample Implied Factors. The remaining question is how to recover both $\boldsymbol{F}_{\text{train}}$ and $\boldsymbol{F}_{\text{test}}$. The answer lies in eigendecomposition: we extract $\boldsymbol{F}_{\text{train}}$ as the leading K eigenvectors of $\boldsymbol{H}_{\text{train}}$, and compute $\boldsymbol{F}_{\text{test}}$ by projecting $\boldsymbol{H}_{\text{test}}$ onto the $\boldsymbol{F}_{\text{train}}$ space:

$$\boldsymbol{F}_{\text{test}} = \boldsymbol{H}_{\text{test}}\boldsymbol{F}_{\text{train}}(\boldsymbol{F}'_{\text{train}}\boldsymbol{F}_{\text{train}})^{-1}.$$

This projection aligns out-of-sample observations with the latent geometry used in-sample, enabling comparisons, interpretation, and contribution analysis in a common factor space. Once the implied factors are obtained, we can proceed as usual: visualize them to gain insight, or feed them into post-estimation regressions as pre-trained features. By construction, $\boldsymbol{F}'_{\text{train}}\boldsymbol{F}_{\text{train}} = I_P$, and therefore we simply have $\boldsymbol{F}_{\text{test}} = \boldsymbol{H}_{\text{test}}\boldsymbol{F}_{\text{train}}$. Note that, unlike in traditional factor analysis (where $\boldsymbol{F}_{\text{train}}$ are linear combinations of $\boldsymbol{X}_{\text{train}}$), we cannot use learned loadings to generate $\boldsymbol{F}_{\text{test}}$. Here, the factors are implicit and unspecified functions of the regressors, not tied to a known linear mapping. Of course, this does not eliminate the need to run the full model in the first place, but once the factors are extracted they allow us to interpret predictions through a linear model in a potentially much lower-dimensional latent space.

While there are many ways to decompose square matrices, the spectral decomposition is particularly useful because it expresses the system in terms of uncorrelated characteristics. This diagonalization clarifies interpretation: total similarity between observations reduces to the sum of orthogonal contributions. The search for orthogonal structures has a long-standing tradition in macroeconomics, and implied factors fit naturally within that lineage. Moreover, just as PCA-based factors are chosen to explain the most variance in $\boldsymbol{X}'_{\text{train}}\boldsymbol{X}_{\text{train}}$, implied factors are designed to capture the most variation in $\boldsymbol{H}_{\text{train}}$, the proximity weights induced by the machine learning algorithm. As a result, only a few are typically needed to faithfully reconstruct OLS-style predictions, thereby enhancing interpretability while preserving fidelity to the original model.

Handling Complex Numbers. While $\boldsymbol{H}_{\text{train}}$ is idempotent and symmetric in ordinary least squares (OLS), and remains symmetric under certain extensions such as regularized least squares, it is not necessarily symmetric for methods like random forests or boosting, where $\boldsymbol{H}_{\text{train}}$ is derived through tree accounting algorithms rather than matrix operations. As a result, its eigendecomposition may yield complex-valued components, typically among the less important directions. In practice, these occurrences are rare and carry little explanatory weight. Throughout our analysis, we retain only the real part of the eigenvectors and use the modulus of the eigenvalues when necessary, ensuring real-valued representations for regression and visualization.

Regularization, Combination, and Realignment. Lastly, the regression on implied factors does in our case use the same observations from which the model was originally trained, but it need not. Since the implied-factor regression involves as many factors as observations, standard tools can be used to select a parsimonious subset. Information criteria (BIC, AIC) or the criteria of Bai and Ng (2002) provide automatic selection rules, while Lasso with cross-validation offers an alternative. One may retain enough factors to reproduce the original predictions exactly, or keep only a subset for additional regularization. A further possibility is to pool implied factors from *different* models and select among them, yielding mixed-factor specifications that combine the predictive representations of multiple machine learning algorithms. On the realignment side, factors may be extracted using the full sample, but the post-regression can be conducted using only recent data to account for possible structural changes, preserving the complex feature representations while ensuring alignment with the most relevant historical period.

2.3 Contribution of Implied Factors to Prediction

Before attempting to interpret implied factors or give them names, it is useful to know how many of them matter and how the predictive content is distributed across them. We therefore quantify how the predictive content of the model builds up as additional implied factors are included, both in sample and out of sample, using a target-weighted spectral decomposition of the relevant hat matrices.

In-sample decomposition. Let $\boldsymbol{y}_{\text{train}} \in \mathbb{R}^N$ denote the training-sample target and $\boldsymbol{H}_{\text{train}} \in \mathbb{R}^{N\times N}$ the in-sample hat matrix. Write its eigendecomposition as $\boldsymbol{H}_{\text{train}} = V\Lambda V'$, where $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots)$ contains eigenvalues in decreasing magnitude and $V = (v_1, v_2, \dots)$ is the associated orthonormal eigenvector matrix. We define the in-sample contribution of component i and the cumulative fraction of in-sample predictive content explained by the first $k \leq K$ factors as

$$\delta_i^{\text{IS}} = \lambda_i^2 \, (v_i' \boldsymbol{y}_{\text{train}})^2, \qquad\qquad C_k^{\text{IS}} = \frac{\sum_{i=1}^{k} \delta_i^{\text{IS}}}{\sum_{i=1}^{K} \delta_i^{\text{IS}}}, \tag{11}$$

respectively. The first captures both the strength of the component in the hat matrix and its alignment with the target. Meanwhile, letting K denote the number of retained implied factors, the latter captures the cumulative fraction of in-sample predictive content explained by the first $k \leq K$ factors, which is non-decreasing in k with $C_K^{\text{IS}} = 1$.

Out-of-sample decomposition. Let $\boldsymbol{F}_{\text{train}} \in \mathbb{R}^{N\times K}$ and $\boldsymbol{F}_{\text{test}} \in \mathbb{R}^{J\times K}$ denote the implied-factor matrices in the training and test samples (the latter constructed via a Nyström-type extension).

The out-of-sample hat matrix is $\boldsymbol{H}_{\text{test}} = \boldsymbol{F}_{\text{test}}\boldsymbol{F}'_{\text{train}}$. Since this matrix is rectangular, we use its SVD, $\boldsymbol{H}_{\text{test}} = U\Sigma V'$, with singular values $\sigma_1 \geq \cdots \geq \sigma_r > 0$ and $r = \text{rank}(\boldsymbol{H}_{\text{test}}) \leq \min(N, J)$. The realized prediction decomposes as

$$\hat{\boldsymbol{y}}_{\text{test}} = \boldsymbol{H}_{\text{test}}\,\boldsymbol{y}_{\text{train}} = \sum_{i=1}^{r} \sigma_i\, u_i\, (v_i'\boldsymbol{y}_{\text{train}}),$$

and so the out-of-sample contribution of component i and cumulative fraction for $k = 1, \ldots, r$, are given by

$$\delta_i^{\text{OOS}} = \sigma_i^2 (v_i'\boldsymbol{y}_{\text{train}})^2, \qquad C_k^{\text{OOS}} = \frac{\sum_{i=1}^{k} \delta_i^{\text{OOS}}}{\sum_{i=1}^{r} \delta_i^{\text{OOS}}}. \tag{12}$$

Because the rank of $\boldsymbol{H}_{\text{test}}$ is bounded by the test-sample length, the decomposition is limited to at most r independent predictive components. Together, the in-sample and out-of-sample decompositions characterize the effective dimensionality of the model's predictive structure and reveal how many independent directions are supported out of sample.

3 An Application to Macroeconomic Forecasting

We apply the implied-factor framework to quarterly U.S. macroeconomic forecasting. We begin by describing the data, evaluation design, and models (Section 3.1). We then assess the out-of-sample performance of the full machine learning models (Section 3.2), ask how many implied factors are needed to reproduce their forecasts (Section 3.3), and examine the common signals that emerge when implied factors are clustered across models (Section 3.4). Finally, we compare implied factors to traditional supervised factor methods (Section 3.5).

3.1 Data, Design, and Models

We study the empirical relevance of implied factors in a macroeconomic forecasting environment. The analysis focuses on three quarterly U.S. macroeconomic aggregates: real GDP growth, headline CPI inflation, and the unemployment rate. Forecasts are produced at horizons $h \in \{1, 2, 4\}$ quarters ahead using direct forecasting. For evaluating out-of-sample forecast accuracy—comparing full ML models, their implied-factor counterparts, and traditional factor methods against an AR(4) benchmark—we adopt an expanding-window scheme in which the training sample begins in 1961Q2 and expands by one year at each step, with the first out-of-sample window covering 2007 and the last covering 2024; the training sample is frozen at 2019Q4 so that no COVID-period data enters the estimation sample. For the implied-factor analyses that

require a fixed train–test split—predictive contributions (Section 3.3) and cross-model clustering (Section 3.4)—we use training on 1962Q2–2006Q4 and testing on 2007Q1–2024Q4.

In both settings, COVID quarters (2020Q1–2020Q4) are excluded when computing forecast accuracy metrics to prevent a small number of extreme observations from dominating model comparisons, though they are retained when displaying factor time series.

Models. We consider several classes of forecasting models. Among machine learning models, we include both linear and nonlinear specifications: ridge regression serves as a regularized linear benchmark, while kernel ridge regression, random forests, gradient-boosted trees, and a feedforward neural network capture progressively richer forms of nonlinearity and interaction structure. For each model, we also consider a restricted implied-factor specification that retains only the leading K implied factors from the model's cross-sample hat matrix and forecasts via OLS on the extracted factors (Section 2); we set $K = 3$ in our main results. To aggregate information across models, mixed-factor specifications pool all implied factors and forecast via OLS or LASSO.

As discussed in the introduction, implied factors inherit supervision through the cross-sample hat matrix rather than through marginal predictor–target relationships. To benchmark this form of supervision, we compare against the leading factor-based methods from the supervised dimension reduction literature—PCR, PLS, sPCA, SPCA, and SsPCA—which also extract a small number of predictive factors from high-dimensional panels, but do so by reweighting or screening predictors based on their marginal association with the target.

AR(4): Fourth-order autoregression on the target variable.

RR: Ridge Regression. High-dimensional linear prediction with ℓ_2 penalty; the regularization parameter λ is selected by ten-fold cross-validation using the one-standard-error rule.

KRR: Kernel Ridge Regression. Extends ridge regression to nonlinear settings via a Laplacian kernel; the kernel bandwidth is set to $\sigma = 0.01$ and regularization to $\lambda = 1/n$.

RF: Random Forest (Breiman, 2001). Aggregates 200 regression trees grown on bootstrap samples with random feature subsets ($\lfloor p/3 \rfloor$ candidate variables per split); subsampling fraction 0.75, minimum node size 25.

LGB: LightGBM (Ke et al., 2017). Sequential gradient boosting with histogram-based splits (LightGBM); 16 leaves per tree, learning rate 0.01, minimum 10 observations per leaf, with early stopping to select the number of boosting rounds.

NN: Feedforward neural network with three hidden layers (50 neurons each), ReLU activations, and dropout regularization (rate 0.5); trained via Adam with early stopping and averaged over 30 bootstrap draws to reduce estimation noise.

IF–m: Implied-factor specification for model $m \in \{\text{KRR, RF, LGB, NN, RR}\}$. Retains only K implied factors and produces forecasts via OLS on the extracted factors (see Section 2). These restricted specifications isolate the low-dimensional predictive structure from the full model's predictions.

MF–LM: Mixed-factor OLS. Regresses the target on all $5K$ implied factors pooled from the five models (KRR, RF, LGB, NN, RR).

MF–LSO: Mixed-factor LASSO. Same pooled factor set as MF–LM, with ℓ_1-penalized selection.

PCR: Principal component regression (Stock and Watson, 2002). Extracts the leading K principal components from the predictor panel and forecasts via OLS on the extracted factors.

PLS: Partial least squares (Kelly and Pruitt, 2013). Sequentially constructs K factors that maximize predictor–target covariance via the NIPALS algorithm.

sPCA: Scaled PCA (Huang et al., 2022). Reweights predictors by the absolute value of their marginal covariance with the target before extracting principal components.

SPCA: Supervised PCA (Hounyo and Li, 2025). Iteratively screens predictors by target correlation, extracts one factor per iteration, and deflates the predictor matrix.

SsPCA: Supervised scaled PCA (Hounyo and Li, 2025). Combines the scaling step of sPCA with the iterative screening and deflation of SPCA.

3.2 Forecasting Performance

Table 1 reports out-of-sample RMSEs relative to the AR(4) benchmark under expanding-window evaluation.

Real GDP. Random forests and ridge regression deliver the strongest short-horizon forecasts, both significantly outperforming the AR(4) at h=1, with the neural network close behind. KRR and LGB, by contrast, fail to improve on the benchmark at the shortest horizon but strengthen markedly at longer horizons: at h=4, LGB leads and four of the five models achieve statistically significant gains. The restricted K=3 implied-factor specifications closely track their full counterparts: IF–RF matches the full random forest at every horizon, and IF–LGB and IF–NN

Table 1: Out-of-sample forecast performance — RMSE relative to AR4

	ML Models					*Implied Factor Models* ($K = 3$)						
	KRR	LGB	NN	RF	RR	IF-KRR	IF-LGB	IF-NN	IF-RF	IF-RR	MF-LM	MF-LSO
Real GDP												
h=1	1.02	1.02	0.87	0.86**	0.86**	1.01	1.02	0.89	**0.85****	0.94	1.12	**0.84***
h=2	0.92*	0.93	0.91	0.87*	0.87**	0.93	0.91	0.91	**0.86***	0.93	1.04	**0.84****
h=4	0.80*	**0.78****	0.82	0.82*	0.79*	0.82*	**0.79***	0.79*	0.83*	0.86	0.95	0.82*
CPI Inflation												
h=1	1.07	1.02	0.99	0.95	1.17	1.10	1.03	1.00	**0.94**	1.15	**0.93**	1.00
h=2	0.94	0.97	0.86*	0.85*	1.07	0.94	0.98	0.89	**0.85***	1.02	**0.81***	0.87*
h=4	**0.88***	1.01	0.93	0.97	1.05	0.89*	1.01	0.98	0.96	1.06	**0.86***	0.93
Unemployment Rate												
h=1	0.97	0.77	**0.70***	0.71	0.72	1.00	0.77	0.74	0.71	0.76	0.74	**0.68**
h=2	0.87	0.84	**0.65***	0.74	0.73	0.89	0.83	**0.68***	0.74	0.76	0.70*	0.70*
h=4	0.85*	0.83**	0.81*	0.81**	0.81*	0.88*	0.80**	0.80*	**0.79****	0.81*	**0.76****	0.82*

Notes: Values show RMSE relative to AR4 benchmark for the full (unrestricted) model predictions (left) and restricted model predictions (right), using expanding-window evaluation (training starts 1961Q2, first test 2007Q1–Q4, expanding by 1 year each step). */**/*** denote one-sided DM test significance at 10/5/1%. Best: **bold green**; second-best: **bold**. The COVID period (2020Q1–2020Q4) is excluded.

achieve significance at h=4. Among the pooled specifications, MF–LSO stands out, delivering the best or near-best performance at every horizon. Overall, a small number of implied factors suffices to preserve the full models' gains over the AR(4), and LASSO-based pooling across models can improve further. The implied-factor analysis below will help us understand what drives these gains and whether different models exploit qualitatively different predictive signals.

CPI inflation. At h=1, random forests and the neural network improve on the AR(4), though the gain is not statistically significant. The picture sharpens at h=2, where RF and NN deliver significant improvements, and at h=4, where KRR emerges as the leader. The restricted specifications mirror these patterns: IF–RF closely tracks the full random forest, and IF–KRR preserves KRR's advantage at h=4. The mixed-factor specifications prove particularly effective for inflation: MF–LM delivers the best performance among all implied-factor models at every horizon, achieving significance at h=2 and h=4, suggesting that pooling complementary signals across models is especially valuable when no single model dominates throughout. The shifting leaderboard across horizons—and the strong showing of pooled specifications—motivates the implied-factor analysis below: by comparing the latent signals extracted from these models, we can investigate which predictive dimensions are relevant at each horizon and how they complement one another.

Unemployment rate. This target is closely related to GDP growth but tends to be less noisy, and forecast improvements are correspondingly the largest of the three variables. The neural network dominates at short horizons, achieving relative RMSEs of 0.70 and 0.65 at h=1

and h=2, respectively; RF and RR follow closely at both horizons. At h=4, all five models significantly outperform the benchmark, with relative RMSEs ranging from 0.81 to 0.85. The restricted specifications largely preserve these gains as IF–RF and IF–NN closely match their full model performance and remain strongly competitive. The pooled specifications perform well throughout: MF–LSO is the best overall model at h=1, and MF–LM achieves the strongest performance at h=4. Across all horizons, the implied-factor models confirm that a few factors capture the bulk of the ML models' advantage over the AR(4). The implied-factor clustering below will allow us to compare the logic behind these forecasts and assess whether the models extract genuinely different signals or merely different quantitative assessments of the same underlying business-cycle dynamics.

3.3 How Many Implied Factors?

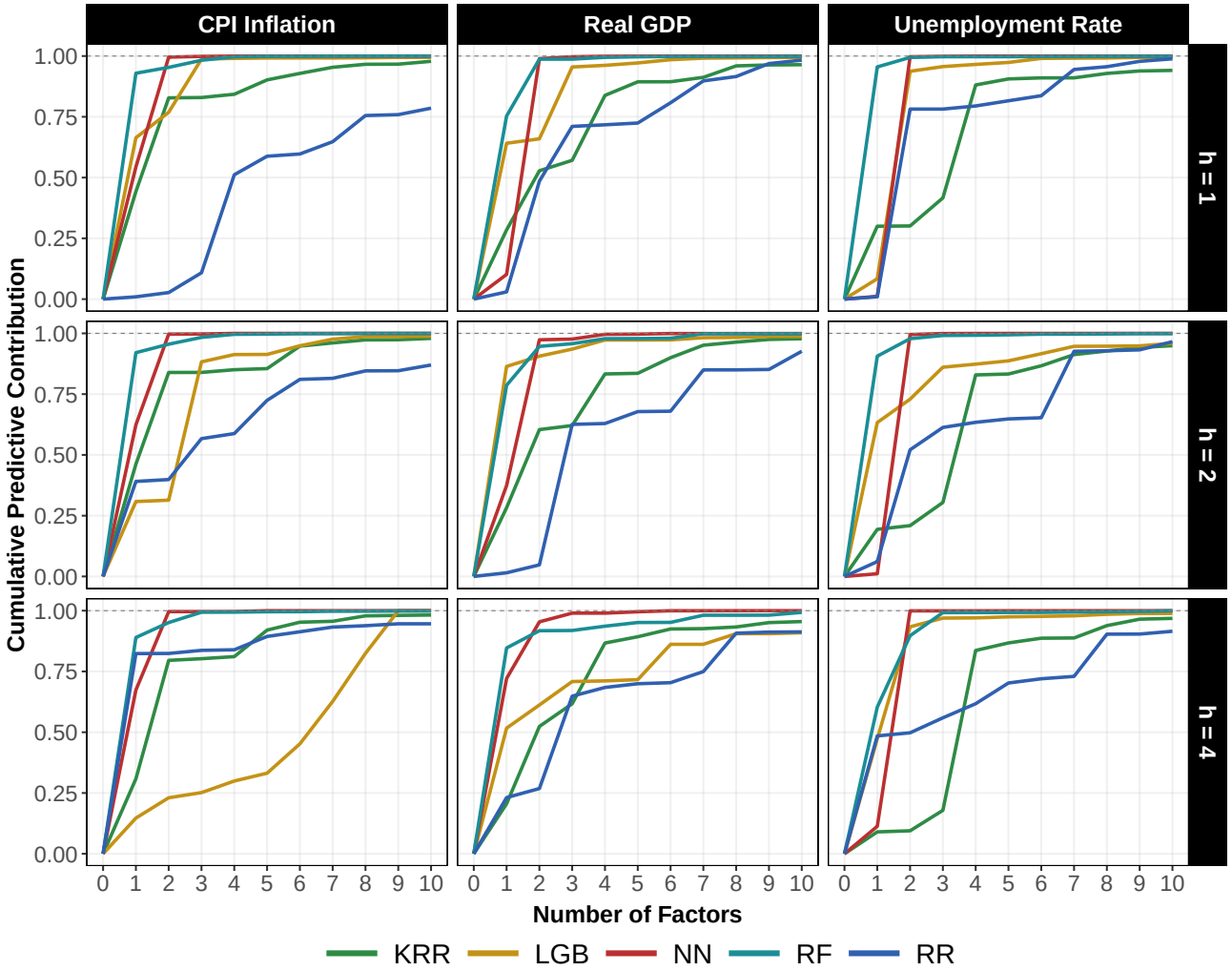
Given that the ML models produce meaningful forecast improvements, we next ask how much of their predictive content can be captured by a small number of implied factors. We first examine how the predictive contribution accumulates as factors are added, and then assess whether restricting to few factors costs forecast accuracy.

Predictive Contribution of Implied Factors. Figure 1 reports the cumulative out-of-sample predictive contribution as implied factors are added sequentially, using the metric proposed in Section 2.3. Because this decomposition requires a fixed hat matrix, we use a fixed train–test split (training on 1962Q2–2006Q4, testing on 2007Q1–2024Q4) rather than the expanding-window scheme employed for forecast evaluation.

For short-run forecasting (h=1), the concentration of predictive content in a handful of factors is striking. For real GDP, two to three factors suffice for NN, RF, and LGB; KRR and RR require more, reflecting the smoother spectral decay of their hat matrices. For CPI inflation, two factors are enough to nearly fully replicate the forecasts of random forests and neural networks. LGB requires roughly three factors to reach the same level. KRR exhibits a different pattern: the cumulative contribution reaches a plateau around two factors, after which the remaining factors contribute only incrementally, with full replicability requiring all ten. Ridge regression climbs even more slowly, reflecting how the ℓ_2 penalty smooths the eigenvalues of the hat matrix and distributes predictive content more diffusely across factors. For the unemployment rate, the concentration is most pronounced: for several models, a single factor captures the bulk of the predictive content, consistent with unemployment being driven by a dominant business-cycle signal.

This pronounced lower-dimensional structure echoes a central finding of the macro factor literature, where a small number of latent factors—typically three to five—capture most aggre-

Figure 1: Out-of-Sample Predictive Contribution of Implied Factors (2007Q1–2024Q4)



Notes: This figure reports the *cumulative out-of-sample predictive contribution* of implied factors within each forecasting model, as increasingly many factors are included (excluding the COVID period, 2020Q1–2020Q4). Columns correspond to target variables, rows correspond to forecast horizons ($h = 1, 2, 4$), and panels correspond to forecasting models. A steeper accumulation indicates that predictive content is concentrated in a small number of implied factors. Results are based on a fixed training sample (1961Q2–2006Q4) and test sample (2007Q1–2024Q4). The corresponding in-sample decomposition is reported in Appendix B, Figure B.1.

gate co-movement and are sufficient for accurate forecasting (Stock and Watson, 1999, 2002; Bai and Ng, 2002). The objects considered here, however, differ conceptually. Traditional factors are extracted from the covariance structure of the predictors and summarize common variation, whereas the implied factors arise from the singular value decomposition of the model's hat matrix and therefore summarize predictive influence. The fact that both approaches reveal a similarly low effective dimension—even in highly flexible nonlinear models—indicates that a relatively small number of factors contains most of the forecast-relevant variation.

At longer horizons (h=2, 4), the patterns for NN and RF remain broadly similar: predictive content stays concentrated in two or three leading factors. Thus, the effective predictive rank for these models does not materially increase with the forecast horizon. While the implied factors

are horizon-specific under direct forecasting—since each horizon corresponds to a distinct predictive mapping—the dimensionality of that mapping remains consistently low. This pattern is reminiscent of findings in the dynamic factor literature, where macroeconomic fluctuations are driven by a small number of latent factors that suffice to generate forecasts at multiple horizons (Stock and Watson, 2002).

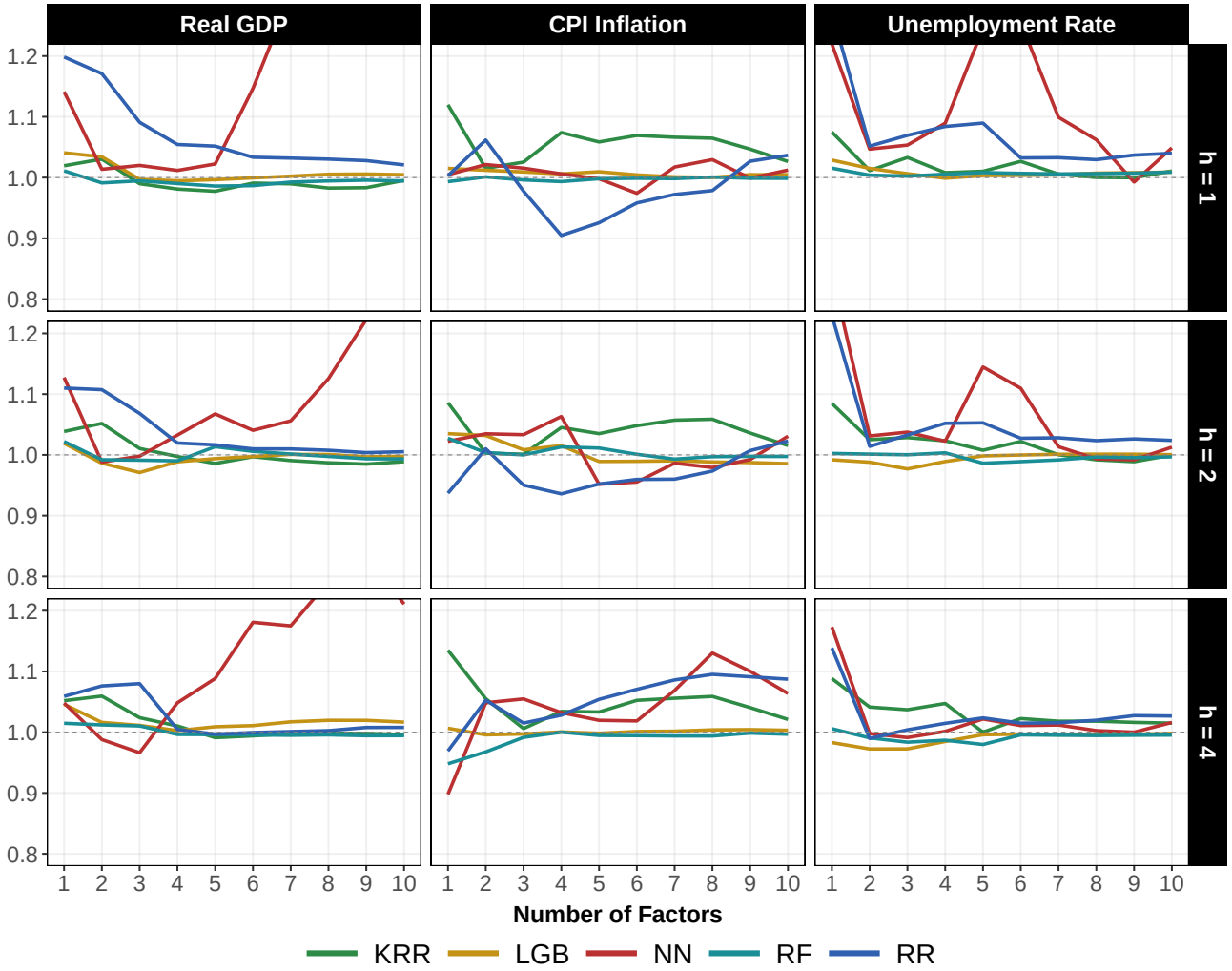
The model that changes most visibly is LGB, particularly for real GDP and CPI inflation, where the cumulative contribution curve becomes markedly smoother. This suggests that the boosting algorithm distributes its predictive content more evenly across factors at longer horizons, possibly reflecting the weaker signal-to-noise ratio. The in-sample decomposition (Appendix B, Figure B.1) exhibits qualitatively similar patterns, suggesting that the spectral properties of the hat matrix are stable across the training and evaluation samples.

Relative RMSE of Restricted Predictions. Having established that a small number of factors captures the bulk of predictive content, we turn to whether restricting the model to K factors actually costs forecast accuracy, or whether dropping the remaining factors might even help through an implicit bias-variance tradeoff. Figure 2 reports the ratio of the out-of-sample RMSE from the K-factor restricted prediction to that of the unrestricted model, for $K = 1, \ldots, 10$. Values below one indicate that the restricted prediction outperforms the full model.

The results confirm that restricted predictions closely reproduce the full model forecasts across targets and horizons. RF stands out for its remarkable stability: the relative RMSE hovers near one at virtually all values of $K > 1$, indicating that using very few implied factors captures the essential predictive content of the random forest. LGB exhibits a similarly muted sensitivity, with ratios remaining close to one from $K = 2$ onward. More generally, tree-based models (RF, LGB) display smoother RMSE paths than models estimated through least squares, consistent with the implicit averaging in ensemble methods producing a more stable predictive structure. Ridge regression shows the steepest initial sensitivity—particularly for real GDP and unemployment at h=1, where the ratio at $K = 1$ reaches 1.20 and 1.27, respectively—but converges rapidly, falling below 1.10 by $K = 3$ or $K = 4$. This pattern aligns with the concentration of RR's predictive content in its first few eigendirections documented in Figure 1. The neural network follows a broadly similar trajectory, and still at $K = 2$ or $K = 3$, ratios sit close to one across all three targets, confirming that a small number of factors suffices to approximate the full network prediction.

The central message is that, by $K = 2$ or $K = 3$, most ratios cluster tightly around one: restricted predictions are essentially interchangeable with the full model forecasts. Factor truncation does not offer a systematic route to improved accuracy, and removing the leading factors consistently hurts. One noteworthy exception is RR for CPI inflation at h=1, where restricting to $K = 4$ yields a ratio of 0.90, implying a meaningful improvement from dropping the trailing eigendirections—a case where the bias–variance tradeoff tilts in favour of the sparser

Figure 2: Relative RMSE of Implied-Factor Predictions (2007–2024, excluding COVID)



Notes: This figure reports the ratio of out-of-sample RMSE from the restricted K-factor implied-factor prediction to the RMSE of the unrestricted (full) model, for $K = 1, \ldots, 10$, using expanding-window evaluation. The COVID period (2020Q1–2020Q4) is excluded. A value below one indicates that the restricted prediction outperforms the full model. Columns correspond to target variables, rows to forecast horizons ($h = 1, 2, 4$), and colors to forecasting models.

specification. While a small number of latent factors often underpins competitive forecasts (Stock and Watson, 2002), the broader forecasting literature shows that accuracy does not always improve monotonically with the number of factors and depends on the horizon and target variable (Faust and Wright, 2013). Moreover, criteria that select only a few factors may omit weaker, yet forecast-relevant components (Bańbura et al., 2010). This is consistent with our finding that factor truncation rarely leads to systematic performance gains.

3.4 Common Signals Across Models

We next ask whether implied factors can be related to known economic dynamics—and if so, whether these interpretable signals are common across models. Using the hierarchical clustering algorithm described in Appendix A.2, we group implied factors across models based on their

pairwise similarity. Because implied factors inherit the sign indeterminacy of eigenvectors, we cluster on absolute similarity and subsequently align signs within each cluster to enable direct visual comparison. Throughout, we report the cumulative predictive contribution measures C_k^{IS} and C_k^{OOS} introduced in Section 2.3 to quantify how much of each model's forecast content is captured by the factors in a given cluster.

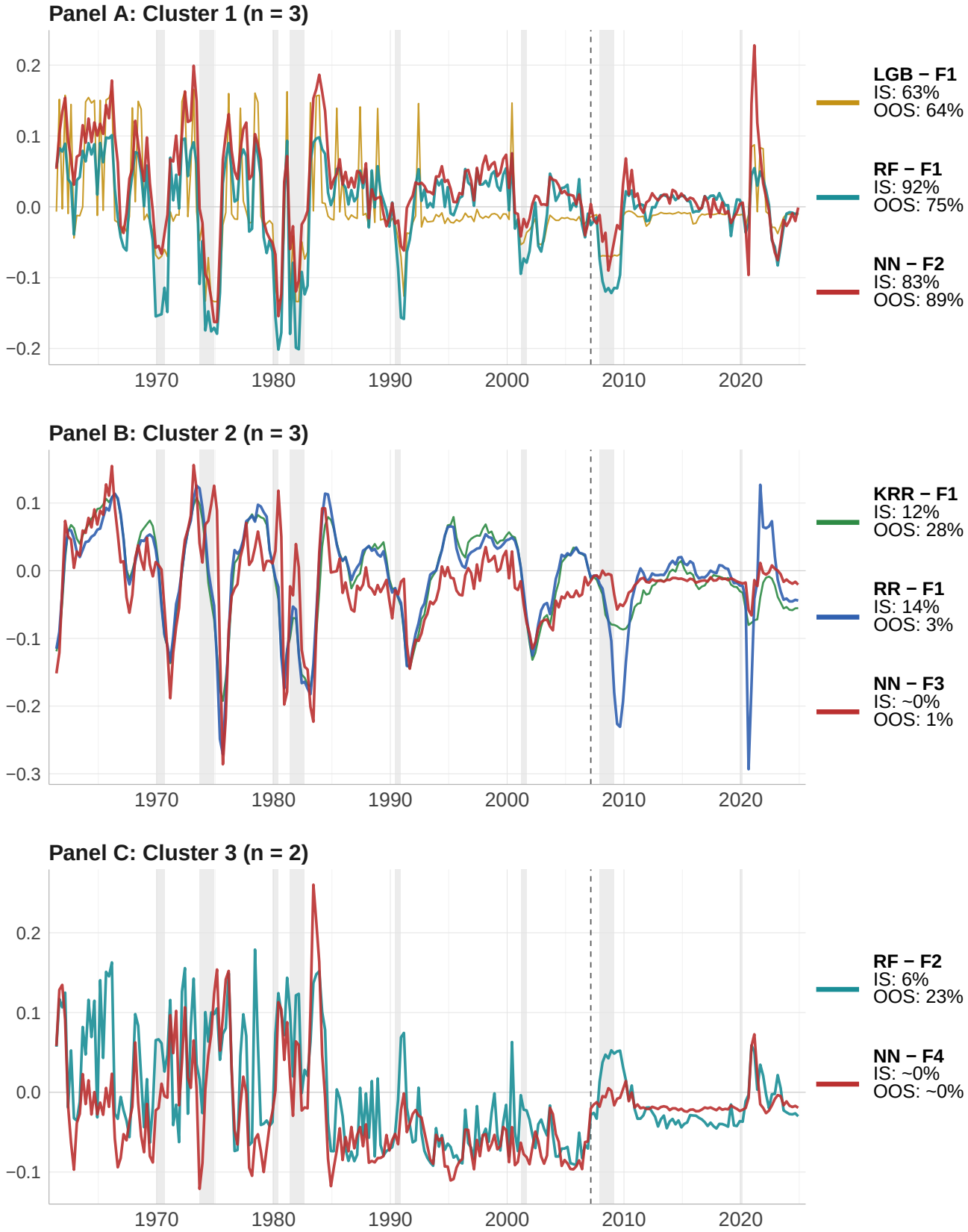
A key feature of this analysis is that the i-th implied factor of one model need not correspond to the i-th factor of another: the eigendecomposition of each model's cross-sample hat matrix produces its own ordering, so that factors with different indices may capture the same signal. The clustering procedure identifies these correspondences automatically. As we document below, across all three target variables, implied factors from different models—and different factor indices—cluster into a small number of common signals, implying that cross-model differences are largely quantitative rather than conceptual.

Real GDP. Three clusters emerge (Figure 3). The first (Panel A) groups RF–F1, LGB–F1, and NN–F2, all carrying high predictive contribution (C_k^{IS}: 63–92%; C_k^{OOS}: 61–91%). This dominant signal captures state-dependent shifts in expected growth. The random forest factor is particularly revealing: it resembles a Markov-switching intercept, with positive values during expansions and sharp drops during recessions, consistent with the tree structure's natural ability to classify regimes. LGB–F1 displays a similar boxy shape, though with occasional spikes. The neural network's second factor (NN–F2, C_k^{IS}: 83%, C_k^{OOS}: 91%) tracks the same dynamics but more smoothly, evoking an output gap measure. Interestingly, around COVID this factor does not plunge as sharply as the tree-based signals; instead, it spikes upward during the recovery, suggesting the neural network reads the post-COVID rebound as a particularly strong positive deviation from trend.

The second cluster (Panel B) groups the leading factors of the two least-squares-based models (RR–F1 and KRR–F1), together with NN–F3. KRR and RR produce strikingly similar in-sample trajectories, both resembling a smoothed principal component of economic activity. This is unsurprising: ridge regression is OLS with an ℓ_2 penalty, and KRR extends it through a kernel, so both inherit a smooth, linear-in-outcome structure. The key difference appears out of sample: the dip during the Great Recession and the COVID episode is markedly less pronounced for KRR than for RR. Moreover, ridge regression's assessment of post-COVID growth is relatively strong, aligning with the neural network's reading.

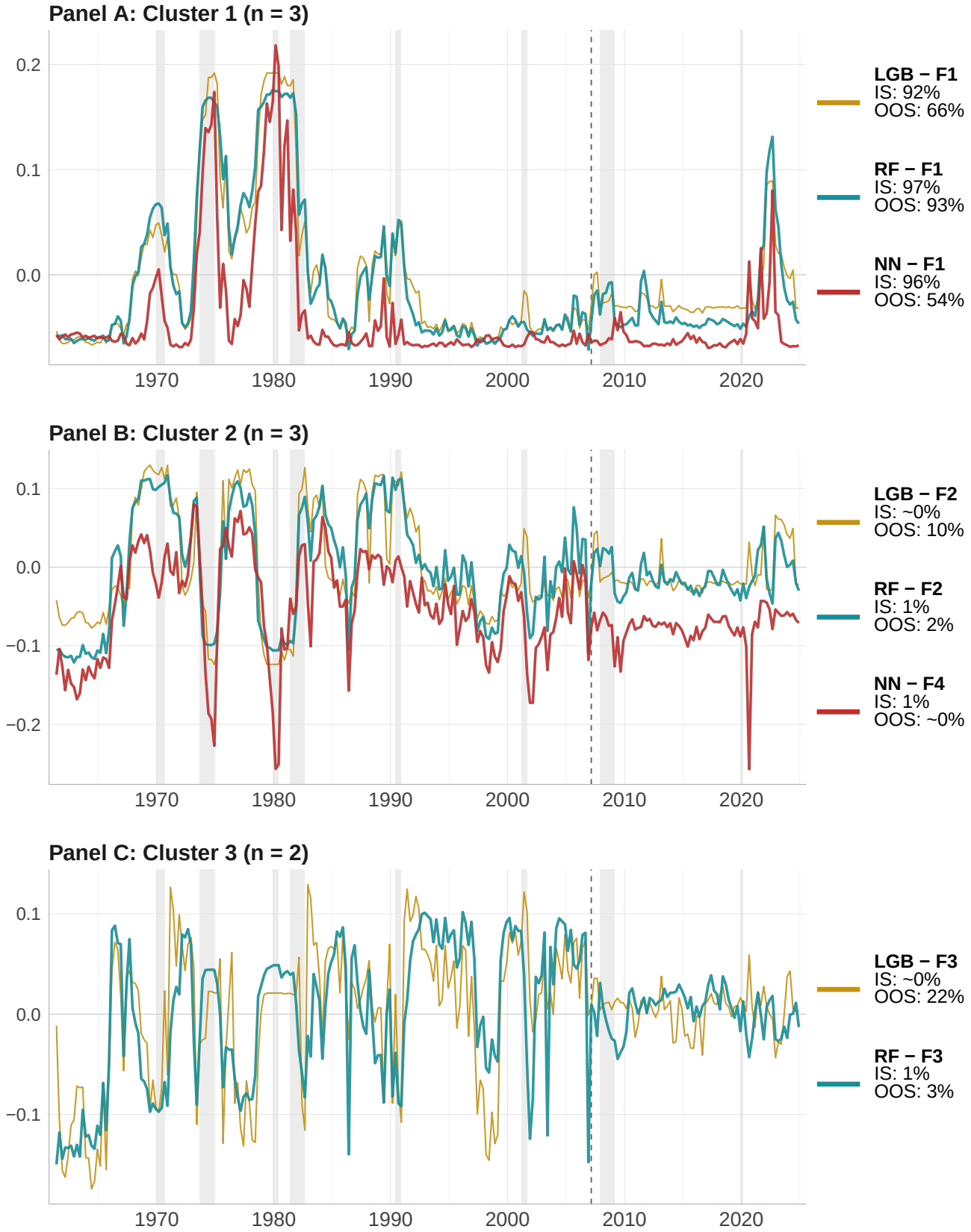
The third cluster (Panel C) pairs RF–F2 (C_k^{IS}: 6%, C_k^{OOS}: 24%) with NN–F4. This signal exhibits high amplitude prior to the mid-1980s followed by sharply reduced variability, a pattern consistent with the Great Moderation. It appears to capture long-run changes in GDP growth volatility, functioning as an uncertainty indicator rather than a level or direction signal.

Figure 3: Implied factors for real GDP — clusters ($h = 1$, 2007Q1–2024Q4)



Notes: Implied-factor signals are grouped using hierarchical clustering based on the similarity metric introduced in Appendix A.2. Absolute similarity is used to ensure robustness to sign indeterminacy Legend entries report the model, factor index, and in-sample (C_k^{IS}) and out-of-sample (C_k^{OOS}) predictive contribution.

Figure 4: Implied factors for CPI inflation — clusters ($h = 1$, 2007Q1–2024Q4)



Notes: Implied-factor signals are grouped using hierarchical clustering based on the similarity metric introduced in Appendix A.2. Absolute similarity is used to ensure robustness to sign indeterminacy. Legend entries report the model, factor index, and in-sample (C_k^{IS}) and out-of-sample (C_k^{OOS}) predictive contribution.

CPI Inflation. Three clusters emerge (Figure 4), all dominated by nonlinear models that perform well pre-COVID. The first (Panel A) groups the leading factors of all three nonlinear models (RF–F1, LGB–F1, NN–F1), which together dominate in-sample predictive content (C_k^{IS}: 92–97%). The signal captures low-frequency, regime-level movements in inflation. The 1970s–early 1980s inflation episode is the most prominent feature, and the factors are well anchored near zero through the 1990s and 2000s, a period of stable inflation expectations. Post-COVID, all three signals pick up again, modestly relative to the 1970s but unmistakably, reflecting the 2021–2023 inflation surge. The neural network's reading is particularly clear on this point. This cluster can be interpreted as a proxy for deviations from long-run inflation expectations: elevated when expectations are de-anchored, quiescent when they are firmly held.

The second cluster (Panel B) groups RF–F2, LGB–F2, and NN–F4, all with modest in-sample contributions but some out-of-sample relevance for LGB. This signal tracks real activity: it follows business cycles, with pronounced dips during recessions, especially visible in the 1970s and around the Great Recession. The neural network shows a substantial COVID dip followed by an upside during the recovery. However, only for LGB does this factor capture a non-negligible C_k^{OOS} share (4%), suggesting that the real-activity channel is secondary for inflation forecasting in this setting.

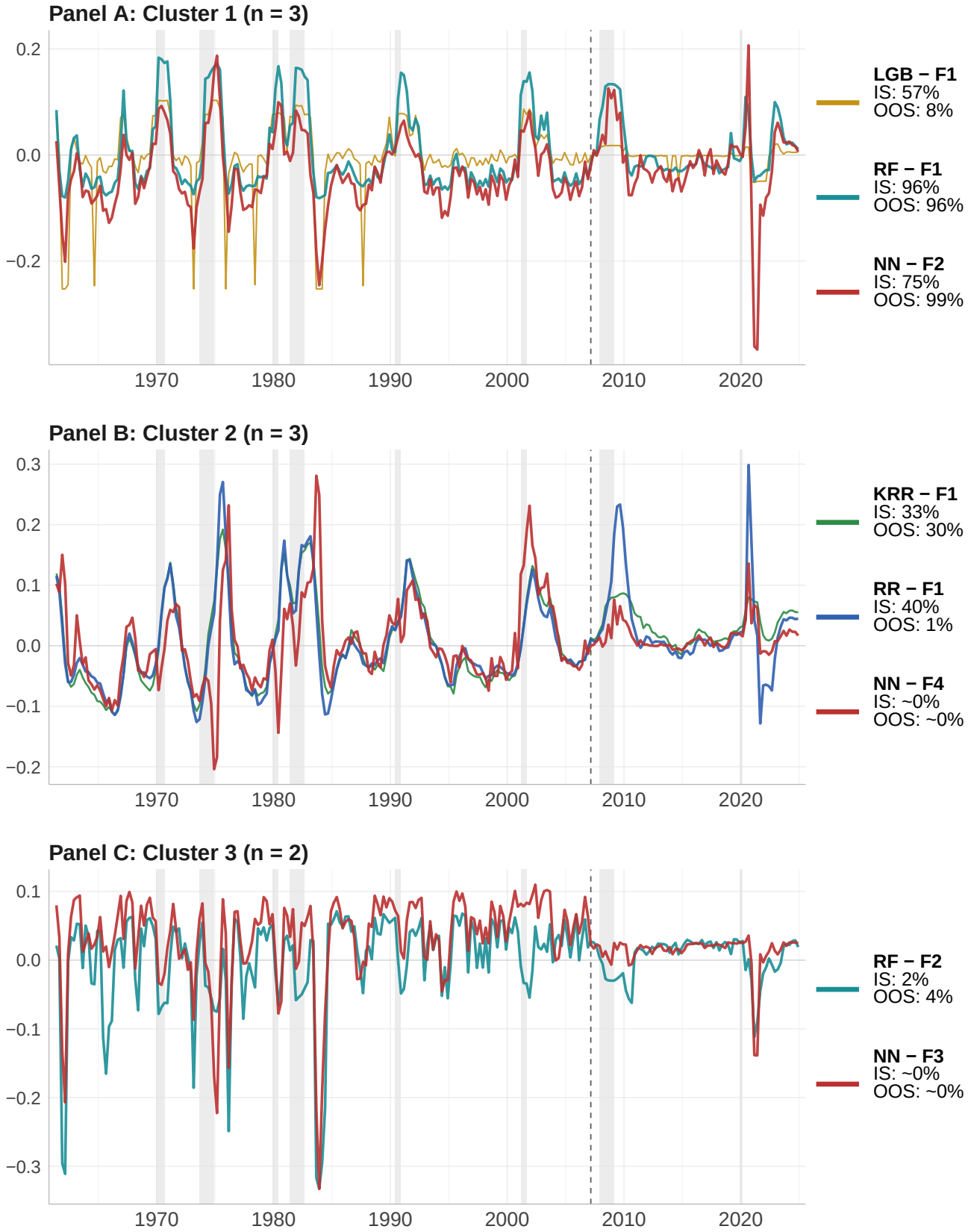
The third cluster (Panel C) pairs LGB–F3 (C_k^{OOS}: 28%) with RF–F3 (C_k^{OOS}: 3%). This is the hardest signal to interpret. For LGB, it contributes meaningfully to C_k^{OOS} despite negligible C_k^{IS}, suggesting it captures dynamics that matter primarily for the evaluation period. The time series shows some medium-frequency variation but no immediately recognizable economic pattern.

Unemployment Rate. A similar three-cluster structure emerges (Figure 5). The first cluster (Panel A) again groups RF–F1 (C_k^{IS}: 96%, C_k^{OOS}: 96%), LGB–F1, and NN–F2. The signal mirrors past unemployment conditions and business-cycle turning points. Random forests produce the sharpest version, resembling a Markov-switching latent state that classifies periods as recession or expansion, while the neural network's second factor provides a smoother rendition that blends elements of the RF and LGB signals. The dominance of this cluster is even more pronounced than for GDP, consistent with unemployment being driven overwhelmingly by a single business-cycle dimension.

The second cluster (Panel B) groups KRR–F1 (C_k^{IS}: 33%, C_k^{OOS}: 32%) and RR–F1 (C_k^{IS}: 40%, C_k^{OOS}: 32%) with NN–F4. As for GDP, the two least-squares-based models produce nearly identical in-sample trajectories, capturing a smooth cyclical component. Out of sample, KRR is again much more muted than RR, particularly around the Great Recession and COVID, where ridge regression exhibits large swings while KRR barely moves.

The third cluster (Panel C) pairs RF–F2 (C_k^{IS}: 2%, C_k^{OOS}: 3%) with NN–F3, both carrying negligible predictive contribution. The signal displays the Great Moderation pattern seen in the

Figure 5: Implied factors for unemployment rate — clusters ($h = 1$, 2007Q1–2024Q4)



Notes: Implied-factor signals are grouped using hierarchical clustering based on the similarity metric introduced in Appendix A.2. Absolute similarity is used to ensure robustness to sign indeterminacy. Legend entries report the model, factor index, and in-sample (C_k^{IS}) and out-of-sample (C_k^{OOS}) predictive contribution.

GDP analysis: high-amplitude fluctuations before the mid-1980s, followed by markedly reduced variability.

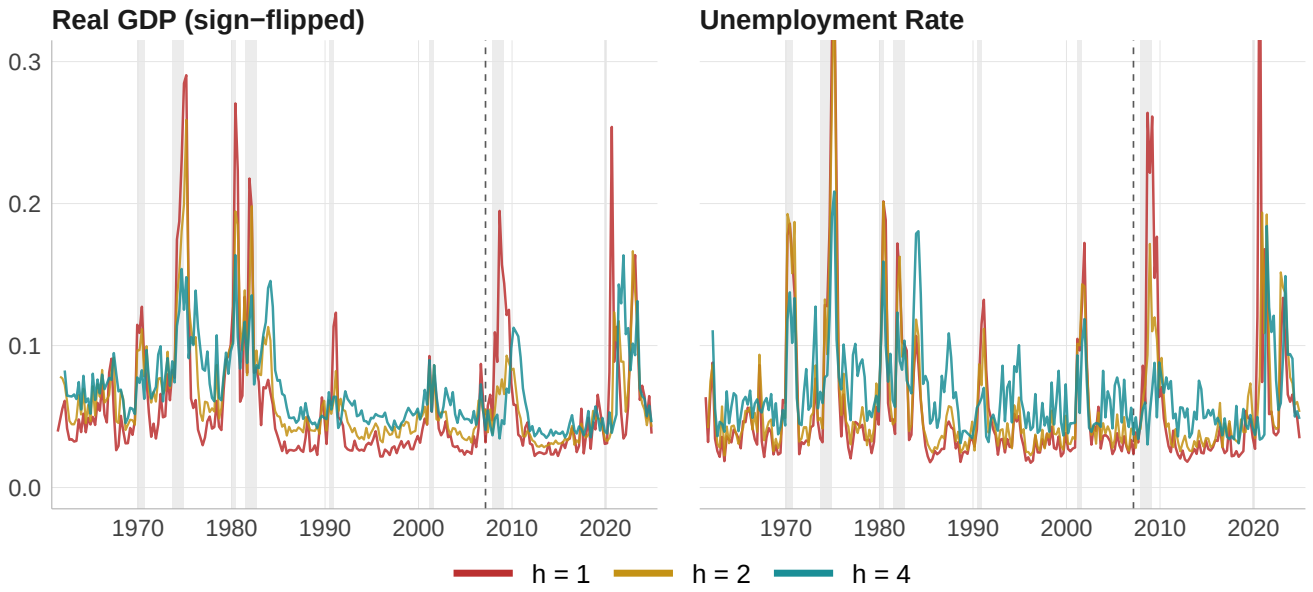
Across all three target variables, the clustering reveals a recurring organizational principle: implied factors from different models and different factor indices track the same underlying economic dynamics, differing primarily in amplitude and persistence rather than in the nature of the signal. Nonlinear models (RF, LGB, NN) tend to produce sharper, more regime-like signals, while least-squares-based models (RR, KRR) yield smoother representations of the same underlying variation.

Lone wolves. Not all implied factors join the dominant clusters. Some remain singletons—factors that do not cluster with any factor from another model—yet carry substantial predictive content. The most notable example is NN–F1 for real GDP and the unemployment rate, which consistently fails to cluster with any other model's factors despite accounting for a meaningful share of the neural network's C_k^{IS} and C_k^{OOS}. This suggests that the neural network's leading factor captures a predictive signal that is genuinely orthogonal to those exploited by tree-based and kernel methods. Other lone wolves contribute modestly and appear largely idiosyncratic.

Figure 6 plots NN–F1 for real GDP and the unemployment rate across all three forecast horizons. Despite being orthogonal to the factors extracted by competing models, the signal is strikingly consistent across horizons: for both targets, the factor displays high-amplitude fluctuations prior to the mid-1980s followed by sharply reduced variability, a pattern consistent with the Great Moderation. This low-frequency envelope is reminiscent of stochastic volatility measures or macroeconomic uncertainty indices, suggesting that the neural network's leading factor captures a time-varying uncertainty signal rather than a level or direction predictor. The factor's persistence across all horizons—under direct forecasting, where each horizon corresponds to a distinct predictive mapping—indicates that the neural network consistently identifies this uncertainty dimension as its primary feature, regardless of the forecast target or horizon. That no other model's factors correlate with this signal underscores the distinctiveness of the neural network's learned representation: while tree-based and kernel methods focus on regime classification and smooth cyclical components, the neural network additionally extracts a volatility channel that is invisible to the other estimators.

Additional clustering results for alternative horizons are reported in Appendix B. The identified clusters remain largely stable across forecast horizons ($h = 2, 4$).

Figure 6: Neural network "lone wolf" factor (NN–F1) across forecast horizons



Notes: This figure plots the first implied factor of the neural network (NN–F1) for real GDP and the unemployment rate across all three forecast horizons ($h = 1, 2, 4$), for the 2007Q1–2024Q4 evaluation period. This factor consistently fails to cluster with any factor from other models. Shaded areas indicate NBER recession dates; the dashed vertical line marks the start of the out-of-sample evaluation window.

3.5 Comparison to Traditional Factor Methods

[Work in progress.] A central question raised by the implied-factor framework is how the factors extracted from supervised ML models relate to those obtained by traditional factor methods commonly used in macroeconomic forecasting. Traditional approaches—principal component regression (PCR), partial least squares (PLS), and recent supervised variants such as sPCA (Bair et al., 2006), SPCA (Hounyo and Li, 2025), and SsPCA (Hounyo and Li, 2025)—extract factors from the predictor panel X either unsupervised (PCA) or with varying degrees of supervision from the target y. These factors summarize common variation or target-relevant variation in X. In contrast, implied factors arise from the eigendecomposition of the cross-sample hat matrix H and thus summarize the predictive mapping itself. The distinction matters because ML models can exploit nonlinear feature interactions and basis expansions that are invisible to linear factor methods, potentially uncovering different predictive dimensions.

We plan to expand this comparison substantially. In particular, we aim to compare the implied factors—both model-specific and clustered across models—to the traditional factors along several dimensions: (i) correlation structure, to assess whether the supervised basis expansion by ML materially changes the extracted factors or whether the linear and nonlinear methods converge on similar signals; (ii) predictive content, to evaluate whether implied factors capture forecast-relevant variation beyond what traditional factors provide; and (iii) economic interpretability, to determine whether the clustered implied factors load on the same macroeconomic themes as

PCA-based factors (e.g., real activity, prices, financial conditions) or whether ML supervision reveals additional predictive dimensions. This comparison is particularly informative because, if implied factors substantially overlap with traditional factors, it would suggest that the forecasting gains from ML models stem primarily from flexible nonlinear transformations of the same underlying signals rather than from the discovery of genuinely different latent predictors.

Table B.1 (Appendix B) provides an initial comparison of forecasting accuracy. We compare the implied-factor specifications (IF–m) against PCR, PLS, sPCA, SPCA, and SsPCA across all three targets and horizons under expanding-window evaluation. Columns labeled IF–KRR, IF–LGB, IF–NN, IF–RF, and IF–RR denote forecasts from the $K = 3$ restricted implied-factor representation of each ML model, not from the full models themselves. Overall, the implied-factor specifications frequently outperform the traditional approaches. The mixed-factor specifications (MF–LM and MF–LSO), which pool implied factors across models, appear particularly promising, often ranking among the top performers. Pooling is beneficial because it combines complementary predictive signals: as the clustering analysis in Section 3.4 showed, different ML models extract factors that are highly correlated in their leading components but can diverge in higher-order factors. By aggregating across these model-specific factors, the mixed specifications exploit a broader set of predictive dimensions while remaining parsimonious.

3.6 Cost-Benefits Analysis of Realignment and Combination

[Work in progress.]

3.7 Local Projections with Implied Factors

[Work in progress.] A natural extension of the implied-factor framework is to use the extracted factors as controls in local projection regressions for estimating impulse response functions to structural shocks. By replacing high-dimensional predictor panels with a small number of implied factors, we can achieve parsimonious conditioning sets that inherit the nonlinear feature representations learned by the underlying ML models. We plan to estimate impulse responses to monetary policy shocks—identified via narrative (Romer and Romer) and high-frequency approaches—using implied factors from random forests and neural networks as controls, and compare the resulting responses to those obtained with standard PCA-based factor augmentation.

4 Conclusion

We have shown that any supervised learning algorithm whose predictions are linear in the training outcomes admits a linear factor representation—its implied factors. The singular value decomposition of the cross-sample hat matrix yields a small number of latent predictors that reproduce the original machine learning forecasts via simple OLS, providing a "linear skeleton" that preserves predictive content while offering a low-dimensional, interpretable summary. Empirically, applied to quarterly forecasting of U.S. GDP, inflation, and unemployment, we find that as few as two or three implied factors suffice to match the out-of-sample accuracy of the full models. Clustering these factors across models reveals that the signals extracted by competing algorithms—linear and nonlinear alike—organize into a few common economic themes, suggesting that cross-model differences are largely quantitative rather than conceptual. Implied factors also compare favorably to traditional supervised factor methods (PCR, PLS, sPCA, and variants), with mixed-factor specifications that pool across models appearing particularly promising.

References

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

Bair, E., Hastie, T., Paul, D., and Tibshirani, R. (2006). Prediction by supervised principal components. *Journal of the American Statistical Association*, 101(473):119–137.

Bańbura, M., Giannone, D., and Reichlin, L. (2010). Large bayesian vector auto regressions. *Journal of Applied Econometrics*, 25(1):71–92.

Bianchi, D., Büchner, M., and Tamoni, A. (2021). Bond risk premiums with machine learning. *The Review of Financial Studies*, 34(2):1046–1089.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Fan, J., Ke, Z. T., Liao, Y., and Neuhierl, A. (2022). Structural deep learning in conditional asset pricing. *Available at SSRN 4117882*.

Farrell, M. H., Liang, T., and Misra, S. (2021). Deep neural networks for estimation and inference. *Econometrica*, 89(1):181–213.

Faust, J. and Wright, J. H. (2013). Forecasting inflation. In *Handbook of Economic Forecasting*, volume 2, pages 2–56. Elsevier.

Gao, Z. and Tsay, R. S. (2024). Supervised dynamic PCA: Linear dynamic forecasting with many predictors. *Journal of the American Statistical Association*, 120(550).

Geertsema, P. and Lu, H. (2023). Instance-based Explanations for Gradient Boosting Machine Predictions with AXIL Weights. *arXiv preprint arXiv:2301.01864*.

Ghorbani, A. and Zou, J. (2019). Data Shapley: Equitable valuation of data for machine learning. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2242–2251.

Goulet Coulombe, P. (2024). The macroeconomy as a random forest. *Journal of Applied Econometrics*, 39:401–421.

Goulet Coulombe, P. (2025a). A neural Phillips curve and a deep output gap. *Journal of Business & Economic Statistics*, 43(3):669–683.

Goulet Coulombe, P. (2025b). Ordinary least squares as an attention mechanism. Technical report.

Goulet Coulombe, P., Göbel, M., and Klieber, K. (2024). Dual interpretation of machine learning forecasts. *Available at SSRN 5029492*.

Gu, S., Kelly, B., and Xiu, D. (2021). Autoencoder asset pricing models. *Journal of Econometrics*, 222(1):429–450.

Hamilton, J. D. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica*, 57(2):357–384.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Hauzenberger, N., Huber, F., and Klieber, K. (2023). Real-time inflation forecasting using non-linear dimension reduction techniques. *International Journal of Forecasting*, 39(2):901–921.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

Hounyo, U. and Li, Z. (2025). Forecasting economic time series in the presence of weak factors: Multiple supervised learning-based approach. *International Journal of Forecasting*. In press.

Huang, D., Jiang, F., Li, K., Tong, G., and Zhou, G. (2022). Scaled pca: A new approach to dimension reduction. *Management Science*, 68(3):1678–1695.

Jia, R., Dao, D., Wang, B., Hubis, F. A., Hynes, N., Gürel, N. M., Li, B., Zhang, C., Song, D., and Spanos, C. J. (2019). Towards efficient data valuation based on the Shapley value. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, pages 1167–1176.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30, pages 3146–3154.

Kelly, B. and Pruitt, S. (2013). Market expectations in the cross-section of present values. *The Journal of Finance*, 68(5):1721–1756.

Koh, P. W. and Liang, P. (2017). Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1885–1894.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 4768–4777. Curran Associates Inc.

Pruthi, G., Liu, F., Kale, S., and Sundararajan, M. (2020). Estimating training data influence by tracing gradient descent. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, pages 19920–19930. Curran Associates Inc.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1135–1144. Association for Computing Machinery.

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

Stock, J. H. and Watson, M. W. (2002). Forecasting using principal components from a large number of predictors. *Journal of the American Statistical Association*, 97(460):1167–1179.

A Appendix

A.1 Construction of Cross-Sample Hat Matrices

As discussed in Section 2.2, the implied-factor decomposition requires the cross-sample hat matrices $\boldsymbol{H}_{\text{train}}$ and $\boldsymbol{H}_{\text{test}}$ satisfying $\hat{\boldsymbol{y}}_{\text{train}} = \boldsymbol{H}_{\text{train}}\boldsymbol{y}_{\text{train}}$ and $\hat{\boldsymbol{y}}_{\text{test}} = \boldsymbol{H}_{\text{test}}\boldsymbol{y}_{\text{train}}$. This appendix details how these matrices are obtained for each model class used in our empirical analysis. The exposition follows Goulet Coulombe et al. (2024), adapted to our notation.

Ridge regression. The ridge estimator $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'_{\text{train}}\boldsymbol{X}_{\text{train}} + \lambda I_P)^{-1}\boldsymbol{X}'_{\text{train}}\boldsymbol{y}_{\text{train}}$ yields hat matrices directly:

$$\boldsymbol{H}_{\text{train}} = \boldsymbol{X}_{\text{train}}(\boldsymbol{X}'_{\text{train}}\boldsymbol{X}_{\text{train}} + \lambda I_P)^{-1}\boldsymbol{X}'_{\text{train}} \qquad \boldsymbol{H}_{\text{test}} = \boldsymbol{X}_{\text{test}}(\boldsymbol{X}'_{\text{train}}\boldsymbol{X}_{\text{train}} + \lambda I_P)^{-1}\boldsymbol{X}'_{\text{train}} \tag{A.1}$$

Equivalently, these can be expressed via the dual formulation. Defining the kernel matrix $\boldsymbol{K} = \boldsymbol{X}_{\text{train}}\boldsymbol{X}'_{\text{train}}$ and the cross-sample kernel vector $K_j = X_j\boldsymbol{X}'_{\text{train}}$ for test observation j, the matrix inversion lemma gives $\boldsymbol{H}_{\text{train}} = \boldsymbol{K}(\boldsymbol{K} + \lambda I_N)^{-1}$ and each row of $\boldsymbol{H}_{\text{test}}$ as $K_j(\boldsymbol{K} + \lambda I_N)^{-1}$. The dual route involves inverting an $N \times N$ matrix rather than a $P \times P$ one, which is computationally advantageous when $P \gg N$.

Kernel ridge regression. KRR replaces the linear inner products in the dual formulation with a kernel function $\mathcal{K}(\cdot, \cdot)$ that implicitly maps inputs into a higher-dimensional feature space Φ. The kernel matrix becomes $\boldsymbol{K}_{ij} = \mathcal{K}(X_i, X_j)$ for training pairs and $K_{ji} = \mathcal{K}(X_j, X_i)$ for a test observation j against training observations. The hat matrices take the same form as in the dual ridge case:

$$\boldsymbol{H}_{\text{train}} = \boldsymbol{K}(\boldsymbol{K} + \lambda I_N)^{-1} \qquad \boldsymbol{H}_{\text{test},j\cdot} = K_j(\boldsymbol{K} + \lambda I_N)^{-1} \tag{A.2}$$

where K_j is the $1 \times N$ vector of kernel evaluations between test observation j and all training observations. In our implementation we use a Laplacian kernel, $\mathcal{K}(X_i, X_j) = \exp(-\|X_i - X_j\|/\sigma)$.

Neural networks. For a feedforward neural network with $L-1$ hidden layers and a linear output layer, the prediction for observation j is $\hat{y}_j = \Psi(X_j)\hat{\boldsymbol{\theta}}_L$, where $\Psi(X_j)$ denotes the $1 \times n_L$ vector of activations from the last hidden layer and $\hat{\boldsymbol{\theta}}_L$ the output-layer weights. Holding the lower layers fixed, the output layer is a linear model in the generated features $\Psi(\boldsymbol{X})$. An auxiliary ridge regression of $\boldsymbol{y}_{\text{train}}$ on $\Psi(\boldsymbol{X}_{\text{train}})$ with a small regularization parameter λ closely replicates

the network's predictions. The hat matrices are then:

$$\boldsymbol{H}_{\text{train}} \cong \boldsymbol{K}^{\Psi}(\boldsymbol{K}^{\Psi} + \lambda I_N)^{-1} \qquad \boldsymbol{H}_{\text{test},j\cdot} \cong K_j^{\Psi}(\boldsymbol{K}^{\Psi} + \lambda I_N)^{-1} \tag{A.3}$$

where $\boldsymbol{K}^{\Psi} = \Psi(\boldsymbol{X}_{\text{train}})\Psi(\boldsymbol{X}_{\text{train}})'$ and $K_j^{\Psi} = \Psi(X_j)\Psi(\boldsymbol{X}_{\text{train}})'$. The "$\cong$" reflects the fact that the network's gradient-based optimization, early stopping, and dropout introduce implicit regularization that departs slightly from the closed-form ridge solution. In practice, the replication accuracy exceeds 99% (Goulet Coulombe et al., 2024). When the network is ensembled over B random initializations, the hat matrices are averaged: $\boldsymbol{H} = \frac{1}{B}\sum_{b=1}^{B} \boldsymbol{H}^{(b)}$.

Random forests. A random forest averages predictions across B trees: $\hat{y}_j = \frac{1}{B}\sum_{b=1}^{B} \mathcal{T}_b(X_j)$. Each tree $\mathcal{T}_b$ partitions the feature space into leaves; the prediction for observation j is the average of training responses in the leaf containing X_j. Denoting by $\mathcal{P}_b(X_j)$ the set of training indices sharing a leaf with j in tree b, the per-tree weight on training observation i is $w_{bji} = \mathbb{I}(i \in \mathcal{P}_b(X_j))/|\mathcal{P}_b(X_j)|$. Aggregating across trees, the (j, i) entry of $\boldsymbol{H}_{\text{test}}$ is:

$$H_{\text{test},ji} = \frac{1}{B}\sum_{b=1}^{B} w_{bji}. \tag{A.4}$$

The in-sample matrix $\boldsymbol{H}_{\text{train}}$ is obtained analogously by setting j equal to a training index. Since each $w_{bji} \geq 0$ and $\sum_i w_{bji} = 1$ for every tree, all weights are nonnegative—random forests cannot assign negative proximity to any training observation.

Gradient boosting. In gradient boosting the prediction is $\hat{y}_j = \nu\sum_{s=1}^{S} \mathcal{T}_s(X_j)$, where ν is the learning rate and $\mathcal{T}_s$ are shallow trees fit sequentially to pseudo-residuals. Because each tree beyond the first targets residuals from the previous iteration rather than the original $\boldsymbol{y}_{\text{train}}$, the leaf-averaging decomposition of random forests does not directly apply. We recover training observation weights using the AXIL (Additive eXplanations with Instance Loadings) algorithm of Geertsema and Lu (2023), which traces the recursive tree-building sequence to express $\hat{y}_j$ as a linear combination of the original training responses: $\hat{y}_j = \sum_{i=1}^{N} w_{ji} y_i$. The resulting weights define the rows of $\boldsymbol{H}_{\text{test}}$ and $\boldsymbol{H}_{\text{train}}$.

A.2 Clustering Implied Factors into Common Signals

Our objective here is to understand whether we can identify *common signals*: groups of implied factors that exhibit strong comovement across models and plausibly reflect the same macroeconomic dynamics.

Clustering implied factors serves four key purposes. First, it provides a systematic way to assess cross-model agreement in the latent information extracted by different forecasting approaches. Second, it helps separate robust economic signals from model-specific noise. Third, it improves interpretability by organizing many implied factors into a smaller set of economically meaningful groups. Finally, it allows us to evaluate whether certain classes of models are better at capturing specific signals (e.g., nonlinear recession dynamics or volatility shifts associated with the Great Moderation) that may be weak or absent in more restrictive specifications.

Construction of the factor panel. Let $m = 1, \ldots, M$ index forecasting models and let $j = 1, \ldots, K_m$ index the factors extracted from model m. For a given out-of-sample vintage and forecast horizon, each model produces a matrix of implied factors $\mathbf{F}^{(m)} = \left(F_{t,1}^{(m)}, \ldots, F_{t,K_m}^{(m)}\right)_{t=1}^{T}$, obtained by stacking in-sample and out-of-sample estimates. The stacked factor panel is then formed as

$$\mathbf{X}_t = \left(F_{t,1}^{(1)}, \ldots, F_{t,K_1}^{(1)}, \ldots, F_{t,1}^{(M)}, \ldots, F_{t,K_M}^{(M)}\right), \qquad t = 1, \ldots, T,$$

where each column of $\mathbf{X}$ corresponds to a model–factor pair.

Scaling and sign normalization. Implied factors are not uniquely identified and may differ across models by sign or scale. Prior to clustering, each factor may optionally be rescaled using the coefficient $\beta_j^{(m)}$ from the forecasting regression of model m. In our baseline specification, factors are left unscaled.

After clustering (described below), we enforce sign coherence in two steps. First, within each cluster we select an anchor series—the factor with the largest average absolute correlation with the other cluster members—and flip the remaining factors in sign so that their correlation with the anchor is nonnegative. Second, we align each cluster to the forecast target: if the average correlation between the cluster's factors and the in-sample target variable Y is negative, the entire cluster is flipped in sign. Only the training portion of Y is used for this alignment to avoid look-ahead bias. Together, these two steps resolve sign indeterminacy both within and across clusters while preserving relative dynamics.

Similarity measure and hierarchical clustering. Let $\mathbf{X}_i$ and $\mathbf{X}_j$ denote two columns of the stacked factor panel. We define similarity using the absolute Pearson correlation, $S_{ij} = |\text{corr}(\mathbf{X}_i, \mathbf{X}_j)|$, computed with pairwise complete observations. The absolute value ensures invariance to sign reversals. We then define the distance $d_{ij} = 1 - S_{ij}$ and apply agglomerative hierarchical clustering with complete linkage. Clusters are obtained by cutting the dendrogram at height $1 - \underline{\rho}$, where $\underline{\rho}$ is a minimum absolute-correlation threshold. The resulting clusters are relabeled in descending order of size.

B Additional empirical results

B.1 Out-of-Sample Forecasting Performance

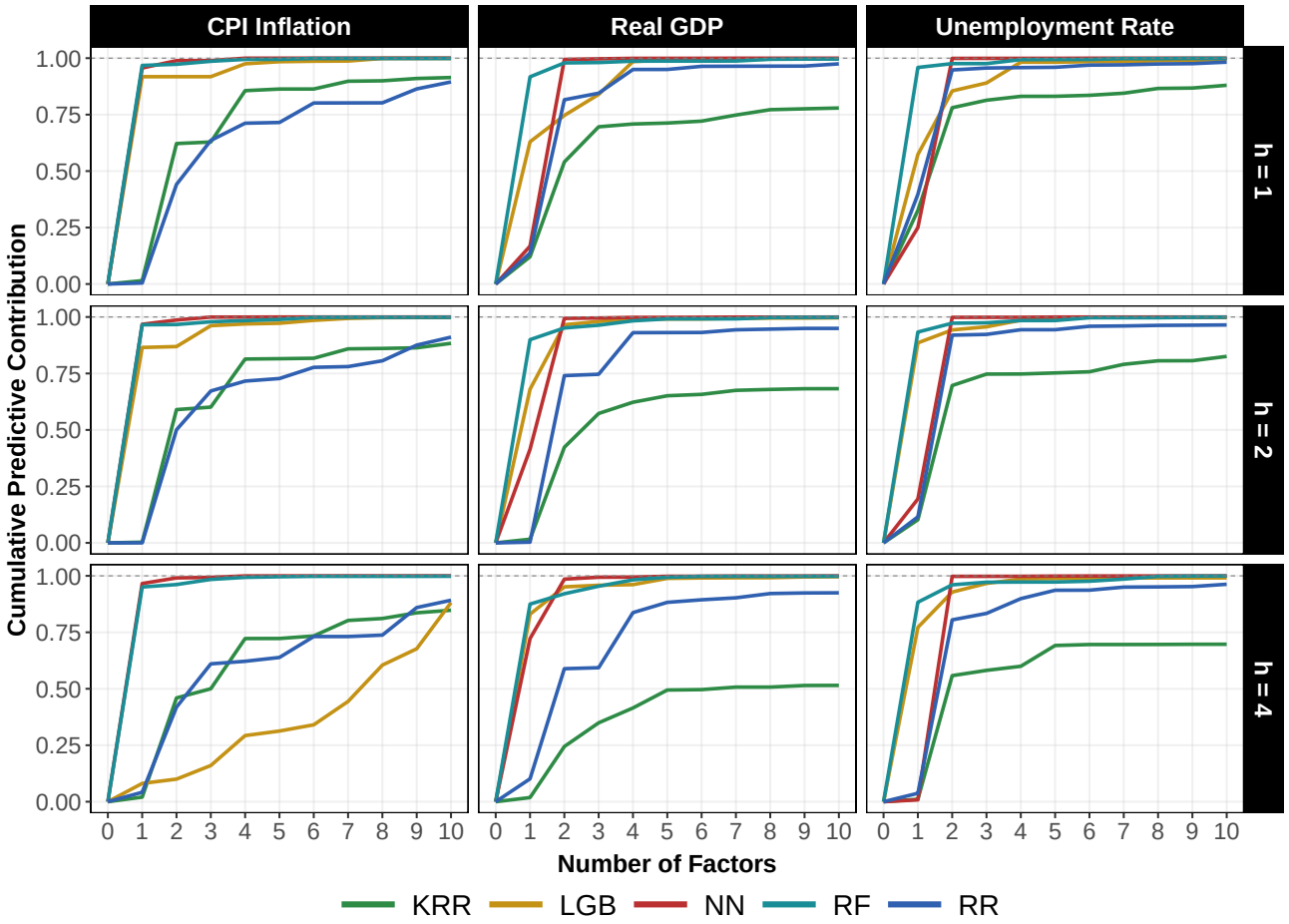
Table B.1: Factor-based forecasting — implied factors vs. traditional methods ($K = 3$)

	Traditional					*Implied Factors – ML model*						
	PCR	PLS	sPCA	SPCA	SsPCA	IF-KRR	IF-LGB	IF-NN	IF-RF	IF-RR	MF-LM	MF-LSO
Real GDP												
h=1	0.96	0.99	0.92	0.95	0.99	1.01	1.02	0.89	**0.85****	0.94	1.12	**0.84***
h=2	0.96	1.04	0.95	0.98	1.00	0.93	0.91	0.91	**0.86***	0.93	1.04	**0.84****
h=4	0.90	1.00	0.95	0.90	1.01	0.82*	**0.79***	**0.79***	0.83*	0.86	0.95	0.82*
CPI Inflation												
h=1	1.16	1.09	1.14	1.05	1.05	1.10	1.03	1.00	**0.94**	1.15	**0.93**	1.00
h=2	1.05	1.22	1.06	0.96	0.96	0.94	0.98	0.89	**0.85***	1.02	**0.81***	0.87*
h=4	1.12	1.34	1.20	1.32	1.23	**0.89***	1.01	0.98	0.96	1.06	**0.86***	0.93
Unemployment Rate												
h=1	0.73	0.80	**0.66**	0.67	**0.58***	1.00	0.77	0.74	0.71	0.76	0.74	0.68
h=2	0.72	0.79	**0.66***	0.75	**0.66***	0.89	0.83	0.68*	0.74	0.76	0.70*	0.70*
h=4	0.78*	0.98	**0.73****	0.87	0.93	0.88*	0.80**	0.80*	0.79**	0.81*	**0.76****	0.82*

RMSE relative to AR4 benchmark (expanding-window evaluation). Traditional methods: PCR (Principal Component Regression), PLS (Partial Least Squares), sPCA (Scaled PCA), SPCA (Supervised PCA), SsPCA (Supervised + Scaled PCA). Implied factor models use $K = 3$ factors per model. MF-LM/MF-LSO combine implied factors from all five ML models. */**/*** denote one-sided DM test significance at 10/5/1%. Best: **bold green**; second-best: **bold**. The COVID period (2020Q1–2020Q4) is excluded.

B.2 Implied Factors vs. Full Model

Figure B.1: In-Sample Predictive Contribution of Implied Factors (2007Q1–2024Q4)



Notes: This figure reports the *cumulative in-sample predictive contribution* of implied factors within each forecasting model, as increasingly many factors are included. Columns correspond to target variables, rows correspond to forecast horizons ($h = 1, 2, 4$), and panels correspond to forecasting models. A steeper accumulation (i.e., reaching high cumulative contribution with few factors) indicates that predictive content is concentrated in a small number of implied factors. The in-sample training period is 1961Q2–2006Q4; the out-of-sample evaluation period is 2007Q1–2024Q4. The COVID period (2020Q1–2020Q4) is excluded from the out-of-sample evaluation.

B.3 Clustering Results

Figure B.2: Implied factors for real GDP — clusters ($h = 2$, 2007Q1–2024Q4).



Figure B.3: Implied factors for real GDP — clusters ($h = 4$, 2007Q1–2024Q4).

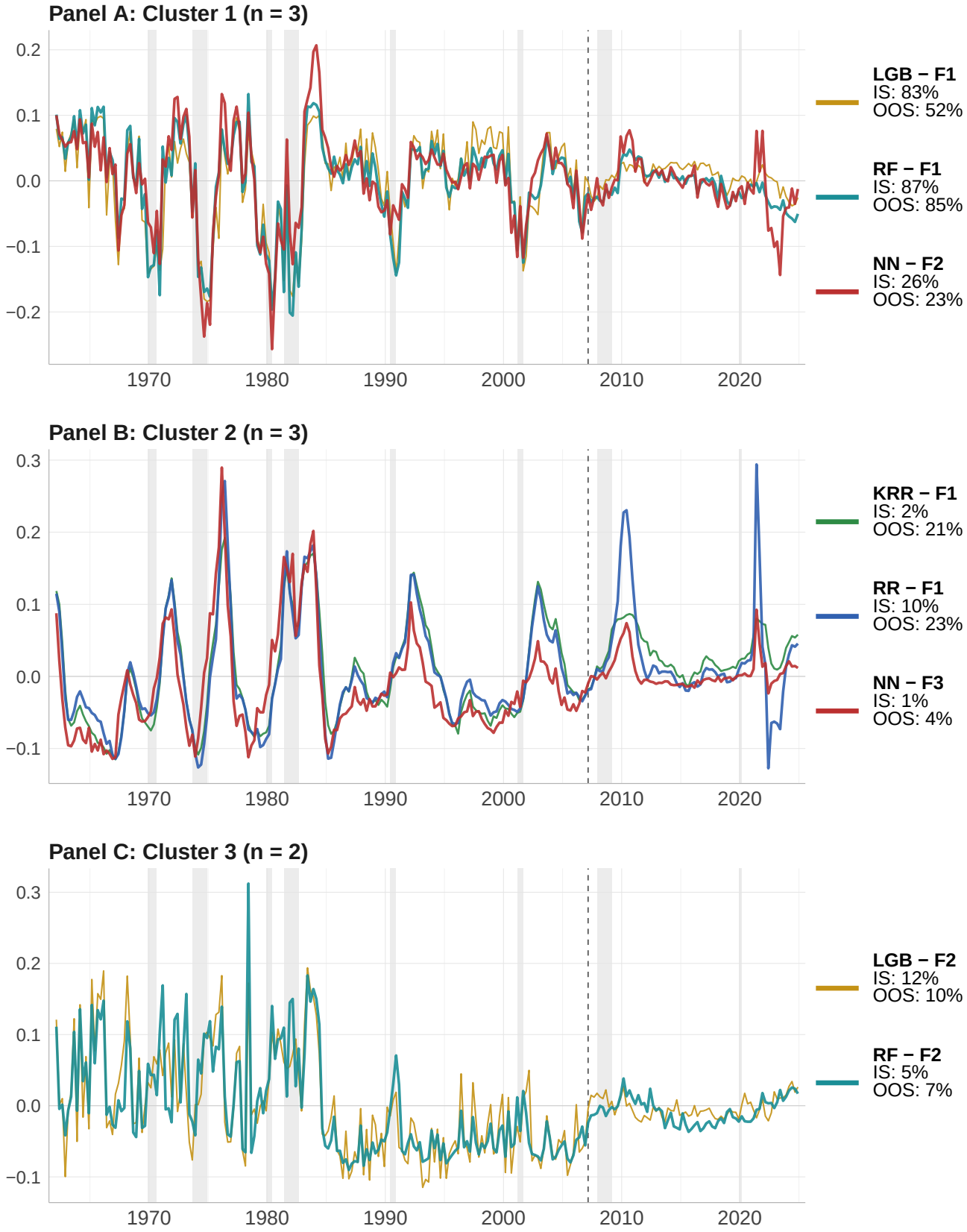


Figure B.4: Implied factors for CPI inflation — clusters ($h = 2$, 2007Q1–2024Q4).

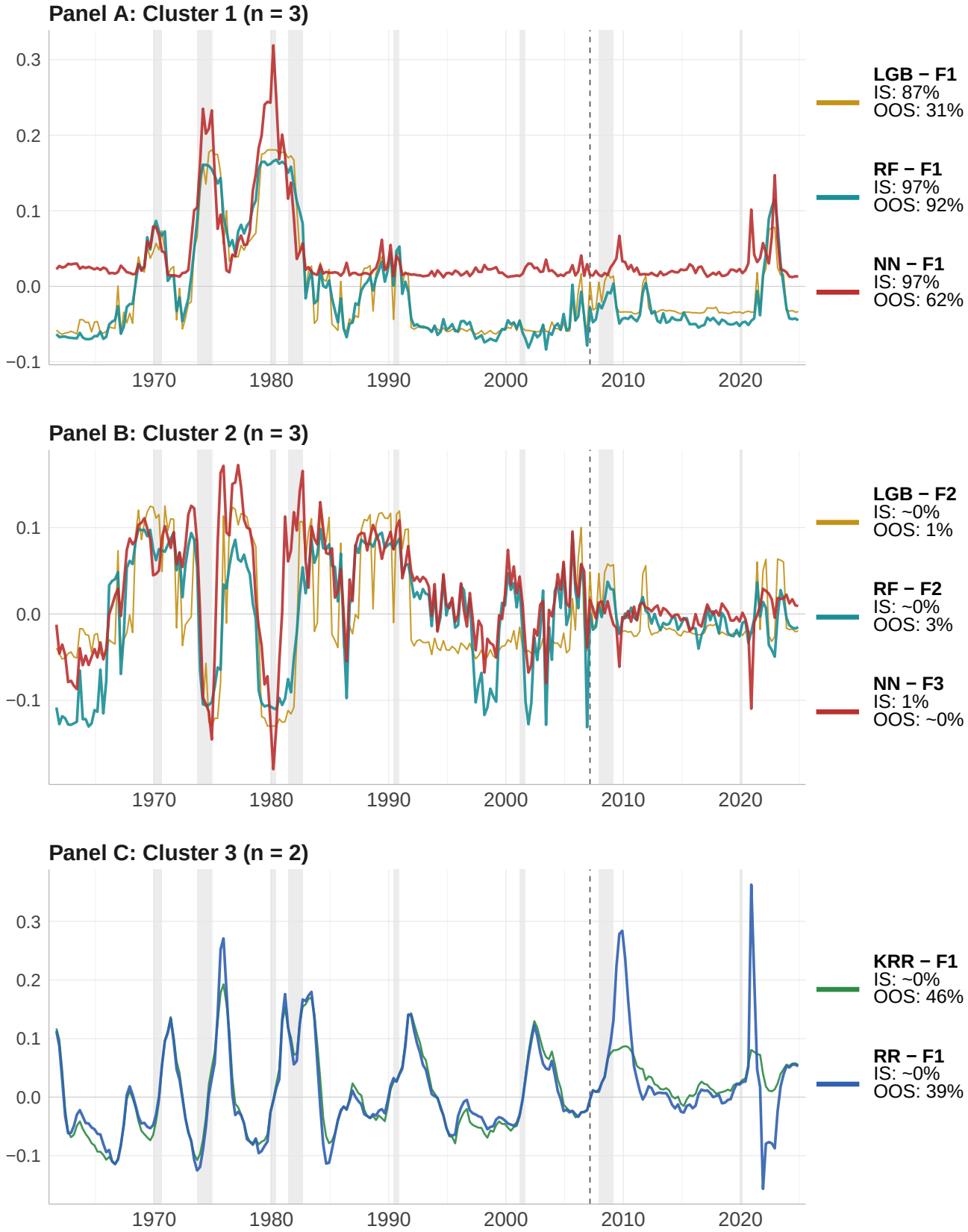


Figure B.5: Implied factors for CPI inflation — clusters ($h = 4$, 2007Q1–2024Q4).

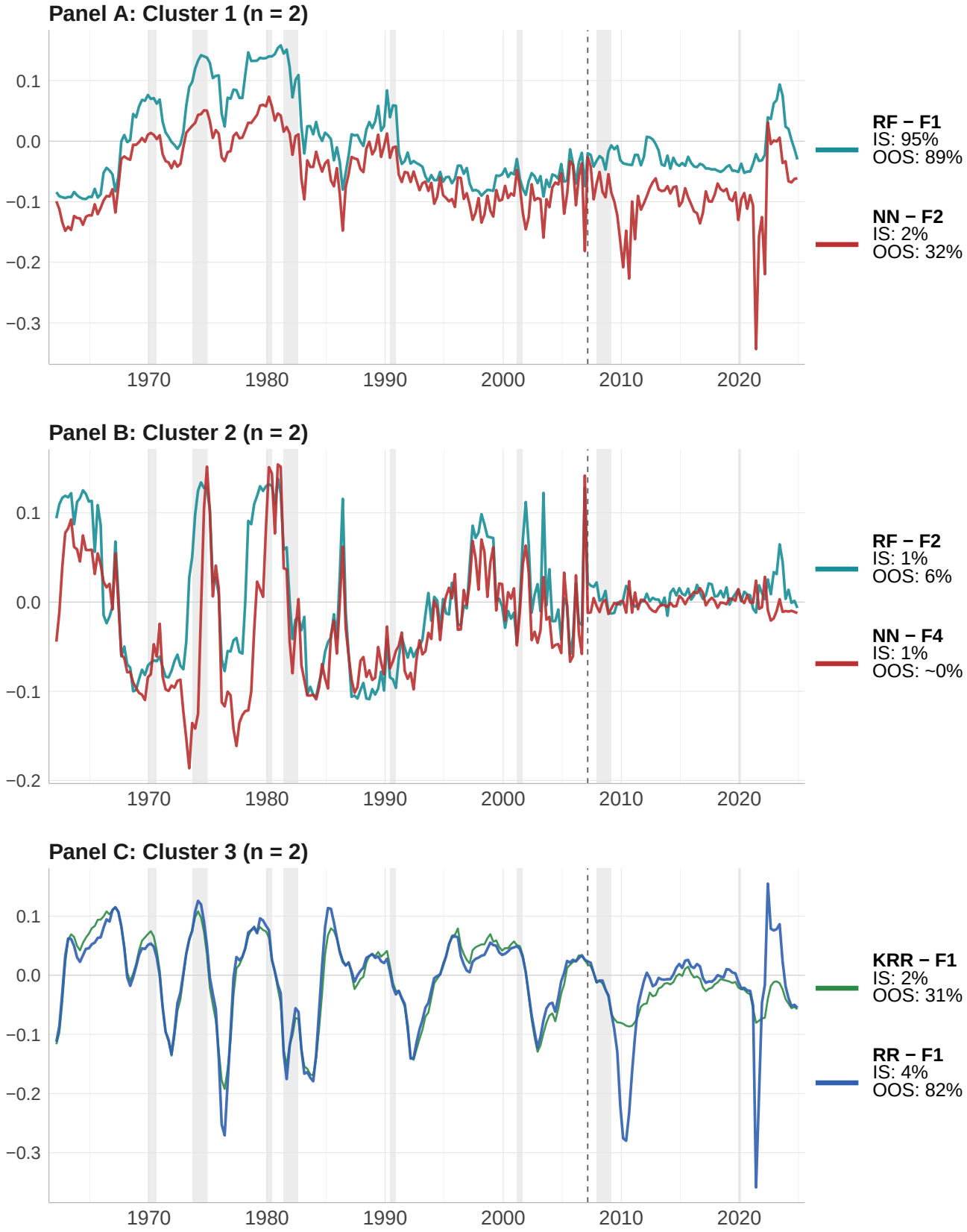


Figure B.6: Implied factors for unemployment rate — clusters ($h = 2$, 2007Q1–2024Q4).



Figure B.7: Implied factors for unemployment rate — clusters ($h = 4$, 2007Q1–2024Q4).

