

# Dynamic Allocation Macro Factor-Mimicking Portfolios

Matéo Molinaro

Lucien Chaudron

27 January 2026

## Abstract

This paper investigates whether macroeconomic information can be effectively translated into investment strategies through the use of macro factor-mimicking portfolios and predictive modeling. We focus on inflation as a representative macroeconomic variable and construct tradable portfolios that replicate inflation exposure by estimating stock-level sensitivities and forming long-short factor-mimicking portfolios. Building on this framework, we employ a range of statistical and machine-learning models to forecast inflation dynamics using a large macroeconomic dataset. These forecasts are then used to implement a dynamic asset allocation strategy that adjusts portfolio exposures in response to predicted macroeconomic conditions. We evaluate the performance of the proposed strategy in terms of return, risk, and risk-adjusted metrics, and compare it to static benchmark allocations. Our results provide evidence on the economic value of combining macro factor-mimicking portfolios with forecasting techniques for dynamic portfolio management.

## Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>   | <b>2</b>  |
| <b>2</b> | <b>Factor Mimicking Portfolios (FMP)</b>                    | <b>2</b>  |
| 2.1      | Data . . . . .  | 2         |
| 2.2      | Methodology . . . . .                                       | 2         |
| 2.2.1    | Measuring stock-level inflation exposure . . . . .          | 2         |
| 2.2.2    | Construction of factor-mimicking portfolios . . . . .       | 3         |
| 2.3      | Results . . . . .   | 3         |
| <b>3</b> | <b>Inflation Forecasting</b>                                | <b>6</b>  |
| 3.1      | Data . . . . .  | 6         |
| 3.2      | Methodology . . . . .                                       | 7         |
| 3.2.1    | Preprocessing . . . . .                                     | 7         |
| 3.2.2    | Regularization . . . . .                                    | 7         |
| 3.2.3    | Factor extraction . . . . .                                 | 7         |
| 3.2.4    | Estimated models . . . . .                                  | 7         |
| 3.2.5    | Expanding walk-forward cross-validation framework . . . . . | 7         |
| 3.2.6    | Results . . . . .   | 8         |
| <b>4</b> | <b>Dynamic Allocation FMP</b>                               | <b>12</b> |
| 4.1      | Methodology . . . . .                                       | 12        |
| 4.2      | Results . . . . .   | 12        |
| <b>5</b> | <b>Conclusion</b>   | <b>15</b> |
| <b>6</b> | <b>Appendix</b>   | <b>16</b> |
| 6.1      | Hyperparameter Grids . . . . .                              | 16        |

# 1 Introduction

Understanding and exploiting the relationship between macroeconomic conditions and asset price dynamics remains a central challenge in portfolio management. Macroeconomic variables such as inflation, interest rates, and business cycle indicators are widely recognized as key drivers of financial markets. However, these variables are not directly tradable and are difficult to incorporate in a systematic and operational manner into portfolio construction.

This project aims to address this challenge by combining macro factor-mimicking portfolios, macroeconomic forecasting, and dynamic asset allocation. We construct tradable portfolios whose returns are explicitly exposed to the dynamics of inflation. These portfolios serve as investable proxies for macroeconomic risks and provide a bridge between macroeconomic information and portfolio decisions. The analysis is organized into three main steps.

First, we construct macro factor-mimicking portfolios (FMPs) for inflation. Building on the methodology proposed by Mikheil Esakia and Felix Goltz (2023)[1], we estimate stock-level sensitivities to inflation. At each rebalancing date, we form a long-short portfolio by taking long positions in the top 10% and short positions in the bottom 10% of stocks ranked by their estimated inflation betas. This step provides a tradable representation of inflation risk.

Second, we model and forecast the dynamics of inflation. We employ a range of statistical and machine-learning techniques to predict future inflation using a broad information set consisting of 125 macroeconomic variables from the FRED-MD dataset. These models aim to capture both linear and non-linear relationships between macroeconomic indicators and inflation dynamics.

Third, we use these inflation forecasts to implement a dynamic asset allocation strategy. Portfolio weights are adjusted over time in response to the predicted evolution of inflation, with the objective of increasing exposure during favorable macroeconomic regimes and reducing exposure during adverse conditions.

Overall, this project investigates whether macro factor-mimicking portfolios, here applied to inflation, but extendable to any macroeconomic variable in the FRED-MD dataset, combined with predictive modeling, can be effectively leveraged to design dynamic allocation strategies that outperform static investment approaches.

## 2 Factor Mimicking Portfolios (FMP)

This section describes the methodology used to estimate the relationship between tradable equity returns and macroeconomic indicators, and to construct macro factor-mimicking portfolios.

### 2.1 Data

The empirical analysis relies on monthly data spanning January 1970 to December 2024. Equity prices and returns are obtained from the Center for Research in Security Prices (CRSP) and include all ordinary common shares listed on the NYSE, AMEX, and NASDAQ. The investment universe is restricted to the top 1,000 firms ranked by market capitalization at each date. Macroeconomic variables are sourced from the FRED-MD dataset. Our code is available on GitHub <sup>1</sup>

### 2.2 Methodology

#### 2.2.1 Measuring stock-level inflation exposure

Following Esakia and Goltz (2023), macroeconomic exposures are estimated at the individual stock level while explicitly controlling for market risk. The objective is to capture heterogeneity in firms' sensitivities to macroeconomic shocks without inducing unintended exposure to the aggregate equity market. For each stock  $i$ , we estimate the following bivariate regression:

$$R_{i,t} - R_{f,t} = \alpha_i + \beta_i^{\text{MKT}} (R_t^{\text{MKT}} - R_{f,t}) + \beta_i^{\text{macro}} (\text{Macro}_t - \mathbb{E}[\text{Macro}_t \mid t-1]) + \varepsilon_{i,t}, \quad (1)$$

---

<sup>1</sup><https://github.com/mateomolinaro1/dynamic-allocation-macro-fmp/tree/master>

where  $R_{i,t}$  denotes the return on stock  $i$ ,  $R_{f,t}$  is the risk-free rate (assumed to be zero in this study),  $R_t^{\text{MKT}}$  represents the market return, proxied by an equally weighted portfolio of all available stocks, and  $\text{Macro}_t - \mathbb{E}[\text{Macro}_t \mid t-1]$  captures innovations in the macroeconomic variable of interest. To improve out-of-sample reliability, several statistical adjustments are implemented. First, to balance estimation precision with time variation in macro exposures, parameters are estimated using weighted least squares with exponentially decaying weights:

$$\min_{\{\alpha_i, \beta_i\}} \sum_{t=1}^T \varepsilon_{i,t}^2 \times \exp\left(-|T-t| \frac{\ln(2)}{60}\right), \quad (2)$$

which assigns half of the total weight to the most recent five years of observations while retaining information from longer historical samples.

Second, estimation risk is explicitly addressed through Bayesian shrinkage following Vasicek (1973). The final macro exposure is obtained as a weighted average of the time-series estimate and a cross-sectional prior:

$$\beta_i^{\text{final}} = \beta_i^{\text{prior}} \frac{\sigma_{i,\text{TS}}^2}{\sigma_{i,\text{TS}}^2 + \sigma_{\text{XS}}^2} + \beta_i^{\text{estimate}} \left(1 - \frac{\sigma_{i,\text{TS}}^2}{\sigma_{i,\text{TS}}^2 + \sigma_{\text{XS}}^2}\right), \quad (3)$$

where  $\sigma_{i,\text{TS}}^2$  denotes the time-series sampling variance of the estimated beta and  $\sigma_{\text{XS}}^2$  the cross-sectional variance across stocks. This shrinkage mechanism assigns less weight to imprecisely estimated exposures and improves the stability of macro betas.

### 2.2.2 Construction of factor-mimicking portfolios

The estimated stock-level macro exposures are then used to construct macro factor-mimicking portfolios, which serve as tradable building blocks for dynamic asset allocation. At each monthly rebalancing date, stocks are sorted based on their Bayesian macro betas. The long leg of the inflation FMP consists of the top 10% of stocks with the highest estimated exposures, while the short leg includes the bottom 10%.

Prior to portfolio formation, Bayesian betas are standardized cross-sectionally using z-scores and winsorized at the 2nd and 98th percentiles to mitigate the impact of extreme values. Transaction costs are not incorporated at the FMP construction stage, as the dynamic allocation strategy can directly trade the assets rather than factor-mimicking portfolios. Transaction costs of 10 basis points are, however, explicitly accounted for in the dynamic allocation phase.

## 2.3 Results

Figure 1 shows the cross-sectional distribution of Bayesian inflation betas across all stocks and time periods. The distribution is centered close to zero but exhibits substantial dispersion, indicating meaningful heterogeneity in firms' inflation exposure. This heterogeneity provides scope for constructing long-short factor-mimicking portfolios. Note that the spikes at the tails are artificial as they reflect the fact that we winsorized/clipped the results for illustrative purpose.

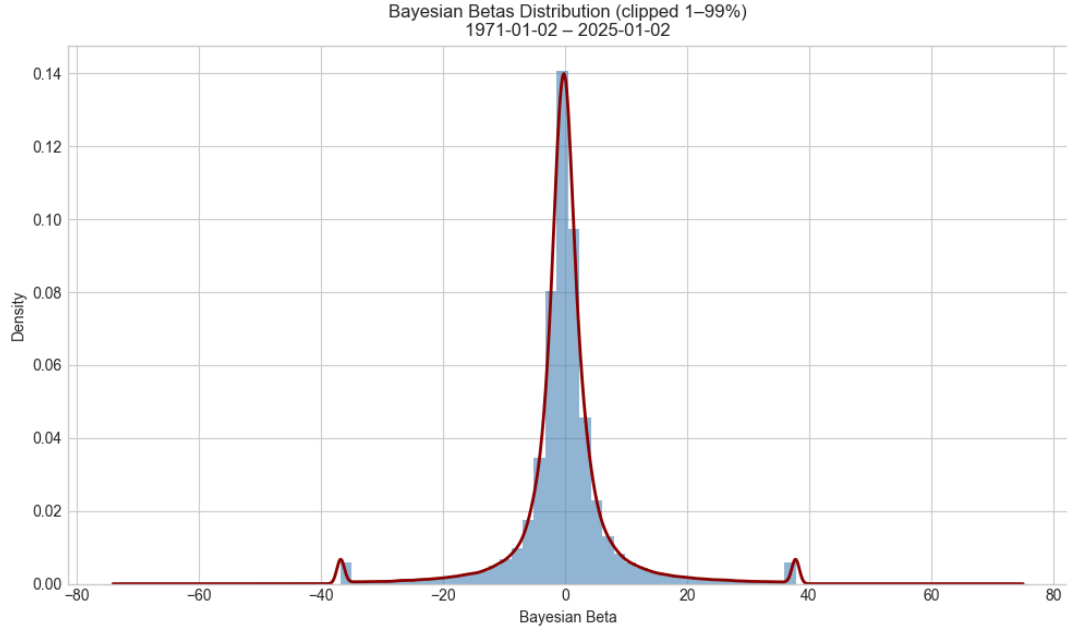


Figure 1: Distribution of the stock-level Bayesian betas across time.

Figure 2 illustrates the evolution over time of the cross-sectional mean, as well as the 10th and 90th percentiles, of Bayesian inflation betas. While the average exposure remains relatively stable, the dispersion varies over time, increasing during certain macroeconomic regimes. This time variation highlights the importance of allowing macro sensitivities to evolve rather than assuming constant exposures.

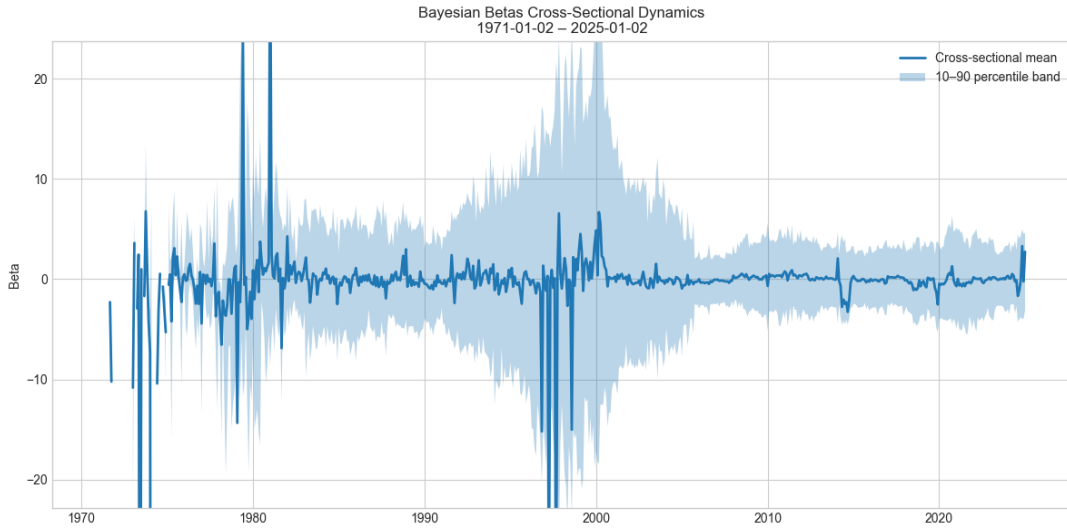


Figure 2: Time-series dynamics of the cross-sectional average as well as 10-90 percentiles.

|                | Bayesian Betas | Non-Bayesian Betas |
|----------------|----------------|--------------------|
| <b>Mean</b>    | -0.17          | -0.17              |
| <b>Median</b>  | -0.10          | -0.12              |
| <b>Std Dev</b> | 126.43         | 135.43             |
| <b>Min</b>     | -60456.15      | -62292.59          |
| <b>Max</b>     | 22780.94       | 25333.50           |

Figure 3: Summary statistics comparison between Bayesian an non-bayesian betas across stocks and time.

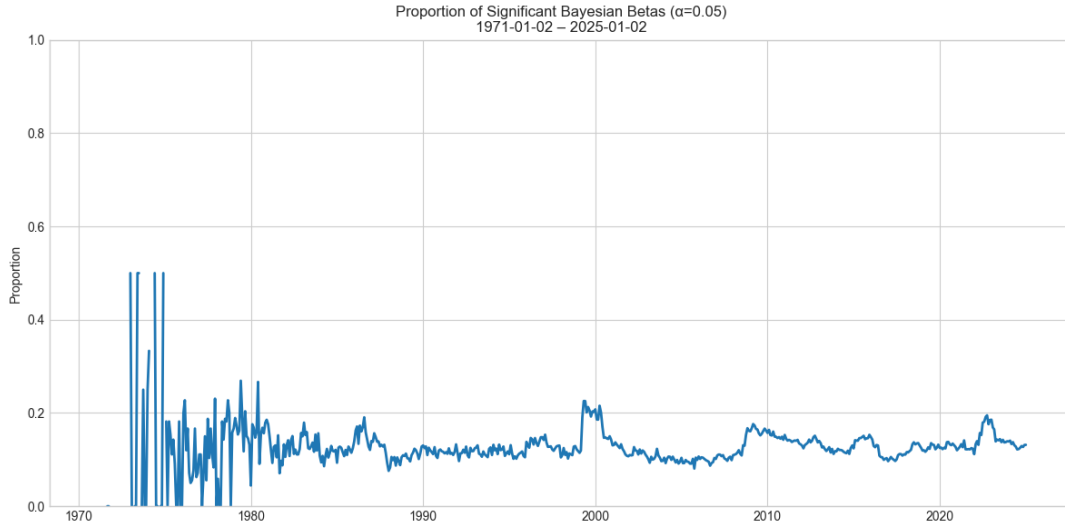


Figure 4: Proportion of significant ( $\alpha=0.05$ , using Newey–West standard errors correction) Bayesian betas overtime.

Figure 4 reports the proportion of stocks with statistically significant inflation betas at the 5% level, using Newey–West corrected standard errors. The proportion fluctuates over time and increases during certain macroeconomic episodes, suggesting that inflation exposure becomes more economically relevant in specific regimes rather than uniformly across the sample.

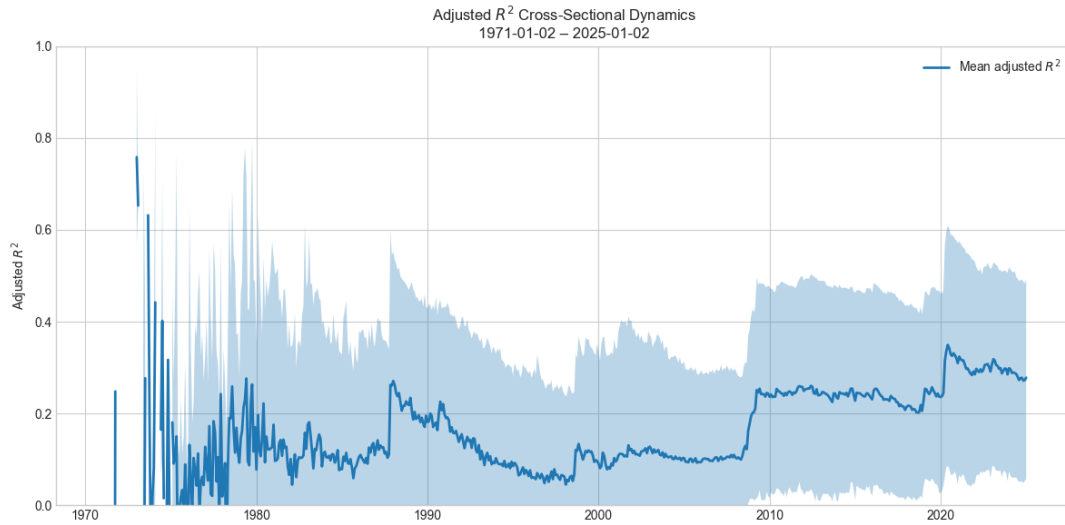


Figure 5: Rsquared (average, 10 and 90 percentiles) overtime from the bivariate linear regression

Figure 5 displays the explanatory power of the bivariate regression model over time. While average R-squared values remain modest, the upper tail indicates that for a subset of firms, inflation innovations contribute meaningfully to return variation.

| Adjusted R-squared |       |
|--------------------|-------|
| Mean               | 0.17  |
| Median             | 0.15  |
| Std Dev            | 0.22  |
| Min                | -2.00 |
| Max                | 1.00  |

Figure 6: Summary statistics of Rsquared from the bivariate linear regression

Consistent with the time-series evidence, Figure 6 confirms that the regression model explains a limited but non-negligible fraction of return variation. The results align with the view that macroeconomic variables are not dominant drivers of returns but can still be exploited in cross-sectional portfolio construction.

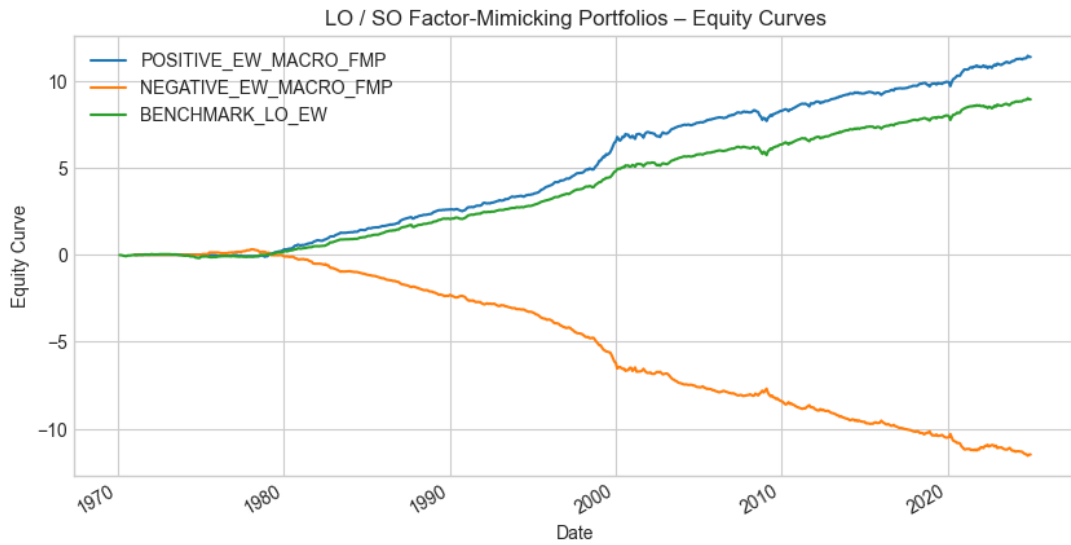


Figure 7: Equity Curve of inflation-FMP long and short leg.

Figure 7 plots the equity curve (cumulative sum of the returns) of the long and short legs of the inflation factor-mimicking portfolio.

|                  | Annualized Return | Annualized Volatility | Sharpe Ratio | Max Drawdown |
|------------------|-------------------|-----------------------|--------------|--------------|
| <b>LO FMP</b>    | 0.21              | 0.17                  | 1.23         | -0.49        |
| <b>SO FMP</b>    | 0.22              | 0.17                  | 1.31         | -0.37        |
| <b>Benchmark</b> | 0.17              | 0.13                  | 1.26         | -0.41        |

Figure 8: Performance metrics of inflation-FMP and Benchmark (EW).

### 3 Inflation Forecasting

This section details our approach regarding inflation forecasting using machine learning methods.

#### 3.1 Data

Our analysis uses the 2024-12 FRED-MD dataset, which is a large macroeconomic database compiled by the Federal Reserve Bank of St. Louis and designed for empirical macroeconomic and financial research. It contains over 125 monthly U.S. macroeconomic time series starting from January 1959, and

integrates transformation codes to ensure stationarity and comparability across series. The dataset is carefully curated, with standardized transformations applied to ensure stationarity and comparability across series.

## 3.2 Methodology

Building on Naghi et. al. (2024), we consider and compare several types of machine learning models (OLS, Lasso, Ridge, Elastic Net, Random Forest, Gradient Boosting, SVR, Feed-forward Neural Network, LightGBM and XGBoost) with different hyperparameter configurations (hyperparameters grid in the appendix). The following steps detail the different steps used in the ML pipeline.

### 3.2.1 Preprocessing

In order to build usable inputs, each variable is transformed according to its predefined transformation code (levels, differences, log-differences, or growth rates) to ensure appropriate scaling and stationarity. Observations (date) with missing values are removed, and the data are finally split into predictors and targets, where the target macroeconomic variable is the 1-month ahead CPIAUCSL ( $h = 1$  forecasting horizon).

### 3.2.2 Regularization

To ensure frugal models and mitigate risks of overfitting, we implemented a regularization procedure using PCA paired with a walk-forward cross-validation framework for optimal out-of-sample (OOS) hyperparameter tuning. Additionally, even though time-constraints prevented us from using them, we also implemented features selectors from the Nowcasting Toolbox, namely a Lasso, Lars, a T-Stat and a correlation selector.

### 3.2.3 Factor extraction

Additionally, we implemented two factor extracting methods to further reduce model size. The first one is a Principal Component Analysis method, the second one is a Dynamic Factor Model (which can also be used to forecast). However, due to time running short, we used only the PCA approach (as DFMs are time-intensive). Hence, we will observe the results for 20 models: 10 using the 125 variables as inputs, and the same 10 models but using 12 principal components (PCs) extracted with PCA. Although choosing the optimal number of PCs can be seen as an additional hyperparameter, we considered here 12 PCs as we have a 12-month validation window. We leave the number of PCs to enter the hyperparameter tuning validation for further research.

### 3.2.4 Estimated models

We considered the following models :

Table 1: Forecasting Models by Methodological Category

| Model Category              | Models  |
|-----------------------------|---|
| Linear Models               | Ordinary Least Squares (OLS)                        |
| Regularized Linear Models   | Ridge Regression, Lasso Regression, Elastic Net     |
| Tree-Based Ensemble Methods | Random Forest, Gradient Boosting, XGBoost, LightGBM |
| Kernel-Based Methods        | Support Vector Regression (SVR)                     |
| Neural Network Models       | Feedforward Neural Network (MLP)                    |

### 3.2.5 Expanding walk-forward cross-validation framework

We implement an expanding window walk-forward framework to evaluate the out-of-sample forecasting performance of the models. At each point in time, models are trained on an expanding sample that starts from an initial minimum number of observations (252) and grows as new data become available. For each re-estimation date, the available training sample is split into a training and a validation subsample, which is used to perform a grid search over hyperparameters that are specific to each model. Hyperparameters selection is based on average validation (RMSE) performance over the validation window. Once the

Figure 1: Training and validation of return prediction model

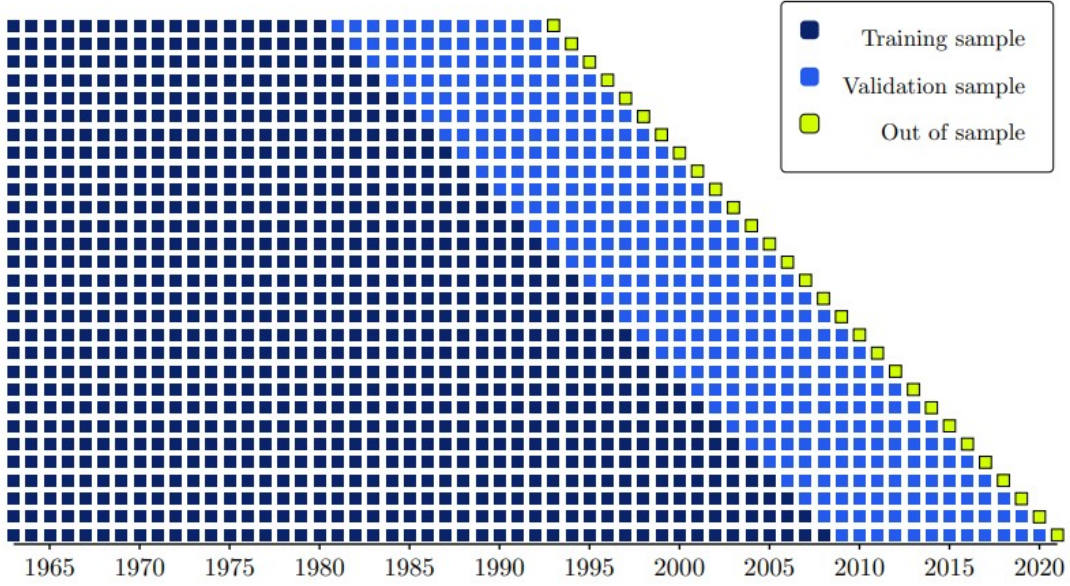


Figure 9: Expanding walk-forward framework

optimal hyperparameters are identified, each model is re-estimated on the full sample up to the end of the validation period and used to generate a one-step-ahead out-of-sample forecast. This walk-forward procedure is repeated over time, ensuring that forecasts are strictly out-of-sample and actualization of model parameters and hyperparameters. See Figure 9 for a visual representation of our training scheme.

### 3.2.6 Results

The results for numeric hyperparameter tuning are presented in Table 11. The graphs displays a fluctuating settings for every hyperparameters, suggesting each time period evaluation came with specific variable relations, pressing the need for model adaptation. This behavior might also be favored by our considered hyperparameters that were also limited in quantity and spaced out (due to computation reasons). Overall, this finding motivates the use of adaptive walk-forward estimation rather than fixed-parameter models.

With the selected hyperparameters, our models performances are summarized in Table 12 for the RMSE over time and in Table 13 for the mean baseline RMSE. The results point out that, while relatively close in absolute differences, some models outperform (but to assess if the difference is statistically significant, a Diebold-Mariano test should be used) the other ones both overtime and in baseline mean RMSE, namely the Random Forest (both with and without a PCA factor extractor implementation), and severall other tree-based approaches (XGBoost and LightGBM). Penalized linear approaches (Lasso and PCA Lasso) also perform well, achieving predicting power thanks to regularization (compared to OLS). Similarly, the Elastic-Net approach (both with and without PCA) displays better predictive power than the lot of our benchmarked models. However, Table 14 reports the proportion of correct sign predictions by our considered models. Several models achieve sign accuracy above 50%, but not the models with the smallest RMSE. This apparent contradiction can arise when a model produces forecasts that are very close to zero and minimize squared errors on average, while occasionally predicting the wrong direction of change. In such cases, small but systematically mis-signed forecasts are penalized only mildly by the RMSE, yet they count fully as errors in terms of directional accuracy. By contrast, models with larger forecast variance may correctly capture the direction of movements more often but incur higher squared errors when the magnitude of deviations is large. As a result, minimizing RMSE may favor conservative, low-volatility forecasts at the expense of correctly predicting the sign of the underlying variable.



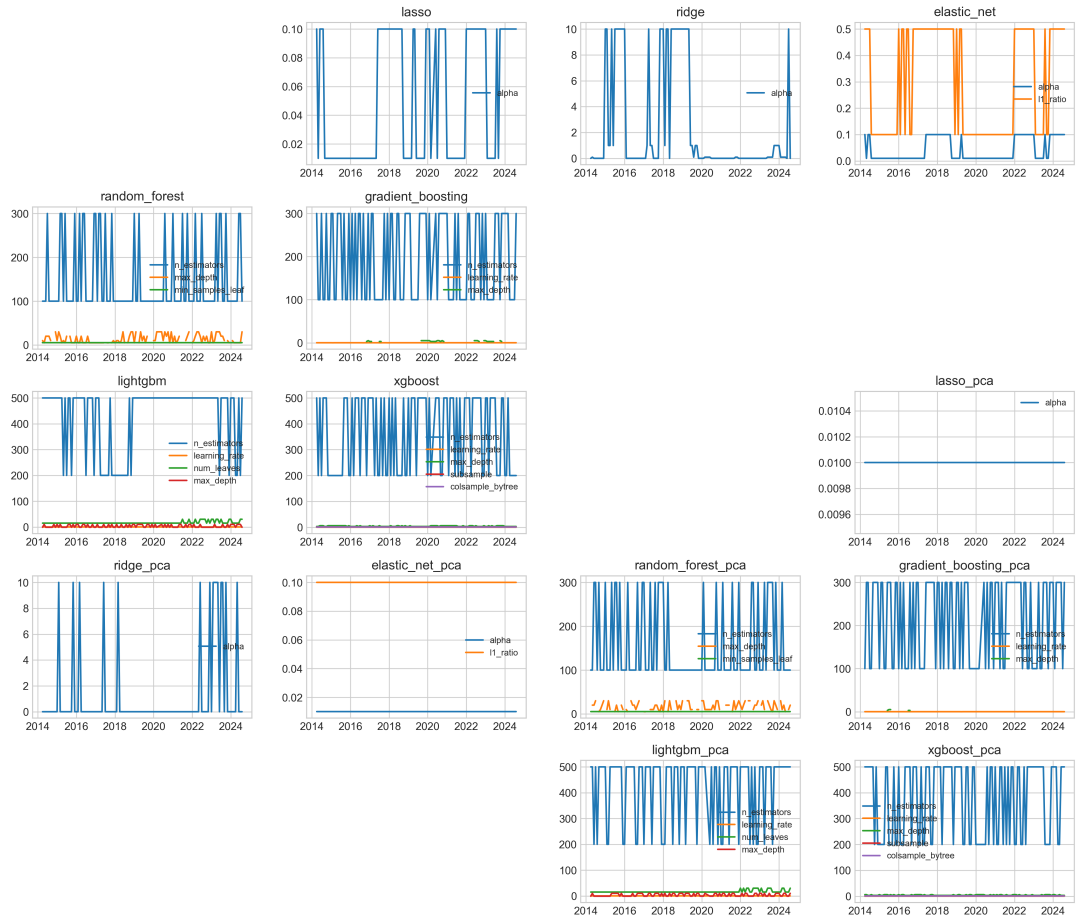


Figure 10: Time-varying optimal hyperparameters (numeric only) for each model

|                        | rank | ols   | lasso | ridge | elastic_net | mean_models |
|------------------------|------|-------|-------|-------|-------------|-------------|
| <b>SRVPRD</b>          | 1    | 3.50  | 0.0   | 0.00  | 0.0         | 0.88        |
| <b>DMANEMP</b>         | 2    | 2.67  | 0.0   | 0.01  | 0.0         | 0.67        |
| <b>NDMANEMP</b>        | 3    | 1.53  | 0.0   | 0.00  | 0.0         | 0.38        |
| <b>PCEPI</b>           | 4    | 1.16  | 0.0   | -0.01 | 0.0         | 0.29        |
| <b>IPMANSICS</b>       | 5    | 0.51  | 0.0   | -0.00 | 0.0         | 0.13        |
| <b>IPMAT</b>           | 6    | 0.45  | 0.0   | 0.01  | 0.0         | 0.11        |
| <b>CONSPI</b>          | 7    | 0.39  | 0.0   | 0.00  | 0.0         | 0.10        |
| <b>M2REAL</b>          | 8    | 0.34  | 0.0   | 0.04  | 0.0         | 0.09        |
| <b>CUSR0000SAS</b>     | 9    | 0.38  | 0.0   | -0.01 | 0.0         | 0.09        |
| <b>CE16OV</b>          | 10   | 0.28  | 0.0   | -0.00 | 0.0         | 0.07        |
| <b>CUSR0000SAC</b>     | 11   | 0.20  | 0.0   | -0.00 | 0.0         | 0.05        |
| <b>RPI</b>             | 12   | 0.17  | 0.0   | 0.01  | 0.0         | 0.04        |
| ...                    | ...  | ...   | ...   | ...   | ...         | ...         |
| <b>W875RX1</b>         | 115  | -0.15 | 0.0   | -0.01 | 0.0         | -0.04       |
| <b>IPDMAT</b>          | 116  | -0.17 | 0.0   | -0.01 | 0.0         | -0.04       |
| <b>CUSR0000SA0L2</b>   | 117  | -0.22 | 0.0   | -0.01 | 0.0         | -0.06       |
| <b>DNDGRG3M086SBEA</b> | 118  | -0.24 | 0.0   | -0.01 | 0.0         | -0.06       |
| <b>USTPU</b>           | 119  | -0.26 | 0.0   | 0.00  | 0.0         | -0.06       |
| <b>CPIULFSL</b>        | 120  | -0.27 | 0.0   | -0.01 | 0.0         | -0.07       |
| <b>CLF16OV</b>         | 121  | -0.34 | 0.0   | -0.00 | 0.0         | -0.09       |
| <b>CUSR0000SA0L5</b>   | 122  | -0.41 | 0.0   | -0.01 | 0.0         | -0.10       |
| <b>INDPRO</b>          | 123  | -0.86 | 0.0   | 0.00  | 0.0         | -0.21       |
| <b>DSERRG3M086SBEA</b> | 124  | -0.89 | 0.0   | -0.01 | 0.0         | -0.22       |
| <b>MANEMP</b>          | 125  | -3.63 | 0.0   | 0.00  | 0.0         | -0.91       |
| <b>PAYEMS</b>          | 126  | -3.72 | 0.0   | 0.00  | 0.0         | -0.93       |

Figure 11: Time-series average coefficients for non-PCA linear models

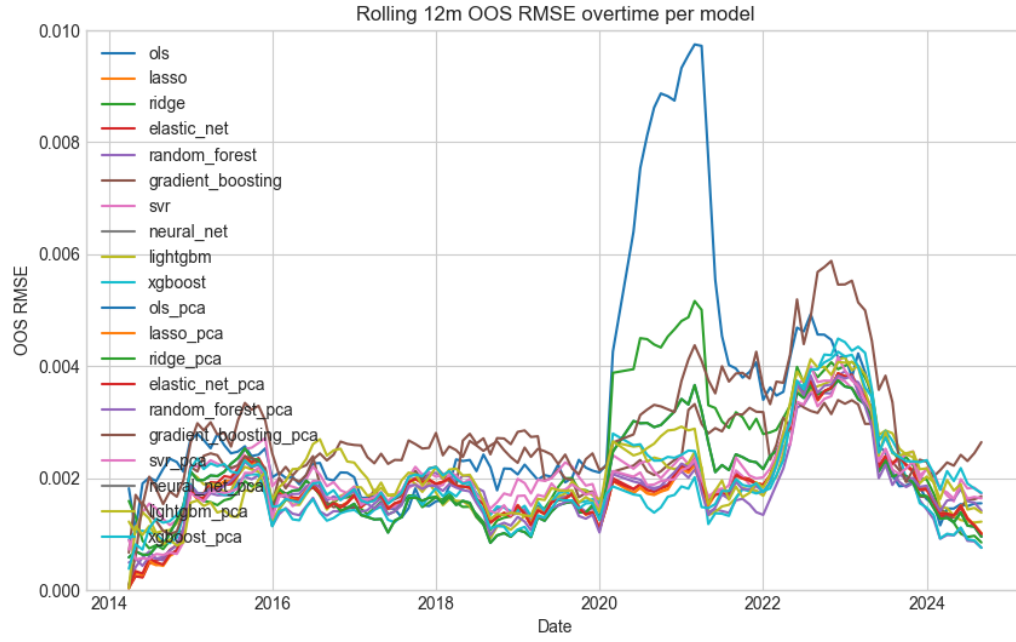


Figure 12: 12-months rolling OOS RMSE per model

|                              | rank | mean_rmse |
|------------------------------|------|-----------|
| <b>random_forest</b>         | 1    | 0.001765  |
| <b>random_forest_pca</b>     | 2    | 0.001868  |
| <b>lasso_pca</b>             | 3    | 0.00187   |
| <b>elastic_net_pca</b>       | 4    | 0.001874  |
| <b>lasso</b>                 | 5    | 0.001876  |
| <b>xgboost</b>               | 6    | 0.00189   |
| <b>elastic_net</b>           | 7    | 0.00189   |
| <b>lightgbm</b>              | 8    | 0.001955  |
| <b>ols_pca</b>               | 9    | 0.001965  |
| <b>ridge_pca</b>             | 10   | 0.001965  |
| <b>xgboost_pca</b>           | 11   | 0.002005  |
| <b>svr_pca</b>               | 12   | 0.002011  |
| <b>svr</b>                   | 13   | 0.00211   |
| <b>lightgbm_pca</b>          | 14   | 0.002122  |
| <b>ridge</b>                 | 15   | 0.002227  |
| <b>gradient_boosting</b>     | 16   | 0.002347  |
| <b>gradient_boosting_pca</b> | 17   | 0.002891  |
| <b>ols</b>                   | 18   | 0.003083  |
| <b>neural_net_pca</b>        | 19   | 0.221804  |
| <b>neural_net</b>            | 20   | 0.685397  |

Figure 13: Time-series average of OOS RMSE per model

|                              | rank | sign_accuracy |
|------------------------------|------|---------------|
| <b>xgboost</b>               | 1    | 0.62          |
| <b>ols</b>                   | 2    | 0.61          |
| <b>lightgbm</b>              | 3    | 0.61          |
| <b>random_forest</b>         | 4    | 0.60          |
| <b>ridge</b>                 | 5    | 0.55          |
| <b>gradient_boosting</b>     | 6    | 0.54          |
| <b>xgboost_pca</b>           | 7    | 0.54          |
| <b>neural_net_pca</b>        | 8    | 0.54          |
| <b>lightgbm_pca</b>          | 9    | 0.54          |
| <b>random_forest_pca</b>     | 10   | 0.53          |
| <b>ridge_pca</b>             | 11   | 0.51          |
| <b>svr_pca</b>               | 12   | 0.51          |
| <b>svr</b>                   | 13   | 0.51          |
| <b>ols_pca</b>               | 14   | 0.51          |
| <b>neural_net</b>            | 15   | 0.50          |
| <b>elastic_net_pca</b>       | 16   | 0.48          |
| <b>elastic_net</b>           | 17   | 0.44          |
| <b>lasso</b>                 | 18   | 0.43          |
| <b>lasso_pca</b>             | 19   | 0.43          |
| <b>gradient_boosting_pca</b> | 20   | 0.42          |

Figure 14: Time-series average of OOS sign accuracy per model

## 4 Dynamic Allocation FMP

### 4.1 Methodology

The different dynamic allocation strategies consist of going long the high-betas inflation-FMP leg if the forecast produced by the model is of positive sign, going long the low-betas inflation-FMP leg if the forecast is negative and going long the benchmark (Long-Only Equally-Weighted) when the forecast is 0.0.

### 4.2 Results

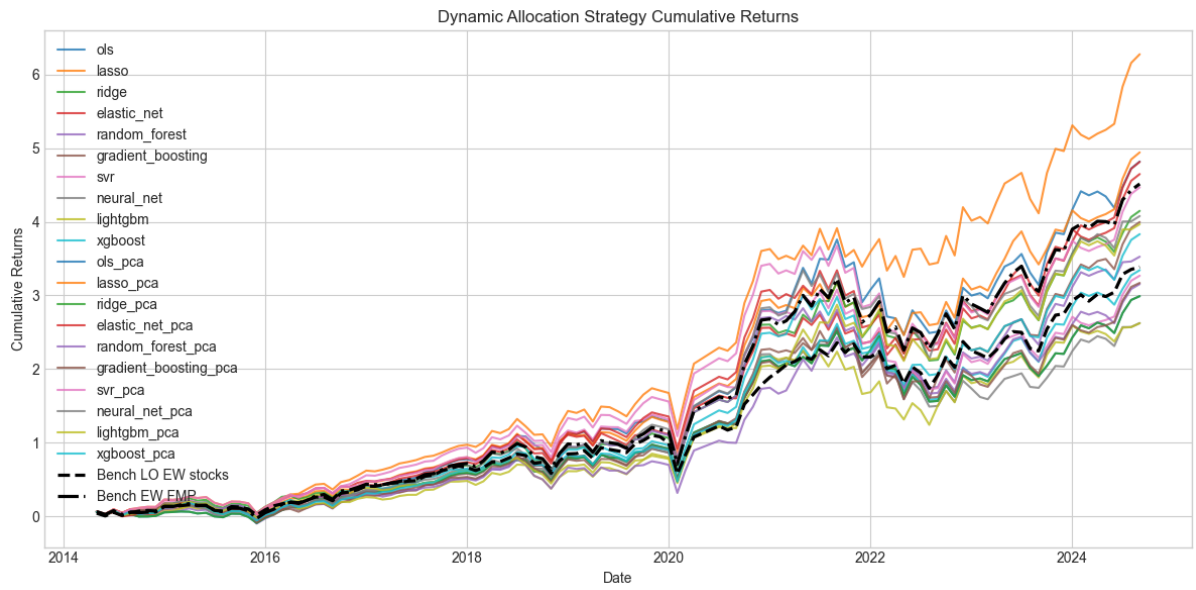


Figure 15: Cumulative Returns of dynamic allocation strategies

Figure 15 shows the cumulative performance of dynamic allocation strategies driven by inflation forecasts. Several strategies outperform static benchmarks, demonstrating that macro forecasts combined with factor-mimicking portfolios can be translated into economically valuable allocation rules.

|                       | Rank | Ann. Return | Ann. Vol | Sharpe | Max Drawdown |
|-----------------------|------|-------------|----------|--------|--------------|
| lasso_pca             | 1    | 21.36       | 19.56    | 1.09   | -20.31       |
| lasso                 | 2    | 18.99       | 19.15    | 0.99   | -20.79       |
| elastic_net           | 3    | 18.40       | 19.51    | 0.94   | -21.05       |
| elastic_net_pca       | 4    | 18.74       | 20.12    | 0.93   | -26.04       |
| Bench EW FMP          | 5    | 18.12       | 19.63    | 0.92   | -22.93       |
| ols                   | 6    | 18.74       | 20.71    | 0.91   | -27.16       |
| Bench LO EW stocks    | 7    | 15.51       | 16.99    | 0.91   | -24.98       |
| svr_pca               | 8    | 18.03       | 20.22    | 0.89   | -29.46       |
| ridge                 | 9    | 17.34       | 20.19    | 0.86   | -25.58       |
| neural_net_pca        | 10   | 17.18       | 19.95    | 0.86   | -27.79       |
| lightgbm_pca          | 11   | 16.93       | 19.74    | 0.86   | -20.31       |
| gradient_boosting     | 12   | 17.00       | 19.98    | 0.85   | -26.24       |
| xgboost_pca           | 13   | 16.61       | 19.98    | 0.83   | -24.40       |
| random_forest         | 14   | 15.87       | 20.24    | 0.78   | -28.78       |
| xgboost               | 15   | 15.41       | 20.77    | 0.74   | -35.43       |
| svr                   | 16   | 15.21       | 20.69    | 0.74   | -33.15       |
| random_forest_pca     | 17   | 14.91       | 20.21    | 0.74   | -25.76       |
| ols_pca               | 18   | 14.46       | 19.91    | 0.73   | -27.19       |
| ridge_pca             | 19   | 14.46       | 19.91    | 0.73   | -27.19       |
| gradient_boosting_pca | 20   | 14.95       | 20.75    | 0.72   | -27.50       |
| neural_net            | 21   | 13.39       | 19.91    | 0.67   | -29.96       |
| lightgbm              | 22   | 13.38       | 20.41    | 0.66   | -31.01       |

Figure 16: Performance metrics of the dynamic allocation strategies

The performance table of Figure 16 highlights improvements in risk-adjusted returns for selected dynamic strategies relative to benchmarks. Gains are driven not only by higher returns but also by improved drawdown control, confirming the diversification and timing benefits of the approach.

## 5 Conclusion

This paper examines whether macroeconomic information can be effectively incorporated into investment strategies through macro factor-mimicking portfolios and predictive modeling. Focusing on inflation, we construct tradable factor-mimicking portfolios based on stock-level macro sensitivities and use machine learning methods to forecast inflation dynamics from a large macroeconomic information set.

Our results show that inflation exposure varies substantially across assets and over time, validating the construction of long-short factor-mimicking portfolios. Forecasting performance varies across models, with nonlinear and tree-based methods outperforming linear benchmarks in out-of-sample RMSE, whilst being less proficient in sign accuracy.

When embedded into a dynamic allocation framework, inflation forecasts generate strategies that outperform static benchmarks on both return and risk-adjusted bases. The results suggest that macro factor-mimicking portfolios provide an effective bridge between macroeconomic forecasting and portfolio management.

Overall, this study provides empirical evidence that combining macroeconomic forecasts with tradable macro proxies can enhance dynamic asset allocation. Future research could extend this framework to additional macroeconomic variables, alternative forecasting horizons or multi-factor allocation schemes.

## References

- [1] Esakia, M., & Goltz, F. (2023). Targeting Macroeconomic Exposures in Equity Portfolios: A Firm-Level Measurement Approach for Out-of-Sample Robustness. *Financial Analysts Journal*. <https://doi.org/10.1080/0015198X.2022.2150500>
- [2] Vasicek, O. A. 1973. "A Note on Using Cross-Sectional Information in Bayesian Estimation of Security Betas." *The Journal of Finance*. doi:10.1111/j.1540-6261.1973.tb01452.x.
- [3] Naghi, Andrea A.; O'Neill, Eoghan; Zaharieva, Martina D. (2023): The benefits of forecasting inflation with machine learning: New evidence. *Journal of Applied Econometrics*. <https://doi.org/10.15456/jae.2023340.1218742483>

## 6 Appendix

### 6.1 Hyperparameter Grids

Table 2 reports the hyperparameter search grids used for each forecasting model in the expanding walk-forward cross-validation procedure.

Table 2: Hyperparameter grids for all machine learning models

| Model                              | Hyperparameter grid   |
|------------------------------------|---|
| OLS                                | No hyperparameters  |
| Lasso                              | $\alpha \in \{0.01, 0.1, 1.0, 10.0\}$   |
| Ridge                              | $\alpha \in \{0.01, 0.1, 1.0, 10.0\}$   |
| Elastic Net                        | $\alpha \in \{0.01, 0.1, 1.0, 10.0\};$<br>$l1\_ratio \in \{0.1, 0.5, 0.9\}$   |
| Random Forest                      | $n\_estimators \in \{100, 300\};$<br>$max\_depth \in \{\text{None}, 5, 10, 20, 30\};$<br>$min\_samples\_leaf = 5$   |
| Gradient Boosting                  | $n\_estimators \in \{100, 300\};$<br>$learning\_rate \in \{0.01, 0.1\};$<br>$max\_depth \in \{\text{None}, 3, 5\}$  |
| Support Vector<br>Regression (SVR) | $C \in \{0.1, 1.0, 10.0\};$<br>$\epsilon \in \{0.01, 0.1\};$<br>$kernel = \text{rbf}$   |
| Neural Network<br>(MLP)            | $hidden\_layer\_sizes \in \{(16), (32), (32, 16)\};$<br>$\alpha \in \{10^{-4}, 10^{-3}\};$<br>$learning\_rate\_init = 0.001$  |
| LightGBM                           | $n\_estimators \in \{200, 500\};$<br>$learning\_rate \in \{0.01, 0.1\};$<br>$num\_leaves \in \{15, 30\};$<br>$max\_depth \in \{-1, 10\}$  |
| XGBoost                            | $n\_estimators \in \{200, 500\};$<br>$learning\_rate \in \{0.01, 0.1\};$<br>$max\_depth \in \{3, 6\};$<br>$subsample \in \{0.8, 1.0\};$<br>$colsample\_bytree \in \{0.8, 1.0\}$ |