New Methods for Mixed-Frequency Modeling Using Machine Learning: With An Application to Forecasting Stock Returns*

Weijia Peng[a], Norman R. Swanson[b], Chun Yao[c]
[a]Sacred Heart University, [b]Rutgers University, [c]Centiva Capital

July 10, 2026

Abstract

This paper proposes and studies three new machine learning methods for incorporating mixed-frequency information into the prediction of lower-frequency time series variables in high-dimensional predictor environments. The methods are designed to preserve within-period dynamics across multiple data frequencies and utilize path embedding and encoding, dimension reduction and shrinkage, and various types of data aggregation. We use monthly stock returns prediction to illustrate implementation of the methods, with a dataset that contains more than 100 daily financial and macroeconomic predictors and a common set of machine learning methods including LASSO and ridge shrinkage operators, random forests, support vector machines, feedforward neural networks, and long-short-term memory networks. Our findings indicate that incorporating higher-frequency information significantly improves forecasting performance and that predictive accuracy depends critically on how such information is incorporated into the forecasting process. Among the proposed methods, one called Group-Structured High-Frequency Path Embedding delivers the most consistent improvements across assets, forecasting models, and evaluation periods. This method also performs favorably when compared against extant mixed-frequency modeling methods such as MIDAS. Overall, the findings highlight the importance of effectively utilizing higher-frequency information in mixed-frequency forecasting and suggest broad applicability to financial and macroeconomic forecasting problems involving predictors and targets observed at different frequencies.

Keywords: Mixed-Frequency Forecasting; High-frequency Data; Machine Learning; Information Representation.

JEL Classification: C22, C52, C53, C58.

Corresponding to: Weijia Peng, Department of Finance, Jack Welch College of Business & Technology, Sacred Heart University, 5151 Park Avenue Fairfield, CT 06825, USA, pengw@sacredheart.edu. Norman R. Swanson, Department of Economics, Rutgers University, 75 Hamilton Street, New Brunswick, NJ 08901, USA, nswanson@rutgers.edu; Chun Yao, Centival Capital, 66 Hudson Blvd E, New York, NY 10001, USA, chunyao.economics@gmail.com. Swanson thanks Rutgers University for research support.

1 Introduction

Forecasting time series variables remains a central problem in finance, economics, and data science. In many applications, decision makers seek to predict outcomes observed at relatively low frequencies, such as monthly asset returns, quarterly macroeconomic indicators, or annual firm performance measures, while relevant predictive information arrives at higher frequencies. In this paper, we introduce three methods for incorporating high-frequency data into lower-frequency prediction problems. The methods utilize path embedding and encoding, dimension reduction and shrinkage, and various types of data aggregation. A key feature of the methods involves the construction and use of mixed-frequency forecasts based on machine learning approaches including ones that employ LASSO and ridge shrinkage operators, random forests, support vector machines, feedforward neural networks, and long-short-term memory (LSTM) networks (see e.g. Jordan and Mitchell (2015), Mullainathan and Spiess (2017), Feng et al. (2020); Gu et al. (2020), Sirignano and Cont (2021), Rolnick et al. (2022), Goulet Coulombe et al. (2022), and Babii et al. (2022)).

All of the above machine learning approaches are designed to exploit large amounts of information and capture nonlinear relationships and complex interactions that can be difficult to model using traditional econometric specifications. At the same time, a growing mixed-frequency forecasting literature has developed methods for linking predictors and targets observed at different sampling frequencies. Important among these are so-called MIDAS regressions (see Ghysels et al. (2004, 2007, 2006), Ghysels (2016), Andreou et al. (2013), Foroni et al. (2015), Babii et al. (2022), and Marcellino and Schumacher (2010)), mixed-frequency state-space models (see Mariano and Murasawa (2003), Giannone et al. (2008), and Schorfheide and Song (2015)), and dynamic factor models (see Stock and Watson (2002), Banbura et al. (2011), and Bok et al. (2018)). Many of the methodologies discussed in these papers has proven effective in a variety forecasting applications, often because of the effective way in which higher-frequency observations are aggregated into lower-frequency forecasts. This paper adds to the corpus of research on the related topic of how to construct informative representations of higher-frequency information in high-dimensional forecasting environments that can subsequently be incorporated into lower frequency forecasts while remaining agnostic regarding weighting functions, lag-decay structures, and/or state-transition processes.

More specifically, we introduce three methods for incorporating high-frequency data into lower-frequency prediction problems. These include: High-Frequency Path Embedding (HPE), Group-Structured High-Frequency Path Embedding with Autoencoding (GHPA), and Diagonal Multi-Horizon Temporal Aggregation (DMHTA). HPE constructs monthly forecasting inputs by collecting the most recent 22 daily observations of all predictors and flattening this daily path into a high-dimensional feature vector. GHPA builds on this path-embedding idea by first using LASSO screening to select relevant predictors and then applying an autoencoder to compress the resulting 22-day predictor paths into a low-dimensional latent representation. Unlike HPE and GHPA, which construct monthly forecasting features from daily predictor paths, DMHTA operates directly in the forecasting stage by generating horizon-specific forecasts and combining overlapping predictions to produce a monthly forecast. DMHTA is related to the forecast

combination and temporal hierarchy literature (see Bates and Granger (1969), Timmermann (2006), and Athanasopoulos et al. (2017)). However, unlike these approaches, DMHTA combines horizon-specific forecasts generated within a single cross-frequency forecasting model rather than combining forecasts generated by alternative models or different temporal aggregation levels. Together, these methods provide alternative ways of utilizing high-frequency information for lower-frequency prediction while balancing information preservation, dimensionality reduction, and computational scalability.

A key distinction between our approach and traditional mixed-frequency methods lies in how higher-frequency information is incorporated into lower-frequency forecasting problems. Existing approaches such as MIDAS regressions, state-space models, and dynamic factor models aggregate higher-frequency information through parametric lag-weighting functions, latent factors, or state-transition structures. By contrast, our method does not impose a particular temporal weighting function, lag-decay structure, or state-transition process. Instead, higher-frequency observations are transformed into structured representations that preserve within-period dynamics, after which the forecasting model learns which aspects of those representations are most predictive of the lower-frequency target. This shifts the focus of mixed-frequency forecasting from estimating aggregation mechanisms to constructing informative cross-frequency feature representations. Such an approach provides greater flexibility and allows predictive information to arise from potentially complex patterns in the high-frequency data without being affected by pre-specified aggregation schemes.

More broadly, our method is related to the growing representation-learning literature in machine learning which seeks to learn informative representations of time series for downstream tasks such as classification, anomaly detection, and forecasting (see Bengio et al. (2013), Goodfellow et al. (2016), Lim et al. (2021), Franceschi et al. (2019), Eldele et al. (2021), and Yue et al. (2022)). These methods generally focus on learning representations from single-frequency time series. By contrast, our methods are designed specifically for forecasting problems involving predictors and targets observed at different frequencies, where preserving within-period dynamics is central to predictive performance.

In order to assess the potential usefulness of our proposed methods, we consider stock return forecasting, which provides a particularly useful setting for studying cross-frequency prediction because financial markets generate vast amounts of information at daily and intraday frequencies, while forecasting targets are often evaluated at monthly horizons. A large literature has examined return predictability using financial, macroeconomic, and technical indicators, and it has been found that forecasting performance remains sensitive to model specification, predictor selection, and information design (see Gu et al. (2021), Gu et al. (2020), Welch and Goyal (2008), Campbell and Thompson (2008), and Timmermann (2008)). These characteristics make stock returns a natural environment for studying how higher-frequency information can be incorporated into lower-frequency forecasting problems.

Summarizing, in a series of stock return forecasting experiments, we evaluate the proposed methods against a conventional monthly-frequency forecasting benchmark using more than 100 predictor variables spanning technical indicators, foreign exchange rates, commodity prices, equity indices, macroeconomic, and financial variables. Forecasts are generated for monthly re-

turns of a broad market ETF and representative firms from the technology, financial, energy, and healthcare sectors. To isolate the effects of information representation, a common set of forecasting base learners, including tree methods, support vector machines, penalized regressions, feedforward neural networks, and LSTM networks are employed across all methods. Forecasting performance is evaluated using mean squared forecast errors, Diebold–Mariano tests, and Model Confidence Set procedures. Our empirical results yield four main findings. First, incorporating higher-frequency information substantially improves forecasting performance relative to conventional monthly-frequency predictors. Second, the representation of higher-frequency information is a critical determinant of forecasting success. Across assets and forecasting models, forecasting performance depends not only on the availability of higher-frequency data, but also on how that information is transformed into lower-frequency forecasting features. Third, GHPA delivers the most robust and consistent forecasting gains across assets, forecasting models, and evaluation periods, suggesting that combining information preservation with dimensionality reduction provides an effective approach for cross-frequency forecasting. Finally, comparison with a Factor-MIDAS benchmark indicates that our methods yield superior forecasts in some scenarios, particularly in the full sample and post-COVID periods that we examine. This finding highlights the value of preserving richer within-period information and allowing machine learning models to learn complex high-frequency representations. More broadly speaking, the above findings highlight the idea that information representation is an important dimension of mixed-frequency forecasting, complementing existing approaches that primarily focus on the estimation of aggregation mechanisms. Additionally, while the empirical application focuses on monthly stock return prediction, the proposed methods are broadly applicable to forecasting problems in finance, macroeconomics, energy markets, climate risk analysis, and other domains involving predictors and targets observed at different frequencies.

The remainder of the paper is organized as follows. Section 2 discusses our so-called "base learner" forecasting models used in conjunction with our mixed-frequency forecasting *methods*, and then turns to a discussion of theoretical motivation and implmentation steps for the 3 *methods*, which include HPE, GHPA, and DMHTA. Section 3 describes the data, forecasting targets, and predictor variables used on our empirical anaysis. Section 4 presents the forecasting design features, model estimation procedures, hyperparameter selection methods, and evaluation metrics that round out our empirical setup. Section 5 reports empirical findings. Sections 6 - 7 contain additional empirical findings associated with various robustness checks on our findings. Finally, Section 8 concludes and discusses directions for future research.

2 Methodology

This section introduces the proposed mixed-frequency forecasting methods. The section is organized as follows. In the first sub-section, we discuss the general forecasting setup. In the second sub-section, we summarize various forecasting *models* that are used in the final step in our forecasting methods. It should be noted that other *models* can also be used. The ones that we list are selected to correspond with those that we subsequently utilize in our empirical experiments. In the third sub-section, we detail our three proposed mixed-frequency forecasting

methods. These three methods for utilizing higher-frequency information in lower-frequency prediction tasks include: HPE, GHPA, and DMHTA.

While the three methods are designed for general cross-frequency forecasting settings, monthly stock return prediction serves as the main empirical application in this study, and our notation is conformable with our empirical analysis, in which monthly returns are predicted using daily data.

2.1 General Forecasting Setup

Following Gu et al. (2020), let y_{m+1} denote a lower-frequency target variable and let $\mathcal{F}_m$ denote the information set available at the forecasting origin. The forecasting problem can be expressed as $y_{m+1} = E_m(y_{m+1}) + \varepsilon_{m+1}$, where $E_m(y_{m+1}) = g(\mathcal{F}_m)$, and $g(\cdot)$ denotes the unknown conditional mean function mapping the available information set into the expected value of the future target variable. In mixed-frequency forecasting problems, the information set $\mathcal{F}_m$ typically contains variables observed at a higher frequency than the target variable. The key challenge is therefore how to transform or aggregate higher-frequency information into forecasting features that can be used to approximate $g(\cdot)$. The objective of forecasting is to construct an approximation to $g(\cdot)$ using observed predictors and a forecasting model.

For notational simplicity, let $\mathbf{x}_m$ denote the predictor vector used by a given forecasting model to approximate the conditional mean function $g(\cdot)$. The following models are used to construct forecasts of y_{m+1}. Note that the first two models are simple econometric benchmarks, while the latter models are based on machine learning methods. In practical applications, forecasting involves a two step procedure. In the first step the (high frequency) data are "processed" using one of the three new mixed-frequency methods outlined in the next section. Note that some of these mixed-frequency methods also contain machine learning elements, such as the use of the least absolute shrinkage operator for high frequency variable selection. In the second step, the processed data are used as inputs for one of the forecasting models (also called our forecasting base learners) listed below, which are estimated using standard methods.

2.2 Forecasting Models Used as Inputs in Mixed-Frequency Methods

Random Walk Model: The random walk model serves as one of our benchmarks and assumes that the expected value of the target variable is constant over time. Forecasts are given by the historical mean,

$$\hat{y}_{m+1} = \bar{y}, \tag{1}$$

where $\bar{y}$ is constructed using data prior to period $m+1$.

Autoregressive Model: The autoregressive model serves as another of our benchmarks, and is specified as follows:

$$y_{m+1} = \alpha + \sum_{j=1}^{p} \phi_j y_{m+1-j} + \varepsilon_{m+1}, \tag{2}$$

where the lag order p is selected using either AIC or BIC, and ε_{m+1} is a stochastic disturbance term.

Least Absolute Shrinkage Operator (LASSO): The LASSO can be viewed as a regression problem that imposes an ℓ_1 penalty on the coefficients in a many predictor linear regression model with coefficients estimated by optimizing the following function (see (Tibshirani, 1996)):

$$\min_{\beta} \left\{ \sum_{m} (y_{m+1} - \mathbf{x}_m^\top \beta)^2 + \lambda \|\beta\|_1 \right\}, \tag{3}$$

where $\|\cdot\|_1$ denotes the $\ell_1 - norm$. The penalty encourages sparse solutions and can also be used to perform variable selection.

Ridge Regression: Ridge regression is similar in structure to the LASSO, but employs ℓ_2 regularization to shrink coefficients toward zero. Unlike the LASSO, all estimated coefficients remain nonzero, so that the ridge estimator is not additionally used for variable selection (see Hoerl and Kennard (1970)). Coefficients are estimated by optimizing the following function:

$$\min_{\beta} \left\{ \sum_{m} (y_{m+1} - \mathbf{x}_m^\top \beta)^2 + \lambda \|\beta\|_2^2 \right\}. \tag{4}$$

Random Forest: Random forests approximate the unknown forecasting function using an ensemble of regression trees estimated from bootstrap samples of the training data (see Breiman (2001)). The forecasting model is:

$$\hat{y}_{m+1} = f(\mathbf{x}_m) = \frac{1}{B} \sum_{b=1}^{B} T_b(\mathbf{x}_m), \tag{5}$$

where $\mathbf{x}_m$ denotes the predictor vector, B is the total number of regression trees, and $T_b(\cdot)$ denotes the prediction from the bth regression tree. A regression tree partitions the predictor space into R_b disjoint terminal regions and predicts

$$T_b(\mathbf{x}) = \sum_{r=1}^{R_b} c_{rb} I(\mathbf{x} \in R_{rb}), \tag{6}$$

where R_{rb} denotes the rth terminal region of tree b, c_{rb} is the average response of the training observations within that region, and $I(\cdot)$ is the indicator function, which equals one if $\mathbf{x} \in R_{rb}$ and zero otherwise. Each tree is estimated independently using a bootstrap sample of the training data, while a random subset of predictors is considered at each split to reduce correlation among trees. The final forecast is obtained by averaging predictions across all trees.

Gradient Boosting: Gradient boosting approximates the forecasting function as a weighted sum of sequentially estimated weak learners (see Friedman (2001)). Specifically,

$$\hat{y}_{m+1} = f(\mathbf{x}_m) = \sum_{b=1}^{B} \gamma_b h_b(\mathbf{x}_m), \tag{7}$$

where $h_b(\cdot)$ denotes the prediction from the bth weak learner, γ_b is the corresponding step size, and B is the total number of boosting iterations. In this study, each weak learner is a shallow regression tree of the form given in equation (6). Unlike random forests, however, the trees are estimated sequentially rather than independently. Specifically, at iteration b, the regression tree is fitted to the negative gradient (pseudo-residuals) of the loss function evaluated using the current ensemble, and the updated forecasting function is obtained by adding the weighted prediction $\gamma_b h_b(\mathbf{x}_m)$ to the existing ensemble. Consequently, boosting constructs the forecasting model through sequential refinement, whereas random forests rely on averaging independently estimated trees.

Support Vector Regression: Support vector regression (SVR) constructs forecasts in a high-dimensional feature space induced by a kernel function (see Drucker et al. (1996) and Cortes and Vapnik (1995)). The forecasting model is:

$$\hat{y}_{m+1} = f(\mathbf{x}_m) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) K(\mathbf{x}_i, \mathbf{x}_m) + b, \tag{8}$$

where N denotes the number of training observations, α_i and α_i^* are the dual optimization coefficients, b is the intercept, and $K(\cdot, \cdot)$ denotes the kernel function. In this study we consider both radial basis function (RBF) and polynomial kernels. The RBF kernel is:

$$K(\mathbf{x}_i, \mathbf{x}) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}\|^2\right), \tag{9}$$

where $\gamma > 0$ controls the kernel bandwidth. The polynomial kernel is

$$K(\mathbf{x}_i, \mathbf{x}) = (\mathbf{x}_i' \mathbf{x} + c)^d, \tag{10}$$

where c is a constant and d denotes the polynomial degree.

Feedforward Neural Networks: Following Kuan and White (1994) and Swanson and White (1997), a feedforward neural network approximates the unknown forecasting function by combining a linear component with nonlinear hidden units. A single-hidden-layer network may be written as

$$f(\mathbf{x}_m, \Theta) = \tilde{\mathbf{x}}_m' \boldsymbol{\kappa} + \sum_{j=1}^{q} G\left(\tilde{\mathbf{x}}_m' \boldsymbol{\pi}_j\right) \lambda_j, \tag{11}$$

where $\tilde{\mathbf{x}}_m = (1, \mathbf{x}_m')'$ is the predictor vector augmented with an intercept, $\Theta = (\boldsymbol{\kappa}', \boldsymbol{\lambda}', \boldsymbol{\pi}')'$ collects all model parameters, $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_q)'$, $\boldsymbol{\pi} = (\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_q)'$, q denotes the number of hidden units, and $G(\cdot)$ denotes a nonlinear activation function. The first term represents the

linear component of the forecasting model, while the second term captures nonlinear interactions through the hidden layer. In this study, we employ neural networks with two and three hidden layers, which extend the single-hidden-layer specification above by stacking additional nonlinear transformations.

Long Short-Term Memory Networks: Long Short-Term Memory (LSTM) networks extend feedforward neural networks by explicitly modeling temporal dependence through recurrent hidden states and memory cells (see Hochreiter and Schmidhuber (1997)). Let $\{\mathbf{x}_t\}_{t=1}^{T}$ denote the input sequence. The hidden state is updated according to

$$\mathbf{f}_t = \sigma(W_f \mathbf{x}_t + U_f \mathbf{h}_{t-1} + \mathbf{b}_f), \tag{12}$$

$$\mathbf{i}_t = \sigma(W_i \mathbf{x}_t + U_i \mathbf{h}_{t-1} + \mathbf{b}_i), \tag{13}$$

$$\mathbf{o}_t = \sigma(W_o \mathbf{x}_t + U_o \mathbf{h}_{t-1} + \mathbf{b}_o), \tag{14}$$

$$\tilde{\mathbf{c}}_t = \tanh(W_c \mathbf{x}_t + U_c \mathbf{h}_{t-1} + \mathbf{b}_c), \tag{15}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \tag{16}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \tag{17}$$

where $\mathbf{f}_t$, $\mathbf{i}_t$, and $\mathbf{o}_t$ denote the forget, input, and output gates, respectively, $\mathbf{c}_t$ is the memory cell, $\mathbf{h}_t$ is the hidden state, $\sigma(\cdot)$ denotes the logistic activation function, and $\odot$ denotes element-wise multiplication. The final forecast is obtained from the last hidden state,

$$\hat{y}_{m+1} = W_o \mathbf{h}_T + b_o, \tag{18}$$

where W_o and b_o denote the output-layer parameters.

2.3 Mixed-Frequency Methods for Incorporating High-Frequency Data into Lower Frequency Forecasts

In this sub-section we detail three methods for incorporating high-frequency data into lower-frequency prediction problems. These include: High-Frequency Path Embedding (HPE), Group-Structured High-Frequency Path Embedding with Autoencoding (GHPA), and Diagonal Multi-Horizon Temporal Aggregation (DMHTA). As discussed above, and without loss of generality, we outline the methods in the context of the empirical experiments reported in a subsequent section. Namely, we assume that the objective is to forecast monthly returns using a dataset that includes daily data, and assume that each month contains approximately 22 days of daily returns data. In this context, HPE constructs monthly forecasting inputs by collecting the most recent 22 daily observations of all predictors and flattening this daily path into a high-dimensional feature vector, without the use of machine learning. GHPA builds on this by first using LASSO screening to select relevant predictors for "flattening". Unlike HPE and GHPA, which construct monthly forecasting features from daily predictor paths, DMHTA operates directly in the forecasting stage by generating horizon-specific forecasts and combining overlapping predictions to

produce a monthly forecast. DMHTA is related to the forecast combination and temporal hierarchy literature (see Bates and Granger (1969), Timmermann (2006), and Athanasopoulos et al. (2017)). However, unlike these approaches, DMHTA combines horizon-specific forecasts generated within a single cross-frequency forecasting model rather than combining forecasts generated by alternative models or different temporal aggregation levels.

2.3.1 Method 1: High-Frequency Path Embedding (HPE)

Consider a lower-frequency forecasting target y_{m+1} and a collection of higher-frequency predictors:

$$\{\mathbf{x}_t\}_{t\in\mathbb{Z}}, \qquad \mathbf{x}_t \in \mathbb{R}^K,$$

where m indexes monthly observations and t indexes daily observations, say, following the setup of our subsequent empirical experiments. The objective is to forecast y_{m+1} using information available prior to the forecasting origin. For each forecasting period m, let τ_m denote the forecasting origin and define the most recent previous D daily predictor observations as:

$$\mathcal{D}_m = \{\mathbf{x}_{\tau_m - d} : d = 1, \ldots, D\}, \tag{19}$$

where D denotes the embedding window length. In our empirical application, we set $D = 22$, corresponding approximately to one trading month.

The HPE method constructs a path matrix containing the recent history of all predictors,

$$\mathbf{X}_m = \left[\mathbf{x} * \tau_m - 1^\top \; \mathbf{x} * \tau_m - 2^\top \vdots \mathbf{x}_{\tau_m - D}^\top\right] \in \mathbb{R}^{D\times K}. \tag{20}$$

The path matrix is then vectorized to obtain a fixed-dimensional feature representation,

$$\tilde{\mathbf{X}}_m = \mathrm{vec}(\mathbf{X}_m) \in \mathbb{R}^{DK}. \tag{21}$$

This embedding preserves both the temporal ordering of observations and the cross-sectional information contained in the predictor set while avoiding explicit aggregation of daily observations.

The forecasting model is given by

$$y_{m+1} = f^\star(\tilde{\mathbf{X}}_m) + \varepsilon_{m+1}. \tag{22}$$

where $f^\star(\cdot)$ denotes an unknown forecasting function approximated by machine learning methods, for example, such as those outlined in Section 2.1 above. This method allows forecasting models to directly learn predictive patterns from the recent high-frequency predictor path without imposing a predefined temporal weighting scheme.

2.3.2 Method 2: Group-Structured High-Frequency Path Embedding with Autoencoding (GHPA)

GHPA extends the HPE approach by combining predictor screening, path embedding, and nonlinear dimensionality reduction. The objective is to construct a compact representation of the high-frequency information that retains predictive content while mitigating overfitting in high-dimensional forecasting environments. The method consists of a number of steps, outlined below.

Step 1 - Predictor Screening: Let $\mathbf{x}_{m,d} \in \mathbb{R}^K$ denote the vector of daily predictors observed on day d within month m. To reduce dimensionality before constructing high-frequency paths, we first perform a LASSO screening step using monthly aggregated predictors.

For each month m, define

$$\bar{\mathbf{x}}_m = \frac{1}{D} \sum_{d=1}^{D} \mathbf{x}_{m,d}, \tag{23}$$

where $D = 22$ denotes the number of daily observations retained in each monthly window. The screening model is

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left[\sum_{m} \left(y_{m+1} - \bar{\mathbf{x}}_m^\top \boldsymbol{\beta} \right)^2 + \lambda \|\boldsymbol{\beta}\|_1 \right], \tag{24}$$

where y_{m+1} denotes the target monthly return. The selected predictor set is

$$\widehat{\mathcal{S}} = \left\{ j \in \{1, \ldots, K\} : \widehat{\beta}_j \neq 0 \right\}. \tag{25}$$

Let $\mathbf{z}_{m,d} = (x_{m,d,j})_{j \in \widehat{\mathcal{S}}} \in \mathbb{R}^p, p = |\widehat{\mathcal{S}}|$, denote the retained daily predictors used to construct the high-frequency path representation.

Step 2 - Path Embedding: Using the screened predictors, construct a high-frequency path representation over the most recent D trading days,

$$\mathbf{X}_m^{(S)} = \begin{bmatrix} \mathbf{z}_{\tau_m - 1}^\top \\ \mathbf{z}_{\tau_m - 2}^\top \\ \vdots \\ \mathbf{z}_{\tau_m - D}^\top \end{bmatrix} \in \mathbb{R}^{D \times p}. \tag{26}$$

The path matrix is vectorized to obtain

$$\tilde{\mathbf{X}}_m^{(S)} = \mathrm{vec}\left(\mathbf{X}_m^{(S)} \right) \in \mathbb{R}^{Dp}. \tag{27}$$

Step 3 - Autoencoder Compression: Because the embedded path can be high-dimensional, we apply an autoencoder to obtain a lower-dimensional representation. Let $h_\theta : \mathbb{R}^{Dp} \to \mathbb{R}^q$ denote the encoder and $g_\phi : \mathbb{R}^q \to \mathbb{R}^{Dp}$ the decoder. The autoencoder is estimated by minimizing

reconstruction loss,

$$(\hat{\theta}, \hat{\phi}) = \arg\min_{\theta,\phi} \sum_{m=1}^{M} \left\| \tilde{\mathbf{X}}_m^{(S)} - g_\phi\Big(h_\theta\Big(\tilde{\mathbf{X}}_m^{(S)}\Big)\Big) \right\|_2^2. \tag{28}$$

The resulting latent representation is

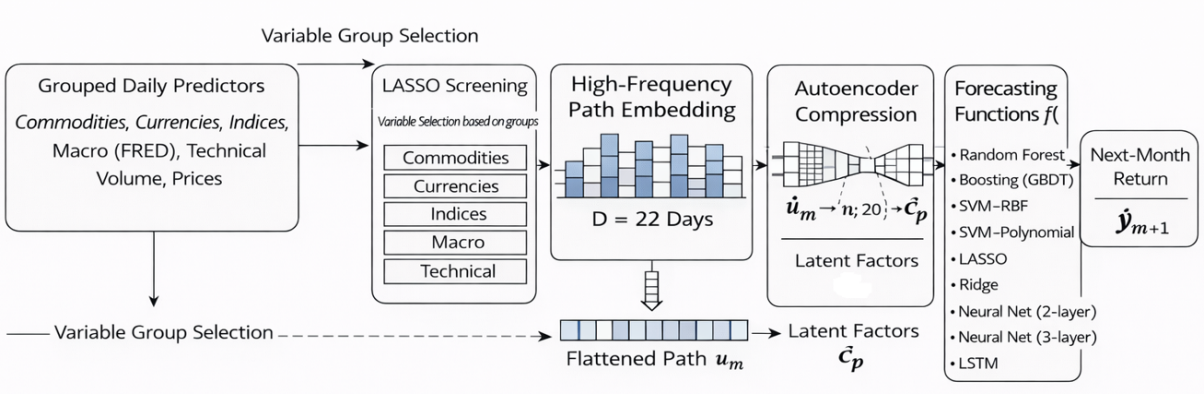
$$\mathbf{c}_m = h_{\hat{\theta}}\Big(\tilde{\mathbf{X}}_m^{(S)}\Big), \qquad \mathbf{c}_m \in \mathbb{R}^q, \qquad q \ll Dp. \tag{29}$$

Step 4 - Forecasting: The compressed features are then used to forecast the lower-frequency target,

$$y_{m+1} = f^\star(\mathbf{c}_m) + \varepsilon_{m+1}, \tag{30}$$

where $f^\star(\cdot)$ is approximated using the forecasting models described in Section 2.1. Figure 1 summarizes the GHPA procedure.

Figure 1: Group-Structured Mixed-Frequency Path Autoencoding (GHPA) Method



Notes: The figure illustrates the six-step GHPA procedure for next-month return prediction. The procedure consists of (1) grouping daily predictors, (2) LASSO screening based on predictor groups, (3) high-frequency path embedding using the most recent $D = 22$ trading days, (4) nonlinear dimensionality reduction via an autoencoder, (5) forecasting using machine learning models (Random Forest, Gradient Boosting, SVM, LASSO, Ridge, Feedforward Neural Networks, and LSTM), and (6) next-month return prediction, $\hat{y}_{m+1}$.

2.3.3 Method 3: Diagonal Multi-Horizon Temporal Aggregation (DMHTA)

DMHTA transforms a panel of high-frequency multi-horizon forecasts into a lower-frequency forecast. Let $\{y_t\}_{t\in\mathcal{T}}$ denote a target variable observed at the higher frequency. Using high-frequency predictors, we estimate horizon-specific forecasting models for horizons $h = 1, \ldots, H$, where $H = 22$ corresponds approximately to one trading month in the case of stock market returns forecasting as in our empirical experiment. Following the direct multi-step forecasting literature (Marcellino et al., 2006), a separate forecasting model is estimated for each horizon.

For each horizon h, one of the forecasting base learners introduced in Section 2.2 is estimated to produce the horizon-specific forecast

$$\hat{y}_{t,h} = f_h(\mathbf{x}_t),$$

where $f_h(\cdot)$ denotes the forecasting function for horizon h. Thus, for each forecasting model, a panel of multi-horizon forecasts is obtained. Collecting forecasts across horizons yields the prediction vector

$$\hat{\mathbf{y}}_t = (\hat{y}_{t,1}, \ldots, \hat{y}_{t,H})^\top \in \mathbb{R}^H.$$

The following steps describe the DMHTA aggregation procedure.

Step 1 - Diagonal Daily-to-Lower-Frequency Alignment: Rather than relying on a single forecast origin, DMHTA combines overlapping forecasts that correspond to the same future realization. Let $\mathcal{D}(m)$ denote the set of trading days in period m,

$$\mathcal{D}(m) = \{t_{m,1} < t_{m,2} < \cdots < t_{m,N_m}\}, \tag{31}$$

where N_m is the number of trading days in the period. Define the reverse within-period index,

$$\tilde{t}_{m,k} = t_{m,N_m-k}, \qquad k = 0, \ldots, N_m - 1, \tag{32}$$

so that $\tilde{t}_{m,0}$ denotes the last trading day of the period, $\tilde{t}_{m,1}$ the second-to-last trading day, and so on. For each $\tilde{t}_{m,k}$, construct a diagonal-aggregated forecast:

$$\hat{g}(\tilde{t}_{m,k}) = \frac{1}{k+1} \sum_{i=0}^{k} \hat{y}_{\tilde{t}_{m,k}-H+i,\, H-i}, \qquad k = 0, \ldots, N_m - 1, \tag{33}$$

where $\hat{y}_{t,h}$ denotes the h-step-ahead forecast produced at time t. As i increases, the forecast origin advances by one period while the forecast horizon decreases by one, tracing a diagonal through the multi-horizon prediction panel. When required observations are unavailable, the corresponding terms are omitted.

Step 2 - Lower-Frequency Forecast Construction: The lower-frequency forecast is obtained by aggregating the diagonal-adjusted forecasts within each period:

$$\hat{y}_{m+1} = \sum_{t \in \mathcal{D}(m)} \hat{g}(t), \tag{34}$$

where $g(\cdot)$ is defined above.

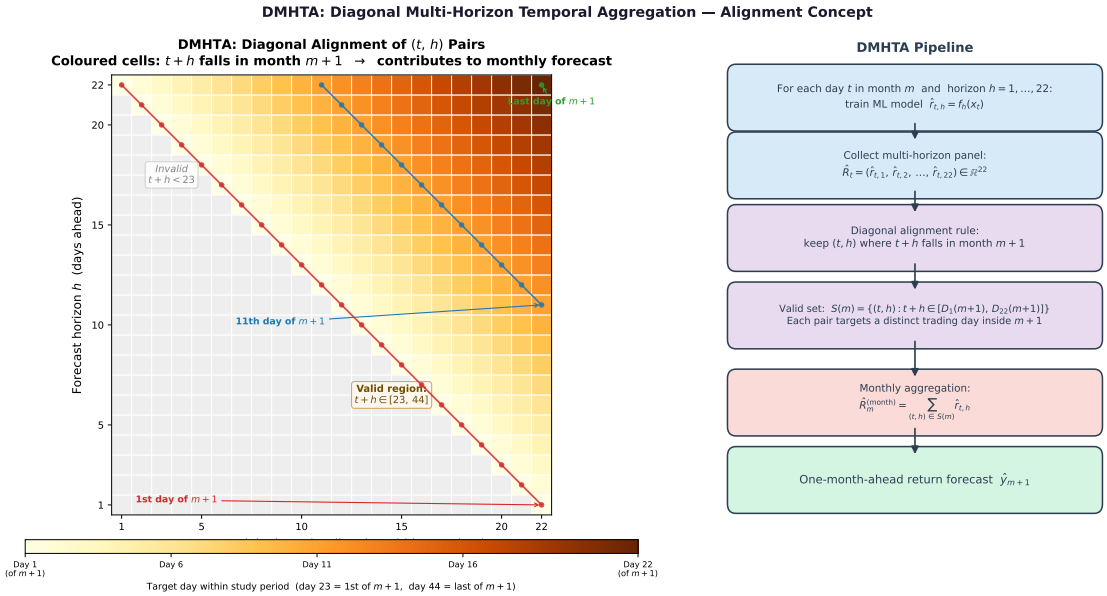
Step 3 - Model-Agnostic Forecast Aggregation: The transformation in (33)–(34) is applied uniformly across all forecasting models. Let $\mathcal{F}$ denote the set of forecasting models considered. For each model $f \in \mathcal{F}$, the resulting lower-frequency forecast is

$$\hat{y}^{(f)}_{m+1} = \sum_{t \in \mathcal{D}(m)} \left(\frac{1}{k(t,m)+1} \sum_{i=0}^{k(t,m)} \hat{y}^{(f)}_{t-H+i,\, H-i} \right), \tag{35}$$

where $k(t,m)$ denotes the reverse within-period position index defined above. This yields a comparable lower-frequency forecast series for each forecasting model. Figure 2 illustrates the

DMHTA procedure, and Figure 3 compares the three proposed methods.

Figure 2: Illustration of the Diagonal Multi-Horizon Temporal Aggregation (DMHTA) Method.



Notes: Left panel: The 22×22 alignment grid, where the horizontal axis indexes forecast-origin days t within month m and the vertical axis indexes forecast horizons h. Colored cells represent (t, h) pairs for which the target day $t+h$ falls within month $m+1$, with the color indicating the corresponding calendar day in month $m+1$. Each anti-diagonal (dashed lines) corresponds to a fixed target day. *Right panel*: The five-step DMHTA procedure from horizon-specific daily forecasting models to the final one-month-ahead return forecast.

3 Data

We collect a predictor dataset from multiple sources, including Yahoo Finance, the Federal Reserve Bank of New York, and the Federal Reserve Bank of St. Louis Economic Database (FRED). We also incorporate the updated predictor dataset from Goyal et al. (2024) in our dataset. These data are useful as they provide a comprehensive set of equity premium predictors widely used in the return predictability literature. See Table 1 for a complete list of variables.

Our sample spans from 2006/06/26 to 2023/12/29, comprising a total of 4,410 trading days. The forecasting targets include monthly returns of the SPY[1], as well as selected large-cap individual stocks including: NVIDIA (NVDA), JPMorgan Chase (JPM), Eli Lilly and Company (LLY) and Exxon Mobil Corp (XOM). Table 2 describes the five forecasting targets.

In summary, we use the 101 predictive variables listed in Table 1 as inputs for forecasting returns on our 5 target variables. These predictors encompass commodity prices, foreign exchange rates, volatility indices, global equity indices, Treasury yields and term spreads, mortgage rates, repo rates, corporate bond yields, technical indicators[2], lagged returns of the target

[1]SPDR S&P 500 ETF Trust.

[2]Technical indicators have been widely studied as predictors of future stock returns and equity risk premia (Neely et al. (2014)).

Figure 3: Three High-Frequency Forecasting Methods for Monthly Return Prediction

Notes: The figure compares the three proposed high-frequency forecasting methods. HPE constructs path embeddings from recent daily observations. GHPA combines group-structured LASSO screening with autoencoder compression. DMHTA aggregates horizon-specific daily forecasts to produce monthly predictions.

assets, and established equity premium predictors. To ensure stationarity, non-stationary variables are transformed using log-differences where appropriate. In addition, variables exhibiting substantially different scales are standardized prior to estimation so that all predictors are on a comparable magnitude.

In our forecasting experiments, we adopt a conventional data partitioning strategy widely used in the forecasting and machine learning literatures, dividing the full sample into training, validation, and testing subsets. Specifically, the dataset is divided such that the training set comprises 50% ($\text{ratio}_{\text{train}} = 0.50$) of the sample, the validation set constitutes 15% of the sample ($\text{ratio}_{\text{vali}} = 0.15$), and the remaining 35% of the sample is allocated to the testing set ($\text{ratio}_{\text{test}} = 0.35$).

4 The Forecasting Experiment and Evaluation Methods

As discussed above, we evaluate the proposed forecasting methods using monthly return forecasts for SPY, JPM, NVDA, XOM, and LLY. The analysis compares the monthly benchmark, HPE, GHPA, and DMHTA methods using a common set of forecasting models (base learners), including Random Forest, Gradient Boosting, Support Vector Machines, LASSO, Ridge, feedforward neural networks, and LSTM networks. Employing each of these seven models across all three mixed-frequency forecasting methods allows differences in predictive performance to be attributed to differences in information representation rather than model choice.

Hyperparameters are selected separately for each asset, mixed-frequency forecasting method, and forecasting model using a validation-based tuning procedure. For models with candidate hyperparameter grids, the optimal specification is chosen by minimizing validation-set mean squared forecast error (MSFE). Feedforward neural networks and LSTM models employ early stopping based on validation performance to mitigate overfitting. The hyperparameter search space varies across forecasting methods to account for differences in input dimensionality. In particular, the HPE method requires stronger regularization because the embedded predictor space is substantially larger than that of the monthly baseline, while the GHPA method does not as dimensionality is reduced through LASSO screening and autoencoder compression.

The out-of-sample evaluation sample period is 2017/12/01 – 2023/12/01. To assess potential structural changes associated with the COVID-19 pandemic, we also partition the evaluation window into a pre-COVID period (2017/12/01 – 2020/02/29) and a post-COVID period (2020/02/29 -– 2023/12/01). All models are estimated using the training sample and subsequently applied to the test sample, ensuring that all one-month-ahead forecasts are generated ex ante. Finally, recall that our benchmark models against which all results are compared include the random walk and autoregressive models given in Section 2.2. These benchmarks are our "non" mixed-frequency models, as they are estimated using only monthly returns data.

Forecasting performance is evaluated using the Mean Squared Forecast Error (MSFE),

$$\text{MSFE} = \frac{1}{T - T_0} \sum_{t=T_0+1}^{T} (r_t - \hat{r}_t)^2, \tag{36}$$

which measures average squared forecast error loss. Lower point MSFE values indicate superior forecasting performance. To formally assess whether differences in point MSFEs are statistically significant, we employ the Harvey et al. (1997) finite-sample corrected Diebold–Mariano (DM) tests (see Diebold and Mariano (1995)). Namely, assume a quadratic loss function, in which case we can define $d_t = e_{i,t}^2 - e_{j,t}^2$, where $e_{i,t}$ and $e_{j,t}$ denote the forecast errors of models i and j, respectively. The null hypothesis of equal predictive accuracy is $H_0 : \mathbb{E}(d_t) = 0$. Because these pairwise comparisons do not provide a joint ranking across multiple competing models, we additionally employ the Model Confidence Set (MCS) procedure of Hansen et al. (2011). The MCS procedure sequentially eliminates inferior forecasting models until a set of statistically indistinguishable superior models remains. We implement the $T_{\max}$ statistic using a quadratic

loss function and compute p-values using stationary bootstrap resampling with 5,000 bootstrap replications.

5 Empirical Results

The results summarized in this section are based on the empirical experiments discussed above, in which the objective is to forecast monthly stock price (index) returns using our three mixed-frequency forecasting methods across multiple assets, forecasting base learners, and market environments. Table 3 summarizes the forecasting models (base learners) and mixed-frequency forecasting methods considered in our analysis, as discussed in the preceding sections. Table 2 describes the target variables for which monthly returns are predicted. Table 4 reports descriptive statistics for the target return series. Note that in the sequel we concentrate our discussion on results for a single target variable, namely SPY. This is done for the sake of brevity, and because our qualitative findings remain unchanged regardless of which target variable we analyze. Results for the other target variables, including NVDA, JPM, XOM, and LLY are gathered in a Supplemental Appendix. Tabulated forecasting results are gathered in Tables 5–7, and Figures 4–6 provide visual summaries of the overall performance patterns across methods and assets.

The most consistent finding across all of our experiments is that incorporating higher-frequency information reduces forecast errors relative to the case where one solely uses monthly-frequency predictors.[3] For example, note that the first column of entries in Table 5 contains MSFEs for our cases where only monthly predictor data re used in the base learners (i.e., as inputs into the random walk and autoregression benchmarks as well as our more sophisticated machine learning forecast models). By reading across the rows in this table, one can compare the MSFEs from the purely monthly data-based approaches with our mixed-frequency approaches. Evidently, while only using monthly data sometimes (albeit rarely) matches the performance of our mixed-frequency methods, it never yields a lower MSFE, except in one cases. Namely, when the neural network model is used as the base learner for our prediction sub-sample from 2020-2023. Indeed, of the 33 cases reported across all three sample periods in the table, our mixed-frequency methods yield lower MSFEs in 23 cases, do as well as monthly data methods in 9 cases, and do slighlty worse in 1 cases.

Equally importantly, the lowest MSFE for each of our 3 sub-samples is never associated with forecasts constructed using solely monthly data. Additionally, the "globally best" base learner and mixed-frequency method MSFEs for each sub-sample are 0.288 (2017-2023 sample with SVM-RBF base learner and GHPA mixed-frequency modeling method), 0.171 (2017-2020 sample with SVM-RBF base learner and GHPA mixed-frequency modeling method), and 0.359 (2020-2023 sample with SVM-RBF base learner and GHPA mixed-frequency modeling method). While the support vector regression with radial basis function is "MSFE-best" (i.e., "wins") in all cases, it should be noted that most of our machine learning base learners also outperform our benchmarks most various cases.

[3]Results reported in our tables under the header "Monthly" refer to cases where only monthly data are used in our forecasting experiments.

Finally, looking across each row of entries in Table 5 note that various methods are statistically superior to the monthly mixed-frequency forecasting benchmark in which only monthly data are used in all calculations. Such models are denoted with starred entries in the table. Stated differently, starred entries denote rejections of the null hypothesis of equal forecast accuracy when comparing all mixed-frequency methods using a particular learner model, based on application of the DM test where the monthly benchmark is used as the null model against which the alternative mixed-frequency methods are compared. In most cases, these rejections occur when our mixed-frequency forecasting methods are implemented using nonlinear machine learning type "base learners", and not our autoregression benchmark model. The model confidence set (MCS) results presented in Table 6 provide complementary evidence. For example, in the full-sample SPY results, the proposed high-frequency methods are consistently retained in the superior model set across nearly all forecasting models, whereas the Monthly benchmark is more frequently assigned substantially lower MCS p-values and is excluded from the superior model set in several cases. Since the MCS identifies methods that cannot be statistically rejected as best-performing rather than providing a strict ranking of forecast accuracy, these findings complement the MSFE and DM results and further support the advantage of incorporating structured high-frequency information.[4]

Turning to the mixed-frequency methods proposed in this paper, note that Group-Structured High-Frequency Path Embedding with Autoencoding (GHPA) yields the greatest number of "wins" when comparing MSFEs across mixed-frequency forecasting methods for a given base learner model, yielding the lowest MSFE in 15 of 27 cases (not including the benchmarks). This results is summarized in the first Panel of Table 7 where the number of base learner model level "wins" are reported. This table also reports analogous summary findings for our other 4 target variables. Upon inspection of the results in this table, we see that across all five target variables GHPA "wins" in 61 of a possible 138 cases, of approximately one half of the time, when comparing the four data sampling methods. Additionally, the monthly method "wins" in only 14 cases.

Figure 4 provides a comprehensive visual summary of forecasting performance across all assets, forecasting models, and information methods. The underlying numerical results are reported in Tables 5–7. Inspection of these figures re-affirms that across assets spanning the broad market (SPY), technology (NVDA), financials (JPM), healthcare (LLY), and energy (XOM), the mixed-frequency forecasting methods often achieve lower MSFE values than the monthly benchmark. Indeed, reductions in MSFE are observed across a wide range of forecasting models and evaluation periods, suggesting that the benefits of incorporating higher-frequency information are systematic rather than driven by isolated asset/model combinations.

Summarizing these findings, recall that DMHTA and HPE represent two distinct ways of preserving intra-monthly information. DMHTA converts daily forecasts into monthly predictions through horizon-specific diagonal aggregation, while HPE directly embeds the full daily predictor path as a high-dimensional monthly feature vector. The empirical results discussed above indicate that both methods improve upon the monthly benchmark, but neither uniformly

[4]Models included in the model confidence sets for each base learner model are bolded entries in the table.

dominates the other. Figure 6 shows win count across all model–asset combinations in the full evaluation sample. The win-count evidence suggests that HPE is slightly more frequently the top-performing method than DMHTA, whereas HPE remains competitive in a number of specifications. The dominance of GHPA in Figure 6 is also reflected in the asset-level forecasting results. The generally strong performance of GHPA suggests that a substantial fraction of useful high-frequency information can be represented within a compact latent space without sacrificing predictive accuracy. In many cases, GHPA achieves forecasting performance comparable to or better than HPE despite operating on a substantially lower-dimensional representation. Overall, the evidence indicates that successful cross-frequency forecasting depends not only on access to high-frequency information, but also on how that information is structured and compressed before entering the forecasting model.

6 Robustness of Empirical Findings

Recall that Table 2 lists our five target variables. These assets span a broad market ETF, financial services, technology, energy, and healthcare sectors and differ substantially in volatility, information flow, market structure, and underlying economic fundamentals, providing a useful setting for evaluating the robustness of the proposed methods across diverse economic environments. Despite these differences, the benefits of incorporating higher-frequency information remain broadly consistent. Across all assets, at least one of the high-frequency methods outperforms the monthly benchmark in terms of MSFE. In many cases, the reductions in MSFE are robust across different forecasting models and subsamples. This consistency suggests that the value of structured high-frequency information representation extends beyond a particular asset class or sector and reflects a more general feature of cross-frequency forecasting problems.

Turning to the three different sample periods analyzed in our experiments, note that forecasting results reported in Tables 5–7 indicate that forecasting gains associated with using mixed-frequency data methods are generally larger during the post-COVID period (2020/02/29 – 2023/12/01).[5] More specifically, our mixed-frequency methods improve forecasting performance in both subsamples, but forecasting gains are generally larger during the post-COVID period. This pattern suggests that preserving intra-period information is always valuable, and becomes increasingly valuable when market conditions are characterized by heightened uncertainty and rapidly changing expectations. The reason why we draw this conclusion is that the post-COVID environment is characterized by heightened volatility, rapid changes in investor expectations, and elevated macroeconomic uncertainty. Under such conditions, monthly observations may provide an incomplete summary of the information arriving throughout the month. We posit that by preserving daily dynamics, HPE and GHPA are better able to capture evolving market conditions and changing information flows. More generally, our results imply that the benefits of mixed-frequency forecasting methods are likely to be greatest in environments characterized by elevated uncertainty, structural change, or rapidly evolving information sets.

[5] Figure 5 summarizes forecasting performance across the three sample periods.

7 Comparison with Factor-MIDAS

To provide a further robustness check against a benchmark from the mixed-frequency forecasting literature, we constructed a Factor-MIDAS model, as in (Marcellino and Schumacher, 2010). A conventional MIDAS (Ghysels et al., 2004, 2006) specification is difficult to implement in our setting because the predictor set contains 101 daily variables whereas the monthly forecasting sample includes only approximately 210 observations. We therefore adopted a dimension-reduction approach that combines LASSO variable screening, principal component analysis (PCA) based factor extraction, and MIDAS weighting similar to that used in (Ghysels et al., 2007). Following the screening procedure used in GHPA, we first applied the LASSO to monthly aggregated predictor information and retained variables with nonzero coefficients. PCA was then used to extract a small number of latent factors from the selected daily predictors, and the resulting factor series were aggregated using MIDAS weights before entering the final forecasting regression. For each month, the most recent 22 trading days of the selected predictors were collected to form a mixed-frequency predictor window. The resulting daily predictor paths were normalized using training-sample Min-Max scaling and transformed into a lower-dimensional factor representation using PCA. Let $F_{m,d}$ denote the latent factor vector on day d within month m. We retained the first five principal components, which were found to explain the majority of the variation in the selected predictor set. The Min-Max scaling parameters and PCA loadings were estimated using the training sample only and subsequently applied unchanged to the validation and testing samples. This procedure ensured that no information from future observations entered the factor construction process and avoided look-ahead bias. The daily factors were then aggregated using a MIDAS specification with exponential Almon lag weights. Specifically, the weighted factor representation used is given by:

$$\widetilde{F}_m = \sum_{j=1}^{22} w(j;\theta_1,\theta_2) F_{m,j},$$

where

$$w(j;\theta_1,\theta_2) = \frac{\exp(\theta_1 j + \theta_2 j^2)}{\sum_{k=1}^{22} \exp(\theta_1 k + \theta_2 k^2)}.$$

The parameters (θ_1, θ_2) were selected using validation-sample mean squared forecast error. The resulting weighted factors were then used in a linear forecasting model for next-month returns,

$$r_{m+1} = \alpha + \beta' \widetilde{F}_m + \varepsilon_{m+1}.$$

This benchmark preserves the core idea of MIDAS by imposing a parametric lag-weighting structure while accommodating the high-dimensional predictor environment through factor extraction. Importantly, the benchmark employs the same predictor universe, sample split, and first-stage variable-selection procedure as GHPA, ensuring a comparable evaluation method.

Table 8 reports the minimum MSFE achieved by Factor-MIDAS and the competing high-frequency methods across assets and evaluation periods. Factor-MIDAS generally outperforms the monthly benchmark and achieves competitive forecasting performance for several assets,

particularly NVDA and LLY. However, the proposed high-frequency methods typically deliver larger and more consistent forecasting gains. Across the fifteen asset–period combinations, the best-performing method is most often GHPA, HPE, or DMHTA, whereas Factor-MIDAS achieves the lowest MSFE in only a small number of cases. These findings suggest that preserving richer within-month information and allowing machine learning models to learn complex high-frequency representations can provide useful predictive benefits that may complement those gains already known to be associated with the use of MIDAS approaches, such as factor-based MIDAS aggregation.

8 Concluding Remarks

This paper introduces three methods for incorporating higher-frequency information into the prediction of lower-frequency outcomes. We propose three cross-frequency information-representation methods: Diagonal Multi-Horizon Temporal Aggregation (DMHTA), High-Frequency Path Embedding (HPE), and Group-Structured High-Frequency Path Embedding with Autoencoding (GHPA). These approaches transform daily information into structured monthly forecasting features while preserving different aspects of within-period dynamics. We use monthly stock return forecasting as a leading example to theoretically motivate and empirically illustrate the usefulness of the methods. Results show that stock returns forecasting performance depends critically on how higher-frequency information is represented when constructing forecasting models. Across a number of assets, linear and nonlinear machine learning type "base learner" forecasting models, and market environments the proposed methods are shown to consistently outperform conventional monthly-frequency predictors.

Among the proposed approaches, GHPA delivers the most robust performance, suggesting that combining structured feature construction with nonlinear compression provides an effective mechanism for extracting predictive information from high-frequency data. More broadly, our empirical findings highlight information representation as an important dimension of mixed-frequency forecasting. While the empirical analysis focuses on monthly stock return prediction, the proposed methods are broadly applicable to forecasting problems in finance, macroeconomics, energy markets, climate risk analysis, and other settings involving predictors and targets observed at different frequencies. Finally, comparison with a Factor-MIDAS benchmark further reinforces the usefulness of the methods that warrants their inclusion in an econometric "toolbox" alongside extant mixed-frequency aggregation approaches.

References

Andreou, E., E. Ghysels, and A. Kourtellos (2013). Should macroeconomic forecasters use daily financial data and how? *Journal of Business & Economic Statistics 31*(2), 240–251.

Athanasopoulos, G., R. J. Hyndman, N. Kourentzes, and F. Petropoulos (2017). Forecasting with temporal hierarchies. *European Journal of Operational Research 262*(1), 60–74.

Babii, A., E. Ghysels, and J. Striaukas (2022). Machine learning time series regressions with an application to nowcasting. *Journal of Business & Economic Statistics 40*(3), 1094–1106.

Banbura, M., D. Giannone, and L. Reichlin (2011). Nowcasting. *Oxford Handbook of Economic Forecasting*, 193–224.

Bates, J. M. and C. W. Granger (1969). The combination of forecasts. *Journal of the operational research society 20*(4), 451–468.

Bengio, Y., A. Courville, and P. Vincent (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence 35*(8), 1798–1828.

Bok, B., D. Caratelli, D. Giannone, A. M. Sbordone, and A. Tambalotti (2018). Macroeconomic nowcasting and forecasting with big data. *Annual Review of Economics 10*(1), 615–643.

Breiman, L. (2001). Random forests. *Machine Learning*.

Campbell, J. Y. and S. B. Thompson (2008). Predicting excess stock returns out of sample: Can anything beat the historical average? *The Review of Financial Studies 21*(4), 1509–1531.

Cortes, C. and V. Vapnik (1995). Support-vector networks. *Machine Learning*.

Diebold, F. X. and R. S. Mariano (1995). Comparing predictive accuracy. *Journal of Business and Economic Statistics 13*(3), 253–263.

Drucker, H., C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik (1996). Support vector regression machines. *Advances in neural information processing systems 9*.

Eldele, E., M. Ragab, Z. Chen, M. Wu, C. K. Kwoh, X. Li, and C. Guan (2021). Time-series representation learning via temporal and contextual contrasting. *arXiv preprint arXiv:2106.14112*.

Feng, G., S. Giglio, and D. Xiu (2020). Taming the factor zoo: A test of new factors. *The Journal of Finance 75*(3), 1327–1370.

Foroni, C., M. Marcellino, and C. Schumacher (2015). Unrestricted mixed data sampling (midas): Midas regressions with unrestricted lag polynomials. *Journal of the Royal Statistical Society Series A: Statistics in Society 178*(1), 57–82.

Franceschi, J.-Y., A. Dieuleveut, and M. Jaggi (2019). Unsupervised scalable representation learning for multivariate time series. *Advances in neural information processing systems 32*.

Friedman, J. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*.

Ghysels, E. (2016). Macroeconomics and the reality of mixed frequency data. *Journal of Econometrics 193*(2), 294–314.

Ghysels, E., P. Santa-Clara, and R. Valkanov (2004). The midas touch: Mixed data sampling regression models. *CIRANO Working Paper*.

Ghysels, E., P. Santa-Clara, and R. Valkanov (2006). Predicting volatility: Getting the most out of return data sampled at different frequencies. *Journal of Econometrics 131*, 59–95.

Ghysels, E., A. Sinko, and R. Valkanov (2007). Midas regressions: Further results and new directions. *Econometric Reviews 26*(1), 53–90.

Giannone, D., L. Reichlin, and D. Small (2008). Nowcasting: The real-time informational content of macroeconomic data. *Journal of Monetary Economics 55*(4), 665–676.

Goodfellow, I., Y. Bengio, A. Courville, and Y. Bengio (2016). *Deep learning*, Volume 1. MIT press Cambridge.

Goulet Coulombe, P., M. Leroux, D. Stevanovic, and S. Surprenant (2022). How is machine learning useful for macroeconomic forecasting? *Journal of Applied Econometrics 37*(5), 920–964.

Goyal, A., I. Welch, and A. Zafirov (2024). A comprehensive 2022 look at the empirical performance of equity premium prediction. *The Review of Financial Studies 37*(11), 3490–3557.

Gu, S., B. Kelly, and D. Xiu (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies 33*(5), 2223–2273.

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

Hansen, P. R., A. Lunde, and J. M. Nason (2011). The model confidence set. *Econometrica 79*(2), 453–497.

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

Hochreiter, S. and J. Schmidhuber (1997). Long short-term memory. *Neural Computation*.

Hoerl, A. and R. Kennard (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*.

Jordan, M. I. and T. M. Mitchell (2015). Machine learning: Trends, perspectives, and prospects. *Science 349*(6245), 255–260.

Kuan, C.-M. and H. White (1994). Artificial neural networks: an econometric perspective. *Econometric Reviews 13*(1), 1–91.

Lim, B., S. Ö. Arık, N. Loeff, and T. Pfister (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International journal of forecasting 37*(4), 1748–1764.

Marcellino, M. and C. Schumacher (2010). Factor midas for nowcasting and forecasting with ragged-edge data: A model comparison for german gdp. *Oxford Bulletin of Economics and Statistics 72*(4), 518–550.

Marcellino, M., J. H. Stock, and M. W. Watson (2006). A comparison of direct and iterated multistep ar methods for forecasting macroeconomic time series. *Journal of econometrics 135*(1-2), 499–526.

Mariano, R. S. and Y. Murasawa (2003). A new coincident index of business cycles based on monthly and quarterly series. *Journal of applied Econometrics 18*(4), 427–443.

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

Neely, C. J., D. E. Rapach, J. Tu, and G. Zhou (2014). Forecasting the equity risk premium: the role of technical indicators. *Management science 60*(7), 1772–1791.

Rolnick, D., P. L. Donti, L. H. Kaack, K. Kochanski, A. Lacoste, K. Sankaran, A. S. Ross, N. Milojevic-Dupont, N. Jaques, A. Waldman-Brown, et al. (2022). Tackling climate change with machine learning. *ACM Computing Surveys (CSUR) 55*(2), 1–96.

Schorfheide, F. and D. Song (2015). Real-time forecasting with a mixed-frequency var. *Journal of Business & Economic Statistics 33*(3), 366–380.

Sirignano, J. and R. Cont (2021). Universal features of price formation in financial markets: perspectives from deep learning. In *Machine learning and AI in finance*, pp. 5–15. Routledge.

Stock, J. H. and M. W. Watson (2002). Macroeconomic forecasting using diffusion indexes. *Journal of Business & Economic Statistics 20*(2), 147–162.

Swanson, N. R. and H. White (1997). Forecasting economic time series using flexible versus fixed specification and linear versus nonlinear econometric models. *International Journal of Forecasting 13*(4), 439–461.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*.

Timmermann, A. (2006). Forecast combinations. *Handbook of economic forecasting 1*, 135–196.

Timmermann, A. (2008). Elusive return predictability. *International Journal of Forecasting 24*(1), 1–18.

Welch, I. and A. Goyal (2008). A comprehensive look at the empirical performance of equity premium prediction. *The Review of Financial Studies 21*(4), 1455–1508.

Yue, Z., Y. Wang, J. Duan, T. Yang, C. Huang, Y. Tong, and B. Xu (2022). Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI conference on artificial intelligence*, Volume 36, pp. 8980–8987.

Table 1: Predictor Variables Used for Stock Return Forecasting

Group	Variable	Description
Technical Indicators		
Technical Indicators	indi_30_90	Indicator constructed using information from 30- to 90-day rolling window.
Technical Indicators	indi_90_120	Indicator constructed using information from 90- to 120-day rolling window.
Technical Indicators	indi_30_120	Indicator constructed using information from 30- to 120-day rolling window.
Technical Indicators	MA_30_90	Moving-average based feature comparing 30-day and 90-day windows.
Technical Indicators	MA_90_120	Moving-average based feature comparing 90-day and 120-day windows.
Technical Indicators	MA_30_120	Moving-average based feature comparing 30-day and 120-day windows.
Technical Indicators	Volume	Trading volume of the target asset.
Currencies (FX rates)		
Currencies	CNY=X	Chinese Yuan per U.S. dollar exchange rate.
Currencies	HKD=X	Hong Kong Dollar per U.S. dollar exchange rate.
Currencies	IDR=X	Indonesian Rupiah per U.S. dollar exchange rate.
Currencies	INR=X	Indian Rupee per U.S. dollar exchange rate.
Currencies	JPY=X	Japanese Yen per U.S. dollar exchange rate.
Currencies	MXN=X	Mexican Peso per U.S. dollar exchange rate.
Currencies	MYR=X	Malaysian Ringgit per U.S. dollar exchange rate.
Currencies	PHP=X	Philippine Peso per U.S. dollar exchange rate.
Currencies	RUB=X	Russian Ruble per U.S. dollar exchange rate.
Currencies	SGD=X	Singapore Dollar per U.S. dollar exchange rate.
Currencies	THB=X	Thai Baht per U.S. dollar exchange rate.
Currencies	ZAR=X	South African Rand per U.S. dollar exchange rate.
Commodities (Futures)		
Commodities	BZ=F	Brent crude oil futures price.
Commodities	CL=F	WTI crude oil futures price.
Commodities	HO=F	Heating oil futures price.
Commodities	NG=F	Natural gas futures price.
Commodities	RB=F	RBOB gasoline futures price.
Commodities	GC=F	Gold futures price.
Commodities	MGC=F	Micro gold futures price.
Commodities	SIL=F	Silver futures price.
Commodities	PL=F	Platinum futures price.
Commodities	PA=F	Palladium futures price.
Commodities	HG=F	Copper futures price.
Commodities	CC=F	Cocoa futures price.
Commodities	CT=F	Cotton futures price.
Commodities	KC=F	Coffee futures price.
Commodities	LBS=F	Lumber futures price.
Commodities	SB=F	Sugar futures price.
Commodities	ZO=F	Oats futures price.
Commodities	ZC=F	Corn futures price.
Commodities	ZR=F	Rough rice futures price.

Table 1 continued.

Group	Variable	Description
Commodities	ZS=F	Soybeans futures price.
Commodities	ZM=F	Soybean meal futures price.
Commodities	GF=F	Feeder cattle futures price.
Commodities	HE=F	Lean hog futures price.
Commodities	LE=F	Live cattle futures price.
Commodities	ZB=F	30-Year U.S. Treasury bond futures price.
Commodities	ZN=F	10-Year U.S. Treasury note futures price.
Commodities	ZF=F	5-Year U.S. Treasury note futures price.
Commodities	ZT=F	2-Year U.S. Treasury note futures price.
Equity and Volatility Indices		
Indices	000001.SS	Shanghai Composite Index.
Indices	399001.SZ	Shenzhen Component Index.
Indices	^FCHI	CAC 40 Index (France).
Indices	^FTSE	FTSE 100 Index (UK).
Indices	^HSI	Hang Seng Index (Hong Kong).
Indices	^N100	Euronext 100 Index.
Indices	^N225	Nikkei 225 Index (Japan).
Indices	^VIX	CBOE Volatility Index.
Indices	NQ=F	NASDAQ-100 futures price.
Indices	RTY=F	Russell 2000 futures price.
Indices	YM=F	Dow Jones Industrial Average futures price.
Macro and FRED Variables		
Macro / FRED	MF	Market factor (excess return on the market portfolio).
Macro / FRED	DFF	Effective Federal Funds Rate.
Macro / FRED	DGS3MO	3-Month Treasury constant maturity rate.
Macro / FRED	DGS1	1-Year Treasury constant maturity rate.
Macro / FRED	DGS2	2-Year Treasury constant maturity rate.
Macro / FRED	DGS3	3-Year Treasury constant maturity rate.
Macro / FRED	DGS5	5-Year Treasury constant maturity rate.
Macro / FRED	DGS10	10-Year Treasury constant maturity rate.
Macro / FRED	DGS20	20-Year Treasury constant maturity rate.
Macro / FRED	DGS30	30-Year Treasury constant maturity rate.
Macro / FRED	DFII5	5-Year inflation-indexed Treasury yield.
Macro / FRED	DFII10	10-Year inflation-indexed Treasury yield.
Macro / FRED	RRPONTSYD	Overnight reverse repurchase agreements, total securities.
Macro / FRED	RPONTTLD	Overnight Treasury reverse repurchase agreements.
Macro / FRED	RPONMBSD	Overnight mortgage-backed securities reverse repos.
Macro / FRED	MORTGAGE15US	15-Year fixed-rate mortgage average.
Macro / FRED	MORTGAGE30US	30-Year fixed-rate mortgage average.
Macro / FRED	BAMLC0A1CAAAEY	ICE BofA AAA U.S. corporate bond yield.
Macro / FRED	BAMLC0A2CAAEY	ICE BofA AA U.S. corporate bond yield.
Macro / FRED	BAMLC0A3CAEY	ICE BofA A U.S. corporate bond yield.
Macro / FRED	BAMLC0A4CBBBEY	ICE BofA BBB U.S. corporate bond yield.
Macro / FRED	BAMLH0A1HYBBEY	ICE BofA BB U.S. high-yield bond yield.
Macro / FRED	BAMLH0A2HYBEY	ICE BofA B U.S. high-yield bond yield.
Macro / FRED	BAMLH0A3HYCEY	ICE BofA CCC U.S. high-yield bond yield.
Macro / FRED	T5YIE	5-Year breakeven inflation rate.
Macro / FRED	T10YIE	10-Year breakeven inflation rate.
Macro / FRED	T5YIFR	5-Year, 5-Year forward inflation expectation.
Macro / FRED	T10Y2Y	10-Year minus 2-Year Treasury yield spread.
Macro / FRED	D12	Dividend-price ratio.
Macro / FRED	E12	Earnings-price ratio.
Macro / FRED	b/m	Book-to-market ratio.

Table 1 continued.

Group	Variable	Description
Macro / FRED	tbl	Treasury bill rate.
Macro / FRED	AAA	AAA-rated corporate bond yield.
Macro / FRED	BAA	BAA-rated corporate bond yield.
Macro / FRED	lty	Long-term government bond yield.
Macro / FRED	ntis	Net equity issuance.
Macro / FRED	Rfree	Risk-free rate.
Macro / FRED	infl	Inflation rate.
Macro / FRED	ltr	Long-term bond return.
Macro / FRED	corpr	Corporate bond return.
Macro / FRED	svar	Stock market variance.
Macro / FRED	CRSP_SPvw	CRSP value-weighted market return.
Macro / FRED	CRSP_SPvwx	CRSP value-weighted excess market return.

Notes: This table reports the predictor variables used in the forecasting models. The suffix "=X" denotes spot foreign exchange rates quoted against the U.S. dollar, while "=F" denotes futures contracts. The predictor set contains 101 variables, including 7 technical indicators, 12 foreign exchange rates, 28 commodity futures, 11 equity and volatility indices, and 43 macroeconomic and financial variables.

Table 2: Forecasting Target Variables

Target	Description	Transformation	Frequency
SPY	SPDR S&P 500 ETF Trust	$\ln(P_t) - \ln(P_{t-1})$	Monthly
NVDA	NVIDIA Corporation	$\ln(P_t) - \ln(P_{t-1})$	Monthly
JPM	JPMorgan Chase & Co.	$\ln(P_t) - \ln(P_{t-1})$	Monthly
XOM	Exxon Mobil Corporation	$\ln(P_t) - \ln(P_{t-1})$	Monthly
LLY	Eli Lilly and Company	$\ln(P_t) - \ln(P_{t-1})$	Monthly

Notes: This table reports the forecasting target variables used in the empirical analysis. All series are transformed into log returns. Sector representations are as follows: NVDA corresponds to the technology sector, JPM to the financial sector, XOM to the energy sector, and LLY to the healthcare sector. The SPY ETF serves as a broad market benchmark.

Table 3: Mixed-Frequency Forecasting Methods and "Base Learner" Forecasting Models

Component	Description
Mixed-Frequency Method	**Monthly baseline:** Directly forecast monthly returns using predictors sampled at the monthly frequency.
	HPE: High-Frequency Path Embedding that represents within-month daily predictor paths as monthly inputs.
	GHPA: Group-Structured High-Frequency Path Embedding with autoencoding that embeds daily paths by groups and compresses them into latent monthly representations.
	DMHTA: Diagonal Multi-Horizon Temporal Aggregation that maps daily predictors into monthly features via multi-horizon aggregation.
Base Learners	Random Walk – benchmark Autoregression (AR Model) – benchmark Random Forest (RandForest) Gradient Boosting (GradBoost) Support Vector Machine w/ Radial Basis (SVM-RBF) Support Vector Machine w/ Polynomial Basis (SVM-Poly) Least Absolute Shrinkage Operator (LASSO) Ridge Feedforward NN w/ 2 layers (Nnet2) Feedforward NN w/ 3 layers (Nnet3) Long Short-Term Memory Network (LSTM)

Notes: All machine learning models listed under "Base Learners" are used as common forecasting functions for each of the three mixed-frequency forecasting methods and the "Monthly baseline" benchmark.

Table 4: Descriptive Statistics of Daily and Monthly Returns

Asset	Mean	Std. Dev.	Skewness	Kurtosis	N
Panel A: Daily Returns					
SPY	0.000458	0.012522	−0.0715	14.1701	4,408
NVDA	0.001633	0.030968	0.1670	8.0887	4,408
JPM	0.000681	0.024075	0.9361	18.1961	4,408
XOM	0.000377	0.017205	0.2475	10.2957	4,408
LLY	0.000744	0.016069	0.7342	12.4004	4,408
Panel B: Monthly Returns					
SPY	0.008942	0.045198	−0.5805	1.0091	210
NVDA	0.033506	0.137780	0.1671	1.0937	210
JPM	0.011590	0.081441	−0.2189	0.8034	210
XOM	0.006841	0.067090	0.3317	3.0544	210
LLY	0.014971	0.064246	0.0860	1.7347	210

Notes: Panel A uses daily closing prices and Panel B uses end-of-month closing prices. In the column labeled "Kurtosis" excess kurtosis is reported (kurtosis minus 3) so that a value of zero corresponds to a normal distribution. The sample period is 2006/06/26 – 2023/12/29.

Table 5: MSFEs of "Learner Models" Using Different Mixed-Frequency Methods

SPY Monthly Returns

	Mixed-Frequency Method			
Learner Model	**Monthly**	**HPE**	**GHPA**	**DMHTA**
Sample Period: 2017/12/01 – 2023/12/01				
Random Walk	**0.301**	–	–	**0.301**
AR Model	**0.303**	–	–	**0.303**
RanForest	0.319	**0.297**	0.303	0.309
Boosting	0.374	0.346	0.302	**0.295***
SVM-RBF	0.428	0.312**	**0.288****	0.301***
SVM-Poly	0.915	0.386***	**0.320*****	1.158
LASSO	**0.301**	**0.301**	**0.301**	0.303
Ridge	0.328	**0.299**	0.301	0.513
Nnet2	1.076	0.302***	**0.293*****	0.302***
Nnet3	0.305	**0.301**	0.302	0.302
LSTM	0.348	0.298	**0.295**	0.300**
Sample Period: 2017/12/01 – 2020/02/29				
Random Walk	**0.181**	–	–	**0.181**
AR Model	**0.182**	–	–	**0.182**
RanForest	0.205	**0.180**	0.182	0.192
Boosting	0.303	0.244	0.185	**0.178****
SVM-RBF	0.280	0.188**	**0.171****	0.180**
SVM-Poly	0.591	0.186***	**0.178*****	1.009
LASSO	**0.181**	**0.181**	**0.181**	0.184
Ridge	0.190	**0.180**	0.181	0.198
Nnet2	0.507	0.185***	**0.181*****	**0.181*****
Nnet3	0.207	0.182	0.184	**0.181**
LSTM	0.213	0.182	**0.177**	0.182
Sample Period: 2020/02/29 – 2023/12/01				
Random Walk	**0.373**	–	–	**0.373**
AR Model	**0.375**	–	–	**0.375**
RanForest	0.387	**0.366**	0.375	0.379
Boosting	0.417	0.407	0.372	**0.364**
SVM-RBF	0.517	0.386*	**0.359***	0.373**
SVM-Poly	1.110	0.506**	**0.405*****	1.248
LASSO	**0.373**	**0.373**	**0.373**	0.375
Ridge	0.412	**0.370**	0.373	0.702
Nnet2	1.417	0.372***	**0.360*****	0.374***
Nnet3	**0.364**	0.373	0.373	0.375
LSTM	0.430	**0.367**	**0.367**	0.371*

Notes: All reported MSFE entries are actual MSFEs multiplied by 10^{-2}. Bold values indicate the lowest MSFE within each model row. Superscripts denote statistical significance based on the application of one-sided Harvey–Leybourne–Newbold corrected Diebold–Mariano tests comparing each high-frequency method with the Monthly benchmark, where starred entries correspond to the following p-values: $^{*}p < 0.10$, $^{**}p < 0.05$, $^{***}p < 0.01$. Monthly refers to the benchmark mixed-frquency method that uses solely monthly-frequency predictors. DMHTA denotes Diagonal Multi-Horizon Temporal Aggregation. HPE denotes High-Frequency Path Embedding. GHPA denotes Group-Structured High-Frequency Path Embedding with Autoencoding. See notes to Sections 2-4 for further details.

Table 6: Model Confidence Set p-Values

SPY Monthly Returns

	Mixed-Frequency Method			
Learner Model	**Monthly**	**HPE**	**GHPA**	**DMHTA**
Sample Period: 2017/12/01 – 2023/12/01				
Random Walk	**1.0000**	–	–	**1.0000**
AR Model	**1.0000**	–	–	**1.0000**
RanForest	**0.6198**	**1.0000**	0.2248	**0.6198**
Boosting	0.0922	0.0922	**1.0000**	**0.9548**
SVM-RBF	0.0064	**0.5208**	**1.0000**	**0.6358**
SVM-Poly	0.0004	0.1088	**1.0000**	0.0000
LASSO	**0.8440**	**1.0000**	**1.0000**	**0.4526**
Ridge	**0.7478**	**1.0000**	**0.7832**	0.1092
Nnet2	0.0670	**0.4804**	**1.0000**	**0.5860**
Nnet3	**0.9660**	**1.0000**	**0.7870**	**0.9660**
LSTM	0.0772	**1.0000**	0.1648	**0.4538**
Sample Period: 2017/12/01 – 2020/02/29				
Random Walk	**1.0000**	–	–	**1.0000**
AR Model	**1.0000**	–	–	**1.0000**
RanForest	**0.5920**	**1.0000**	**0.6918**	**0.6918**
Boosting	0.0380	0.1120	**1.0000**	**0.7374**
SVM-RBF	0.0098	**0.4784**	**1.0000**	**0.7222**
SVM-Poly	0.0002	**1.0000**	0.0672	0.0000
LASSO	**0.8044**	**1.0000**	**1.0000**	**0.4422**
Ridge	**0.8858**	**1.0000**	**0.8858**	**0.8858**
Nnet2	0.0140	**1.0000**	**0.9172**	**0.9172**
Nnet3	**0.5278**	**1.0000**	**0.8174**	**0.8174**
LSTM	**0.3692**	**1.0000**	**0.6374**	**0.6374**
Sample Period: 2020/02/29 – 2023/12/01				
Random Walk	**1.0000**	–	–	**1.0000**
AR Model	**1.0000**	–	–	**1.0000**
RanForest	**0.8756**	**1.0000**	0.2308	**0.8756**
Boosting	**0.4364**	**0.4364**	**0.9366**	**1.0000**
SVM-RBF	0.0432	**0.6566**	**1.0000**	**0.7560**
SVM-Poly	0.0042	0.0738	**1.0000**	0.0000
LASSO	**0.9104**	**1.0000**	**1.0000**	**0.7528**
Ridge	**0.8632**	**1.0000**	**0.9722**	0.0460
Nnet2	0.1006	**0.3706**	**1.0000**	**0.6730**
Nnet3	**1.0000**	**0.9460**	**0.8970**	**0.8970**
LSTM	0.1128	**1.0000**	**0.2646**	**0.6474**

Notes: See notes to Table 5. Each cell reports the Model Confidence Set (MCS) p-value from the application of the $T_{\max}$ statistic of Hansen, Lunde & Nason (2011). Bold values denote methods with p-values > 0.25, indicating strong inclusion in the MCS. A method is excluded from the $\alpha = 10\%$ MCS if its p-value < 0.10.

Table 7: Model-Level "Win" Counts Across Assets and Evaluation Periods Based on MSFEs

Asset	Period	Monthly	HPE	GHPA	DMHTA
SPY	2017/12/01 – 2023/12/01	0	4	**5**	1
	2017/12/01 – 2020/02/29	0	3	**5**	2
	2020/02/29 – 2023/12/01	1	3	**5**	1
NVDA	2017/12/01 – 2023/12/01	0	0	**8**	1
	2017/12/01 – 2020/02/29	1	3	**5**	0
	2020/02/29 – 2023/12/01	0	1	**7**	1
JPM	2017/12/01 – 2023/12/01	1	2	**3**	**3**
	2017/12/01 – 2020/02/29	1	**5**	2	1
	2020/02/29 – 2023/12/01	1	1	3	**4**
XOM	2017/12/01 – 2023/12/01	1	2	2	**4**
	2017/12/01 – 2020/02/29	**4**	1	2	2
	2020/02/29 – 2023/12/01	0	2	1	**6**
LLY	2017/12/01 – 2023/12/01	1	2	**5**	1
	2017/12/01 – 2020/02/29	2	2	**3**	2
	2020/02/29 – 2023/12/01	1	2	**5**	1
Total Wins (out of 138)		14	33	**61**	30

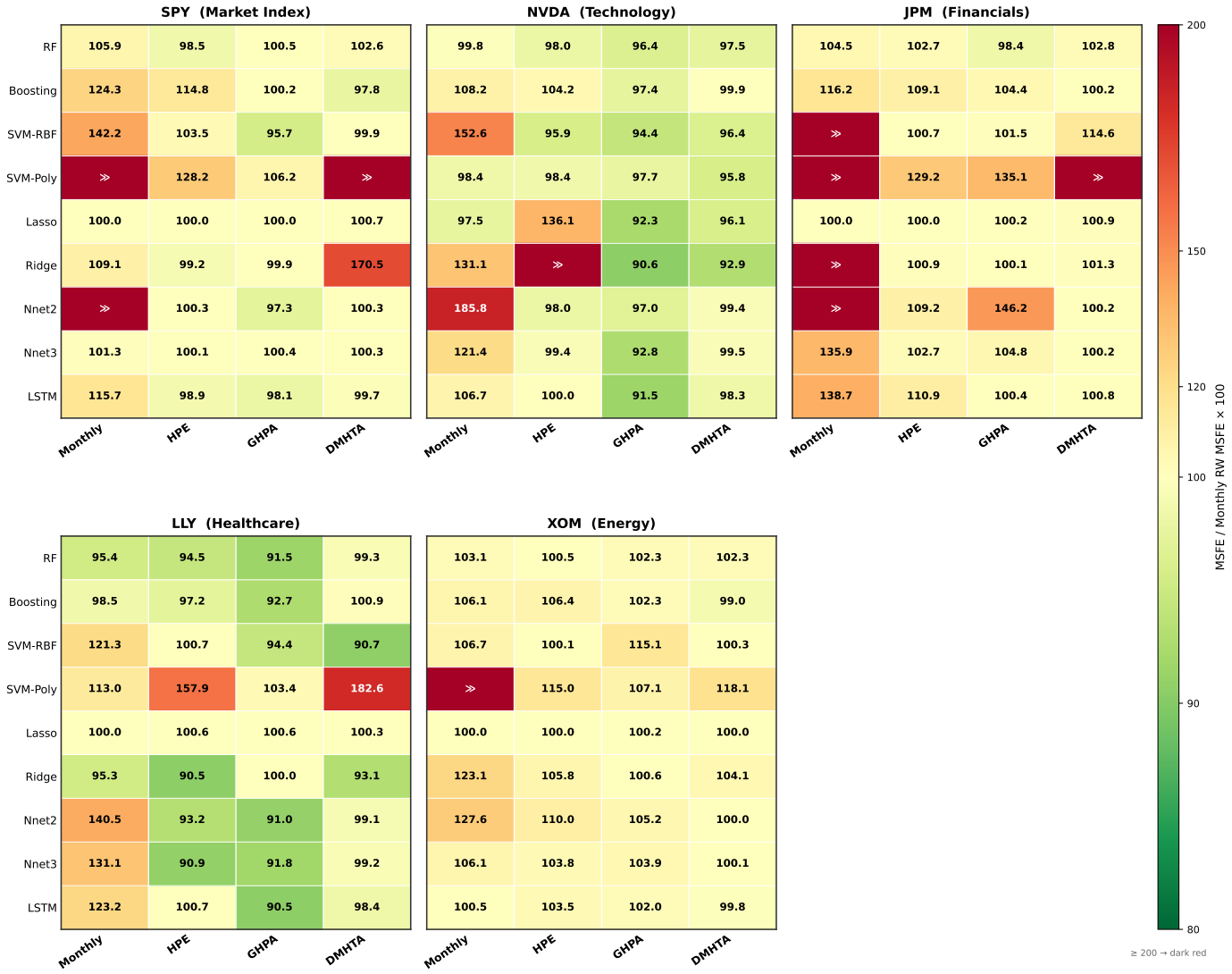
Notes: Entries report the number of model-level wins out of nine machine-learning models within each asset-period combination. A win is recorded when a framework achieves the *lowest* MSFE among the four competing frameworks for a given model; ties are credited to all tied methods, yielding a grand total slightly above $5 \times 3 \times 9 = 135$. The grand total is 138 rather than 135 because SPY–LASSO produces a two-way tie between HPE and GHPA across all three evaluation periods (both predict approximately the training-period mean), contributing 3 extra win credits (one per period). Random Walk and AR benchmarks are excluded because they are not applicable to the HPE and GHPA representations. Bold values indicate the framework with the largest number of wins within each asset-period combination. The final row aggregates wins across all five assets, three evaluation periods, and nine machine-learning models.

Table 8: MSFEs When Comparing Factor-MIDAS With Various Other Mixed-Frequency Forecasting Methods

Sample Period	Asset	Monthly	HPE	GHPA	DMHTA	Factor-MIDAS
2017/12/01 – 2023/12/01	SPY	0.301	0.297	**0.288**	0.295	0.300
	NVDA	2.315	2.276	**2.150**	2.206	2.189
	JPM	0.616	0.616	**0.606**	0.617	0.631
	XOM	0.883	0.884	0.885	**0.875**	0.880
	LLY	0.581	**0.551**	0.552	0.553	0.562
2017/12/01 – 2020/02/29	SPY	0.181	0.180	**0.171**	0.178	0.194
	NVDA	1.703	1.504	1.623	1.723	**1.453**
	JPM	**0.305**	0.344	0.335	0.363	0.374
	XOM	**0.385**	0.421	0.437	0.455	0.452
	LLY	0.319	0.318	0.310	0.332	**0.263**
2020/02/29 – 2023/12/01	SPY	0.364	0.366	**0.359**	0.364	0.363
	NVDA	2.631	2.578	**2.467**	2.497	2.631
	JPM	0.768	0.768	**0.753**	0.769	0.785
	XOM	1.132	1.134	1.137	**1.122**	1.136
	LLY	0.729	**0.684**	0.690	0.685	0.742

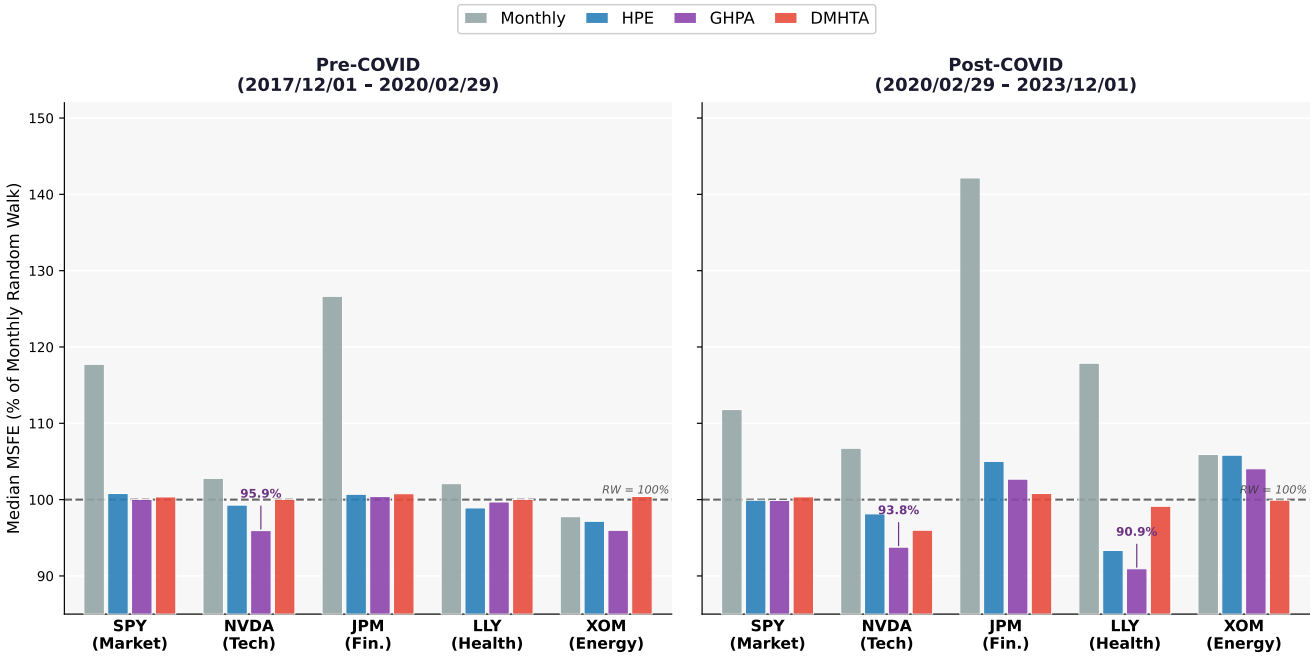
Notes: See notes to Tables 5-6. Tabulated entries are MSFEs multiplied by 10^{-2}. Reported values are the *minimum* MSFEs achieved across all nine competing base learner models (see list of "base learners" in Table 3). Bold values denote the best-performing (lowest MSFE) method for each asset–period pair.

Figure 4: MSFEs of "Learner Models" Relative to the Random Walk Benchmark



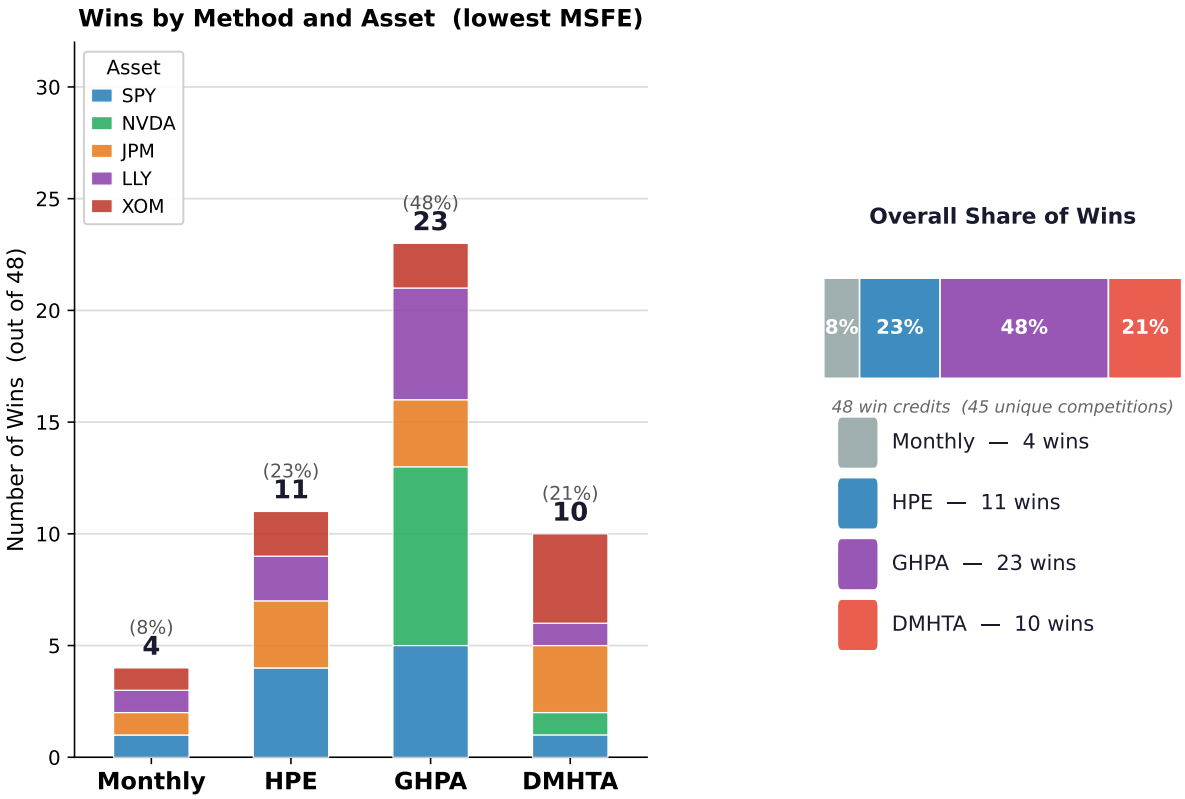
Notes: Entries report the MSFEs relative to the (non-mixed frequency) Monthly Random Walk benchmark for the full sample period (2017/12/01–2023/12/01). Darker green cells indicate lower forecast errors implying "better" forecasting performance, while darker red cells indicate higher forecast errors. Values are reported as $100 \times \text{MSFE}/\text{MSFE}_{RW}$, where MSFE_{RW} denotes the Monthly Random Walk benchmark (training-sample mean forecast). Values below 100 indicate outperformance relative to the benchmark, whereas values above 100 indicate underperformance. For readability, cells with $\text{MSFE}/\text{MSFE}_{RW} \geq 2$ (i.e., at least 200% of the benchmark MSFE) are denoted by "≫".

Figure 5: Median Relative MSFEs by Method, Asset, and Market Regime



Notes: Each bar reports median relative MSFEs across the nine machine learning base learners for the specified forecasting method, asset, and evaluation sub-period. Relative MSFEs are computed using the (non-mixed frequency) Monthly Random Walk benchmark, with values below one indicating outperformance relative to the benchmark. The two evaluation sub-periods correspond to the pre-COVID period (2017/12/01–2020/02/29) and the post-COVID period (2020/02/29–2023/12/01). Lower values indicate "better" forecasting performance.

Figure 6: Mixed-Frequency Forecasting Methods Win Count



Notes: The figure summarizes the number of wins achieved by each forecasting method across all machine learning model–asset combinations for the full evaluation period (2017/12/01–2023/12/01). A "win" is assigned to the method that attains the lowest MSFE for a given machine learning model and asset pair. Stacked bar segments indicate the contribution of each asset to the total number of wins.