Structured Deep Learning for Intraday Realized Volatility*

First Author† Second Author‡ Third Author§

December 10, 2025

Abstract

We examine how temporal structure, path geometry, and cross-sectional commonality can be exploited to improve intraday realized volatility forecasts. We develop two recurrent architectures—a multi-scale attention mechanism that learns horizon-specific dynamics and a signature-augmented design that encodes geometric information from high-frequency price paths—and introduce a cluster-based training scheme that pools stocks within industry groups and conditions on cluster-level realized volatility. Using minute-level data for large Chinese A-share stocks, we compare these models with standard econometric benchmarks, tree-based learners, and conventional recurrent networks. The evidence shows that nonlinear models dominate linear specifications, that multi-scale attention and signature augmentation yield systematic gains at short and longer horizons, and that the cluster-based scheme delivers the strongest and most robust improvements. Diebold–Mariano tests and Model Confidence Set procedures confirm a clear hierarchy of pooling strategies, with cluster-based training outperforming market-augmented, fully pooled, and stock-specific estimation. These results underscore sector-level comovement as a persistent and exploitable source of predictive power in intraday volatility.

Keywords: realized volatility forecasting, Attention-based neural networks, Path signatures,Cluster-based pooling,Commonality
JEL Codes: C45, C53, G17.

1 Introduction

Realized volatility constructed from high-frequency returns has become an essential input for empirical finance. It supports intraday risk management, derivative pricing, execution decisions, and market surveillance. As minute-level data become widely available, the demand for accurate short-horizon volatility forecasts has increased. Market conditions evolve quickly within the trading day, and high-frequency realized volatility often reflects these shifts in real time.

*Here are the thank-you remarks, funding information, etc. When writing your own thesis, change it to your own acknowledgments.

†Affiliation 1. Email: first.author@example.com.
‡Affiliation 2. Email: second.author@example.com.
§Affiliation 3.

Recent progress in machine learning has renewed interest in volatility forecasting. Studies using LSTM networks, convolution-based architectures, boosted trees, and hybrid deep-learning models report meaningful improvements over traditional econometric benchmarks such as HAR and GARCH-type specifications (Christensen et al., 2022; Bucci, 2020; Moreno-Pino and Zohren, 2024a; Gao et al., 2023; Liu et al., 2024a). These results suggest that nonlinear features embedded in high-frequency data can be exploited effectively. However, existing approaches still face important limitations when applied to intraday volatility.

A first limitation concerns temporal scale. Volatility responds to information at heterogeneous horizons, and the relevant scales vary across the trading day. Fixed lag structures used in econometric models capture these components only coarsely. Standard recurrent neural networks process sequences sequentially but do not explicitly learn which time segments contain the most predictive information. As a result, they often miss interactions between short-lived fluctuations and more persistent components.

A second limitation relates to the geometry of high-frequency price paths. Research in rough volatility and path signatures shows that the shape of a path carries information beyond raw increments (Gatheral et al., 2018; Bennedsen et al., 2022). High-frequency price series display irregularity, asymmetry, and higher-order interactions, especially around market events. Models that use sequences of returns alone find it difficult to extract these geometric features in a systematic way.

A third limitation arises from cross-sectional structure. Volatility co-moves strongly across stocks and sectors, and these common shocks propagate rapidly. Studies on volatility commonality, connectedness, and cross-sectional pooling document that incorporating such structure improves forecast accuracy (Herskovic et al., 2016; Bollerslev et al., 2018; Mensi et al., 2021; Zhang et al., 2024a). Yet most deep-learning frameworks train on either a single asset or a fully pooled sample, without distinguishing clusters of stocks that share structural patterns.

This paper addresses these limitations by developing two deep-learning architectures and a structured training scheme tailored to high-frequency realized volatility. The first architecture, the *Multi-Scale Attention* (MSA) family, enriches recurrent networks with a mechanism that identifies informative temporal horizons and models how these features interact. Instead of relying on pre-specified scales, the network learns them directly from the data. This design allows the model to adapt when the dominant patterns shift across horizons.

The second architecture, the *Signature-Augmented* (SA) family, incorporates path signatures into LSTM and GRU networks. Path signatures summarize the geometry of the price path and capture higher-order interactions among increments. When combined with a recurrent backbone, they allow the model to use both sequential information and geometric features that are especially relevant when volatility exhibits roughness or long-memory behavior.

We further introduce a *cluster-based training scheme* that groups stocks using industry and volatility commonality. This scheme sits between stock-specific and universal pooling approaches and leverages stable cross-sectional patterns in realized volatility.

Using minute-level data for 100 large A-share stocks from 2019 to 2024, we evaluate forecasting performance across four horizons—10-, 30-, 60-, and 240-minute—and four training schemes.

Benchmarks include OLS, LASSO, HAR, XGBoost, multilayer perceptrons, LSTM, and GRU. Forecast accuracy is assessed using MSE, RMSE, MAE, QLIKE, Realized Utility, Diebold–Mariano tests, and Model Confidence Set procedures.

Three main results emerge. The MSA architecture performs best at short horizons, where multi-scale interactions matter most. The SA architecture achieves strong gains at the daily horizon, consistent with the role of path geometry and long-memory components. The cluster training scheme improves all models and performs consistently across horizons, indicating that cross-sectional structure is a persistent feature of A-share volatility. These improvements are statistically significant and robust across evaluation metrics.

The contributions of this paper are fourfold. First, we propose two new deep-learning families—MSA and SA—that jointly capture multi-scale temporal structure and path-dependent geometric information. Second, we introduce a cluster-based training paradigm that exploits stable cross-sectional commonality. Third, we conduct a large-scale empirical analysis of 100 A-share stocks and provide a comprehensive comparison across econometric, machine-learning, and deep-learning models. Fourth, we document new evidence on the structure of intraday volatility in the Chinese market and show how it shapes the relative performance of different forecasting approaches.

The remainder of the paper is organized as follows. Section 2 reviews the related literature and positions our study within the existing work on high-frequency realized volatility and machine-learning forecasting. Section 3 introduces the proposed Signature-Augmented and Multi-Scale Attention architectures, together with the four training schemes, including the cluster-based approach. Section 4 describes the minute-level data, the construction of realized volatility measures, and the forecasting design across intraday horizons. Section 5 presents the forecasting results, statistical evaluation, and robustness analyses. Section 6 examines why the cluster-based training scheme outperforms alternative pooling strategies, drawing on economic reasoning, regression evidence, estimated coefficients, and adjusted R^2 comparisons. Section 7 concludes.

2 Literature Review

This section connects our study to three related strands of research: high-frequency realized volatility and econometric models; machine-learning and deep-learning approaches to volatility forecasting; and the literature on volatility commonality and cross-sectional dependence. These streams together frame the methodological and empirical motivations for the models developed in this paper.

2.1 High-Frequency Realized Volatility and Econometric Models

A large body of work has established realized volatility as a reliable nonparametric measure of ex-post return variation. Early contributions document the asymptotic properties of realized measures and interpret their behavior under market microstructure noise (Andersen et al., 2001a; Barndorff-Nielsen and Shephard, 2002, 2004). Subsequent studies show that volatility constructed from high-frequency data exhibits jumps, roughness, and strong persistence, which complicates modeling and forecasting. The Heterogeneous Autoregressive (HAR) model captures multi-scale volatility

dynamics through a structured lag design and has become a benchmark for realized volatility forecasting (Corsi, 2009). Extensions incorporate jumps, leverage effects, and long-memory features, while other studies examine the role of intraday periodicity and market microstructure patterns. These insights point to the intrinsically multi-scale nature of realized volatility.

Recent research highlights the importance of high-frequency structure for short-horizon forecasts. Studies document systematic patterns in intraday volatility, including strong variations around the open and close, intensity shocks linked to scheduled announcements, and changes in persistence across the day. For example, Boubaker et al. (2022) identify structural breaks and recurrent intraday patterns using high-frequency Asian market data, while Liu and Wen (2024) show that A-share volatility contains distinct short- and long-horizon components that evolve with information arrivals. Industry-specific variables and market conditions also shape realized volatility dynamics (Niu et al., 2023; Gunnarsson et al., 2024). These findings reinforce the view that volatility is driven by a combination of transient microstructure shocks and more persistent macro-driven movements, implying that effective forecasting models must account for interactions across multiple time scales.

Despite these advances, traditional econometric models face limitations when applied to intraday realized volatility. Fixed-scale structures such as HAR impose rigid temporal horizons, and linear specifications struggle to capture nonlinear transitions between different volatility regimes. Moreover, classical models do not incorporate geometric information contained in high-frequency price trajectories, nor do they exploit structured cross-sectional patterns across assets. These limitations motivate the development of flexible, data-driven approaches capable of modeling multi-scale temporal dependence and richer functional properties of high-frequency data.

2.2 Machine Learning and Deep Learning for Volatility Forecasting

Although deep learning is formally a subset of machine learning, it is standard in modern computational research to distinguish traditional machine-learning methods— such as support vector machines, random forests, and gradient boosting— from deep-learning neural architectures such as multilayer perceptrons, convolutional networks, LSTMs, and Transformers. This methodological separation is well justified by the foundational literature in the field. Classical machine-learning pipelines rely on hand-crafted or pre-extracted features fed into shallow learning algorithms, as emphasized by Bengio et al. (2013). In contrast, deep learning employs multi-layer, trainable nonlinear transformations to learn hierarchical representations directly from raw data, thereby enabling an end-to-end representation-learning paradigm (LeCun et al., 2015). The authoritative textbook by Bengio et al. (2017) further contrasts traditional feature-engineering workflows with deep neural networks, highlighting their fundamental differences in model structure, training mechanisms, input requirements, and scalability. Therefore, while deep learning is theoretically contained within the broader set of machine-learning methods, treating traditional machine-learning algorithms and deep-learning models as two distinct families of forecasting approaches is both conceptually sound and empirically necessary.

Machine learning has become an active area of research in volatility forecasting. Studies using tree-based methods, feed-forward networks, and recurrent neural networks report substantial

improvements over econometric benchmarks. For instance, Christensen et al. (2022) show that nonlinear models outperform HAR-type structures at both daily and intraday horizons, while Zhang et al. (2024a) demonstrate that modern networks capture complex relationships between realized volatility and cross-sectional predictors. Research in this area has expanded rapidly, covering convolutional architectures, transformer-style attention mechanisms, and hybrid designs that incorporate market microstructure variables.

High-frequency applications of deep learning have produced large gains in predictive accuracy. LSTM-based and GRU-based architectures have been used to model minute-level volatility, jumps, and return innovations. Liu et al. (2024b) develop an LSTM-HIT model that incorporates high-frequency intensity features and achieve strong performance at short horizons. Zhang et al. (2024b) investigate how positive and negative jumps contribute to volatility forecasting, while Gao et al. (2023) highlight the benefits of combining jump decomposition with nonlinear sequence models. Deep-learning studies on Asian markets also show meaningful gains, particularly when incorporating microstructure covariates or cross-asset information (Wu et al., 2024).

Even so, three structural limitations persist in the current literature. First, most architectures operate on a single temporal scale and therefore fail to extract interactions between short-lived fluctuations and more persistent movements. While convolutional networks and attention-based mechanisms partially address this issue, existing designs are not tailored for multi-scale volatility forecasting. Second, standard deep-learning models rely on raw input sequences and do not incorporate geometric or roughness-related information from high-frequency price paths. Advances in signature methods offer tools for extracting such structure (Lyons, 2014; Chevyrev and Kormilitzin, 2025), but these ideas have rarely been used in volatility forecasting. Third, deep-learning studies typically adopt either stock-specific or fully pooled training schemes, which overlook persistent cross-sectional patterns. Recent work on representation learning and cross-sectional deep learning (Li and Tang, 2025; Chen et al., 2024; Sirignano and Cont, 2021) provides motivation for integrating structured pooling into volatility forecasting models.

These observations motivate a new class of architectures that jointly address multi-scale temporal structure, path-dependent geometric information, and cross-sectional heterogeneity. Our proposed Multi-Scale Attention (MSA) and Signature-Augmented (SA) families build on these insights and contribute to the next generation of high-frequency volatility forecasting models.

2.3 Volatility Commonality and Cross-Sectional Dependence

The third strand of literature examines the cross-sectional structure of volatility. Early studies document strong commonality in liquidity and volatility across assets (Chordia et al., 2000; Herskovic et al., 2016). More recent work shows that volatility co-moves across sectors due to macroeconomic news, policy expectations, and market-wide information flows. Studies using network methods document spillovers and cluster-specific dynamics (Bollerslev et al., 2018; Mensi et al., 2021). These findings confirm that realized volatility contains persistent cross-sectional components that can be exploited for forecasting.

The relevance of cross-sectional structure is further emphasized in recent machine-learning studies. Christensen et al. (2022) and Zhang et al. (2024a) show that pooling information across stocks

improves forecast accuracy, especially when commonality is strong at intraday horizons. Research linking industry-level predictors and high-frequency volatility (Niu et al., 2023; Li et al., 2025b) provides additional support. In the A-share market, Niu et al. (2023) and Wu and Xie (2023) report significant spillovers and network-driven clustering using high-frequency data.

These findings collectively suggest that volatility forecasting models benefit from incorporating cross-sectional structure. Stock-specific models fail to capture common shocks, while fully pooled models ignore systematic differences across industries or volatility regimes. This motivates the cluster-based training scheme developed in this paper, which seeks to balance asset-specific information with stable group-level patterns and to provide a flexible platform for modeling cross-sectional commonality in realized volatility.

2.4 Summary and Implications

The existing literature provides several important insights for modeling and forecasting high-frequency realized volatility. Econometric studies emphasize the multi-scale structure of volatility and document persistent roughness, jumps, and intraday patterns that vary across horizons. Machine-learning and deep-learning research demonstrates meaningful improvements in predictive accuracy, yet most existing architectures remain restricted to single-scale inputs, raw-sequence representations, and homogeneous pooling schemes. Meanwhile, evidence from volatility commonality highlights stable cross-sectional components that propagate through sectors and industry clusters. Taken together, these findings point to a gap in the current forecasting literature: effective models must jointly account for heterogeneous temporal structure, geometric features of high-frequency price paths, and cross-sectional dependence. Our methodological framework addresses these needs by integrating multi-scale attention, signature-based representations, and cluster-based training into recurrent neural networks, thereby providing a flexible and data-driven approach to capturing the complex structure of high-frequency realized volatility.

3 Methodology

3.1 Forecasting Framework and Notation

Stock prices are modeled as realizations of a continuous-time stochastic process. Let $P_{i,t}(s)$ denote the price of stock i on day t at intraday time s. Following the standard approach in high-frequency financial econometrics, the log-price evolves as a semimartingale with stochastic volatility,

$$d\ln P_{i,t}(s) = \mu_{i,t}(s)\,ds + \sigma_{i,t}(s)\,dW_{i,t}(s), \tag{1}$$

where $\sigma_{i,t}(s)$ denotes the instantaneous volatility and $W_{i,t}(s)$ is a Brownian motion. This specification encompasses geometric Brownian motion as a special case and provides a flexible structure for describing intraday price variation (Andersen and Bollerslev, 1998; Barndorff-Nielsen and Shephard, 2002).

Within this framework, the latent *integrated variance* (IV) over a generic forecasting horizon h

is defined as

$$IV_{i,t}^{(h)} = \int_{s}^{s+h} \sigma_{i,t}(u)^2 \, du, \tag{2}$$

where $h \in \{10, 30, 60, 240\}$ minutes and s denotes the current intraday time. IV captures the continuous-time accumulation of volatility, but it is not directly observable. A large body of research shows that, under mild conditions and as the sampling grid becomes dense, the sum of squared intraday returns consistently estimates the integrated variance (Andersen et al., 2003; Barndorff-Nielsen and Shephard, 2002). Let $P_{i,t,m}$ denote the m-th intraday price for stock i on day t, and define the log-return

$$r_{i,t,m} = \ln P_{i,t,m} - \ln P_{i,t,m-1}. \tag{3}$$

For an aggregation horizon $h \in \{10, 30, 60, 240\}$ minutes consisting of M_h intraday observations, the realized variance proxy is

$$RV_{i,t}^{\text{raw},(h)} = \sum_{m=1}^{M_h} r_{i,t,m}^2. \tag{4}$$

Realized variance is highly skewed and heavy-tailed in practice. Empirical evidence suggests that the logarithmic transformation yields a distribution that is closer to Gaussian and improves the behavior of forecasting models. Andersen et al. (2001b) document that the natural logarithm of realized volatility is approximately normal, and the transformation has since become standard in both traditional econometric models and modern machine-learning applications. For example, Bucci (2020), Christensen et al. (2022) and Zhang et al. (2024a) all apply the log transformation in their realized-volatility forecasting setups, following the common practice of stabilizing variance and improving numerical behavior in neural-network models. In line with this convention, we define throughout the paper:

$$RV_{i,t}^{(h)} = \log\left(RV_{i,t}^{\text{raw},(h)}\right), \tag{5}$$

and refer to $RV_{i,t}^{(h)}$ simply as *realized volatility* or RV.

Our forecasting task is to rolling forecast the realized volatility at horizon h, based on the information available at intraday time t. Formally, the forecasting target is

$$\widehat{RV}_{i,t+h}^{(h)} = f_\theta(\mathcal{X}_{i,t}), \tag{6}$$

where $RV_{i,t+h}^{(h)}$ denotes the realized volatility computed over the future interval $[t,\, t+h]$.

For each trading day, this setup produces multiple intraday forecasts depending on the length of h. For example, when $h = 10$ minutes, we obtain 24 forecasts per day corresponding to the intervals 9:30–9:40, 9:40–9:50, ..., 14:50–15:00. Note that the A-share market in China operates for four hours each trading day, from 9:30 to 11:30 and 13:00 to 15:00. Likewise, a 30-minute horizon ($h = 30$) yields 8 forecasts per day; similar partitions are used for the 60-minute and the 240-minute (entire trading session) horizons. The information set $\mathcal{X}_{i,t}$ contains all predictors available at time t. The baseline inputs include a sequence of lagged realized volatilities $\{RV_{i,t}^{(h)}, RV_{i,t-h}^{(h)}, RV_{i,t-2h}^{(h)}, \ldots\}$, capturing the persistent nature of intraday volatility.

Given the large cross-section of assets, we complement individual stock features with aggregate volatility measures. The market-level realized volatility is defined as

$$RV^{(h)}_{\text{mkt},t} = \frac{1}{N}\sum_{i=1}^{N} RV^{(h)}_{i,t}, \tag{7}$$

and the cluster-level realized volatility for cluster g is

$$RV^{(h)}_{\text{clu}(\#g),t} = \frac{1}{|g|}\sum_{i\in g} RV^{(h)}_{i,t}. \tag{8}$$

These aggregates capture market-wide and sector-specific volatility dynamics and serve as key predictors under the augmented and cluster-based training schemes described later in the methodology.

The definitions and notation in this section establish the statistical environment for all forecasting models considered in the paper, including econometric benchmarks, traditional machine-learning algorithms, and the neural-network architectures developed in the subsequent subsections.

3.2 Benchmark Models

Unified forecasting representation. All forecasting models in this study share a common target and a consistent structure of predictor variables. For each asset i and horizon $h \in \{10, 30, 60, 240\}$ minutes, the quantity of interest is the future log-realized volatility $RV^{(h)}_{i,t+h}$ computed over the interval $[t,\, t+h]$. The baseline predictor set consists of a high-dimensional vector of lagged intraday log-realized volatilities for asset i, stacking multiple past buckets and preceding trading days. Formally, we denote this vector by

$$\mathbf{r}^{(h)}_{i,t} = \left(RV^{(h)}_{i,t}, RV^{(h)}_{i,t-h}, RV^{(h)}_{i,t-2h}, \dots\right)'.$$

In some specifications, this set may be augmented with aggregate predictors such as market-level or cluster-level realized volatility, depending on the training scheme introduced in Section 3.5. Let $\mathbf{a}^{(h)}_t$ denote these optional aggregate variables. All models considered in the paper can therefore be expressed in the general form

$$\widehat{RV}^{(h)}_{i,t+h} = f_\theta\left(\mathbf{r}^{(h)}_{i,t},\, \mathbf{a}^{(h)}_t\right), \tag{9}$$

where $f_\theta(\cdot)$ represents the forecasting function, with θ indexing the parameters of the econometric, machine-learning, or neural-network model. This unified representation provides a consistent foundation for the benchmark models and the proposed architectures introduced in the following subsections.

3.2.1 HAR-d Model

The Heterogeneous Autoregressive (HAR) model of Corsi (2009) is a widely used benchmark for realized volatility forecasting. The model approximates long-memory behavior by combining lagged

volatility measures computed over different time horizons. In the daily setting, the standard HAR specification for asset i takes the form

$$RV_{i,t+1}^{(d)} = \alpha_i + \beta_i^{(d)} RV_{i,t}^{(d)} + \beta_i^{(w)} RV_{i,t}^{(w)} + \beta_i^{(m)} RV_{i,t}^{(m)} + \varepsilon_{i,t+1}, \tag{10}$$

where $RV_{i,t}^{(d)}$ is the previous day's realized volatility, $RV_{i,t}^{(w)}$ is the weekly average $RV_{i,t}^{(w)} = \frac{1}{5}\sum_{k=0}^{4} RV_{i,t-k}^{(d)}$, and $RV_{i,t}^{(m)}$ is the monthly average $RV_{i,t}^{(m)} = \frac{1}{21}\sum_{k=0}^{20} RV_{i,t-k}^{(d)}$. The three components represent short-, medium-, and longer-term volatility movements, and the structure is known to perform well in environments where volatility displays persistent multi-scale dynamics.

To adapt the HAR model to intraday forecasting, Zhang et al. (2024a) propose an extension that incorporates two additional elements: a diurnal adjustment term and an intraday lag component. Following their specification, let $RV_{i,t}^{(h)}$ denote the log-realized volatility over the interval $[t-h,\, t]$ and let $RV_{i,t+h}^{(h)}$ denote the future realized volatility used as the prediction target. The diurnal-adjusted HAR model (HAR-d) is then written as

$$RV_{i,t+h}^{(h)} = \alpha_i + \beta_i^{(\tau)} D_{i,\tau_{t+h}} + \beta_i^{(s)} RV_{i,t}^{(h)} + \beta_i^{(d)} RV_{i,t}^{(d)} + \beta_i^{(w)} RV_{i,t}^{(w)} + \beta_i^{(m)} RV_{i,t}^{(m)} + \varepsilon_{i,t+h}^{(h)}. \tag{11}$$

The first new term, $D_{i,\tau_{t+h}}$, captures the intraday seasonal pattern of realized volatility. Here τ_{t+h} indexes the "bucket-of-the-day" corresponding to the interval $[t,\, t+h]$, such as 9:30–9:40 for $h = 10$ minutes. The value $D_{i,\tau_{t+h}}$ is computed as the average realized volatility of stock i at the same intraday bucket over the past 21 trading days. It reflects predictable patterns within the trading day caused by recurring liquidity cycles, trading intensity, and scheduled events.

The second new component, $RV_{i,t}^{(h)}$, is the lag-one intraday realized volatility. It summarizes the most recent high-frequency movement over the preceding horizon h, such as the 10-minute RV for the interval immediately prior to t. This term is particularly important in high-frequency applications, where short-term volatility clustering is pronounced.

The remaining three components, $RV_{i,t}^{(d)}$, $RV_{i,t}^{(w)}$, and $RV_{i,t}^{(m)}$, retain the structure of the daily HAR model. They are constructed from aggregated daily realized volatilities and capture persistent dynamics at longer temporal scales, even when the forecasting target is intraday. For example, $RV_{i,t}^{(d)}$ denotes the preceding day's realized volatility, while $RV_{i,t}^{(w)}$ and $RV_{i,t}^{(m)}$ represent the weekly and monthly averages defined above.

Equation (11) reduces to the standard daily HAR model in Equation (10) when the diurnal term and the intraday component are removed. Thus, HAR-d provides a natural and parsimonious extension of the classical HAR structure to the intraday environment. In our empirical analysis, we employ HAR-d as a structured linear benchmark. It offers a transparent point of comparison for evaluating the improvements brought by flexible machine-learning models and by the signature- and attention-based neural architectures introduced later in the methodology.

3.2.2 Ordinary Least Squares (OLS)

The OLS specification treats realized volatility forecasting as a high-dimensional linear regression problem based on past intraday realized volatilities. For each asset i and horizon $h \in$

$\{10, 30, 60, 240\}$ minutes, let $RV_{i,t+h}^{(h)}$ denote the log-realized volatility over the future interval $[t, t+h]$. We construct a predictor vector from multiple lagged intraday realized volatilities for asset i at horizon h.

Specifically, let

$$\mathbf{r}_{i,t}^{(h)} = \left(RV_{i,t}^{(h)}, RV_{i,t-h}^{(h)}, RV_{i,t-2h}^{(h)}, \dots\right)'$$

collect a sequence of past intraday realized volatilities observed in preceding buckets and trading days. The exact number of lags is chosen in the implementation and may vary across horizons, but in all cases $\mathbf{r}_{i,t}^{(h)}$ is a high-dimensional summary of the recent intraday volatility history for asset i. In some specifications, this predictor vector may be augmented with aggregate variables such as market- or cluster-level realized volatility; the precise configurations are described in the subsection 3.5 about training schemes below.

The OLS model is then given by

$$RV_{i,t+h}^{(h)} = \alpha_i + \boldsymbol{\beta}'\mathbf{r}_{i,t}^{(h)} + \varepsilon_{i,t+h}^{(h)}, \tag{12}$$

where $\boldsymbol{\beta}$ is a vector of slope coefficients and $\varepsilon_{i,t+h}^{(h)}$ is an error term with zero mean.

OLS provides a transparent linear benchmark that exploits a rich set of lagged intraday predictors while keeping the functional form simple. Similar high-dimensional linear frameworks have been used to model and forecast realized volatility and multivariate stochastic volatility in data-rich environments (e.g. Poignard and Asai, 2023; Branco et al., 2024). In our study, this specification serves as a reference point for assessing the incremental gains from regularization and from the nonlinear and sequential models considered later in the paper.

3.2.3 LASSO Regression

The LASSO model uses the same type of predictors as the OLS specification but introduces an ℓ_1 penalty on the slope coefficients. This penalty shrinks many coefficients towards zero and performs implicit variable selection, which is attractive when the number of lagged intraday predictors is large relative to the effective sample size.

Let $y_{i,t+h}^{(h)} = RV_{i,t+h}^{(h)}$ denote the forecasting target, and define the regressor vector $\mathbf{x}_{i,t}^{(h)} = \mathbf{r}_{i,t}^{(h)}$, where $\mathbf{r}_{i,t}^{(h)}$ is the vector of multiple lagged intraday realized volatilities described above. In some variants, $\mathbf{x}_{i,t}^{(h)}$ may also include aggregate volatility predictors such as lagged market- or cluster-level realized volatility; these extensions are specified when we discuss the different training schemes.

For a given horizon h, the LASSO estimator solves

$$\left(\widehat{\alpha}_i, \widehat{\boldsymbol{\beta}}\right) = \arg\min_{\alpha_i, \boldsymbol{\beta}} \left\{ \frac{1}{T} \sum_t \left(y_{i,t+h}^{(h)} - \alpha_i - \boldsymbol{\beta}'\mathbf{x}_{i,t}^{(h)} \right)^2 + \lambda \|\boldsymbol{\beta}\|_1 \right\}, \tag{13}$$

where $\lambda > 0$ controls the strength of the regularization and $\|\boldsymbol{\beta}\|_1$ is the ℓ_1 norm of $\boldsymbol{\beta}$.

The tuning parameter λ is selected using a time-series cross-validation scheme that respects the temporal ordering of the data. We partition the sample into five consecutive folds and, for each candidate value of λ, evaluate forecasting performance in a rolling-origin (walk-forward) fashion.

This procedure preserves the chronological structure of the intraday series and avoids the look-ahead bias that would arise from random reshuffling of observations.

Regularized linear models such as LASSO, elastic net, and ridge regression have been shown to perform well in volatility forecasting tasks with many predictors, where multicollinearity and noise are important concerns (Hao et al., 2020; Ma et al., 2023; Li et al., 2025a). Empirical evidence suggests that LASSO-based specifications can outperform unpenalized OLS and remain competitive with more complex nonlinear models, especially when combined with feature-processing steps such as principal components analysis (Qiao et al., 2024).

Within our framework, the LASSO regression serves as a flexible linear benchmark that exploits a high-dimensional set of past intraday realized volatilities, while controlling model complexity through data-driven regularization.

3.2.4 XGBoost

Gradient boosting decision trees provide a flexible approach for capturing nonlinear relationships in high-dimensional data. Among these methods, XGBoost (Chen, 2016) has become one of the most widely used variants due to its computational efficiency, predictive accuracy, and explicit regularization. A growing literature demonstrates its effectiveness in realized volatility forecasting and in handling feature-rich environments that arise in high-frequency financial applications (e.g. Ding et al., 2022; Zhang et al., 2024a). XGBoost has also been incorporated into hybrid architectures that combine boosted trees with recurrent or attention-based networks, highlighting its ability to capture nonlinearities, local threshold effects, and interaction structures in volatility and return dynamics (Nobre and Neves, 2019).

The predictor vector of XGBoost consists primarily of multiple lagged intraday log-realized volatilities for each asset, collected across recent buckets and trading days to summarize local volatility behavior. In certain training schemes, these predictors may be expanded to include market- or cluster-level realized volatility, but the core specification relies solely on the asset's own volatility history.

Formally, XGBoost constructs the forecast as an additive ensemble of regression trees:

$$\widehat{RV}_{i,t+h}^{(h)} = \widehat{RV}_{i,t+h}^{(h,0)} + \sum_{m=1}^{M} \eta \, f_m(\mathbf{x}_{i,t}), \tag{14}$$

where $\widehat{RV}_{i,t+h}^{(h,0)}$ is an initial estimate (typically a constant), $\eta \in (0, 1]$ is the learning rate, and each f_m is a decision tree belonging to a function class $\mathcal{F}$ defined by its structure and leaf weights. This representation highlights the basic idea of boosting: the final forecast is the accumulated contribution of many small trees, with each tree providing a focused correction to the current approximation of the target.

The model is estimated in a stagewise manner. After $m-1$ trees have been added, the prediction is updated recursively according to

$$\widehat{RV}_{i,t+h}^{(h,m)} = \widehat{RV}_{i,t+h}^{(h,m-1)} + \eta \, f_m(\mathbf{x}_{i,t}), \tag{15}$$

so that each new tree makes a small, directed adjustment to the existing ensemble. From a practical perspective, this update rule means that the model learns in incremental steps: instead of fitting a single complex model at once, boosting builds the predictor gradually by repeatedly refining the remaining errors.

At boosting stage m, the newly added tree f_m solves a regularized optimization problem:

$$f_m = \arg\min_{f \in \mathcal{F}} \left\{ \sum_t \ell\left(RV_{i,t+h}^{(h)}, \widehat{RV}_{i,t+h}^{(h,m-1)} + f(\mathbf{x}_{i,t}) \right) + \Omega(f) \right\}, \tag{16}$$

where $\ell(\cdot)$ denotes a differentiable loss function and $\Omega(f)$ is a regularization term penalizing overly complex trees. Conceptually, this step identifies the tree that best reduces the current forecasting errors while avoiding unnecessary structural complexity. Each new tree therefore functions as a targeted correction to the existing ensemble, addressing whatever patterns remain unexplained by the trees added so far.

A common regularizer is

$$\Omega(f) = \gamma T_f + \frac{\lambda}{2} \sum_{j=1}^{T_f} w_{f,j}^2, \tag{17}$$

where T_f is the number of leaves in tree f and $w_{f,j}$ is the prediction associated with leaf j. The parameter γ discourages trees with too many leaves, thereby limiting excessive partitioning of the feature space, whereas λ shrinks the leaf weights toward zero, preventing large updates driven by noise or transient intraday fluctuations. Together, these penalties help stabilize the model and improve its generalization performance—properties that are particularly important when forecasting high-frequency realized volatility.

The learning rate η further modulates how much each tree influences the final forecast. A smaller η forces each tree to make only a modest adjustment to the existing ensemble; while this requires more boosting iterations, it usually yields a smoother and more robust predictor. Viewed from a forecasting perspective, each individual tree introduces a set of nonlinear threshold rules, and the additive ensemble allows these localized rules to interact flexibly. As a result, XGBoost can accommodate both abrupt changes and subtle interaction effects in intraday volatility dynamics, making it a powerful nonlinear benchmark against which we evaluate the more structured neural network architectures introduced later.

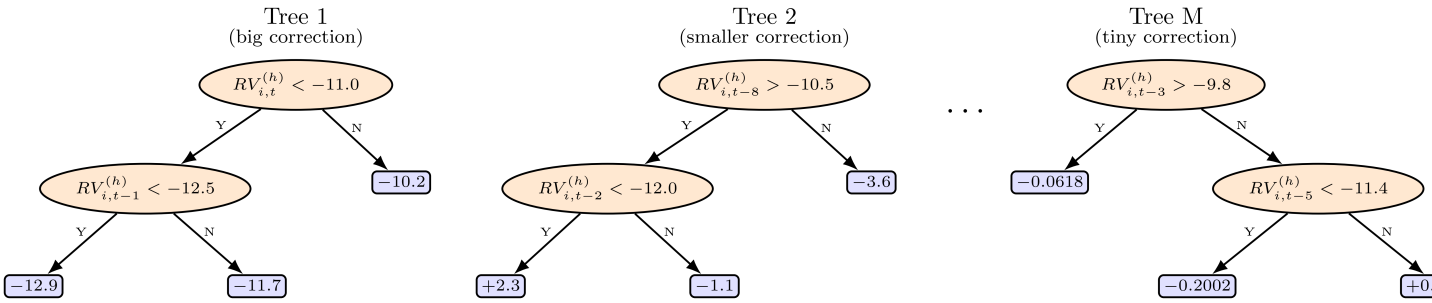


Figure 1: Illustration of the XGBoost boosting tree model.

3.2.5 Multilayer Perceptron (MLP)

Multilayer perceptrons provide a natural nonlinear extension of the linear benchmark models discussed above. An MLP maps the predictor vector $\mathbf{x}_{i,t}$ to the target through a sequence of affine transformations followed by nonlinear activation functions. Because of their universal approximation property, MLPs can capture a broad range of nonlinear relationships among lagged intraday log-realized volatilities while preserving a relatively simple and transparent structure. In empirical finance, MLPs have been applied widely to volatility forecasting, financial distress prediction, and hybrid deep-learning architectures that combine recurrent or attention-based components (e.g. Wu et al., 2022; Pradeepkumar and Ravi, 2017; Yang et al., 2024; García-Medina and Aguayo-Moreno, 2023).

For our forecasting task, the MLP takes multiple lagged intraday log-realized volatilities as inputs. These inputs summarize recent volatility dynamics at the asset level, and in some training schemes the predictor set may be augmented with market- or cluster-level information. Let $\mathbf{x}_{i,t} \in \mathbb{R}^d$ denote the predictor vector. A standard L-layer feedforward network computes

$$\mathbf{h}^{(1)} = \phi\Big(\mathbf{W}^{(1)}\mathbf{x}_{i,t} + \mathbf{b}^{(1)}\Big), \tag{18}$$

$$\mathbf{h}^{(\ell)} = \phi\Big(\mathbf{W}^{(\ell)}\mathbf{h}^{(\ell-1)} + \mathbf{b}^{(\ell)}\Big), \qquad \ell = 2, \dots, L, \tag{19}$$

where $\phi(\cdot)$ is a nonlinear activation function such as ReLU or tanh, and $\mathbf{W}^{(\ell)}$, $\mathbf{b}^{(\ell)}$ are the weights and biases of layer ℓ. The final forecast is obtained from the output layer,

$$\widehat{RV}^{(h)}_{i,t+h} = \mathbf{W}^{(L+1)}\mathbf{h}^{(L)} + \mathbf{b}^{(L+1)}. \tag{20}$$

These equations highlight the structure of the model: each hidden layer extracts progressively more nonlinear combinations of the lagged intraday log-realized volatilities, and the output layer converts these representations into a forecast of the future realized volatility at horizon h.

A substantial literature documents the effectiveness of MLPs in financial applications. Studies show that MLPs offer competitive performance in stock-return and volatility forecasting, particularly when combined with optimization schemes such as genetic algorithms or particle swarm methods (Namdari and Durrani, 2021; Li and Tang, 2025). In volatility modeling, MLPs often serve as baselines in hybrid architectures, where they complement recurrent networks by learning nonlinear transformations of short-run predictors (Yang et al., 2024). More recent work shows that MLPs can match or exceed the performance of recurrent architectures in settings with strong nonlinearities but limited long-range dependence, including cryptocurrency and high-frequency equity volatility forecasting (García-Medina and Aguayo-Moreno, 2023; Dudek et al., 2024).

Taken together, these findings indicate that MLPs provide a useful nonlinear benchmark for our forecasting environment. They combine simplicity and computational efficiency with the ability to learn rich nonlinear patterns in intraday realized volatility, offering a baseline against which the more structured recurrent and attention-based models introduced later can be compared.

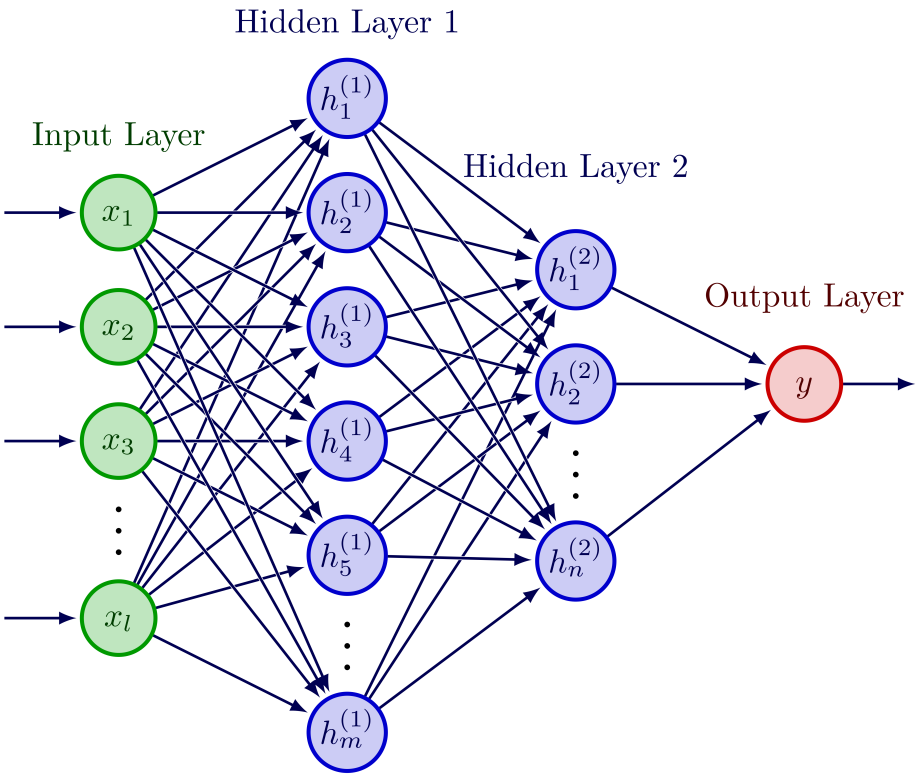


Figure 2: Illustration of the multilayer perceptron architecture.

3.2.6 Basic LSTM

Recurrent neural networks provide a natural way to handle ordered data, but standard architectures often struggle with long-range dependencies and unstable gradients. Long short-term memory (LSTM) networks address these issues by combining recurrent dynamics with a gated memory mechanism, and have been shown to perform well in a wide range of financial forecasting applications, including return prediction and volatility modeling (e.g. Fischer and Krauss, 2018; Zhang et al., 2020b; Cao et al., 2019). Recent studies further report that LSTM-based and hybrid LSTM models can outperform traditional econometric benchmarks such as GARCH and HAR in realized volatility and intraday volatility forecasting, especially in environments with pronounced nonlinearities, structural breaks, or regime changes (Gajamannage et al., 2023; Astudillo et al., 2025).

In our setting, the LSTM is used as a recurrent benchmark that takes a sequence of lagged intraday log-realized volatilities as input and produces a forecast of the future log-realized volatility at horizon h. For each asset i and forecast origin (t, h), we construct an input sequence $\{x_1, \ldots, x_T\}$ that stacks recent log-realized volatilities (and, in some specifications, additional covariates) ordered in calendar time. The LSTM processes this sequence step by step, updating a hidden state h_t and a cell state c_t that together summarize the information from the past.

The dynamics of a single LSTM layer follow the standard gated formulation. At time step t, given the current input x_t and the previous hidden state h_{t-1}, the input gate i_t, forget gate f_t,

output gate o_t, and candidate cell state $\tilde{c}_t$ are computed as

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)\,, \tag{21}$$
$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)\,, \tag{22}$$
$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c)\,, \tag{23}$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)\,, \tag{24}$$

where $\sigma(\cdot)$ denotes the logistic sigmoid function, $\tanh(\cdot)$ is the hyperbolic tangent, and $W.$, $U.$, $b.$ are learnable weight matrices and bias vectors. The cell state c_t and hidden state h_t are then updated via

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \tag{25}$$
$$h_t = o_t \odot \tanh(c_t), \tag{26}$$

where $\odot$ denotes the element-wise (Hadamard) product. In the context of realized volatility forecasting, the forget gate controls how much of the previous volatility information is retained, the input gate determines the impact of new intraday shocks on the latent volatility state, and the cell state c_t accumulates these effects over time. This structure allows the LSTM to preserve slow-moving volatility components while still reacting to abrupt intraday jumps.

To link the recurrent representation to our forecasting target, we use the final hidden state h_T of the last LSTM layer as a summary of recent intraday volatility dynamics and map it to the future log-realized volatility through a linear output layer,

$$\widehat{RV}^{(h)}_{i,t+h} = w_{\text{out}}^{\top} h_T + b_{\text{out}}, \tag{27}$$

where w_{out} and b_{out} are output parameters. Intuitively, h_T embeds information about both short-lived volatility spikes and more persistent movements across the input window, and the output layer translates this representation into a point forecast for the next intraday bucket.

Empirical evidence from equity, futures, and cryptocurrency markets suggests that LSTM models are particularly useful when volatility exhibits a mixture of persistent components and short-lived bursts, and when the predictive signal is distributed over multiple lags rather than concentrated in a small set of features (Gajamannage et al., 2023; Astudillo et al., 2025). In our application, the LSTM therefore serves as a natural recurrent benchmark: it is designed to exploit the sequential structure of the intraday log-realized volatility series, yet it does not impose the additional path-based or multi-scale structure that characterizes the signature-augmented and attention-based architectures introduced in the next subsections. This makes the LSTM a useful reference point for assessing the added value of those more specialized models.

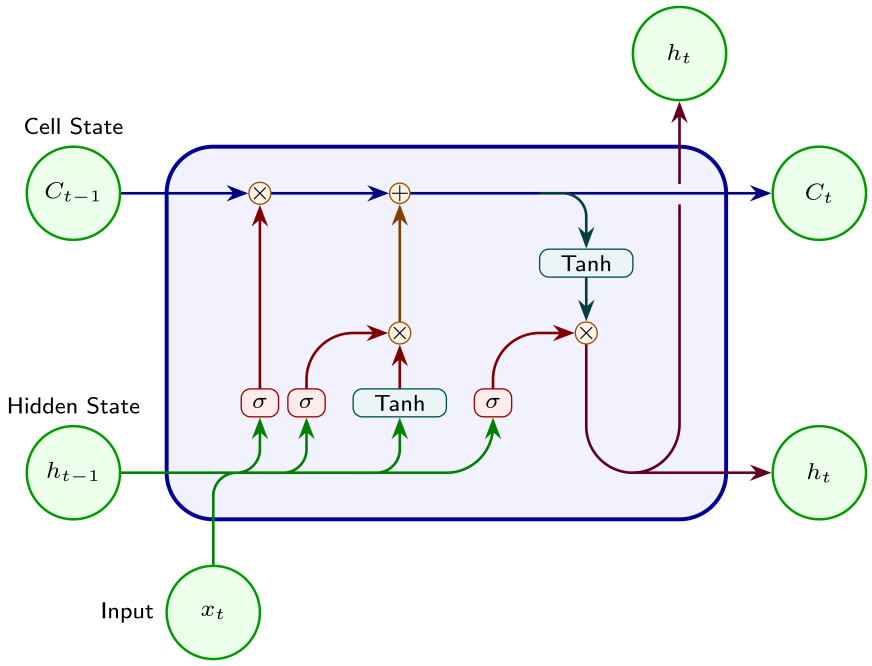


Figure 3: Illustration of the LSTM gating mechanism.

3.2.7 Basic GRU

Gated recurrent units (GRUs) provide a streamlined alternative to the LSTM architecture. They retain the key gating ideas that allow recurrent networks to capture nonlinear time dependence, yet achieve this with a simpler structure and fewer parameters. This parsimony makes GRUs attractive in high-frequency forecasting settings, where the data are noisy, the forecasting horizon is short, and excessive parameterization may reduce robustness. Empirical studies show that GRUs deliver competitive—and often superior—performance relative to LSTMs and traditional econometric benchmarks in financial volatility forecasting, exchange-rate modeling, and multivariate market prediction (e.g. Di Persio et al., 2023; Yu et al., 2023). Hybrid CNN–GRU architectures have also been shown to improve high-frequency volatility forecasts in both U.S. and Chinese equity markets (Song et al., 2024).

In our framework, the GRU is used as a recurrent benchmark alongside the LSTM. While both models rely on gating mechanisms to filter incoming information and preserve relevant components of past volatility, their designs differ in how memory is maintained. The LSTM uses a separate cell state and three gates; the GRU merges these elements into a single hidden state updated through two gates. This difference leads to a more compact parameterization that can be advantageous in high-frequency realized volatility forecasting, where the sequence length is moderate but the number of assets and forecast origins is large. Using both LSTM and GRU therefore allows us to evaluate whether the additional flexibility of the LSTM cell materially improves forecasts relative to the simpler GRU structure.

Formally, the GRU computes its hidden state using an update gate z_t and a reset gate r_t. Given

the input x_t (a lagged intraday log-realized volatility or derived feature) and the previous hidden state h_{t-1}, the gates and candidate hidden state are

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)\,, \tag{28}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)\,, \tag{29}$$

$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h)\,, \tag{30}$$

where $W.$, $U.$, and $b.$ are learnable parameters. The hidden state evolves according to

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t. \tag{31}$$

In this structure, the update gate determines how strongly past volatility information is retained, and the reset gate controls how past states influence the candidate update. The model can therefore respond quickly to short-lived volatility bursts while still preserving longer-run dependence, a combination that is particularly relevant for intraday realized volatility series.

As with the LSTM, the final hidden state h_T summarizes recent volatility dynamics and is mapped to the forecast through a linear output layer,

$$\widehat{RV}^{(h)}_{i,t+h} = w_{\text{out}}^{\top} h_T + b_{\text{out}}. \tag{32}$$

Recent empirical evidence suggests that GRUs may outperform LSTMs in settings where the predictive signal is relatively local in time or where the reduced parameterization helps stabilize the optimization (e.g. Di Persio et al., 2023; Song et al., 2024; Yu et al., 2023). In our application, the GRU thus serves as a complementary recurrent baseline: it retains the core dependence-tracking features of the LSTM while offering a more compact structure. Together, the LSTM and GRU allow us to assess the value of recurrent architectures before introducing the signature-augmented and multi-scale attention models developed in later subsections.

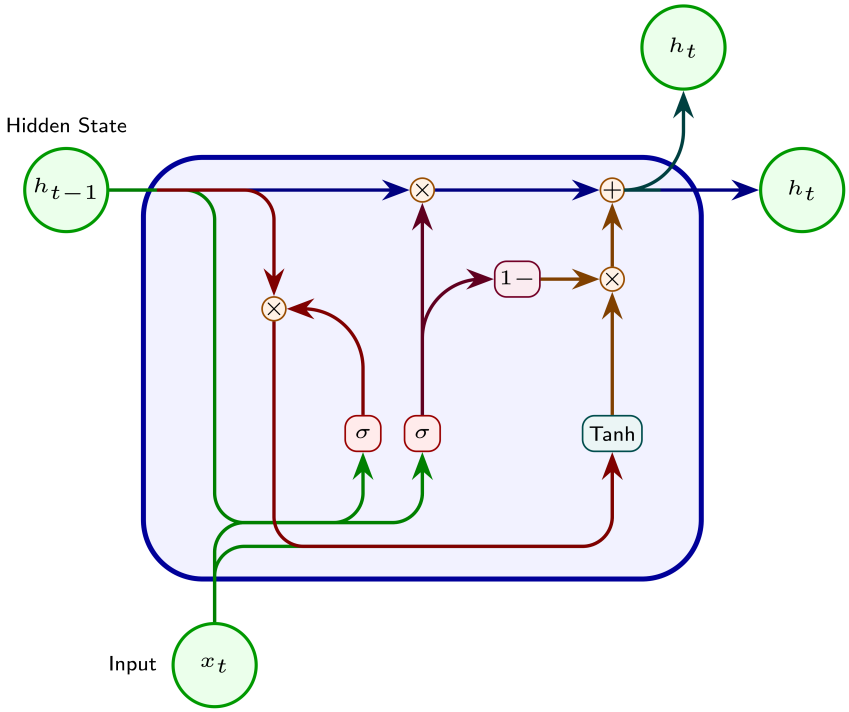


Figure 4: Illustration of the GRU gating mechanism.

3.3 Signature-Augmented Recurrent Models (SA-LSTM and SA-GRU)

This subsection introduces the signature-augmented recurrent architectures used in our forecasting framework. Intraday volatility often exhibits abrupt shifts, short-lived bursts, and irregular local patterns that are not well captured by conventional low-order lag structures. Path signatures offer a principled way to represent such behaviour by encoding the ordered fluctuations and interaction patterns of the price path through iterated integrals (Lyons, 1998; Friz et al., 2024). The expressive power of signatures is well established: they are universal, stable, and uniquely determine the underlying path under mild regularity (Geng, 2017; Boedihardjo et al., 2020). Recent studies also demonstrate the effectiveness of truncated signatures as feature representations for time-series modelling, especially when combined with suitable lifts such as the lead–lag transform (Fermanian, 2021; Kalsi et al., 2020).

We compute a truncated signature from a short history of the high-frequency log-price path, apply an attention layer that selects the most informative signature components, and incorporate the resulting descriptor into an LSTM or GRU. The resulting architectures are referred to as SA-LSTM and SA-GRU.

3.3.1 Path Signatures

To obtain non-trivial higher-order signature terms, the high-frequency log-price path is first represented through a canonical two-dimensional lift. This construction is standard in rough-path theory and corresponds to the minimal embedding that retains higher-order iterated integral information (Friz et al., 2024). In empirical applications, lifted paths have been shown to capture short-horizon dynamics and local irregularities in financial time series (**?**Li and Liu, 2022). Importantly, the lift in our setting is used solely for generating signature features; the forecasting model continues to rely only on realized volatility as the predictive input.

Consider an N-dimensional lifted path $X : [0, 1] \to \mathbb{R}^N$, written as

$$X(s) = \big(X^1(s), \ldots, X^N(s)\big).$$

The signature of X is a sequence of iterated integrals that records how the components of the path evolve jointly over time. The first level collects the net changes:

$$S(X)^n_{0,1} = \int_0^1 dX^n_s, \qquad n = 1, \ldots, N. \tag{33}$$

Higher levels summarise the ordered interactions between coordinates. The second level is

$$S(X)^{n,m}_{0,1} = \int_0^1 S(X)^n_{0,s}\, dX^m_s, \qquad n, m \in \{1, \ldots, N\}, \tag{34}$$

and the k-th level ($k \geq 2$) is defined recursively as

$$S(X)^{i_1,\ldots,i_k}_{0,1} = \int_0^1 S(X)^{i_1,\ldots,i_{k-1}}_{0,s}\, dX^{i_k}_s, \qquad i_j \in \{1, \ldots, N\}. \tag{35}$$

Collectively, these iterated integrals form the full signature

$$S(X)_{0,1} = \big(S^{(1)}(X),\ S^{(2)}(X),\ \ldots\big), \tag{36}$$

which is truncated at a finite depth M in empirical work:

$$S_{\leq M}(X)_{0,1} = \big(S^{(1)}(X), S^{(2)}(X), S^{(3)}(X), \ldots, S^{(M)}(X)\big). \tag{37}$$

If the lifted path has dimension N, the truncated signature contains

$$D_M = \sum_{k=1}^{M} N^k \tag{38}$$

components. A two-dimensional lift with depth $M = 3$ therefore yields $2 + 2^2 + 2^3 = 14$ signature terms (13 excluding the constant element).

From a modelling perspective, the truncated signature provides a compact summary of the recent high-frequency path. It captures direction, persistence, and cross-effects across coordinates,

which are often relevant for short-horizon volatility dynamics (Colmenarejo et al., 2020; Améndola et al., 2019; Galuppi, 2019). The truncation depth M determines the level of detail retained and is selected through validation.

3.3.2 Attention-Weighted Signature Descriptor

For each forecasting origin t, we extract a short intraday path X_t and compute its truncated signature:

$$s_t = S_{\leq M}(X_t) \in \mathbb{R}^{D_M}. \tag{39}$$

As the dimension of s_t grows with M, not all components are equally informative. An attention layer assigns data-driven weights to the signature channels.

Let W_a and b_a denote attention parameters. Raw scores are computed as

$$u_t = W_a s_t + b_a, \tag{40}$$

and transformed into normalised weights via a softmax:

$$\alpha_t = \mathrm{softmax}(u_t). \tag{41}$$

The attention-weighted descriptor is

$$\tilde{s}_t = \alpha_t \odot s_t, \tag{42}$$

with $\odot$ denoting the Hadamard product. The vector $\tilde{s}_t$ summarises the signature terms most relevant for forecasting at time t.

3.3.3 SA-LSTM and SA-GRU Architectures

All recurrent models in this study use the same baseline predictors: a sequence of p lagged log-realised volatilities,

$$x_t^{(\mathrm{lag})} = \left(RV_{t-1}, RV_{t-2}, \ldots, RV_{t-p}\right)^\top. \tag{43}$$

To incorporate the information extracted from the signature–attention module, each time step uses its own descriptor. For the $(t-j)$-th lag, let $\tilde{s}_{t-j}$ denote the attention-weighted signature computed from the corresponding prefix path. The augmented input matrix is therefore

$$Z_t = \begin{bmatrix} RV_{t-1} & \tilde{s}_{t-1}^\top \\ RV_{t-2} & \tilde{s}_{t-2}^\top \\ \vdots & \vdots \\ RV_{t-p} & \tilde{s}_{t-p}^\top \end{bmatrix} \in \mathbb{R}^{p \times (1+D_M)}. \tag{44}$$

Each row combines the realized volatility at a given lag with the descriptor summarising the path information available up to that lag. This construction allows the recurrent layer to condition its dynamics on both the local volatility history and the evolving signature-based representation of the underlying path.

This augmented input allows the recurrent network to process both the short-run volatility history and a global summary of the recent path.

SA-LSTM. Let $\mathrm{LSTM}_\phi(\cdot)$ denote an LSTM layer with parameters ϕ. The hidden state produced by the signature-augmented LSTM satisfies

$$h_t^{\text{SA-LSTM}} = \mathrm{LSTM}_\phi(Z_t), \tag{45}$$

and the one-step-ahead forecast is

$$\widehat{RV}_t = w^\top h_t^{\text{SA-LSTM}} + b. \tag{46}$$

SA-GRU. Replacing the LSTM layer with a GRU cell yields the SA-GRU architecture. Let $\mathrm{GRU}_\psi(\cdot)$ denote a GRU layer with parameters ψ. The hidden state satisfies

$$h_t^{\text{SA-GRU}} = \mathrm{GRU}_\psi(Z_t), \tag{47}$$

and the forecast is

$$\widehat{RV}_t = w^\top h_t^{\text{SA-GRU}} + b. \tag{48}$$

Both SA-LSTM and SA-GRU employ the same signature-attention module and differ only in the recurrent update rule.

3.3.4 Interpretation

The signature-augmented architecture supplements the recurrent network with a structured summary of the recent intraday path. Signatures capture high-order ordered variations that cannot be recovered from low-order lags alone (Friz et al., 2024; Fermanian, 2021). The attention mechanism extracts the components most relevant for short-horizon volatility prediction, while the recurrent layer processes both local fluctuations and global geometric information. This design enables the model to remain sensitive to abrupt intraday changes, nonlinear interactions, and the short-term geometric structure of the price path.

A schematic illustration of the architecture is provided in Figure 5.

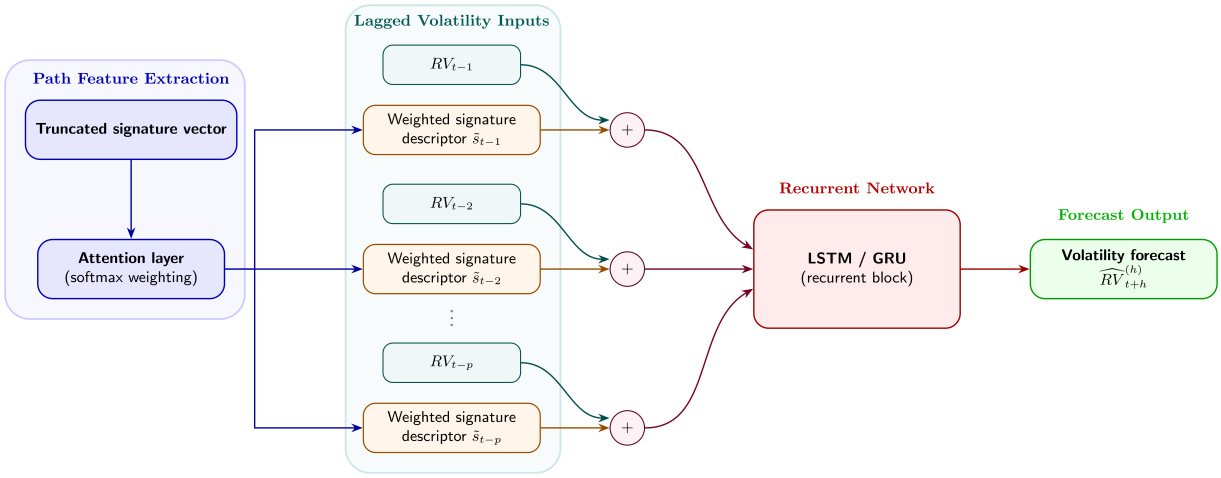


Figure 5: Illustration of the signature attention recurrent architecture.

3.4 Multi-Scale–Augmented Recurrent Models (MSA-LSTM and MSA-GRU)

This subsection presents the multi-scale–augmented recurrent architectures (MSA-LSTM and MSA-GRU). The motivation is straightforward: realized volatility evolves at multiple horizons, and a model based only on short lags cannot disentangle these layers (Andersen et al., 2003; Corsi, 2009; Baruník and Křehlík, 2018). The MSA framework extracts short-, medium-, and longer-run components of past volatility using a small set of causal convolutions defined over fixed calendar windows, and forms a compact descriptor that is concatenated with the usual lagged-volatility sequence before entering the recurrent layer. This multi-horizon construction is broadly consistent with the philosophy behind the HAR model, which also aggregates volatility over several calendar windows, though our implementation is substantially more flexible: the filters are learned rather than fixed, and the final mapping to the forecast is carried out by an LSTM or GRU rather than a linear regression.

3.4.1 Multi-Scale Volatility Structure and Motivation

Empirical evidence suggests that high-frequency realized volatility combines several components operating at different horizons. (Andersen et al., 2003; Corsi, 2009). Very short-run movements reflect microstructural noise and transient order-flow shocks; intermediate horizons capture intraday clustering and spillovers across adjacent days; longer horizons are driven by more persistent forces such as macroeconomic conditions and risk premia. A specification based purely on a small number of lags treats all horizons symmetrically and may fail to separate these layers.

The MSA architecture is designed to make this horizon structure explicit. Instead of relying on a single lag polynomial, we introduce a few convolutional filters whose effective lengths correspond to fixed calendar horizons of three, five, and ten trading days. These filters are applied to the high-frequency realised-volatility sequence in a causal way, using only past information, and an attention mechanism then learns how much weight to place on each horizon at a given forecasting

origin. The recurrent layer therefore conditions its dynamics on an adaptive combination of short-, medium-, and longer-run volatility summaries.

3.4.2 Multi-Scale Convolutional Features

The MSA models enrich the baseline lagged-volatility input by applying a small set of one-dimensional causal convolutions to the past realised-volatility sequence. Each convolution uses a kernel whose length corresponds to a fixed calendar horizon, so that the resulting feature sequences capture short-, medium-, and longer-run patterns in past volatility, inspired by Borovykh et al. (2018); Chen et al. (2020); Moreno-Pino and Zohren (2024b).

We consider three horizons: three days, five days, and ten days. Because the number of intraday observations per day differs across sampling frequencies, the effective kernel length depends on the forecasting interval. For instance, forecasting ten-minute realized volatility involves 24 intraday observations per day, so the three convolutional kernels span 3×24, 5×24, and 10×24 time steps. At the thirty-minute frequency the corresponding kernel lengths become 3×8, 5×8, and 10×8; at the sixty-minute frequency they become 3×4, 5×4, and 10×4; and at the daily frequency they reduce directly to $\{3, 5, 10\}$ lags.

Let Conv_H denote the convolution associated with horizon $H \in \{3, 5, 10\}$. Applying Conv_H to the lagged realised-volatility vector produces a horizon-specific feature sequence

$$h_t^{(H)} = \mathrm{Conv}_H\left(x_t^{(\mathrm{lag})}\right) \in \mathbb{R}^p. \tag{49}$$

The convolutions are implemented in a strictly causal manner, so that each entry of $h_t^{(H)}$ depends only on information available at or before the corresponding lag. This guarantees that the extracted multi-scale features conform to the forecasting information set and do not introduce look-ahead bias.

3.4.3 Scale-Attention Descriptor

The three sequences $\{h_t^{(3)}, h_t^{(5)}, h_t^{(10)}\}$ provide parallel views of the recent volatility history at different horizons. Their relative importance need not be constant over time. To allow the model to adjust the emphasis placed on each horizon, we combine these sequences through an attention mechanism operating across scales(Tran et al., 2018; Yang et al., 2021; Chatigny et al., 2021).

For each horizon $H \in \{3, 5, 10\}$ we construct a scalar score $u_t^{(H)}$ that summarises the information in $h_t^{(H)}$,

$$u_t^{(H)} = w^\top \tanh\left(W h_t^{(H)} + b\right), \qquad H \in \{3, 5, 10\}, \tag{50}$$

where W, w, and b are attention parameters shared across horizons. These scores are mapped into normalised horizon weights via a softmax transformation,

$$\alpha_t^{(H)} = \frac{\exp\left(u_t^{(H)}\right)}{\sum_{H' \in \{3,5,10\}} \exp\left(u_t^{(H')}\right)}, \qquad H \in \{3, 5, 10\}. \tag{51}$$

The weights $\alpha_t^{(H)}$ indicate how much the model relies on each horizon at time t.

Using these weights, we form a single multi-scale feature sequence $h_t^{(\mathrm{ms})} \in \mathbb{R}^p$ that is aligned with the lagged volatility sequence,

$$h_t^{(\mathrm{ms})} = \sum_{H \in \{3,5,10\}} \alpha_t^{(H)} \, h_t^{(H)}. \tag{52}$$

The sequence $h_t^{(\mathrm{ms})}$ plays the same role in the MSA models as the attention-weighted signature descriptor $\tilde{s}_t$ does in the SA models: it provides a compact, data-driven summary of higher-order information, here in the form of multi-horizon volatility filters rather than path-signature terms.

3.4.4 MSA-LSTM and MSA-GRU Architectures

To feed the multi-scale descriptor into the recurrent network, we concatenate $h_t^{(\mathrm{ms})}$ to the baseline lag sequence $x_t^{(\mathrm{lag})}$ on a per-lag basis. For forecasting origin t, the augmented input matrix is defined as

$$Z_t = \begin{bmatrix} RV_{t-1} & h_{t,1}^{(\mathrm{ms})} \\ RV_{t-2} & h_{t,2}^{(\mathrm{ms})} \\ \vdots & \vdots \\ RV_{t-p} & h_{t,p}^{(\mathrm{ms})} \end{bmatrix} \in \mathbb{R}^{p \times 2}. \tag{53}$$

Each row of Z_t contains the realized volatility at a given lag together with the corresponding multi-scale summary of volatility around that lag. This construction mirrors the SA input matrix, where each lag is paired with its own descriptor computed from the relevant path prefix.

Let $\mathrm{LSTM}_\varphi(\cdot)$ denote an LSTM layer with parameters φ, and let $\mathrm{GRU}_\psi(\cdot)$ denote a GRU layer with parameters ψ. The hidden state produced by the MSA-LSTM satisfies

$$h_t^{\text{MSA-LSTM}} = \mathrm{LSTM}_\varphi(Z_t), \tag{54}$$

and the corresponding one-step-ahead forecast is

$$\widehat{RV}_t = w^\top h_t^{\text{MSA-LSTM}} + b, \tag{55}$$

for some output weights w and bias b. Replacing the LSTM layer with a GRU cell yields the MSA-GRU architecture,

$$h_t^{\text{MSA-GRU}} = \mathrm{GRU}_\psi(Z_t), \tag{56}$$

with forecast

$$\widehat{RV}_t = w^\top h_t^{\text{MSA-GRU}} + b. \tag{57}$$

Thus, SA and MSA models share the same recurrent and output layers and differ only in the way the auxiliary descriptor entering Z_t is constructed.

3.4.5 Interpretation, Relation to SA Models, and Link to HAR

The MSA architecture augments the recurrent network with an explicit decomposition of volatility history across a small set of fixed calendar horizons. The convolutional filters behave like data-driven horizon-specific components: short-horizon filters respond primarily to very recent volatility shocks and microstructural disturbances, medium-horizon filters track clustering over a handful of days, and long-horizon filters capture more persistent movements. The attention mechanism then determines, at each forecasting origin, which of these horizons is most informative for predicting the next-period realized volatility. In this way, the model separates the volatility signal into coarse horizon layers before passing it to the LSTM or GRU.

A useful benchmark for interpreting the MSA design is the heterogeneous autoregressive (HAR) model for realized volatility. In the HAR specification, future volatility is modelled as a linear combination of averages over past daily, weekly, and monthly horizons, for example one-day, five-day, and twenty-one-day realized volatility (see also Clements and Preve, 2021). When the forecasting interval is 240 minutes (one observation per trading day in our setting), the three MSA convolutional kernels span three, five, and ten trading days and thus resemble rolling windows over the recent past. The resulting filtered sequences can be viewed as flexible counterparts of short-, medium-, and longer-horizon volatility components. However, in contrast to the HAR model, the weights within each window are learned rather than fixed to a simple arithmetic average, and the final mapping from these components to the forecast is performed by an LSTM or GRU instead of a linear regression. From this perspective, the 240-minute MSA specification can be interpreted as a nonlinear, data-driven extension of the classical HAR framework (see Corsi, 2009).

Compared with the SA models, which encode fine structure in high-frequency volatility paths through signature-based features, the MSA models operate on a simpler input (realized volatility alone) and organise information along the time-scale dimension. Both families implement the same high-level idea: enrich the baseline lag sequence with a structured, learned summary of richer dynamics, then let an LSTM or GRU exploit this augmented input. In empirical applications, the two approaches are expected to be complementary. The SA models are well suited to capturing irregular local patterns in very high-frequency data, whereas the MSA models provide a transparent way to account for multi-horizon persistence and offer a natural bridge to established econometric models such as HAR while allowing for nonlinearity and richer dynamics(Moreno-Pino and Zohren, 2024b; Yang et al., 2021; Chatigny et al., 2021).

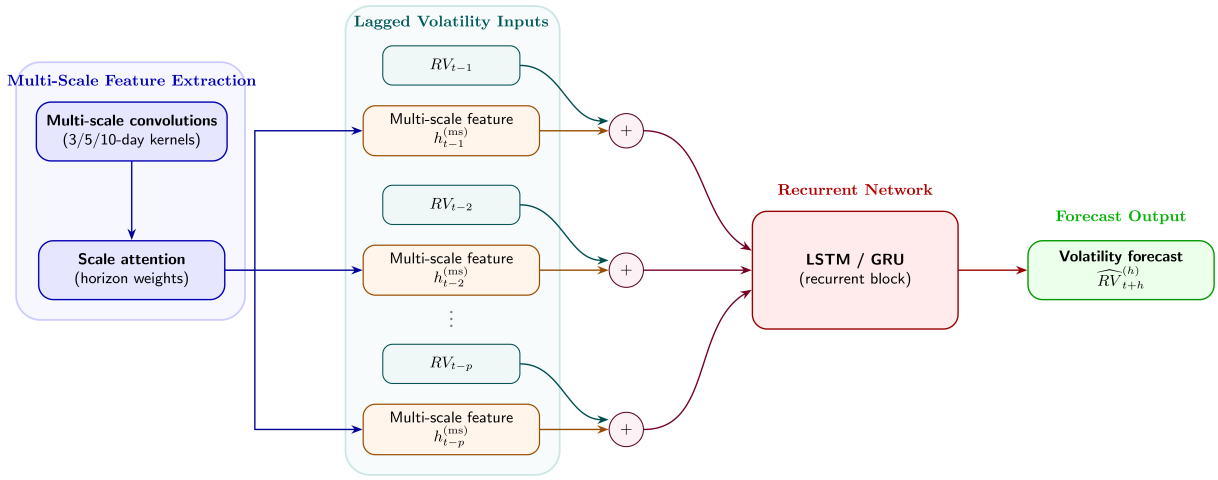


Figure 6: Illustration of the multi-scale attention recurrent architecture.

3.5 Training Schemes

Forecasting models are estimated under four training schemes that differ in how information is shared across assets and how cross-sectional structure is incorporated. These schemes reflect standard approaches in multi-asset prediction, including single-task models, pooled estimation, augmented specifications, and cluster-specific learning. Related methodologies have been widely used in panel data machine learning, multi-task learning, and grouped forecasting models (Gu et al., 2020; **?**; Gajamannage et al., 2023; Shu et al., 2025). Our first three schemes—Single, Universal, and Augmented—follow the terminology of Zhang et al. (2024a), while the fourth scheme extends their framework by exploiting GICS-based industry structure.

3.5.1 Single (Stock-Specific) Training

The Single scheme estimates a separate model for each stock. Parameters are fully asset-specific, and the network is trained exclusively on the historical series of that stock:

$$\widehat{RV}_{i,t+h}^{(h)} = f_{\theta_i}\left(\mathbf{r}_{i,t}^{(h)}\right). \tag{58}$$

This approach captures stock-level idiosyncrasies but ignores systematic comovement and reduces the effective sample size for each model. It is analogous to single-task learning and asset-by-asset estimation used in sequential forecasting studies such as Gajamannage et al. (2023).

3.5.2 Universal (Fully Pooled) Training

The Universal scheme pools all stocks into a single large training set and estimates one model with a shared parameter vector:

$$\widehat{RV}_{i,t+h}^{(h)} = f_{\theta}\left(\mathbf{r}_{i,t}^{(h)}\right). \tag{59}$$

This pooled estimator exploits cross-sectional information and stabilizes parameter estimates, consistent with panel-data machine learning and pooled estimation frameworks (Park et al., 2022). It corresponds to the Universal scheme in Zhang et al. (2024a).

3.5.3 Augmented (Pooled + Market Volatility Feature)

The Augmented scheme retains the fully pooled structure but adds an additional predictor capturing aggregate market volatility:

$$\text{MarketRV}_t^{(h)} = \frac{1}{N}\sum_{i=1}^{N} RV_{i,t}^{(h)}. \tag{60}$$

The forecasting model becomes:

$$\widehat{RV}_{i,t+h}^{(h)} = f_\theta\left(\mathbf{r}_{i,t}^{(h)}, \text{MarketRV}_t^{(h)}\right). \tag{61}$$

This scheme incorporates a cross-sectional factor summarizing market-wide volatility conditions. It mirrors enhanced pooled models in cross-asset forecasting (e.g., Shu et al. (2025)) and corresponds to the Augmented scheme of Zhang et al. (2024a).

3.5.4 Cluster (Group-Pooled + ClusterRV Feature)

To exploit sector-level comovement while preserving cross-sectional heterogeneity, we introduce a cluster-based training scheme. Stocks are grouped according to the Global Industry Classification Standard (GICS), producing economically meaningful clusters. Table 1 provides the sector composition of the sample. Within each cluster g, stocks share the same parameter vector:

$$\widehat{RV}_{i,t+h}^{(h)} = f_{\theta_{g(i)}}\left(\mathbf{r}_{i,t}^{(h)}, \text{ClusterRV}_{g(i),t}^{(h)}\right), \tag{62}$$

where the cluster-level realized volatility is defined as

$$\text{ClusterRV}_{g,t}^{(h)} = \frac{1}{|g|}\sum_{i\in g} RV_{i,t}^{(h)}. \tag{63}$$

This specification combines partial pooling (shared parameters within clusters) with cluster-specific augmentation. It extends the pooled augmented framework of Zhang et al. (2024a) and aligns with cluster-specific and multi-task learning methods in the forecasting literature (Park et al., 2022). Using GICS sectors ensures that the clusters reflect well-established industry-level volatility commonality.

Remark. The focus of this subsection is on the cross-sectional structure of parameter sharing. Details regarding loss functions, optimization, early stopping, hyperparameter ranges, and practical implementation are reported in Appendix A. These procedures are identical across all four training schemes unless otherwise noted.

4 Data Description

4.1 Data Sources and Sampling

The empirical analysis uses minute-level Level-1 data from the Wind database. The sample covers all regular trading sessions of the Shanghai and Shenzhen A-share markets from 2 January 2019 to 31 December 2024. This period spans several major market events, including the COVID-19 outbreak, episodes of U.S.–China trade tensions, the Russia–Ukraine conflict, and subsequent domestic policy-driven cycles. providing a wide range of volatility conditions.

The stock universe is based on the constituents of the CSI 300 index as of December 2018, a date prior to the start of the forecasting period. The CSI 300 is a free-float market-capitalisation-weighted benchmark composed of large and liquid A-share companies. Using the pre-sample index composition avoids look-ahead concerns and ensures that the stock selection does not rely on information from the evaluation window. Among the 2018 constituents, we retain stocks with complete minute-level records throughout Jan 2 2019–Dec 31 2024. Within this set of stocks, we sort by the 2018 index weights and select the top 100 names. This approach emphasises firms with high liquidity and stable intraday trading activity, reducing problems caused by thin trading or long stretches of zero returns.

For each stock, the dataset provides open, high, low, and close transaction prices at the one-minute frequency. The open is the first trade within the minute; the high and low are the highest and lowest transaction prices; and the close is the last trade of the minute. We also observe traded volume, traded value (in CNY), and the best bid and ask quotes together with their posted depths.

The Chinese A-share market follows a split-session structure. An opening call auction is conducted from 9:15 to 9:25, followed by continuous trading from 9:30 to 11:30 and from 13:00 to 15:00, and a short closing call auction at the end of the session. The four-hour continuous trading window determines the number of intraday observations available for constructing realised-volatility measures. Based on the minute-by-minute prices, we form non-overlapping realised-volatility horizons of 10, 30, 60, and 240 minutes, corresponding to 24, 8, 4, and 1 intervals per trading day.

Data cleaning follows standard practice in high-frequency empirical work. We remove non-trading days and full-day suspensions. Minute timestamps are aligned across stocks to form a common grid. Minutes without transactions are filled using the last available price so that log returns can be computed on an evenly spaced series. Obvious data errors, such as zero prices or out-of-range quotes, are discarded.

Sector classification follows the first-level GICS sectors. For the stocks in our sample, the GICS mapping coincides with the first-level CSI industry classifications. Table 1 reports the sector distribution of the 100 selected stocks.

Table 1: Industry Distribution of the Sample Stocks (GICS Classification)

Industry	Count	Tickers
Communication Services	2	SH.600050, SZ.002027
Consumer Discretionary	8	SH.600104, SH.600660, SH.600690, SH.600741, SH.601888, SZ.000333, SZ.000651, SZ.002594
Consumer Staples	8	SH.600519, SH.600887, SH.601933, SH.603288, SZ.000568, SZ.000858, SZ.000895, SZ.002304
Energy	4	SH.600028, SH.601088, SH.601225, SH.601857
Financials	26	SH.600000, SH.600015, SH.600016, SH.600036, SH.600837, SH.600919, SH.600999, SH.601009, SH.601166, SH.601169, SH.601229, SH.601288, SH.601318, SH.601328, SH.601336, SH.601398, SH.601601, SH.601628, SH.601688, SH.601818, SH.601939, SH.601988, SZ.000001, SZ.000166, SZ.000776, SZ.300059
Health Care	8	SH.600196, SH.600276, SH.600436, SZ.000538, SZ.000661, SZ.300003, SZ.300015, SZ.300142
Industrials	16	SH.600029, SH.600031, SH.600089, SH.600406, SH.601006, SH.601111, SH.601186, SH.601390, SH.601668, SH.601669, SH.601766, SH.601800, SH.601989, SZ.000338, SZ.002008, SZ.002202
Information Technology	11	SH.600271, SH.600487, SH.600570, SH.600703, SH.601012, SZ.000100, SZ.000725, SZ.002044, SZ.002230, SZ.002415, SZ.002475
Materials	9	SH.600010, SH.600019, SH.600309, SH.600352, SH.600547, SH.600585, SH.601600, SH.601899, SH.603993
Real Estate	4	SH.600048, SH.600606, SZ.000002, SZ.000069
Utilities	4	SH.600011, SH.600795, SH.600886, SH.601985

4.2 Realized Volatility Construction

The realized volatility measures used in the empirical analysis follow the notation and framework introduced in Section 3.1. For each stock i and horizon $h \in \{10, 30, 60, 240\}$ minutes, log-realised variance $RV_{i,t}^{(h)}$ is obtained from non-overlapping intraday intervals by summing squared high-frequency returns and applying a logarithmic transformation to stabilise the distribution. This construction is standard in the realised-volatility literature and ensures that the discrete-time proxy is a consistent estimator of the underlying integrated variance under mild regularity conditions (Barndorff-Nielsen and Shephard, 2002; Andersen et al., 2003, 2001b). Throughout the paper we work with $\{RV_{i,t}^{(h)}\}$ as the volatility target and as the main input to the forecasting models, and use the same notation when forming market- and cluster-level aggregates defined in Section 3.1.

To illustrate the realised-volatility measures introduced above, Figure 7 plots the cross-sectional

mean of log-realized volatility at the 10-, 30-, 60-, and 240-minute horizons. Grey vertical bands mark days when the daily realized volatility exceeds its 90th percentile, and the grey dashed line shows a ten-observation moving average that highlights lower-frequency movements.

Three features stand out. First, major volatility spikes appear at almost identical dates across all horizons, implying that these surges reflect broad market-level shocks rather than horizon-specific noise. Second, shorter-horizon series exhibit sharper and more irregular fluctuations, while the daily horizon produces a smoother profile in which the same episodes unfold more gradually. Third, despite these scale differences, the underlying trend is nearly the same across horizons, confirming that $RV_{i,t}^{(h)}$ at different frequencies are coherent aggregations of the same volatility process.

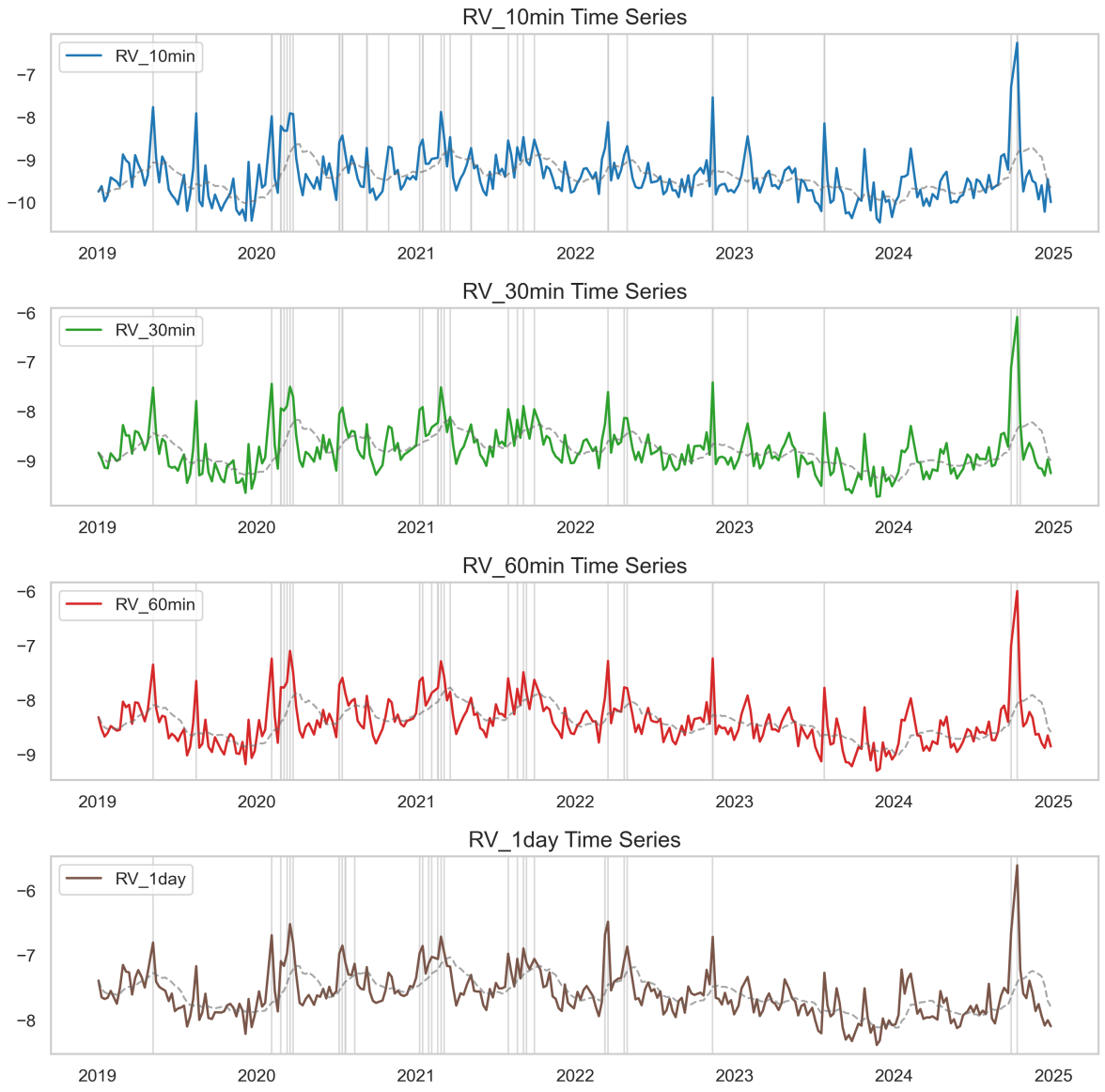


Figure 7: Time series of log-realized volatility at the 10-, 30-, 60-, and 240-minute horizons. Each panel reports the cross-sectional mean of $RV_{i,t}^{(h)}$. Grey vertical bands indicate days for which the daily realized volatility exceeds its 90th percentile. The grey dashed line shows a ten-observation moving average. Across all horizons, major volatility spikes occur at the same dates, with shorter horizons exhibiting more jagged fluctuations and the daily horizon displaying a smoother profile.

4.3 Descriptive Evidence and Volatility Commonality

To summarise the main empirical features of intraday volatility, this subsection reports evidence on cross-sectional correlation patterns, the time-series behaviour of daily volatility, and diurnal intraday profiles. Before constructing these statistics, we apply a mild winsorisation to returns and realised volatilities at the stock level. Following Zhang et al. (2024a), observations below the 0.5th percentile of a given series are set to that percentile, and observations above the 99.5th percentile are truncated at the 99.5th percentile. This procedure reduces the influence of extreme values and

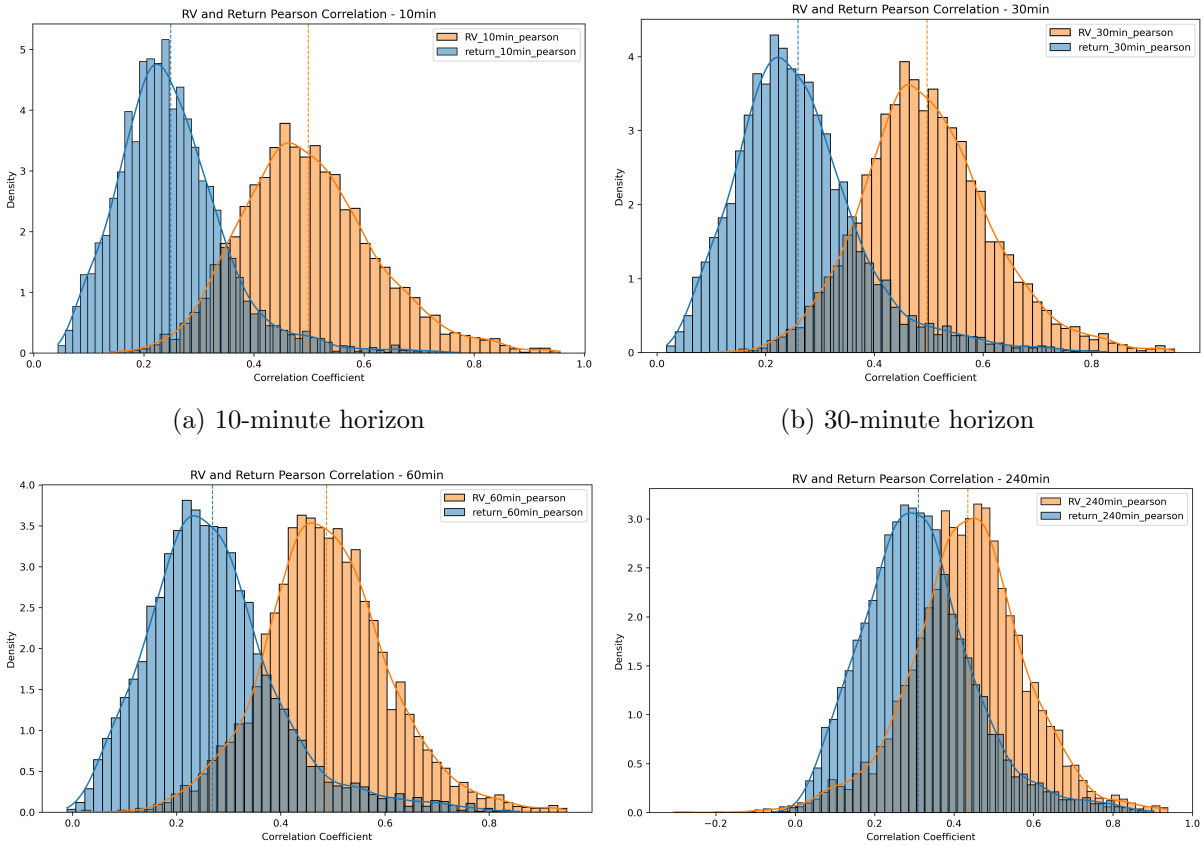
data errors while preserving the overall shape of the distributions.

4.3.1 Cross-Sectional Correlation Patterns

Figure 8 reports the cross-sectional Pearson correlations of returns and realized volatility for horizons $h \in \{10, 30, 60, 240\}$ minutes. For each day and horizon, we compute all pairwise correlations across the 100 sample stocks, separately for log-returns and log-realized volatility, and pool these values over time to form the empirical distributions shown in the figure. The orange histograms correspond to $RV_{i,t}^{(h)}$, while the blue histograms correspond to returns.

Two features stand out. First, the cross-sectional correlation of realized volatility is substantially higher than that of returns at all horizons. For the 10-minute horizon, for example, the mean (median) pairwise correlation of realized volatility is about 0.50 (0.49), whereas the corresponding values for returns are around 0.25 (0.24). Similar gaps arise at 30 and 60 minutes, and even at the daily horizon the mean volatility correlation of about 0.44 still exceeds the mean return correlation of roughly 0.31. These numbers indicate that volatility movements are much more synchronised across stocks than price changes themselves, consistent with the interpretation of volatility as a common risk factor that loads on many assets simultaneously.

Second, the degree of commonality is strong and remarkably stable across intraday horizons. For $h = 10, 30$, and 60 minutes, the mean correlations for realized volatility cluster tightly around 0.49–0.50, while the daily horizon displays a moderate decline to the mid-0.4 range, likely reflecting the influence of overnight information and non-trading hours. By contrast, the cross-sectional correlations of returns rise more noticeably with the horizon, from a mean of about 0.25 at 10 minutes to roughly 0.31 at the daily frequency, as idiosyncratic noise is averaged out. Taken together, these patterns show that volatility exhibits strong and pervasive comovement across the universe of large A-share stocks, whereas returns remain only weakly correlated. The evidence closely parallels the findings of Zhang et al. (2024a) for U.S. equities, suggesting that volatility commonality is a robust feature shared by both markets. This pronounced cross-sectional dependence motivates the use of market- and cluster-level volatility aggregates in the forecasting models and underpins the design of the cluster-based training scheme introduced later in Section 3.5.



(a) 10-minute horizon

(b) 30-minute horizon

(c) 60-minute horizon

(d) 240-minute (daily) horizon

Figure 8: Cross-sectional Pearson correlations of returns and realized volatility at different intraday horizons. The figure plots the empirical distributions of pairwise correlations across the 100 sample stocks for log-returns (blue) and log-realized volatility (orange), after winsorising each series at the 0.5th and 99.5th percentiles at the stock level.

4.3.2 Time-Series Behaviour of Daily Volatility

Figure 9 summarises the time-series evolution of daily realized volatility over the sample period together with the cumulative market return. The solid line shows the cross-sectional average of log-realized volatility at the daily horizon ($h = 240$ minutes), while the shaded bands report the 25th–75th and 5th–95th cross-sectional percentiles. A second vertical axis displays the cumulative return on a value-weighted portfolio formed from the 100 sample stocks, where weights are based on pre-sample CSI 300 constituents and normalised to sum to one.

High-volatility episodes, highlighted by the grey shaded regions, are defined as days on which the cross-sectional mean of daily realized volatility exceeds its 90th percentile. These episodes exhibit pronounced clustering: once the market enters a high-volatility state, realized volatility tends to remain elevated for a prolonged period. The shaded intervals almost entirely coincide with major macroeconomic and policy events, including the onset of the COVID-19 pandemic, the escalation of the Russia–Ukraine conflict, and several phases of domestic policy tightening or stimulus. During these windows, the cumulative market return swings sharply, with either steep drawdowns or rapid

rebounds, underscoring the close connection between volatility spikes and large price movements. Outside such episodes, both the level and dispersion of log-realized volatility remain relatively subdued, indicating that the cross-section of stocks tends to move in a more orderly fashion in tranquil times.

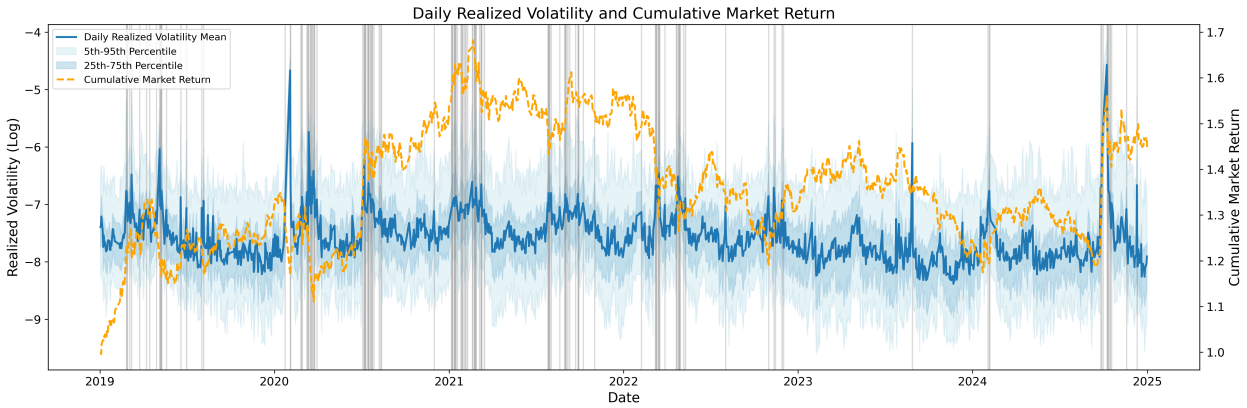


Figure 9: Daily realized volatility and cumulative market return. The solid line plots the cross-sectional mean of daily log-realized volatility, with the dark (light) shaded areas indicating the 25th–75th (5th–95th) cross-sectional percentiles. Grey vertical bands mark days on which the mean daily realized volatility exceeds its 90th percentile. The dashed line reports the cumulative return on a value-weighted portfolio of the 100 sample stocks (right axis), with weights proportional to their CSI 300 index weights at the end of 2018.

4.3.3 Intraday Diurnal Patterns

Finally, Figure 10 depicts the average intraday profile of log-realized volatility at the 10-minute horizon, together with the corresponding trading volume pattern. For each 10-minute interval within the continuous trading session, we compute the cross-sectional mean of $RV_{i,t}^{(10)}$ and its interquartile range, and then average these statistics across all trading days in the sample. The right axis reports the mean total trading volume (in shares) aggregated across all sample stocks for each intraday time bucket.

The diurnal volatility pattern exhibits a pronounced "high-at-open, low-at-midday, flat-towards-close" structure. Volatility is highest immediately after the market opens, declines steadily throughout the morning, and reaches its minimum around the midday break, before rising slightly at the reopening and stabilising towards the close. Trading volume follows a similar shape, with spikes at the open and, to a lesser extent, near the end of the afternoon session, indicating that periods of intense trading activity coincide with elevated volatility. This intraday profile reflects key institutional features of the Chinese A-share market, including the T+1 trading rule, price limits, and the split-session trading schedule with a long midday recess(Tian and Guo, 2007; Zhang et al., 2019; Chen et al., 2025). Compared with the U-shaped intraday patterns often reported for major developed markets, the A-share market displays a more front-loaded volatility structure, with relatively high activity concentrated in the first part of the trading day(Guo, 2006; Zhang et al., 2024a).

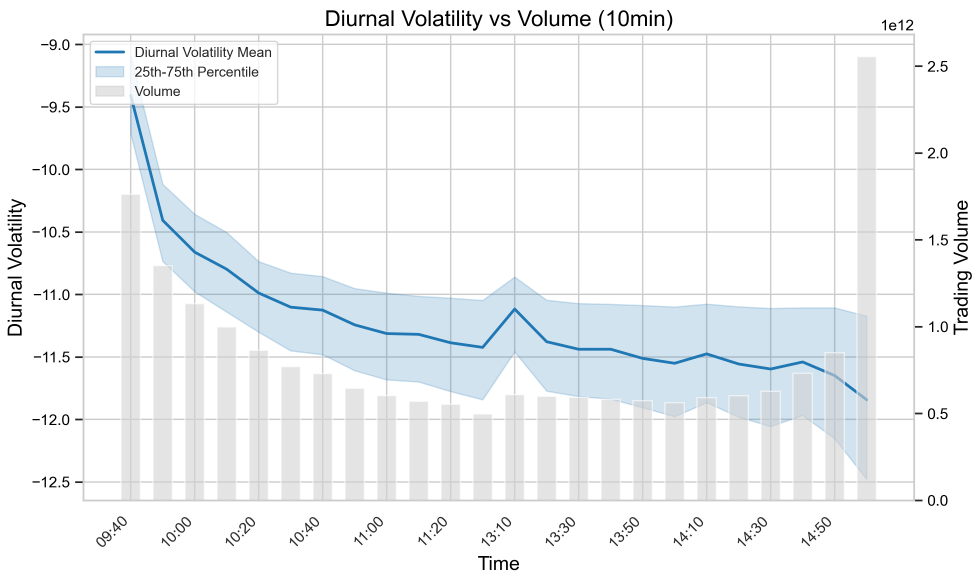


Figure 10: Intraday pattern of log-realized volatility and trading volume (10-minute horizon). The solid line shows the average cross-sectional mean of 10-minute log-realized volatility across all trading days, with the shaded band indicating the interquartile range. The grey bars represent the average total trading volume (in shares) across the 100 sample stocks for each 10-minute interval (right axis).

4.3.4 Commonality Estimation

The patterns documented above indicate that a substantial part of intraday volatility is driven by common forces rather than purely idiosyncratic shocks. To quantify the extent to which a single aggregate factor explains the cross-section of volatility, we follow the approach used in recent empirical studies and measure commonality through the explanatory power of a simple market-level realised-volatility index (Dang et al., 2015; Mumtaz and Theodoridis, 2017; Zhang et al., 2020a; Qiu et al., 2020; Zhang et al., 2024a). For each stock i and horizon h, we estimate the linear specification

$$RV_{i,t}^{(h)} = \alpha_i^{(h)} + \beta_i^{(h)} RV_{M,t}^{(h)} + \varepsilon_{i,t}^{(h)}, \tag{64}$$

where $RV_{M,t}^{(h)}$ denotes the equal-weighted average realized volatility across all stocks in the sample. The adjusted R^2 from (64) summarises how much of each stock's volatility is accounted for by the market component, and the cross-sectional average of these adjusted R^2 statistics serves as our measure of volatility commonality at horizon h. This metric is more informative than simple pairwise correlations, as it captures the proportion of systematic variation attributable to a single common factor.

Although the discussion in this subsection focuses on market-level commonality, an analogous measure can be constructed using cluster-level realised-volatility indices formed from the GICS sector groups. We mention this extension only briefly here, as the cluster-based commonality statistics play an important role later in Section 6 when analysing why the cluster-based training scheme improves forecasting performance and how group-level risk factors shape cross-sectional

volatility dynamics.

Figure 11 presents the time-series evolution of average adjusted R^2 for the four intraday horizons examined in this study. Three empirical regularities stand out.

First, commonality is materially stronger at higher intraday frequencies. Across the full sample, the 60-minute horizon exhibits the highest level of commonality, followed by the 30-minute and 10-minute horizons, while the daily (240-minute) measure is noticeably lower. This ordering is consistent with evidence reported for U.S. equities, indicating that the structure of intraday volatility in the A-share market shares important similarities with that of developed markets (Zhang et al., 2024a).

Second, the four horizons display remarkably similar dynamics over time. Periods of market stress—such as the onset of the COVID-19 pandemic, episodes of heightened geopolitical tension, and several phases of domestic policy adjustments—coincide with sharp increases in commonality across all horizons. This synchronous rise indicates that during turbulent episodes, a larger share of volatility is driven by pervasive system-wide shocks rather than firm-specific events (see Wang et al., 2020; Zhang et al., 2022).

Third, the daily (240-minute) commonality series is both lower in level and more volatile than its intraday counterparts. This behaviour is consistent with the influence of overnight information, discrete macro announcements, and non-trading hours, all of which limit the explanatory power of a contemporaneous market-level volatility index (see Liang et al., 2020; Wang et al., 2020).

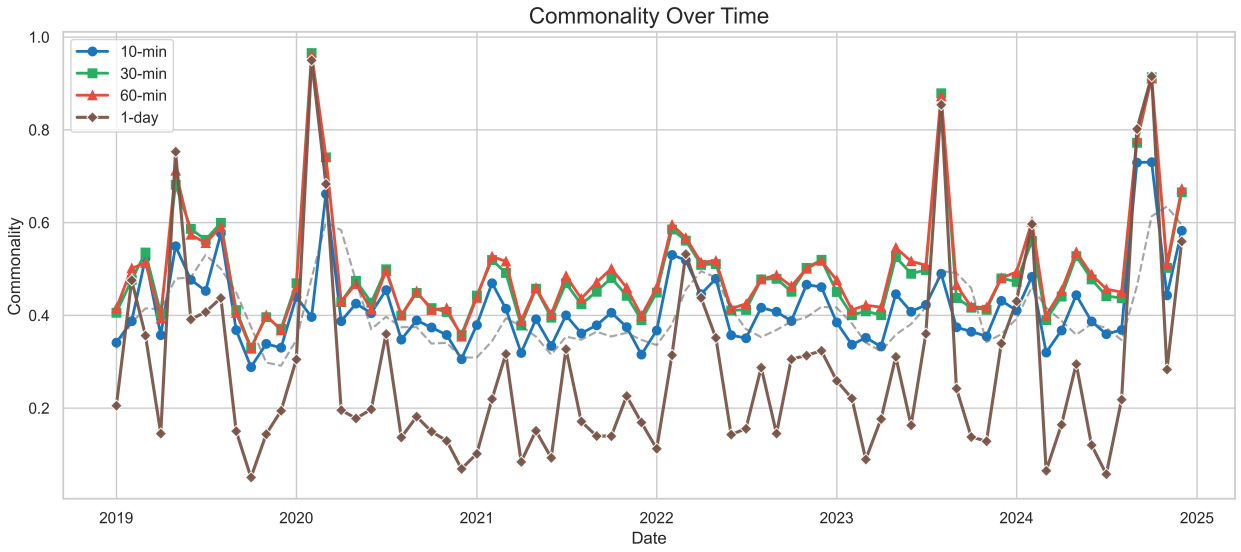


Figure 11: Monthly commonality in realized volatility across the four intraday horizons. The figure plots the cross-sectional average adjusted R^2 obtained from regressing each stock's realized volatility on the corresponding market-level measure. Commonality is substantially higher at intraday frequencies than at the daily horizon and is largest for the 60-minute interval. All horizons display similar time variation, with sharp increases during major market stress episodes, indicating that a larger fraction of volatility is driven by systematic forces in turbulent periods.

Figure 12 reports the intraday behaviour of commonality for the 30-minute horizon. A distinct

pattern emerges: commonality peaks at the market open and declines sharply thereafter, remaining relatively low for the rest of the trading day. This front-loaded structure contrasts with the more monotonic rise toward the close reported for developed markets and highlights the institutional features of the Chinese A-share market (see Zhang et al., 2024a). At the open, overnight macroeconomic news, policy signals, and global market movements are incorporated simultaneously into prices across many stocks, generating a surge in synchronised volatility. The high participation of retail investors further contributes to this synchronous response, as order submissions tend to cluster at the start of the session. Once the initial burst of information is absorbed, volatility becomes less coordinated across firms, and commonality stabilises at a lower level (see Huo and Ahmed, 2017; Wang et al., 2020).

Taken together, these results indicate that volatility in the A-share market contains a sizeable and time-varying common component. This evidence provides a natural motivation for incorporating market- and group-level structure into our forecasting framework and lays the empirical foundation for the cluster-based training scheme analysed later in Section 6.

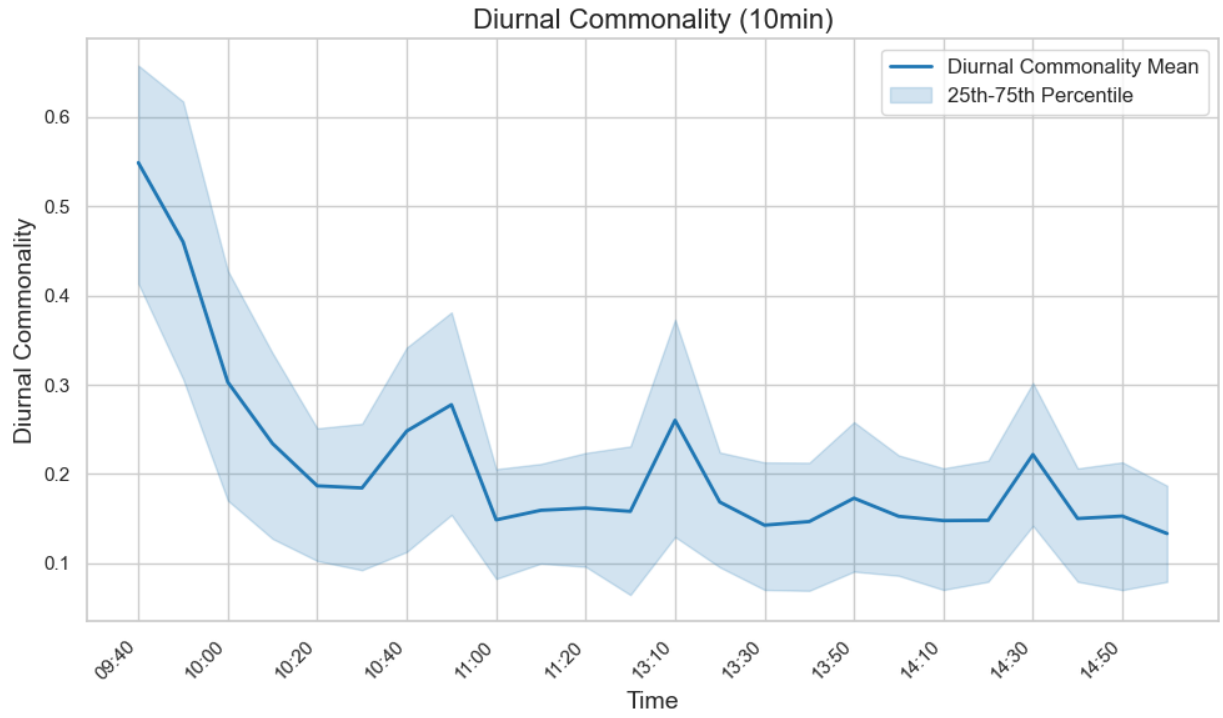


Figure 12: Intraday pattern of volatility commonality at the 10-minute horizon. For each intraday time bucket, the figure reports the cross-sectional average adjusted R^2 from regressions of stock-level realized volatility on the market-level index. Commonality peaks at the market open and declines sharply thereafter, reflecting the synchronous incorporation of overnight information and the concentrated trading activity at the start of the session. Following the opening period, commonality remains at a relatively low and stable level for the rest of the trading day.

5 Main Empirical Results

This section presents the main empirical findings on the out-of-sample forecasting performance of the proposed models.

5.1 Forecast Evaluation Metrics and Statistical Inference

This subsection describes the statistical and economic criteria used to evaluate forecasting accuracy. We employ a set of loss functions widely used in the realised-volatility literature and complement them with economic utility measures that quantify the portfolio value of volatility forecasts. Statistical significance is assessed using Diebold–Mariano tests with HAC standard errors and the Model Confidence Set procedure.

Statistical loss functions. Let $RV_{i,t}^{(h)}$ denote the realized volatility and $\widehat{RV}_{i,t}^{(h)}$ its forecast for stock i at horizon h. We evaluate point forecasts using mean squared error (MSE), mean absolute error (MAE), and the out-of-sample R^2:

$$\text{MSE} = \frac{1}{N \cdot \#\mathcal{T}_{\text{test}}} \sum_{i=1}^{N} \sum_{t \in \#\mathcal{T}_{\text{test}}} \left(RV_{i,t}^{(h)} - \widehat{RV}_{i,t}^{(h)} \right)^2, \tag{65}$$

$$\text{MAE} = \frac{1}{N \cdot \#\mathcal{T}_{\text{test}}} \sum_{i=1}^{N} \sum_{t \in \#\mathcal{T}_{\text{test}}} \left| RV_{i,t}^{(h)} - \widehat{RV}_{i,t}^{(h)} \right|, \tag{66}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N} \sum_{t \in \#\mathcal{T}_{\text{test}}} \left(RV_{i,t}^{(h)} - \widehat{RV}_{i,t}^{(h)} \right)^2}{\sum_{i=1}^{N} \sum_{t \in \#\mathcal{T}_{\text{test}}} \left(RV_{i,t}^{(h)} - \bar{RV}_{i}^{(h)} \right)^2}, \tag{67}$$

where $\widehat{RV}_{i,t}^{(h)}$ is the predicted value of $RV_{i,t}^{(h)}$, $\bar{RV}_{i}^{(h)}$ is the mean realized volatility for asset i over the test period, N is the number of sample stocks, and $\#\mathcal{T}_{\text{test}}$ is the number of observations in the test sample.

Because realized volatility is heavy-tailed and estimated with noise, a large body of research recommends the quasi-likelihood loss (QLIKE) as a robust criterion for comparing volatility forecasts. Following recent studies using log-transformed realised variances (see Bucci, 2020; Christensen et al., 2022; Zhang et al., 2024a), we adopt the log-QLIKE form:

$$\text{QLIKE} = \frac{1}{N \cdot \#\mathcal{T}_{\text{test}}} \sum_{i=1}^{N} \sum_{t \in \#\mathcal{T}_{\text{test}}} \left[\exp\left(RV_{i,t}^{(h)} - \widehat{RV}_{i,t}^{(h)} \right) - \left(RV_{i,t}^{(h)} - \widehat{RV}_{i,t}^{(h)} \right) - 1 \right]. \tag{68}$$

Although this expression differs algebraically from the textbook variance-level QLIKE, it remains the appropriate likelihood-based score when forecasts are produced for $\log RV$, as in Bucci (2020); Zhang et al. (2024a).

Economic evaluation via realized utility. Forecast accuracy has direct implications for portfolio choice, Bollerslev et al. (2018) introduced a utility-based framework that measures the utility gains for a mean–variance investor facing time-varying volatility while assuming a constant Sharpe ratio. To quantify the economic value of competing volatility forecasts, we follow the above framework and use the CRRA-based mean–variance approximation used in the volatility-forecasting literature. Let r^e_{t+1} denote the excess return of the risky asset and let the investor allocate a fraction x_t of wealth to the risky asset. Under a standard second-order expansion of expected utility, the investor's conditional objective is

$$U_t(x_t) = x_t\,\mathbb{E}_t(r^e_{t+1}) - \frac{\gamma}{2}\,x_t^2\,\mathbb{E}_t(RV_{t+1})\,, \tag{69}$$

where γ is the coefficient of relative risk aversion. Maximising (69) yields the optimal allocation

$$x^*_t = \frac{\mathbb{E}_t(r^e_{t+1})}{\gamma\,\mathbb{E}_t(RV_{t+1})} = \frac{SR}{\gamma}\,\frac{1}{\sqrt{\mathbb{E}_t(RV_{t+1})}}, \tag{70}$$

where SR is the conditional Sharpe ratio (assumed constant following standard practice). Equation (70) shows that investors target a constant level of portfolio volatility; better volatility forecasts therefore translate directly into more accurate risk scaling.

Let $\widehat{RV}_{t+1}$ denote the model-implied forecast for RV_{t+1}. Replacing conditional expectations in (69) with realised quantities yields the realised utility (RU):

$$\text{RU} = \frac{1}{T_{\text{test}}}\sum_{t\in\mathcal{T}_{\text{test}}}\left(\frac{SR}{\gamma}\,\frac{\sqrt{RV_{t+1}}}{\sqrt{\widehat{RV}_{t+1}}} - \frac{SR^2}{2\gamma}\,\frac{RV_{t+1}}{\widehat{RV}_{t+1}}\right). \tag{71}$$

Consistent with Bollerslev et al. (2018), Zhang et al. (2024a) and Li and Tang (2025), we adopt the standard calibration in which the conditional Sharpe ratio is fixed at $SR = 0.4$ and the coefficient of relative risk aversion is set to $\gamma = 2$. These values anchor the utility calculations to a common economic scale and ensure that our results are directly comparable to prior studies.

We also consider a transaction-cost-adjusted measure, denoted RU-TC. Following Zhang et al. (2024a) and Li and Tang (2025), transaction costs are assumed to be linear in the absolute change in portfolio positions. For each stock, the proportional cost parameter κ_i is taken to be the full median bid–ask spread over the past 90 trading days, which provides a stable estimate of trading frictions while filtering out transient microstructure noise. The cost associated with adjusting the optimal position from $x^*_{i,t-1}$ to $x^*_{i,t}$ is

$$\text{TC}_{i,t} = \kappa_i\left|x^*_{i,t} - x^*_{i,t-1}\right|.$$

The transaction-cost-adjusted realised utility is therefore

$$\text{RU-TC} = \frac{1}{N\cdot\#\mathcal{T}_{\text{test}}}\sum_{i=1}^{N}\sum_{t\in\#\mathcal{T}_{\text{test}}}\left[\left(\frac{SR}{\gamma}\,\frac{\sqrt{RV_{i,t+1}}}{\sqrt{\widehat{RV}_{i,t+1}}} - \frac{SR^2}{2\gamma}\,\frac{RV_{i,t+1}}{\widehat{RV}_{i,t+1}}\right) - \text{TC}_{i,t}\right].$$

RU reflects the welfare gains from improved risk scaling, while RU-TC captures these benefits net of the turnover implied by the forecast-driven adjustments in portfolio weights. Both measures are averaged across stocks and all out-of-sample forecast periods to form the economic evaluation criteria.

Diebold–Mariano tests. To assess whether differences in forecast accuracy are statistically significant, we use the Diebold–Mariano (DM) test. For two competing models, let

$$d_t = L_t^{(A)} - L_t^{(B)}$$

denote the loss differential, where $L_t^{(\cdot)}$ is one of the loss measures above. The hypotheses are

$$H_0 : \mathbb{E}(d_t) = 0, \qquad H_1 : \mathbb{E}(d_t) < 0,$$

so that the left-tailed alternative corresponds to model A having lower expected loss. The DM statistic is

$$DM = \frac{\bar{d}}{\sqrt{\widehat{\mathrm{Var}}(\bar{d})}}, \qquad \bar{d} = \frac{1}{T_{\text{test}}} \sum_t d_t. \tag{72}$$

Loss differences d_t are serially correlated because rolling forecasts and overlapping horizons create dependence across adjacent forecast errors. Thus the variance of $\bar{d}$ must be estimated using a heteroskedasticity–autocorrelation consistent (HAC) estimator. We adopt the Newey–West long-run variance estimator (Newey and West, 1987; Andrews, 1991):

$$\widehat{\mathrm{Var}}(\bar{d}) = \frac{1}{T_{\text{test}}} \left[\hat{\gamma}(0) + 2 \sum_{k=1}^{K} \left(1 - \frac{k}{K+1} \right) \hat{\gamma}(k) \right], \tag{73}$$

where $\hat{\gamma}(k)$ is the sample autocovariance at lag k and K is the truncation parameter. This long-run variance estimator adjusts for both autocorrelation and time-varying volatility and yields valid inference for overlapping forecast errors.

Model Confidence Set. While DM tests evaluate pairwise differences, the Model Confidence Set (MCS) procedure (Hansen et al., 2011) identifies the set of models that cannot be statistically distinguished from the best-performing model at a chosen confidence level. Let $L_{m,t}$ be the loss of model m and let

$$d_{mn,t} = L_{m,t} - L_{n,t}$$

be pairwise loss differences across models. The null hypothesis of equal predictive ability is

$$H_0 : \mathbb{E}(d_{mn,t}) = 0 \quad \text{for all } m, n.$$

The MCS algorithm iteratively removes the model whose performance is most inconsistent with H_0 until the null cannot be rejected. The resulting set—the superior set of models—provides a robust, multiple-comparison-adjusted assessment of forecasting performance.

Further implementation details, including the bootstrap procedure and elimination statistics, are reported in Appendix E.

5.2 Experimental Design and Implementation

The forecasting exercises follow a strictly forward-looking walk-forward design. For each stock, the realised-volatility series from January 2019 to December 2024 is divided into a sequence of overlapping estimation windows and non-overlapping test windows. Each estimation window spans one calendar year, and the subsequent six months constitute the corresponding out-of-sample test period. The first test window covers January–June 2020. We then shift the entire estimation window forward by six months and repeat the procedure until the end of 2024. This produces a panel of rolling forecasts that mimics the information set of an investor who periodically re-estimates models as new data arrive. Figure 13 provides a visual summary of the scheme.

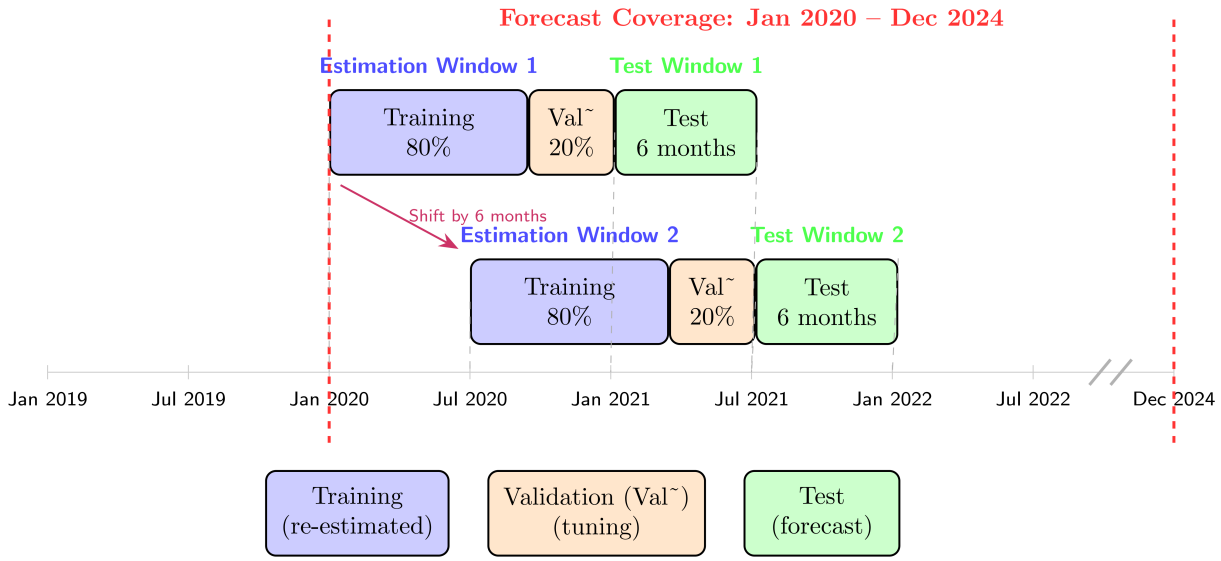


Figure 13: Illustration of the rolling estimation and evaluation scheme.

Within each one-year estimation window, observations are split into an 80% training part and a 20% validation part. All models are re-estimated from scratch in every window using only the data available at that point in time. Hyperparameter tuning—when required—is conducted exclusively on the validation portion, while the subsequent six-month period is reserved for genuine out-of-sample evaluation. Test observations never influence model selection or parameter estimation. By construction, the procedure rules out any look-ahead bias and ensures that each forecast reflects the information set that would have been available in real time.

The estimation strategy is adapted to the structure of each model class. The OLS and HAR–diurnal benchmarks contain no hyperparameters. Their coefficients are therefore estimated once per window using the full estimation sample, combining the training and validation observations to maximise statistical efficiency. For the LASSO and XGBoost models, hyperparameters are chosen by a simple grid search over pre-specified candidate values within each estimation window. The corresponding

search spaces are reported in Appendix A.4.

Neural-network architectures require more careful treatment because their training paths are sensitive to the random initialisation of network weights. To reduce variance arising from different random seeds, each neural network is trained multiple times within every estimation window under distinct seeds, and the resulting forecasts are averaged. This ensemble step improves stability without altering the underlying design and is now routine in forecast-oriented deep-learning applications (Gu et al., 2020). Hyperparameters are selected via random search on the validation split, and the configuration that yields the lowest validation loss is retained to generate forecasts in the corresponding test window.

All input variables are standardised within each estimation window following the procedure described in Appendix A.2. For every stock and forecast horizon, the mean and standard deviation used to normalise realized volatility are computed from the training portion alone. The same transformation is then applied to the validation and test observations in that window. This stock-specific normalisation is standard in high-dimensional forecasting and pooled machine-learning designs, as it aligns the scale of inputs across assets and prevents differences in unconditional volatility levels from influencing the training process (e.g. Gu et al., 2020; Zhang et al., 2024a). For pooled training schemes—Universal, Augmented, and Cluster—mini-batches may include observations from multiple stocks and dates, but the time ordering of each individual series is always respected. The Single scheme trains a separate model for each stock using only its own normalised history.

5.3 Main Empirical Findings

5.3.1 Overall out-of-sample performance

Tables 2–5 report the out-of-sample performance of all model–scheme combinations at the four intraday horizons. The evaluation metrics comprise MSE, MAE, the out-of-sample R^2, the log-QLIKE loss, and the realised-utility measures RU and RU-TC introduced in Section 5.1. For ease of exposition, the discussion below focuses on QLIKE and the utility-based criteria, while the other statistics serve as robustness checks.

Several broad patterns emerge. First, across all horizons, the linear benchmarks—OLS, HAR–diurnal, and LASSO—systematically trail the nonlinear specifications. Tree-based XGBoost improves meaningfully on the linear models, but is in turn dominated by the recurrent neural networks, especially once multi-scale and signature-based features are incorporated. Second, the choice of training scheme matters. For almost every model class and horizon, the Single scheme is weakest, the Universal and Augmented schemes deliver noticeable gains, and the Cluster scheme delivers the best or near-best performance. This ordering is visible both in loss-based metrics and in the realised-utility measures, indicating that pooling information within economically homogeneous clusters is more effective than either stock-by-stock estimation or unconditional pooling.

Turning to individual horizons, the 10-minute results in Table 2 show that the best performance is achieved by the MSA-LSTM trained under the Cluster scheme. Its QLIKE loss is about 14% lower than that of the OLS benchmark under the Single scheme, and it delivers higher RU and RU-TC, with utility gains on the order of 0.4 units relative to the linear models. The SA-GRU with Cluster training is the second-best performer at this horizon: it is slightly weaker than the MSA-

LSTM–Cluster combination in QLIKE, but yields very similar realised utilities. Both specifications clearly dominate all alternatives, including the corresponding Single, Universal, and Augmented variants.

At the 30-minute horizon (Table 3), the picture is even sharper. The MSA-LSTM and MSA-GRU under Cluster training form a clear top pair. Their QLIKE values are roughly one-third lower than those of the OLS and HAR–diurnal benchmarks, and they also attain the highest RU and RU-TC among all entries in the table. The Universal and Augmented schemes for these architectures perform well but remain uniformly weaker than the Cluster counterparts, underscoring the value of aligning the training design with cross-sectional volatility commonality.

For the 60-minute horizon (Table 4), the MSA-GRU with Cluster training emerges as the single best-performing specification, with the MSA-LSTM–Cluster combination a close second. Relative to the OLS Single benchmark, the QLIKE reduction is close to 40%, and realised utility increases by around 0.3 units, both before and after transaction costs. Again, the gains from clustering are visible within each model family: for example, Cluster-based MSA-GRU dominates its Single, Universal, and Augmented variants across all reported metrics.

Finally, at the 240-minute (daily) horizon in Table 5, the relative ordering of models is broadly consistent with the shorter horizons, although the performance gaps are somewhat smaller. The SA-GRU trained under the Cluster scheme achieves the lowest QLIKE and the highest RU and RU-TC, with the MSA-GRU–Cluster specification ranking second. Even at this lower-frequency horizon, these cluster-trained recurrent networks outperform both traditional econometric benchmarks and tree-based models, confirming that the proposed architectures retain an advantage beyond very short forecasting horizons.

Across forecast horizons, the QLIKE metric reveals a clear and persistent pattern. The lowest QLIKE values—indicating the smallest forecast errors—occur at the 60-minute horizon, followed closely by the 30-minute horizon. Predictive accuracy weakens at the 10-minute horizon, where high-frequency noise becomes more prominent, and is the lowest at the daily (240-minute) horizon. This ordering is common across all model families: multi-scale (MSA), signature-based (SA), and standard recurrent networks uniformly achieve their strongest out-of-sample performance at intermediate intraday frequencies (30–60 minutes), obtain smaller improvements at the ultra-short 10-minute horizon, and exhibit noticeably weaker gains at the daily horizon. Overall, forecastability is highest at intermediate intraday horizons, somewhat weaker at very short horizons, and lowest at the daily horizon, as measured by QLIKE.

The cross-horizon pattern in QLIKE aligns closely with the market-level volatility commonality documented earlier. Using Market-RV as the common factor, the adjusted R^2 from the market–individual regressions peaks at the 30–60 minute frequencies, declines at the 10-minute horizon, and is lowest at the daily horizon (see Figure 11). When market-wide components account for a larger share of individual-stock volatility, the series contain a greater amount of shared structure that can be exploited through pooling. This mechanism helps explain why the Cluster, Augmented, and Universal schemes—which all rely on pooling information across assets—consistently outperform the Single scheme. Relative to Single, these pooling-based designs draw on a broader cross-section of volatility histories and can therefore learn market-wide patterns that are common

across stocks while filtering out idiosyncratic noise. As a result, the gains from pooling are most pronounced at horizons with stronger market-level commonality, and remain positive across all horizons even when common components are weaker.

Entries in the Model and Scheme columns marked with an asterisk belong to the Model Confidence Set (MCS) at the chosen confidence level. Consistent with this classification, the out-of-sample QLIKE values in Tables 2–5 indicate that the Cluster-based SA and MSA architectures dominate across horizons. Moreover, the formal MCS procedure reported in Section 5.4 corroborates this ranking: after adjusting for multiple comparisons, the surviving superior models are precisely those Cluster-based SA and MSA specifications that achieve the lowest QLIKE losses.

Table 2: Out-of-sample Forecast Performance: 10-minute Horizon

Model	Scheme	MSE	MAE	R^2	QLIKE	RU	RU-TC
ols	single	0.8954	0.5790	0.4422	0.8313	3.35	2.89
	universal	0.9309	0.5996	0.4255	0.8270	3.33	2.85
	augmented	0.8940	0.5772	0.4452	0.8136	3.37	2.90
	cluster	0.8325	0.5715	0.4813	0.7805	3.45	2.89
har-d	single	0.9054	0.5806	0.4369	0.8693	3.36	2.85
	universal	0.9008	0.5787	0.4419	0.8662	3.36	2.86
	augmented	0.9056	0.5796	0.4391	0.8747	3.37	2.86
	cluster	0.8999	0.5791	0.4418	0.8591	3.37	2.86
lasso	single	0.9461	0.5916	0.4140	0.9229	3.27	2.76
	universal	0.9731	0.6047	0.4018	0.9156	3.23	2.84
	augmented	0.9347	0.5869	0.4207	0.9110	3.22	2.77
	cluster	0.8444	0.5718	0.4745	0.8061	3.45	2.92
xgb	single	0.6147	0.4778	0.6176	0.7605	3.61	3.23
	universal	0.7082	0.5104	0.5589	0.7865	3.55	3.17
	augmented	0.7047	0.5027	0.5639	0.8020	3.54	3.16
	cluster	0.6990	0.5043	0.5655	0.7946	3.56	3.17
mlp	single	0.8936	0.5656	0.4484	1.0929	2.83	2.43
	universal	0.9053	0.5845	0.4351	0.8222	3.33	2.78
	augmented	0.9132	0.5868	0.4294	0.8465	3.32	2.77
	cluster	0.9070	0.5850	0.4332	0.8291	3.32	2.77
lstm	single	0.9338	0.5757	0.4207	0.9377	3.33	2.97
	universal	0.7231	0.5067	0.5514	0.8506	3.52	3.18
	augmented	0.6746	0.4894	0.5814	0.8307	3.56	3.21
	cluster	0.6279	0.4721	0.6105	0.8108	3.60	3.24
gru	single	0.9375	0.5755	0.4194	0.9424	3.32	2.97
	universal	0.7260	0.5064	0.5504	0.8572	3.52	3.18
	augmented	0.6774	0.4892	0.5805	0.8377	3.56	3.21
	cluster	0.6304	0.4719	0.6096	0.8184	3.60	3.24
salstm	single	0.9411	0.5812	0.4164	0.9521	3.29	2.91
	universal	0.5857	0.4560	0.6366	0.7921	3.63	3.26
	augmented	0.5344	0.4356	0.6685	0.7659	3.67	3.27
	cluster	0.4862	0.4155	0.6983	0.7377	3.71	3.29
sagru	single	0.9468	0.5818	0.4134	0.9594	3.28	2.90

Model	Scheme	MSE	MAE	R^2	QLIKE	RU	RU-TC
	universal	0.5881	0.4558	0.6358	0.8004	3.63	3.26
	augmented	0.5365	0.4354	0.6677	0.7768	3.67	3.28
	cluster	0.4882	0.4153	0.6977	0.7519	3.71	3.29
msalstm∗	single	0.9386	0.5765	0.4181	0.9427	3.32	2.97
	universal	0.5348	0.4357	0.6682	0.7661	3.67	3.27
	augmented	0.4874	0.4160	0.6976	0.7385	3.71	3.28
	cluster∗	0.4430	0.3966	0.7251	0.7089	3.74	3.29
msagru	single	0.9440	0.5771	0.4139	0.9415	3.32	2.96
	universal	0.5370	0.4355	0.6675	0.7770	3.67	3.28
	augmented	0.4894	0.4158	0.6969	0.7526	3.71	3.29
	cluster	0.4448	0.3964	0.7245	0.7230	3.74	3.29

Notes: The red (blue) shading marks the best (second-best) model–scheme combination under QLIKE. An entry with * denotes inclusion in the MCS at the 5% significance level.

Table 3: Out-of-sample Forecast Performance: 30-minute Horizon

Model	Scheme	MSE	MAE	R^2	QLIKE	RU	RU-TC
ols	single	0.3781	0.4175	0.5935	0.3636	3.58	3.00
	universal	0.4211	0.4388	0.5444	0.3690	3.55	2.92
	augmented	0.3761	0.4151	0.5973	0.3471	3.59	3.00
	cluster	0.3417	0.4014	0.6210	0.3195	3.67	3.05
har-d	single	0.3818	0.4194	0.5891	0.3760	3.59	2.98
	universal	0.3758	0.4165	0.5964	0.3663	3.59	2.99
	augmented	0.3824	0.4170	0.5892	0.3763	3.59	2.98
	cluster	0.3776	0.4174	0.5946	0.3639	3.59	2.99
lasso	single	0.3871	0.4198	0.5817	0.3844	3.57	2.98
	universal	0.4367	0.4424	0.5348	0.3933	3.52	2.97
	augmented	0.3809	0.4151	0.5914	0.3665	3.58	3.01
	cluster	0.3462	0.4021	0.6171	0.3286	3.66	3.07
xgb	single	0.2572	0.3307	0.7176	0.2957	3.77	3.28
	universal	0.3271	0.3779	0.6453	0.3323	3.70	3.20
	augmented	0.2808	0.3477	0.6970	0.3053	3.74	3.23
	cluster	0.2560	0.3331	0.7233	0.2910	3.77	3.26
mlp	single	0.5143	0.4733	0.4534	0.5143	3.29	2.75
	universal	0.3765	0.4179	0.5919	0.3529	3.57	2.97
	augmented	0.3762	0.4182	0.5924	0.3568	3.56	2.96
	cluster	0.3757	0.4180	0.5944	0.3567	3.56	2.96
lstm	single	0.4129	0.4280	0.5514	0.4272	3.51	3.01
	universal	0.2633	0.3210	0.7066	0.2846	3.76	3.25
	augmented	0.2439	0.3034	0.7291	0.2727	3.79	3.27
	cluster	0.2219	0.2842	0.7534	0.2597	3.81	3.28
gru	single	0.4089	0.4262	0.5569	0.4271	3.52	3.02
	universal	0.2612	0.3202	0.7092	0.2835	3.76	3.26
	augmented	0.2410	0.3020	0.7319	0.2716	3.79	3.27

Model	Scheme	MSE	MAE	R^2	QLIKE	RU	RU-TC
	cluster	0.2199	0.2835	0.7556	0.2589	3.82	3.28
salstm	single	0.4278	0.4364	0.5387	0.4482	3.48	2.97
	universal	0.2327	0.3018	0.7408	0.2664	3.80	3.28
	augmented	0.2065	0.2791	0.7707	0.2511	3.83	3.29
	cluster	0.1797	0.2558	0.8002	0.2350	3.86	3.30
sagru	single	0.4286	0.4378	0.5373	0.4484	3.47	2.97
	universal	0.2308	0.3010	0.7430	0.2657	3.80	3.28
	augmented	0.2039	0.2778	0.7731	0.2502	3.83	3.29
	cluster	0.1782	0.2552	0.8020	0.2345	3.86	3.30
msalstm∗	single	0.4199	0.4301	0.5452	0.4330	3.51	3.01
	universal	0.2133	0.2889	0.7624	0.2548	3.82	3.29
	augmented	0.1889	0.2670	0.7902	0.2405	3.85	3.30
	cluster∗	0.1641	0.2444	0.8176	0.2252	3.87	3.30
msagru∗	single	0.4156	0.4294	0.5499	0.4311	3.51	3.01
	universal	0.2116	0.2882	0.7644	0.2542	3.82	3.29
	augmented	0.1866	0.2657	0.7924	0.2394	3.85	3.30
	cluster∗	0.1627	0.2438	0.8192	0.2246	3.87	3.30

Notes: The red (blue) shading marks the best (second-best) model–scheme combination under QLIKE. An entry with * denotes inclusion in the MCS at the 5% significance level.

Table 4: Out-of-sample Forecast Performance: 60-minute Horizon

Model	Scheme	MSE	MAE	R^2	QLIKE	RU	RU-TC
ols	single	0.3283	0.3808	0.5975	0.3191	3.63	3.08
	universal	0.3719	0.4022	0.5386	0.3208	3.61	3.03
	augmented	0.3274	0.3786	0.6001	0.3080	3.64	3.08
	cluster	0.2940	0.3640	0.6242	0.2802	3.71	3.13
har-d	single	0.3196	0.3784	0.6065	0.3179	3.64	3.08
	universal	0.3137	0.3748	0.6138	0.3076	3.65	3.09
	augmented	0.3220	0.3765	0.6034	0.3179	3.65	3.08
	cluster	0.3153	0.3761	0.6119	0.3065	3.65	3.09
lasso	single	0.3220	0.3780	0.6018	0.3252	3.63	3.10
	universal	0.3756	0.4011	0.5394	0.3252	3.61	3.09
	augmented	0.3301	0.3779	0.5958	0.3267	3.63	3.10
	cluster	0.2976	0.3632	0.6207	0.2915	3.71	3.15
xgb	single	0.2231	0.3045	0.7191	0.2514	3.80	3.31
	universal	0.3159	0.3699	0.6051	0.3061	3.69	3.20
	augmented	0.2687	0.3395	0.6657	0.2784	3.73	3.26
	cluster	0.2506	0.3284	0.6883	0.2672	3.77	3.28
mlp	single	0.1930	0.2876	0.7660	0.2385	3.80	3.29
	universal	0.2231	0.3159	0.7231	0.2335	3.76	3.23
	augmented	0.1978	0.2971	0.7548	0.2142	3.79	3.26
	cluster	0.1741	0.2790	0.7845	0.1861	3.82	3.29
lstm	single	0.3476	0.3915	0.5718	0.3585	3.58	3.10

Model	Scheme	MSE	MAE	R^2	QLIKE	RU	RU-TC
	universal	0.1882	0.2663	0.7625	0.2235	3.84	3.32
	augmented	0.1738	0.2522	0.7798	0.2146	3.85	3.33
	cluster	0.1584	0.2374	0.7988	0.2050	3.87	3.34
gru	single	0.3438	0.3901	0.5777	0.3575	3.58	3.11
	universal	0.1875	0.2654	0.7643	0.2233	3.84	3.32
	augmented	0.1728	0.2512	0.7820	0.2141	3.85	3.33
	cluster	0.1582	0.2370	0.7999	0.2049	3.87	3.34
salstm	single	0.3705	0.4054	0.5474	0.3911	3.52	3.05
	universal	0.1787	0.2589	0.7750	0.2176	3.85	3.32
	augmented	0.1656	0.2462	0.7910	0.2096	3.86	3.33
	cluster	0.1486	0.2267	0.8117	0.1986	3.88	3.34
sagru	single	0.3664	0.4039	0.5510	0.3836	3.53	3.06
	universal	0.1769	0.2570	0.7772	0.2163	3.85	3.33
	augmented	0.1633	0.2438	0.7934	0.2089	3.87	3.34
	cluster	0.1461	0.2240	0.8149	0.1977	3.89	3.35
msalstm	single	0.3467	0.3906	0.5729	0.3583	3.57	3.10
	universal	0.1597	0.2377	0.7989	0.2057	3.87	3.34
	augmented	0.1485	0.2261	0.8132	0.1997	3.88	3.34
	cluster	0.1346	0.2109	0.8298	0.1911	3.90	3.35
msagru∗	single	0.3502	0.3921	0.5682	0.3628	3.57	3.09
	universal	0.1577	0.2355	0.8017	0.2048	3.87	3.34
	augmented	0.1447	0.2214	0.8173	0.1974	3.89	3.35
	cluster∗	0.1324	0.2081	0.8324	0.1895	3.90	3.35

Notes: The red (blue) shading marks the best (second-best) model–scheme combination under QLIKE. An entry with * denotes inclusion in the MCS at the 5% significance level.

Table 5: Out-of-sample Forecast Performance: 240-minute Horizon

Model	Scheme	MSE	MAE	R^2	QLIKE	RU	RU-TC
ols	single	0.2198	0.3287	0.4633	0.1688	3.71	3.64
	universal	0.2310	0.3336	0.4190	0.1576	3.73	3.64
	augmented	0.2195	0.3271	0.4632	0.1602	3.71	3.64
	cluster	0.1880	0.3139	0.5182	0.1188	3.78	3.69
har-d	single	0.2185	0.3252	0.4675	0.1632	3.71	3.65
	universal	0.2125	0.3206	0.4806	0.1564	3.72	3.66
	augmented	0.2210	0.3266	0.4576	0.1640	3.71	3.64
	cluster	0.2173	0.3238	0.4696	0.1566	3.72	3.66
lasso	single	0.2200	0.3279	0.4645	0.1618	3.71	3.66
	universal	0.2147	0.3199	0.4590	0.1448	3.74	3.68
	augmented	0.2237	0.3303	0.4527	0.1645	3.71	3.66
	cluster	0.1707	0.2954	0.5629	0.1111	3.80	3.73
xgb	single	0.1676	0.2896	0.5933	0.1076	3.81	3.78
	universal	0.2206	0.3372	0.4689	0.1386	3.75	3.71
	augmented	0.1816	0.3040	0.5603	0.1130	3.80	3.75

Model	Scheme	MSE	MAE	R^2	QLIKE	RU	RU-TC
	cluster	0.1767	0.3022	0.5744	0.1071	3.81	3.77
mlp	single	0.2383	0.3454	0.4138	0.1776	3.68	3.64
	universal	0.2314	0.3376	0.4387	0.1701	3.69	3.63
	augmented	0.2343	0.3392	0.4317	0.1727	3.69	3.63
	cluster	0.2297	0.3367	0.4390	0.1691	3.69	3.63
lstm	single	0.2294	0.3333	0.4423	0.1755	3.69	3.64
	universal	0.1772	0.3204	0.5145	0.0983	3.81	3.63
	augmented	0.1756	0.3203	0.5067	0.0964	3.81	3.63
	cluster	0.1541	0.2978	0.5703	0.0850	3.84	3.67
gru	single	0.2358	0.3383	0.4257	0.1803	3.68	3.64
	universal	0.1598	0.3058	0.5492	0.0887	3.83	3.65
	augmented	0.1600	0.3068	0.5488	0.0882	3.83	3.65
	cluster	0.1361	0.2799	0.6133	0.0758	3.86	3.69
salstm	single	0.2710	0.3732	0.3390	0.1990	3.64	3.58
	universal	0.1794	0.3241	0.5068	0.0975	3.81	3.63
	augmented	0.1772	0.3221	0.5097	0.0973	3.81	3.63
	cluster	0.1413	0.2864	0.6062	0.0771	3.85	3.69
sagru∗	single	0.2693	0.3718	0.3420	0.1969	3.64	3.58
	universal	0.1608	0.3074	0.5534	0.0872	3.83	3.65
	augmented	0.1633	0.3103	0.5411	0.0887	3.83	3.65
	cluster∗	0.1254	0.2704	0.6442	0.0677	3.87	3.71
msalstm	single	0.2384	0.3422	0.4209	0.1800	3.68	3.64
	universal	0.1674	0.3138	0.5386	0.0916	3.82	3.64
	augmented	0.1650	0.3126	0.5345	0.0909	3.82	3.64
	cluster	0.1416	0.2857	0.6065	0.0795	3.85	3.68
msagru	single	0.2382	0.3411	0.4206	0.1808	3.68	3.63
	universal	0.1531	0.2992	0.5655	0.0839	3.84	3.66
	augmented	0.1486	0.2967	0.5703	0.0820	3.84	3.66
	cluster	0.1289	0.2730	0.6284	0.0718	3.86	3.70

Notes: The red (blue) shading marks the best (second-best) model–scheme combination under QLIKE. An entry with * denotes inclusion in the MCS at the 5% significance level.

5.3.2 Consistency of Evaluation Metrics

To assess whether different forecast evaluation criteria deliver consistent rankings across model–scheme combinations, we compute Spearman rank correlations for each metric pair at every horizon and then average the resulting matrices across the four intraday intervals. Figure 14 displays the heatmap based on this *average* correlation matrix, while the horizon-specific heatmaps are provided in Appendix F.

The results reveal a remarkably stable dependence structure. Error-based metrics—MSE, MAE, and QLIKE—are almost perfectly correlated with one another, with average Spearman coefficients between 0.95 and 0.99. These three measures are strongly negatively correlated with the accuracy- and utility-based metrics (R^2, RU, and RU-TC), reflecting their opposing optimisation directions:

the former reward smaller forecast errors, whereas the latter reward higher predictive accuracy and investment utility. The utility-based metrics themselves are highly concordant, with correlations in the 0.84–0.90 range, and R^2 aligns closely with both RU and RU-TC.

Importantly, this dependence structure is not driven by any single horizon. The horizon-specific heatmaps (Appendix F) show that the same pattern appears at 10-, 30-, 60-, and 240-minute intervals. Thus, the relative performance of models is *highly robust* to the choice of evaluation metric. This consistency justifies our emphasis on QLIKE and the two realised-utility measures in the main discussion, as these metrics are representative of the broader set of criteria and lead to the same empirical conclusions.

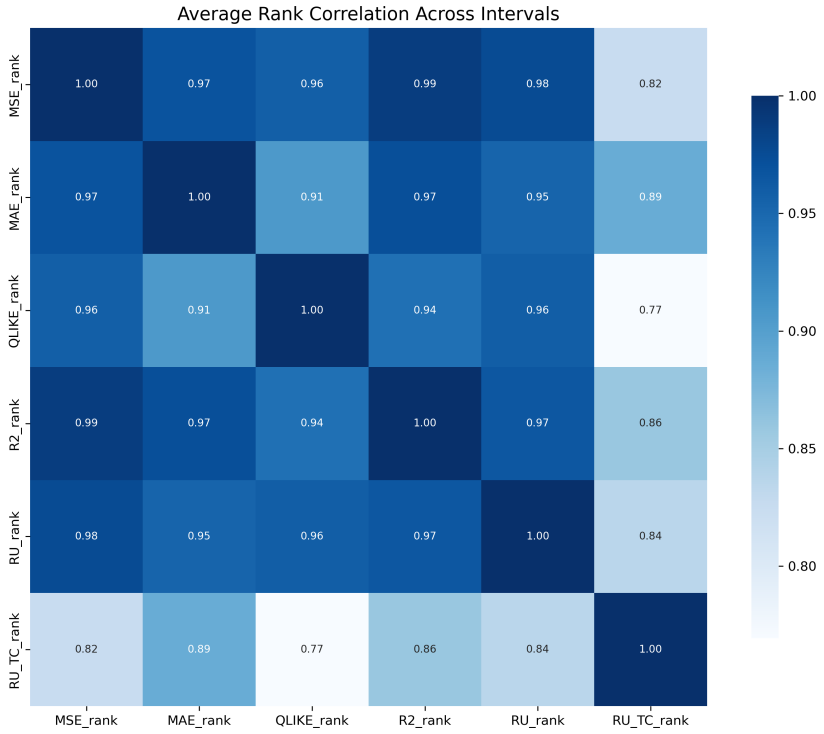


Figure 14: Average Spearman rank correlation matrix across the four intraday horizons (10, 30, 60, and 240 minutes). Darker colours indicate stronger associations. The figure shows that error-based metrics (MSE, MAE, QLIKE) are nearly perfectly correlated with one another, and strongly negatively correlated with accuracy- and utility-based metrics (R^2, RU, RU-TC). The horizon-specific matrices are reported in Appendix F.

To summarise performance across evaluation criteria, we convert each metric into a ranking of the model–scheme combinations, harmonising directions so that smaller (larger) values correspond to better performance for error-based (accuracy- and utility-based) measures. For each entry we compute the average rank over the six metrics—MSE, MAE, QLIKE, R^2, RU, and RU-TC—and aggregate these scores at the model and scheme levels. This procedure delivers a horizon-specific composite ranking that synthesises information from all evaluation criteria.

Table 6 reports the best and weakest performers at each horizon. A striking pattern emerges. Across all intraday intervals, the Cluster scheme attains the lowest composite rank, while the Single

scheme is invariably the weakest. This dominance of cluster-based pooling is fully consistent with the out-of-sample comparisons and with the MCS results in Section 4.

The model-level rankings display a similarly coherent structure. At the 10-minute horizon, MSALSTM achieves the strongest overall performance, whereas LASSO ranks last. From 30 minutes onward, MSAGRU is the best-performing model across horizons, with the weakest performers being LASSO (30 minutes), OLS (60 minutes), and MLP (240 minutes). In aggregate, the multi-scale recurrent architectures (MSA-LSTM and MSA-GRU) dominate the rankings, highlighting the importance of capturing multi-horizon dependence in realised-volatility dynamics.

As a robustness check, we repeat the ranking exercise using individual metrics rather than composite averages. Focusing on MSE, QLIKE, and RU—the three criteria most widely used in volatility-forecasting research—we find that the identity of the best and weakest model–scheme combinations remains identical to those reported in Table 6. This confirms that our conclusions do not rely on the averaging step and are not driven by any single evaluation metric.

Taken together, these results reinforce the earlier evidence and align closely with the MCS-based superior sets reported in Section 4.

Table 6: Best and weakest performers by horizon based on composite rankings.

Horizon	Best Model	Weakest Model	Best Scheme	Weakest Scheme
10 min	MSALSTM	LASSO	Cluster	Single
30 min	MSAGRU	LASSO	Cluster	Single
60 min	MSAGRU	OLS	Cluster	Single
240 min	MSAGRU	MLP	Cluster	Single

5.4 Statistical Significance: DM Tests and MCS Results

The performance differences documented in Section 5 raise a natural question: are these improvements statistically meaningful? To address this, we combine pairwise Diebold–Mariano (DM) tests with the Model Confidence Set (MCS) procedure, providing a unified assessment of statistical significance across horizons.

DM test evidence. We use pairwise DM tests based on QLIKE loss differentials to examine whether the relative performance of the four training schemes is statistically robust across model architectures. For each horizon, every model trained under a given scheme is compared with its counterpart trained under alternative schemes. Table 7 reports, for each ordered pair of schemes, how many of the eleven model architectures favour the scheme in the row at the 5% significance level. The detailed pairwise DM statistic matrices for each horizon and scheme pairing (Universal, Augmented, and Cluster versus Single) are reported in Appendix G.

Table 7: DM-based dominance counts among training schemes at the 5% significance level. Entries show, out of eleven model architectures, how many exhibit significantly lower QLIKE loss under the scheme in the row compared with the scheme in the column.

Horizon	Cluster $\succ$ Augmented	Cluster $\succ$ Universal	Cluster $\succ$ Single	Augmented $\succ$ Universal	Augmented $\succ$ Single	Universal $\succ$ Single
10 min	9	8	10	8	8	10
30 min	10	9	10	7	7	11
60 min	8	8	10	6	8	10
240 min	6	6	10	5	6	10

The dominance patterns are clear. Cluster-based training stands at the top of the hierarchy. At the 10- and 30-minute horizons it outperforms Augmented and Universal in most architectures and dominates Single almost uniformly. Its advantage over Single remains strong even at the 60- and 240-minute horizons. Augmented ranks below Cluster but above Universal: it improves on Single at all horizons and outperforms Universal in a non-trivial share of cases, particularly at short horizons. Universal in turn performs better than Single in the large majority of architectures.

These results yield a consistent ranking of the four schemes:

$$\text{Cluster} \succ \text{Augmented} \succ \text{Universal} \succ \text{Single},$$

with the largest gaps appearing at higher-frequency horizons. This hierarchy is intuitive. Richer pooling structures stabilise short-horizon volatility forecasts by exploiting cross-sectional signals that individual stocks do not contain on their own.

The gains are driven mainly by the neural-network architectures. Across the pooled schemes, the SA- and MSA-based recurrent models account for a large share of the significant wins in Table 7, whereas linear benchmarks rarely dominate. The DM evidence therefore corroborates the accuracy-, utility-, and ranking-based results in Section 5.3 and sets the stage for the MCS analysis that follows.

MCS results. While DM tests provide pairwise evidence, the MCS procedure identifies the set of models that cannot be rejected as inferior at a given confidence level. We implement the algorithm of Hansen et al. (2011) with QLIKE loss and block-bootstrap resampling. Table 8 reports the resulting superior sets at the 5% significance level.

Across all horizons, the superior sets are extremely sparse and remarkably consistent. At the 5- and 10-minute horizons, the sole surviving specification is the MSALSTM–Cluster combination. At the 30-minute horizon, the superior set contains MSALSTM–Cluster and MSAGRU–Cluster, reflecting their near-ties in the loss-based and ranking-based comparisons. At the 60-minute horizon, only MSAGRU–Cluster remains in the MCS, and at the daily (240-minute) horizon the superior set reduces to SAGRU–Cluster. Two features stand out. First, every superior set consists exclusively of Cluster-trained recurrent models, confirming the dominant role of cross-sectional pooling. Second, the winning architectures coincide exactly with the top performers in the earlier accuracy, utility, and composite-ranking analyses. Thus, statistical evidence and economic performance fully align: the improvements of cluster-based SA/MSA models are both economically large and statistically validated.

Table 8: Model Confidence Set (MCS) superior sets at the 5% significance level based on QLIKE loss. The table reports, for each intraday horizon, the set of model–scheme combinations that cannot be rejected as inferior by the MCS procedure of Hansen et al. (2011).

Horizon	Superior set
5 min	MSALSTM–Cluster
10 min	MSALSTM–Cluster
30 min	MSALSTM–Cluster; MSAGRU–Cluster
60 min	MSAGRU–Cluster
240 min	SAGRU–Cluster

Synthesis. Taken together, the DM and MCS results consolidate the main conclusions of this section. Pooling information across economically related stocks yields clear and statistically significant gains in forecast accuracy, and these gains are strongest for multi-scale recurrent architectures. At every horizon, the set of statistically superior models is small, stable, and entirely composed of Cluster-trained SA/MSA networks, providing robust support for the modelling and training choices adopted in this study.

5.5 Forecasting Improvements Across Training Schemes

A central question in multi-asset volatility forecasting is how different training schemes enhance prediction accuracy relative to stock-specific estimation. To build intuition, and in line with the structure of Zhang et al. (2024a), we use the OLS model as an illustrative example. OLS offers a transparent environment in which improvements can be linked directly to the way cross-sectional information is incorporated.

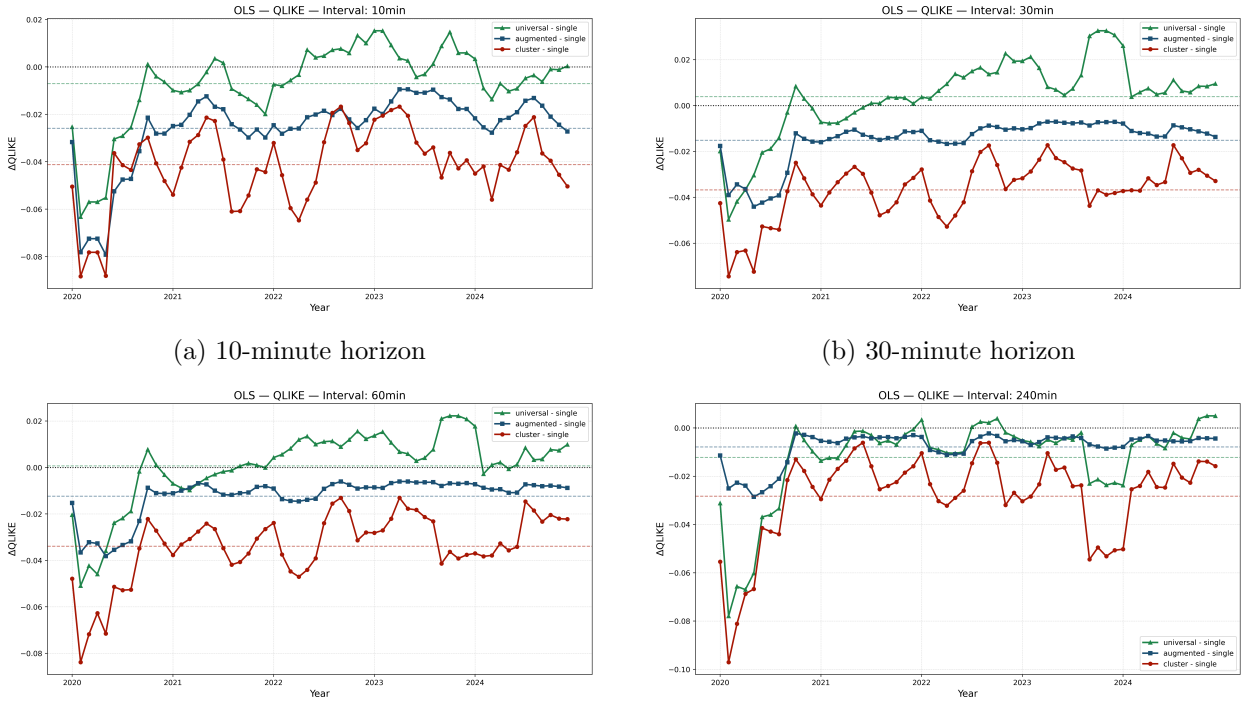
Time-series patterns of forecasting gains. For each horizon, we compute the monthly difference in QLIKE loss between a given training scheme and the stock-specific benchmark. This metric shows when each scheme delivers gains and how the magnitude of these gains evolves over time.

Across all horizons, the ordering of performance is stable:

$$\text{Cluster} \succ \text{Augmented} \succ \text{Universal} \succ \text{Single}.$$

The **Cluster** specification provides the largest and most persistent reductions in QLIKE. The **Augmented** model also improves performance throughout, with dynamics that mirror the mechanisms noted in earlier studies of pooled volatility forecasting. The **Universal** variant yields only modest and horizon-dependent effects, offering small but steady improvements.

The four panels in Figure 15 summarise these time-series patterns across horizons.



(a) 10-minute horizon

(b) 30-minute horizon

(c) 60-minute horizon

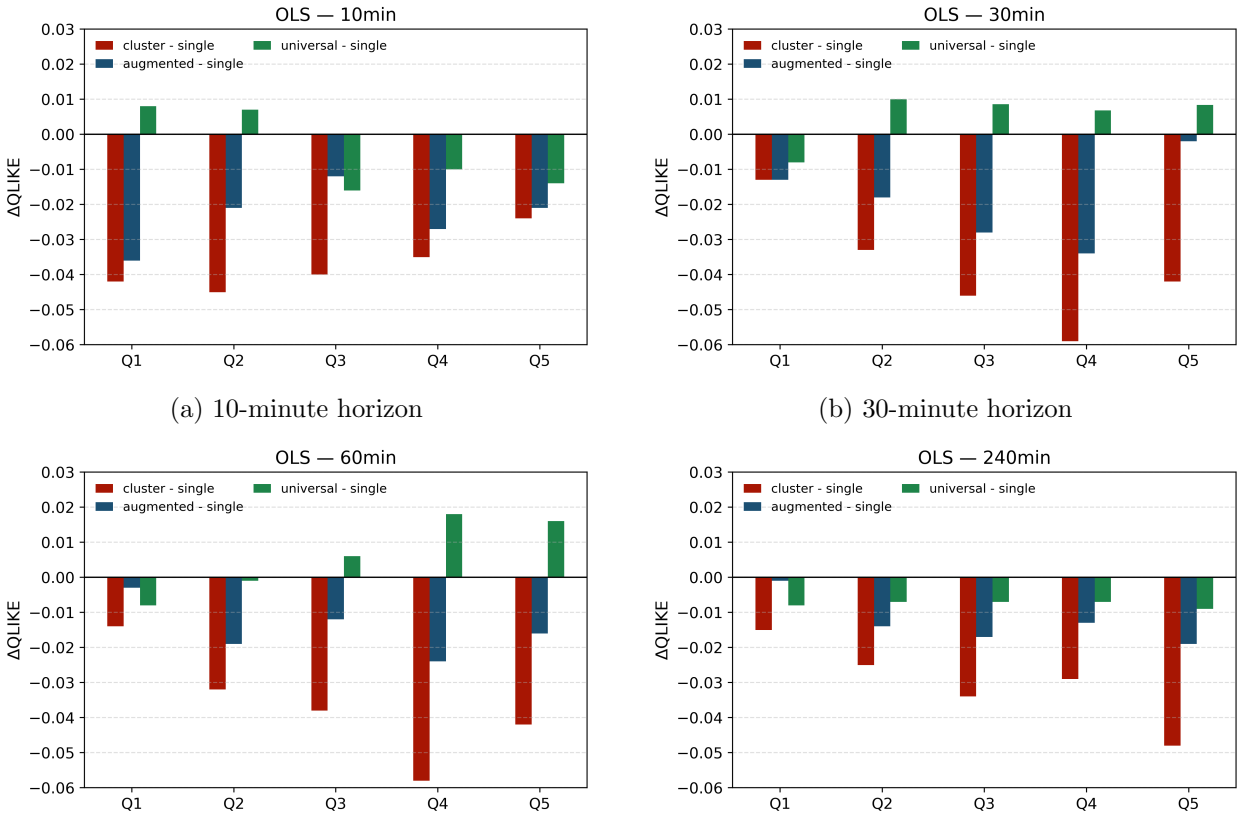
(d) 240-minute horizon

Figure 15: Time-series differences in QLIKE loss between each training scheme and the stock-specific baseline for the OLS model. Each panel reports the monthly average of the loss differential over the forecast evaluation period. Negative values indicate lower QLIKE relative to the Single scheme.

Cross-sectional patterns and volatility commonality. To examine which stocks benefit most from pooling, we sort stocks into quintiles (Q1–Q5) based on their volatility commonality. Commonality is measured using the adjusted R^2 from a regression of stock-level realized volatility on an aggregate volatility series.

This cross-sectional view produces clear patterns. For all horizons, the gains from the **Cluster** and **Augmented** schemes increase from Q1 to Q5. Stocks with stronger co-movement with market- or sector-level volatility display the largest reductions in QLIKE. Both schemes also deliver meaningful improvements in the lower-commonality groups, suggesting that structured pooling stabilises estimation and filters out idiosyncratic noise.The **Universal** model shows only mild variation and stays relatively flat across quintiles, reinforcing the dominance of the Cluster and Augmented designs.

Figure 16 summarises the cross-sectional patterns for all four horizons.



(a) 10-minute horizon

(b) 30-minute horizon

(c) 60-minute horizon

(d) 240-minute horizon

Figure 16: Cross-sectional differences in QLIKE loss across commonality quintiles. Stocks are sorted into Q1–Q5 based on their volatility commonality, with Q5 representing the highest-commonality group. Negative values indicate lower QLIKE relative to the Single scheme.

Summary. The evidence from both perspectives points to the same conclusion: while pooling is useful, *the structure of pooling is essential*. Sector-level clustering and market-based augmentation deliver large and consistent improvements over stock-specific estimation and outperform unstructured pooling by a wide margin. These findings extend earlier insights on volatility commonality and highlight its central role in short-horizon forecasting.

6 Why Cluster-Based Training Works

6.1 Market- vs. Cluster-Level RV: Motivation and Exploratory Evidence

Our cluster-based training schemes are motivated by the idea that individual-stock RV is driven by both market-wide and sector-specific components (Christensen and Prabhala, 1998; Barigozzi and Hallin, 2017; Laborda and Olmo, 2021). If sector-level RV captures systematic forces that are not fully summarised by the market aggregate, then conditioning on sector information may improve RV forecasts relative to purely market-based pooling.

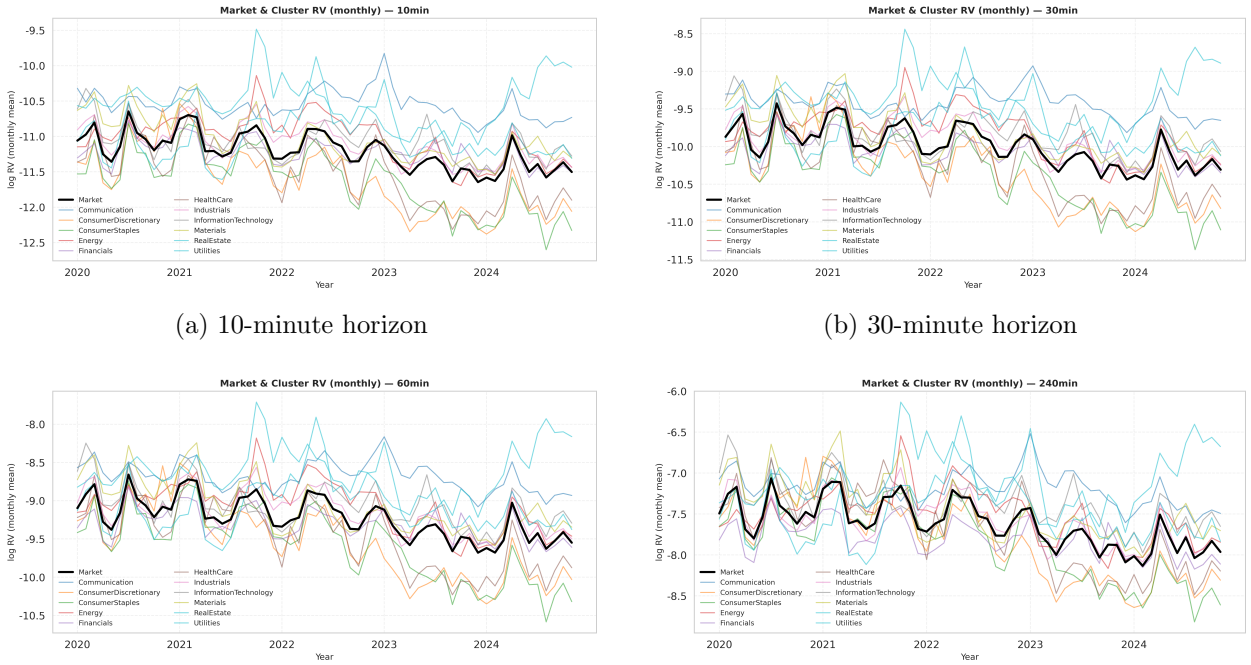
To obtain a first view of this structure, we construct RV for the market portfolio and for each

GICS sector and average these series at the monthly frequency. Figure 17 reports the time series of log RV for the market (bold line) and for all sectors (thin lines) at the 10-, 30-, 60-, and 240-minute horizons, based on equal-weighted averages across the stocks in each group.

Two facts emerge clearly from the four panels. First, sector-level RV co-moves closely with the market series. Peaks and troughs in market RV are mirrored by most sectors at all horizons, indicating a strong common component in high-frequency RV. Second, the sector curves display substantial dispersion around the market path. In many periods, some sectors lie persistently above or below the market line, and sector-level RV typically fluctuates more strongly than the aggregate. This pattern suggests that sector portfolios are exposed to additional shocks that are largely shared within the sector but only partly reflected at the market level.

These observations are directly relevant for our modelling choices. Sector RV can be viewed as a finer decomposition of the systematic environment faced by individual stocks. Pooling stocks within sectors allows the forecasting model to exploit both the market-wide component and the sector-specific variation that is common within a cluster (Leippold et al., 2022). This provides an intermediate design between stock-by-stock estimation and fully pooled training, and helps to explain why cluster-based schemes may deliver gains over universal or augmented pooling in the empirical results reported below.

Taken together, the two features highlighted in Figure 17—the strong co-movement between sector- and market-level RV on the one hand, and the persistent cross-sector dispersion on the other—demonstrate that RV contains both a "market common" and a "sector heterogeneous" structural component. It is precisely this dual structure that enables sector-based clustering to capture market-wide and sector-specific systematic forces simultaneously within the training sample. This observation provides direct conceptual support for why cluster-based training should work in practice and helps explain its superior empirical performance relative to fully pooled or stock-specific approaches in the results that follow.



(a) 10-minute horizon

(b) 30-minute horizon

(c) 60-minute horizon

(d) 240-minute horizon

Figure 17: Monthly realized volatility for the market and for each GICS sector at the 10-, 30-, 60-, and 240-minute horizons. Market realized volatility is shown in bold. Sector curves follow the broad market cycle but exhibit stronger variation and heterogeneous timing across industries.

6.2 Naïve Regression Decomposition

To understand whether sector-level realized volatility (RV) contains information that is not already summarised by the market aggregate, we estimate three simple regressions for each stock. These models are intentionally transparent. They allow us to isolate how closely individual-stock RV aligns with movements at the market level and at the sector level, and whether one of these aggregates captures variation that the other cannot. This diagnostic step helps clarify why sector-based pooling may provide a more effective learning structure than either stock-specific estimation or fully pooled training.

For stock i, the three specifications are:

$$\text{Model 1:} \quad RV_{i,t} = \alpha_i + \beta_{1,i}\, RV_{\text{mkt},t} + \varepsilon_{i,t}, \tag{74}$$

$$\text{Model 2:} \quad RV_{i,t} = \alpha_i + \beta_{2,i}\, RV_{\text{clu},t} + \varepsilon_{i,t}, \tag{75}$$

$$\text{Model 3:} \quad RV_{i,t} = \alpha_i + \beta_{1,i}\, RV_{\text{mkt},t} + \beta_{2,i}\, RV_{\text{clu},t} + \varepsilon_{i,t}. \tag{76}$$

Model 1 evaluates the alignment between individual RV and broad market conditions. Model 2 focuses on industry-level conditions. Model 3 assesses the marginal contribution of each aggregate when both are included. Adjusted R^2 is used as a consistent measure of relative model adequacy across the three specifications; its role here is to compare the in-sample fit after accounting for model dimensionality, rather than to interpret the proportion of variance "explained" in a structural sense.

Across all intraday horizons, Model 2 delivers a higher adjusted R^2 than Model 1. This pattern appears in the means, medians, and quartiles (Tables 9–12). For example, the mean adjusted R^2 increases from 0.34 to 0.43 at the 10-minute horizon, from 0.47 to 0.56 at 30 minutes, from 0.50 to 0.60 at 60 minutes, and from 0.35 to 0.49 at 240 minutes. These differences suggest that sector-level RV aligns more closely with stock-level RV than the market aggregate does, once differences in model complexity are controlled for. The shifts are systematic rather than driven by outliers: the entire cross-sectional distribution moves upward.

The joint regression (Model 3) further clarifies the relation between the two aggregates. The coefficient on cluster RV remains large and statistically significant across horizons, while the coefficient on market RV is small, often close to zero, and unstable in sign. This coefficient pattern indicates that sector-level movements subsume most of the variation that can be captured by market-level RV. Model 3 achieves the highest adjusted R^2, although the gains over Model 2 are modest, implying that the incremental information in market-level RV is limited once sector conditions are included.

To evaluate whether the differences in adjusted R^2 between the market and cluster regressions are statistically meaningful, we compare the two specifications stock by stock. For each stock i, define the improvement

$$\Delta_i = adR^2_{i,\text{clu}} - adR^2_{i,\text{mkt}}.$$

We first apply a paired t-test, which assesses whether the *average* improvement is positive:

$$H_0 : \mathbb{E}[\Delta_i] = 0 \qquad \text{vs.} \qquad H_1 : \mathbb{E}[\Delta_i] > 0, \tag{77}$$

with test statistic

$$t = \frac{\bar{\Delta}}{s_\Delta/\sqrt{n}}, \tag{78}$$

where $\bar{\Delta}$ is the sample mean of Δ_i and s_Δ its standard deviation. The paired t-test provides a direct measure of whether the typical improvement in fit is materially above zero under the assumption of approximate symmetry and moderate tail behavior in the distribution of differences.

Because adjusted R^2 differences may exhibit mild skewness across stocks, we complement the t-test with the Wilcoxon signed-rank test. The Wilcoxon test evaluates whether the *median* improvement is positive:

$$H_0 : \text{median}(\Delta_i) = 0 \qquad \text{vs.} \qquad H_1 : \text{median}(\Delta_i) > 0. \tag{79}$$

Rather than relying on means or normality, the test ranks the absolute differences $|\Delta_i|$ and attaches signs according to whether Δ_i is positive. The resulting statistic captures whether positive improvements arise systematically and with non-negligible magnitude. This makes the Wilcoxon test a natural robustness check, ensuring that our conclusions do not depend on distributional assumptions or a few extreme observations.

The results, reported in Table 13, are highly consistent across horizons. Mean improvements range from 0.086 to 0.145, and between 90% and 93% of stocks exhibit higher adjusted R^2 under

the cluster regression. Both the paired t-statistics and Wilcoxon signed-rank statistics indicate significance at the 1% level. These findings reject the null of equal model fit and demonstrate that the advantage of the cluster specification is broad-based rather than concentrated in a small subset of stocks. From an economic perspective, the results suggest that sector-level realized volatility captures systematic components of individual-stock RV that are not fully reflected in the market aggregate, and that this incremental information is pervasive across the cross-section.

Taken together, the naïve regressions reveal a clear pattern: individual-stock RV loads heavily on sector-level conditions and only weakly on market-wide movements. This layered structure—a broad market component combined with a stronger sector-specific component— provides direct economic support for the cluster-based learning schemes used in the forecasting models that follow.

Table 9: Summary Statistics of Regressions (10-minute Horizon)

Group	**Metric**	Mean	Median	Q1	Q3
Model 1:	$RV_i \sim RV_{\text{mkt}}$				
	β	0.9846	1.1312	0.6001	1.3979
	p-value	0.0001	0.0000	0.0000	0.0000
	Adjusted R^2	0.3436	0.3988	0.1795	0.4984
Model 2:	$RV_i \sim RV_{\text{clu}}$				
	β	1.0000	0.9820	0.7444	1.2551
	p-value	0.0000	0.0000	0.0000	0.0000
	Adjusted R^2	0.4292	0.4770	0.3360	0.5673
Model 3:	$RV_i \sim RV_{\text{clu}} + RV_{\text{mkt}}$				
	β_{clu}	1.0000	0.9458	0.6882	1.1818
	p-value$_{\text{clu}}$	0.0000	0.0000	0.0000	0.0000
	β_{mkt}	0.0000	0.0963	−0.2199	0.4125
	p-value$_{\text{mkt}}$	0.0260	0.0000	0.0000	0.0000
	Adjusted R^2 (joint)	0.4520	0.5096	0.3410	0.5889

Table 10: Summary Statistics of Regressions (30-minute Horizon)

Group	Metric	Mean	Median	Q1	Q3
Model 1:	$RV_i \sim RV_{\text{mkt}}$				
	β	0.9960	1.1579	0.6265	1.4010
	p-value	0.0000	0.0000	0.0000	0.0000
	Adjusted R^2	0.4657	0.5129	0.3678	0.6088
Model 2:	$RV_i \sim RV_{\text{clu}}$				
	β	1.0000	1.0116	0.7473	1.2714
	p-value	0.0000	0.0000	0.0000	0.0000
	Adjusted R^2	0.5647	0.6178	0.4999	0.6992
Model 3:	$RV_i \sim RV_{\text{clu}} + RV_{\text{mkt}}$				
	β_{clu}	1.0000	0.9435	0.6890	1.2876
	p-value$_{\text{clu}}$	0.0028	0.0000	0.0000	0.0000
	β_{mkt}	0.0000	−0.0393	−0.2295	0.2140
	p-value$_{\text{mkt}}$	0.0338	0.0000	0.0000	0.0000
	Adjusted R^2 (joint)	0.5765	0.6265	0.5132	0.7098

Table 11: Summary Statistics of Regressions (60-minute Horizon)

Group	Metric	Mean	Median	Q1	Q3
Model 1:	$RV_i \sim RV_{\text{mkt}}$				
	β	0.9956	1.1559	0.6185	1.4060
	p-value	0.0000	0.0000	0.0000	0.0000
	Adjusted R^2	0.5010	0.5582	0.3908	0.6461
Model 2:	$RV_i \sim RV_{\text{clu}}$				
	β	1.0000	1.0147	0.7465	1.2819
	p-value	0.0000	0.0000	0.0000	0.0000
	Adjusted R^2	0.5999	0.6496	0.5256	0.7357
Model 3:	$RV_i \sim RV_{\text{clu}} + RV_{\text{mkt}}$				
	β_{clu}	1.0000	0.9446	0.6761	1.3006
	p-value$_{\text{clu}}$	0.0099	0.0000	0.0000	0.0000
	β_{mkt}	0.0000	−0.0263	−0.2385	0.2261
	p-value$_{\text{mkt}}$	0.0610	0.0000	0.0000	0.0002
	Adjusted R^2 (joint)	0.6138	0.6608	0.5499	0.7514

Table 12: Summary Statistics of Regressions (240-minute Horizon)

Group	Metric	Mean	Median	Q1	Q3
Model 1:	$RV_i \sim RV_{\text{mkt}}$				
	β	0.9923	0.9803	0.6260	1.4363
	p-value	0.0009	0.0000	0.0000	0.0000
	Adjusted R^2	0.3477	0.3771	0.2251	0.4795
Model 2:	$RV_i \sim RV_{\text{clu}}$				
	β	1.0000	1.0246	0.7263	1.2739
	p-value	0.0027	0.0000	0.0000	0.0000
	Adjusted R^2	0.4926	0.5054	0.3826	0.6343
Model 3:	$RV_i \sim RV_{\text{clu}} + RV_{\text{mkt}}$				
	β_{clu}	1.0000	0.9348	0.6912	1.2752
	p-value$_{\text{clu}}$	0.0016	0.0000	0.0000	0.0000
	β_{mkt}	0.0000	0.0631	−0.3592	0.3236
	p-value$_{\text{mkt}}$	0.0393	0.0000	0.0000	0.0025
	Adjusted R^2 (joint)	0.5193	0.5366	0.4080	0.6405

Table 13: Differences in Adjusted R^2: Cluster vs. Market Models

Interval	n	Mean ΔadR^2	Median ΔadR^2	Q1	Q3	Share($\Delta adR^2 > 0$)	t-stat	Wilcoxon W
10min	100	0.0856	0.0621	0.0335	0.1060	0.90	7.8002***	212***
30min	100	0.0991	0.0809	0.0365	0.1212	0.93	10.1632***	60***
60min	100	0.0989	0.0894	0.0359	0.1219	0.92	9.8389***	93***
240min	100	0.1449	0.1352	0.0385	0.2007	0.91	10.0551***	202***

Notes: $\Delta adR^2 = adR^2_{\text{clu}} - adR^2_{\text{mkt}}$. Stars indicate significance of the paired t-test and Wilcoxon signed-rank test: ***, **, and * denote the 1%, 5%, and 10% levels, respectively.

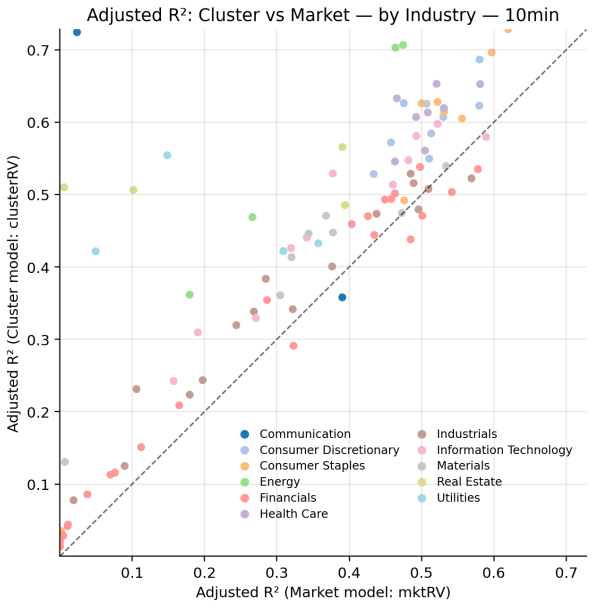
6.3 Adjusted R^2 Comparison Across Industries

A complementary view of the relative fit of the market and sector regressions is obtained from scatter plots of adjusted R^2 across stocks and industries. For each intraday horizon, we plot the adjusted R^2 from the market regression (Model 1) on the horizontal axis and the adjusted R^2 from the sector regression (Model 2) on the vertical axis, using a different colour for each GICS sector. Figure 18 displays the resulting cloud of points for the 10-, 30-, 60-, and 240-minute horizons together with the 45-degree line that corresponds to equal fit of the two models.

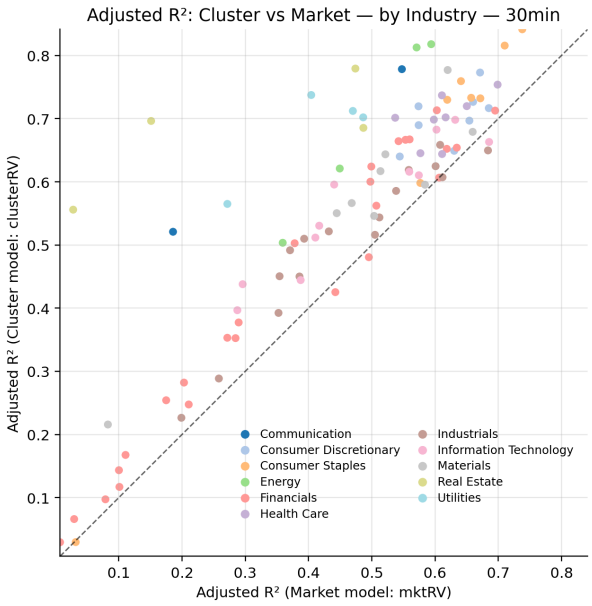
The scatter plots reveal a striking and robust pattern. At all horizons, the bulk of observations lies above the 45-degree line, indicating that sector-level RV delivers a higher adjusted R^2 than market-level RV for the vast majority of stocks. Points that fall below the line are rare and typically close to it, so that losses from replacing the market aggregate with sector RV are small when they occur. The dominance of sector RV is most pronounced at the intermediate horizons (30 and 60 minutes), where the cloud is shifted farthest above the line and the improvements in

adjusted R^2 are largest. At the very short (10-minute) and daily (240-minute) horizons, the points are more dispersed, yet they remain overwhelmingly concentrated above the 45-degree line. This pattern aligns with the evidence in Table 13 and with the distributional histograms of Δ_i reported in Appendix H.

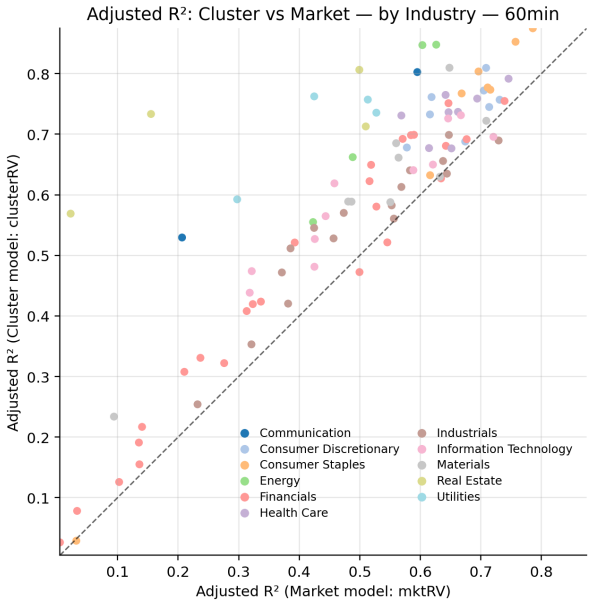
Colour-coding by industry highlights a clear clustering structure. Stocks within the same sector tend to occupy similar regions of the scatter plot: for example, energy and materials firms often lie in the upper-right area, with relatively high adjusted R^2 under both specifications but a particularly strong advantage for sector RV, while some service-oriented sectors exhibit more moderate levels of fit and smaller gains. These patterns are in line with the notion that industry-level shocks—including policy interventions, regulatory changes, and sector-specific demand or supply conditions—propagate strongly within sectors and are only partly reflected in the broad market index. Sector RV therefore captures systematic forces that are shared within the cluster but attenuated at the aggregate level.



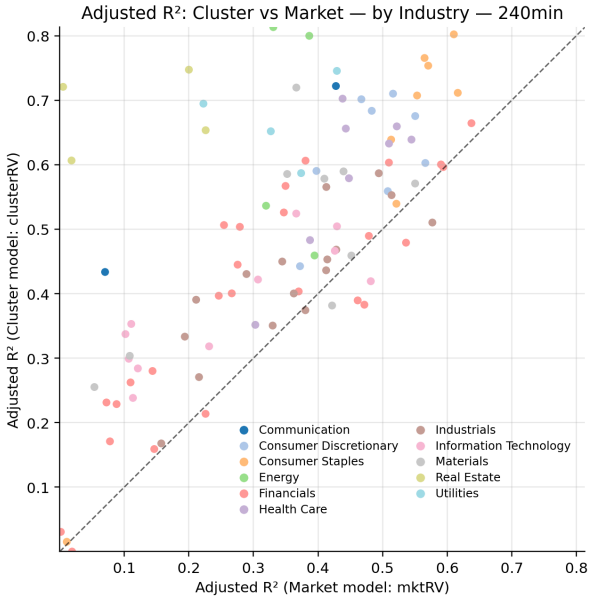
(a) 10-minute horizon



(b) 30-minute horizon



(c) 60-minute horizon



(d) 240-minute horizon

Figure 18: Comparison of adjusted R^2 from sector and market regressions across four intraday horizons. Each point corresponds to one stock, colours denote sectors, and the dashed 45-degree line marks equal fit of the two models. Points predominantly lie above the line, indicating that sector-level RV typically yields higher adjusted R^2, especially at the 30- and 60-minute horizons.

Implications for Forecasting Performance. The cross-sectional patterns in Figure 18 help explain the ranking of forecasting schemes in the main results. Because sector RV aligns more closely with stock-level RV than the market aggregate and does so in a way that is tightly organised by industry, pooling observations within sectors allows the forecasting models to exploit a rich

sector-specific systematic component in addition to the market-wide one. This mechanism naturally favours the cluster-based training schemes over both stock-specific estimation, which ignores cross-sectional information, and fully pooled approaches, which aggregate all assets into a single universal model. In line with this interpretation, the cluster schemes deliver the lowest QLIKE loss, the highest realised-utility gains, and the strongest dominance in the DM and MCS comparisons reported in the subsequent sections.

7 Conclusion

This study investigates the forecasting of realized volatility in the Chinese A-share market using minute-by-minute Level-1 data and a broad range of modelling approaches. We examine both intraday and end-of-day horizons through a unified empirical framework that spans linear benchmarks, tree-based methods, and several neural-network architectures. Within this setting we introduce two new classes of models—Signature Attention (SA) and Multi-Scale Attention (MSA)—that enrich the representation of historical information, together with a sector-based cluster training scheme designed to reflect the structure of volatility comovement in the cross-section. These components form a coherent set of tools for studying how temporal patterns and cross-sectional organisation jointly shape volatility dynamics.

The empirical results show that the proposed SA and MSA models provide stable improvements across horizons. Their designs allow neural networks to organise information in ways that conventional architectures find difficult: the SA models emphasise the geometric structure of past movements, while the MSA models capture fluctuations operating at distinct temporal scales. Other neural networks also perform better than linear benchmarks, indicating the value of nonlinear filtering in high-frequency environments, and tree-based methods deliver competitive performance in several cases. The general pattern is that models capable of representing richer temporal structure tend to deliver more reliable forecasts.

Training design is equally important. Pooling information across assets almost always yields more accurate forecasts than estimating separate models for each stock, suggesting that volatility contains systematic components that can be exploited even at high frequencies. Within this broader finding, the cluster scheme provides a disciplined way to incorporate industry-level structure. It enables models to learn patterns that are shared within economic groups yet remain difficult to extract under undifferentiated pooling or simple augmentation with market-level indicators. The evidence points to the usefulness of training strategies that respect the organisation of the cross-section rather than treating assets as interchangeable.

Several avenues for further work are natural extensions of this study. The modelling principles underlying the SA and MSA architectures may be applied to multivariate volatility measures, to execution-related risk inputs, or to learning problems in which the shape of the return path plays a central role. At the same time, structure-aware pooling can itself be refined, allowing interactions between sector-level signals and asset-specific behaviour to be incorporated in more flexible ways. Such developments may offer a broader perspective on how cross-sectional information can be integrated into high-frequency volatility models.

References

Améndola, C., Friz, P., and Sturmfels, B. (2019). Varieties of signature tensors. In *Forum of Mathematics, Sigma*, volume 7, page e10. Cambridge University Press.

Andersen, T. G. and Bollerslev, T. (1998). Deutsche mark–dollar volatility: intraday activity patterns, macroeconomic announcements, and longer run dependencies. *the Journal of Finance*, 53(1):219–265.

Andersen, T. G., Bollerslev, T., Diebold, F. X., and Ebens, H. (2001a). The distribution of realized stock return volatility. *Journal of financial economics*, 61(1):43–76.

Andersen, T. G., Bollerslev, T., Diebold, F. X., and Ebens, H. (2001b). The distribution of realized stock return volatility. *Journal of financial economics*, 61(1):43–76.

Andersen, T. G., Bollerslev, T., Diebold, F. X., and Labys, P. (2003). Modeling and forecasting realized volatility. *Econometrica*, 71(2):579–625.

Andrews, D. W. (1991). Heteroskedasticity and autocorrelation consistent covariance matrix estimation. *Econometrica: Journal of the Econometric Society*, pages 817–858.

Astudillo, K., Flores, M., Soliz, M., Ferreira, G., and Varela-Aldás, J. (2025). A hybrid gas-att-lstm architecture for predicting non-stationary financial time series. *Mathematics*, 13(14):2300.

Barigozzi, M. and Hallin, M. (2017). Generalized dynamic factor models and volatilities: estimation and forecasting. *Journal of Econometrics*, 201(2):307–321.

Barndorff-Nielsen, O. E. and Shephard, N. (2002). Estimating quadratic variation using realized variance. *Journal of Applied econometrics*, 17(5):457–477.

Barndorff-Nielsen, O. E. and Shephard, N. (2004). Power and bipower variation with stochastic volatility and jumps. *Journal of financial econometrics*, 2(1):1–37.

Baruník, J. and Křehlík, T. (2018). Measuring the frequency dynamics of financial connectedness and systemic risk. *Journal of Financial Econometrics*, 16(2):271–296.

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

Bengio, Y., Goodfellow, I., Courville, A., et al. (2017). *Deep learning*, volume 1. MIT press Cambridge, MA, USA.

Bennedsen, M., Lunde, A., and Pakkanen, M. S. (2022). Decoupling the short- and long-term behavior of stochastic volatility. *Journal of Financial Econometrics*, 20(5):961–1006.

Boedihardjo, H., Geng, X., and Souris, N. P. (2020). Path developments and tail asymptotics of signature for pure rough paths. *Advances in Mathematics*, 364:107043.

Bollerslev, T., Hood, B., Huss, J., and Pedersen, L. H. (2018). Risk everywhere: Modeling and managing volatility. *The Review of Financial Studies*, 31(7):2729–2773.

Bollerslev, T., Patton, A. J., and Quaedvlieg, R. (2016). Exploiting the errors: A simple approach for improved volatility forecasting. *Journal of Econometrics*, 192(1):1–18.

Borovykh, A., Bohte, S., and Oosterlee, C. W. (2018). Dilated convolutional neural networks for time series forecasting. *Journal of Computational Finance*, 22(4):73–101.

Boubaker, S., Liu, Z., and Zhai, L. (2022). Change-points and functional features of intraday volatility in china stock market. *Annals of Operations Research*, 352(3):563–582.

Branco, R. R., Rubesam, A., and Zevallos, M. (2024). Forecasting realized volatility: Does anything beat linear models? *Journal of Empirical Finance*, 78:101524.

Bucci, A. (2020). Realized volatility forecasting with neural networks. *Journal of Financial Econometrics*, 18(3):502–531.

Cao, J., Li, Z., and Li, J. (2019). Financial time series forecasting model based on ceemdan and lstm. *Physica A: Statistical mechanics and its applications*, 519:127–139.

Chatigny, P., Patenaude, J.-M., and Wang, S. (2021). Spatiotemporal adaptive neural network for long-term forecasting of financial time series. *International Journal of Approximate Reasoning*, 132:70–85.

Chen, J., Haboub, A., Khan, A., and Mahmud, S. (2025). Investor clientele and intraday patterns in the cross section of stock returns. *Review of Quantitative Finance and Accounting*, 64(2):757–797.

Chen, L., Pelger, M., and Zhu, J. (2024). Deep learning in asset pricing. *Management Science*, 70(2):714–750.

Chen, T. (2016). Xgboost: A scalable tree boosting system. *Cornell University*.

Chen, Y., Kang, Y., Chen, Y., and Wang, Z. (2020). Probabilistic forecasting with temporal convolutional neural network. *Neurocomputing*, 399:491–501.

Chevyrev, I. and Kormilitzin, A. (2025). A primer on the signature method in machine learning. In *Signature Methods in Finance: An Introduction with Computational Applications*, pages 3–64. Springer.

Chordia, T., Roll, R., and Subrahmanyam, A. (2000). Commonality in liquidity. *Journal of financial economics*, 56(1):3–28.

Christensen, B. J. and Prabhala, N. R. (1998). The relation between implied and realized volatility. *Journal of financial economics*, 50(2):125–150.

Christensen, K., Siggaard, M., and Veliyev, B. (2022). A machine learning approach to volatility forecasting. *Journal of Financial Econometrics*, 21(5):1680–1727.

Clements, A. and Preve, D. P. (2021). A practical guide to harnessing the har volatility model. *Journal of Banking & Finance*, 133:106285.

Colmenarejo, L., Galuppi, F., and Michałek, M. (2020). Toric geometry of path signature varieties. *Advances in Applied Mathematics*, 121:102102.

Corsi, F. (2009). A simple approximate long-memory model of realized volatility. *Journal of financial econometrics*, 7(2):174–196.

Dang, T. L., Moshirian, F., and Zhang, B. (2015). Commonality in news around the world. *Journal of Financial Economics*, 116(1):82–110.

Di Persio, L., Garbelli, M., Mottaghi, F., and Wallbaum, K. (2023). Volatility forecasting with hybrid neural networks methods for risk parity investment strategies. *Expert Systems with Applications*, 229:120418.

Ding, S., Cui, T., and Zhang, Y. (2022). Futures volatility forecasting based on big data analytics with incorporating an order imbalance effect. *International Review of Financial Analysis*, 83:102255.

Dudek, G., Fiszeder, P., Kobus, P., and Orzeszko, W. (2024). Forecasting cryptocurrencies volatility using statistical and machine learning methods: A comparative study. *Applied soft computing*, 151:111132.

Fermanian, A. (2021). Embedding and learning with signatures. *Computational Statistics & Data Analysis*, 157:107148.

Fischer, T. and Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European journal of operational research*, 270(2):654–669.

Friz, P. K., Lyons, T., and Seigal, A. (2024). Rectifiable paths with polynomial log-signature are straight lines. *Bulletin of the London Mathematical Society*, 56(9):2922–2934.

Gajamannage, K., Park, Y., and Jayathilake, D. I. (2023). Real-time forecasting of time series in financial markets using sequentially trained dual-lstms. *Expert Systems with Applications*, 223:119879.

Galuppi, F. (2019). The rough veronese variety. *Linear algebra and its applications*, 583:282–299.

Gao, Y., He, D., Mu, Y., and Zhao, H. (2023). Realised volatility prediction of high-frequency data with jumps based on machine learning. *Connection Science*, 35(1):2210265.

García-Medina, A. and Aguayo-Moreno, E. (2023). Lstm–garch hybrid model for the prediction of volatility in cryptocurrency portfolios. *Computational Economics*, 63(4):1511–1542.

Gatheral, J., Jaisson, T., and Rosenbaum, M. (2018). Volatility is rough. *Quantitative Finance*, 18(6):933–949.

Geng, X. (2017). Reconstruction for the signature of a rough path. *Proceedings of the London Mathematical Society*, 114(3):495–526.

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

Gunnarsson, E. S., Isern, H. R., Kaloudis, A., Risstad, M., Vigdel, B., and Westgaard, S. (2024). Prediction of realized volatility and implied volatility indices using ai and machine learning: A review. *International review of financial analysis*, 93:103221.

Guo, M. (2006). *Intraday return, volatility and liquidity: An investigation of the market microstructure of the Chinese stock market*. PhD thesis, PhD Thesis, University of Western Sydney.

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

Hao, X., Zhao, Y., and Wang, Y. (2020). Forecasting the real prices of crude oil using robust regression models with regularization constraints. *Energy Economics*, 86:104683.

Herrera, A. M., Hu, L., and Pastor, D. (2018). Forecasting crude oil price volatility. *International Journal of Forecasting*, 34(4):622–635.

Herskovic, B., Kelly, B., Lustig, H., and Van Nieuwerburgh, S. (2016). The common factor in idiosyncratic volatility: Quantitative asset pricing implications. *Journal of Financial Economics*, 119(2):249–283.

Huo, R. and Ahmed, A. D. (2017). Return and volatility spillovers effects: Evaluating the impact of shanghai-hong kong stock connect. *Economic Modelling*, 61:260–272.

Kalsi, J., Lyons, T., and Arribas, I. P. (2020). Optimal execution with rough path signatures. *SIAM Journal on Financial Mathematics*, 11(2):470–493.

Laborda, R. and Olmo, J. (2021). Volatility spillover between economic sectors in financial crisis prediction: Evidence spanning the great financial crisis and covid-19 pandemic. *Research in International Business and Finance*, 57:101402.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

Leippold, M., Wang, Q., and Zhou, W. (2022). Machine learning in the chinese stock market. *Journal of Financial Economics*, 145(2):64–82.

Li, C. and Liu, K. (2022). Path signature-based phase space reconstruction for stock trend prediction. *International Journal of Data Science and Analytics*, 14(3):293–304.

Li, S. Z. and Tang, Y. (2025). Automated volatility forecasting. *Management Science*, 71(7):6248–6274.

Li, X., Liang, C., and Ma, F. (2025a). Forecasting stock market volatility with a large number of predictors: New evidence from the ms-midas-lasso model. *Annals of Operations Research*, 352(3):613–652.

Li, Y., Qiao, Y., and Lei, S. (2025b). Ripple effect of esg sentiment: How news stirs the waves in china's a-share market. *International Review of Financial Analysis*, 97:103856.

Liang, C., Tang, L., Li, Y., and Wei, Y. (2020). Which sentiment index is more informative to forecast stock market volatility? evidence from china. *International Review of Financial Analysis*, 71:101552.

Liu, G., Zhuang, Z., and Wang, M. (2024a). Forecasting the high-frequency volatility based on the lstm-hit model. *Journal of Forecasting*, 43(5):1356–1373.

Liu, G., Zhuang, Z., and Wang, M. (2024b). Forecasting the high-frequency volatility based on the lstm-hit model. *Journal of Forecasting*, 43(5):1356–1373.

Liu, W. and Wen, Z. (2024). The time secret of chinese a-share systematic risk: Overnight and intraday. *Emerging Markets Finance and Trade*, 60(1):99–112.

Lyons, T. (2014). Rough paths, signatures and the modelling of functions on streams. *arXiv preprint arXiv:1405.4537*.

Lyons, T. J. (1998). Differential equations driven by rough signals. *Revista Matemática Iberoamericana*, 14(2):215–310.

Ma, F., Wang, J., Wahab, M., and Ma, Y. (2023). Stock market volatility predictability in a data-rich world: A new insight. *International Journal of Forecasting*, 39(4):1804–1819.

Mensi, W., Nekhili, R., Vo, X. V., Suleman, T., and Kang, S. H. (2021). Asymmetric volatility connectedness among us stock sectors. *The North American Journal of Economics and Finance*, 56:101327.

Moreno-Pino, F. and Zohren, S. (2024a). Deepvol: Volatility forecasting from high-frequency data with dilated causal convolutions. *Quantitative Finance*, 24(8):1105–1127.

Moreno-Pino, F. and Zohren, S. (2024b). Deepvol: Volatility forecasting from high-frequency data with dilated causal convolutions. *Quantitative Finance*, 24(8):1105–1127.

Mumtaz, H. and Theodoridis, K. (2017). Common and country specific economic uncertainty. *Journal of International Economics*, 105:205–216.

Namdari, A. and Durrani, T. S. (2021). A multilayer feedforward perceptron model in neural networks for predicting stock market short-term trends. *Operations Research Forum*, 2:38.

Newey, W. K. and West, K. D. (1987). A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix. *Econometrica*, 55(3):703–708.

Niu, Z., Wang, C., and Zhang, H. (2023). Forecasting stock market volatility with various geopolitical risks categories: New evidence from machine learning models. *International Review of Financial Analysis*, 89:102738.

Nobre, J. and Neves, R. F. (2019). Combining principal component analysis, discrete wavelet transform and xgboost to trade in the financial markets. *Expert Systems with Applications*, 125:181–194.

Park, H. J., Kim, Y., and Kim, H. Y. (2022). Stock market forecasting using a multi-task approach integrating long short-term memory and the random forest framework. *Applied Soft Computing*, 114:108106.

Patton, A. J. (2011). Volatility forecast comparison using imperfect volatility proxies. *Journal of econometrics*, 160(1):246–256.

Patton, A. J. and Sheppard, K. (2009). Evaluating volatility and correlation forecasts. In *Handbook of financial time series*, pages 801–838. Springer.

Poignard, B. and Asai, M. (2023). High-dimensional sparse multivariate stochastic volatility models. *Journal of Time Series Analysis*, 44(1):4–22.

Pradeepkumar, D. and Ravi, V. (2017). Forecasting financial time series volatility using particle swarm optimization trained quantile regression neural network. *Applied Soft Computing*, 58:35–52.

Qiao, G., Pan, Y., Liang, C., Wang, L., and Wang, J. (2024). Forecasting chinese crude oil futures volatility: New evidence based on dual feature processing of large-scale variables. *Journal of Forecasting*, 43(7):2495–2521.

Qiu, Y., Xie, T., Yu, J., and Zhou, Q. (2020). Forecasting equity index volatility by measuring the linkage among component stocks*. *Journal of Financial Econometrics*, 20(1):160–186.

Shu, Y., Yu, C., and Mulvey, J. M. (2025). Dynamic asset allocation with asset-specific regime forecasts. *Annals of Operations Research*, 346(1):285–318.

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

Song, Y., Lei, B., Tang, X., and Li, C. (2024). Volatility forecasting for stock market index based on complex network and hybrid deep learning model. *Journal of Forecasting*, 43(3):544–566.

Tian, G. G. and Guo, M. (2007). Interday and intraday volatility: Additional evidence from the shanghai stock exchange. *Review of Quantitative Finance and Accounting*, 28(3):287–306.

Tran, D. T., Iosifidis, A., Kanniainen, J., and Gabbouj, M. (2018). Temporal attention-augmented bilinear network for financial time-series data analysis. *IEEE transactions on neural networks and learning systems*, 30(5):1407–1418.

Wang, Z., Li, Y., and He, F. (2020). Asymmetric volatility spillovers between economic policy uncertainty and stock markets: Evidence from china. *Research in International Business and Finance*, 53:101233.

Wu, D., Ma, X., and Olson, D. L. (2022). Financial distress prediction using integrated z-score and multilayer perceptron neural networks. *Decision Support Systems*, 159:113814.

Wu, H. and Xie, Q. (2023). Volatility spillovers and asymmetric effects of chinese a-share markets—enterprise-level data based on high-dimensional social network models. *Applied Economics*, 56(57):7732–7756.

Wu, X., Zhao, A., Wang, Y., and Han, Y. (2024). Forecasting chinese stock market volatility with high-frequency intraday and current return information. *Pacific-Basin Finance Journal*, 86:102458.

Yang, Q., Yu, Y., Dai, D., He, Q., and Lin, Y. (2024). Can hybrid model improve the forecasting performance of stock price index amid covid-19? contextual evidence from the meemd-lstm-mlp approach. *The North American Journal of Economics and Finance*, 74:102252.

Yang, Z., Keung, J., Kabir, M. A., Yu, X., Tang, Y., Zhang, M., and Feng, S. (2021). Acomnn: Attention enhanced compound neural network for financial time-series forecasting with cross-regional features. *Applied Soft Computing*, 111:107649.

Yu, Y., Lin, Y., Hou, X., and Zhang, X. (2023). Novel optimization approach for realized volatility forecast of stock price index based on deep reinforcement learning model. *Expert Systems with Applications*, 233:120880.

Zhang, C., Zhang, Y., Cucuringu, M., and Qian, Z. (2024a). Volatility forecasting with machine learning and intraday commonality. *Journal of Financial Econometrics*, 22(2):492–530.

Zhang, Q., Zhang, P., and Zhou, F. (2022). Intraday and interday features in the high-frequency data: Pre-and post-crisis evidence in china's stock market. *Expert Systems with Applications*, 209:118321.

Zhang, Y., Ma, F., and Liao, Y. (2020a). Forecasting global equity market volatilities. *International Journal of Forecasting*, 36(4):1454–1475.

Zhang, Y., Ma, F., and Zhu, B. (2019). Intraday momentum and stock return predictability: Evidence from china. *Economic Modelling*, 76:319–329.

Zhang, Y., Song, Y., Peng, Y., and Wang, H. (2024b). Volatility forecasting incorporating intraday positive and negative jumps based on deep learning model. *Journal of Forecasting*, 43(7):2749–2765.

Zhang, Y., Yan, B., and Aasma, M. (2020b). A novel deep learning framework: Prediction and analysis of financial time series using ceemd and lstm. *Expert systems with applications*, 159:113609.

A Implementation and Hyperparameter Details

A.1 Data Splits and Rolling Evaluation

We adopt a strictly forward-looking rolling-window design. For each stock, the data between January 2019 and December 2024 are partitioned into a sequence of overlapping estimation and test windows. Each estimation window spans one calendar year and is split into an 80% training part and a 20% validation part, while the subsequent half-year forms the corresponding test window (see Figure 13). The window then shifts forward by half a year, and the procedure is repeated: models are re-estimated from scratch using only the current estimation window, all hyperparameter tuning is based on the associated validation split, and forecasts are generated for the following half-year test period. Test observations never enter the training or validation sets at any stage. The resulting forecast panel covers the period from January 2020 to December 2024.

A.2 Standardisation

To stabilise the numerical behaviour of the models and make the scale of inputs comparable across assets, RV is standardised on a stock-by-stock basis within each estimation window. For stock i and horizon h, let $\mu_{i,\text{train}}^{(h)}$ and $\sigma_{i,\text{train}}^{(h)}$ denote the sample mean and standard deviation of $RV_{i,t}^{(h)}$ computed using only the training portion of a given estimation window. The standardised predictor is then defined as

$$\widetilde{RV}_{i,t}^{(h)} = \frac{RV_{i,t}^{(h)} - \mu_{i,\text{train}}^{(h)}}{\sigma_{i,\text{train}}^{(h)}}. \tag{80}$$

The same transformation, based on $\mu_{i,\text{train}}^{(h)}$ and $\sigma_{i,\text{train}}^{(h)}$, is applied to the validation and test observations within that window. Stock-specific standardisation is widely used in high-dimensional forecasting and pooled machine-learning designs (e.g. Gu et al., 2020; Zhang et al., 2024a). It ensures comparability across assets and avoids distortions from differences in unconditional volatility levels. For pooled schemes (Universal, Augmented, Cluster), mini-batches may contain observations from multiple stocks and dates, but the time ordering of each stock is always respected.

A.3 Hyperparameter Tuning and Loss Function

Hyperparameters are tuned in a way that is standard in forecast-oriented machine-learning applications. The OLS and HAR benchmarks have no tuning parameters. For the LASSO and XGBoost models, we use grid search over pre-specified hyperparameter grids. For the neural-network models (MLP, LSTM, GRU, SA-LSTM, SA-GRU, MSA-LSTM, MSA-GRU), we rely on random search over continuous and discrete ranges. The search spaces are fixed ex ante and reported in Tables A.1 and A.2. Within each estimation window, candidate hyperparameter configurations are evaluated on the validation split, and the configuration that yields the lowest average loss on the validation observations is retained for that model class and horizon. The subsequent half-year test window is reserved exclusively for out-of-sample evaluation and does not enter the tuning procedure.

All neural-network models are trained under the QLIKE loss, which is the standard objective in realized-volatility forecasting and is widely recommended for comparing conditional variance

forecasts when volatility is observed only through noisy realised measures (Patton, 2011; Herrera et al., 2018). Compared with mean-squared-error (MSE) criteria, QLIKE is better aligned with the positive and highly heterogeneous nature of volatility and is less sensitive to measurement error in realized volatility proxies (Bollerslev et al., 2016). In robustness checks, we also experimented with training and evaluating models under MSE-style losses. The qualitative ranking of models is broadly similar, but the QLIKE-based specifications deliver slightly better out-of-sample performance, and QLIKE remains our primary loss function for both estimation and forecast comparison.

A.4 Hyperparameter Search Space

The hyperparameter grids for the linear and tree-based models are summarised in Table A.1. The OLS and HAR specifications use fixed lag structures and therefore require no tuning. The LASSO model employs a log-scale grid for its penalty parameter, and the XGBoost model uses conventional ranges for tree depth, learning rate, and regularisation-oriented parameters. These settings follow standard practice in forecast-oriented gradient-boosting applications and are designed to provide sufficient flexibility without excessive complexity.

Table A.2 reports the search spaces for the neural-network models. The MLP models vary in depth, width, dropout rate, learning rate, batch size, and weight decay. The LSTM and GRU models tune the number of recurrent layers, the hidden size, dropout, the learning rate, batch size, and weight decay, while keeping the output activation linear and applying a fixed gradient-clipping threshold. The SA-LSTM and SA-GRU architectures include additional hyperparameters related to the signature-based descriptor and attention mechanism, such as the signature truncation depth, the projection dimension, the number of attention heads, and the hidden dimension of the attention network. The MSA-LSTM and MSA-GRU models incorporate tuning parameters for the multi-scale convolutional block, including the number of filters per scale, the dimension of the multi-scale descriptor, and kernel sizes representing short-, medium-, and longer-run components. Across all neural-network models, batch sizes are chosen from {256, 512, 1024}, reflecting the large cross-sectional and time-series sample sizes.

Table A.1: Hyperparameter Search Space for Linear and Tree-Based Models

Model	Hyperparameter	Search Space / Setting
OLS	—	No hyperparameters
HAR	—	No hyperparameters
LASSO	Penalty λ	$\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$ (log-scale grid)
XGBoost	Number of trees (n_estimators)	$\{300, 500, 700, 900\}$
	Maximum depth (max_depth)	$\{3, 5, 7\}$
	Learning rate	$\{0.01, 0.05, 0.10\}$
	Subsample ratio	$\{0.6, 0.8, 1.0\}$
	Column subsampling	$\{0.6, 0.8, 1.0\}$

Table A.2: Hyperparameter Search Space for Neural Network Models

Model	Hyperparameter	Search Space / Setting
MLP	Hidden layers	$\{2, 3, 4\}$
	Hidden units per layer	$\{64, 128, 256, 512\}$
	Activation (hidden)	ReLU
	Activation (output)	Linear
	Dropout	$\{0.0, 0.2, 0.4\}$
	Learning rate	log-uniform $[10^{-4}, 5 \times 10^{-3}]$
	Batch size	$\{256, 512, 1024\}$
	Weight decay	$\{0, 10^{-5}, 10^{-4}\}$
LSTM / GRU	Recurrent layers	$\{1, 2, 3\}$
	Hidden size	$\{64, 128, 256\}$
	Dropout	$\{0.0, 0.2, 0.4\}$
	Learning rate	log-uniform $[10^{-4}, 5 \times 10^{-3}]$
	Batch size	$\{256, 512, 1024\}$
	Weight decay	$\{0, 10^{-5}, 10^{-4}\}$
	Gradient clipping	Global norm $= 5$ (fixed)
SA-LSTM / SA-GRU	Signature depth	$\{2, 3, 4\}$
	Projection dimension	$\{32, 64, 128\}$
	Attention heads	$\{1, 2\}$
	Attention hidden dimension	$\{32, 64, 128\}$
	Recurrent layers	$\{1, 2, 3\}$
	Hidden size	$\{64, 128, 256\}$
	Dropout	$\{0.0, 0.2, 0.4\}$
	Learning rate	log-uniform $[10^{-4}, 5 \times 10^{-3}]$
	Batch size	$\{256, 512, 1024\}$
	Weight decay	$\{0, 10^{-5}, 10^{-4}\}$
MSA-LSTM / MSA-GRU	Kernel sizes	$\{3, 5, 10\}$ (fixed set)
	Filters per scale	$\{8, 16, 32\}$
	Descriptor dimension	$\{32, 64, 128\}$
	Recurrent layers	$\{1, 2, 3\}$
	Hidden size	$\{64, 128, 256\}$
	Dropout	$\{0.0, 0.2, 0.4\}$
	Learning rate	log-uniform $[10^{-4}, 5 \times 10^{-3}]$
	Batch size	$\{256, 512, 1024\}$
	Weight decay	$\{0, 10^{-5}, 10^{-4}\}$

A.5 Optimisation and Training Schedule

All neural-network models are trained with the Adam optimiser using default momentum parameters. The learning rate, batch size, and weight decay are selected as part of the hyperparameter search. A learning-rate scheduler monitors the validation loss within each estimation window and decreases the learning rate when the loss fails to improve for a fixed number of epochs. Training proceeds for up to 300 epochs, with early stopping triggered after a patience period without validation improvement. Regularisation combines dropout, L2 weight decay, and gradient clipping with a fixed global-norm threshold. These optimisation settings follow standard practice in deep time-series forecasting and are held constant across all training schemes, conditional on the tuned hyperparameters.

Multiple random seeds are used to stabilise training. For each hyperparameter configuration, the model is trained under three different random initialisations, validation losses are averaged when comparing candidate configurations, and out-of-sample forecasts are obtained by averaging the predictions from the three resulting networks.

A.6 Software and Reproducibility

All forecasting experiments are implemented in Python. Neural-network models are estimated using a modern deep-learning framework, while linear and tree-based models rely on widely used machine-learning libraries. Computations are run on a GPU-enabled environment, though the procedures do not depend on any hardware-specific features. To enhance reproducibility, random seeds are set for Python, NumPy, and the deep-learning library, and the same data-splitting, validation, tuning, and forecasting protocols are applied uniformly across all windows, models, horizons, and training schemes.

B Additional Details for Forecast Evaluation Metrics

This appendix provides further details for the loss functions and utility-based evaluation measures introduced in Section 5.1.

B.1 Log-QLIKE for log-realized volatility

The standard QLIKE loss is defined for level variances and takes the form

$$L_{\mathrm{QLIKE}}(RV, \widehat{RV}) = \frac{RV}{\widehat{RV}} - \log\left(\frac{RV}{\widehat{RV}}\right) - 1.$$

Because we work exclusively with log-realized volatility, the forecast target is $RV = \log(RV^{\mathrm{raw}})$. Following Bucci (2020); Zhang et al. (2024a), we adopt the likelihood-equivalent loss for the log specification:

$$L^{\log}_{\mathrm{QLIKE}} = \exp(RV - \widehat{RV}) - (RV - \widehat{RV}) - 1.$$

This transformation maintains the desirable properties emphasised by Patton and Sheppard (2009), including robustness to noise in realised variance and strict consistency with likelihood-based fore-

cast comparison.

B.2 Out-of-sample R^2

The out-of-sample R^2 used in our evaluation differs from the in-sample coefficient of determination commonly reported in regression analysis. It compares a model's forecast errors with those of a naive mean forecast and is defined as

$$R^2_{\text{oos}} = 1 - \frac{\sum_{i,t}(RV_{i,t} - \widehat{RV}_{i,t})^2}{\sum_{i,t}(RV_{i,t} - \bar{RV}_i)^2},$$

where $\bar{RV}_i = \frac{1}{T_{\text{test}}} \sum_t RV_{i,t}$ is the average realized volatility for asset i computed over the test period.

Because $\bar{RV}_i$ is unknown ex ante and is constructed from the same out-of-sample observations, the denominator does not correspond to the total variation around a fixed population mean. Consequently, R^2_{oos} is not constrained to the $[0, 1]$ interval: it may take negative values when a model have relative poor performance. This property is standard for forecast-based R^2 measures and reflects their role as relative performance metrics rather than measures of explained variation.

C Utility-Based Evaluation: Derivations and Implementation

This section provides the derivation of the realised utility (RU) and risk-targeted utility (RU_TC) measures used in the empirical analysis. The exposition follows the mean–variance approximation for CRRA utility commonly employed in financial econometrics.

C.1 Conditional expected utility

Let wealth evolve according to

$$W_{t+1} = W_t \left(1 + r^f_t + x_t r^e_{t+1}\right),$$

where r^e_{t+1} is the excess return on the risky asset and x_t is the portfolio weight. For a CRRA investor with utility $u(W)$, the second-order Taylor expansion around $\mathbb{E}_t(W_{t+1})$ yields

$$\mathbb{E}_t[u(W_{t+1})] \approx W_t \left[x_t \, \mathbb{E}_t(r^e_{t+1}) - \frac{\gamma}{2} \, x_t^2 \, \mathbb{E}_t(r^{e\,2}_{t+1})\right],$$

where γ is relative risk aversion and $\mathbb{E}_t(r^{e\,2}_{t+1})$ is approximated using $\mathbb{E}_t(RV_{t+1})$.

C.2 Optimal position

Maximising the quadratic approximation yields

$$x^*_t = \frac{\mathbb{E}_t(r^e_{t+1})}{\gamma \, \mathbb{E}_t(RV_{t+1})}.$$

Letting

$$SR = \frac{\mathbb{E}_t(r^e_{t+1})}{\sqrt{\mathbb{E}_t(RV_{t+1})}}$$

denote the conditional Sharpe ratio (assumed constant), we obtain the risk-scaling rule

$$x^*_t = \frac{SR}{\gamma} \frac{1}{\sqrt{\mathbb{E}_t(RV_{t+1})}}.$$

The corresponding conditional utility is

$$U_t(x^*_t) = \frac{SR^2}{2\gamma} W_t,$$

illustrating that more accurate volatility forecasts raise realised utility by improving risk scaling.

C.3 Realised utility (RU) and transaction-cost-adjusted utility (RU-TC)

This subsection collects the derivations underlying the realised utility measures reported in the main text. The starting point is a representative investor with CRRA preferences,

$$u(W) = \frac{W^{1-\gamma}}{1-\gamma}, \qquad \gamma > 0,$$

who allocates a fraction x_t of wealth to a risky asset with excess return r^e_{t+1} and holds the remainder in a risk-free asset.

Mean–variance approximation and optimal position. Let wealth evolve according to

$$W_{t+1} = W_t\big(1 + x_t r^e_{t+1}\big).$$

A second-order expansion of $\mathbb{E}_t[u(W_{t+1})]$ around $\mathbb{E}_t(r^e_{t+1})$ yields the familiar mean–variance approximation (see, e.g., Bollerslev et al., 2018):

$$\mathbb{E}_t[u(W_{t+1})] \approx u(W_t) + W_t \left[x_t\, \mathbb{E}_t(r^e_{t+1}) - \frac{\gamma}{2}\, x_t^2\, \mathbb{E}_t(r^{e\,2}_{t+1})\right]. \tag{81}$$

Suppressing the constant $u(W_t)$, the investor maximises

$$\widetilde{U}_t(x_t) = x_t\, \mu_t - \frac{\gamma}{2}\, x_t^2\, \sigma_t^2,$$

where $\mu_t = \mathbb{E}_t(r^e_{t+1})$ and $\sigma_t^2 = \mathbb{E}_t(r^{e\,2}_{t+1})$. The first-order condition yields the optimal position

$$x^*_t = \frac{\mu_t}{\gamma\, \sigma_t^2}. \tag{82}$$

Define the conditional Sharpe ratio as

$$SR_t = \frac{\mu_t}{\sigma_t}.$$

In line with the empirical literature on volatility forecasting, we assume that the Sharpe ratio is constant over time and equal to a fixed value SR (e.g. Bollerslev et al., 2018; Zhang et al., 2024a; Li and Tang, 2025). Under this calibration, $\mu_t = SR\,\sigma_t$ and (82) becomes

$$x_t^* = \frac{SR}{\gamma}\frac{1}{\sigma_t}. \tag{83}$$

Substituting (83) into $\widetilde{U}_t(x_t)$ yields a closed-form expression for the maximised expected utility. Using $\mu_t = SR \cdot \sigma_t$ and $x_t^* = SR/(\gamma\sigma_t)$, straightforward algebra gives

$$\widetilde{U}_t(x_t^*) = \frac{SR^2}{2\gamma}. \tag{84}$$

This value does not depend on σ_t, reflecting the fact that once the conditional Sharpe ratio is fixed, the optimal mean–variance trade-off implied by the quadratic approximation collapses to a constant. This constant benchmark is the starting point for the realised-utility expressions used in the main text.

Realised utility (RU). In the empirical implementation, conditional volatility is proxied by realized volatility. Let RV_{t+1} denote the ex post realized volatility and let $\widehat{RV}_{t+1}$ be the model's forecast for $\mathbb{E}_t(RV_{t+1})$. We work with $\sigma_t \approx \sqrt{RV_{t+1}}$ ex post and $\widehat{\sigma}_t \approx \sqrt{\widehat{RV}_{t+1}}$ ex ante. Replacing conditional moments by these proxies and averaging over the out-of-sample period yields the realised utility measure,

$$\mathrm{RU} = \frac{1}{\#\mathcal{T}_{\text{test}}} \sum_{t \in \#\mathcal{T}_{\text{test}}} \left(\frac{SR}{\gamma} \frac{\sqrt{RV_{t+1}}}{\sqrt{\widehat{RV}_{t+1}}} - \frac{SR^2}{2\gamma} \frac{RV_{t+1}}{\widehat{RV}_{t+1}} \right), \tag{85}$$

which corresponds to the expression reported in Section 5.1. In the panel setting, we compute RU by averaging (85) across stocks:

$$\mathrm{RU} = \frac{1}{N \cdot \#\mathcal{T}_{\text{test}}} \sum_{i=1}^{N} \sum_{t \in \#\mathcal{T}_{\text{test}}} \left(\frac{SR}{\gamma} \frac{\sqrt{RV_{i,t+1}}}{\sqrt{\widehat{RV}_{i,t+1}}} - \frac{SR^2}{2\gamma} \frac{RV_{i,t+1}}{\widehat{RV}_{i,t+1}} \right). \tag{86}$$

In the main analysis we fix $SR = 0.4$ and $\gamma = 2$, in line with Bollerslev et al. (2018), Christensen et al. (2022), and Zhang et al. (2024a), so that the RU values are directly comparable to existing studies.

Transaction-cost-adjusted utility (RU-TC). Volatility forecasts also affect the turnover of the optimal portfolio. To account for this, we compute a transaction-cost-adjusted utility measure, denoted RU-TC. Following Zhang et al. (2024a) and Li and Tang (2025), we assume that transaction costs are linear in the absolute change in positions. For each stock i, the proportional cost parameter κ_i is taken as the full median bid–ask spread over the last 90 trading days, which summarises typical trading frictions while smoothing out short-lived microstructure noise.

Let $x_{i,t}^*$ denote the optimal position implied by (83) using the forecast $\widehat{RV}_{i,t+1}$ for stock i at

time t. The trading cost associated with adjusting the position from $x^*_{i,t-1}$ to $x^*_{i,t}$ is

$$\mathrm{TC}_{i,t} = \kappa_i \left| x^*_{i,t} - x^*_{i,t-1} \right|. \tag{87}$$

The transaction-cost-adjusted realised utility is then defined as

$$\text{RU-TC} = \frac{1}{N \cdot \#\mathcal{T}_{\text{test}}} \sum_{i=1}^{N} \sum_{t \in \#\mathcal{T}_{\text{test}}} \left[\left(\frac{SR}{\gamma} \frac{\sqrt{RV_{i,t+1}}}{\sqrt{\widehat{RV}_{i,t+1}}} - \frac{SR^2}{2\gamma} \frac{RV_{i,t+1}}{\widehat{RV}_{i,t+1}} \right) - \mathrm{TC}_{i,t} \right]. \tag{88}$$

By construction, RU measures the welfare gains from improved risk scaling in the absence of trading frictions, whereas RU-TC captures the net benefit once portfolio turnover and transaction costs are taken into account. Both measures are used in the empirical analysis to assess the economic value of competing volatility forecasts.

D HAC Variance Estimation and the DM Statistic

This appendix provides the details for the heteroskedasticity–autocorrelation consistent (HAC) variance estimator used in the Diebold–Mariano tests.

D.1 Autocovariance structure

Let d_t denote the loss differential between two forecasting models. Rolling forecasts and overlapping horizons induce serial correlation in $\{d_t\}$. Define the sample autocovariance at lag k:

$$\hat{\gamma}(k) = \frac{1}{T_{\text{test}}} \sum_{t=k+1}^{T_{\text{test}}} (d_t - \bar{d})(d_{t-k} - \bar{d}), \quad \bar{d} = \frac{1}{T_{\text{test}}} \sum_{t} d_t.$$

The variance of $\bar{d}$ depends on the full set of autocovariances, not only on $\hat{\gamma}(0)$. Thus a long-run variance estimator is required.

D.2 Newey–West long-run variance

We estimate the variance of $\bar{d}$ using the Newey–West estimator (Newey and West, 1987; Andrews, 1991):

$$\widehat{\mathrm{Var}}(\bar{d}) = \frac{1}{T_{\text{test}}} \left[\hat{\gamma}(0) + 2 \sum_{k=1}^{K} \left(1 - \frac{k}{K+1} \right) \hat{\gamma}(k) \right].$$

The Bartlett kernel $1 - k/(K+1)$ ensures non-negative definiteness. We set the truncation lag to $K = \lfloor T_{\text{test}}^{1/3} \rfloor$, a standard choice that balances bias and variance. This estimator is consistent for the long-run variance of $\bar{d}$ under general forms of heteroskedasticity and autocorrelation.

D.3 DM statistic

The DM statistic is

$$DM = \frac{\bar{d}}{\sqrt{\widehat{\mathrm{Var}}(\bar{d})}},$$

with the one-sided alternative $H_1 : \mathbb{E}(d_t) < 0$. Under mild conditions, DM is asymptotically standard normal.

E Model Confidence Set (MCS) Procedure

This appendix summarises the implementation of the MCS procedure of Hansen et al. (2011).

E.1 Equal predictive ability and loss differences

Let $\mathcal{M}$ denote the set of competing models. For each pair (m, n), define the loss differential

$$d_{mn,t} = L_{m,t} - L_{n,t}.$$

The null hypothesis of equal predictive ability (EPA) is

$$H_0 : \ \mathbb{E}(d_{mn,t}) = 0 \quad \text{for all } m, n \in \mathcal{M}.$$

E.2 Test statistics

The MCS procedure evaluates H_0 using either the $T_{\max}$ or T_R statistic. Define

$$\bar{d}_{mn} = \frac{1}{T_{\text{test}}} \sum_t d_{mn,t},$$

and let $\widehat{\mathrm{Var}}(\bar{d}_{mn})$ be a HAC estimator computed as in Appendix D. The statistics are

$$T_{\max} = \max_{m,n} \frac{|\bar{d}_{mn}|}{\sqrt{\widehat{\mathrm{Var}}(\bar{d}_{mn})}},$$

$$T_R = \frac{1}{|\mathcal{M}|} \sum_{m,n} \frac{|\bar{d}_{mn}|}{\sqrt{\widehat{\mathrm{Var}}(\bar{d}_{mn})}}.$$

E.3 Bootstrap and sequential elimination

Critical values are obtained by a block bootstrap to preserve the serial dependence of losses. If H_0 is rejected, the worst-performing model—defined as the one with the largest average loss relative to the others—is removed, and the test is repeated on the reduced model set. Iteration continues until the EPA null cannot be rejected. The remaining set is the superior set of models, denoted $\widehat{\mathcal{M}}^*$.

F Horizon-Specific Rank Correlation Matrices

This appendix reports the horizon-specific Spearman rank correlation matrices underlying the summary heatmap in Figure 14. The four panels in Figure F.1 correspond to the 10-, 30-, 60-, and 240-minute intervals. Although the overall dependence structure of the evaluation metrics is stable across horizons, several horizon-specific features are worth highlighting.

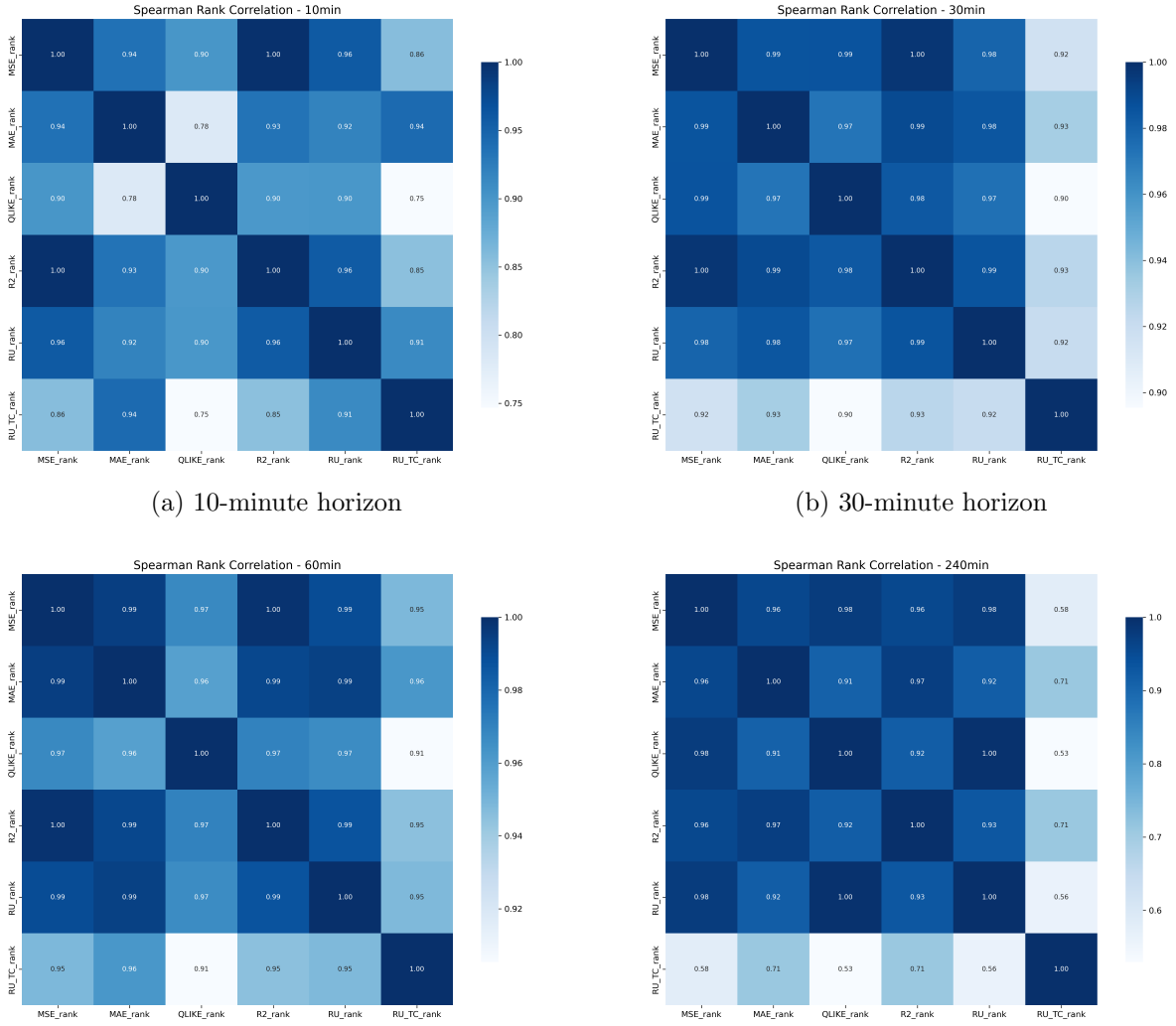
10-minute horizon. At the highest sampling frequency (Figure F.1a), the correlation structure is broadly strong but noticeably more heterogeneous than at longer horizons. MSE and MAE remain almost perfectly correlated (0.94), and both align closely with QLIKE (0.90). These three error-based metrics show the expected strong negative relationship with R^2 and the utility measures (correlations around -0.90). However, the realised utility after transaction costs (RU-TC) exhibits weaker associations with the error-based metrics (0.75–0.86), reflecting the fact that transaction-cost adjustments introduce additional noise at very high frequencies. This horizon shows the lowest cross-metric coherence among all four intervals, consistent with the greater role of microstructure noise and the more volatile nature of ultra-short horizon forecasts.

30-minute horizon. At 30 minutes (Figure F.1b), the correlation matrix becomes substantially tighter. Error-based metrics are nearly perfectly correlated (0.97–0.99), and their relationships with R^2, RU, and RU-TC strengthen considerably (absolute correlations around 0.90–0.99). This horizon shows highly uniform dependence patterns, indicating that different evaluation criteria produce almost identical orderings of model performance. The higher coherence reflects the reduction in microstructure-induced distortions while preserving sufficient intraday variation to identify differences across models.

60-minute horizon. The 60-minute horizon (Figure F.1c) delivers the strongest and most stable correlation structure among all intervals. Virtually all pairwise correlations exceed 0.95 in absolute value, including those involving RU-TC. QLIKE correlates at 0.96–0.97 with MSE and MAE, and the utility-based metrics are almost perfectly aligned with R^2. This horizon exhibits the cleanest one-factor structure across metrics, consistent with the fact that the 60-minute frequency maximises the strength of market-wide volatility commonality and reduces the influence of high-frequency noise. Accordingly, all metrics deliver almost indistinguishable rankings of model–scheme combinations.

240-minute (daily) horizon. At the daily horizon (Figure F.1d), the overall dependence pattern remains strong, but RU-TC again becomes an outlier, with markedly lower correlations (0.53–0.71) relative to QLIKE, MSE, and R^2. This weakening reflects the fact that transaction costs accumulate proportionally with position changes; at lower sampling frequencies, daily rebalancing generates utility patterns that differ somewhat from statistical forecast accuracy. Excluding RU-TC, the remaining metrics retain very high correlations (0.90–0.99), suggesting that forecast accuracy and utility remain tightly connected at the daily horizon.

Cross-horizon comparison. Taken together, the heatmaps show that metric consistency improves systematically from 10 minutes to 60 minutes, peaking at the 60-minute horizon, and weakens slightly at the 240-minute horizon, primarily due to RU-TC. The high coherence at 30–60 minutes aligns with the earlier evidence on market-level volatility commonality, which is strongest at intermediate frequencies. As a result, all six metrics deliver nearly identical model rankings at these horizons. Even at 10 and 240 minutes—where microstructure noise or transaction-cost effects weaken a few correlations—the direction of all relationships remains consistent. Overall, the horizon-specific matrices reinforce the conclusion drawn in Section 5.3: different evaluation criteria provide a highly robust and mutually consistent ordering of forecasting models.



(a) 10-minute horizon

(b) 30-minute horizon

(c) 60-minute horizon

(d) 240-minute horizon

Figure F.1: Spearman rank correlation matrices for each intraday horizon. The structure of correlations is highly consistent across horizons: error-based metrics (MSE, MAE, QLIKE) are strongly positively correlated; these metrics are strongly negatively correlated with accuracy and utility metrics (R^2, RU, RU-TC); and the utility-based metrics themselves exhibit strong positive correlation.

G Detailed DM test matrices

This appendix reports the full pairwise Diebold–Mariano (DM) statistic matrices (see Tables from G.1 to G.4) underpinning the dominance counts in Table 7. All statistics are based on QLIKE loss differentials. For the square matrices, let L^{row} and L^{col} denote the QLIKE losses of the two training schemes being compared. Each entry reports the two-sided DM statistic for testing $H_0 : \mathbb{E}(L^{\text{row}} - L^{\text{col}}) = 0$ based on the loss differential $d_t = L_t^{\text{row}} - L_t^{\text{col}}$. A positive value therefore indicates lower loss for the scheme listed in the row (the row scheme outperforms the column scheme), whereas a negative value indicates the opposite. In each panel, the final row labelled "Single vs." reports DM statistics for the loss differential $L^{\text{Single}} - L^{\text{scheme}}$; positive values in this row thus indicate that the pooled scheme (Universal, Augmented, or Cluster) delivers lower QLIKE loss than stock-specific training for the corresponding architecture. Stars denote rejection of equal predictive ability at the 10%, 5%, and 1% significance levels (*, **, and ***, respectively). Taken together, the matrices confirm a clear and robust ranking of the training schemes. Pooling across assets almost never deteriorates performance and typically delivers sizeable reductions in QLIKE relative to stock-specific estimation. Cluster-based training yields the most systematic gains, followed by the augmented and universal schemes, in line with the dominance patterns reported in the main text. The improvements are particularly pronounced for the SA- and MSA-based recurrent architectures, while more parsimonious linear benchmarks display smaller but still consistent benefits from cross-sectional information.

10-minute horizon (Table G.1). At the 10-minute horizon, the DM statistics indicate that all three pooled schemes substantially outperform stock-specific training for the majority of model architectures. Cluster-based training records the most frequent and most pronounced rejections of equal predictive accuracy, with augmented and universal pooling also delivering strong and widespread gains. The largest improvements arise for the neural-network specifications, especially the SA- and MSA-augmented recurrent models, reflecting their ability to exploit rich cross-sectional signals at very short horizons.

30-minute horizon (Table G.2). For the 30-minute horizon, the matrices continue to show broad DM support for pooled training relative to stock-specific models. Cluster-based estimation remains the most dominant scheme, with augmented and universal pooling also achieving significant reductions in QLIKE for most architectures. Again, the gains are strongest for the recurrent and signature-based networks, while linear benchmarks exhibit more moderate but still frequent improvements.

60-minute horizon (Table G.3). At the 60-minute horizon, the relative advantages of the pooled schemes remain evident, although the magnitude of the DM statistics declines compared with shorter horizons. Cluster-based training continues to outperform stock-specific estimation for most models, and the augmented and universal schemes also register a high share of significant DM rejections in their favour. The SA- and MSA-based recurrent architectures still account for

a large portion of these wins, suggesting that they are able to translate additional cross-sectional information into more accurate medium-horizon forecasts.

240-minute (daily) horizon (Table G.4). For the 240-minute (daily) horizon, the matrices show that cross-sectional pooling continues to be beneficial, but with smaller DM values and fewer very large statistics than at intraday horizons. Cluster-based training remains the most robust performer, while augmented and universal pooling also tend to improve upon stock-specific estimation in a majority of cases. The recurrent and signature-augmented models continue to realise noticeable gains, indicating that exploiting volatility commonality is helpful even at the daily horizon, albeit with reduced incremental value compared with higher-frequency settings.

Table G.1: DM test matrices at the 10-minute horizon

(1) Universal vs. Single

Universal \ Universal	ols	lasso	hard	xgb	mlp	lstm	gru	salstm	sagru	msalstm	msagru
ols		4.14***	6.78***	-3.42***	28.19***	28.25***	31.15***	27.46***	30.33***	27.01***	29.86***
lasso			4.31***	-4.50***	23.81***	25.14***	27.90***	22.93***	25.55***	21.89***	24.44***
hard				-11.70***	26.46***	23.72***	27.45***	21.23***	24.82***	20.01***	23.51***
xgb					32.47***	26.40***	29.39***	24.12***	26.83***	23.11***	25.70***
mlp						-2.40**	1.44	-6.77***	-3.14***	-8.60***	-5.09***
lstm							9.50***	-25.08***	-1.34	-25.08***	-5.32***
gru								-18.78***	-27.37***	-21.49***	-27.37***
salstm									9.50***	-25.08***	4.72***
sagru										-13.96***	-27.37***
msalstm											9.50***
msagru											
Single vs.	8.23***	9.74***	7.23***	-10.18***	21.53***	25.08***	27.37***	33.04***	30.30***	24.97***	18.20***

(2) Augmented vs. Single

Augmented \ Augmented	ols	lasso	hard	xgb	mlp	lstm	gru	salstm	sagru	msalstm	msagru
ols		17.31***	-0.68	-7.12***	31.84***	20.16***	23.81***	17.38***	20.88***	16.16***	19.58***
lasso			-14.78***	-20.61***	10.45***	4.36***	7.63***	0.53	3.51***	-0.89	1.97**
hard				-6.69***	30.24***	19.95***	23.50***	16.87***	20.18***	15.58***	18.77***
xgb					42.29***	28.16***	32.34***	24.00***	27.69***	22.34***	25.81***
mlp						-9.48***	-5.83***	-14.14***	-10.90***	-15.77***	-12.71***
lstm							9.50***	-25.08***	-2.37**	-25.08***	-6.24***
gru								-19.52***	-27.37***	-22.06***	-27.37***
salstm									9.50***	-25.08***	4.66***
sagru										-14.01***	-27.37***
msalstm											9.50***
msagru											
Single vs.	-0.94	-7.69***	3.48***	-24.83***	17.50***	25.08***	27.37***	33.69***	31.23***	25.64***	19.40***

(3) Cluster vs. Single

Cluster \ Cluster	ols	lasso	hard	xgb	mlp	lstm	gru	salstm	sagru	msalstm	msagru
ols		4.22***	11.61***	5.49***	29.92***	24.58***	26.77***	23.20***	25.25***	22.65***	24.65***
lasso			10.18***	4.51***	28.32***	23.34***	25.52***	21.73***	23.76***	21.11***	23.08***
hard				-7.87***	33.19***	19.16***	22.67***	15.76***	19.00***	14.46***	17.58***
xgb					39.90***	25.97***	29.83***	21.79***	25.07***	20.30***	23.38***
mlp						-11.76***	-7.85***	-17.17***	-13.74***	-18.87***	-15.62***
lstm							9.50***	-25.08***	-3.39***	-25.08***	-7.13***
gru								-20.22***	-27.37***	-22.60***	-27.37***
salstm									9.50***	-25.08***	4.60***
sagru										-14.06***	-27.37***
msalstm											9.50***
msagru											
Single vs.	11.22***	13.00***	4.02***	-23.99***	18.61***	25.08***	27.37***	34.07***	31.88***	26.14***	20.39***

Table G.2: DM test matrices at the 30-minute horizon

(1) Universal vs. Single

Universal \ Universal	ols	lasso	hard	xgb	mlp	lstm	gru	salstm	sagru	msalstm	msagru
ols		4.27***	6.48***	-0.69	11.20***	13.46***	13.55***	13.34***	13.42***	13.23***	13.30***
lasso			4.76***	-1.61	9.09***	12.18***	12.23***	11.83***	11.86***	11.55***	11.58***
hard				-7.17***	6.22***	10.73***	10.83***	10.08***	10.15***	9.55***	9.60***
xgb					11.18***	16.12***	16.02***	15.17***	15.06***	14.49***	14.37***
mlp						5.83***	5.78***	4.70***	4.62***	3.83***	3.73***
lstm							-1.05	-11.76***	-8.19***	-11.76***	-9.82***
gru								-8.20***	-11.84***	-10.41***	-11.84***
salstm									-1.05	-11.76***	-6.89***
sagru										-6.28***	-11.84***
msalstm											-1.05
msagru											
Single vs.	7.73***	7.28***	4.25***	2.02**	9.25***	8.98***	7.50***	22.41***	22.97***	13.10***	11.01***

(2) Augmented vs. Single

Augmented \ Augmented	ols	lasso	hard	xgb	mlp	lstm	gru	salstm	sagru	msalstm	msagru
ols		5.32***	7.11***	-9.07***	11.09***	7.24***	7.77***	6.29***	6.79***	5.72***	6.20***
lasso			3.08***	-11.50***	8.14***	5.31***	5.85***	4.07***	4.58***	3.35***	3.83***
hard				-13.25***	5.02***	2.64***	3.09***	1.22	1.63	0.46	0.85
xgb					20.10***	23.05***	23.69***	21.45***	21.91***	20.39***	20.77***
mlp						-2.77***	-2.30**	-4.91***	-4.46***	-6.05***	-5.61***
lstm							3.12***	-11.51***	-5.59***	-11.51***	-7.49***
gru								-11.82***	-11.97***	-12.82***	-11.97***
salstm									3.12***	-11.51***	-2.60***
sagru										-9.12***	-11.97***
msalstm											3.12***
msagru											
Single vs.	-2.25**	-0.39	-5.37***	-0.81	6.51***	9.28***	7.40***	22.29***	22.42***	13.36***	11.01***

(3) Cluster vs. Single

Cluster \ Cluster	ols	lasso	hard	xgb	mlp	lstm	gru	salstm	sagru	msalstm	msagru
ols		-0.04	8.10***	-1.18	13.21***	11.85***	11.97***	11.43***	11.54***	11.23***	11.34***
lasso			7.99***	-1.17	13.09***	11.85***	11.97***	11.39***	11.49***	11.17***	11.27***
hard				-13.93***	8.68***	6.84***	7.07***	5.37***	5.57***	4.70***	4.89***
xgb					24.42***	28.70***	28.84***	26.13***	26.24***	24.87***	24.97***
mlp						-0.95	-0.73	-3.46***	-3.25***	-4.55***	-4.34***
lstm							1.24	-11.66***	-8.01***	-11.66***	-9.33***
gru								-9.86***	-11.80***	-10.84***	-11.80***
salstm									1.24	-11.66***	-4.14***
sagru										-6.47***	-11.80***
msalstm											1.24
msagru											
Single vs.	7.22***	8.44***	0.45	9.69***	7.29***	9.40***	7.70***	22.00***	22.27***	13.35***	11.25***

Table G.3: DM test matrices at the 60-minute horizon

(1) Universal vs. Single

Universal \ Universal	ols	lasso	hard	xgb	mlp	lstm	gru	salstm	sagru	msalstm	msagru
ols		-1.82*	8.73***	0.41	17.40***	14.16***	14.79***	12.01***	12.23***	13.92***	14.81***
lasso			9.00***	0.74	16.72***	14.32***	14.95***	12.36***	12.51***	14.10***	14.92***
hard				-6.95***	7.82***	5.48***	6.53***	3.49***	3.48***	6.38***	7.28***
xgb					10.62***	10.11***	10.68***	9.70***	9.48***	11.04***	11.39***
mlp						-3.86***	-2.98***	-5.64***	-5.73***	-2.22**	-1.58
lstm							3.66***	-5.85***	-6.82***	3.22***	6.14***
gru								-8.40***	-9.77***	1.07	3.95***
salstm									-0.05	8.16***	10.96***
sagru										8.84***	12.68***
msalstm											2.64***
msagru											
Single vs.	9.51***	11.40***	4.05***	1.49	6.92***	10.25***	8.36***	22.01***	21.75***	13.45***	15.09***

(2) Augmented vs. Single

Augmented \ Augmented	ols	lasso	hard	xgb	mlp	lstm	gru	salstm	sagru	msalstm	msagru
ols		-0.42	6.71***	-7.62***	8.37***	1.83*	2.66***	-0.08	0.71	3.73***	3.12***
lasso			5.78***	-7.58***	8.44***	2.12**	2.92***	0.16	0.97	3.97***	3.33***
hard				-11.16***	2.70***	-3.20***	-2.62***	-4.88***	-4.15***	-1.20	-1.94*
xgb					14.65***	11.94***	12.49***	10.49***	11.21***	13.42***	12.62***
mlp						-8.01***	-7.45***	-9.65***	-8.93***	-4.93***	-6.11***
lstm							2.86***	-5.46***	-3.76***	5.29***	3.82***
gru								-7.74***	-6.52***	3.77***	2.11**
salstm									3.59***	10.14***	8.27***
sagru										8.65***	7.06***
msalstm											-2.31**
msagru											
Single vs.	-2.24**	2.82***	-4.61***	-4.48***	5.28***	10.09***	8.26***	21.81***	21.06***	12.75***	15.12***

(3) Cluster vs. Single

Cluster \ Cluster	ols	lasso	hard	xgb	mlp	lstm	gru	salstm	sagru	msalstm	msagru
ols		1.62	7.66***	0.09	12.84***	10.28***	10.53***	8.89***	9.41***	10.88***	10.53***
lasso			7.43***	-0.18	12.45***	10.09***	10.34***	8.68***	9.18***	10.71***	10.32***
hard				-10.35***	5.72***	2.60***	3.14***	0.71	1.18	4.52***	3.62***
xgb					13.67***	12.97***	13.60***	12.12***	12.24***	14.67***	14.11***
mlp						-4.27***	-3.77***	-6.16***	-5.98***	-1.53	-2.74***
lstm							1.71*	-5.92***	-5.60***	4.94***	3.01***
gru								-6.85***	-7.53***	4.18***	1.98**
salstm									1.77*	9.13***	8.72***
sagru										8.96***	8.35***
msalstm											-2.84***
msagru											
Single vs.	7.10***	8.66***	1.51	2.83***	6.77***	10.11***	8.32***	21.75***	20.96***	12.35***	15.12***

Table G.4: DM test matrices at the 240-minute (daily) horizon

(1) Universal vs. Single

Universal \ Universal	ols	lasso	hard	xgb	mlp	lstm	gru	salstm	sagru	msalstm	msagru
ols		11.52***	7.90***	-0.93	13.05***	4.84***	6.03***	2.09**	3.67***	6.59***	7.21***
lasso			4.54***	-2.67***	10.45***	3.08***	4.07***	0.30	1.64	4.73***	5.14***
hard				-6.88***	10.29***	0.78	1.64	-2.81***	-1.52	2.62***	2.86***
xgb					13.94***	5.76***	6.64***	4.23***	5.26***	7.47***	7.72***
mlp						-3.85***	-3.58***	-8.15***	-7.21***	-2.42**	-2.70***
lstm							1.12	-3.67***	-2.62***	2.18**	2.28**
gru								-4.63***	-3.74***	1.41	1.46
salstm									2.36**	5.63***	5.77***
sagru										4.68***	4.98***
msalstm											-0.09
msagru											
Single vs.	9.95***	4.60***	5.36***	4.76***	-0.78	6.76***	6.47***	8.97***	6.58***	4.30***	6.32***

(2) Augmented vs. Single

Augmented \ Augmented	ols	lasso	hard	xgb	mlp	lstm	gru	salstm	sagru	msalstm	msagru
ols		-4.96***	2.92***	-8.86***	9.08***	-1.70*	-0.85	-3.91***	-4.26***	0.62	1.64
lasso			6.16***	-7.77***	11.38***	-0.13	0.91	-2.30**	-2.38**	2.30**	3.46***
hard				-10.70***	7.57***	-2.57**	-1.85*	-4.98***	-5.26***	-0.41	0.46
xgb					20.11***	4.17***	5.59***	3.34***	3.47***	6.81***	8.07***
mlp						-6.07***	-5.83***	-9.55***	-9.89***	-4.35***	-3.86***
lstm							1.63	-2.20**	-2.29**	3.17***	4.16***
gru								-3.49***	-3.92***	1.93*	3.23***
salstm									0.17	5.09***	6.02***
sagru										5.53***	6.78***
msalstm											1.37
msagru											
Single vs.	-2.13**	0.78	-3.10***	1.15	-3.88***	7.32***	6.84***	7.85***	6.95***	4.34***	5.47***

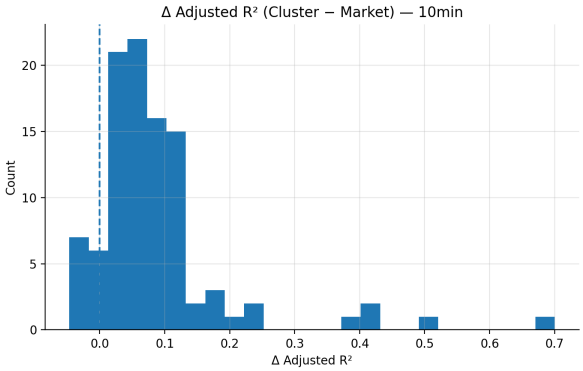
(3) Cluster vs. Single

Cluster \ Cluster	ols	lasso	hard	xgb	mlp	lstm	gru	salstm	sagru	msalstm	msagru
ols		3.85***	5.28***	-1.36	9.55***	3.66***	4.01***	3.46***	3.87***	6.92***	6.40***
lasso			4.58***	-2.22**	9.21***	3.05***	3.34***	2.78***	3.11***	6.41***	5.82***
hard				-8.66***	9.77***	0.10	0.01	-0.78	-0.78	4.40***	3.39***
xgb					15.96***	5.92***	6.38***	6.71***	6.23***	9.84***	9.08***
mlp						-4.11***	-4.80***	-5.93***	-6.12***	-0.59	-1.80*
lstm							-0.17	-0.94	-0.96	5.27***	3.79***
gru								-0.91	-0.98	5.78***	4.62***
salstm									0.08	6.10***	4.90***
sagru										6.64***	5.54***
msalstm											-2.17**
msagru											
Single vs.	5.89***	4.07***	2.64***	5.16***	-1.08	6.91***	7.55***	5.85***	4.90***	2.28**	5.10***

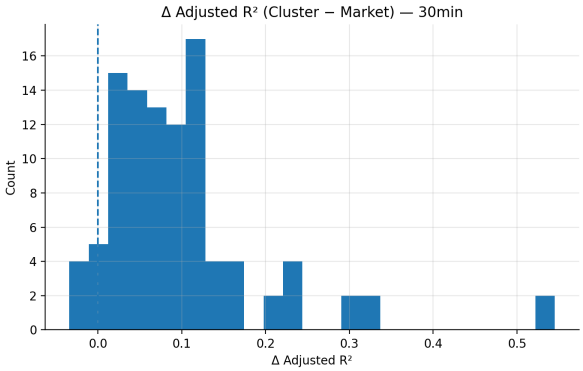
H Distribution of ΔAdjusted R^2

This appendix complements the regression evidence in Section 6.2 by documenting the full cross-sectional distribution of improvements in adjusted R^2 when the sector-level RV replaces the market aggregate in the naïve regressions. For each horizon, we plot the histogram of $\Delta_i = adR^2_{i,\text{clu}} - adR^2_{i,\text{mkt}}$. These figures serve as a distributional counterpart to Table 13, allowing us to verify that the superiority of the cluster specification is a broad-based cross-sectional phenomenon rather than the result of a few extreme observations.

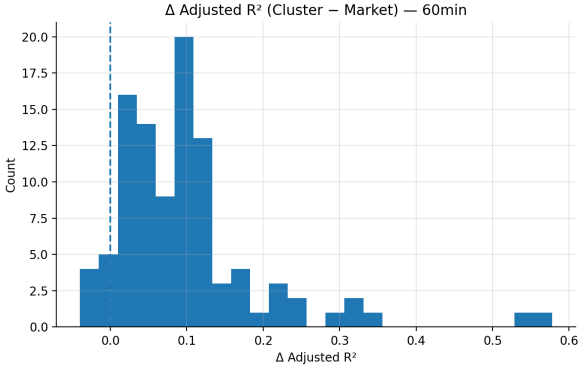
Across all horizons, the distributions are strongly right-skewed and centred well above zero. Most stocks exhibit improvements in the range of 0.03–0.15, with only a small number of slightly negative values. The pattern is most pronounced at the 30- and 60-minute horizons, where the gains are both larger and more concentrated, consistent with the tighter separation observed in the industry scatter plots. At the 240-minute horizon, the distribution displays a thicker right tail, indicating that some stocks benefit particularly strongly from sector-level information at lower frequencies. Taken together, the histograms reinforce the conclusion that sector-level RV captures systematic variation that is not fully reflected in the market aggregate, thereby supporting the empirical advantage of cluster-based training documented in the main text.



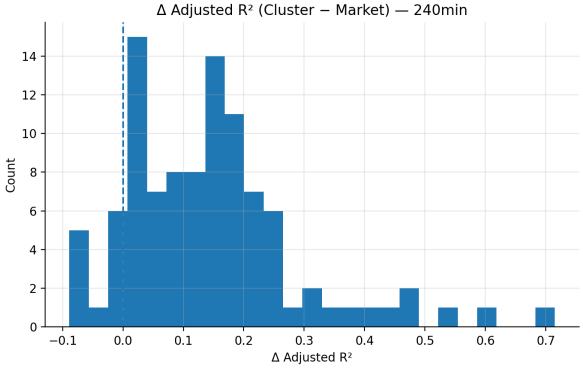
(a) 10-minute horizon



(b) 30-minute horizon



(c) 60-minute horizon



(d) 240-minute horizon

Figure H.1: Distribution of Δadjusted $R^2 = adR^2_{\text{clu}} - adR^2_{\text{mkt}}$ across stocks at four intraday horizons. The distributions are consistently right-skewed and centred above zero, indicating that the sector regression improves in-sample fit for the vast majority of stocks. The gains are most concentrated at the 30- and 60-minute horizons, while the 240-minute horizon exhibits a thicker right tail.