Shape Is Useful:

Geometric Analog Forecasting of Realized Volatility

Tongren Xiao Shuhang Chen

Abstract

This paper proposes a nonparametric forecasting model based on path geometric similarity. We embed volatility paths into a geometric space and retrieve historical paths according to their geometric similarity. A Transformer learns the combination weights of geometric similarity channels, which determine the retrieval weights of candidate paths. We then aggregate the future volatility changes associated with the retrieved paths to obtain the final forecast. This framework provides a comprehensive update to classical analog forecasting models. Unlike existing volatility forecasting models, our model relies neither on mathematical models nor on end-to-end deep learning techniques; instead, it forecasts entirely from the shapes of historical paths. Empirical results show that the model achieves strong forecasting performance at short and medium horizons. More importantly, using only realized volatility (RV), the model outperforms benchmark models that use additional information such as implied volatility (IV) and returns in large-sample short- and medium-horizon forecasts. We acknowledge that the model performs poorly at longer horizons, which remains an important direction for future research. A comprehensive series of empirical evaluations under different settings further confirms the robustness of our results.

Keywords: Realized volatility; Nonparametric forecasting; Nearest-neighbor methods; Time-series similarity; Geometric representation

1 Introduction

The modeling and forecasting of stock return volatility play important roles in risk management, asset allocation, and derivative pricing. With the widespread use of high-frequency financial data, realized volatility (RV) has become an important measure of asset price fluctuations. Relative to conventional volatility measures constructed from daily closing prices, RV exploits intraday price movements and therefore provides more detailed market information (Andersen et al., 2003). Research on RV forecasting has consequently continued to develop.

Existing RV forecasting models mainly follow two routes. One class of models is based on prespecified mathematical relationships and forms forecasts from multiscale statistics of historical volatility or other parametric states. The heterogeneous autoregressive model (HAR) is one of the most representative examples (Corsi, 2009). It aggregates historical RV at daily, weekly, and monthly scales to characterize the multiscale dependence of volatility. Subsequent studies have incorporated implied volatility (Christensen and Prabhala, 1998), cross-asset information, and network-neighborhood aggregation, leading to extensions such as HAR-IV and GHAR (Zhang et al., 2024). The other class adopts end-to-end deep learning techniques that directly learn the mapping from historical observables to future volatility. Among them, models such as GNNHAR further use graph neural networks to learn nonlinear dependencies across multi-asset series (Zhang et al., 2025). Overall, existing studies have continually increased both model complexity and the breadth of the information sets under consideration.

However, mainstream models have generally not treated the geometric form of historical RV paths as predictive information. By "geometric form,"we do not mean the absolute volatility level of a path, but rather its relative rises and falls, turning points, and patterns of change. Whether the geometric form of the current path provides incremental information about future RV remains insufficiently studied. Mathematical models typically compress historical paths into fixed-horizon lagged statistics or parametric states; graph models represent cross-asset information through neighborhood aggregation; and end-to-end models map input sequences directly to forecasts. Under these approaches, the evolution of a path is not identified and used as an independent source of information. In other words, existing models mainly focus on the numerical level of a path, fixed-horizon statistics, or prespecified dynamic relationships, rather than directly exploiting the predictive information embedded in the geometric form of historical RV paths.

Against this background, we ask two related questions. First, can the geometric information in historical RV paths produce accurate forecasts relative to conventional models that do not use IV? Second, can this information alone match or outperform models that additionally use IV? The second question is more demanding because it compares path geometry with a broader forecasting information set rather than with a different representation of historical volatility alone.

In fact, treating the form of a path or curve as information that can be represented, compared, and retrieved has been widely used in fields such as pattern recognition and nonlinear time-series analysis (Takens, 1981; Farmer and Sidorowich, 1987; Yakowitz, 1987). In financial volatility forecasting, Andrada-Félix et al. (2016) introduced nearest-neighbor forecasts into realized-variance forecasting and combined them with model-based forecasts. Because such methods rely on a single scale such as Euclidean distance and retrieve observations only within a single asset, they cannot precisely characterize the geometric properties of paths; their reported nearest-neighbor forecasts also did not consistently outperform traditional benchmarks such as HAR. Subsequently, as parametric models and end-to-end deep learning models developed rapidly, this simple approach of retrieval forecasting based on path similarity was largely abandoned and forgotten, which is regrettable.

This paper proposes a nonparametric forecasting model based entirely on path geometric similarity, termed the Shape Retrieval Model (SRM), for realized-volatility forecasting. The model first represents the one-dimensional RV sequence prior to the forecast origin as a historical path and maps it into a three-dimensional geometric curve through delay embedding, thereby retaining richer local geometric information. It then constructs five similarity channels—curve, velocity, acceleration, curvature, and global geometric measures—to characterize geometric differences between two paths from multiple perspectives. The Transformer learns the combination weights of these five channels from the current path state. The resulting composite similarity determines the retrieval weights of historical candidate paths, and the model generates the final forecast by aggregating the subsequent RV changes of these paths. Thus, the Transformer in SRM is only a similarity-weight learner: the entire forecast is generated through historical path retrieval rather than end-to-end volatility prediction.

Our main contribution is a novel nonparametric forecasting model based entirely on the geometric form of historical RV information. We use an S&P 100 stock sample to compare SRM with a range of benchmark models out of sample. The results show that SRM has strong forecasting ability at short and medium horizons. In particular, SRM using only RV outperforms benchmarks that use additional information, including IV, returns, and graph structures. Notably, SRM performs less well than the benchmarks at longer horizons, which is an issue that requires further improvement.

The remainder of this paper is organized as follows. Section 2 defines the forecasting setting and reviews the existing model classes relevant to our comparison. Section 3 introduces the Shape Retrieval Model. Section 4 describes the data, experimental design, and main empirical results. Section 5 provides further analysis, Section 6 presents robustness results, and Section 7 concludes.

2 Preliminaries and Existing Forecasting Models

This section specifies the common forecasting problem and summarizes the three existing modeling routes relevant to this paper. The purpose is to distinguish the information set and forecast-generation mechanism of SRM from those of classical parametric models, end-to-end deep learning models, and conventional analog forecasting.

2.1 Classical models[1]

This section reviews the classical realized-volatility forecasting models used as benchmarks in this paper, including HARM, HARM+IV, GHARM, and GHARM+IV. Let the price process of asset i be

$$d \log P_{i,s} = \mu_i ds + \sigma_{i,s} dW_{i,s},$$

where μ_i is the drift, $\sigma_{i,s}$ is the instantaneous volatility, and $W_{i,s}$ is a standard Brownian motion. The integrated variance on day t is

$$IV_{i,t} = \int_{t-1}^{t} \sigma_{i,s}^2 ds.$$

For an intraday sampling interval Δ, define

$$r_{i,t}(l) = \log P_{i,t}(l\Delta) - \log P_{i,t}((l-1)\Delta),$$

and

$$RV_{i,t} = \sum_{l=1}^{M} r_{i,t}^2(l),$$

where M denotes the number of intraday observations. Let $\mathbf{v}_t = (RV_{1,t}, \ldots, RV_{N,t})^\top$ denote the cross-sectional vector of realized volatilities. For $h \in \{1, 5, 21\}$, the common forecast target is

$$\mathbf{Y}_t^{(h)} = \frac{1}{h} \sum_{r=0}^{h-1} \mathbf{v}_{t+r}. \tag{1}$$

All forecasts are formed at the end of day $t-1$ and use only the information set $\mathcal{F}_{t-1}$.

Corsi (2009) proposed the heterogeneous autoregressive model (HAR), which incorporates lagged realized-volatility components over different time horizons. Let

$$\mathbf{v}_{i,t-1} = \left(RV_{i,t-1}, \quad \frac{1}{4} \sum_{\ell=2}^{5} RV_{i,t-\ell}, \quad \frac{1}{17} \sum_{\ell=6}^{22} RV_{i,t-\ell} \right)^\top$$

[1] Throughout this paper, bold capital letters denote matrices, bold lowercase letters denote vectors, and plain letters denote scalars.

denote the daily, weekly, and monthly HAR predictors. The pooled HARM specification is

$$Y_{i,t}^{(h)} = \alpha_i + \mathbf{v}_{i,t-1}^\top \boldsymbol{\beta}^{(h)} + u_{i,t}^{(h)}, \tag{2}$$

where α_i is an asset-specific intercept and $\boldsymbol{\beta}^{(h)}$ is shared across assets. HARM therefore preserves cross-sectional heterogeneity in levels while imposing a common multiscale persistence relation.

Let $q_{i,t-1}$ denote the implied-volatility regressor available at the forecast origin after the unit conversion described in Section 4. The IV-augmented HARM is written as

$$Y_{i,t}^{(h)} = \alpha_i + \mathbf{v}_{i,t-1}^\top \boldsymbol{\beta}^{(h)} + \eta^{(h)} q_{i,t-1} + u_{i,t}^{(h)}. \tag{3}$$

Equation (3) asks whether the forward-looking information contained in IV improves upon the backward-looking persistence summarized by HAR components.

Graph-based extensions augment these own-asset predictors with information from neighboring assets. Let $\mathbf{r}_t = (r_{1,t}, \ldots, r_{N,t})^\top$ be the vector of daily returns. Using only returns available before the test period, the graphical lasso estimates a precision matrix $\widehat{\boldsymbol{\Theta}}$, from which the binary adjacency matrix is constructed as

$$A_{ij} = \mathbb{I}\Big(\widehat{\Theta}_{ij} \neq 0\Big)\, \mathbb{I}(i \neq j).$$

Let $D_{ii} = \sum_{j=1}^N A_{ij}$ and define the symmetrically normalized graph matrix by

$$\mathbf{W} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}.$$

For an isolated node, the corresponding diagonal element of $\mathbf{D}^{-1/2}$ is defined as zero. For $\mathbf{V}_{t-1} = (\mathbf{v}_{1,t-1}, \ldots, \mathbf{v}_{N,t-1})^\top$, the vector $(\mathbf{W}\mathbf{V}_{t-1})_{i,\cdot}$ aggregates the HAR predictors of the neighbors of asset i. The GHARM specification is therefore

$$Y_{i,t}^{(h)} = \alpha_i + \mathbf{v}_{i,t-1}^\top \boldsymbol{\beta}^{(h)} + (\mathbf{W}\mathbf{V}_{t-1})_{i,\cdot}\boldsymbol{\gamma}^{(h)} + u_{i,t}^{(h)}, \tag{4}$$

and GHARM+IV adds the term $\eta^{(h)} q_{i,t-1}$ to Eq. (4). These models are based on prespecified parametric relations. Existing work notes that linear realized-volatility models may suffer from misspecification in the presence of nonlinear volatility dynamics (Buccheri and Corsi, 2019), while linear graph aggregation may be inadequate when cross-asset spillovers are nonlinear (Zhang et al., 2025).

2.2 End-to-end deep learning models

Unlike the classical models in Section 2.1, end-to-end deep learning models generally do not prespecify the functional relation between historical variables and future realized volatility. Instead, they jointly learn the model parameters by minimizing a forecasting loss. If $\mathbf{X}_{t-1}$ denotes the variables available at the forecast origin, such a model can be written generally as

$$\widehat{\mathbf{Y}}_t^{(h)} = f_{\boldsymbol{\theta}}(\mathbf{X}_{t-1}),$$

where $\widehat{\mathbf{Y}}_t^{(h)}$ is the predicted vector of future realized volatilities, $f_{\boldsymbol{\theta}}(\cdot)$ is a deep-learning mapping determined by parameters $\boldsymbol{\theta}$, and $\mathbf{X}_{t-1}$ contains the variables available by the forecast origin. The defining feature is that the network output itself is the forecast of future realized volatility.

2.2.1 GNNHAR

Modern volatility forecasting often considers interactions across assets, and GNNHAR is a representative model in this class. Compared with the GHARM model in Section 2.1, which characterizes cross-asset relations through prespecified linear neighborhood aggregation, GNNHAR replaces this linear aggregation with a graph neural network and thereby learns nonlinear volatility spillovers.

The core technique of GNNHAR is the graph neural network. Its basic idea is to aggregate information from each node and its neighboring nodes to learn a new vector representation for every node. Let $\mathbf{H}^{(\ell)}$ denote the node-representation matrix at layer ℓ. A generic graph-convolution layer can be written as

$$\mathbf{H}^{(\ell+1)} = \sigma\left(\widetilde{\mathbf{W}}\mathbf{H}^{(\ell)}\boldsymbol{\Theta}^{(\ell)}\right),$$

where $\mathbf{H}^{(\ell)}$ and $\mathbf{H}^{(\ell+1)}$ are the node-representation matrices at layers ℓ and $\ell+1$, $\widetilde{\mathbf{W}}$ is the graph-propagation matrix, $\boldsymbol{\Theta}^{(\ell)}$ is the trainable parameter matrix at layer ℓ, and $\sigma(\cdot)$ is a nonlinear activation function. Stacking graph-convolution layers allows the model to use information from higher-order neighbors.

Zhang et al. (2025) use graph neural networks with HAR features as node inputs to construct GNNHAR. Let $\mathbf{H}^{(0)} = V_{t-1}$. The model can be represented generally as

$$\mathbf{H}^{(\ell+1)} = \mathrm{GNN}\left(\mathbf{H}^{(\ell)}, \mathbf{A}\right),$$

where $\mathbf{A}$ denotes the asset-relation graph and $\mathrm{GNN}(\cdot)$ denotes a graph-neural-network layer that performs neighborhood aggregation and nonlinear transformation. After L graph-propagation layers, the forecasting relation is

$$E(\mathbf{Y}_t^{(h)} \mid \mathcal{F}_{t-1}) = \boldsymbol{\alpha} + V_{t-1}\boldsymbol{\beta} + \mathbf{H}^{(L)}\boldsymbol{\gamma},$$

where $\mathbf{Y}_t^{(h)}$ is the vector of future realized-volatility targets, $\mathcal{F}_{t-1}$ is the information set available by day $t-1$, $\boldsymbol{\alpha}$ is the vector of asset-specific intercepts, V_{t-1} is the HAR input-feature matrix, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are parameters to be estimated, and $\mathbf{H}^{(L)}$ is the node-representation matrix output by the Lth layer. The graph-propagation and forecasting parameters are trained jointly to generate the future realized-volatility forecasts directly.

2.2.2 Transformer-based forecasting models

Unlike GNNHAR, which relies on a prespecified asset-relation graph, Transformer-based forecasting models primarily use self-attention to learn dependencies across positions in an input sequence. For query, key, and value matrices $(\mathbf{Q}, \mathbf{K}, \mathbf{V})$, the basic self-attention operation is

$$\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{d_k}}\right)\mathbf{V}.$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ denote the query, key, and value matrices, respectively, $\mathbf{K}^{\top}$ is the transpose of the key matrix, and d_k is the dimension of the key vectors. The operation assigns attention weights to different elements of the value matrix according to the similarity between queries and keys.

The Transformer architecture was first proposed by Vaswani et al. (2017). Subsequently, models such as the Temporal Fusion Transformer applied attention mechanisms to multi-horizon time-series forecasting (Lim et al., 2021). These models generally encode the historical sequence with a Transformer and then use a forecasting layer to output the future target directly:

$$\widehat{\mathbf{Y}}_t^{(h)} = g_{\boldsymbol{\theta}}[\mathrm{Transformer}(\mathbf{X}_{t-1})]\,,$$

where $\mathrm{Transformer}(\mathbf{X}_{t-1})$ is the hidden representation obtained by encoding the input sequence with attention, $g_{\boldsymbol{\theta}}(\cdot)$ is a forecasting layer determined by parameters $\boldsymbol{\theta}$, and $\widehat{\mathbf{Y}}_t^{(h)}$ is the final forecast of future realized volatility.

Although GNNHAR and Transformer-based models use different network architectures, both directly learn a mapping from input information to future realized volatility, and the network output constitutes the final forecast.

2.3 Traditional analog forecasting

Analog forecasting is a nonparametric forecasting method that differs fundamentally from the models reviewed above. Rather than estimating a mapping from historical variables to future volatility, it searches the historical sample for paths closest to the current state and constructs a forecast directly from the subsequent outcomes of those paths. This is an extremely simple idea. Related analog-based approaches have long been used in weather forecasting and have developed into analog ensemble methods that construct predictive distributions from historically similar weather states (Lorenz, 1969) and (Delle Monache et al., 2013); however, their application to realized-volatility forecasting has remained relatively limited.

To describe the basic form of this method, let the historical-path length for asset i at forecast origin t be L. The current path is written as

$$\mathbf{p}_{i,t} = (RV_{i,t-L}, \dots, RV_{i,t-1})^\top$$

where $RV_{i,t}$ is the realized volatility of asset i on date t, and $\mathbf{p}_{i,t}$ contains only information available before the forecast origin. For a historical candidate endpoint s, the corresponding path is

$$\mathbf{p}_{i,s} = (RV_{i,s-L}, \dots, RV_{i,s-1})^\top.$$

Given a prespecified distance function $d(\cdot, \cdot)$, the model selects the K historical paths that are closest to the current path:

$$\mathcal{N}_{i,t}(K) = \arg\min_{\substack{\mathcal{S} \\ |\mathcal{S}|=K}} \sum_{s \in \mathcal{S}} d(\mathbf{p}_{i,t}, \mathbf{p}_{i,s}), \tag{5}$$

where $\mathcal{N}_{i,t}(K)$ denotes the set of K nearest neighbors of the current path, and $d(\cdot, \cdot)$ is usually Euclidean distance or another prespecified sequence distance. The forecast is then constructed by aggregating the future realized volatilities of these neighboring paths:

$$\widehat{Y}_{i,t}^{(h),\mathrm{NN}} = \mathcal{M}\left\{ Y_{i,s}^{(h)} : s \in \mathcal{N}_{i,t}(K) \right\}, \tag{6}$$

where

$$Y_{i,s}^{(h)} = \frac{1}{h} \sum_{r=1}^{h} RV_{i,s+r}$$

is the average realized volatility over the h trading days following historical path s, and $\mathcal{M}(\cdot)$ denotes an aggregation function, such as the median, the mean, or a kernel-weighted mean.

Andrada-Félix et al. (2016) provided an early application of this approach to realized-variance forecasting for the S&P 100 index. They used fixed-length histories of realized variance, selected nearest neighbors by Euclidean distance, and formed forecasts from the median realized variance subsequently observed after those neighbors. Both the history length and the number of neighbors were selected by in-sample mean squared error. Their results show that nearest-neighbor forecasts did not consistently outperform model-based forecasts, including HAR, under conventional statistical accuracy measures. Their relative advantage was instead reflected in risk-adjusted trading performance during turbulent markets, while directional combinations of nearest-neighbor and model-based forecasts performed best.

3 Methodology

To address the limitations of the existing model classes, we propose a forecasting model based on the traditional analog-forecasting approach. Section 3.1 introduces the Shape Retrieval Model; Section 3.2 describes geometric similarity; and Section 3.3 presents adaptive weighting.

3.1 Shape Retrieval Model

Traditional analog models generally measure path similarity using distances defined on one-dimensional numerical sequences and conduct retrieval within a single asset. Such models have difficulty distinguishing local path movements from global structure and cannot exploit similar volatility states across assets. To overcome these limitations, we propose the Shape Retrieval Model (SRM). The model represents historical RV paths as three-dimensional geometric curves, defines path similarity through multiple geometric channels, and retrieves similar states from the historical paths of all sample assets.

Let $RV_{i,t} > 0$ denote the realized volatility of asset i on trading day t, where $i = 1, \dots, N$. The model forms its forecast after the end of trading day $t-1$. For forecast horizon $h \geq 1$, the forecast target is

$$Y_{i,t}^{(h)} = \frac{1}{h} \sum_{r=0}^{h-1} RV_{i,t+r}.$$

Let $x_{i,u} = \log[\max(RV_{i,u}, \varepsilon)]$, where $\varepsilon > 0$ is a fixed numerical-stability constant. Given path length L, the query path of asset i at forecast origin t is defined as

$$\mathbf{p}_{i,t} = (x_{i,t-L}, \dots, x_{i,t-1})^\top,$$

and the candidate path of asset j ending at historical endpoint s is defined as

$$\mathbf{p}_{j,s} = (x_{j,s-L+1}, \dots, x_{j,s})^\top.$$

The subsequent relative volatility change associated with the candidate path is defined as

$$R_{j,s}^{(h)} = \log\left[\frac{h^{-1} \sum_{r=1}^{h} RV_{j,s+r}}{RV_{j,s} + \varepsilon}\right].$$

In out-of-sample forecasting, let b_w denote the endpoint of the historical information available in rolling window w. The candidate set is defined as

$$\mathcal{M}_w^{(h)} = \{(j,s) : j \in \{1, \dots, N\}, \quad L \leq s, \quad s+h \leq b_w\}.$$

Here, j denotes the candidate asset and s denotes the historical endpoint of the candidate path. Let $D^{(h)}_{i,t,j,s}$ denote the composite geometric measure between the query and candidate paths, as defined in Section 3.2. SRM selects from $\mathcal{M}^{(h)}_w$ the K_{NN} paths with the smallest measure, and hence the greatest geometric similarity, to form the nearest-neighbor set $\mathcal{N}^{(h)}_{i,t}$. Given temperature parameter $T_{w,h} > 0$, the retrieval weight of a candidate path is

$$\begin{gathered}\omega^{(h)}_{i,t,j,s} = \frac{\exp[-D^{(h)}_{i,t,j,s}/T_{w,h}]}{\sum_{(j',s')\in\mathcal{N}^{(h)}_{i,t}} \exp[-D^{(h)}_{i,t,j',s'}/T_{w,h}]}, \\ (j,s) \in \mathcal{N}^{(h)}_{i,t}.\end{gathered} \tag{7}$$

The temperature parameter controls the concentration of retrieval weights across neighboring paths. The SRM forecast is obtained by aggregating the subsequent relative changes of these historical paths:

$$\widehat{Y}^{(h)}_{i,t} = RV_{i,t-1} \exp\left[\sum_{(j,s)\in\mathcal{N}^{(h)}_{i,t}} \omega^{(h)}_{i,t,j,s} R^{(h)}_{j,s} \right]. \tag{8}$$

The exponential term in Eq. (8) is the weighted geometric mean of the candidate paths' relative changes. The future volatility of asset i evolves from its most recent volatility level before the forecast origin. Accordingly, Eq. (8) uses the target asset's own $RV_{i,t-1}$ as the level anchor and applies the subsequent relative changes of similar paths to this anchor. Cross-asset candidate paths provide only relative-change patterns; their own volatility levels do not enter the target asset's forecast.

3.2 Geometric Similarity

SRM characterizes path similarity through a three-dimensional path representation and five geometric channels. To remove differences in path level and scale, we first standardize each path. For any path $\mathbf{p} = (p_1, \ldots, p_L)^\top$, define

$$z_r = \frac{p_r - \bar{p}}{\sigma_p + \varepsilon}, \qquad \bar{p} = \frac{1}{L}\sum_{r=1}^{L} p_r, \qquad \sigma_p^2 = \frac{1}{L}\sum_{r=1}^{L}(p_r - \bar{p})^2.$$

We use each position and its first two lagged values as three-dimensional coordinates so that local movement direction and curvature can be measured in the same space:[2]

$$\mathbf{c}_r = \left(z_r, z_{\max(1,r-1)}, z_{\max(1,r-2)}\right)^\top, \qquad r = 1, \ldots, L.$$

[2] Because historical RV paths are observed at discrete trading dates, we use finite differences and discrete geometric quantities rather than the continuous forms commonly used in differential geometry.

On this basis, the path's velocity, acceleration, and curvature are defined as

$$\begin{aligned}\mathbf{v}_r &= \frac{\mathbf{c}_{r+1} - \mathbf{c}_{r-1}}{2},\\ \mathbf{a}_r &= \mathbf{c}_{r+1} - 2\mathbf{c}_r + \mathbf{c}_{r-1},\\ \kappa_r &= \frac{\|\mathbf{v}_r \times \mathbf{a}_r\|_2}{(\|\mathbf{v}_r\|_2^2 + \delta)^{3/2}}.\end{aligned} \tag{9}$$

Here, $r = 2, \ldots, L-1$ and $\delta > 0$; one-sided differences are used for endpoint velocities, while endpoint accelerations take the values of their adjacent interior points. Curve position represents path form, velocity and acceleration describe local movement and its adjustment, and curvature captures local turning. To complement these local quantities, define

$$\mathbf{G}(\mathbf{p}) = \left(\ell(\mathbf{p}), \Delta(\mathbf{p}), \frac{\ell(\mathbf{p})}{\Delta(\mathbf{p}) + \delta}\right)^{\top},$$

where $\ell(\mathbf{p}) = \sum_{r=1}^{L-1} \|\mathbf{c}_{r+1} - \mathbf{c}_r\|_2$ is curve length and $\Delta(\mathbf{p}) = \|\mathbf{c}_L - \mathbf{c}_1\|_2$ is endpoint displacement.

Write the vector representations of the five channels as

$$\begin{aligned}&\mathbf{u}_C(\mathbf{p}) = \mathrm{vec}(\mathbf{C}), \quad \mathbf{u}_V(\mathbf{p}) = \mathrm{vec}(\mathbf{V}),\\ &\mathbf{u}_A(\mathbf{p}) = \mathrm{vec}(\mathbf{A}), \quad \mathbf{u}_K(\mathbf{p}) = (\kappa_1, \ldots, \kappa_L)^{\top},\\ &\mathbf{u}_G(\mathbf{p}) = \mathbf{G}(\mathbf{p}),\end{aligned}$$

where $\mathrm{vec}(\cdot)$ arranges matrix entries into a column vector in row-major order, $\mathbf{C} = (\mathbf{c}_1, \ldots, \mathbf{c}_L)^{\top}$, $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_L)^{\top}$, and $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_L)^{\top}$. The corresponding dimensions are $n_C = n_V = n_A = 3L$, $n_K = L$, and $n_G = 3$. The distance between the query and candidate paths in channel k is written uniformly as

$$d_{i,t,j,s,k} = \left[\frac{1}{n_k} \|\mathbf{u}_k(\mathbf{p}_{i,t}) - \mathbf{u}_k(\mathbf{p}_{j,s})\|_2^2\right]^{1/2}. \tag{10}$$

For each query asset and rolling window, distances from different channels are standardized using the median and interquartile range obtained by comparing that asset's last eligible training query with all eligible cross-asset candidate paths in the training period:

$$\widetilde{d}_{i,t,j,s,k} = \frac{d_{i,t,j,s,k} - m_{i,w,h,k}}{q_{i,w,h,k} + \delta},$$

where $m_{i,w,h,k}$ and $q_{i,w,h,k}$ are the corresponding median and interquartile range. This standardization allows the five distances to be combined on a common scale.

3.3 Adaptive Weighting

The importance of different geometric channels may vary with the state of the query path. To convert geometric representations of different dimensions into a common Transformer input, for $k \in \{C, V, A, K\}$ define the corresponding channel state as

$$\begin{aligned}\mathbf{g}_{i,t,k} &= \left[\mathrm{Mean}(\mathbf{u}_k(\mathbf{p}_{i,t})), \mathrm{RMS}(\mathbf{u}_k(\mathbf{p}_{i,t})), (\mathbf{u}_k(\mathbf{p}_{i,t}))_{n_k}\right]^\top,\\ \mathbf{g}_{i,t,G} &= \mathbf{G}(\mathbf{p}_{i,t}).\end{aligned}$$

Here, $\mathrm{Mean}(\cdot)$ and $\mathrm{RMS}(\cdot)$ denote the mean and root-mean-square of a channel representation, respectively, and $(\mathbf{u}_k)_{n_k}$ denotes its last component. Let

$$\mathbf{Q}_{i,t} = \left(\mathbf{g}_{i,t,C}, \mathbf{g}_{i,t,V}, \mathbf{g}_{i,t,A}, \mathbf{g}_{i,t,K}, \mathbf{g}_{i,t,G}\right)^\top \in \mathbb{R}^{5\times 3}.$$

After adding channel-identity embeddings that distinguish the five channels and a horizon embedding, the Transformer outputs

$$\begin{aligned}\boldsymbol{\beta}_{i,t,h} &= \mathrm{softmax}\left[f_{\boldsymbol{\theta}}(\mathbf{Q}_{i,t}, e_h)\right],\\ \sum_{k\in\{C,V,A,K,G\}} \beta_{i,t,h,k} &= 1.\end{aligned} \tag{11}$$

Here, $\boldsymbol{\theta}$ denotes the trainable Transformer parameters, e_h denotes the horizon information, and $\beta_{i,t,h,k} \geq 0$ is the weight assigned to channel k. In the implementation, the horizon embedding distinguishes the one-day horizon from the multi-day horizons ($h = 5, 21$). The composite geometric measure is therefore defined as

$$D^{(h)}_{i,t,j,s} = \sum_{k\in\{C,V,A,K,G\}} \beta_{i,t,h,k}\widetilde{d}_{i,t,j,s,k}. \tag{12}$$

For a given query path and forecast horizon, the channel weights remain unchanged across all candidate paths, and the retrieval weight of each candidate path is determined by its composite geometric measure. In SRM, the Transformer is used only to organize different geometric information and does not directly generate future RV forecasts; the final forecast continues to be obtained by aggregating the subsequent changes of historically similar paths.

3.4 Estimation Criterion

Mean squared error is the most commonly used in-sample estimation criterion in volatility forecasting. However, realized volatility is generally treated as an imperfect proxy for latent volatility, and forecast errors are heterogeneous across volatility levels. In this setting, QLIKE emphasizes the proportional discrepancy between the forecast and the realization and has desirable robustness properties in the presence of measurement error in volatility proxies (Patton, 2011). We therefore pay particular attention to the forecasting performance of models estimated using QLIKE.

For model F, the MSE estimation criterion is defined as

$$\mathrm{MSE}_F = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\mathcal{T}_{\mathrm{train}}|}\sum_{t\in\mathcal{T}_{\mathrm{train}}}\left(Y_{i,t}^{(h)} - \widehat{Y}_{i,t}^{(h,F)}\right)^2, \tag{13}$$

where N is the number of assets, $\mathcal{T}_{\mathrm{train}}$ is the training sample, $Y_{i,t}^{(h)}$ is the realized target, and $\widehat{Y}_{i,t}^{(h,F)}$ is the forecast of model F. The QLIKE estimation criterion is defined as

$$\mathrm{QLIKE}_F = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\mathcal{T}_{\mathrm{train}}|}\sum_{t\in\mathcal{T}_{\mathrm{train}}}\left[\frac{Y_{i,t}^{(h)}}{\widehat{Y}_{i,t}^{(h,F)}} - \log\left(\frac{Y_{i,t}^{(h)}}{\widehat{Y}_{i,t}^{(h,F)}}\right) - 1\right]. \tag{14}$$

Both criteria are minimized. We do not impose the prior assumption that either criterion is uniformly superior. Instead, MSE and QLIKE are treated as two representative estimation schemes: SRM and the four benchmark models are estimated separately under each criterion and compared under the same out-of-sample design.

4 Empirical Analysis

4.1 Data and Experimental Design

We evaluate the out-of-sample performance of the models on the S&P 100 equity panel. To ensure comparability across models, the sample retains only stocks with complete realized-volatility (RV), implied-volatility (IV), and daily-return observations over the common trading days. Table 1 summarizes the sample construction. The ticker symbols are listed in Appendix A, where we also report descriptive statistics for the volatility estimates.

Table 1: Sample construction

Sample	Sample period	Trading days	Effective stocks
S&P 100	9 Jun. 2021–9 Jun. 2026	1,256	91

Note: A stock is retained only when its RV, IV, and daily-return observations are complete over all common trading days.

We compare the out-of-sample RV forecasts of SRM and the four benchmark models introduced in Sections 2 and 3. All models use the same training, validation, and test blocks within each rolling window, and forecast the same target on the same test dates. Each window contains 600 trading days for training, followed by 200 trading days for validation. An h-day purge separates the training, validation, and test blocks. The following

21 trading days form the test block, and the rolling window advances by 21 trading days. Once the validation block has ended, each model uses the union of the training and validation origins for its final pre-test estimation and is then held fixed throughout the test block. For SRM, validation outcomes are used before this refit to select the stopping point and retrieval temperature.

We consider forecast horizons of $h = 1$, $h = 5$, and $h = 21$. For asset i, a forecast formed at the end of day $t - 1$ targets

$$Y_{i,t}^{(h)} = \frac{1}{h} \sum_{r=0}^{h-1} RV_{i,t+r}.$$

Thus, $h = 1$, $h = 5$, and $h = 21$ correspond to the next-day RV and the average RV over the subsequent five and 21 trading days, respectively. This target definition is common to SRM and all benchmark models.

To examine the role of the estimation criterion, we estimate all five models separately under MSE and QLIKE. Forecast performance is evaluated by MSE, QLIKE, and MAE. The main results report forecast losses under both MSE and QLIKE estimation, thereby separating the role of the model architecture from that of the estimation criterion.

4.2 Main Results

We organize the main results around the two questions stated in the Introduction. Section 4.2.1 compares SRM with HARM and GHARM, neither of which uses IV. Section 4.2.2 compares SRM with HARM+IV and GHARM+IV, which additionally use forward-looking IV.

4.2.1 Comparison with Models without IV

Table 2 reports the comparison with HARM and GHARM. Panels A and B contain the MSE- and QLIKE-estimated forecasts, respectively.

Table 2: SRM and models without IV on S&P 100

Model	$h=1$			$h=5$			$h=21$		
	MSE	QLIKE	MAE	MSE	QLIKE	MAE	MSE	QLIKE	MAE
Panel A: MSE estimation									
HARM	7.952521	0.003870	**1.260781**	18.685401	0.008806	2.494335	**59.073077**	**0.026160**	**5.171324**
GHARM	7.954865	0.003861	1.280607	18.641559	0.008739	2.527492	61.340851	0.026486	5.272023
SRM	**7.347110**	**0.003824**	1.281628	**15.263724**	**0.008018**	**2.253593**	62.534382	0.029097	5.308454
Panel B: QLIKE estimation									
HARM	7.982958	0.003851	**1.255222**	18.891879	0.008676	2.503819	**59.541349**	**0.025913**	**5.184390**
GHARM	7.970248	0.003834	1.270597	18.748372	0.008558	2.522628	61.320873	0.026049	5.268992
SRM	**7.358237**	**0.003804**	1.280529	**15.471597**	**0.008015**	**2.262870**	64.391834	0.028951	5.422775

Note: Panel A and Panel B report forecasts estimated by MSE and QLIKE, respectively. All three models use historical RV; GHARM also uses returns to estimate the asset graph. The lowest loss within each panel and horizon is in bold.

At the one-day horizon, SRM has the lowest MSE and QLIKE under both estimation criteria. Its MSE is 7.61% lower than that of HARM under MSE estimation and 7.83% lower under QLIKE estimation. HARM retains the lowest MAE, although the differences among the three models are small.

The five-day results are stronger. SRM has the lowest MSE, QLIKE, and MAE under both estimation criteria. Relative to HARM, QLIKE-estimated SRM lowers MSE, QLIKE, and MAE by 18.10%, 7.62%, and 9.62%, respectively. The DM–HLN tests in Table 4 reject equal QLIKE predictive accuracy against both HARM and GHARM. Thus, historical path geometry provides predictive information beyond the conventional specifications without IV at the short and medium horizons.

This result reverses at $h = 21$. All three SRM losses exceed those of HARM and GHARM under both estimation criteria, and the QLIKE differences are statistically significant. The geometric information in the 22-day path therefore improves short- and medium-horizon forecasts but is insufficient for the monthly horizon.

4.2.2 Comparison with IV-Augmented Models

We next compare SRM, which uses RV alone, with HARM+IV and GHARM+IV. The latter models add IV, while GHARM+IV also uses daily returns to estimate the asset graph. Table 3 reports the results.

Table 3: SRM and IV-augmented models on S&P 100

	$h = 1$			$h = 5$			$h = 21$		
Model	MSE	QLIKE	MAE	MSE	QLIKE	MAE	MSE	QLIKE	MAE
Panel A: MSE estimation									
HARM+IV	7.699478	0.003782	1.313510	16.587135	0.008073	2.476665	**44.618000**	**0.019999**	**4.467461**
GHARM+IV	7.704829	**0.003779**	1.317262	16.611630	0.008050	2.482645	45.334074	0.020027	4.516073
SRM	**7.347110**	0.003824	**1.281628**	**15.263724**	**0.008018**	**2.253593**	62.534382	0.029097	5.308454
Panel B: QLIKE estimation									
HARM+IV	7.746688	0.003754	1.296741	16.909493	0.007892	2.480192	**45.092197**	**0.019670**	**4.479493**
GHARM+IV	7.743830	**0.003747**	1.301770	16.858009	**0.007856**	2.482108	45.662524	0.019753	4.518830
SRM	**7.358237**	0.003804	**1.280529**	**15.471597**	0.008015	**2.262870**	64.391834	0.028951	5.422775

Note: Panel A and Panel B report forecasts estimated by MSE and QLIKE, respectively. SRM uses RV alone; HARM+IV uses RV and IV; GHARM+IV additionally uses returns to estimate the asset graph. The lowest loss within each panel and horizon is in bold.

At $h = 1$, SRM has the lowest MSE and MAE under both estimation criteria, whereas the IV-augmented models have lower QLIKE. Under QLIKE estimation, SRM lowers MSE by approximately 5% relative to the best IV-augmented benchmark, but its QLIKE is 1.54% higher. The DM–HLN tests do not reject equal QLIKE predictive accuracy between SRM and either IV-augmented model.

At $h = 5$, MSE-estimated SRM has the lowest loss under all three evaluation measures. Relative to the strongest IV-augmented benchmark for each measure, its MSE, QLIKE, and MAE are 7.98%, 0.39%, and 9.01% lower. Under QLIKE estimation, SRM continues to lower MSE and MAE by 8.22% and 8.76%, while its QLIKE is 2.02% higher. These QLIKE differences are not statistically significant. Hence, RV path geometry can match the QLIKE accuracy of the richer information set at the medium horizon while producing materially lower level-based errors.

The IV-augmented models dominate at $h = 21$. Under QLIKE estimation, SRM's MSE, QLIKE, and MAE exceed those of GHARM+IV by 41.00%, 46.57%, and 20.00%, respectively. Path geometry alone is therefore competitive with additional information at short and medium horizons, but not at the monthly horizon.

Table 4: DM–HLN tests for QLIKE-estimated forecasts

	$h = 1$		$h = 5$		$h = 21$	
Benchmark	DM–HLN	p-value	DM–HLN	p-value	DM–HLN	p-value
HARM	0.661	0.509	3.290	0.001	−2.340	0.020
HARM+IV	−0.720	0.472	−0.537	0.591	−7.154	< 0.001
GHARM	0.399	0.690	2.384	0.018	−3.300	0.001
GHARM+IV	−0.795	0.427	−0.650	0.516	−7.892	< 0.001

Note: Positive statistics favor SRM. Two-sided p-values use daily cross-sectional average loss differentials.

4.3 Market Regimes

We next examine whether the relative performance of SRM varies across market states. Within each rolling window, test dates above the pre-test 90th percentile of the cross-sectional average target are classified as high-volatility periods, and the remaining dates as normal periods. Table 5 reports the percentage loss reduction of QLIKE-estimated SRM relative to HARM and GHARM+IV; positive values indicate lower SRM losses.

For the five-day horizon, all three SRM losses are lower than those of HARM in both market states. During high-volatility periods, its MSE, QLIKE, and MAE are 22.94%, 13.39%, and 17.42% lower, respectively. Relative to GHARM+IV, however, its MSE and MAE are 3.00% and 6.58% lower, whereas its QLIKE is 10.47% higher. The one-day high-volatility results are similar: MSE and MAE are 2.87% and 1.66% lower than those of GHARM+IV, while QLIKE is 5.30% higher. Hence, during high-volatility periods, the reductions relative to GHARM+IV occur in MSE and MAE, not in QLIKE.

Table 5: Relative SRM losses across market regimes (%)

		Relative to HARM			Relative to GHARM+IV		
h	Regime	MSE	QLIKE	MAE	MSE	QLIKE	MAE
1	High	9.65	2.96	3.01	2.87	−5.30	1.66
	Normal	7.18	0.87	−3.46	5.69	−0.82	1.63
5	High	22.94	13.39	17.42	3.00	−10.47	6.58
	Normal	15.66	5.92	6.64	10.45	0.05	9.57
21	High	8.30	8.51	5.52	−18.84	−22.78	−11.91
	Normal	−18.60	−20.74	−8.49	−55.25	−56.82	−22.99

Note: High-volatility dates exceed the pre-test 90th percentile of the cross-sectional average target; all other dates are classified as normal. Positive entries indicate lower SRM loss.

For the 21-day horizon, all three SRM losses exceed those of HARM during normal periods and those of GHARM+IV in both market states.

5 Further Analysis

This section examines three features of SRM. We first compare the two estimation criteria, then assess the contribution of cross-asset retrieval, and finally examine the geometric-channel weights.

5.1 Estimation Criterion

Table 6 reports the losses of SRM estimated by MSE and QLIKE. QLIKE estimation produces lower QLIKE at all three horizons. MSE estimation yields lower MSE at each horizon and lower MAE at the five- and 21-day horizons.

Table 6: SRM forecast losses under alternative estimation criteria

h	Estimation criterion	MSE	QLIKE	MAE
1	MSE	7.347110	0.003824	1.281628
	QLIKE	7.358237	0.003804	1.280529
5	MSE	15.263724	0.008018	2.253593
	QLIKE	15.471597	0.008015	2.262870
21	MSE	62.534382	0.029097	5.308454
	QLIKE	64.391834	0.028951	5.422775

Note: Each row reports the out-of-sample losses of SRM estimated by the indicated criterion on the S&P 100 sample.

The differences between the two estimates are small relative to the differences across forecast horizons. Under both criteria, SRM performs well at the one- and five-day horizons but poorly at the 21-day horizon. Hence, the main results are not specific to a single estimation criterion.

5.2 Same-Asset and Cross-Asset Retrieval

SRM searches the historical paths of all sample assets. To examine whether paths from other assets add useful information, we restrict the candidate set to the target asset and re-estimate the model. All other settings remain unchanged.

Table 7 shows that cross-asset retrieval lowers MSE, QLIKE, and MAE at every horizon. At the five-day horizon, the reductions are 5.23%, 3.65%, and 3.08%, respectively. The corresponding reductions at the one- and 21-day horizons are smaller but remain positive for all three losses. Thus, the historical paths of other assets contain analog states that are not available in the target asset's own history. The improvement at the 21-day horizon does not, however, eliminate SRM's long-horizon disadvantage.

5.3 Geometric Channel Weights

Table 8 reports the average geometric-channel weights of the QLIKE-estimated SRM. Velocity receives the largest weight at the one-day horizon. Its weight declines as the

Table 7: Same-asset and cross-asset retrieval

h	Retrieval scope	MSE	QLIKE	MAE	ΔMSE (%)	ΔQLIKE (%)	ΔMAE (%)
1	Same asset	7.674975	0.003881	1.298407	—	—	—
	Cross asset	7.358237	0.003804	1.280529	−4.13	−1.97	−1.38
5	Same asset	16.325556	0.008319	2.334889	—	—	—
	Cross asset	15.471597	0.008015	2.262870	−5.23	−3.65	−3.08
21	Same asset	68.193285	0.029838	5.581062	—	—	—
	Cross asset	64.391834	0.028951	5.422775	−5.57	−2.97	−2.84

Note: Results are for the S&P 100 sample and QLIKE estimation. The last three columns report percentage changes relative to same-asset retrieval; negative values indicate lower loss. Both specifications use the same test observations and differ only in the candidate-asset set.

horizon increases, whereas the weights on curvature and global geometry rise. At the 21-day horizon, curvature and global geometry jointly account for 79.4% of the total weight.

Table 8: Average geometric-channel weights

h	Curve	Velocity	Acceleration	Curvature	Global	Effective channels
1	0.190	0.431	0.166	0.124	0.089	3.34
5	0.159	0.295	0.132	0.241	0.172	3.47
21	0.069	0.073	0.065	0.481	0.313	2.34

Note: The table reports averages over all S&P 100 test observations for SRM estimated by QLIKE. Effective channels equal the inverse Herfindahl index of the five weights, averaged across observations.

The weights also vary across market states. At $h = 5$, the curvature weight increases from 22.1% in normal periods to 32.7% in high-volatility periods, and the global-geometry weight rises from 15.6% to 23.9%. The velocity weight falls from 31.4% to 21.7%. SRM therefore places greater weight on turning behavior and global path form when volatility is high.

The weight distribution does not by itself imply better forecasts. Curvature and global geometry receive the largest weights at $h = 21$, where SRM performs poorly. The learned weights describe how the model organizes geometric information; they do not establish the predictive contribution of an individual channel.

6 Robustness

We examine whether the results depend on the number of retrieved paths. The baseline SRM selects the 20 paths with the smallest composite geometric measure. We repeat the five-day QLIKE-estimated forecasts using $K_{\mathrm{NN}} \in \{10, 20, 40, 60\}$ and leave all other settings unchanged.

Table 9: Sensitivity to the number of retrieved paths

K_{NN}	MSE	QLIKE	MAE	ΔMSE (%)	ΔQLIKE (%)	ΔMAE (%)
10	15.983811	0.008303	2.332098	3.31	3.59	3.06
20	15.471597	0.008015	2.262870	0.00	0.00	0.00
40	15.192197	0.007869	2.225142	−1.81	−1.82	−1.67
60	15.142005	0.007821	2.213417	−2.13	−2.42	−2.19

Note: Results are for the S&P 100 sample, $h = 5$, and QLIKE estimation. The last three columns report percentage changes relative to $K_{\mathrm{NN}} = 20$; negative values indicate lower loss.

Forecast losses decline as the number of retrieved paths increases. Relative to the baseline, $K_{\mathrm{NN}} = 40$ lowers all three losses by approximately 2%. Increasing the number to 60 yields a further, but smaller, reduction. The main findings are therefore not tied to the baseline choice of 20 neighbors.

7 Conclusion

This paper proposes SRM, a nonparametric model that forecasts realized volatility from the geometric similarity of historical RV paths. SRM represents one-dimensional RV paths as three-dimensional curves, measures similarity through five geometric channels, and uses a Transformer to combine these channels. The forecast itself remains an aggregation of the subsequent volatility changes of retrieved historical paths.

The empirical results show that path geometry contains useful information for short- and medium-horizon volatility forecasting. At the one- and five-day horizons, SRM produces lower MSE than all four benchmarks under both estimation criteria. At the five-day horizon, its MSE and MAE are also lower than those of the IV-augmented benchmarks, although SRM uses only RV. Its performance deteriorates at the 21-day horizon.

Further analysis yields three findings. First, using MSE rather than QLIKE for estimation does not change the horizon pattern of the results. Second, cross-asset retrieval improves all three losses relative to same-asset retrieval. Third, the channel weights vary across horizons and market states, while the results remain stable when the number of retrieved paths changes. These findings support geometric path similarity as a source of short- and medium-horizon predictive information. Improving long-horizon forecasts remains an open issue.

A S&P 100 Sample Audit

The common sample contains 1,256 trading days from 9 June 2021 to 9 June 2026. After intersecting the RV, IV, and daily-return panels and requiring finite observations throughout the common period, 91 stocks are retained. The retained tickers are:

AAPL, ABBV, ABT, ACN, ADBE, AMAT, AMD, AMGN, AMT, AMZN, AVGO, AXP, BA, BAC, BKNG, BLK, BMY, C, CAT, CL, CMCSA, COF, COP, CRM, CSCO, CVS, CVX, DE, DHR, DIS, DUK, EMR, GD, GILD, GM, GOOG, GOOGL, GS, HD, HON, INTC, INTU, ISRG, JNJ, JPM, KO, LIN, LLY, LMT, LOW, LRCX, MA, MCD, MDLZ, MDT, META, MRK, MS, MSFT, MU, NEE, NFLX, NKE, NOW, NVDA, ORCL, PEP, PFE, PG, PLTR, PM, QCOM, RTX, SBUX, SCHW, SO, SPG, TMO, TMUS, TSLA, TXN, UBER, UNH, UNP, UPS, USB, V, VZ, WFC, WMT, XOM.

The ten common-universe tickers excluded by the completeness rule are BNY, BRK.B, COST, FDX, GEV, IBM, MMM, MO, T, and GE. The complete machine-readable audit is included with the replication materials.

References

1. Corsi, F. (2009). A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics*, 7(2), 174–196. DOI: 10.1093/jjfinec/nbp001.
2. Buccheri, G., and Corsi, F. (2019). HARK the SHARK: Realized volatility modeling with measurement errors and nonlinear dependencies. *Journal of Financial Econometrics*. DOI: 10.1093/jjfinec/nbz025.
3. Andersen, T. G., Bollerslev, T., Diebold, F. X., and Labys, P. (2003). Modeling and forecasting realized volatility. *Econometrica*, 71(2), 579–625. DOI: 10.1111/1468-0262.00418.
4. Christensen, B. J., and Prabhala, N. R. (1998). The relation between implied and realized volatility. *Journal of Financial Economics*, 50(2), 125–150. DOI: 10.1016/S0304-405X(98)00034-8.
5. Andrada-Félix, J., Fernández-Rodríguez, F., and Fuertes, A.-M. (2016). Combining nearest neighbor predictions and model-based predictions of realized variance: Does it pay? *International Journal of Forecasting*, 32, 695–715. DOI: 10.1016/j.ijforecast.2015.10.004.

6. Lorenz, E. N. (1969). Atmospheric predictability as revealed by naturally occurring analogues. *Journal of the Atmospheric Sciences*, 26(4), 636–646. DOI.
7. Delle Monache, L., Eckel, F. A., Rife, D. L., Nagarajan, B., and Searight, K. (2013). Probabilistic weather prediction with an analog ensemble. *Monthly Weather Review*, 141(10), 3498–3516. DOI.
8. Takens, F. (1981). Detecting strange attractors in turbulence. In D. Rand and L.-S. Young (Eds.), *Dynamical Systems and Turbulence, Warwick 1980*, Lecture Notes in Mathematics 898, 366–381. Springer. DOI: 10.1007/BFb0091924.
9. Farmer, J. D., and Sidorowich, J. J. (1987). Predicting chaotic time series. *Physical Review Letters*, 59(8), 845–848. DOI: 10.1103/PhysRevLett.59.845.
10. Yakowitz, S. (1987). Nearest-neighbour methods for time series analysis. *Journal of Time Series Analysis*, 8(2), 235–247. DOI: 10.1111/j.1467-9892.1987.tb00435.x.
11. Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3), 432–441. DOI: 10.1093/biostatistics/kxm045.
12. Patton, A. J. (2011). Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics*, 160(1), 246–256. DOI: 10.1016/j.jeconom.2010.03.034.
13. Zhang, C., Pu, X., Cucuringu, M., and Dong, X. (2024). Graph-based methods for forecasting realized covariances. *Journal of Financial Econometrics*, 23(2). DOI.
14. Zhang, C., Pu, X., Cucuringu, M., and Dong, X. (2025). Forecasting realized volatility with spillover effects: Perspectives from graph neural networks. *International Journal of Forecasting*, 41, 377–397. DOI: 10.1016/j.ijforecast.2024.09.002.
15. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30. URL.
16. Lim, B., Arık, S. Ö., Loeff, N., and Pfister, T. (2021). Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4), 1748–1764. DOI: 10.1016/j.ijforecast.2021.03.012.