International Journal of Forecasting 41 (2025) 377–397

Contents lists available at ScienceDirect

International Journal of Forecasting

journal homepage: www.elsevier.com/locate/ijforecast

Forecasting realized volatility with spillover effects: Perspectives from graph neural networks☆

Chao Zhang [a,b,*,1], Xingyue Pu [b,d,1], Mihai Cucuringu [b,c,e], Xiaowen Dong [b,d]

[a] FinTech Thrust, HKUST(GZ), Guangzhou, China
[b] Oxford-Man Institute of Quantitative Finance, University of Oxford, Oxford, UK
[c] Department of Statistics, University of Oxford, Oxford, UK
[d] Department of Engineering Science, University of Oxford, Oxford, UK
[e] The Alan Turing Institute, London, UK

ARTICLE INFO

Article history:
Dataset link: LOBSTER, github.com/chaozhang-ox/GNNHAR

Keywords:
Graph neural network
Realized volatility
Spillover effect
Quasi-likelihood
Nonlinearity

ABSTRACT

We present a novel nonparametric methodology for modeling and forecasting multivariate realized volatilities using customized graph neural networks to incorporate spillover effects across stocks. The proposed model offers the benefits of incorporating spillover effects from multi-hop neighbors, capturing nonlinear relationships, and flexible training with different loss functions. The empirical findings suggest that incorporating spillover effects from multi-hop neighbors alone does not yield a clear advantage in terms of predictive accuracy. Furthermore, modeling nonlinear spillover effects enhances the forecasting accuracy of realized volatilities, particularly for short-term horizons of up to one week. More importantly, our results consistently indicate that training with the quasi-likelihood loss leads to substantial improvements in model performance compared to the commonly used mean squared error, primarily due to its superior handling of heteroskedasticity. A comprehensive series of empirical evaluations in alternative settings confirm the robustness of our results.

1. Introduction

Modeling and forecasting stock return volatility plays a crucial role in the theory and practice of finance. Extensive attention has been devoted to this subject within the literature, encompassing numerous ARCH, GARCH, and stochastic volatility models. Due to the availability of high-frequency data, realized volatility (RV), calculated from the sum of squared intraday returns, has gained popularity in recent years. For example, Corsi (2009) put forward the heterogeneous autoregressive (HAR) model for predicting daily RVs using various lagged RV components over different time horizons. While these methods provided valuable insights into the dynamic dependence of volatilities, they neglected the volatility spillover effect among assets, as highlighted by Bollerslev, Hood, et al. (2018).

The volatility spillover effect refers to the phenomenon that certain big shocks of a specific asset (or market) may have an influence on the volatilities of other assets (or markets). Essentially, the discovery of volatility spillover effects is expected to benefit the understanding and forecasting of volatilities. For example, Buncic and Gisler (2016) documented that the VIX of the U.S. market plays an important role in forecasting the volatilities of other global assets markets. Degiannakis and Filis (2017) examined the cross-asset spillover effects from stocks, currencies, and commodities to improve RV predictions

☆ An earlier version of this article circulated under the title "Graph Neural Networks for Forecasting Realized Volatility with (Nonlinear) Spillover Effects".

* Corresponding author at: FinTech Thrust, HKUST(GZ), Guangzhou, China.
E-mail address: chaoz@hkust-gz.edu.cn (C. Zhang).

[1] The first two authors contributed equally to this work.

https://doi.org/10.1016/j.ijforecast.2024.09.002

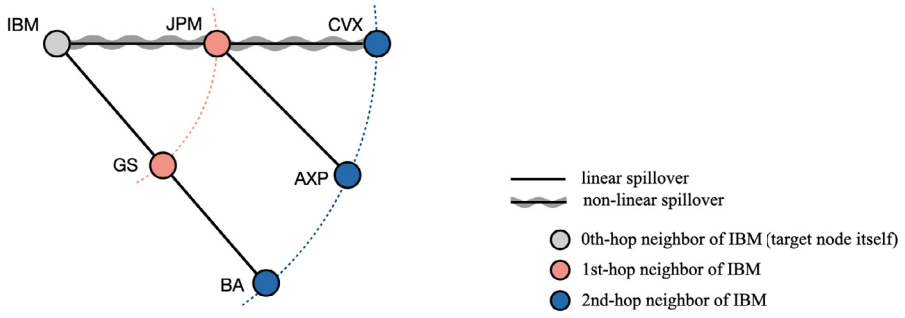


Fig. 1. Illustration of multi-hop and nonlinear volatility spillover.
Note: The target node represents the volatility of IBM. The connections are only for illustration, and hence not necessarily consistent with our experiments.

of crude oil. Bollerslev, Hood, et al. (2018), Li and Tang (2021) utilized the commonality in risk structures to improve the forecasting of future volatility. Basturk et al. (2022), Zhang et al. (2024) applied neural networks to predict volatility, finding that cross-sectional data pooling enhances forecasting accuracy.

There are a number of studies dedicated to incorporating the spillover effect into volatility modeling, e.g. BEKK-GARCH (Engle & Kroner, 1995) and VAR-GARCH (Ling & McAleer, 2003). In terms of modeling RV, Wilms et al. (2021) employed vector autoregression (VAR) to obtain the multivariate volatility forecasts for stock market indices. However, in high-dimensional scenarios, the aforementioned models may deliver poor out-of-sample forecasts due to the curse of dimensionality, as emphasized by Callot et al. (2017). Hecq et al. (2023) studied volatility spillovers using Granger causality analysis with VARs based on penalized least squares estimations. Most recently, Zhang et al. (2022) introduced graph-based methods to capture volatility spillover effects, and proposed a parsimonious model to augment HAR via neighborhood aggregation on a graph that represents a financial network, denoted graph HAR (or GHAR). In these graphs, each asset is modeled as a node, and an edge connecting two nodes encodes the existence of the spillover effect between their volatilities.

One natural question following GHAR is whether there exists any spillover effect between nodes that are beyond one step, also known as multi-hop neighbors (see the detailed definitions in Section 2.1). For example, as illustrated in Fig. 1, for the target node (i.e. IBM), in addition to the spillover effect of one-hop neighbors (i.e. JPM and GS), we are also interested in whether there is any spillover effect from two-hop neighbors (i.e. AXP, CVX, and BA). To the best of our knowledge, the exploration of spillover effects from multi-hop neighbors has not been extensively addressed in the literature on volatility modeling.[2]

In addition to multi-hop effects, another interesting question is whether the volatility spillover is nonlinear. Choudhry et al. (2016) documented the existence of significant nonlinear spillover effects among four major markets—the U.S., Canada, Japan, and the U.K.—via a nonlinear causality test proposed by Bai et al. (2010). Wang et al. (2018) attempted to capture the nonlinear relationship between the volatilities of stocks and crude oil by incorporating the asymmetric effect of oil prices and regime shifts. While the existence of nonlinear volatility spillover effects has been documented and examined in previous studies, in this paper, we employ a deep learning approach (a graph neural network) to unveil nonparametric evidence about the existence of a nonlinear mechanism between cross-sectional volatilities, without explicitly assuming the presence of asymmetric effects or regime shifts in the pairwise interactions.

From a machine learning perspective, the incorporation of multi-hop neighbors expands the set of features, and the potential presence of nonlinear spillover effects introduces new functional forms to describe volatility dynamics. It is also worth emphasizing that the choice of estimation criterion (EC) plays a crucial role, as it represents the objective function for estimating model parameters. Traditional econometric models, such as GARCH, commonly employ conditional quasi-likelihood (QL) based on normal distributions for parameter estimation. Conversely, models focused on forecasting realized volatilities, such as HAR, utilize the mean squared error (MSE) as their EC. Therefore, an important question arises as to whether a preferred EC exists (Cipollini et al., 2020), especially when combined with the aforementioned aspects, namely the effect of multi-hop neighbors and non-linear relationships.

In the present work, we explore these three questions using graph neural networks (GNNs). GNNs are a class of deep learning models designed for performing inferences on graphs and graph-structured data. They are capable of learning node and graph-level representations that are useful for a wide range of tasks involving graph analysis, such as node classification, node regression, and graph clustering. GNNs have demonstrated successful applications in various financial domains, including stock movement prediction (Chen et al., 2018; Sawhney et al., 2020), credit risk prediction (Liang et al., 2021; Wang et al., 2019), and payment fraud detection (Liu et al., 2019, 2018). A recent study by Chen and Robert (2022) utilized a graph transformer network for intraday volatility

[2] Two-hop connections have been studied in the context of cascading effects of financial networks, e.g. Acemoglu et al. (2010), where the shocks that occur to an individual firm would propagate through the rest of the economy. Consequently, downstream firms more than one hop away may also suffer from the impact.

forecasting. However, it is worth mentioning that their approach had some limitations, particularly in terms of interpretability and benchmarking. In addition, they did not thoroughly investigate factors such as multi-hop neighbors, nonlinearity, or the impact of estimation criteria, which are the focus of our current study.

In particular, we design a GNN-based framework to model volatility spillover effects and enhance volatility predictions. By replacing the linear neighborhood aggregation in the GHAR of Zhang et al. (2022) with a nonlinear operation, the proposed model is able to automatically learn the nonlinear spillover effects. Furthermore, the multi-layer setting of GNNs allows us to explore this nonlinearity in the multi-hop setting, i.e. spillover to neighbors that are more than one hop away in the financial network. Another notable advantage of our model lies in its flexibility to accommodate various ECs during the training phase.[3] It should be emphasized that the goal here is not only to extend the original HAR model with neighborhood information but also to provide new perspectives from GNNs for the nonparametric modeling of volatility spillover effects, further improving the volatility forecasts.

The main contributions of our work are summarized as follows. First, we examine the spillover effect from multi-hop neighbors in the financial graph, and observe that the multi-hop spillover effect is not necessary, as long as zero-hop and one-hop neighbors are included. Second, we establish that the proposed GNN model with nonlinear operations significantly improves the forecasting performance of GHAR, indicating the existence of nonlinear spillover effects on one-hop neighbors. Third, compared to MSE-trained models, models employing QL as the EC generally achieve substantial improvements in predictive accuracy. With a further endeavor, we establish a natural link between QL-trained models and the multiplicative error model (Engle, 2002), highlighting their superior handling of error heteroskedasticity by assigning different degrees of importance to observations. Overall, our proposed GNN model trained with QL exhibits an average forecast error in MSE (resp. QL) approximately 13% (resp. 4%) lower than that of the standard HAR model. Additionally, we examine the robustness of our proposed models across various market conditions, an alternative data-splitting scheme, and an alternative universe, consistently observing enhanced prediction accuracy across all experimental settings.

The remainder of this paper is organized as follows. Section 2 contains preliminaries on the mathematical definitions of graphs, a brief review of GNN models, and two baseline models (HAR and GHAR). In Section 3, we introduce the proposed model (GNNHAR), evaluation criterion, and forecast evaluation approaches. Section 4 outlines the experimental setup and provides the key out-of-sample results across various forecast horizons and market regimes. Furthermore, in Section 5, we conduct an extensive analysis concerning the impact of QL, nonlinearity, and multi-hop neighbors. In Section 6, we perform several robustness tests. We conclude our work and highlight future research directions in Section 7.

[3] Note that the adaptability in selecting ECs is not exclusive to GNN models and can be applied to various ML models.

2. Preliminaries

In this section, we summarize the preliminary concepts and models. In particular, we provide the mathematical definitions of graphs and multi-hop neighbors in Section 2.1. In Section 2.2, we briefly review two popular graph neural networks that inspired our work. Section 2.3 revisits the baseline model HAR for forecasting realized volatilities, while Section 2.4 reviews another baseline model GHAR. Throughout this paper, capital bold letters indicate matrices, lowercase bold letters indicate vectors, and plain letters indicate scalars.

2.1. Graph definitions

Definition 2.1 (*Graphs*)**.** A graph $\mathcal{G}$ is defined as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ is a set of N nodes and $\mathcal{E}$ is a set of edges, where $e_{ij} = (v_i, v_j) \in \mathcal{E}$ denotes an edge connecting node v_i and node v_j.

Definition 2.2 (*Adjacency Matrix*)**.** An adjacency matrix $\mathbf{A}$ is a square matrix whose dimension is $N \times N$, where $\mathbf{A}[i, j]$ represents the connection between v_i and v_j in the graph $\mathcal{G}$. If $\mathbf{A}[i, j] \in \{0, 1\},\ \forall i, j$, the graph is a binary graph.[4] The diagonal elements of $\mathbf{A}$ are all zero, since edges from a node to itself are typically not considered in graphs. In this article, we mainly consider binary graphs without self-connections.

Definition 2.3 (*K-hop Neighbors*)**.** Following Feng et al. (2022), we use the K-hop neighbors of node v to represent all the neighbors that have distance from node v less than or equal to K, based on the shortest path distance (SPD) kernel. In contrast, k-hop neighbors represent the neighbors with exact distance k from node v. Finally, we denote $Q^K_{v,\mathcal{G}}$ as the set of K-hop neighbors of node v in graph $\mathcal{G}$.

Example 1 (*A Graph with 5 Nodes*)**.** In Fig. 2(a), we plot an example graph with five nodes and five undirected edges, where the node v_1 is colored as a target node. Nodes v_2 and v_4 are the one-hop and two-hop neighbors of v_1, respectively. Fig. 2(b) shows its adjacency matrix.

2.2. A brief review of GNNs

Graph neural networks (GNNs) are a class of deep learning models designed for performing inferences on graphs. The main idea is to learn a vector representation for every node defined on a graph while preserving both the graph topology structure and node content information (Wu et al., 2020). The node representations, for example, can be further applied to node classification or regression. To this end, many GNN variants utilize the idea of neighborhood aggregation to develop the layer-wise forward propagation rules. In essence, neighborhood

[4] An adjacency matrix can be weighted, where $\mathbf{A}[i, j] \geq 0,\ \forall i, j$ represents the strength/intensity of the connection between nodes v_i and v_j.

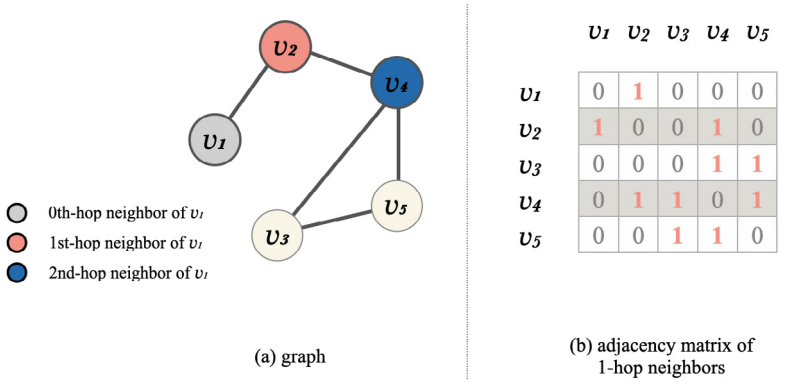


Fig. 2. Illustration of a graph and its corresponding adjacency matrix.

aggregation effectively generates a node v's representation by aggregating its own feature vector $\boldsymbol{h}_v \in \mathbb{R}^D$ and the feature vectors of its connected nodes $\boldsymbol{h}_u \in \mathbb{R}^D$, where $u \in Q^1_{v,\mathcal{G}}$. Common examples of aggregation functions include sum, mean, and maximum. Early attempts at GNNs—regarding which, see Dai et al. (2018) and Scarselli et al. (2008)—update node representations by aggregating neighborhood information recursively until a stable equilibrium is reached. More efficiently, a novel notion of a convolution operator can be defined on irregular graphs to process neighborhood aggregation in parallel (so-called graph convolution).[5] A considerable number of GNN variants and architectures are built from different graph convolution operators. We provide a brief introduction to a specific GNN architecture that is relevant to our volatility forecasting models.

The graph convolutional network (GCN) was introduced by Kipf and Welling (2017). It approximates the graph convolution with the following layer-wise propagation rule[6]:

$$\boldsymbol{H}^{(l+1)} = \sigma\left(\tilde{\boldsymbol{O}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{O}}^{-\frac{1}{2}}\boldsymbol{H}^{(l)}\boldsymbol{\Theta}^{(l)}\right), \tag{1}$$

where $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}_N$ is the adjacency matrix of the graph $\mathcal{G}$ with added self-connections, and $\tilde{\boldsymbol{O}}$ is a diagonal matrix with $\tilde{\boldsymbol{O}}_{ii} = \sum_j \tilde{\boldsymbol{A}}_{ij}$. $\boldsymbol{A}$ is the regular adjacency matrix of one-hop neighbors. $\tilde{\boldsymbol{O}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{O}}^{-\frac{1}{2}}$ is the normalized adjacency matrix, introduced to stabilize the training of the GNN models. $\boldsymbol{\Theta}^{(l)} \in \mathbb{R}^{D^{(l)}\times D^{(l+1)}}$ is the layer-specific trainable weight matrix. $\boldsymbol{H}^{(l)} \in \mathbb{R}^{N\times D^{(l)}}$ is the matrix of node representations at the lth layer. $\boldsymbol{H}^{(0)}$ is the input node features. $\sigma(\cdot)$ denotes a nonlinear activation function, such as $\mathrm{ReLU}(\cdot) = \max(0, \cdot)$.

When addressing various research problems, the above GNN layers can be combined with other deep learning layers in an end-to-end learning framework. Additionally, the exploration of multi-hop effects can be achieved by straightforwardly stacking multiple GNN layers within a model. A model that incorporates K-layer GNN layers is commonly referred to as a K-layer GNN model.

Definition 2.4 (*Receptive Field*)**.** In a GNN model, the receptive field of a target node is the set of nodes of the graph that determine its representations; see Alon and Yahav (2020) and Feng et al. (2022).

Proposition 2.1. *After K layers of graph convolution in a GNN model, every node representation is determined by the information from the nodes within K hops; see Feng et al. (2022).*

The above proposition states that the size of the receptive field of every node is associated with the number of layers in a GNN model. Alon and Yahav (2020) found that when K is unnecessarily large, any two nodes could easily have highly overlapping receptive fields, and consequently attain highly similar node representations, which leads to the problem of over-smoothing (see Chen et al., 2020; Li et al., 2018). Therefore, a large K does not always help, and on the contrary, it may lead to indistinguishable node representations and thus weaken the forecasting or classification accuracy.

2.3. Forecasting RV with HAR

Assume the price process $P_{i,s}$ of a financial asset i follows

$$\mathrm{d}\log P_{i,s} = \mu_i \mathrm{d}s + \sigma_{i,s}\mathrm{d}W^i_s, \tag{2}$$

where μ_i is the drift, $\sigma_{i,s}$ is the instantaneous volatility, and W^i_s is the standard Brownian motion. The integrated variance (IV) of asset i at day t is defined as $IV_{i,t} = \int_{t-1}^{t}\sigma^2_{i,s}\mathrm{d}s$.

Andersen et al. (2001) and Barndorff-Nielsen and Shephard (2002) showed that the sum of squared intraday returns is a consistent estimator of the unobserved $IV_{i,t}$. The daily RV for a particular asset i at day t is defined

[5] Convolution operations have been widely applied to regular grid data, e.g. image pixels. Recently, they have been extended to graph-structured data. More details can be found in Shuman et al. (2013).

[6] The GCN propagation rule approximates the graph convolution with the first-order Chebyshev spectral polynomials (ChebyNet). It alleviates gradient vanishing/exploding and stabilizes the training in ChebyNet by introducing a normalization step on $\boldsymbol{A}$. More details about ChebyNet can be found in Defferrard et al. (2016).

as $RV_{i,t} = \sum_{l=1}^{M} r_{i,t(l)}^2$, where $r_{i,t(l)}$ is the lth Δ-min log returns during day t, i.e. $r_{i,t(l)} = \log p_{i,t(l\Delta)} - \log p_{i,t((l-1)\Delta)}$, and $p_{i,t(l\Delta)}$ is the price at time $l\Delta$ at day t. We refer to $\boldsymbol{v}_t = (RV_{1,t}, \ldots, RV_{N,t})'$ as the vector of cross-sectional realized volatilities. Here, we consider five-minute windows in a trading day, following Liu et al. (2015).[7]

Corsi (2009) proposed a heterogeneous autoregressive regression (HAR) model for modeling and forecasting RV where the lagged daily, weekly, and monthly volatility components are incorporated as features. Bollerslev, Hood, et al. (2018) recommended using pooled panel data instead of time-series data to improve the accuracy of RV forecasts. We adopt this approach to make the most of cross-sectional information. As a result, we model the cross-sectional RV for day t as follows:

$$\begin{aligned}\textbf{HAR}: \quad \mathbb{E}(\boldsymbol{v}_t|\mathcal{F}_{t-1}) &= \boldsymbol{\alpha} + \beta_d \boldsymbol{v}_{t-1} + \beta_w \boldsymbol{v}_{t-5:t-2} \\ &\quad + \beta_m \boldsymbol{v}_{t-22:t-6}, \\ &= \boldsymbol{\alpha} + \boldsymbol{V}_{:t-1}\boldsymbol{\beta},\end{aligned} \tag{3}$$

where $\mathcal{F}_{t-1}$ is the information set consisting of all relevant information up to and including $t-1$. $\boldsymbol{v}_{t-5:t-2} = \frac{1}{4}\sum_{k=2}^{5} \boldsymbol{v}_{t-k}$ and $\boldsymbol{v}_{t-22:t-6} = \frac{1}{17}\sum_{k=6}^{22} \boldsymbol{v}_{t-k}$ denote the weekly and monthly lagged RV, respectively,[8] and $\boldsymbol{V}_{:t-1} = [\boldsymbol{v}_{t-1}, \boldsymbol{v}_{t-5:t-2}, \boldsymbol{v}_{t-22:t-6}] \in \mathbb{R}^{N\times 3}$. The choice of a daily, weekly, and monthly lag aims to capture the long-memory dynamic dependencies observed in most RV series.

2.4. Graph HAR (GHAR)

Zhang et al. (2022) augmented the HAR model to capture the volatility spillover effect via linear neighborhood aggregation on graphs.[9] GHAR is defined as

$$\begin{aligned}\textbf{GHAR}(\boldsymbol{A}): \quad \mathbb{E}(\boldsymbol{v}_t|\mathcal{F}_{t-1}) &= \boldsymbol{\alpha} + \beta_d \boldsymbol{v}_{t-1} + \beta_w \boldsymbol{v}_{t-5:t-2} \\ &\quad + \beta_m \boldsymbol{v}_{t-22:t-6} \\ &\quad + \gamma_d \boldsymbol{W}\cdot\boldsymbol{v}_{t-1} + \gamma_w \boldsymbol{W} \\ &\quad \cdot \boldsymbol{v}_{t-5:t-2} + \gamma_m \boldsymbol{W}\cdot\boldsymbol{v}_{t-22:t-6}, \\ &= \boldsymbol{\alpha} + \boldsymbol{V}_{:t-1}\boldsymbol{\beta} + \boldsymbol{W}\boldsymbol{V}_{:t-1}\boldsymbol{\gamma},\end{aligned} \tag{4}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^N$, $\boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{R}^3$ are parameters to be estimated. $\boldsymbol{W} = \boldsymbol{O}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{O}^{-\frac{1}{2}}$ is the normalized adjacency matrix without self-connections, where $\boldsymbol{O} = \text{diag}\{n_1, \ldots, n_N\}$ and $n_i = \sum_j \boldsymbol{A}[i,j]$, $\forall i$.[10]

$\boldsymbol{W}\cdot\boldsymbol{v}_{t-1}$ represents the neighborhood aggregation over daily horizons, and similarly for weekly and monthly horizons. γ_d, γ_w, and γ_m represent the effects of connected neighbors over different horizons. If we employ an empty graph, i.e. the elements of $\boldsymbol{A}$ are all zeros, (4) reduces to (3). When the off-diagonal elements of $\boldsymbol{A}$ are all ones, i.e. a complete graph, $\boldsymbol{W}\cdot\boldsymbol{v}_{t-1}$ represents the global volatility, as studied by Bollerslev, Hood, et al. (2018).

3. Proposed methodology

To investigate the presence of multi-hop and nonlinear effects in modeling volatility spillovers, we propose a new class of forecasting models based on the GNNs in Section 3.1. Furthermore, Section 3.2 highlights the significance of using various criteria to estimate model coefficients. In Section 3.3, we introduce the forecast evaluation methods and emphasize the differences between estimation criteria and forecast evaluations.

3.1. GNN-enhanced HAR (GNNHAR)

As introduced in (4), GHAR in Zhang et al. (2022) assumes a linear relationship between the volatilities of two connected assets. However, if the spillover effect is nonlinear, linear models are misspecified and are likely to generate less accurate forecasts. Additionally, GHAR considers only the zero-hop and one-hop neighbors, and this lack of consideration for multi-hop neighbors may lead to incomplete information and less accurate predictions. In light of the abilities of GNNs discussed in Section 2, we propose the following GNN architecture for modeling the volatility spillover effect, allowing for nonlinearity and multi-hop neighbors to improve the prediction accuracy.

$$\textbf{GNN}(\boldsymbol{H}^{(l)}, \boldsymbol{A}): \quad \boldsymbol{H}^{(l+1)} = \text{ReLU}\left(\boldsymbol{O}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{O}^{-\frac{1}{2}}\boldsymbol{H}^{(l)}\boldsymbol{\Theta}^{(l)}\right), \tag{5}$$

where $\boldsymbol{W} = \boldsymbol{O}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{O}^{-\frac{1}{2}}$ is the normalized adjacency matrix, used to avoid numerical instabilities and exploding/vanishing gradients during the training phrase. Note that $\boldsymbol{H}^{(\boldsymbol{0})} = \boldsymbol{V}_{:t-1} \in \mathbb{R}^{N\times 3}$, which is the matrix composed of the past daily, weekly, and monthly volatilities. $\boldsymbol{H}^{(l)} \in \mathbb{R}^{N\times D^{(l)}}$ is a matrix of node representations at the lth layer of the GNN, where $D^{(l)}$ is the dimension of node representations. $\boldsymbol{\Theta}^{(l)} \in \mathbb{R}^{D^{(l)}\times D^{(l+1)}}$ is a matrix of trainable parameters (see Fig. 3).

In contrast to the GCN architecture shown in (1), our proposed GNN propagation rule does not include self-connections; i.e. the diagonal elements in $\boldsymbol{A}$ are zeros. We conjecture that the mechanism of an individual stock's past volatility on its future volatility differs from the spillover effect. As a result, we apply the above GNN propagation in (5) solely to model the spillover effect, while the impact of a stock's own past volatility is modeled using the same linear model as in HAR.[11] This allows for a clear and straightforward explanation of the performance gain of our proposed model compared to the baseline models, HAR and GHAR.

[7] We also adopt the subsampling averaging method (see Andersen et al., 2011; Sheppard, 2010; Varneskov & Voev, 2013) to improve the above *RV* estimation, which uses all Δ-minute returns, not just non-overlapping ones.

[8] Compared to the original definition in Corsi (2009), the current definition, in line with Cipollini et al. (2021) and Patton and Sheppard (2015), aims to isolate the impact of individual past components more distinctly. In the context of linear models, both definitions yield equivalent outcomes, except for the coefficients.

[9] A related model is the spatial autoregressive/lag model, which operates as a simultaneous spatial model but does not account for temporal dependency (Anselin, 2022; LeSage, 1999).

[10] It is worth noting that for GHAR, the normalization of $\boldsymbol{W}$ does not impact the forecasting performance directly. However, it does assist with evaluating the relative effect of zero-hop neighbors in comparison to one-hop neighbors.

[11] In this paper, our primary focus is on investigating spillover effects. However, for a more comprehensive understanding of the nonlinearity between a stock's past volatility and its future volatility, we refer to Bucci (2020), Li and Tang (2021) and Zhang et al. (2024).

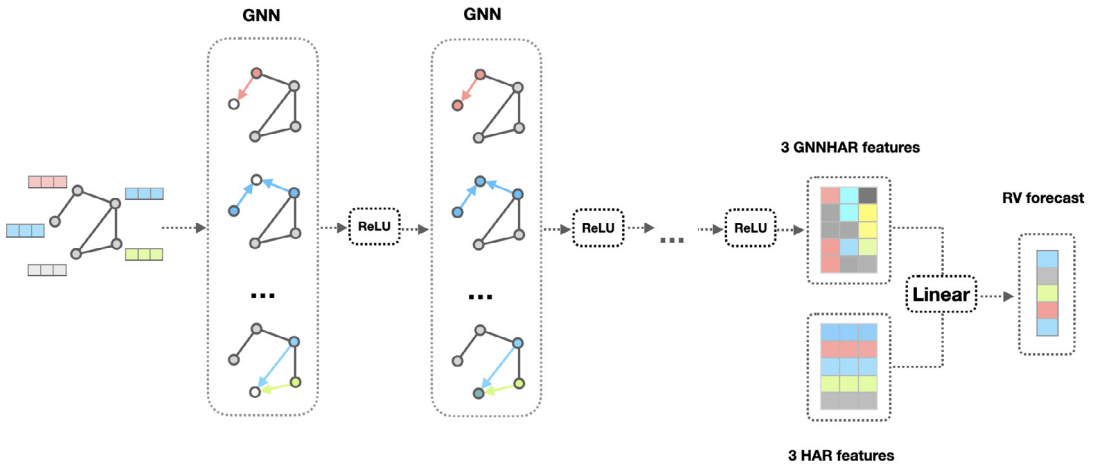


Fig. 3. Illustration of the GNNHAR model.

We introduce a GNN-enhanced HAR model, referred to as GNNHAR1L in (6), by replacing the linear neighborhood aggregation in GHAR (i.e. the term $\boldsymbol{WV}_{:t-1}\boldsymbol{\gamma}$ in (4)) with the proposed GNN layer in (5). It is worth noting that the main difference between GNNHAR1L and GHAR is that GNNHAR1L uses a graph convolutional layer with a nonlinear activation function, in the form of

$$\begin{aligned} \boldsymbol{H}^{(1)} &= \mathrm{GNN}(\boldsymbol{V}_{:t-1},\ \boldsymbol{A}), \\ \textbf{GNNHAR1L}(\boldsymbol{A}):\quad \mathbb{E}(\boldsymbol{v}_t|\mathcal{F}_{t-1}) &= \boldsymbol{\alpha} + \boldsymbol{V}_{:t-1}\boldsymbol{\beta} + \boldsymbol{H}^{(1)}\boldsymbol{\gamma}. \end{aligned} \tag{6}$$

As introduced in Section 2, the nonlinear multi-hop effects can be explored by stacking multiple layers of the GNN. We denote the two-layer and three-layer models as GNNHAR2L and GNNHAR3L, respectively.[12] Specifically,

$$\begin{aligned} \boldsymbol{H}^{(2)} &= \mathrm{GNN}(\boldsymbol{H}^{(1)},\ \boldsymbol{A}), \\ \textbf{GNNHAR2L}(\boldsymbol{A}):\quad \mathbb{E}(\boldsymbol{v}_t|\mathcal{F}_{t-1}) &= \boldsymbol{\alpha} + \boldsymbol{V}_{:t-1}\boldsymbol{\beta} + \boldsymbol{H}^{(2)}\boldsymbol{\gamma}. \end{aligned} \tag{7}$$

$$\begin{aligned} \boldsymbol{H}^{(3)} &= \mathrm{GNN}(\boldsymbol{H}^{(2)},\ \boldsymbol{A}), \\ \textbf{GNNHAR3L}(\boldsymbol{A}):\quad \mathbb{E}(\boldsymbol{v}_t|\mathcal{F}_{t-1}) &= \boldsymbol{\alpha} + \boldsymbol{V}_{:t-1}\boldsymbol{\beta} + \boldsymbol{H}^{(3)}\boldsymbol{\gamma}. \end{aligned} \tag{8}$$

Our empirical analysis (deferred to Appendix A) indicates that each node in the volatility spillover graphs for the components of the DJIA 30 index, chosen by GLASSO (see 3.1.1), is connected to other nodes within a maximum of three steps (i.e. the graph has a diameter of length three, which is the size of the longest shortest pairwise path distance in the graph).

Consequently, by employing a three-layer GNN, we can guarantee that the volatility representation of each asset encompasses information from all other assets. Hence, there is no requirement to investigate beyond a three-layer GNN. Nevertheless, it is worth noting that for different universes or graphs, the number of GNN layers may need to be re-evaluated according to the distribution of SPDs.

3.1.1. Graph construction

Before training the GNN models or GHAR, it is essential to predefine the adjacency matrix or graph structure. In much of the GNN literature, the graph structure, such as a citation network, is explicitly defined. Unlike these applications, financial graphs require estimation, typically from time series analyses of price-based economic variables. For example, Diebold and Yılmaz (2014) studied the connectedness built from variance decompositions, while Karpman et al. (2023) leveraged random forests alongside high-frequency trading data to infer edge relationships. Zhang et al. (2022) constructed different types of graphs for volatility modeling and concluded that adjacency matrices obtained through graphical LASSO (GLASSO) effectively capture the relationships between individual volatilities, thereby enhancing forecasting accuracy.

GLASSO was proposed by Friedman et al. (2008) as a sparsity-penalized maximum likelihood estimator for the precision matrix $\boldsymbol{\Theta}$ (i.e. the inverse of the covariance matrix). It assumes that the input N-dim vector is drawn from a multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} = \boldsymbol{\Theta}^{-1}$.[13] A principal advantage of GLASSO is its capacity to reveal the conditional independence between variables, here the assets, through the estimated precision matrix. If the ij-th entry of the precision matrix is zero, the ith asset and jth asset are conditionally independent. Therefore, the adjacency matrix $\boldsymbol{A}$ obtained by applying GLASSO to multivariate daily returns is defined as $\boldsymbol{A}[i,j] = 1$ if $\boldsymbol{\Theta}[i,j] \neq 0$; otherwise $\boldsymbol{A}[i,j] = 0$. Based on these compelling results of GLASSO, we adopt it to construct the adjacency matrix for graph-based models throughout this paper.[14]

[12] Furthermore, we introduce a linear model that incorporates multi-hop neighbors for volatility forecasting. Additional results regarding this model can be found in Appendix E.

[13] The Gaussian assumption might seem overly simplistic given the complex and often non-Gaussian nature of financial returns, which can exhibit heavy tails and skewness. However, GLASSO serves as a starting point for estimating the conditional dependencies of financial assets. One interesting direction of future work would be to explore models that can accommodate these unique characteristics of financial returns, like Liu et al. (2009) and Voorman et al. (2014).

[14] Note that the hyperparameter that determines the sparsity of GLASSO graphs is chosen by k-fold cross-validation. This training/validation setup aligns with the GNN model training/validation setup detailed in Section 4.1.

3.2. Estimation criterion

The standard HAR model described in (3) is often estimated via ordinary least squares (OLS). In other words, the estimation criterion (EC) for its in-sample training is the MSE. When the errors in (3) are independent, homoscedastic, and normally (Gaussian) distributed, the OLS estimator is consistent under the asymptotic sense. Nonetheless, given the stylized facts of RV (such as heteroskedasticity and so on), the OLS estimator may not be an ideal choice and a better estimator may be available. For example, Hansen and Dumitrescu (2022) proved that the likelihood-based estimator is asymptotically efficient, although the likelihood-based estimator can also be vastly inferior if the underlying statistical model is misspecified. Clements and Preve (2021) empirically compared various estimation criteria on HAR and found that simple weighted least squares can yield substantial improvements to the predictive ability of the standard HAR.

Meanwhile, QL has served as a commonly employed metric for estimating traditional econometric models, including GARCH. Fan et al. (2014) and Hall and Yao (2003) demonstrated that the conditional Gaussian QL estimator is always consistent, even when the error term deviates from a normal distribution.

Utilizing the flexibility of neural networks and stochastic gradient descent algorithms, we are able to investigate whether different estimation criteria would result in disparate model predictions. Specifically, our primary focus revolves around the following estimation criteria: MSE and QL, defined as follows:

- MSE:

$$\frac{1}{N}\sum_{i=1}^{N}\frac{1}{\#\mathcal{T}_{train}}\sum_{t\in\mathcal{T}_{train}}\left(RV_{i,t}-\widehat{RV}_{i,t}^{(F)}\right)^2, \tag{9}$$

- QL:

$$\frac{1}{N}\sum_{i=1}^{N}\frac{1}{\#\mathcal{T}_{train}}\sum_{t\in\mathcal{T}_{train}}\left[\frac{RV_{i,t}}{\widehat{RV}_{i,t}^{(F)}}-\log\left(\frac{RV_{i,t}}{\widehat{RV}_{i,t}^{(F)}}\right)-1\right], \tag{10}$$

where $\widehat{RV}_{i,t}^{(F)}$ represents the predicted value of $RV_{i,t}$ by a specific model F, N is the number of stocks in our universe, $\mathcal{T}_{train}$ is the training period, and $\#\mathcal{T}_{train}$ is the length of the training period.

Lower values are preferred for both measures. For clarity, we use F_M (F_Q) to denote model F trained with MSE (QL). To the best of our knowledge, adopting QL as the estimation criterion to optimize volatility models, especially those grounded on neural networks, has not yet drawn considerable attention in the literature. An exception can be found in the work of Cipollini et al. (2020), who conducted an empirical assessment of the impact of various error criteria on linear HAR models. They observed that using QL led to slightly improved forecasts, though without offering further theoretical explanations.

In Appendix B, we show that the models trained with QL are linked to the multiplicative error model (MEM) by Engle (2002). Hence, the comparison between models trained with MSE and QL essentially boils down to the comparison between additive models and multiplicative models (see below). According to Cipollini et al. (2021), those additive models have issues related to the heteroskedasticity of errors (u_t). However, when considering multiplicative models, the errors (z_t) tend to be homoskedastic.

$$RV_t=\begin{cases}\mathbb{E}\left(RV_t|\mathcal{F}_{t-1}\right)+u_t, & u_t \text{ zero mean}\\ \mathbb{E}\left(RV_t|\mathcal{F}_{t-1}\right)\times z_t, & z_t \text{ unit mean.}\end{cases}$$

From an empirical perspective, Clements and Preve (2021), Patton and Sheppard (2015) and Reisenhofer et al. (2022) estimated their models using different schemes of weighted least squares (WLS) to assign less importance during estimation to periods where volatility is less precisely estimated.

Next, we examine the weighting scheme implicitly employed in QL-trained HAR models. Fig. 4 first displays the aforementioned EC for different forecasts $\widehat{RV}$ when $RV = 1$. Notably, the QL function exhibits asymmetry and imposes a higher penalty on under-predictions. This feature becomes particularly significant during turbulent periods, as the volatility forecasts tend to be smaller than the actual shocks. By placing emphasis on those under-predictions, models trained with QL have the potential to achieve improved prediction accuracy during such turbulent periods.

Proposition 3.1. *The optimization of HAR models trained with QL can be achieved through iteratively reweighted least squares (IRLS), employing weights $w_{k-1,t} = 1/\widehat{RV}_{k-1,t}^2$ at iteration k. Here, $\widehat{RV}_{k-1,t}$ represents the fitted value from the preceding iteration (with the initial iteration performed using OLS).*

This proposition further validates the observation that models trained with QL give greater emphasis to under-predictions. Its proof is provided in Appendix C. In line with Cipollini et al. (2021) and Clements and Preve (2021), we are not asserting the optimality of the weighting scheme in QL-trained models. Additionally, our analysis, limited to comparing two statistical loss functions for realized volatility, may not comprehensively justify the superiority of QL across various applications in finance.[15] Nonetheless, they could serve as valuable benchmarks, due to their natural relations with the MEM and WLS, and their desirable theoretical properties.[16]

3.3. Forecast evaluation approaches

Regarding the performance of forecasts in out-of-sample tests, we continue to employ MSE and QL as our

[15] For example, Goyenko et al. (2024) proposed an economic loss for comparing volume forecasts within a mean-variance portfolio framework that trades off tracking error versus net-of-cost performance.

[16] The estimator obtained through maximum likelihood exhibits desirable properties, such as consistency and asymptotic theory (see Bauwens et al., 2012; Caporin et al., 2017).

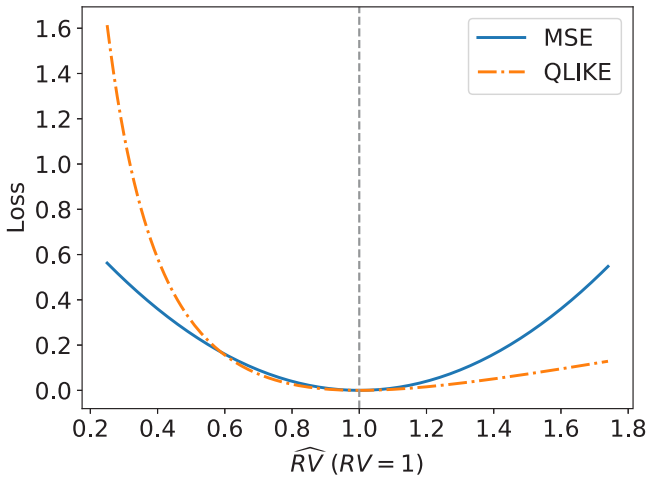


Fig. 4. Comparison of the MSE and QL loss functions.

evaluation methods. However, it is important to distinguish between the concept of forecast loss (FL) and the estimation criterion (EC), as they serve distinct purposes. FL assesses the performance of RV forecasts during out-of-sample testing, while the EC is utilized for model estimation within the in-sample period (Cipollini et al., 2020).

In order to determine the significance of the performance improvement compared to the baseline models, we employ two commonly used statistical tests found in the literature. As suggested by Patton and Sheppard (2009), QL demonstrates greater statistical power than MSE in the Diebold–Mariano (DM) test. Consequently, our focus in the analysis of the out-of-sample results is primarily on QL.

- The model confidence set (MCS) was proposed by Hansen et al. (2011) to identify a subset of models with significantly superior performance from model candidates at a given level of confidence. The MCS procedure renders it possible to make statements about the statistical significance from multiple pairwise comparisons. For additional details, we refer to the studies of Hansen et al. (2003, 2011).
- The Diebold–Mariano (DM) test was proposed by Diebold and Mariano (1995) to examine whether there are significant differences between two time-series forecasts. The DM test was further modified by Harvey et al. (1997) to account for serial dependence in forecasts. In addition to comparing errors for each individual stock, we also follow Gu et al. (2020) to compare the cross-sectional average of prediction errors from two models. Further details on the DM test are available in Diebold and Mariano (1995).

4. Empirical analysis

In this section, we first introduce the data and provide details regarding the implementation. Subsequently, we present the main findings and conduct a stratified analysis to evaluate the performance across different market regimes.

4.1. Setup

The intraday data of Dow Jones Industrial Average (DJIA) components are obtained from the LOBSTER database.[17] The time period under consideration is from July 1, 2007 to Jun 30, 2021.[18] Following Bollerslev et al. (2016), we include only those stocks among the DJIA components that traded continuously throughout the entire period. As a result, 27 stocks are included in the final sample. Their ticker symbols are listed in Appendix A, where we also present summary statistics for the volatility estimates. Additionally, for robustness checks, we consider a larger universe of S&P 100 components. Further details regarding this analysis can be found in Section 6.2.

Our out-of-sample forecast comparisons are based on the RV forecasts for the set of models introduced in Sections 2 and 3. All models are recalibrated every month based on a rolling sample window of the past 1000 days, following Bollerslev et al. (2016), Bollerslev, Patton, and Quaedvlieg (2018), Symitsi et al. (2018) and Pascalau and Poirier (2021). Specifically, we use 36-month data for model training, and the recent 12-month data as the validation set to tune the hyperparameters and prevent overfitting.[19] Finally, testing data are the samples in the following month; they are out-of-sample in order to provide objective assessments of the model performance. To this end, in aggregate, we obtain a 10-year out-of-sample period, that is, from July 1, 2011 to June 30, 2021.

The parameters in HAR_M and GHAR_M are estimated by OLS using both the training and validation data, as there is no requirement for hyperparameter tuning. To estimate the parameters in the proposed GNNHARs, we

[17] https://lobsterdata.com/

[18] The LOBSTER database contains data from June 27, 2007 up to the day before yesterday.

[19] To examine the impact of the validation dataset, we perform a robustness check for GNNHAR models in Section 6.1, and we conclude that the other choice of validation data does not significantly alter our findings.

adopt the Adam optimizer (Kingma & Ba, 2014).[20] When QL is chosen as the EC, there are no available estimators in closed form. Therefore, we also employ Adam to optimize HAR_Q and $GHAR_Q$ using both the training and validation data. Given the stochastic nature of the optimizer,[21] we employ an ensemble approach to enhance the robustness of GNNHAR models and QL-trained linear models (see Gu et al., 2020; Zhang et al., 2024). We train multiple models with random initialization and obtain final predictions by averaging the outputs of all networks. For further details on the hyperparameter choices in GNNHAR, refer to Appendix D.

One-day forecasting is not the only time horizon of interest to practitioners. Following the convention established in the literature (Symitsi et al., 2018; Zhang et al., 2022), we also examine whether the proposed methods can be applied to various forecasting horizons, e.g. one week or one month. The weekly and monthly target volatilities are defined as $\boldsymbol{v}_{t:t+h} = \sum_{k=0}^{h} \boldsymbol{v}_{t+k}$, where $h = 4$ and $h = 21$, respectively.

4.2. Main results

We begin our empirical analysis by comparing the out-of-sample performance of the competing models under consideration. Table 1 presents the ratio of forecast losses for each model relative to the HAR_M model (i.e. HAR estimated by OLS).

Table 1 first highlights the consistent improvement of the GHAR model over the standard HAR model in terms of forecast loss (FL), implying the importance of graph information. Furthermore, the first two columns of Table 1, which represent the results for the one-day horizon, demonstrate that our proposed GNNHAR model with a single hidden layer ($GNNHAR1L_M$) further improves the performance of the linear model $GHAR_M$. This finding underscores the significance of incorporating nonlinearity when modeling the spillover effect. However, it is worth noting that the performance starts to decline when additional GNN layers are added, particularly with three layers.

When considering models trained with QL, the results for the one-day horizon reveal that HAR_Q achieves better forecasts than its counterpart HAR_M. $GNNHAR1L_Q$ further improves the predictive accuracy of $GNNHAR1L_M$ and yields the best (resp. second-best) out-of-sample performance in terms of MSE (resp. QL). Specifically, at the daily forecast horizon, $GNNHAR1L_Q$ has about 13% (resp. 4%) lower average forecast error in MSE (resp. QL) compared to the standard HAR_M model. In addition, the MCS test indicates that both $GNNHAR1L_Q$ and $GNNHAR2L_Q$ are included in the subset of best models, based on the QL forecast loss. Interestingly, $GNNHAR3L_Q$ delivers worse out-of-sample performance than GNNs with one or two layers, yet still outperforms its counterpart trained with MSE. These findings suggest that QL might serve as a more effective in-sample estimation criterion than MSE. In the subsequent sections, we provide further analysis to delve into these results.

Table 1
Out-of-sample forecast losses.

	One day		One week		One month	
	MSE	QL	MSE	QL	MSE	QL
HAR_M	1.000	1.000	1.000	1.000	1.000	1.000
$GHAR_M$	0.927	0.983	0.904	0.987	0.975*	1.036
$GNNHAR1L_M$	0.907	0.979	0.940	0.943	1.021	0.968
$GNNHAR2L_M$	0.967	0.977	1.034	0.953	1.134	1.032
$GNNHAR3L_M$	1.210	0.982	1.014	0.961	1.046	0.958
HAR_Q	0.927	0.981	0.939	0.945	1.069	0.986
$GHAR_Q$	0.886	0.983	0.842*	0.936	1.151	0.954*
$GNNHAR1L_Q$	0.867*	0.961*	0.855	0.913*	1.179	0.965
$GNNHAR2L_Q$	0.879	0.959*	0.873	0.920	1.736	0.947*
$GNNHAR3L_Q$	0.894	0.963	1.185	0.942	1.502	0.971

Note: The table reports the ratios of forecast losses of various models compared to the standard HAR_M model over the one-day, one-week, and one-month horizons. For each horizon, the model with the best out-of-sample performance in MSE (QL) is highlighted in red (blue).
* Asterisk indicates models that yield as accurate forecasts as the best model at the 5% significance level based on the MCS test.

The results for weekly and monthly horizons presented in Table 1 demonstrate that models incorporating graph information (including GHAR and various GNNHAR models) exhibit significantly superior forecast accuracy compared to the HAR model over longer horizons, up to one week. Specifically, when examining the QL loss for the one-week forecast horizon, we observe that $GNNHAR1L_Q$ achieves the best out-of-sample performance. However, as the prediction horizon extends, the ratios approach or even exceed one, particularly for MSE. This suggests that longer-term forecasting becomes less sensitive to graph information. Additionally, we notice that the discrepancy between the ratios based on MSE and QL becomes more pronounced over longer horizons. One possible explanation is that the QL loss is generally less affected by extreme observations in the testing samples (see Patton, 2011). This is particularly relevant considering that such extreme observations may occur more frequently over longer horizons.

4.3. Market regimes

To assess the stability of performance across different market regimes, we perform a stratified out-of-sample analysis on two sub-samples: relatively calm periods when the RV of the S&P 500 ETF index is below the 90% quantile of its entire sample distribution, and the turbulent periods when the RV is above its 90% quantile (see Pascalau & Poirier, 2021; Zhang et al., 2022).

The results presented in Table 2 demonstrate that the enhancements achieved through the introduction of nonlinearity and the selection of QL as the EC are generally consistent across different market regimes. Specifically, when considering calm days and the daily forecast horizon, the models $GNNHAR1L_M$ and $GNNHAR2L_M$ appear to be the most effective based on the MSE loss. On the other hand, when evaluating accuracy in terms of QL,

[20] Adam is a popular stochastic optimization algorithm for deep learning models and is very efficient at finding the local minimum, especially with those non-convex and less smooth loss functions.

[21] The stochastic optimization algorithms might end up with different local minima with different initial values.

the models GNNHAR1L$_Q$ and GNNHAR2L$_Q$ provide the most precise forecasts. This outcome is expected since the volatility process tends to be more stable during calm periods. Consequently, if the forecast user has a specific preference for a particular loss function, it would be advisable to optimize the model parameters accordingly. In other words, for stationary time series, the alignment of the training loss (i.e. EC) and the testing loss (i.e. FL) may produce improved forecasts.

Nevertheless, when examining turbulent days and the daily forecast horizon, models trained with QL exhibit greater percentage improvements compared to those trained with MSE across both losses. For instance, the average forecast MSE (QL) loss of GNNHAR1L$_Q$ is approximately 13% (2%) lower than GNNHAR1L$_M$. This suggests that models trained with QL may possess unique characteristics distinct from their MSE-trained counterparts during turbulent periods. This intriguing discovery is explored and analyzed in the subsequent section.

In addition, when considering longer forecast horizons and periods of calmness, GNNHAR1L$_M$ produces significantly more accurate out-of-sample forecasts relative to other models in terms of MSE. Regarding the QL accuracy, GNNHAR1L$_Q$ outperforms other models for the weekly horizon, while GNNHAR2L$_M$ emerges as the top-performing model for the monthly horizon. When transitioning to the volatile periods, we continue to observe the superiority of QL-trained models (especially GHAR$_Q$) over MSE-trained models, with the exception being the monthly forecast horizon and considering MSE as the FL.

Table 2
Stratified out-of-sample forecast losses.

	One day		One week		One month	
	MSE	QL	MSE	QL	MSE	QL
	Panel A: Bottom 90%					
HAR$_M$	1.000	1.000	1.000	1.000	1.000	1.000
GHAR$_M$	0.961	0.998	0.949	1.001	0.967	1.027
GNNHAR1L$_M$	0.943*	0.998	0.883*	0.960*	0.923*	0.924*
GNNHAR2L$_M$	0.944*	0.990	0.901	0.954*	0.946*	0.921*
GNNHAR3L$_M$	0.957	0.987	0.911	0.965	0.937*	0.930*
HAR$_Q$	1.010	0.984	1.005	0.955*	1.159	0.942*
GHAR$_Q$	0.989	1.007	1.076	1.001	1.257	1.084
GNNHAR1L$_Q$	0.967	0.978*	0.944	0.943*	1.478	0.977
GNNHAR2L$_Q$	0.976	0.979*	0.985	0.947*	1.433	0.973
GNNHAR3L$_Q$	0.970	0.980*	1.062	0.957	1.662	0.969
	Panel B: Top 10%					
HAR$_M$	1.000	1.000	1.000	1.000	1.000	1.000
GHAR$_M$	0.916	0.910	0.897	0.959	0.976*	1.043
GNNHAR1L$_M$	0.895	0.903	0.949	0.908	1.033	1.007
GNNHAR2L$_M$	1.102	0.915	1.056	0.951	1.157	1.131
GNNHAR3L$_M$	1.293	0.958	1.030	0.952	1.059	0.982
HAR$_Q$	0.900	0.965	0.928	0.925	1.059	1.024
GHAR$_Q$	0.852	0.867*	0.804*	0.799*	1.149	0.841*
GNNHAR1L$_Q$	0.834*	0.879	0.841	0.848	1.143	0.955
GNNHAR2L$_Q$	0.848	0.862*	0.924	0.861	1.773	0.886
GNNHAR3L$_Q$	0.868	0.882	1.205	0.909	1.483	0.973

Note: The table reports stratified losses during trading days with the bottom 90% (Panel A) and the top 10% (Panel B) RV of the S&P 500 ETF index over the one-day, one-week, and one-month horizons. For each horizon, the model with the best out-of-sample performance in MSE (QL) is highlighted in red (blue).

* Asterisk indicates models that yield as accurate forecasts as the best model at the 5% significance level based on the MCS test.

5. Discussion

The objective of this section is to examine the reasons behind the superior performance of our proposed GNNHAR models trained with QL. Our analysis begins by investigating the impact of the choice of EC on the predictive accuracy of the models. We then delve into exploring the influence of model nonlinearity, followed by an examination of the predictive information obtained from multi-hop neighbors.

5.1. Impact of evaluation criterion

As mentioned above, QL deals with over- and under-predictions differently, which may account for the overall better performance of QL-trained models compared to MSE-trained models. In light of this observation, we examine the forecast errors ($\widehat{RV}^{(F)}_{i,t} - RV_{i,t}$) and forecast ratios ($\widehat{RV}^{(F)}_{i,t} / RV_{i,t}$) over the entire testing period and various sub-periods.[22]

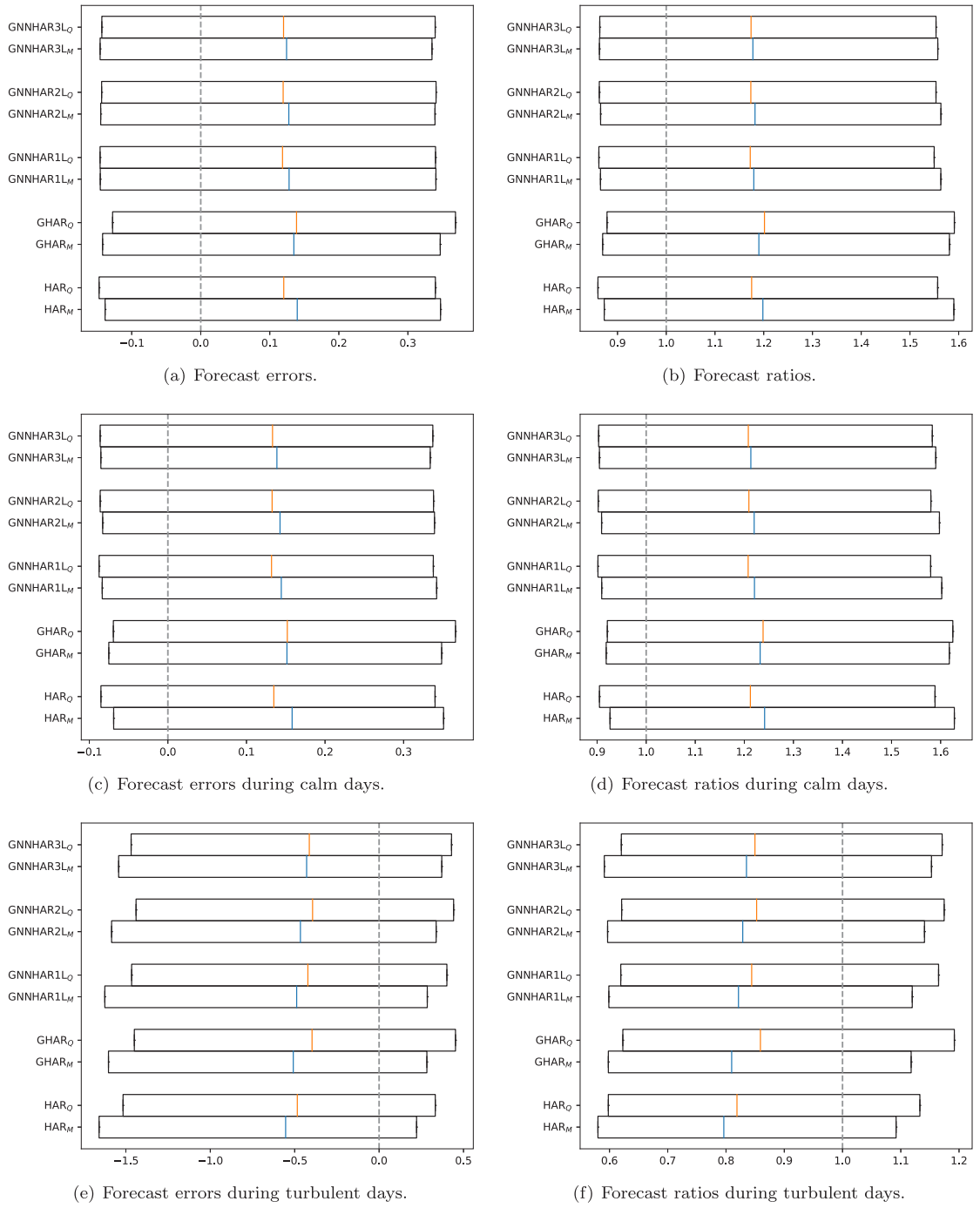
Fig. 5 presents boxplots for forecast errors and ratios of various models. From subplots (a) and (b), we observe that in general, all models tend to exhibit a bias towards over-predictions (i.e. positive errors or ratios greater than one) rather than under-predictions, aligning with the findings of Clements and Preve (2021). Subplots (c) and (d) further unveil that this over-prediction tendency is primarily observed during calm periods. Conversely, subplots (e) and (f) indicate that these models are more inclined to under-predict volatilities during turbulent periods. This observation is not surprising, as the models do not explicitly incorporate any exogenous variables to aid in detecting changes in market conditions.

Furthermore, subplots (a) and (b) demonstrate that the bulk of the forecast errors (resp. ratios) of QL-trained models are generally closer to zero (resp. one) compared to MSE-trained models. Specifically, subplots (c) and (d) reveal that QL-trained models exhibit a reduced tendency to over-predict during calm periods, while subplots (e) and (f) suggest that they are less prone to excessive under-prediction during turbulent periods, when compared to the MSE-trained models.

5.2. Impact of nonlinearity

To examine the necessity of nonlinear relations, we provide the following analysis to shed light on the competitive performance of these models, particularly during volatile periods. Inspired by Chinco et al. (2019), we introduce, for each day t, the following metric to evaluate the fraction of variance of model F which is unexplained

[22] It is worth noting that the MSE loss is solely dependent on the forecast error, while QL exclusively relies on the forecast ratio, as corroborated by Patton (2011).



(a) Forecast errors.

(b) Forecast ratios.

(c) Forecast errors during calm days.

(d) Forecast ratios during calm days.

(e) Forecast errors during turbulent days.

(f) Forecast ratios during turbulent days.

Fig. 5. Grouped boxplots for models trained with MSE or QL.
Note: This figure presents boxplots illustrating three summary statistics: the median, and the Q1 and Q3 quantiles. Each group consists of two sets of boxplots, with the top (resp. bottom) set representing models utilizing QL (resp. MSE) as EC. (a)–(b) Forecast errors or ratios over the entire testing period. (c)–(d) Forecast errors or ratios over calm periods. (e)–(f) Forecast errors or ratios over turbulent periods.

Table 3
FVUs compared to HAR_M.

	One day		One week		One month	
	Calm	Turb	Calm	Turb	Calm	Turb
HAR_M	0.000	0.000	0.000	0.000	0.000	0.000
$GHAR_M$	0.044	0.061	0.054	0.099	0.066	0.092
$GNNHAR1L_M$	0.077	0.165	0.117	0.244	0.178	0.300
$GNNHAR2L_M$	0.080	0.205	0.114	0.304	0.207	0.441
$GNNHAR3L_M$	0.079	0.300	0.130	0.246	0.218	0.272
HAR_Q	0.033	0.056	0.068	0.139	0.184	0.263
$GHAR_Q$	0.077	0.128	0.108	0.216	0.228	0.779
$GNNHAR1L_Q$	0.060	0.134	0.102	0.244	0.216	0.886
$GNNHAR2L_Q$	0.060	0.184	0.118	0.379	0.283	1.391
$GNNHAR3L_Q$	0.070	0.212	0.163	0.764	0.292	1.236

Note: The table reports the fractions of variance unexplained (FVUs) of multiple models compared by the baseline HAR, across different market regimes.

(FVU) by the standard HAR_M model[23]:

$$\mathrm{FVU}_t = \frac{\sum_{i=1}^{N}\left(\widehat{RV}_{i,t}^{(F)} - \widehat{RV}_{i,t}^{(\mathrm{HAR}_M)}\right)^2}{\sum_{i=1}^{N}\left(\widehat{RV}_{i,t}^{(F)} - \overline{RV}_{t}^{(F)}\right)^2}, \tag{11}$$

where $\overline{RV}_t^{(F)}$ is the average forecast RV of model F across stocks on day t. At one extreme, $\mathrm{FVU}_t = 0$ means that the HAR_M's RV forecasts explain all of the variation in the predicted RVs provided by F, whereas, at the other extreme, $\mathrm{FVU}_t = 1$ denotes that HAR_M explains none of this variation.

Table 3 displays the FVUs of each model in comparison to HAR_M. It is worth noting that nonlinear models, particularly those with multiple hidden layers, exhibit higher FVU values, as anticipated. In addition, the results for one-week and one-month horizons in Table 3 suggest that the nonlinearity in volatility models seems to strengthen as the forecasting horizons increase. It is important to mention that the distinction between GHAR and GNNHAR1L lies in the presence of an additional hidden layer with a nonlinear activation function in GNNHAR1L. Consequently, the extra FVUs observed in GNNHAR1L can be considered as a measure of the degree of nonlinearity.

By comparing the first column and second column in Table 3, we observe higher FVU scores during turbulent days, regardless of the choice of EC. This suggests that nonlinear spillover effects are most likely to exist in turbulent periods, rather than in calm periods. In light of the results in Table 2, it can be inferred that a suitable level of model nonlinearity, such as that exhibited by GNNHAR1L, leads to improved predictive power during turbulent days. However, we find that overly complex models, such as GNNHAR3L, are unable to outperform the linear baseline. As a result, GNNHAR1L shows significant promise as a model for capturing nonlinearity while avoiding the overfitting problem.

[23] In fact, $\mathrm{FVU}_t = 1 - R^2(\widehat{RV}_{i,t}^{(F)}, \widehat{RV}_{i,t}^{(\mathrm{HAR}_M)})$, where R^2 is the coefficient of determination between the predicted RVs from the target model and the baseline model.

5.3. Impact of multi-hop neighbors

We utilize the DM test to evaluate the statistical significance of two-hop neighbors by comparing the performance of GNNHAR2L and GNNHAR1L. Here, a positive (resp. negative) DM test value indicates the superiority of the GNNHAR1L (resp. GNNHAR2L) model. A p-value less than a given significance level a rejects the null hypothesis that GNNHAR2L and GNNHAR1L have the same forecasting power at the $1 - a$ confidence level.[24]

Fig. 6 illustrates the main results from the above hypothesis test. In terms of individual stocks, $GNNHAR2L_M$ is only superior to $GNNHAR1L_M$ in forecasting AXP's volatilities, at the 5% confidence level. When considering the cross-sectional performance, the p-value is around 75%, from which we cannot reject the null hypothesis. This suggests that once the impact from itself and its one-hop neighbors has been taken into account, two-hop neighbors are not deemed necessary. The comparison between $GNNHAR2L_Q$ and $GNNHAR1L_Q$ indeed supports these findings.

GNNs are known to suffer from the problem of over-smoothing, which is defined as the high similarity of node representations obtained at the output layer of GNNs; see Li et al. (2018). Such high similarity is often observed when stacking with multiple GNN layers that are more than necessary. With K layers, every node receives information from its K-hop neighbors.[25] When K is large, node representations obtained from GNN information propagation become indistinguishable and weaken the forecasting accuracy.

Following the convention in the GNN literature (e.g. Chen et al., 2020), we use the mean average distance (MAD) to measure the similarity of node representations and identify whether there is any sign of over-smoothing in our GNNHAR models. The MAD takes as input the node representations $\boldsymbol{H} \in \mathbb{R}^{N\times D}$ obtained at the final layer of the GNN, that is $\boldsymbol{H} = \mathrm{GNN}(\boldsymbol{V}_{:t-1}, \boldsymbol{A})$ in (6). It is defined as follows[26]:

$$\mathrm{MAD} = \frac{\sum_{i=1}^{N}\bar{d}_i}{\sum_{i=1}^{N}\mathbb{1}_{\bar{d}_i>0}}, \quad \text{where } \bar{d}_i = \frac{\sum_{j=1}^{N}\bar{\boldsymbol{D}}_{ij}}{\sum_{j=1}^{N}\mathbb{1}_{\bar{\boldsymbol{D}}_{ij}>0}}. \tag{12}$$

Here, $\bar{\boldsymbol{D}}$ is the masked cosine distance matrix, i.e. $\bar{\boldsymbol{D}} = \boldsymbol{D} \circ \boldsymbol{A}$, where $\circ$ denotes the Hadamard product (element-wise multiplication), and $\boldsymbol{D}_{ij} = 1 - \frac{\boldsymbol{H}[i,:]\cdot\boldsymbol{H}[j,:]}{\|\boldsymbol{H}[i,:]\|\|\boldsymbol{H}[j,:]\|}$. In the above definition, $\bar{d}_i$ is the average distance between the representations of node i and its connected nodes. Overall, MAD represents an average level of how a node representation is similar to the representations of its connected neighbors in a graph.

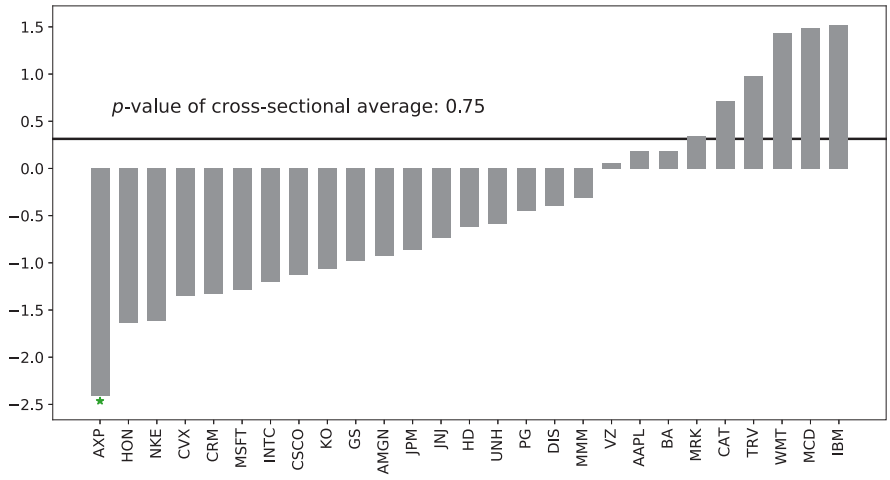
In Fig. 7, three boxes represent GNNHAR models with one, two, and three GNN layers trained with MSE.[27] Each

[24] We also conducted the same test to compare linear multi-hop graph models, i.e. GHAR and GHAR2Hop (see Appendix E) and the conclusions were similar.

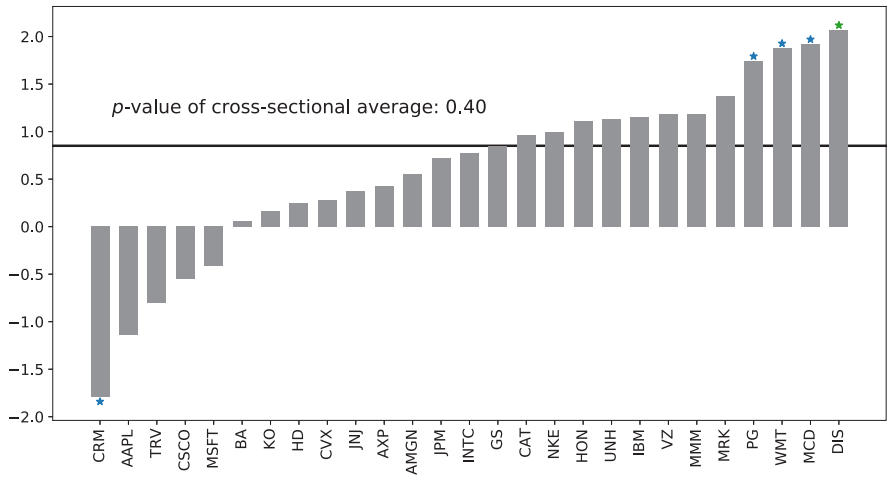
[25] This is also known as the receptive field of a GNN. Details are provided in Section 2.

[26] $\boldsymbol{H}$ is the (unweighted) average of the hidden representations obtained from GNNHARs in our ensemble set.

[27] Similar results (unreported) were observed for GNNHARs trained with QL.



(a) $\mathrm{GNNHAR2L}_M$ vs $\mathrm{GNNHAR1L}_M$



(b) $\mathrm{GNNHAR2L}_Q$ vs $\mathrm{GNNHAR1L}_Q$

Fig. 6. DM test between GNNHAR2L and GNNHAR1L.
Note: A positive (negative) number indicates superiority for the GNNHAR1L (GNNHAR2L) model. The *y*-axis represents the DM test values based on a QL between GNNHAR2L and GNNHAR1L, while the *x*-axis lists the stock symbols. Stars indicate the *p*-value, with orange, green, and blue representing significance at the 1%, 5%, and 10% levels, respectively. The horizon line represents the cross-sectional DM test value and its corresponding *p*-value. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

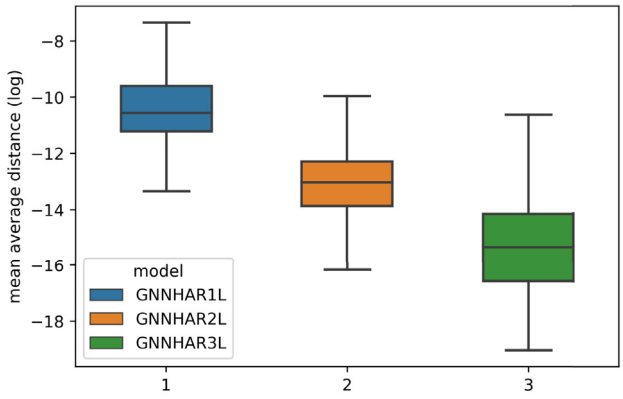


Fig. 7. Smoothness of GNNHARs.
Note: A small mean average distance (MAD) value indicates high similarity between node representations at the output layer of the GNN.

Table 4
Out-of-sample forecast losses under a smaller validation dataset.

	One day		One week		One month	
	MSE	QL	MSE	QL	MSE	QL
HAR_M	1.000	1.000	1.000	1.000	1.000	1.000
$GHAR_M$	0.927	0.983	0.904	0.987	0.975*	1.031
$GNNHAR1L_M$	0.942	0.978	0.931	0.945	1.008	0.975
$GNNHAR2L_M$	0.984	0.984	1.005	0.956	1.138	1.033
$GNNHAR3L_M$	1.078	1.002	1.035	0.954	1.068	0.958
HAR_Q	0.936	0.986	0.945	0.944	1.218	0.959
$GHAR_Q$	0.942	0.982	0.993	0.945	1.174	0.954
$GNNHAR1L_Q$	0.889*	0.967*	0.875*	0.912	1.226	0.961
$GNNHAR2L_Q$	0.896	0.968*	0.861*	0.907*	1.510	0.925*
$GNNHAR3L_Q$	1.152	0.981	1.060	0.929	1.572	0.972

Note: The table reports the out-of-sample losses of various models using 47 months as training data and the most recent one month as validation data. For each horizon, the model with the best out-of-sample performance in MSE (QL) is highlighted in red (blue).
* Asterisk indicates models that yield as accurate forecasts as the best model at the 5% significance level based on the MCS test.

box corresponds to the MAD values on a logarithmic scale, calculated across all out-of-sample samples. As the number of GNN layers increases, there is a decrease in log MAD that corresponds to an increase in smoothness. The three-layer GNNHAR has the lowest MAD score, suggesting potential over-smoothing of node representations. Specifically, the rows of $\text{GNN}(\boldsymbol{V}_{:t-1}, \boldsymbol{A})$ from GNNHAR3L in (6) become too similar to provide any node-specific predictive information. This partially explains the inferior performance of GNNHAR3L, as shown in Table 1.

6. Robustness tests

After presenting the main empirical results and analyzing the model performance across different market periods, we shift our focus to evaluating the robustness of the proposed models by considering two aspects: (i) an alternative validation set size, and (ii) a larger universe.

6.1. Alternative validation set size

Our main analysis is based on rolling samples of four years, using the first approximately three years as training data, and the recent year as validation data. Using a smaller validation dataset, such as one month, does not significantly alter our findings, as shown in Table 4.

6.2. Larger universe

To further assess the robustness of our findings and ascertain that they are not specific to the stocks under current consideration, we repeat the out-of-sample analysis using a larger dataset, including the components of the S&P 100 index.[28] The experimental setups and the hyperparameter choices in GNNHAR remain the same as those described in Section 4.1. As illustrated in Table A.2, in the volatility spillover graphs for the S&P 100 index components, each node is connected to other nodes

[28] Details about the data are provided in Appendix A.

Table 5
Out-of-sample forecast losses on S&P 100.

	One day		One week		One month	
	MSE	QL	MSE	QL	MSE	QL
HAR_M	1.000	1.000	1.000	1.000	1.000	1.000
$GHAR1L_M$	0.948	0.988	0.909	0.994	0.972*	0.986
$GNNHAR1L_M$	0.963	0.986	0.951	0.944	1.027	1.092
$GNNHAR2L_M$	1.072	0.988	1.031	0.954	1.092	1.000
$GNNHAR3L_M$	1.061	0.986	1.029	0.959	0.992	0.967
$GNNHAR4L_M$	1.047	0.992	1.042	0.975	1.079	0.978
$GNNHAR5L_M$	1.090	0.997	1.057	0.986	1.109	1.038
HAR_Q	0.949	0.983	0.937	0.947	1.171	0.991
$GHAR_Q$	0.919	0.984	0.850*	0.922	1.154	0.939*
$GNNHAR1L_Q$	0.917*	0.969	0.858	0.916	1.231	1.017
$GNNHAR2L_Q$	0.915*	0.969	0.909	0.915*	1.206	0.941*
$GNNHAR3L_Q$	0.938	0.966*	1.178	0.968	1.523	0.946
$GNNHAR4L_Q$	0.985	0.970	1.165	0.972	1.563	0.971
$GNNHAR5L_Q$	0.951	0.968	1.193	0.975	1.741	0.989

Note: The table reports the ratios of forecast losses of various models compared to the standard HAR_M model over one-day, one-week, and one-month horizons. For each horizon, the model with the best out-of-sample performance in MSE (QL) is highlighted in red (blue).
* Asterisk indicates models that yield as accurate forecasts as the best model at the 5% significance level based on the MCS test.

within a maximum of five steps. Consequently, we extend our analysis to include four- and five-layer versions of the GNNHAR model.

The out-of-sample forecasting performance on the volatilities of S&P 100 components is presented in Table 5. Firstly, we observe that GHAR consistently enhances forecasting accuracy compared to the traditional HAR model. Additionally, the nonlinear variant, GNNHAR1L, further improves upon the performance of GHAR over the one-day horizon. Generally, as we increase the number of layers in the GNNHAR models, their forecasting performance tends to decline. Nevertheless, we still observe the benefits of training models with the QL loss function. In summary, the findings presented in Table 5 align closely with those observed for the DJIA 30, providing consistent results across both datasets.

7. Conclusion

In this article, we proposed a novel methodology, GNNHAR, for modeling and forecasting RV while taking into account volatility spillover effects in the U.S. equity market. Our analysis suggests that the information from the multi-hop neighbors in the financial graph does not offer a clear advantage in predicting the volatility of any target stock. However, nonlinear spillover effects help improve the forecasting accuracy of the RV. Moreover, we found that utilizing QL as the training loss function leads to more accurate volatility forecasts than using the conventional MSE. Additionally, QL-trained nonlinear models demonstrated greater resilience during turbulent periods compared to calmer market conditions, unlike standard linear models which struggle in such regimes. Our comprehensive evaluation tests in alternative settings confirmed the robustness and effectiveness of our proposed methodology.

Table A.1
Summary statistics of realized volatility.

Ticker	Mean	Std	Min	25%	50%	75%	Max	DJIA	S&P 100
AAPL	2.30	3.39	0.07	0.70	1.25	2.46	38.30	✓	✓
ABT	1.41	1.95	0.12	0.57	0.89	1.50	34.32		✓
ACN	1.72	2.79	0.14	0.58	0.92	1.76	54.88		✓
ADBE	2.53	3.34	0.16	0.93	1.54	2.76	45.55		✓
ADP	1.41	2.51	0.10	0.49	0.78	1.39	44.36		✓
AMGN	1.91	2.34	0.16	0.82	1.27	2.14	33.44	✓	✓
AMT	2.16	3.83	0.19	0.68	1.11	2.10	53.19		✓
AMZN	3.22	4.48	0.11	1.02	1.84	3.59	62.14		✓
AXP	3.19	6.32	0.12	0.64	1.15	2.67	91.45	✓	✓
BA	2.69	5.00	0.13	0.78	1.35	2.60	90.65	✓	✓
BAC	4.93	11.48	0.10	1.01	1.81	3.68	135.30		✓
BDX	1.37	1.84	0.13	0.54	0.86	1.48	28.52		✓
BMY	1.77	2.20	0.08	0.72	1.14	1.93	30.75		✓
BSX	3.15	4.39	0.20	1.14	1.92	3.35	55.28		✓
C	5.48	14.6	0.15	0.99	1.82	3.94	257.34		✓
CAT	2.79	4.00	0.15	0.94	1.58	2.89	45.26	✓	✓
CB	1.82	3.66	0.07	0.44	0.75	1.62	61.54		✓
CI	3.65	6.92	0.19	1.01	1.75	3.28	164.21		✓
CMCSA	2.35	3.57	0.13	0.78	1.29	2.47	43.26		✓
CME	3.07	5.49	0.18	0.84	1.38	2.72	68.79		✓
COP	3.12	5.18	0.16	0.98	1.71	3.26	75.84		✓
COST	1.44	2.11	0.0	0.51	0.79	1.44	26.30		✓
CRM	4.00	4.93	0.22	1.44	2.41	4.64	61.67	✓	✓
CSCO	1.98	2.92	0.14	0.70	1.13	2.09	43.74	✓	✓
CVS	1.99	3.15	0.13	0.70	1.17	2.03	53.28		✓
CVX	2.03	3.51	0.13	0.61	1.07	2.04	48.07	✓	✓
D	1.44	2.56	0.1	0.56	0.85	1.40	40.39		✓
DHR	1.6	2.41	0.14	0.54	0.95	1.67	29.78		✓
DIS	1.89	3.04	0.12	0.60	1.01	1.88	40.56	✓	✓
DUK	1.32	2.20	0.06	0.50	0.78	1.32	36.07		✓
FIS	1.89	3.48	0.15	0.59	0.97	1.74	62.40		✓
FISV	1.71	2.82	0.15	0.58	0.93	1.69	53.36		✓
GE	3.08	5.54	0.09	0.68	1.43	3.05	77.33		✓
GILD	2.36	2.67	0.23	1.03	1.55	2.64	33.62		✓
GOOG	1.94	2.72	0.11	0.64	1.08	2.07	30.36		✓
GS	3.24	6.27	0.19	0.92	1.49	2.81	112.41	✓	✓
HD	2.11	3.59	0.15	0.62	1.02	2.01	48.22	✓	✓
HON	1.85	3.25	0.1	0.52	0.97	1.84	49.64	✓	✓
IBM	1.38	2.33	0.11	0.47	0.75	1.34	30.22	✓	✓
INTC	2.29	3.12	0.14	0.86	1.39	2.44	42.90	✓	✓
INTU	2.00	2.81	0.15	0.75	1.22	2.15	38.91		✓
Ticker	Mean	Std	Min	25%	50%	75%	Max	DJIA	S&P 100
ISRG	3.19	4.31	0.22	1.10	1.81	3.38	46.66		✓
JNJ	0.92	1.56	0.06	0.35	0.54	0.90	24.74	✓	✓
JPM	3.46	7.04	0.15	0.74	1.36	2.82	108.17	✓	✓
KO	0.99	1.68	0.07	0.37	0.58	1.00	25.00	✓	✓
LLY	1.59	2.29	0.13	0.61	0.98	1.70	35.90		✓
LMT	1.64	2.59	0.12	0.56	0.94	1.64	35.79		✓
LOW	2.71	4.20	0.17	0.88	1.45	2.77	73.32		✓
MA	2.86	4.60	0.13	0.73	1.31	2.79	52.20		✓
MCD	1.17	2.15	0.08	0.39	0.61	1.13	37.57	✓	✓
MDT	1.5	2.19	0.13	0.59	0.93	1.57	36.66		✓
MMM	1.43	2.25	0.08	0.46	0.81	1.49	31.11	✓	✓
MO	1.41	2.25	0.06	0.52	0.84	1.43	39.67		✓
MRK	1.65	2.45	0.12	0.58	0.92	1.74	30.99	✓	✓
MS	5.74	14.30	0.20	1.25	2.18	4.33	286.91		✓
MSFT	1.82	2.51	0.11	0.67	1.09	1.92	30.64	✓	✓
NFLX	5.53	5.69	0.36	2.14	3.78	6.75	72.86		✓
NKE	2.03	3.00	0.14	0.74	1.15	2.02	47.87	✓	✓
NVDA	5.14	6.03	0.40	1.83	3.2	5.96	72.25		✓
ORCL	1.90	2.84	0.08	0.66	1.14	2.05	44.23		✓
PEP	1.02	1.78	0.06	0.37	0.58	1.01	28.18		✓
PFE	1.55	2.07	0.14	0.59	0.95	1.67	26.54		✓
PG	1.00	1.76	0.09	0.38	0.58	0.98	31.60	✓	✓
PNC	3.64	7.52	0.16	0.79	1.38	3.04	141.27		✓
QCOM	2.46	3.39	0.10	0.81	1.49	2.77	42.15		✓

(*continued on next page*)

Table A.1 (*continued*).

Ticker	Mean	Std	Min	25%	50%	75%	Max	DJIA	S&P 100
SBUX	2.45	3.90	0.18	0.71	1.24	2.48	63.45		✓
SO	1.19	1.98	0.12	0.47	0.72	1.22	36.40		✓
SYK	1.67	2.61	0.08	0.62	0.98	1.76	49.51		✓
T	1.49	2.55	0.08	0.47	0.76	1.39	32.03		✓
TGT	2.46	4.02	0.11	0.76	1.24	2.34	53.02		✓
TJX	2.33	3.34	0.16	0.76	1.24	2.53	55.49		✓
TMO	1.89	2.74	0.16	0.71	1.14	1.99	40.82		✓
TRV	2.04	4.09	0.11	0.49	0.81	1.76	57.95	✓	
TXN	2.33	3.02	0.16	0.84	1.41	2.57	48.68		✓
UNH	2.70	4.34	0.16	0.78	1.35	2.57	52.54	✓	✓
UNP	2.53	3.94	0.14	0.83	1.39	2.52	45.94		✓
UPS	1.58	2.35	0.10	0.51	0.88	1.72	31.67		✓
USB	3.20	6.88	0.13	0.62	1.16	2.64	95.38		✓
VZ	1.40	2.36	0.10	0.50	0.77	1.33	34.19	✓	✓
WFC	4.05	8.89	0.11	0.73	1.39	3.24	106.81		✓
WMT	1.18	1.76	0.11	0.45	0.67	1.18	27.18	✓	✓

Note: The table reports summary statistics for the daily realized volatility of stocks in the DJIA 30 or S&P 100. The statistics are averaged across each trading day.

An intriguing avenue for future exploration involves expanding the predictor set to incorporate additional information sources, such as limit order books, options, and news (Li & Tang, 2021). Another interesting direction to explore is the robustness of the proposed methods when applied to different approaches to constructing financial graphs, such as those based on supply chains (Herskovic et al., 2020) and analyst co-coverage (Ali & Hirshleifer, 2020). It would be valuable to investigate whether these graphs provide unique information content and have the potential to enhance performance.

CRediT authorship contribution statement

Chao Zhang: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Xingyue Pu:** Writing – review & editing, Visualization, Validation, Software, Investigation, Formal analysis, Conceptualization. **Mihai Cucuringu:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization. **Xiaowen Dong:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data and code availability

The data used in this paper is sourced from LOBSTER, which is subject to licensing restrictions and must be purchased by users, as redistribution is not permitted. The source code is available at: github.com/chaozhang-ox/GNNHAR.

Table A.2
Frequency (in percentage) of the shortest path distance.

SPD	1	2	3	4	5
DJIA	57.7	41.8	0.5	0.0	0.0
S&P 100	24.3	61.2	12.0	2.2	0.3

Note: For example, in the case of the S&P 100, 12% of pairs of nodes have their shortest path distance of size three.

Acknowledgments

This work is supported by the Guangzhou-HKUST(GZ) Joint Funding Program (No. 2024A03J0630). The authors also thank the Oxford Suzhou Centre for Advanced Research for providing the computational facilities.

Appendix A. Data statistics

See Tables A.1 and A.2.

Appendix B. QL-trained models and MEM

Let y_t be a non-negative random variable, such as RV_t. y_t follows an MEM model if it can be expressed as

$$y_t = \nu_t z_t, \quad \nu_t = g\left(\theta; \mathcal{F}_{t-1}\right), \quad z_t \overset{\text{i.i.d.}}{\sim} D^+\left(1, \sigma^2\right). \quad (13)$$

Here, the term ν_t represents the non-negative expectation of y_t conditional on the information set $\mathcal{F}_{t-1}$ available at time $t-1$, and ν_t is determined by a function g with parameters θ. z_t is a conditionally unpredictable homoskedastic component, with non-negative support and unit expected value. The standard MEM aligns with the autoregressive structure of the well-known GARCH(1,1) for ν_t. In this paper, we utilize GNNs as g to model ν_t.

Supposing z_t is gamma-distributed[29] with scale 1 and shape 1, the density of y_t is

$$f_y(y_t) \propto \frac{1}{\nu_t} e^{-\frac{y_t}{\nu_t}}.$$

[29] In the univariate case, there are other distributions satisfying these characteristics, such as log-normal, beta, etc.

Subsequently, the negative log likelihood, after omitting constants, can be expressed as follows[30]:

$$L(\theta) = \sum_{t=1}^{T} \left[\log(\nu_t) + \frac{y_t}{\nu_t} \right]. \tag{14}$$

This is equivalent to (10) if we substitute ν_t with $\widehat{RV}_t^{(F)}$ and y_t with RV_t, up to a constant factor.

Appendix C. Training HAR via QL

Denote $\boldsymbol{\beta} = (\alpha, \beta_d, \beta_w, \beta_m)' \in \mathbb{R}^4$, $\boldsymbol{x}_t = (1, RV_{t-1}, RV_{t-5:t-2}, RV_{t-22:t-6})' \in \mathbb{R}^4$, and $\boldsymbol{X} = (\boldsymbol{x}_{23}, \ldots, \boldsymbol{x}_T)' \in \mathbb{R}^{(T-22)\times 4}$. The QL loss of the HAR model for a single time series is

$$\mathcal{L}_Q = \sum_t \left[\frac{RV_t}{\boldsymbol{\beta}'\boldsymbol{x}_t} - \log \frac{RV_t}{\boldsymbol{\beta}'\boldsymbol{x}_t} - 1 \right]. \tag{15}$$

Then score function is given by

$$\begin{aligned} \frac{\partial \mathcal{L}_Q}{\partial \boldsymbol{\beta}} &= \sum_t \frac{-RV_t}{(\boldsymbol{\beta}'\boldsymbol{x}_t)^2}\boldsymbol{x}_t + \frac{1}{\boldsymbol{\beta}'\boldsymbol{x}_t}\boldsymbol{x}_t \\ &= \sum_t \frac{\boldsymbol{\beta}'\boldsymbol{x}_t - RV_t}{(\boldsymbol{\beta}'\boldsymbol{x}_t)^2}\boldsymbol{x}_t \\ &= \sum_t w_{\boldsymbol{\beta},t}(\boldsymbol{\beta}'\boldsymbol{x}_t - RV_t)\boldsymbol{x}_t \\ &= \boldsymbol{X}'\boldsymbol{W}_{\boldsymbol{\beta}}(\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Y}) \end{aligned} \tag{16}$$

where $w_{\boldsymbol{\beta},t} = \frac{1}{(\boldsymbol{\beta}'\boldsymbol{x}_t)^2}$ and $\boldsymbol{W}_{\boldsymbol{\beta}} = \operatorname{diag}\{\ldots w_{\boldsymbol{\beta},t} \ldots\}$. This leads to the first-order condition $\boldsymbol{X}'\boldsymbol{W}_{\boldsymbol{\beta}}(\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Y}) = 0$.[31] The optimal solution $\boldsymbol{\beta}$ appears in the weights $\boldsymbol{W}_{\boldsymbol{\beta}}$. Iteratively reweighted least squares (IRLS) is therefore recommended:

1. Select initial estimates $\boldsymbol{\beta}_0$, such as the OLS.
2. At each iteration k, calculate the predictions $\widehat{RV}_{k-1,t} = \boldsymbol{\beta}_{k-1}'\boldsymbol{x}_t$ from the previous iteration, and the associated weights $w_{k-1,t} = 1/\widehat{RV}_{k-1,t}^2$ and $\boldsymbol{W}_{k-1} = \operatorname{diag}\{\ldots w_{k-1,t} \ldots\}$.
3. Solve for new WLS estimates
$$\boldsymbol{\beta}_k = \left[\boldsymbol{X}'\boldsymbol{W}_{k-1}\boldsymbol{X}\right]^{-1}\boldsymbol{X}'\boldsymbol{W}_{k-1}\boldsymbol{Y}. \tag{17}$$
4. Steps 2 and 3 are repeated until the estimated coefficients converge.

To gain further insights into the impact of the EC, we present the trajectories of β_d in the HAR models estimated using MSE or QL in Fig. C.1. As anticipated, there are substantial temporal variations in the rolling estimates of both models. In general, the estimates of β_d in HAR_Q exhibit greater variability compared to those in HAR_M, which can be attributed to the stochastic nature of the optimization algorithm employed in HAR_Q. However, the estimates of β_d in HAR_M reveal two prominent changes occurring during December 2015 to February 2016 and March 2020 to April 2020, albeit in different directions.[32] On the other hand, the β_d in HAR_Q exhibits an increasing trend during turbulent periods. This suggests that QL-trained models have the ability to swiftly adapt to market changes and assign greater importance to observations associated with recent significant events. Future studies exploring the relationship between different estimators of HAR are therefore recommended.

Appendix D. Hyperparameter tuning

Following the convention of stochastic optimization (Kingma & Ba, 2014), we set the batch size to 32.[33] The learning rate for Adam is set to 10^{-3}. We stop the training procedure early if there is a sign of overfitting, that is, if the training loss keeps dropping but validation loss increases beyond a tolerance level.

To a large extent, the dimension of hidden representations or the number of hidden neurons in the lth layer, i.e. $D^{(l)}$ in (5), reflects the complexity of our models. Inadequate dimensions may lack the capability to effectively capture the underlying data structure, while excessively large dimensions could lead to overfitting and poor generalization performance. To mitigate this issue, we use a grid search over $D^{(l)} \in \{3, 6, 9, 16, 32\}$ on validation datasets. Fig. D.1 shows that a hidden dimension of nine in a one-layer GNNHAR model leads to the smallest MSE and QL on the validation data. The same conclusion holds true for the QL-trained models as well. When multiple GNN layers are utilized, we maintain the same $D^{(l)}$ value as determined in the one-layer model.

Appendix E. GHAR with multi-hop (GHAR2Hop)

It is important to highlight that HAR can be interpreted as a model that only considers the zero-hop neighbors, i.e. the target node itself, while the GHAR takes into account both the zero-hop and one-hop neighbors. In order to explore the potential benefits of multi-hop neighbors for enhancing volatility forecasting, we delve into the investigation of whether they provide additional predictive power. To address this novel question, we consider the following model:

$$\begin{aligned} \textbf{GHAR2Hop}(\boldsymbol{A}): \quad \boldsymbol{RV}_t &= \boldsymbol{\alpha} + \boldsymbol{V}_{:t-1}\boldsymbol{\beta} + \boldsymbol{W}\boldsymbol{V}_{:t-1}\boldsymbol{\gamma} \\ &\quad + \textsc{Hop2}(\boldsymbol{A})\boldsymbol{V}_{:t-1}\boldsymbol{\delta} + \boldsymbol{u}_t, \end{aligned} \tag{18}$$

where $\textsc{Hop2}(\boldsymbol{A})$ maps the raw adjacent matrix (for one-hop neighbors) to the adjacent matrix of two-hop neighbors. Specifically, $\textsc{Hop2}(\boldsymbol{A}) = \text{XOR}(\boldsymbol{A}^2 \wedge (\neg\boldsymbol{A}), \boldsymbol{I}_N)$. $\boldsymbol{A}^2[i,j]$ has a non-zero if it is possible to go from node i to node j in two or fewer steps, $\neg\boldsymbol{A}$ excludes the one-hop neighbors, and XOR confirms the diagonal of the two-hop adjacent matrix to be zero. For a visual representation

[30] Bauwens et al. (2012) documented that the solution by maximum likelihood does not depend on the dispersion parameter a in Gamma$(a, 1/a)$.

[31] Note that the OLS estimator (i.e. trained with MSE) satisfies $\boldsymbol{X}'(\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Y}) = 0$.

[32] These two periods correspond to significant market changes, namely the Chinese stock market turbulence and the Covid-19 pandemic, respectively.

[33] Mini-batch training is believed to improve generalization performance; see Masters and Luschi (2018).

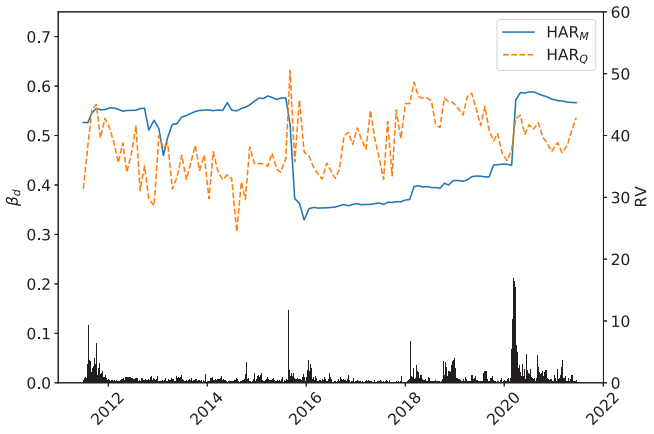


Fig. C.1. Trajectories of β_d in HAR trained with different losses.
Note: The left y-axis represents the estimated values of β_d every month, while the right y-axis represents the daily RV of the S&P 500 ETF shown in bar charts.

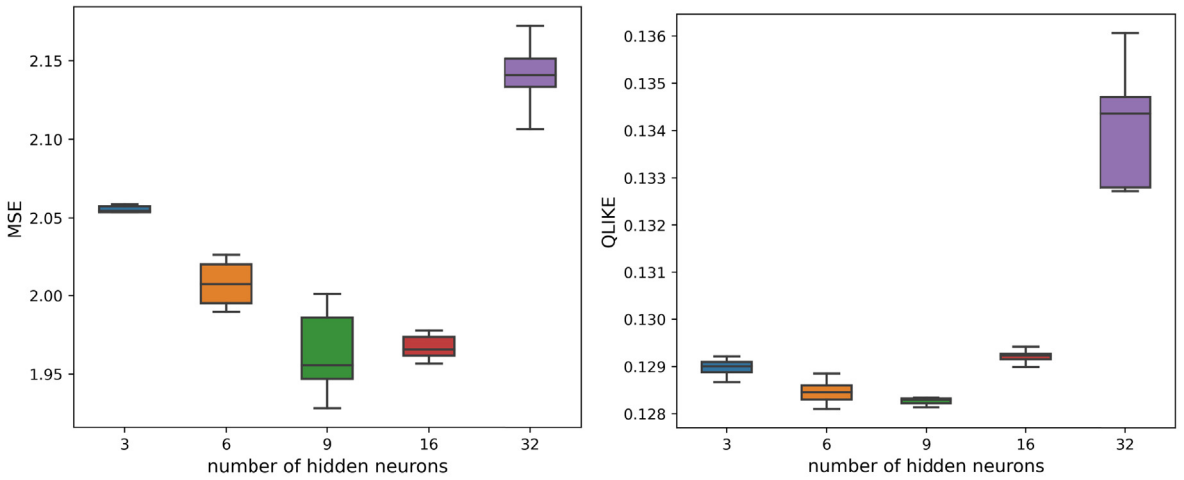


Fig. D.1. Validation performance under different dimensions of hidden representations in GNNHAR1L$_M$.
Note: Each box is obtained from 10 replicated experiments with different random initial parameters.

and further details, we refer the reader to Example 1 and Fig. 2. In our experiments, we used the normalized adjacent matrix of two-hop neighbors and estimated (18) through OLS.

The DM test results between GHAR2Hop and GHAR are presented in Fig. E.1. The cross-sectional DM test value is approximately -1, with a corresponding p-value of approximately 35%. These results reinforce the primary findings regarding the role of multi-hop neighbors, indicating that including two-hop neighbors may not provide substantial additional predictive power.

In Fig. E.2, we conduct a detailed examination of the coefficients associated with K-hop neighbors across different forecasting horizons. Based on the given definitions, the zero-hop coefficients for the daily (resp. weekly and monthly) horizon represent β_d (resp. β_w and β_m), the one-hop coefficients correspond to γ_d (resp. γ_w and γ_m), and the two-hop coefficients denote δ_d (resp. δ_w and δ_m). Fig. E.2 reveals that the coefficients at zero hops are positive over three horizons (i.e. $\beta_d, \beta_w, \beta_m > 0$), consistent with previous literature (Bollerslev, Patton, & Quaedvlieg, 2018). We also observe that the daily coefficients are positive on average but rapidly decay with distance (i.e. $\beta_d > \gamma_d > \delta_d$). Specifically, the daily coefficient associated with two-hop neighbors is approximately one-eighth (one-sixteenth) relative to the coefficient of their one-hop (zero-hop) counterparts. Another interesting observation is that the weekly and monthly coefficients are negative, potentially due to high collinearity, as highlighted by Zhang et al. (2022). Nonetheless, the magnitude of these coefficients diminishes as the distance increases, suggesting that the influence of the two-hop neighbors may be negligible.

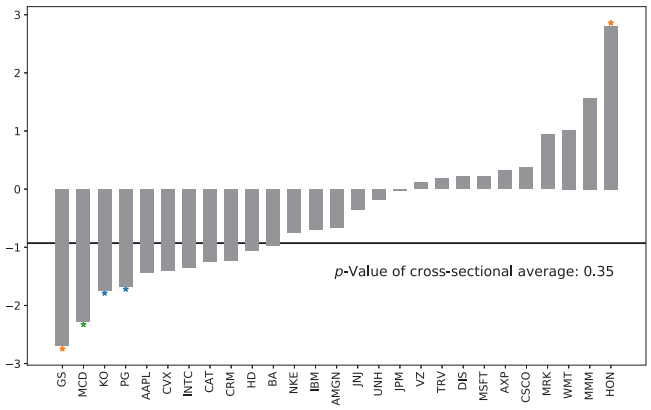


Fig. E.1. DM test between GHAR2Hop and GHAR.
Note: A positive (negative) number indicates superiority for the GHAR (GHAR2Hop) model. The *y*-axis represents the DM test values based on QLs between GHAR2Hop and GHAR, while the *x*-axis lists the stock symbols. Stars indicate the *p*-values, with orange, green, and blue representing significance at the 1%, 5%, and 10% levels, respectively. The horizon line represents the cross-sectional DM test value and its corresponding *p*-value. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

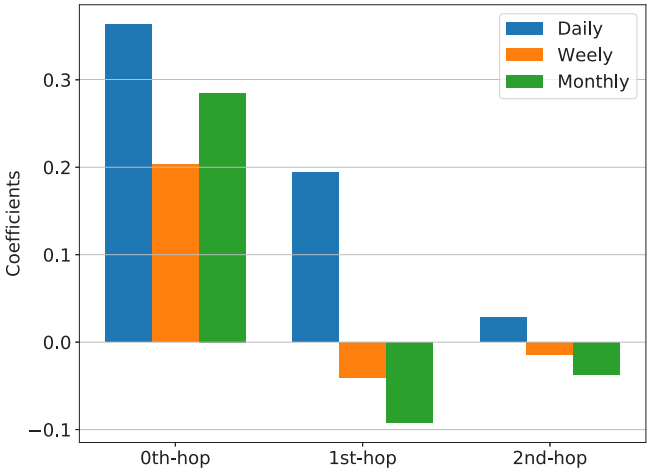


Fig. E.2. Coefficients in GHAR2Hop. *Note:* This figure describes the average coefficients of different hop neighborhoods over multiple horizons.

References

Acemoglu, Daron, Ozdaglar, Asuman, & Tahbaz-Salehi, Alireza (2010). *Cascades in networks and aggregate volatility*: *Technical report*, National Bureau of Economic Research.

Ali, Usman, & Hirshleifer, David (2020). Shared analyst coverage: Unifying momentum spillover effects. *Journal of Financial Economics*, *136*(3), 649–675.

Alon, Uri, & Yahav, Eran (2020). On the bottleneck of graph neural networks and its practical implications. In *International conference on learning representations*.

Andersen, Torben G., Bollerslev, Tim, Diebold, Francis X., & Ebens, Heiko (2001). The distribution of realized stock return volatility. *Journal of Financial Economics*, *61*(1), 43–76.

Andersen, Torben G., Bollerslev, Tim, & Meddahi, Nour (2011). Realized volatility forecasting and market microstructure noise. *Journal of Econometrics*, *160*(1), 220–234.

Anselin, Luc (2022). Spatial econometrics. In *Handbook of spatial analysis in the social sciences* (pp. 101–122).

Bai, Zhidong, Wong, Wing-Keung, & Zhang, Bingzhi (2010). Multivariate linear and nonlinear causality tests. *Mathematics and Computers in Simulation*, *81*(1), 5–17.

Barndorff-Nielsen, Ole E., & Shephard, Neil (2002). Econometric analysis of realized volatility and its use in estimating stochastic volatility models. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *64*(2), 253–280.

Basturk, Nalan, Schotman, Peter C., & Schyns, Hugo (2022). A neural network with shared dynamics for multi-step prediction of value-at-risk and volatility. Available at SSRN 3871096.

Bauwens, Luc, Hafner, Christian M., & Laurent, Sébastien (2012). *Handbook of volatility models and their applications*: *vol. 3*, John Wiley & Sons.

Bollerslev, Tim, Hood, Benjamin, Huss, John, & Pedersen, Lasse Heje (2018). Risk everywhere: Modeling and managing volatility. *The Review of Financial Studies*, *31*(7), 2729–2773.

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

Bollerslev, Tim, Patton, Andrew J., & Quaedvlieg, Rogier (2018). Modeling and forecasting (un) reliable realized covariances for more reliable financial decisions. *Journal of Econometrics*, *207*(1), 71–91.

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

Buncic, Daniel, & Gisler, Katja I. M. (2016). Global equity market volatility spillovers: A broader role for the United States. *International Journal of Forecasting*, *32*(4), 1317–1339.

Callot, Laurent A. F., Kock, Anders B., & Medeiros, Marcelo C. (2017). Modeling and forecasting large realized covariance matrices and portfolio choice. *Journal of Applied Econometrics*, *32*(1), 140–158.

Caporin, Massimiliano, Rossi, Eduardo, & De Magistris, Paolo Santucci (2017). Chasing volatility: A persistent multiplicative error model with jumps. *Journal of Econometrics*, *198*(1), 122–145.

Chen, Deli, Lin, Yankai, Li, Wei, Li, Peng, Zhou, Jie, & Sun, Xu (2020). Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *AAAI conference on artificial intelligence, vol. 34* (pp. 3438–3445).

Chen, Qinkai, & Robert, Christian-Yann (2022). Multivariate realized volatility forecasting with graph neural network. In *Proceedings of the third ACM international conference on AI in finance* (pp. 156–164).

Chen, Yingmei, Wei, Zhongyu, & Huang, Xuanjing (2018). Incorporating corporation relationship via graph convolutional neural networks for stock price prediction. In *Proceedings of the 27th ACM international conference on information and knowledge management* (pp. 1655–1658).

Chinco, Alex, Clark-Joseph, Adam D., & Ye, Mao (2019). Sparse signals in the cross-section of returns. *The Journal of Finance*, *74*(1), 449–492.

Choudhry, Taufiq, Papadimitriou, Fotios I., & Shabi, Sarosh (2016). Stock market volatility and business cycle: Evidence from linear and nonlinear causality tests. *Journal of Banking & Finance*, *66*, 89–101.

Cipollini, Fabrizio, Gallo, Giampiero M., & Otranto, Edoardo (2021). Realized volatility forecasting: Robustness to measurement errors. *International Journal of Forecasting*, *37*(1), 44–57.

Cipollini, Fabrizio, Gallo, Giampiero M., & Palandri, Alessandro (2020). Realized variance modeling: Decoupling forecasting from estimation. *Journal of Financial Econometrics*, *18*(3), 532–555.

Clements, Adam, & Preve, Daniel P. A. (2021). A practical guide to harnessing the HAR volatility model. *Journal of Banking & Finance*, *133*, Article 106285.

Corsi, Fulvio (2009). A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics*, *7*(2), 174–196.

Dai, Hanjun, Kozareva, Zornitsa, Dai, Bo, Smola, Alex, & Song, Le (2018). Learning steady-states of iterative algorithms over graphs. In *International conference on machine learning* (pp. 1106–1114). PMLR.

Defferrard, Michaël, Bresson, Xavier, & Vandergheynst, Pierre (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*.

Degiannakis, Stavros, & Filis, George (2017). Forecasting oil price realized volatility using information channels from other asset classes. *Journal of International Money and Finance*, *76*, 28–49.

Diebold, Francis X., & Mariano, Roberto S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, *13*(3), 253–263.

Diebold, Francis X., & Yılmaz, Kamil (2014). On the network topology of variance decompositions: Measuring the connectedness of financial firms. *Journal of Econometrics*, *182*(1), 119–134.

Engle, Robert (2002). New frontiers for ARCH models. *Journal of Applied Econometrics*, *17*(5), 425–446.

Engle, Robert F., & Kroner, Kenneth F. (1995). Multivariate simultaneous generalized ARCH. *Econometric Theory*, *11*(1), 122–150.

Fan, Jianqing, Qi, Lei, & Xiu, Dacheng (2014). Quasi-maximum likelihood estimation of GARCH models with heavy-tailed likelihoods. *Journal of Business & Economic Statistics*, *32*(2), 178–191.

Feng, Jiarui, Chen, Yixin, Li, Fuhai, Sarkar, Anindya, & Zhang, Muhan (2022). How powerful are K-hop message passing graph neural networks. In *Advances in neural information processing systems*.

Friedman, Jerome, Hastie, Trevor, & Tibshirani, Robert (2008). Sparse inverse covariance estimation with the graphical LASSO. *Biostatistics*, *9*(3), 432–441.

Goyenko, Ruslan, Kelly, Bryan T., Moskowitz, Tobias J., Su, Yinan, & Zhang, Chao (2024). Trading volume alpha. Available at SSRN.

Gu, Shihao, Kelly, Bryan, & Xiu, Dacheng (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, *33*(5), 2223–2273.

Hall, Peter, & Yao, Qiwei (2003). Inference in ARCH and GARCH models with heavy-tailed errors. *Econometrica*, *71*(1), 285–317.

Hansen, Peter Reinhard, & Dumitrescu, Elena-Ivona (2022). How should parameter estimation be tailored to the objective? *Journal of Econometrics*, *230*(2), 535–558.

Hansen, Peter Reinhard, Lunde, Asger, & Nason, James M. (2003). Choosing the best volatility models: The model confidence set approach. *Oxford Bulletin of Economics and Statistics*, *65*, 839–861.

Hansen, Peter Reinhard, Lunde, Asger, & Nason, James M. (2011). The model confidence set. *Econometrica*, *79*(2), 453–497.

Harvey, David, Leybourne, Stephen, & Newbold, Paul (1997). Testing the equality of prediction mean squared errors. *International Journal of Forecasting*, *13*(2), 281–291.

Hecq, Alain, Margaritella, Luca, & Smeekes, Stephan (2023). Granger causality testing in high-dimensional VARs: A post-double-selection procedure. *Journal of Financial Econometrics*, *21*(3), 915–958.

Herskovic, Bernard, Kelly, Bryan, Lustig, Hanno, & Van Nieuwerburgh, Stijn (2020). Firm volatility in granular networks. *Journal of Political Economy*, *128*(11), 4097–4162.

Karpman, Kara, Basu, Sumanta, Easley, David, & Kim, Sanghee (2023). Learning financial networks with high-frequency trade data. *Data Science in Science*, *2*(1), Article 2166624.

Kingma, Diederik P., & Ba, Jimmy (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kipf, Thomas N., & Welling, Max (2017). Semi-supervised classification with graph convolutional networks. In *International conference on learning representations*.

LeSage, James. P. (1999). *The theory and practice of spatial econometrics*: *vol. 28*, (no. 11), (pp. 1–39). Toledo, Ohio: University of Toledo.

Li, Qimai, Han, Zhichao, & Wu, Xiao-Ming (2018). Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI conference on artificial intelligence*.

Li, Sophia Zhengzi, & Tang, Yushan (2021). Automated volatility forecasting. Available at SSRN 3776915.

Liang, Ting, Zeng, Guanxiong, Zhong, Qiwei, Chi, Jianfeng, Feng, Jinghua, Ao, Xiang, & Tang, Jiayu (2021). Credit risk and limits forecasting in e-commerce consumer lending service via multi-view-aware mixture-of-experts nets. In *Proceedings of the 14th ACM international conference on web search and data mining* (pp. 229–237).

Ling, Shiqing, & McAleer, Michael (2003). Asymptotic theory for a vector ARMA-GARCH model. *Econometric Theory*, *19*(2), 280–310.

Liu, Ziqi, Chen, Chaochao, Li, Longfei, Zhou, Jun, Li, Xiaolong, Song, Le, & Qi, Yuan (2019). Geniepath: Graph neural networks with adaptive receptive paths. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 4424–4431).

Liu, Ziqi, Chen, Chaochao, Yang, Xinxing, Zhou, Jun, Li, Xiaolong, & Song, Le (2018). Heterogeneous graph neural networks for malicious account detection. In *Proceedings of the 27th ACM international conference on information and knowledge management* (pp. 2077–2085).

Liu, Han, Lafferty, John, & Wasserman, Larry (2009). The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *Journal of Machine Learning Research*, *10*(10), 2295–2328.

Liu, Lily Y., Patton, Andrew J., & Sheppard, Kevin (2015). Does anything beat 5-minute RV? A comparison of realized measures across multiple asset classes. *Journal of Econometrics*, *187*(1), 293–311.

Masters, Dominic, & Luschi, Carlo (2018). Revisiting small batch training for deep neural networks. arXiv preprint arXiv:1804.07612.

Pascalau, Razvan, & Poirier, Ryan (2021). Increasing the information content of realized volatility forecasts. *Journal of Financial Econometrics*, *21*(4), 1064–1098.

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

Patton, Andrew J., & Sheppard, Kevin (2009). Evaluating volatility and correlation forecasts. In *Handbook of financial time series* (pp. 801–838). Springer.

Patton, Andrew J., & Sheppard, Kevin (2015). Good volatility, bad volatility: Signed jumps and the persistence of volatility. *The Review of Economics and Statistics*, *97*(3), 683–697.

Reisenhofer, Rafael, Bayer, Xandro, & Hautsch, Nikolaus (2022). HARNet: A convolutional neural network for realized volatility forecasting. arXiv preprint arXiv:2205.07719.

Sawhney, Ramit, Agarwal, Shivam, Wadhwa, Arnav, & Shah, Rajiv (2020). Deep attentive learning for stock movement prediction from social media text and company correlations. In *Proceedings of the 2020 conference on empirical methods in natural language processing* (pp. 8415–8426).

Scarselli, Franco, Gori, Marco, Tsoi, Ah Chung, Hagenbuchner, Markus, & Monfardini, Gabriele (2008). The graph neural network model. *IEEE Transactions on Neural Networks*, *20*(1), 61–80.

Sheppard, Kevin (2010). *Financial econometrics notes* (pp. 333–426). University of Oxford.

Shuman, David I., Narang, Sunil K., Frossard, Pascal, Ortega, Antonio, & Vandergheynst, Pierre (2013). The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, *30*(3), 83–98.

Symitsi, Efthymia, Symeonidis, Lazaros, Kourtis, Apostolos, & Markellos, Raphael (2018). Covariance forecasting in equity markets. *Journal of Banking & Finance*, *96*, 153–168.

Varneskov, Rasmus, & Voev, Valeri (2013). The role of realized ex-post covariance measures and dynamic model choice on the quality of covariance forecasts. *Journal of Empirical Finance*, *20*, 83–95.

Voorman, Arend, Shojaie, Ali, & Witten, Daniela (2014). Graph estimation with joint additive models. *Biometrika*, *101*(1), 85–101.

Wang, Daixin, Lin, Jianbin, Cui, Peng, Jia, Quanhui, Wang, Zhen, Fang, Yanming, Yu, Quan, Zhou, Jun, Yang, Shuang, & Qi, Yuan (2019). A semi-supervised graph attentive network for financial fraud detection. In *2019 IEEE international conference on data mining* (pp. 598–607). IEEE.

Wang, Yudong, Wei, Yu, Wu, Chongfeng, & Yin, Libo (2018). Oil and the short-term predictability of stock return volatility. *Journal of Empirical Finance*, *47*, 90–104.

Wilms, Ines, Rombouts, Jeroen, & Croux, Christophe (2021). Multivariate volatility forecasts for stock market indices. *International Journal of Forecasting*, *37*(2), 484–499.

Wu, Zonghan, Pan, Shirui, Chen, Fengwen, Long, Guodong, Zhang, Chengqi, & Philip, S. Yu (2020). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(1), 4–24.

Zhang, Chao, Pu, Xingyue Stacy, Cucuringu, Mihai, & Dong, Xiaowen (2022). Graph-based methods for forecasting realized covariances. Available at SSRN.

Zhang, Chao, Zhang, Yihuang, Cucuringu, Mihai, & Qian, Zhongmin (2024). Volatility forecasting with machine learning and intraday commonality. *Journal of Financial Econometrics*, *22*(2), 492–530.