Multivariate Stochastic Volatility Models based on Generalized Fisher Transformation*

Han Chen
Hunan University

Yijie Fei
Hunan University

Jun Yu
University of Macau

May 21, 2025

Abstract

Modeling multivariate stochastic volatility (MSV) can pose significant challenges, particularly when both variances and covariances are time-varying. In this study, we tackle these complexities by introducing novel MSV models based on the generalized Fisher transformation (GFT) proposed by Archakov and Hansen (2021). Our model exhibits remarkable flexibility, ensuring the positive-definiteness of the variance-covariance matrix, and disentangling the driving forces of volatilities and correlations. To conduct Bayesian analysis of the models, we employ a Particle Gibbs Ancestor Sampling (PGAS) method, facilitating efficient Bayesian model comparisons. Furthermore, we extend our MSV model to cover leverage effects and incorporate realized measures. Our simulation studies demonstrate that the proposed method performs well for our GFT-based MSV model. Furthermore, empirical studies based on equity returns show that the MSV models outperform alternative specifications in both in-sample and out-of-sample performances.

Keywords: Multivariate stochastic volatility; Dynamic correlation; Leverage effect; Particle filter; Markov chain Monte Carlo; Realized measures
JEL Codes: G10, C53, C12, C32, C58

*The author for correspondence is Yijie Fei. We would like to thank Viktor Todorov (the co-editor), an associate editor, two referees, Peter Hansen, and Yanping Yi for useful discussions and comments. Han Chen (Email: hanchen@hnu.edu.cn) and Yijie Fei (Email: yijiefei@hnu.edu.cn), College of Finance and Statistics, Hunan University, 109 Shijiachong Road, Changsha, China. Jun Yu (Email: junyu@um.edu.mo), Faculty of Business Administration, University of Macau, Avenida da Universidade, Macao, China. Yijie Fei acknowledges the financial support from Fundamental Research Funds for the Central Universities in China and Natural Science Foundation of Hunan Province (No.2025JJ60440).

1 Introduction

The characterization of the dynamic behavior of return volatility is crucial for asset pricing, portfolio allocation, and risk management. Univariate volatility models have been extensively studied in the literature since the seminal paper by Engle (1982). These models can be broadly categorized into two types: GARCH-based and stochastic volatility (SV) models. In recent decades, there has been a growing focus on multivariate financial data analysis. It is now widely recognized that analyzing asset returns individually is insufficient, and the dependence structure among assets must be taken into account. To address this, a plethora of multivariate extensions to univariate GARCH and SV models have been proposed and applied in practice. Multivariate GARCH (MGARCH) models have been extensively reviewed in Bauwens et al. (2006), while multivariate SV (MSV) models have been reviewed in Asai et al. (2006). These multivariate models enable us to capture the co-movements of volatilities and correlations among multiple assets, providing a more accurate representation of the underlying dependence structure. The development of multivariate models has significantly improved our understanding of asset return dynamics and has important practical implications for financial risk management and investment strategies.

The first multivariate stochastic volatility (MSV) model, proposed by Harvey et al. (1994), is an extension of the constant conditional correlation (CCC) model in multivariate GARCH (MGARCH). In this basic setup, each asset's volatility is modeled by a univariate stochastic volatility process, while the correlation matrix among all assets remains constant over time. However, this assumption is rather restrictive. Subsequent efforts have been devoted to relaxing this assumption in the MSV literature. For instance, Yu and Meyer (2006) proposed a model that mirrors the dynamic conditional correlation (DCC) model of Engle (2002) in MGARCH. The DCC-based model allows for time-varying correlation among assets while still assuming that each asset's volatility follows a univariate stochastic volatility process. Another parametrization based on DCC can be found in Asai and McAleer (2009b). Other studies have proposed even more flexible models that allow for both time-varying volatilities and correlations among assets.

In this paper, we propose a new MSV model that builds upon a recently developed parameterization of the correlation matrix. This parameterization, first introduced in Archakov and Hansen (2021), is a generalization of the well-known Fisher z-transformation (GFT hereafter) from the bivariate case to the multivariate case. It has been successfully used in other models, such as the multivariate realized GARCH model of Archakov et al. (2024a) and the dynamic conditional score model of Hafner and Wang (2023). It has also been suggested for the purpose of generating random correlation matrices in Archakov et al. (2024b) and

macroeconomic forecasting in Arias et al. (2023). Recent simulation and empirical evidence in Bucci et al. (2022) shows that this parameterization provides more accurate forecasts of the realized covariance matrix than other existing methods. Our paper is among the first studies that introduce GFT to the MSV literature.[1]

Our new MSV model allows the underlying latent variables that determine the correlations among assets to have an unrestricted domain because the correlation matrix is always valid by construction. In addition, the shocks to the volatility dynamics and the correlation dynamics are fully separated in our model. This is an appealing feature, as in practice, these two types of shocks may be determined by completely distinct factors. Finally, our model is invariant to the reordering of assets, which eliminates the need for an ex-ante ordering of assets. All of these features indicate that our model is highly flexible and imposes a minimal level of ex-ante restrictions.

Beyond our basic model, we also propose two extensions that have been proved to be beneficial in modeling multivariate volatility. The first one includes asymmetric effects in our model and the second incorporates information from realized measures when high-frequency data is available.

The importance of accommodating asymmetric effects in the volatility literature has long been recognized. For equity returns, it has been emphasized that bad news has a greater impact on future volatility than good news, known as the leverage effect. This has been incorporated in several existing MSV models, such as Asai and McAleer (2006) and Asai and McAleer (2009a). To allow for asymmetric effects across multiple assets in an MSV model, Ishihara et al. (2016) and Asai et al. (2022) propose to consider the lower-diagonal elements of the matrix logarithm of the covariance matrix and assume that the return vector is correlated with modeled variables, which they call cross-leverage. However, this approach is inconsistent with the original idea of leverage effect since the latent variables are generated by complex nonlinear transformations of the covariance matrix and correspond to both volatility and correlation. As argued in Asai et al. (2006), leverage should refer only to the negative correlations between the current return and future volatility. To address this issue, in the present paper, we extend the basic MSV model to allow explicitly for volatilities and returns to be correlated.

Classical MSV modelling relies solely on daily return data for estimation, thus not fully harnessing the available information. An additional valuable source for capturing return fluctuations is realized volatility (RV), computed from intra-day high-frequency data; for a

[1]While Arias et al. (2023) introduce the Fisher transformation to a MSV model, both volatility and correlation are assumed to be non-stationary in Arias et al. (2023). This random walk assumption is well known to be violated for financial assets. In our model, we do not make the random walk assumption.

comprehensive overview, refer to Andersen et al. (2010). Research indicates that models integrating realized measures can significantly enhance parameter estimation efficiency and model fit, as highlighted in works by Hansen et al. (2012) and Hurn et al. (2020). Motivated by these insights, literature has introduced SV models – termed RSV models – that leverage both return series and RV data. Prior studies integrating univariate SV models with realized measures include Koopman and Scharth (2012), Venter and de Jongh (2014), and Asai et al. (2017). Recent advancements in multivariate modeling along this line include Shirota et al. (2017), Kurose and Omori (2020), Yamauchi and Omori (2020) and Asai et al. (2022). In our study, we extend this approach by integrating realized measures into our MSV model. This augmentation involves applying the new transformation to the realized covariance matrix, furnishing additional measurements to the latent variables. As suggested by Yamauchi and Omori (2020), this additional information plays a crucial role in stabilizing parameter estimation processes.

In our study, we introduce a Bayesian statistical framework for analyzing the proposed MSV models. Different from the conventional Bayesian MSV literature, which predominantly employs standard Markov chain Monte Carlo (MCMC) techniques, we adopt a recently developed Particle MCMC (PMCMC) algorithm. PMCMC algorithms have gained attraction following the seminal work by Andrieu et al. (2010) and have found applications across diverse domains. While theoretically versatile for a wide spectrum of models, the practical efficacy of PMCMC algorithms hinges on several factors and necessitates meticulous evaluation. In the present paper, we opt for the Particle Gibbs Ancestor Sampling (PGAS) method of Lindsten et al. (2014), a refined version of the Particle Gibbs (PG) sampler that offers enhanced mixing properties, even with a small number of particles.

Our inference procedure involves an inverse transformation that is generally time-consuming due to the lack of a closed-form solution. Inefficient handling of this transformation could impose constraints on the scalability of our model. To surmount this obstacle, we propose ways to improve the numerical method of Archakov and Hansen (2021). We present extensive simulation evidence to justify our choice of the estimation strategy and provide useful guidance for empirical applications.

The rest of the paper is organized as follows. Section 2 introduces the new parametrization of correlation matrix, and presents our basic MSV model. Section 3 introduces the estimation and inferential method based on the PGAS algorithm. Section 4 focuses on the efficient treatment of the inversion transform in our inference procedure. Section 5 reports simulation evidence to support our proposed method. Section 6 extends our basic model to incorporate the leverage effect and the realized measures. Empirical studies are provided in Section 7. Section 8 concludes. The Online Supplement includes additional details and

materials that complement and support our main text.

Throughout the paper, we let $diag(A)$ denote the column vector formed by the diagonal elements of a square matrix A or the diagonal matrix whose diagonal elements are elements in A if A is a column vector;[2] $vech(A)$ denote the $p(p+1)/2 \times 1$ column-vector obtained by vectorizing only the lower triangular part of a p-dimensional matrix A (including the diagonal elements); $vecl(A)$ denote the $p(p-1)/2 \times 1$ column-vector containing all lower off-diagonal elements of A (excluding the diagonal elements); I_p denote a p-dimensional identity matrix, $\mathbb{1}_p$ denote a p-dimensional column vector of ones; $I(x)$ denote the indicator function.

2 A New Multivariate Stochastic Volatility Model

In this section, we introduce the generalized Fisher transformation (GFT) of Archakov and Hansen (2021) (AH hereafter), and propose a new MSV model that utilizes GFT. Section A.1 in Online Supplement provides a comprehensive review of existing MSV models, with a focus on the specification of dynamic covariance matrix.

2.1 Generalized Fisher transformation of correlation matrix

When the correlation coefficient between two random variables, say ρ, is to be modeled, an essential constraint is that its value must be within the interval $(-1, 1)$. To avoid the complexity introduced by this constraint in modeling, one can instead model the Fisher z-transformation of ρ, defined as

$$g = \frac{1}{2} \log \frac{1+\rho}{1-\rho} := F(\rho) \tag{1}$$

It is easy to show that

$$\rho = F^{-1}(g) = \frac{\exp(2g) - 1}{\exp(2g) + 1} \in (-1, 1), \forall g \in (-\infty, \infty)\,. \tag{2}$$

Therefore, one can impose any structure on $F(\rho)$ and transform it back to obtain ρ without worrying about the validity of the resulting correlation coefficient. This idea was first introduced to the MSV literature by Yu and Meyer (2006) when the number of assets is two.[3] Unfortunately, it is acknowledged by Yu and Meyer (2006) that this approach "is not easy to be generalized into higher dimension situations". In particular, a pairwise transformation

[2] If A is a square matrix, $diag(diag(A))$ is a diagonal matrix whose diagonal elements are the diagonal elements in A.

[3] Note that the expression used in Yu and Meyer (2006) is slightly different from (1), with the latter one aligning with the definition of GFT in case $p = 2$.

applied to each entry in a high-dimensional correlation matrix, though seems to be natural, is not a valid choice as it fails to ensure the positive-definiteness of the resulting correlation matrix in general.

Clearly, it is desirable to obtain a valid high-dimensional extension to the Fisher z-transformation. This is the exact contribution made in AH. To fix the idea, let R be a valid p-dimensional correlation matrix and[4]

$$G = \log R = \sum_{k=1}^{\infty} \frac{(-1)^k (R-I)^k}{k}.$$

Note that the convergence of the infinite summation and hence, the existence of G are ensured by the fact that R is a correlation matrix. Furthermore, let $q = vecl(G)$. In summary, GFT of R is defined by the mapping $q = vecl(\log R)$. One of key theoretical contributions of AH is demonstrating that this mapping is bijective. Thus, given any $\frac{p(p-1)}{2}$-dimensional vector q, there exists a unique and valid p-dimensional correlation matrix R. Although the inverse mapping from q to R does not have a closed-form expression when $p > 2$, R can be obtained numerically from q using an iterative algorithm as shown in AH; see Section 4 below for more discussions on computational issues.

When $p = 2$, AH show that the above-defined transformation reduces to the Fisher z-transformation. The new transformation retains the advantages of the Fisher z-transformation and enjoy some additional desirable properties. First and foremost, it is very flexible in the sense that, when modeling q, no algebraic constraint is needed. This suggests that we can consider any reasonable dynamics for q without worrying about the positive-definiteness of the resulting correlation matrix. Second, compared with original elements in R, the sample distribution of elements in q is often closer to Gaussian due to the use of log transformation. Hence, it is reasonable to model elements of q via a Gaussian process. Third, this transformation is invariant to the order of the variables. This is in sharp contrast to that based on the Cholesky decomposition. Fourth, although elements of q depend on R in a nonlinear way, many interesting properties in R carry over to $G = \log(R)$, including the equicorrelation structure and the block-equicorrelation structure; see Archakov and Hansen (2024). For the sake of notational simplicity, in the rest of the paper, we refer to the mapping $vecl(\log(\cdot))$ as $F(\cdot)$ and its inverse as $F^{-1}(\cdot)$.

[4]This formulation of matrix logarithm is correct only for R sufficiently close to the identity matrix, for the sum to exist. See Higham (2008) for a more general definition of matrix logarithm.

2.2 Basic MSV-GFT model

To introduce our basic MSV model, for $t = 1, \cdots, T$, let $r_t = (r_{1t}, ..., r_{pt})'$ denote the $p \times 1$ vector of asset returns and $h_t = (h_{1t}, ..., h_{pt})'$ the vector of latent log-volatilities of these returns at time t. Let $V_t = \exp(diag(h_t))$. Let $q_t = (q_{1t}, ..., q_{dt})'$ denote the vector of latent variables at time t that underlie all the correlation coefficients in R_t, where $d = \frac{p(p-1)}{2}$. In particular, q_t is connected to R_t through the transformation detailed in Section 2.1. Our basic MSV model, which we refer to as MSV-GFT, is given by

$$r_t = V_t^{1/2}\epsilon_t,\ \epsilon_t \sim N(0, R_t), \tag{3a}$$

$$V_t = \exp(diag(h_t)), \tag{3b}$$

$$q_t = F(R_t), \tag{3c}$$

$$h_{t+1} = \mu_h + \Phi_h(h_t - \mu_h) + \eta_{ht},\ \eta_{ht} \sim N(0, \Sigma_h), \tag{3d}$$

$$q_{t+1} = \mu_q + \Phi_q(q_t - \mu_q) + \eta_{qt},\ \eta_{qt} \sim N(0, \Sigma_q), \tag{3e}$$

$$h_0 \sim N\left(\mu_h, (I_p - \Phi_h^2)^{-1}\Sigma_h\right),\ q_0 \sim N\left(\mu_q, (I_d - \Phi_q^2)^{-1}\Sigma_q\right), \tag{3f}$$

where $\epsilon_t = (\epsilon_{1t}, ..., \epsilon_{pt})'$, $\eta_{ht} = (\eta_{h1t}, ..., \eta_{hpt})'$, $\eta_{qt} = (\eta_{q1t}, ..., \eta_{qdt})'$, $\mu_h = (\mu_{h1}, ..., \mu_{hp})'$, $\mu_q = (\mu_{q1}, ..., \mu_{qd})'$, $\Phi_h = diag((\phi_{h1}, ..., \phi_{hp})')$, $\Phi_q = diag((\phi_{q1}, ..., \phi_{qd})')$, and $t = 1, ..., T$. It is assumed that ϵ_t, η_{ht} and η_{qt} are independent. This implies that no leverage (neither self-leverage or cross-leverage) effect is allowed. Such an assumption will be relaxed in Section 6.1. It also implies that the shocks to the volatility dynamics (i.e. η_{ht}) are completely separated from those to the correlation dynamics (i.e. η_{qt}). To reduce the number of parameters, we further assume that $\Sigma_h = diag((\sigma_{h1}^2, ..., \sigma_{hp}^2)')$ and $\Sigma_q = diag((\sigma_{q1}^2, ..., \sigma_{qd}^2)')$.

In MSV-GFT, h_t is a p-dimensional latent variable that determines the volatilities via the exponential transformation and q_t is a d-dimensional latent variable that determines the correlation coefficients via the F transformation. Elements of two types of latent variables are assumed to follow independent Gaussian AR(1) processes.[5] It is important to note that in MSV-GFT, persistence in elements of q_t can be heterogeneous across pairs. This is in sharp contrast to models based on the idea of DCC or the Wishart autoregression, where persistence of all the correlation sequences is assumed to be the same. Yamauchi and Omori (2020) propose to model the dynamics of Fisher-transformed pairwise correlations by random walks without drift. This is equivalent to imposing $\mu_q = 0$ and $\Phi_h = I_d$ in (3e). Our

[5]Section 7.3 provides empirical evidence based on high-frequency data that supports the independent Gaussian assumption for GFT-transformed correlations q_t. See also Section 4.4 of Archakov et al. (2024a) for a similar finding.

specification is apparently more flexible and realistic than theirs.

3 Inference of MSV-GFT Model

Due to the difficulty of evaluating the likelihood function, the literature on MSV models relies on Bayesian methods to carry out statistical inference. In this section, we discuss in details the estimation of our MSV-GFT model within a Bayesian framework.

3.1 Gibbs sampler based on particle filter

In this paper, instead of using standard MCMC techniques,[6] we apply a PMCMC method known as PG, due to Andrieu et al. (2010), to estimate the proposed MSV model.[7] The intuition is to construct a high-dimensional efficient Markov kernel for latent processes using the particle filter. See Section B in Online Supplement for a brief introduction to PG.

As a PMCMC method, PG enjoys a few desirable properties compared with standard MCMC methods. First, relative to the single-move sampler, a significant improvement can be achieved in terms of efficiency by PG.

Second, unlike the multi-move samplers that are model dependent, PG requires a minimal modification across different models, as long as they could be cast into a state-space form.

Third, an important by-product of the filtering strategy is the evaluation of likelihood $p(r|\theta)$, where $r = (r_1, \cdots, r_T)'$. Once $p(r|\theta)$ is known, the marginal likelihood $p(r)$ can be calculated easily. Two popular approaches have been used in practice to compare competing Bayesian models. The first one is based on the Bayes factor and the second one on the Deviance Information Criterion (DIC).[8] The computation of the Bayes factor requires $p(r)$ while the computation of DIC requires $p(r|\theta)$. Hence, model comparison is straightforward in PG.

[6] See Section A.2 in Online Supplement for a review of other Bayesian estimation methods for MSV model, with discussions on their pros and cons.

[7] Another PMCMC method potentially applicable here is Particle Metropolis-Hasting. See Xu and Jasra (2019) for its application in MSV model with constant correlation matrix and cross-leverage. It is not chosen, however, as it requires an accurate estimation of the likelihood and hence a very large number of particles.

[8] When comparing two candidate models (nested or non-nested), the log marginal likelihood of the first model minus that of the second model leads to the log Bayes factor (BF); see Kass and Raftery (1995). DIC is a Bayesian version of AIC with the aim of favouring models that are likely to make good predictions; see Spiegelhalter et al. (2002) and Li et al. (2020). The smaller DIC, the better the model.

3.2 Particle Gibbs with ancestor sampling

As noted in Lindsten et al. (2014) and Chopin and Singh (2015), the mixing of the Markov kernel induced by PG can be rather slow when there is path degeneracy. For the high-dimensional problem, such as the one we consider in this paper, path degeneracy is inevitable. To overcome this problem, Lindsten et al. (2014) propose to use an additional step called ancestor sampling in PG. The PGAS algorithm enjoys fast mixing of the Markov kernel even only a seemingly small number of particles are used in the underlying particle filter. Informally, in the original PG, when degeneracy occurs, the particle system collapses toward the chosen reference trajectory. Whereas, in the PGAS, it degenerates toward something entirely different. As a consequence, the update rates of latent variables are much higher with the additional ancestor sampling step. Therefore, the mixing is much faster.[9] This approach has also been used in Gong and Stoffer (2021) for efficient fitting of stochastic volatility. They show that, for univariate SV model, PGAS algorithm mixes well enough with only 20 particles.

For our purpose, a fast mixing under a small number of particles is highly desirable, as our likelihood function contains a component that has no closed-form solution and thus must be computed numerically. Although the cost for one-time computation is relatively low, it soon becomes infeasible when a vast number of particles are included in the system. Indeed, for MCMC with S iterations, if the sample size is T and N particles are used, $F^{-1}(\cdot)$ must be evaluated $S \times T \times N$ times. As S and T are usually quite large in practice, we can gain a lot in terms of computational efficiency by using the PGAS algorithm. In summary, we believe that PGAS is a suitable estimation tool given our model setup. Its performance will be further examined in simulation in Section 5.

3.3 Bayesian inference of MSV-GFT

We now present the Bayesian analysis of our MSV-GFT model. The first step is to specify the prior distributions of all the parameters $\theta = (\mu_h, \mu_q, \phi_h, \phi_q, \sigma_h^2, \sigma_q^2)'$. In this regard, our specification follows those adopted in Kim et al. (1998). For μ_h and μ_q, we assume independent multivariate normal distributions. The persistence parameters ϕ_h and ϕ_q are assumed to have Beta priors. The prior distribution of σ_h and σ_q are chosen to be inverse gamma. In summary, for $i = 1, ..., p$ and $j = 1, ..., d$, we choose the following prior distributions:

- $\mu_{hi} \sim N(m_{\mu 0}, s_{\mu 0}^2)$ and $\mu_{qj} \sim N(m_{\mu 0}, s_{\mu 0}^2)$;
- $\frac{\phi_{hi}+1}{2} \sim Beta(a, b)$ and $\frac{\phi_{qj}+1}{2} \sim Beta(a, b)$;

[9] Lindsten et al. (2014) also show that for a state-space model, PGAS is probabilistically equivalent to the particle Gibbs sampler with a backward smoothing step under certain conditions.

- $\sigma_{hi}^2 \sim IG(\frac{n_{m0}}{2}, \frac{d_{m0}}{2})$ and $\sigma_{qj}^2 \sim IG(\frac{n_{m0}}{2}, \frac{d_{m0}}{2})$,

where $m_{\mu 0}, s_{\mu 0}^2, a, b, n_{m0}, d_{m0}$ are hyperparameters.

To carry out the inference, we implement a Gibbs sampler with four blocks. In the following, we use $\theta_{/\alpha}$ to denote the parameters θ excluding α. The algorithm proceeds as:

1. Initialize h, q and θ.
2. Draw $h, q|r, \theta$.
3. Draw $\mu_h, \mu_q|r, h, q, \theta_{/(\mu_h,\mu_q)}$.
4. Draw $\phi_h, \phi_q|r, h, q, \theta_{/(\phi_h,\phi_q)}$.
5. Draw $\sigma_h^2, \sigma_q^2|r, h, q, \theta_{/(\sigma_h^2,\sigma_q^2)}$.

Iteration over steps 2-5 consists of a complete sweep of MCMC sampler. We apply PGAS introduced in Section 3.2 to sample the latent variables h and q given all the observations r and one particular set of parameter values. The detailed description of the algorithm is presented in Section C in Online Supplement. On the other hand, from the joint posterior density, it is straightforward to sample each element in θ given one realization of latent processes h and q. The details are provided in Section E in Online Supplement.

4 Inverting GFT

4.1 Review of AH's method

For the Bayesian method introduced in Section 3, the most time-consuming step is the evaluation of $F^{-1}(\cdot)$. The scalability of our model depends critically on how this step is efficiently handled.

In general a closed-form expression for $F^{-1}(\cdot)$ is not available. AH proposes a numerical solution to $F^{-1}(\cdot)$ as a root-finding problem. The idea is as follows. Since $\log R_t$ must be symmetric, it is uniquely identified through its diagonal elements $z_t = (z_{1t}, \cdots, z_{pt})'$ given all the off-diagonal elements q_t. As $R_t = \exp(\log R_t)$, finding a valid correlation matrix R_t given q_t is thus equivalent to finding an appropriate $p \times 1$ vector z_t. Using the fact that all the diagonal elements of a correlation matrix must be one, z_t can be found through solving the following equation

$$diag\left(e^{A[z_t]}\right) = \mathbb{1}_p, \tag{4}$$

where A is a symmetric matrix with $vecl(A) = q_t$ and $A[z_t]$ highlights the fact that $diag(A) = z_t$. As long as we find z_t^* that solves (4), the correlation matrix R_t is straightforwardly

recovered through $R_t = \exp(A[z_t^*])$. AH point out that (4) is a system of nonlinear equations and propose a fixed-point iteration method to solve this root-finding problem.

Algorithm 1: AH's method

Data: q ; // $\frac{p(p-1)}{2} \times 1$ vector

Result: z ; // $p \times 1$ vector

Set initial Value: $z_0 = \mathbf{0}_p$ **and** $f(z_0) = \inf$

for k = 0 : MaxIteration **do**

  if $||f(z)||_2 < \epsilon$ **then**

    Return z

  else

    Update z: $z_{k+1} = z_k - f(z_k)$; // Complexity: $O(p^3)$

We summarize AH's method in Algorithm 1.[10] The time complexity of this algorithm is $O(p^3K)$, which is determined by two factors. The first factor is the cost for each iteration, which is dominated by the matrix exponential operation with the $O(p^3)$ time complexity. The second factor is the number of iterations (K) before convergence. AH show that in general $K = O(\log p)$. However, the exact value of K is sensitive to the correlation structure. Simply put, when R is nearly singular, a large K is needed for the fixed-point algorithm to converge. In the following section, we consider two modifications that may potentially reduce the computational cost of inverting GFT.

4.2 Newton's Method and Broyden's Method

It is well-known that faster convergence to find roots may be achieved by using the Jacobian matrix. This motivates us to consider two alternative methods to AH's algorithm. The first approach is the classical Newton's method. Specifically, in each iteration, we utilize the closed-form expression of the Jacobian matrix, whose analytic expressions are given in Appendix of Archakov and Hansen (2021). This approach is summarized in Algorithm 2 below with the definition of $J(z)$ given in Section D of Online Supplement.

Unlike the fixed-point algorithm, Newton's method converges only if the initial value is in the neighborhood of the true root. Therefore, we propose to first conduct a few fixed-point updates before starting Newton's iteration. In practice, we find that running the fixed-point update just once is enough before implementing Newton's method in almost all cases.[11]

While Newton's method usually converges faster than the fixed-point iteration, it requires

[10] When implementing all algorithms, we set the maximum number of iterations to 1000 and the convergence criterion ϵ to 1×10^{-6}.

[11] Indeed, in our experiment, only 3 out of 10000 cases require more than one updates.

Algorithm 2: GFTI based on Newton's method

Data: q ; // $\frac{p(p-1)}{2} \times 1$ vector
Result: z ; // $p \times 1$ vector
Set initial Value: $z_0 = -f(\mathbf{0}_p)$ and $f(z_0) = \inf$
for $\underline{\text{k} = 0 : \text{MaxIteration}}$ **do**
  if $\underline{||f(z_k)||_2 < \epsilon}$ **then**
    Return z
  else
    Compute Jacobian: $J_k = J(z_k)$; // Complexity: $O(p^4)$
    Update z: $z_{k+1} = z_k - J_k^{-1} f(z_k)$; // Complexity: $O(p^3)$

calculation of Jacobian, which increases the time complexity to $O(p^4)$ and can be quite time-consuming if p is very large. To avoid the substantial cost required in computing Jacobian, we consider a quasi-Newton approach known as Broyden's method. The fundamental idea is to compute the Jacobian matrix only in the first iteration and then perform rank-one updates in subsequent iterations. Broyden's method is reported in Algorithm 3.

Algorithm 3: GFTI based on Broyden's method

Data: q ; // $\frac{p(p-1)}{2} \times 1$ vector
Result: z ; // $p \times 1$ vector
Set initial Value: $z_0 = -f(\mathbf{0}_p)$ and $f(z_0) = \inf$
for $\underline{\text{k} = 0 : \text{MaxIteration}}$ **do**
  if $\underline{||f(z_k)||_2 < \epsilon}$ **then**
    Return z
  else
    if $\underline{\text{k} = 0}$ **then**
      Compute Jacobian: $J_k = J(z_k)$; // Complexity: $O(p^4)$
    else
      Update Jacobian:
      $\Delta f(z_k) = f(z_k) - f(z_{k-1})$; // Complexity: $O(p^3)$
      $\Delta z_k = z_k - z_{k-1}$
      $J_k = J_{k-1} + \frac{\Delta f(z_k) - J_{k-1}\Delta z_k}{||\Delta z_k||_2} \Delta z_k'$
    Update z: $z_{k+1} = z_k - J_k^{-1} f(z_k)$; // Complexity: $O(p^3)$

As we only need to compute the Jacobian once in Broyden's method, the total time complexity is close to $O(p^3)$. In comparison to AH's fixed-point technique, Broyden's method exhibits accelerated convergence rates but demands a higher computational time per iteration. Relative to Newton's method, when p is sufficiently large, we anticipate a reduction

in the computational time per iteration by Broyden's method, albeit at the cost of slower convergence.

4.3 Comparison with AH's algorithm

In line with AH's analysis, the convergence rate in AH's method depends upon the singularity level of the correlation matrix, a characteristic often gauged by the minimum eigenvalue denoted as $\lambda_{\min}$.[12] For a correlation matrix to be invertible, $\lambda_{\min}$ must fall within the range of zero to one. If the correlation matrix is near-singular ($\lambda_{\min} \approx 0$), AH's method requires a large number of iterations, potentially rendering alternative algorithms utilizing Jacobian information more computationally effective. Conversely, in scenarios where the correlation matrix approaches an identity matrix ($\lambda_{\min} \approx 1$), the convergence in AH's method is expedited, obviating the necessity for Jacobian matrix computations. Thus, the comparative computational efficiency gains or losses associated with the adoption of Jacobian-based methodologies demand meticulous evaluation through comprehensive empirical investigations. We design two experiments to compare the performance of three methods.

Following AH, we first compare three methods based on the following Toeplitz-type correlation matrix

$$R = [R_{ij}], \quad i = 1, \ldots, p, \quad j = 1, \ldots, p,$$

where $R_{ij} = \rho^{|i-j|}$ with $\rho = [0.5, 0.9, 0.99]$ and $p \in \{3, 4, \cdots, 100\}$. For the dimensions considered, three values of ρ corresponds to $\log(\lambda_{\min}) \in [-0.9, 1.1]$, $[-2.67, -2.94]$ and $[-5.01, -5.29]$, respectively. In general, a larger value of ρ implies that the correlation matrix has a smaller $\lambda_{\min}$ and hence is closer to singularity.

The top panels of Figure 1 depict the average number of iterations required by the three methods as the model dimension p changes. It can be seen that, for Newton's method and Broyden's method, the required number of iterations are insensitive to both p and ρ. In almost all cases, both algorithms converge after 4 or 5 iterations. This stands in stark contrast to AH's method, which is remarkably sensitive to ρ. For instance, AH's method demands approximately 45 iterations for convergence, a figure that decreases to around 20 iterations when $\rho = 0.9$. Even in the least singular scenario ($\rho = 0.5$), AH's algorithm necessitates a higher number of iterations to achieve convergence in comparison to the other two methods.

However, a lower iteration count does not necessarily translate to an overall reduction in computational expenses. As previously mentioned, both Newton's method and Broyden's

[12] An operational challenge arises from the fact that the exact value of $\lambda_{\min}$ remains unknown until after the GFT inversion is conducted. Nevertheless, as highlighted in AH, the maximum absolute value of transformed variables q can serve as a dependable proxy for $\lambda_{\min}$.

method entail the computation of the Jacobian matrix. To assess the balance between convergence speed and Jacobian computation, the lower panels of Figure 1 depict the CPU time taken by the three methods against p. We observe that, due to the $O(p^4)$ time complexity at each iteration, Newton's method encounters a sharp escalation in CPU time when p increases. For example, when $\rho = 0.5$ and $p = 100$, the total CPU time required for Newton's method surpasses that of the other two methods by more than threefold. Consequently, it is apparent that Newton's method is ill-suited for scenarios with large p. Broyden's method, on the other hand, exhibits significant computational advantage over AH's method for $\rho = 0.99$, but is less attractive when $\rho = 0.5$. Intuitively, if the correlation matrix is close to singularity (i.e. a large ρ), Broyden's method demands fewer iterations. Given that it computes the Jacobian only once, the overall computational burden is lighter compared to AH's method. However, when the correlation matrix veers away from singularity (i.e., at lower ρ values), the fixed-point algorithm requires only marginally more iterations than Broyden's method. In such scenarios, the expenses incurred due to Jacobian calculation outweigh the savings derived from a reduced iteration count.

Next, we make a comparison based on randomly generated correlation matrices. For each $p \in \{5, 10, 15, 20\}$, we generate $100,000$ distinct correlation matrices and categorize them into different groups based on the $\log(\lambda_{\min})$ value.[13] The generating mechanism is akin to that in AH. Figure 2 depicts the CPU time taken by the three methods against $\log(\lambda_{\min})$. Notably, all three algorithms exhibit increased computational demands with higher p. However, the cumulative computational costs of Newton's method and Broyden's method demonstrate significantly lower sensitivity to $\log(\lambda_{\min})$ in comparison to AH's method. Conversely, the computational expenses associated with AH's method surge notably as $\log(\lambda_{\min})$ becomes more negative (i.e. the correlation matrix is closer to singularity). When $\log(\lambda_{\min})$ approaches zero, signifying closeness to an identity matrix, Broyden's method emerges as the most cost-effective option among the three methods, as depicted in the insets of each subplot.[14]

In conclusion, our experimental findings indicate that the performance advantages of the three algorithms hinge on both the dimensionality and singularity characteristics of the correlation matrix. Given the necessity to generate a substantial number of correlation matrices within the particle-filter-based method, with some matrices possessing small $\lambda_{\min}$, we determine that Broyden's method generally surpasses the other two options. Consequently, it has been selected for the inversion of GFT in both simulation and empirical studies. While

[13] The range of $\log(\lambda_{\min})$ is set to $[-20, 0]$, which is partitioned into 50 groups. For each group, we calculate the average CPU time for all matrices belonging to the same group.

[14] This trend holds except in scenarios where $\log(\lambda_{\min})$ is extremely close to zero, a scenario which is empirically not relevant.

it is important to note that Broyden's method might become less advantageous compared to AH's algorithm in scenarios with very high p, it is crucial to acknowledge that attempting to directly estimate the MSV model with a large p remains unfeasible regardless of the algorithm utilized. For moderate values of p, our analysis reveals that Broyden's method can potentially provide a substantial time-saving advantage, reducing computational time by approximately 85% in comparison to AH's method.

5 Simulation Studies

To investigate the performance of our estimation procedure, we conduct some simulation exercises in this section. The design of our experiment is frequentist in nature, as we fix the parameters at their true values and generate data from the same data generating process with 1000 replications. We use the posterior mean as a point estimator for all the parameters. Since the true values are known, we are thus able to calculate estimation bias (defined as the difference between the true values and the average value of the posterior means) and the standard deviation.[15]

For the purpose of evaluating the sampling efficiency of PGAS algorithm, following Kim et al. (1998), we calculate the average inefficiency factor (IF), which is defined as the variance of sample mean from MCMC sampling divided by that from a hypothetical sampler which draws independent samples. The variance of MCMC sample mean is the square of numerical standard error estimated by

$$NSE = 1 + \frac{2B_M}{B_M - 1} \sum_{i=1}^{B_M} K\left(\frac{i}{B_M}\right) \hat{\rho}(i),$$

where $\hat{\rho}(i)$ is estimated autocorrelation at lag i, B_M is the bandwidth and $K(\cdot)$ is the Parzen kernel. We choose the bandwidth B_M to be 1000. A smaller IF indicates a better mixing of the Markov chain and thereby a higher sampling efficiency.

Our data generating process is the basic MSV-GFT model with $p = 4$. There are 24 parameters in the model, whose true values are given by:

1. $\mu_{h1} = \mu_{h2} = \mu_{h3} = \mu_{h4} = 0.3$ and $\mu_{q1} = \mu_{q2} = \mu_{q3} = \mu_{q4} = 0.7$,
2. $\phi_{h1} = \phi_{h2} = \phi_{h3} = \phi_{h4} = 0.9$ and $\phi_{q1} = \phi_{q2} = \phi_{q3} = \phi_{q4} = 0.8$,
3. $\sigma^2_{h1} = \sigma^2_{h2} = \sigma^2_{h3} = \sigma^2_{h4} = 0.05$ and $\sigma^2_{q1} = \sigma^2_{q2} = \sigma^2_{q3} = \sigma^2_{q4} = 0.05$.

[15] Here, the standard deviation refers to the variation across replications, rather than the numerical standard error of MCMC sampler introduced below.

All the simulation results reported in this section are based on 5000 MCMC iterations, among which the first 1000 samples are discarded as burn-in period.[16] We consider three different sample sizes, namely $T = 500, 1000, 2000$, as well as three numbers of particles, namely $N = 50, 100, 200$. It is worthwhile to mention that, the simulated data used across different particle numbers for given sample size are the same, while it changes when the sample size increases. To save the space, we only report the results for h_1 and q_1. The results for other latent processes are similar and hence omitted.

Table 1 reports the average values of the posterior means, standard deviations and IFs across replications of μ_{h1}, ϕ_{h1}, σ^2_{h1}, μ_{q1}, ϕ_{q1} and σ^2_{q1}. It can be seen that even for a small sample size (such as 500) and a relatively small number of particles (such as 50), the posterior means for both μ_{h1} and μ_{q1} are reasonably close to their respective true values, although there is an downward bias for both μ_{h1} and μ_{q1}. Nevertheless, it can be seen that the bias shrinks towards zero when T expands. As expected, the standard deviations for both μ_{h1} and μ_{q1} substantially decrease as T increases while an increasing number of particles has no effect in this regard.

Meanwhile, the persistence parameters ϕ_{h1} and ϕ_{q1} can be estimated accurately, even with 500 observations and 50 particles. The estimates have very small biases and low standard deviations. With 200 particles, the bias almost completely vanishes. Substantial downward biases are observed for σ^2_{h1} and σ^2_{q1} when sample size is 500. This bias is insensitive to the number of particles. Fortunately, it can be improved when more observations are available. Indeed, we observe that if $T = 2000$, the bias becomes much smaller for σ^2_{h1} and completely vanishes for σ^2_{q1}.

Finally, the IF varies little as we change the sample size, but improves when the number of particles increases. Consistent with earlier studies, the IF is the lowest for μ's and the highest for σ^2's. Compared with the traditional single-move or multi-move Gibbs sampler (for example, see Kim et al. (1998)), our new PGAS sampler enjoys a much better mixing property. In summary, the simulation results confirm that our chosen approach works well for the model considered in our study. In light of the good performance, 200 particles are used for the empirical applications reported later.

To offer an overview of the computational demands of our model and the proposed estimation strategy, the final column in Table 1 presents the computational costs (measured in CPU hours) associated with our inference procedure across different combinations of (T, N). This CPU time estimation is based on running 5000 MCMC iterations in MATLAB R2023b on a desktop computer featuring an AMD Ryzen 9 7950X 16-Core Processor running at 4.50 GHz. Evidently, the computational burden increases linearly with both the sample

[16] Examination of the autocorrelation function suggests that MCMC well converges after 1000 iterations.

size and the number of particles. For instance, with 2000 observations and 200 particles, the complete in-sample analysis requires approximately 8.5 hours to execute.

6 Model Extensions with Leverage Effects and Realized Measures

6.1 MSVL-GFT model and its Bayesian inference

To incorporate asymmetric effect, we assume that

$$\begin{pmatrix} \epsilon_t \\ \eta_{ht} \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} R_t & R_t^{\frac{1}{2}} \Omega \Sigma_h^{\frac{1}{2}} \\ \Sigma_h^{\frac{1}{2}} \Omega R_t^{\frac{1}{2}} & \Sigma_h \end{pmatrix} \right), \tag{3g}$$

where $\Omega = diag(\rho)$ and $\rho = (\rho_1, ..., \rho_p)'$. The model defined by equations (3a)-(3g) is referred to as MSVL-GFT.

The Bayesian analysis for MSVL-GFT is a simple extension to that for MSV-GFT with additional p parameters characterizing the leverage effect ρ. Following Yu (2005), we set the prior distributions for these extra parameters as $\rho_i \sim U(-1, 1)$. Redefining $\theta = (\mu_h, \mu_q, \phi_h, \phi_q, \sigma_h^2, \sigma_q^2, \rho, \gamma_\mu)'$, we can make inference for MSVL-GFT model by implementing a Gibbs sampler as in Section 3.3 with an additional block drawing $\rho | r, h, \theta_{/(\rho)}$.

6.2 RMSV(L)-GFT model and its Bayesian inference

When realized measures for the latent process h_t and q_t are available, we expect that incorporating them into our baseline MSV-GFT model may improve its empirical performance. Specifically, we assume that the researchers have access to the $p \times p$ realized covariance matrices (denoted by C_t^r) computed from intra-day high-frequency returns. Let

$$C_t^r = (V_t^r)^{1/2} R_t^r (V_t^r)^{1/2}, \tag{5}$$

where the superscripts denote realized measures. V_t^r is a diagonal matrix collecting all the realized variances and R_t^r is the realized correlation matrix. Equation (5) is the realized version of traditional variance-correlation decomposition. Since the latent variables in MSV-GFT are the transformation of the original variances and the correlation coefficients, we apply the same transformation to the realized covariance. Specifically, we define

$$\begin{aligned} h_t^r &= (h_{1t}^r, ..., h_{pt}^r)' = diag(\log(V_t^r)^{1/2}), \\ q_t^r &= (q_{1t}^r, ..., q_{dt}^r)' = F(R_t^r). \end{aligned} \tag{6}$$

It can be seen that h_t^r and q_t^r are the observed empirical measures of the latent variables, h_t and q_t, respectively. We hence expect that time variation in these realized measures contains information about the dynamics of the corresponding latent variables.

As is well known in the literature, there may exist non-trivial measurement errors in h_t^r and q_t^r, because neither are perfect measurements of the latent variables due to the microstructure noise, nontrading hours, nonsynchronous trading, and so forth. Bearing this in mind, we model the relationship between the latent variables and their realized counterparts by

$$h_t^r = \psi_h + h_t + \xi_{ht},\ \xi_{ht} \sim N(0, \Sigma_h^r), \tag{7}$$

$$q_t^r = \psi_q + q_t + \xi_{qt},\ \xi_{qt} \sim N(0, \Sigma_q^r), \tag{8}$$

where $\psi_h = (\psi_{h1}, ..., \psi_{hp})'$ and $\psi_q = (\psi_{q1}, ..., \psi_{qd})'$ capture potential approximation errors in the realized measures.[17] We further assume $\Sigma_h^r = diag((\eta_{h1}^2, ...\eta_{hp}^2)')$ and $\Sigma_q^r = diag((\eta_{q1}^2, ..., \eta_{qd}^2)')$. Combining equations (3a)-(3f) with equations (7)-(8), we get the realized MSV-GFT (RMSV-GFT) model. Besides, it may also be desirable to allow leverage effects defined by (3g) in RMSV-GFT. Such a specification is referred to as RMSVL-GFT.

It can be seen that extra measurement equations have been added to the MSV-GFT model. These additional equations are based on the transformation of realized measure and the same transformation applied to the latent covariance matrix. In the literature, it has been shown that the realized volatility converges to the integrated volatility and the same applies to the logarithmic versions. However, Barndorff-Nielsen and Shephard (2002) argued that the approximation of the log integrated volatility by the log realized volatility usually performs better in practice. This property has been used in Hansen and Huang (2016) to introduce a realized EGARCH model and in Phillips and Yu (2009) to construct a two-stage method to estimate continuous time models.

Similar to MSV-GFT model, RMSV(L)-GFT model can be estimated using our PGAS-based MCMC algorithm as well. Details of inference can be found in Section F of Online Supplement. Note that the estimation time of models with the leverage effect and/or realized measures is almost the same as the baseline MSV-GFT model. This is not surprising because the main bulk of the computational cost lies in the inversion of GFT and adding either the leverage effect or realized measures does not change the times or speed of inverting GFT.

[17] Equation (7) has been used for constructing univariate realized stochastic volatility models by, for instance, Takahashi et al. (2009) and Koopman and Scharth (2012). Equation (8) is proposed in Yamauchi and Omori (2020) as a building block of their RMSV model based on the pairwise Fisher transformation.

7 Empirical Studies

In this section, we conduct an empirical analysis of the proposed models and compare them with several existing competitors, evaluating the performance of alternative MSV models through both in-sample fit and out-of-sample forecasting.

7.1 Data description

Our analysis focuses on daily close-to-close log-returns of six stocks, namely, JPMorgan Chase & Co (JPM), Goldman Sachs Group Inc (GS), Honeywell International Inc (HON), Caterpillar Inc (CAT), Johnson & Johnson (JNJ) and Amgen Inc (AMGN).[18] Our full sample period is from January 3, 2006 to December 31, 2015, covering 2516 trading days. The log-return sequences are plotted in Figure 3 as the red dashed line. Panel (a) of Table 2 presents a set of summary statistics, as well as the pairwise sample correlations. It can be observed that the returns exhibit strong positive co-movement and the degree of co-movement varies across different pairs, with the correlation coefficient ranging from 0.32 to 0.74.[19]

For each stocks, intraday transaction data were obtained from the TAQ database, which were cleaned using the method suggested in Barndorff-Nielsen et al. (2009). From the high-frequency data, we compute the realized kernel estimates of the 6×6 integrated covariance for each trading day. From the realized covariance, we obtain h_t^r and q_t^r by applying the logarithm transformation and GFT, respectively. The daily annualized realized volatility sequences are plotted in Figure 3 as the blue solid line. Time series of the realized correlations are presented in Figure 4 with the red dotted line depicting the sample average and the blue dashed line showing the corresponding correlations computed using daily data. Note that both the level and the persistence of these sequences exhibit some heterogeneity. Panel (b) of Table 2 presents a set of summary statistics for each realized volatility sequence, together with the time series average of all realized correlations. An interesting finding from this table is that the level of co-movement implied by the realized correlations is about half of that suggested by the daily returns, which can be visualized in Figure 4 as well.

7.2 Specifications of competing models

For comparison, we consider the following three categories of model specifications:

1. MSV models

[18] The data for daily returns were obtained from Yahoo Finance at https://finance.yahoo.com/.

[19] As suggested by a referee, it might be instructive to estimate and analyze alternative sets of variables with different dynamic correlation structures to demonstrate the robustness of empirical results. Due to the huge computational cost of our analysis, this exercise will be left for future research.

(a) MSV-CC defined by (3a)-(3d) and $R_t = R$ for all t.

(b) MSV-GFT.

(c) MSV-Chol. This is a model based on Cholesky decomposition proposed by Lopes et al. (2010).[20]

(d) MSV-DCC. This is the model proposed in Asai and McAleer (2009b), where a DCC-type structure with a Wishart transition dynamics is used to characterize the movement of the correlation matrix.

2. MSV models with the leverage effect

(a) MSVL-CC. This is the MSV-CC model with the leverage effect defined in (3g).

(b) MSVL-GFT.

(c) MSVL-Chol. This is the MSV-Chol model with the leverage effect proposed in Shirota et al. (2017).[21]

(d) MSVL-DCC. This is the MSV-DCC model with additional assumption (3g).

3. MSV models that incorporate the realized measures.

(a) RMSV-CC. This is the MSV-CC model with additional assumption (7).

(b) RMSVL-CC. This is the MSVL-CC model with additional assumption (7).

(c) RMSV-GFT.

(d) RMSVL-GFT.

It is important to note that all candidate models except for those based on the Cholesky decomposition share the same parametrization of the volatility dynamics. The key difference among these models is in the way how the correlation dynamics is specified. To facilitate the comparison, all models are analyzed under the Bayesian framework and estimated by MCMC. In particular, models based on the DCC specification are estimated by a single-move sampler following Asai and McAleer (2009a), while the estimation of MSV(L)-Chol is conducted using the PGAS algorithm introduced in Section 3.2 with 100 particles.

[20] When estimating the MSV-Chol model, we arbitrarily choose the order of JPM, GS, HON, CAT, JNJ and AMGN. The results may be sensitive to ordering; see Arias et al. (2023) for detailed discussions. For the application considered here, there is no reason to prefer one particular ordering a priori. Ideally, one could try all possible orderings (in total $6! = 720$) and pick the best one. This strategy is practically infeasible.

[21] Note that for the MSVL-Chol model, the leverage effect is defined as the negative correlation between the innovation to returns and that to the diagonal elements of the Cholesky decomposition. This is different from the definition in all other MSV models that we consider.

7.3 In-sample analysis

We first consider the full sample analysis using all available data. We draw 20000 MCMC samples and discard first 2000 as the burn-in period. The number of particles used is set to 200. Before reporting in-sample estimation results, we first examine the validity of the independent Gaussian assumption that we made for the elements of q_t in (3e). Though q_t is not observed, we can take advantage of the corresponding realized measure as a reasonable proxy. Following Archakov et al. (2024a), we depict the Q-Q plots for the transformed realized correlations, q_t^r, in Figure 5. These plots indicate that all GFT-transformed realized correlations can be well modelled by Gaussian distributions, corroborating the discovery in Archakov et al. (2024a) and the specification outlined in equation (3e). To further support the use of GTF, in Figure 6, we present the contour plots for selected pairs of realized correlations, together with the bivariate Fisher-transformed and generalized Fisher-transformed counterparts. For three subplots in the first row, ρ_{ij} denotes the realized correlation between asset i and j, with $i = 1, \cdots, 6$ corresponds to {JPM, GS, HON, CAT, JNJ, AMGN}. For three subplots in the second row, g_{ij} denotes the bivariate Fisher-transformation, defined by (1), of ρ_{ij}. For three subplots in the third row, q_i denotes the i^{th} variable generated by applying GFT on 6×6 realized correlation matrices. An noteworthy finding from Figure 6 is that GFT produces variables much less correlated than both realized correlations and their bivariate Fisher-transformed counterparts. This result provides further evidence supporting our assumption of independence when specifying shocks to elements of q_t. In MSV models proposed in Yamauchi and Omori (2020), all pairwise Fisher-transformed variables are assumed to be generated by independent random walks.[22] Figure 6 questions the validity of such an assumption and suggests that modeling GFT-transformed variables independently aligns more with reality, at least for equity returns.[23]

We now report the parameter estimates for the RMSVL-GFT model.[24] Table 3 presents the posterior statistics of parameters related to log volatility sequences h_t. The statistics we consider include the posterior means (in the first row), the posterior standard deviations (in the parenthesis), and the 95% credible intervals (in the fourth row). Also reported are

[22] A subtle point is that the dependence among these variables is in fact implicitly introduced in their estimation procedure. Note that the single-move algorithm they propose is based on conditional restrictions so that the range of a correlation is determined by all other correlations.

[23] As suggested by a referee, in Section G of Online Supplement, we make a detailed comparison between our GFT-based specification and correlation modeling based on pairwise Fisher transformation. We find that the former is significantly superior, both in-sample and out-of-sample.

[24] For $p = 6$, RMSVL-GFT has 21 latent variables, including 6 log volatilities and 15 GFT-transformed correlations. Each latent variable induces 3 parameters in autoregressive dynamics and 2 in measurement equation of realized measure. Along with 6 coefficients for leverage effect, the total number of model parameters is 111.

the IFs (in the third row). Our observations reveal that the posterior means and standard deviations of all parameters linked to volatility dynamics closely align with existing literature. In particular, all log volatility sequences have a very high level of persistence, with the autoregressive root of ϕ_h close to but smaller than 1. The IFs are relatively small across the board, indicating effective mixing of the MCMC draws. Furthermore, the leverage effect ρ is always significantly negative with a value ranging from -0.1832 to -0.4536. Consistent with findings from prior studies like Koopman and Scharth (2012) and Yamauchi and Omori (2020), the bias parameters ψ_h is consistently significantly negative. This is well anticipated as realized volatility contains the information during market trading hours and thereby accounts for only a fraction of variation of close-to-close returns.

Table 4 follows a structure akin to Table 3 but focuses on parameters that characterize transformed correlations q_t. First and foremost, it can be seen that the posterior means of ϕ_q varies a lot, with the minimum value being 0.71 and maximum 0.97. This suggests a great deal of heterogeneity in the level of persistency in the elements of q_t. While the transformed correlation is close to a unit root process in some cases, it can be quite stationary in other cases. Second, we observe that the posterior means of μ_q also differ considerably among q's. Third, the posterior means of σ_q^2 are all significantly different from zero, suggesting the correlation coefficients are time-varying. These findings forcefully highlights the importance of allowing unique dynamics for each correlation sequence. Another notable finding from Table 4 is that the estimated bias parameters ψ_q are always significantly negative, implying that q_t^r is a biased version of underlying transformed correlation q_t and on average suffers from underestimation. This finding aligns with the results in Yamauchi and Omori (2020), where a negative bias in Fisher-transformed realized pairwise correlations is noted. It is worth noting that this bias aligns with the trend of smaller average realized correlations compared to the daily sample correlations as observed in Table 2 and Figure 4.

An interesting observation from parameter estimation is that the magnitude of σ_q^2 is quite small, suggesting that q_t is closer to be deterministic. This finding aligns with the result in Yamauchi and Omori (2020). To explain this phenomenon, note that (7) and (8) in RMSVL-GFT impose a restriction that the unconditional variance of realized measures is equal to that of the latent variables plus the variance of the noise. As a consequence, the values of σ_q^2 are related to the unconditional variance of q_t^r as well as the signal-to-noise ratio of the realized correlations. Figure 7 plots the sequences of posterior means of the selected log volatilities (top panel) and those of the pairwise correlations (bottom panel), both filtered from the RMSVL-GFT model (the red dash-dotted line). Also plotted are the corresponding realized measures with bias ψ_h and ψ_q removed (the blue solid line). The latent variables obtained from the RMSVL-GFT model exhibit a very similar pattern to

their realized counterparts. However, the total variation of correlation is smaller. Moreover, we observe that for correlations, model-implied sequences are much smoother than observed realized measures, which suggests that realized correlations contain a large noise. This is not the case for the realized volatility, whose dynamics can be largely explained by the variation of the latent variable h_t.[25] Such a conclusion can also be obtained if we contrast R-squared of (7) with that of (8). Indeed, for the measurement equations of h_t, we observe that R^2 is higher than 85% in most cases. For q_t, on the other hand, we find it usually less than 30%. Our estimation results therefore highlight the difference in information content between realized volatilities and correlations.

To assess whether the flexibility in MSV-GFT leads to better in-sample statistical performance, we first compare the marginal likelihoods of daily returns, using the approach suggested by Chib (1995). When computing the likelihood ordinate, we use the auxiliary particle filter of Pitt and Shephard (1999). We also compare DIC values of alternative models using DIC_1 of Spiegelhalter et al. (2002); see Li et al. (2020) for discussions why DIC_1 is used for latent variable models. As our main interest lies in the evaluation of relative merits of various model specifications, we only consider candidates that base solely on daily returns in this exercise. The results are presented in Table 5, which also summarizes the number of parameters in each model. Within models without realized measures and leverage effect, our new MSV-GFT model markedly outperforms all other MSV candidates based on both the log marginal likelihood and DIC, providing compelling evidence in favor of the specification based on GFT. Moreover, as expected, the in-sample fitness of MSV-GFT can be further improved by introducing the leverage effect.

7.4 Out-of-sample performance

In this subsection, comparisons of short-term out-of-sample forecasting ability are conducted based on both statistical and economic loss functions. The forecast period is from January 2, 2013 to December 31, 2015, spanning three years and consisting of 755 trading days. For each out-of-sample trading day (say t), we use expanding window approach to re-estimate the model and generate the one-step-ahead forecast of the covariance matrix $\hat{C}_{t|t-1}$ by computing the posterior mean of C_t conditional on all available observations up to period $t-1$.

Our first exercise focuses on the statistical performance reflected by the likelihood of return series. In particular, for each specification and in each out-of-sample trading day, we first evaluate the predictive log-likelihood of daily returns. For trading day t, this quantity

[25] Similar patterns can be found in Figure 2 of Yamauchi and Omori (2020).

is defined as

$$\log p(r_t|r_{1:t-1}),\ t \in \{T_0+1, \cdots, T\},$$

for MSV models without realized measures and

$$\log p(r_t|r_{1:t-1}, h^r_{1:t-1}, q^r_{1:t-1}),\ t \in \{T_0+1, \cdots, T\},$$

for RMSV models, where T_0 denotes the in-sample size and $x_{1:t-1} = (x_1', \cdots, x_{t-1}')'$. We then obtain the average out-of-sample predictive return log-likelihoods of each candidate (R)MSV model. The results covering entire out-of-sample period as well as three sub-years can be found in Table 6, where the improvement of log-likelihood relative to that of MSV-CC are reported. Italic figures in the parenthesis are the corresponding p-values of model confidence set (MCS) of Hansen et al. (2011).[26] MCS_1 are p-values when comparison is confined to models without realized measures and MCS_2 are for the set of all candidates. Underlined values in Table 6 are for the best performing models according to MCS_1, while boldface numbers are for the best performing models according to MCS_2.

As expected, all values reported are positive, indicating a better forecasting ability of dynamic models over the simple constant correlation model. Confining to competitors based on daily returns only, we observe that the GFT-based models dominate except for second sub-year, with or without incorporating the leverage effect. Overall, the DCC specification ranks third, outperformed by MSV(L)-Chol. Another notable finding from Table 6 is that incorporating realized measures into our GFT-based specification leads to additional dramatic improvement of the out-of-sample predictive return likelihood. This can be explained by the fact that taking into account the realized information can significantly stabilize the inference of parameters by reducing the variance of estimators. Overall, our analysis suggests the most flexible RMSVL-GFT model is preferred in terms of its ability in predicting the return distribution.

We then turn to the economic loss function, which is also practically relevant for covariance modeling. We construct the global minimum variance (GMV) portfolio for each model and compare their average squared returns. According to Markowitz (1952), the GMV portfolio is optimal as it has the smallest variance among all portfolios on the efficient frontier. At period $t-1$, we construct the GMV portfolio with the optimal weights $w_t = (w_{1t}, ..., w_{pt})$ where[27]

$$w_t = \frac{\hat{C}^{-1}_{t|t-1}\mathbb{1}_p}{\mathbb{1}_p'\hat{C}^{-1}_{t|t-1}\mathbb{1}_p}, \tag{9}$$

[26] A p-value larger than, say 0.1, indicates that the model belongs to the set of the best performers at the 90% confidence level.

[27] We assume negative weights are allowed so that short-sells are possible.

and the optimal portfolio return at time t is then obtained as $R_t^p = w_t' r_t$.

In addition to the MSV models discussed earlier, we also consider a portfolio with equal weights as a benchmark, which is frequently used in practice. Furthermore, to investigate the relative merits of our parameter-driven MSV models compared with observation-driven models, in our analysis, we consider the DCC model of Engle (2002) and BEKK model of Engle and Kroner (1995). Following a suggestion from a reviewer, we also incorporate into our analysis a simplified version of the GFT-based dynamic conditional score model of Hafner and Wang (2023). In particular, returns are assumed to be conditionally Gaussian, with the conditional volatility of each asset following the GARCH(1,1) process and the log correlation matrix being driven by the conditional scores of the likelihood. Using our notation, the dynamics of transformed correlations in this model can be expressed as

$$q_{t+1} = \omega + A\frac{\partial l_t}{\partial q_t} + Bq_t,$$

where ω is a $d \times 1$ vector, A and B are $d \times d$ diagonal matrix and l_t is the log conditional likelihood of r_t. The detailed expression of $\frac{\partial l_t}{\partial q_t}$ can be found in Theorem 1 of Hafner and Wang (2023). We estimate this model using the two-step approach explained in Section 3.2 of Hafner and Wang (2023). We refer to this model as GARCH-GFT-DCS.

To enable a fair comparison across models, we assume that all stocks have equal expected returns and focus solely on the variance of the portfolio. Specifically, we measure the portfolio variance by computing the average squared return over out-of-sample periods. To check the robustness of our analysis, we again present the analogous results for each of the three years in the out-of-sample period. The results are shown in Table 7, accompanied by corresponding MCS p-values. Similar to Table 6, MCS_1 are p-values when comparison is confined to models without realized measures and MCS_2 are for all candidates. Underlined GMV values identify best models without realized measures according to MCS_1, while boldface numbers identify the best model among all competitors according to MCS_2.

The portfolios based on equal weights and GARCH-type models consistently display notably higher variances in almost all scenarios, making these strategies less favorable options. Interestingly, among the GARCH-type specifications, the GARCH-GFT-DSC model exhibits best performance. Besides, from the insights provided in Table 7, several key conclusions can be drawn. First and foremost, confining to MSV models without leverage and realized measures, we observe that MSV-GFT dominates all other competitors, both for the full out-of-sample period and the two out of three sub-periods. More impressively, for the full period, among all models without realized information and leverage, only the MSV-GFT models belong to the 90% MCS_2. This outcomes strongly underscore the efficacy of utiliz-

ing GFT for dynamic correlation matrix specification. Secondly, our analysis indicates that MSVL models consistently outperform their MSV counterparts in most instances, aligning with the significant impact of the leverage effect observed in-sample. Indeed, we find that the MSVL-GFT model ranks highest according to MCS_1 except for 2014, in which MSVL-DCC is better. Thirdly, the highly flexible RMSVL-GFT model yields the lowest overall average squared return and it also excels in 2015. Following closely are the RMSV-GFT and RMSV(L)-CC models, each incorporating additional measurements derived from realized data. Furthermore, it is worth noting that, for all choices of sample period, models with realized information always belong to the 90% MCS_2.[28] Across all sub-periods, the RMSV models demonstrate a marked advantage over their MSV counterparts. These findings underscore the substantial benefits of integrating both daily and high-frequency data in enhancing asset allocation strategies.

In summary, our analysis indicates that the MSV models utilizing GFT, along with the inclusion of the leverage effect and/or realized measures, offer more dependable results for out-of-sample covariance forecasting and portfolio construction.

8 Conclusion

We present a new approach to modeling multivariate stochastic volatility in this paper. Our approach uses a generalized version of Fisher's z-transformation to dynamically characterize the correlation structure in a highly flexible manner. One key advantage of our model is that it can automatically generate a positive-definite correlation matrix, while also completely separate the driving forces underlying volatilities and correlations. We go a step further by extending the model to incorporate both the leverage effect and the realized measures.

In contrast to numerous existing studies that rely on conventional Bayesian inference methods, we utilize a Gibbs sampler coupled with a particle filter to conduct inference for our model. A novel contribution we make to the literature is our introduction of two alternative algorithms, namely Newton's method and Broyden's method, alongside AH's algorithm to solve systems of nonlinear equations – an essential step in the estimation process. We conduct experiments to assess the performance of these three algorithms and advocate for the practical application of Broyden's method based on our findings. We showcase the efficacy of our estimation approach for the model. Overall, our proposed model stands out as a potent and versatile tool for capturing the intricate dynamics of multivariate stochastic volatility within financial markets.

[28] Note that if we only consider the universe of four RMSV models, then RMSV(L)-CC will not belong to 90% MCS.

Our empirical results (with $p = 6$) highlight that this adaptable approach to modeling multivariate stochastic volatility enhances the in-sample fitting to stock return volatilities. Furthermore, by integrating the leverage effect and realized measures into the updated model specification, we observe improvements in both in-sample and out-of-sample forecasting accuracies compared to numerous existing models.

While a single digit for the dimension of assets appears a restriction, estimation of a low-dimensional MSV model can help understand important features in data and hence, provide guidance to choose more restrictive MSV models for high dimensional data. For example, after we estimate our six-dimensional MSV model, we find that it is critical to allow the pairwise correlation coefficient sequences to have different levels of persistence. A reasonable restrictive MSV model for high-dimensional data must retain this feature.

An unrestrictive MSV model with p being hundreds or even thousands of assets will impose a significant computational challenge because the computational burden of implementing the model increases exponentially with the number of assets. In such scenarios, dimension reduction strategies such as enforcing a block structure or introducing latent common factors become necessary. There are two potential factor-based specifications that one may consider.

The first one assumes

$$r_t = \Lambda f_t + \Omega^{1/2} e_t, \tag{10}$$

where Λ is a $p \times K$ matrix of factor loadings with $K << p$, f_t is a K-dimensional vector of factors that is assumed to follow Model (3a)-(3f), Ω is a $p \times p$ (possibly diagonal) covariance matrix, and e_t is a vector of p independent standard normal variates without serial dependence. The second one imposes a factor structure directly on q_t, that is,

$$q_t = \Lambda f_t, \tag{11}$$

where Λ is a $p(p-1)/2 \times K$ loading matrix with $K << p(p-1)/2$ and f_t is a K-dimensional vector of factors. We may then specify a new model by adding equation (11) to (3a)-(3f) and replace q_t in (3e) by f_t. In the Section H of Online Supplement, we discuss these specifications in greater details.

It is well known in the literature that even when p is in the range of hundreds or even thousands, the number of volatility factors (i.e., K in equation (10) and (11)) is always a low single digit (see for example, Ding et al. (2025) and references therein). As a result, we expect Broyden's method and hence, our estimation method may continue to be useful. We defer the exploration of how to estimate high-dimensional MSV models and the investigation of the relative performance of alternative models to future endeavors.

References

Andersen, T. G., T. Bollerslev, P. Frederiksen, and M. Ørregaard Nielsen (2010): "Continuous-time models, realized volatilities, and testable distributional implications for daily stock returns," Journal of Applied Econometrics, 25, 233–261.

Andrieu, C., A. Doucet, and R. Holenstein (2010): "Particle markov chain monte carlo methods," Journal of the Royal Statistical Society: Series B, 72, 269–342.

Archakov, I. and P. R. Hansen (2021): "A new parametrization of correlation matrices," Econometrica, 89, 1699–1715.

——— (2024): "A canonical representation of block matrices with applications to covariance and correlation matrices," Review of Economics and Statistics, 1–39.

Archakov, I., P. R. Hansen, and A. Lunde (2024a): "A Multivariate Realized GARCH Model," preprint.

Archakov, I., P. R. Hansen, and Y. Luo (2024b): "A new method for generating random correlation matrices," The Econometrics Journal, 27, 188–212.

Arias, J. E., J. F. Rubio-Ramirez, and M. Shin (2023): "Macroeconomic forecasting and variable ordering in multivariate stochastic volatility models," Journal of Econometrics, 235, 1054–1086.

Arulampalam, M. S., S. Maskell, N. Gordon, and T. Clapp (2002): "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," IEEE Transactions on Signal Processing, 50, 174–188.

Asai, M., C.-L. Chang, and M. McAleer (2017): "Realized stochastic volatility with general asymmetry and long memory," Journal of Econometrics, 199, 202–212.

——— (2022): "Realized matrix-exponential stochastic volatility with asymmetry, long memory and higher-moment spillovers," Journal of Econometrics, 227, 285–304.

Asai, M. and M. McAleer (2006): "Asymmetric multivariate stochastic volatility," Econometric Reviews, 25, 453–473.

——— (2009a): "Multivariate stochastic volatility, leverage and news impact surfaces," The Econometrics Journal, 12, 292–309.

——— (2009b): "The structure of dynamic correlations in multivariate stochastic volatility models," Journal of Econometrics, 150, 182–192.

Asai, M., M. McAleer, and J. Yu (2006): "Multivariate stochastic volatility: a review," Econometric Reviews, 25, 145–175.

Barndorff-Nielsen, O., P. R. Hansen, A. Lunde, and N. Shephard (2009): "Realized kernels in practice: trades and quotes," The Econometrics Journal, C1–C32.

Barndorff-Nielsen, O. E. and N. Shephard (2002): "Econometric analysis of realized volatility and its use in estimating stochastic volatility models," Journal of the Royal Statistical Society: Series B, 64, 253–280.

Bauwens, L., S. Laurent, and J. V. Rombouts (2006): "Multivariate GARCH models: a survey," Journal of Applied Econometrics, 21, 79–109.

Bucci, A., L. Ippoliti, and P. Valentini (2022): "Comparing unconstrained parametrization methods for return covariance matrix prediction," Statistics and Computing, 32, 1–20.

Chan, D., R. Kohn, and C. Kirby (2006): "Multivariate stochastic volatility models with correlated errors," Econometric Reviews, 25, 245–274.

Chib, S. (1995): "Marginal likelihood from the Gibbs output," Journal of the American Statistical Association, 90, 1313–1321.

Chib, S., Y. Omori, and M. Asai (2009): "Multivariate stochastic volatility," in Handbook of Financial Time Series, Springer, 365–400.

Chopin, N. and S. S. Singh (2015): "On particle Gibbs sampling," Bernoulli, 21, 1855–1883.

De Jong, P. and N. Shephard (1995): "The simulation smoother for time series models," Biometrika, 82, 339–350.

Dellaportas, P., M. K. Titsias, K. Petrova, and A. Plataniotis (2023): "Scalable inference for a full multivariate stochastic volatility model," Journal of Econometrics, 232, 501–520.

Ding, Y., R. Engle, Y. Li, and X. Zheng (2025): "Multiplicative factor model for volatility," Journal of Econometrics, 249, 105959.

Engle, R. (2002): "Dynamic conditional correlation: A simple class of multivariate generalized autoregressive conditional heteroskedasticity models," Journal of Business & Economic Statistics, 20, 339–350.

Engle, R. and B. Kelly (2012): "Dynamic equicorrelation," Journal of Business & Economic Statistics, 30, 212–228.

Engle, R. and K. Kroner (1995): "Multivariate Simultaneous Generalized ARCH," Econometric Theory, 11, 122–150.

Engle, R. F. (1982): "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation," Econometrica, 987–1007.

Gong, C. and D. S. Stoffer (2021): "A note on efficient fitting of stochastic volatility models," Journal of Time Series Analysis, 42, 186–200.

Gouriéroux, C., J. Jasiak, and R. Sufana (2009): "The Wishart autoregressive process of multivariate stochastic volatility," Journal of Econometrics, 150, 167–181.

Hafner, C. M. and L. Wang (2023): "A dynamic conditional score model for the log correlation matrix," Journal of Econometrics, 237, 105176.

Hansen, P. R. and Z. Huang (2016): "Exponential GARCH modeling with realized measures of volatility," Journal of Business & Economic Statistics, 34, 269–287.

Hansen, P. R., Z. Huang, and H. H. Shek (2012): "Realized GARCH: a joint model for returns and realized measures of volatility," Journal of Applied Econometrics, 27, 877–906.

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

Harvey, A., E. Ruiz, and N. Shephard (1994): "Multivariate stochastic variance models," The Review of Economic Studies, 61, 247–264.

Higham, N. J. (2002): "Computing the nearest correlation matrix—a problem from finance," IMA journal of Numerical Analysis, 22, 329–343.

——— (2008): Functions of matrices: theory and computation, SIAM.

Hurn, S., V. Martin, P. Phillips, and J. Yu (2020): Financial Econometric Modeling, Oxford University Press.

Ishihara, T. and Y. Omori (2012): "Efficient Bayesian estimation of a multivariate stochastic volatility model with cross leverage and heavy-tailed errors," Computational Statistics & Data Analysis, 56, 3674–3689.

Ishihara, T., Y. Omori, and M. Asai (2016): "Matrix exponential stochastic volatility with cross leverage," Computational Statistics & Data Analysis, 100, 331–350.

Johansen, A. M. and A. Doucet (2008): "A note on auxiliary particle filters," Statistics & Probability Letters, 78, 1498–1504.

Kass, R. E. and A. E. Raftery (1995): "Bayes factors," Journal of the American statistical Association, 90, 773–795.

Kim, S., N. Shephard, and S. Chib (1998): "Stochastic volatility: likelihood inference and comparison with ARCH models," The Review of Economic Studies, 65, 361–393.

Koopman, S. J. and M. Scharth (2012): "The analysis of stochastic volatility in the presence of daily realized measures," Journal of Financial Econometrics, 11, 76–115.

Kurose, Y. and Y. Omori (2016): "Dynamic equicorrelation stochastic volatility," Computational Statistics & Data Analysis, 100, 795–813.

——— (2020): "Multiple-block dynamic equicorrelations with realized measures, leverage and endogeneity," Econometrics and Statistics, 13, 46–68.

Li, Y., J. Yu, and T. Zeng (2020): "Deviance information criterion for latent variable models and misspecified models," Journal of Econometrics, 216, 450–493.

Lindsten, F., M. I. Jordan, and T. B. Schön (2014): "Particle Gibbs with ancestor sampling," The Journal of Machine Learning Research, 15, 2145–2184.

Lopes, H. F., R. McCulloch, and R. Tsay (2010): "Cholesky stochastic volatility," Unpublished Technical Report, University of Chicago, Booth Business School, 2.

Markowitz, H. (1952): "Portfolio analysis," Journal of Finance, 8, 77–91.

Nakajima, J. (2017): "Bayesian analysis of multivariate stochastic volatility with skew return distribution," Econometric Reviews, 36, 546–562.

Philipov, A. and M. E. Glickman (2006): "Multivariate stochastic volatility via Wishart processes," Journal of Business & Economic Statistics, 24, 313–328.

Phillips, P. C. and J. Yu (2009): "A two-stage realized volatility approach to estimation of diffusion processes with discrete data," Journal of Econometrics, 150, 139–150.

Pitt, M. K. and N. Shephard (1999): "Filtering via simulation: Auxiliary particle filters," Journal of the American Statistical Association, 94, 590–599.

Shirota, S., Y. Omori, H. F. Lopes, and H. Piao (2017): "Cholesky realized stochastic volatility model," Econometrics and Statistics, 3, 34–59.

Spiegelhalter, D. J., N. G. Best, B. P. Carlin, and A. Van Der Linde (2002): "Bayesian measures of model complexity and fit," Journal of the Royal Statistical Society: Series B, 64, 583–639.

Takahashi, M., Y. Omori, and T. Watanabe (2009): "Estimating stochastic volatility models using daily returns and realized volatility simultaneously," Computational Statistics & Data Analysis, 53, 2404–2426.

Tong, C., P. R. Hansen, and I. Archakov (2024): "Cluster GARCH," arXiv preprint arXiv:2406.06860.

Venter, J. H. and P. J. de Jongh (2014): "Extended stochastic volatility models incorporating realised measures," Computational Statistics and Data Analysis, 76, 687–707.

Watanabe, T. and Y. Omori (2004): "A multi-move sampler for estimating non-Gaussian time series models: Comments on Shephard & Pitt (1997)," Biometrika, 91, 246–248.

Xu, Y. and A. Jasra (2019): "Particle filters for inference of high-dimensional multivariate stochastic volatility models with cross-leverage effects," Foundations of Data Science, 1, 61.

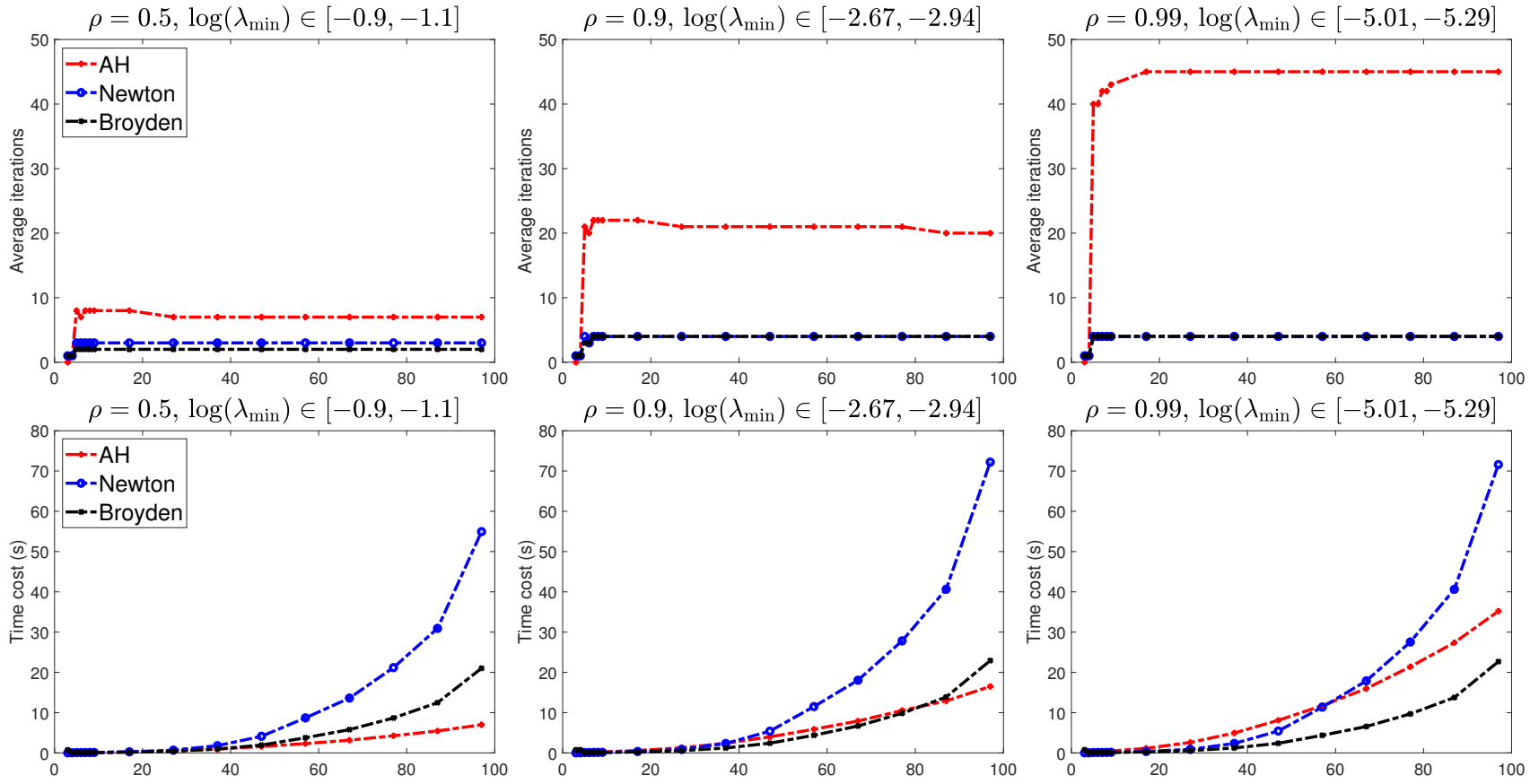
Yamauchi, Y. and Y. Omori (2020): "Multivariate stochastic volatility model with realized volatilities and pairwise realized correlations," Journal of Business & Economic Statistics, 38, 839–855.

Yu, J. (2005): "On leverage in a stochastic volatility model," Journal of Econometrics, 127, 165–178.

Yu, J. and R. Meyer (2006): "Multivariate stochastic volatility models: Bayesian estimation and model comparison," Econometric Reviews, 25, 361–384.

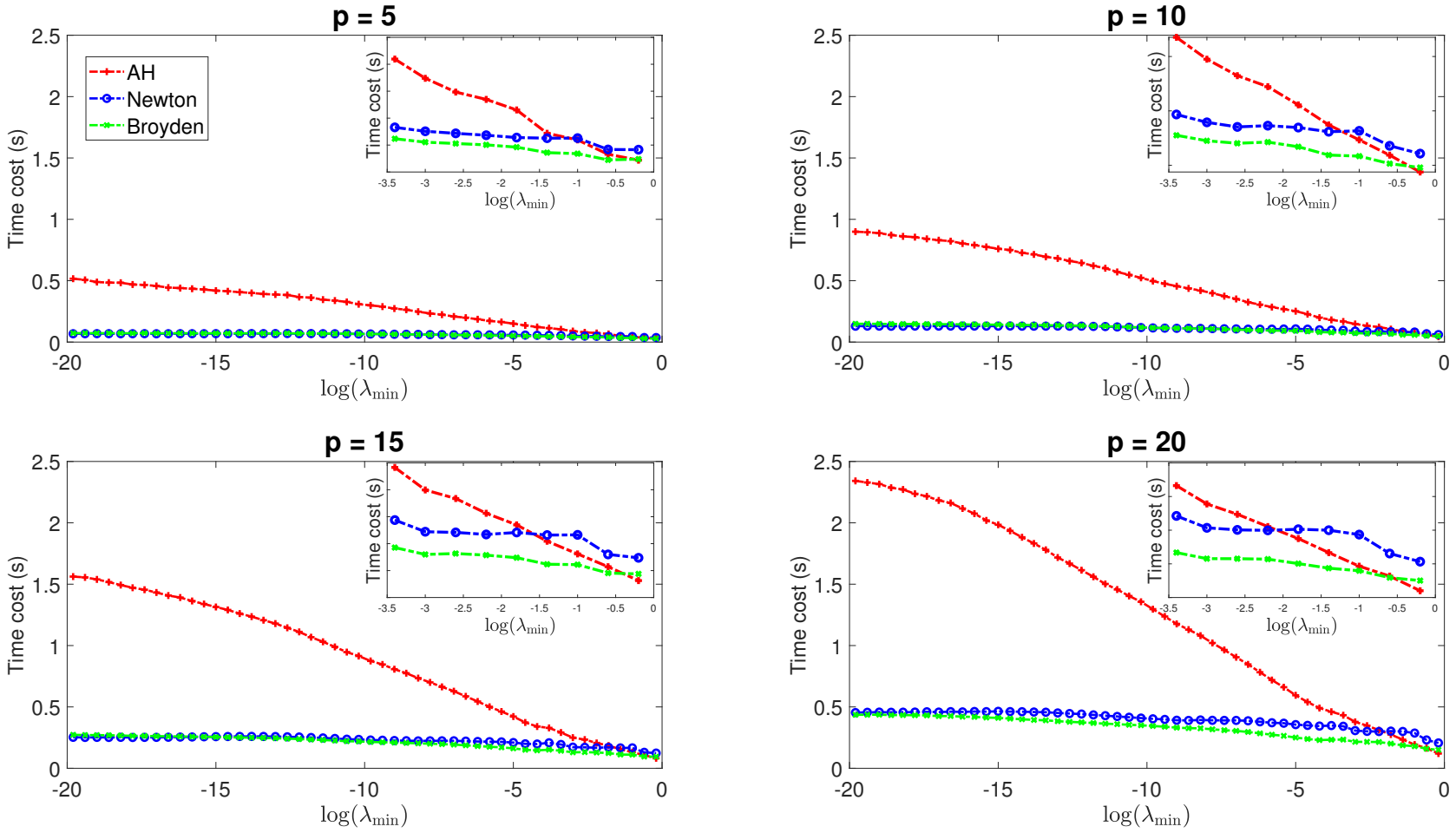
Zaharieva, M. D., M. Trede, and B. Wilfling (2020): "Bayesian semiparametric multivariate stochastic volatility with application," Econometric Reviews, 39, 947–970.

Figure 1: Comparison of alternative root-finding methods based on Toeplitz matrices.



Notes: This figure plots the average required iterations (top panels) and the computational time (bottom panels) for the three algorithms as a function of p. All plots are based on the Toeplitz matrices, which becomes more singular as the value of ρ increases.

Figure 2: Comparison of alternative root-finding methods based on randomly generated matrices.



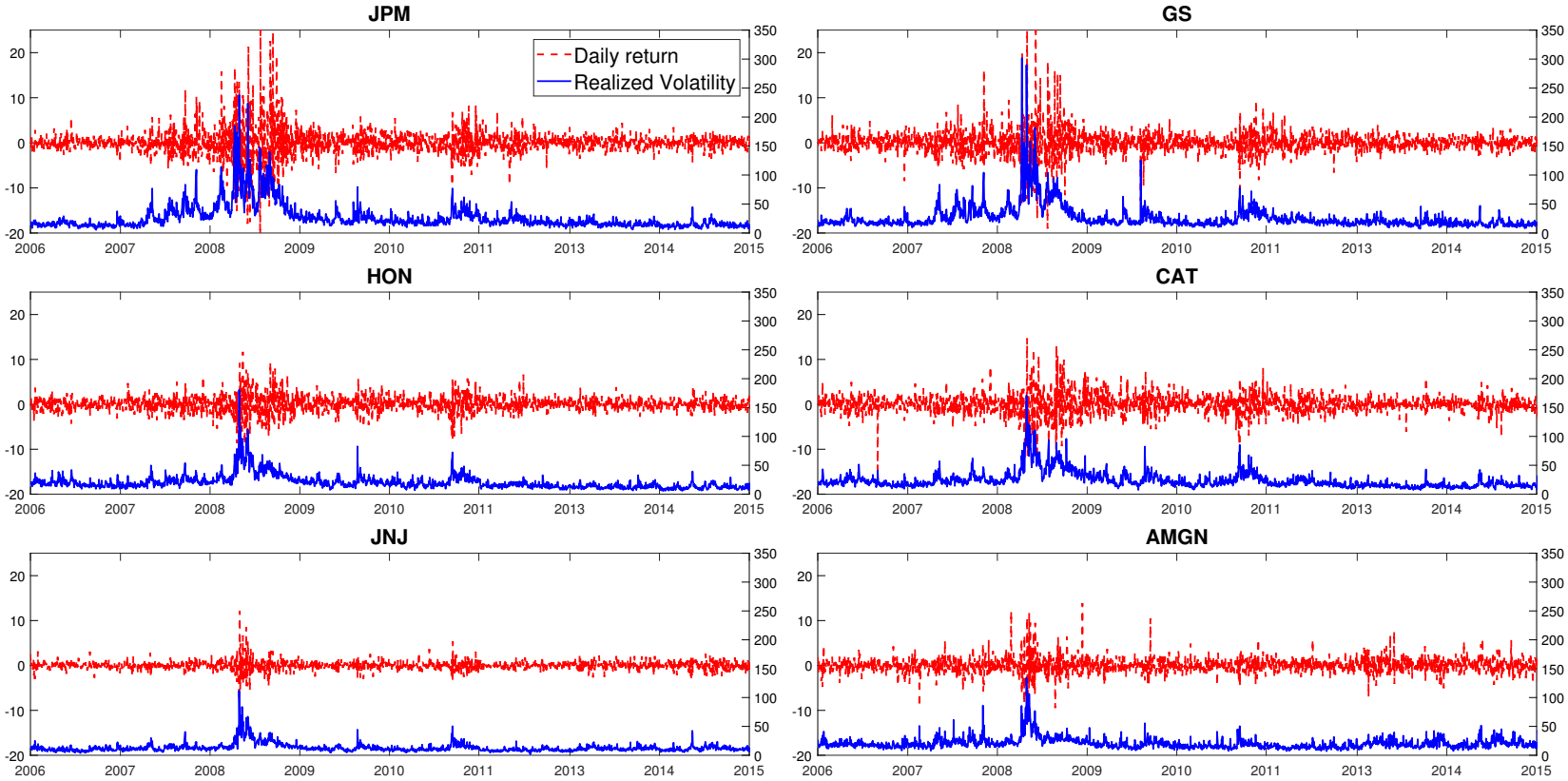
Notes: This figure plots the average computational time for the three algorithms as a function of $\log(\lambda_{\min})$ when $p \in \{5, 10, 15, 20\}$. For each p and $\log(\lambda_{\min})$, a large number of correlation matrices are generated randomly. The inset in each subplot zooms in the area near zero.

Table 1: Simulation results for MSV-GFT with $p = 4$

T	N		Volatility			Correlation			Time cost (h)
			μ_{h1}	ϕ_{h1}	σ_{h1}	μ_{q1}	ϕ_{q1}	σ_{q1}	
		true	0.3	0.9	0.05	0.7	0.8	0.05	
		Mean	0.295	0.899	0.047	0.676	0.811	0.048	
500	50	std	0.141	0.045	0.023	0.085	0.081	0.026	0.49
		IF	22.536	123.418	218.199	33.591	143.971	245.731	
		Mean	0.296	0.900	0.047	0.676	0.811	0.048	
500	100	std	0.142	0.045	0.023	0.085	0.081	0.026	1.00
		IF	11.692	98.971	175.112	21.513	115.681	198.900	
		Mean	0.296	0.900	0.047	0.676	0.811	0.048	
500	200	std	0.142	0.045	0.023	0.084	0.081	0.026	2.09
		IF	6.735	80.288	142.103	15.589	94.680	160.444	
		Mean	0.296	0.900	0.049	0.688	0.803	0.050	
1000	50	std	0.091	0.032	0.017	0.054	0.066	0.021	1.00
		IF	17.709	136.979	219.401	32.800	179.264	279.643	
		Mean	0.296	0.900	0.049	0.688	0.803	0.050	
1000	100	std	0.091	0.032	0.017	0.054	0.067	0.021	2.00
		IF	9.051	107.458	170.971	23.102	147.555	229.836	
		Mean	0.297	0.900	0.049	0.688	0.803	0.050	
1000	200	std	0.091	0.032	0.017	0.055	0.067	0.021	4.25
		IF	5.377	88.112	139.847	17.727	128.492	196.301	
		Mean	0.298	0.901	0.049	0.692	0.800	0.050	
2000	50	std	0.062	0.023	0.012	0.037	0.051	0.016	1.99
		IF	15.464	139.000	216.425	32.151	208.696	303.972	
		Mean	0.298	0.900	0.049	0.693	0.799	0.050	
2000	100	std	0.062	0.023	0.012	0.037	0.051	0.016	4.01
		IF	8.205	106.982	165.828	23.064	174.045	252.155	
		Mean	0.298	0.901	0.049	0.692	0.801	0.050	
2000	200	std	0.062	0.023	0.012	0.037	0.052	0.016	8.51
		IF	4.853	91.032	140.521	18.461	154.004	220.199	

Notes: T is the number of observations for each asset. N is the number of particles used in PGAS. Mean, std and IF are the average value of posterior means, the standard error of the posterior means, and the average inefficiency factor, respectively. All these three statistics are computed across 1000 replications. The computational time is the number of hours for 5000 MCMC iterations in MATLAB R2023b on a desktop computer with an AMD Ryzen 9 7950X 16-Core Processor and 4.50 GHz memory.

Figure 3: Daily return and realized volatility.



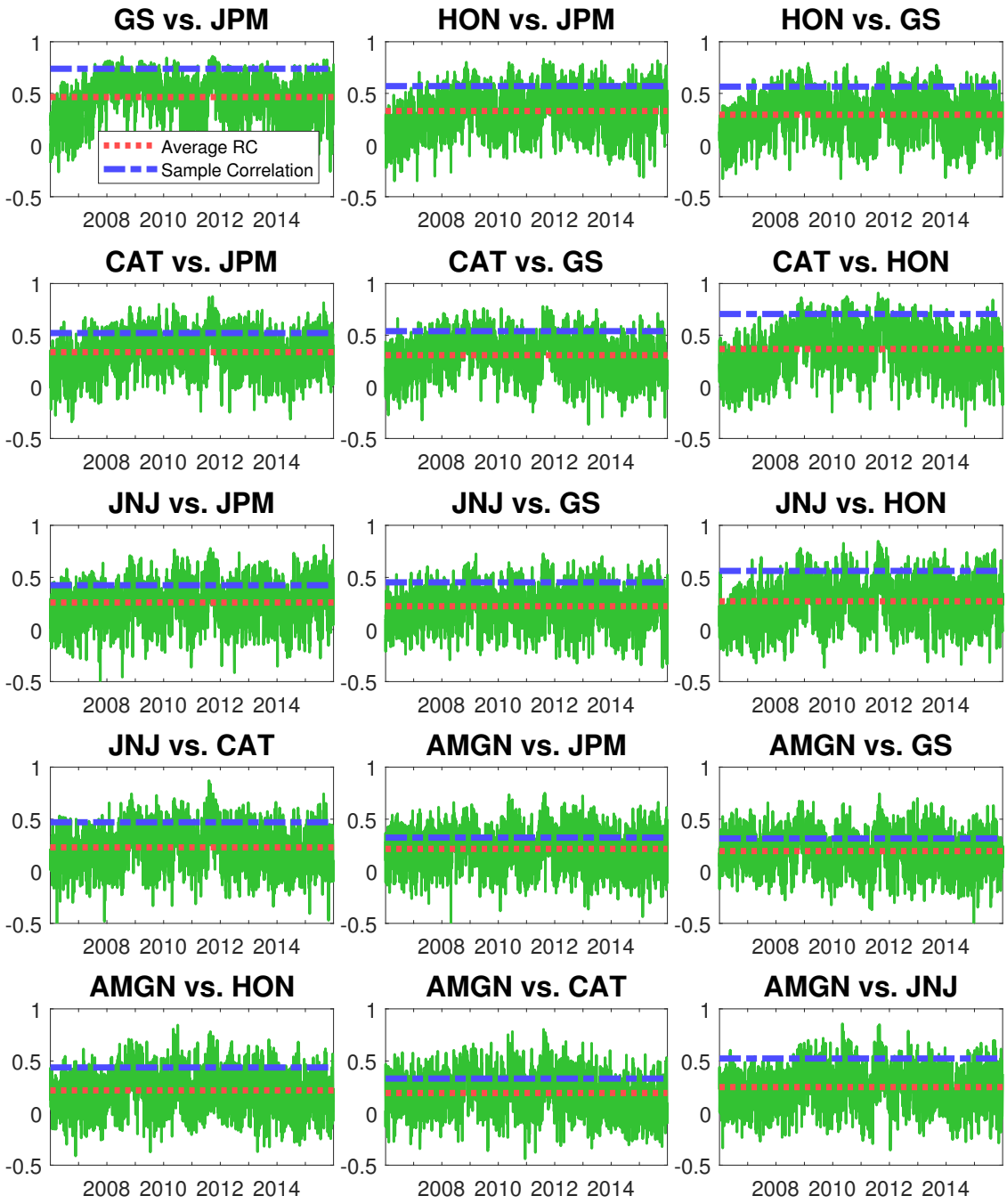
Notes: This figure plots the time series of daily returns (the red dashed line and the left y-axis) and the corresponding daily annualized realized volatilities (the blue solid line and the right y-axis) for six equities (JPM, GS, HON, CAT, JNJ and AMGN) considered in the empirical application. The sample period is from January 3 2006 to December 30 2015.

Table 2: Descriptive statistics for empirical analysis

(a). daily returns						
	JPM	GS	HON	CAT	JNJ	AMGN
Mean	0.0579	0.0456	0.0557	0.0289	0.0260	0.0427
Std	2.7668	2.5390	1.7267	2.0997	1.0265	1.7104
Skewness	1.0053	0.9618	0.0592	0.1055	0.6825	0.6735
Kurtosis	18.8782	20.4724	7.5136	8.7327	17.1011	10.4860
Max	25.0379	26.4218	11.6905	14.6937	12.1892	13.8755
Min	-20.7863	-19.0057	-9.4644	-14.5468	-7.7052	-9.4674
JB	0.001	0.001	0.001	0.001	0.001	0.001
sample correlation	1					
	0.74	1				
	0.58	0.57	1			
	0.52	0.54	0.71	1		
	0.43	0.46	0.56	0.47	1	
	0.33	0.32	0.44	0.33	0.52	1
(b). realized volatilities/correlations						
	JPM	GS	HON	CAT	JNJ	AMGN
Mean	4.4968	4.2611	2.2009	3.0795	0.8862	2.0891
Std	11.5413	13.7358	4.5665	5.5186	1.8443	2.9782
Skewness	8.7142	15.9009	12.3628	7.7342	12.5043	9.6341
Kurtosis	117.8498	350.1991	268.5557	101.7275	249.3983	159.9696
Max	224.8679	361.6185	129.3694	115.8542	49.9017	71.4977
Min	0.1139	0.2076	0.1198	0.1932	0.0112	0.1962
JB	0.001	0.001	0.001	0.001	0.001	0.001
average realized correlations	1					
	0.46	1				
	0.33	0.29	1			
	0.34	0.31	0.37	1		
	0.26	0.22	0.27	0.23	1	
	0.22	0.19	0.22	0.19	0.25	1

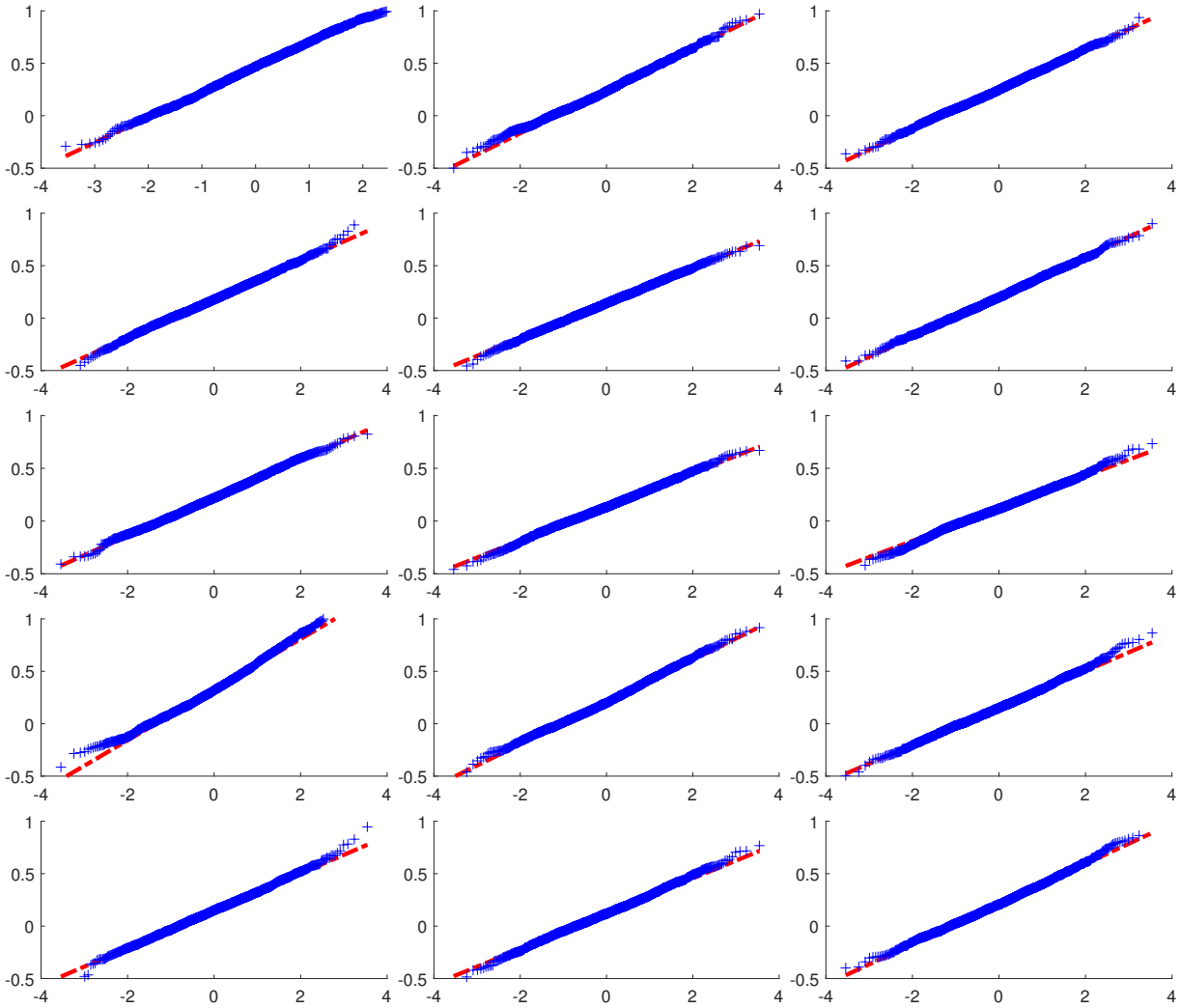
Notes: This table reports the summary statistics for the daily log-returns and realized measures of six equities. The sample period is from January 3, 2006 to December 31, 2015. JB denotes the p-values of Jarque-Bera normality test. For panel (b), the upper part contains the summary statistics for realized volatilities, while the lower part reports the time series average of each realized correlation.

Figure 4: Dynamics of realized correlations.



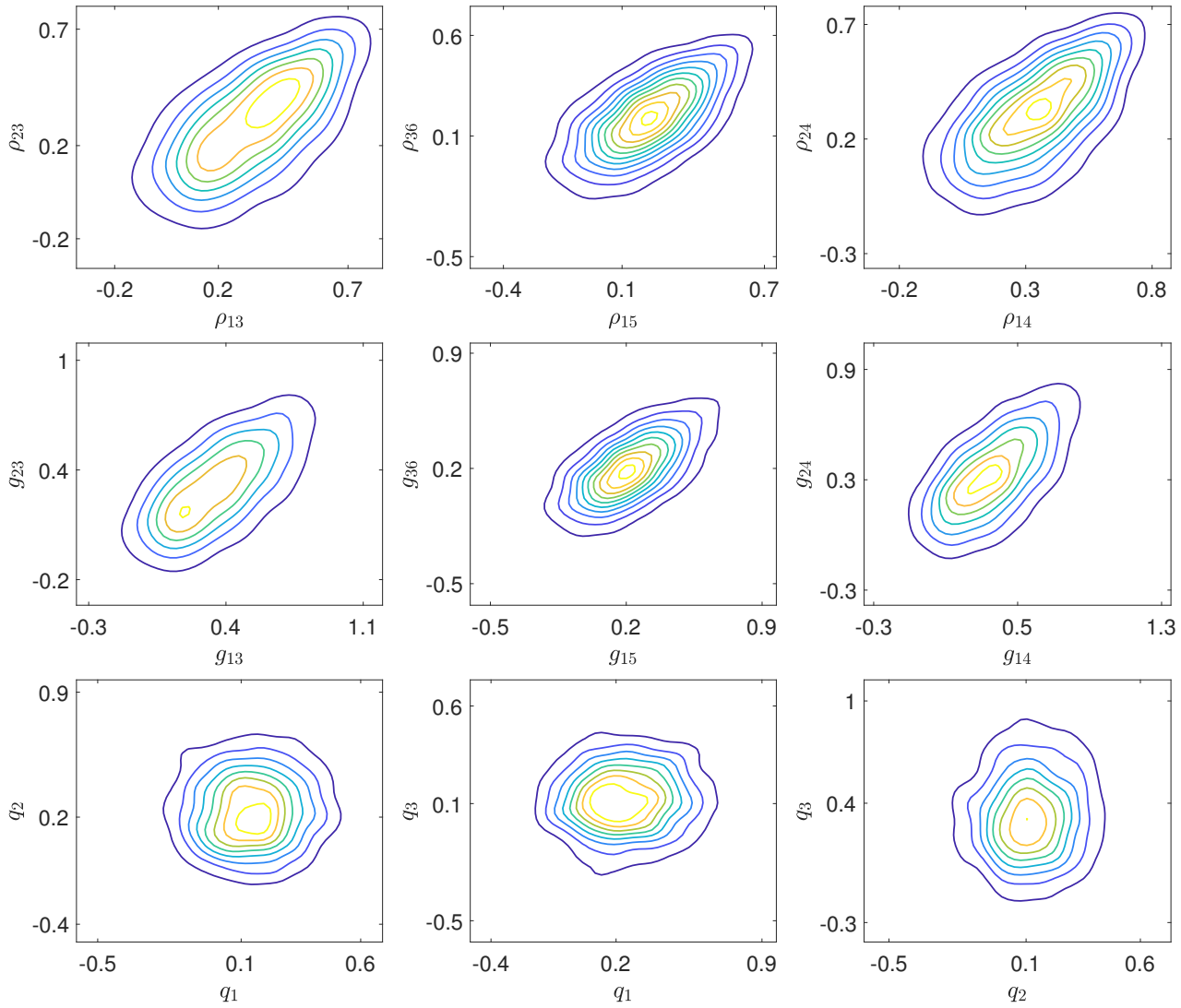
Notes: This figure plots the time series of all pairwise realized correlations considered in the empirical application. The red dotted line depicts the time series average for each realized correlation sequences. The blue dashed line depicts the corresponding sample correlations computed using daily data. The sample period is from January 3 2006 to December 30 2015.

Figure 5: Q-Q plots for transformed realized correlations.



Notes: This figure depicts contour plots of selected realized correlations considered in the empirical application. The results are based on the sample period from January 3 2006 to December 30 2015, which has 2,516 daily observations. The quantiles of their empirical distributions are plotted against the corresponding quantiles of the normal distribution.

Figure 6: Contour plots for selected (transformed) realized correlations.



Notes: This figure depicts contour plots of selected pairs of realized correlations (the first row), bivariate Fisher-transformed counterparts (the second row) and GFT variables (the third row). For subplots in the first row, ρ_{ij} denotes the realized correlation between asset i and j, with $i = 1, \cdots, 6$ corresponds to {JPM, GS, HON, CAT, JNJ, AMGN}. For subplots in the second row, g_{ij} denote the bivariate Fisher-transformation of ρ_{ij}. For subplots in the third row, q_i denote the i^{th} variable generated by applying GFT on 6×6 realized correlation matrices.

Table 3: In-sample volatility estimation results for RMSVL-GFT model

	μ_h	ϕ_h	σ_h^2	ψ_h	η_h^2	ρ
JPM	0.9066	0.9742	0.0624	-0.3367	0.1145	-0.2821
	(0.2035)	(0.0050)	(0.0054)	(0.0301)	(0.0058)	-0.0262
	1.5871	5.1228	22.5061	31.4243	14.0086	13.8807
	[0.5003,1.3053]	[0.9643,0.9836]	[0.0524,0.0736]	[-0.3943,-0.2767]	[0.1033,0.1261]	[-0.3344,-0.2319]
GS	0.9700	0.9723	0.0461	-0.3059	0.1405	-0.3197
	(0.1631)	(0.0053)	(0.0042)	(0.0290)	(0.0060)	(0.0267)
	1.9536	5.0324	17.5200	22.6178	9.5674	16.0523
	[0.6469,1.2880]	[0.9618,0.9822]	[0.0383,0.0546]	[-0.3619,-0.2481]	[0.1291,0.1524]	[-0.3726,-0.2682]
HON	0.5549	0.9705	0.0425	-0.3307	0.1301	-0.4536
	(0.1464)	(0.0056)	(0.0044)	(0.0283)	(0.0060)	(0.0353)
	1.8690	9.2886	35.0743	28.1210	18.1513	33.5945
	[0.2635,0.8474]	[0.9592,0.9812]	[0.0344,0.0516]	[-0.3871,-0.2749]	[0.1185,0.1421]	[-0.5245,-0.3861]
CAT	0.9246	0.9728	0.0387	-0.3421	0.1134	-0.2259
	(0.1539)	(0.0053)	(0.0040)	(0.0292)	(0.0052)	(0.0267)
	2.9799	8.3413	28.8584	36.1329	13.1749	12.7290
	[0.6191,1.2260]	[0.9619,0.9829]	[0.0312,0.0471]	[-0.4013,-0.2859]	[0.1035,0.1240]	[-0.2800,-0.1756]
JNJ	-0.5300	0.9593	0.0414	-0.0598	0.1322	-0.2917
	(0.1066)	(0.0068)	(0.0042)	(0.0310)	(0.0058)	(0.0266)
	4.8144	10.1458	25.9069	34.8798	11.5275	12.9620
	[-0.7402,-0.3186]	[0.9456,0.9719]	[0.0338,0.0502]	[-0.1200,0.0007]	[0.1213,0.1440]	[-0.3449,-0.2399]
AMGN	0.5731	0.9291	0.0566	-0.1813	0.1424	-0.1832
	(0.0739)	(0.0102)	(0.0063)	(0.0297)	(0.0067)	(0.0235)
	5.8587	21.3422	40.1928	26.7798	19.7796	7.9495
	[0.4287,0.7184]	[0.9085,0.9482]	[0.0451,0.0700]	[-0.2389,-0.1235]	[0.1295,0.1559]	[-0.2308,-0.1381]

Notes: This table reports the in-sample volatility-related parameter estimation results based on RMSVL-GFT. The sample period is from January 3, 2006 to December 31, 2015. For each parameter, we report its posterior mean, posterior standard deviation (number in the parenthesis), inefficiency factor, as well as the 95% credible interval using the 2.5th and 97.5th percentiles of the MCMC draws.

Table 4: In-sample correlation estimation results for RMSVL-GFT model

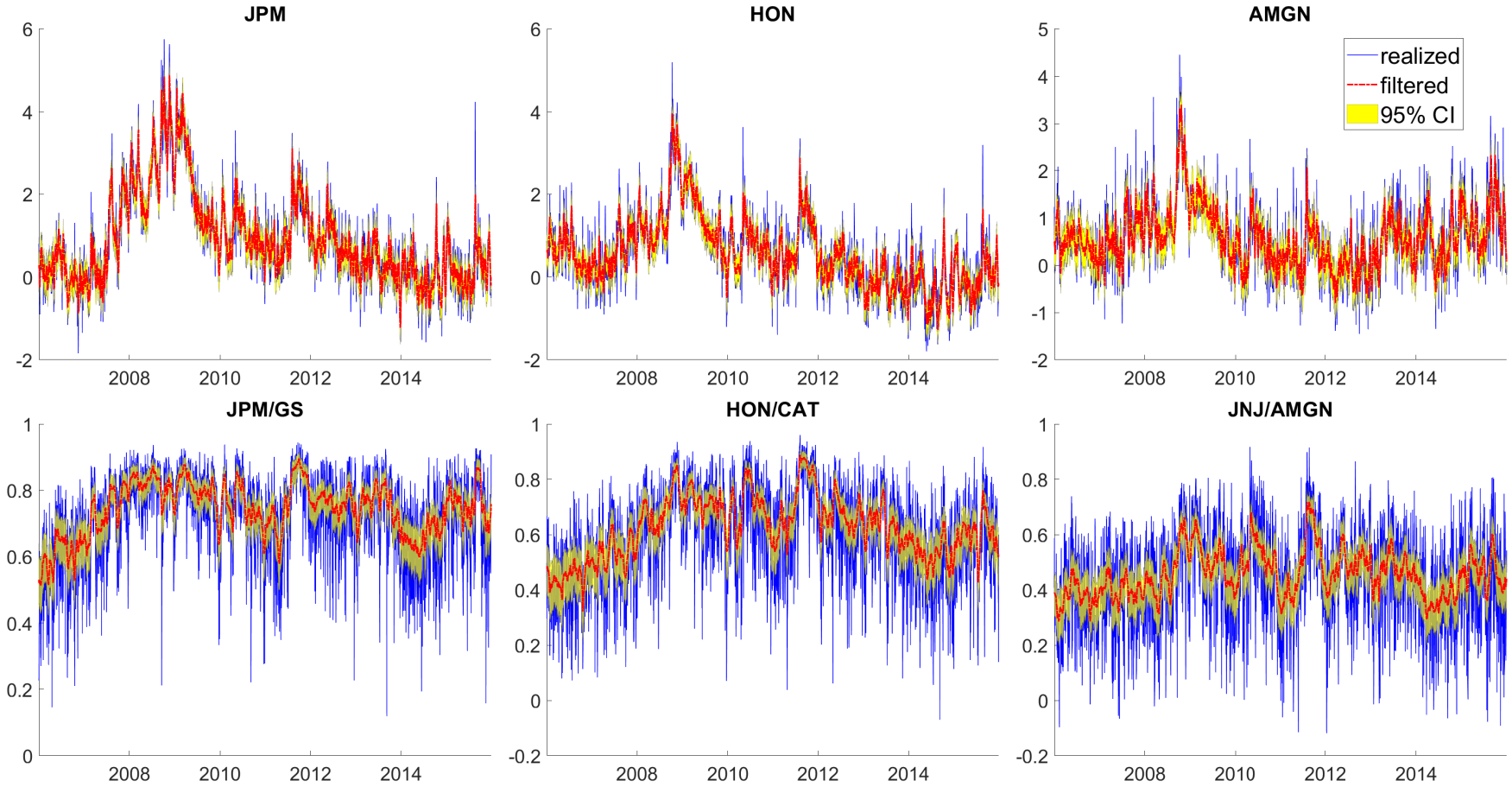
	μ_q	ϕ_q	σ_q^2	ψ_q	η_q^2
q_1	0.8209	0.9718	0.0013	-0.3619	0.0332
	(0.0315)	(0.0068)	(0.0002)	(0.0158)	(0.0011)
	81.2793	135.9620	276.5013	328.3459	38.5393
	[0.7574,0.8813]	[0.9578,0.9843]	[0.0010,0.0018]	[-0.3861,-0.3299]	[0.0312,0.0354]
q_2	0.3870	0.9129	0.0016	-0.1437	0.0297
	(0.0152)	(0.0195)	(0.0004)	(0.0126)	(0.0010)
	201.5734	253.1221	330.0832	301.0712	76.2051
	[0.3607,0.4215]	[0.8710,0.9466]	[0.0009,0.0025]	[-0.1761,-0.1248]	[0.0277,0.0318]
q_3	0.2959	0.9020	0.0017	-0.0427	0.0266
	(0.0126)	(0.0223)	(0.0004)	(0.0096)	(0.0010)
	158.5033	263.6150	328.2505	259.4481	95.9343
	[0.2709,0.3196]	[0.8575,0.9441]	[0.0008,0.0026]	[-0.0605,-0.0242]	[0.0247,0.0285]
q_4	0.3008	0.8813	0.0022	-0.1196	0.0253
	(0.0162)	(0.0236)	(0.0005)	(0.0145)	(0.0010)
	259.8706	247.8345	321.7477	327.5475	121.6425
	[0.2713,0.3323]	[0.8340,0.9237]	[0.0012,0.0031]	[-0.1469,-0.0943]	[0.0235,0.0272]
q_5	0.1696	0.7306	0.0034	-0.0284	0.0226
	(0.0178)	(0.0759)	(0.0012)	(0.0178)	(0.0012)
	325.5295	326.6328	349.2177	336.9915	223.8341
	[0.1372,0.2130]	[0.5655,0.8552]	[0.0017,0.0063]	[-0.0721,0.0037]	[0.0200,0.0247]
q_6	0.3128	0.9625	0.0006	-0.1065	0.0272
	(0.0159)	(0.0088)	(0.0001)	(0.0086)	(0.0008)
	80.5803	171.1830	285.5487	279.1870	18.7110
	[0.2812,0.3437]	[0.9423,0.9773]	[0.0004,0.0009]	[-0.1235,-0.0886]	[0.0257,0.0289]
q_7	0.3211	0.9507	0.0007	-0.0966	0.0263
	(0.0154)	(0.0101)	(0.0001)	(0.0119)	(0.0008)
	164.8006	157.5036	275.1092	319.9081	22.1908
	[0.2888,0.3495]	[0.9285,0.9683]	[0.0005,0.0010]	[-0.1166,-0.0694]	[0.0248,0.0279]
q_8	0.2270	0.7097	0.0032	-0.0899	0.0212
	(0.0111)	(0.0663)	(0.0010)	(0.0108)	(0.0010)
	269.9458	299.4386	325.6539	287.5220	177.0645
	[0.2040,0.2473]	[0.5513,0.8102]	[0.0019,0.0058]	[-0.1088,-0.0672]	[0.0188,0.0230]
q_9	0.1738	0.7293	0.0025	-0.0538	0.0218
	(0.0138)	(0.0759)	(0.0008)	(0.0135)	(0.0009)
	310.3938	324.1396	346.0053	319.2184	133.4191
	[0.1516,0.2020]	[0.5617,0.8583]	[0.0012,0.0042]	[-0.0813,-0.0322]	[0.0201,0.0235]

(To be continued)

q_{10}	0.5888	0.9746	0.0012	-0.2580	0.0350
	(0.0318)	(0.0060)	(0.0002)	(0.0138)	(0.0011)
	55.8941	128.2431	288.2593	312.3583	42.3190
	[0.5246,0.6497]	[0.9623,0.9859]	[0.0009,0.0017]	[-0.2804,-0.2269]	[0.0328,0.0372]
q_{11}	0.3472	0.9432	0.0012	-0.1335	0.0300
	(0.0162)	(0.0125)	(0.0003)	(0.0102)	(0.0010)
	121.9742	211.4949	318.4378	275.3242	51.9069
	[0.3166,0.3792]	[0.9148,0.9634]	[0.0009,0.0019]	[-0.1538,-0.1147]	[0.0281,0.0319]
q_{12}	0.2706	0.9195	0.0009	-0.1145	0.0277
	(0.0138)	(0.0173)	(0.0002)	(0.0122)	(0.0009)
	228.7777	226.2120	292.3517	314.5477	25.7573
	[0.2421,0.2969]	[0.8819,0.9506]	[0.0006,0.0013]	[-0.1375,-0.0884]	[0.0261,0.0294]
q_{13}	0.1984	0.9124	0.0013	-0.0475	0.0267
	(0.0149)	(0.0204)	(0.0003)	(0.0130)	(0.0009)
	229.7456	261.5111	326.0511	311.8691	53.1734
	[0.1692,0.2251]	[0.8638,0.9465]	[0.0008,0.0020]	[-0.0685,-0.0228]	[0.0250,0.0285]
q_{14}	0.1351	0.8536	0.0021	-0.0116	0.0240
	(0.0120)	(0.0292)	(0.0005)	(0.0107)	(0.0009)
	215.9103	256.1655	314.9568	279.6245	105.6430
	[0.1094,0.1580]	[0.7909,0.9037]	[0.0014,0.0035]	[-0.0321,0.0129]	[0.0222,0.0257]
q_{15}	0.3801	0.9591	0.0007	-0.1647	0.0287
	(0.0169)	(0.0118)	(0.0002)	(0.0117)	(0.0009)
	136.7918	255.0702	344.3820	319.4239	50.2741
	[0.3475,0.4132]	[0.9293,0.9772]	[0.0004,0.0012]	[-0.1867,-0.1447]	[0.0270,0.0305]

Notes: This table reports the in-sample correlation-related parameter estimation results based on RMSVL-GFT. The sample period is from January 3, 2006 to December 31, 2015. For each parameter, we report its posterior mean, posterior standard deviation (number in the parenthesis), inefficiency factor, as well as the 95% credible interval constructed using the 2.5th and 97.5th percentiles of the MCMC draws.

Figure 7: Filtered and (bias-corrected) realized measures in the empirical application.



Notes: This figure plots the posterior mean of selected log-volatility sequences (top panel) and pairwise correlation sequences (bottom panel) implied by RMSVL-GFT (the red dash-dotted line), accompanied by the corresponding bias-corrected realized measures (the blue solid line) and 95% credible intervals (yellow shaded area). The sample period is from January 3 2006 to December 30 2015.

Table 5: In-sample model comparison

	# of params	log-lik	DIC
MSV			
MSV-CC	39	-24171	48356
MSV-DCC	41	-24120	48283
MSV-GFT	63	-24005	48226
MSV-Chol	63	-24061	48252
MSVL			
MSVL-CC	45	-24102	48258
MSVL-DCC	47	-24077	48193
MSVL-GFT	69	-23996	48150
MSVL-Chol	69	-24003	48187

Notes: This table reports the number of parameters (# of params), the log marginal likelihood (log-lik) and deviance information criterion (DIC) values of competing models. The results are based on daily returns of JPM, GS, HON, CAT, JNJ and AMGN from January 3, 2006 to December 30, 2015.

Table 6: Model comparison based on out-of-sample predictive likelihood of returns.

	Overall			2013			2014			2015		
	pred-ll	MCS_1	MCS_2	pred-ll	MCS_1	MCS_2	pred-ll	MCS_1	MCS_2	pred-ll	MCS_1	MCS_2
MSV												
MSV-CC	.000	(*0.019*)	(*0.003*)	.000	(*0.716*)	(*0.406*)	.000	(*0.383*)	(*0.323*)	.000	(*0.017*)	(*0.013*)
MSV-DCC	.082	(*0.691*)	(*0.304*)	.060	(*0.993*)	(*0.724*)	.069	(*1.000*)	(*0.892*)	.116	(*0.064*)	(*0.083*)
MSV-GFT	.104	(*0.885*)	(*0.574*)	.072	(*0.993*)	(*0.724*)	.046	(*0.985*)	(*0.892*)	.193	(*0.511*)	(*0.622*)
MSV-Chol	.091	(*0.885*)	(*0.574*)	.058	(*0.993*)	(*0.794*)	.043	(*0.985*)	(*0.892*)	.172	(*0.436*)	(*0.305*)
MSVL												
MSVL-CC	.013	(*0.040*)	(*0.004*)	.003	(*0.735*)	(*0.498*)	.003	(*0.383*)	(*0.126*)	.033	(*0.005*)	(*0.003*)
MSVL-DCC	.089	(*0.885*)	(*0.479*)	.063	(*0.993*)	(*0.794*)	.055	(*0.985*)	(*0.892*)	.150	(*0.074*)	(*0.141*)
MSVL-GFT	.119	(*1.000*)	(*0.574*)	.081	(*1.000*)	(*0.794*)	.045	(*0.985*)	(*0.867*)	.232	(*1.000*)	(*0.622*)
MSVL-Chol	.092	(*0.885*)	(*0.574*)	.067	(*0.993*)	(*0.794*)	.037	(*0.985*)	(*0.892*)	.172	(*0.511*)	(*0.305*)
RMSV(L)												
RMSV-CC	.108		(*0.574*)	.118		(*0.794*)	.009		(*0.605*)	.197		(*0.622*)
RMSV-GFT	.175		(*0.574*)	.175		(*0.946*)	.090		(*0.892*)	**.262**		(*1.000*)
RMSVL-CC	.108		(*0.574*)	.119		(*0.794*)	.010		(*0.640*)	.197		(*0.622*)
RMSVL-GFT	**.176**		(*1.000*)	**.175**		(*1.000*)	**.092**		(*1.000*)	.261		(*0.622*)

Notes: This table reports the predictive log-likelihoods (pred-ll) of returns for all competing MSV models relative to MSV-CC. MCS_2 is the p-value for model confidence set for all candidate models. MCS_1 is the p-value when comparison is confined to models without realized measures. Underlined values for the best models according to MCS_1. Boldface values are for the best model according to MCS_2.

Table 7: Model comparison based on out-of-sample portfolio construction.

	Overall			2013			2014			2015		
	GMV	MCS_1	MCS_2	GMV	MCS_1	MCS_2	GMV	MCS_1	MCS_2	GMV	MCS_1	MCS_2
MSV												
MSV-CC	0.797	(*0.073*)	(*0.003*)	0.592	(*0.000*)	(*0.010*)	0.722	(*0.571*)	(*0.107*)	1.079	(*0.344*)	(*0.181*)
MSV-DCC	0.783	(*0.201*)	(*0.044*)	0.575	(*0.076*)	(*0.322*)	0.716	(*1.000*)	(*0.115*)	1.060	(*0.393*)	(*0.295*)
MSV-GFT	0.767	(*0.588*)	(*0.257*)	0.548	(*0.488*)	(*0.794*)	0.743	(*0.571*)	(*0.107*)	1.012	(*0.560*)	(*0.599*)
MSV-Chol	0.803	(*0.073*)	(*0.003*)	0.597	(*0.488*)	(*0.417*)	0.756	(*0.571*)	(*0.107*)	1.056	(*0.281*)	(*0.031*)
MSVL												
MSVL-CC	0.786	(*0.073*)	(*0.003*)	0.583	(*0.067*)	(*0.067*)	0.720	(*0.571*)	(*0.107*)	1.055	(*0.393*)	(*0.092*)
MSVL-DCC	0.772	(*0.588*)	(*0.050*)	0.567	(*0.417*)	(*0.417*)	<u>0.713</u>	(*0.728*)	(*0.107*)	1.037	(*0.560*)	(*0.295*)
MSVL-GFT	<u>0.756</u>	(*1.000*)	(*0.367*)	**<u>0.537</u>**	(*1.000*)	(*1.000*)	0.741	(*0.554*)	(*0.081*)	<u>0.992</u>	(*1.000*)	(*0.763*)
MSVL-Chol	0.801	(*0.017*)	(*0.000*)	0.586	(*0.417*)	(*0.417*)	0.769	(*0.296*)	(*0.001*)	1.049	(*0.281*)	(*0.015*)
GARCH												
GARCH-DCC	0.811	(*0.003*)	(*0.000*)	0.617	(*0.000*)	(*0.000*)	0.746	(*0.426*)	(*0.032*)	1.071	(*0.344*)	(*0.074*)
GARCH-BEKK	0.849	(*0.003*)	(*0.000*)	0.652	(*0.076*)	(*0.051*)	0.794	(*0.426*)	(*0.107*)	1.103	(*0.344*)	(*0.029*)
GARCH-GFT-DCS	0.809	(*0.003*)	(*0.000*)	0.607	(*0.000*)	(*0.000*)	0.759	(*0.426*)	(*0.050*)	1.063	(*0.281*)	(*0.031*)
RMSV(L)												
RMSV-CC	0.734		(*0.367*)	0.563		(*0.417*)	0.672		(*0.139*)	0.968		(*0.763*)
RMSV-GFT	0.719		(*0.933*)	0.545		(*0.970*)	**0.669**		(*1.000*)	0.945		(*0.763*)
RMSVL-CC	0.734		(*0.367*)	0.563		(*0.417*)	0.672		(*0.115*)	0.967		(*0.763*)
RMSVL-GFT	**0.719**		(*1.000*)	0.546		(*0.945*)	0.669		(*0.150*)	**0.944**		(*1.000*)
EW	0.948	(*0.003*)	(*0.000*)	0.758	(*0.000*)	(*0.000*)	0.761	(*0.554*)	(*0.107*)	1.327	(*0.073*)	(*0.006*)

Notes: This table reports the average squared returns of the global minimum variance (GMV) portfolio for all competing MSV and MGARCH models, together with that of a equal-weight portfolio. MCS_2 are model confidence set p-values based on absolute portfolio returns when all candidate models are considered. MCS_1 are p-values when comparison is confined to models without realized measures. Underlined GMV values identify best models without realized measures according to MCS_1. GMV values in boldface identify the best model according to MCS_2.

Online Supplement to "Multivariate Stochastic Volatility Models based on Generalized Fisher Transformation" by Han Chen, Yijie Fei, Jun Yu

(Not for Publication)[1]

This Online Supplement includes details of review of existing MSV models and their inference, introduction to PG, details of PGAS algorithm, expression of the Jacobian for GFTI, details of sampling model parameters of MSV-GFT, inference for RMSVL-GFT model, extension to high dimensions and comparison with estimation based on pairwise Fisher transformation.

A Review of existing MSV models and their inference

A.1 Model specifications

The existing MSV literature is vast, with numerous studies proposing alternative modeling strategies and estimation techniques. Two comprehensive reviews of the literature are Asai et al. (2006) and Chib et al. (2009). In this section, we focus on the models proposed over the last decade, which have aimed to address the challenge of ensuring the positive-definiteness of the variance-covariance matrix. We provide a critical review of these recent models and their parametrizations.[2]

The basic structure of the MSV model is

$$r_t|C_t \sim N(0, C_t),$$

where r_t is a vector of q asset returns. We aim at characterizing the dynamics of its variance-covariance matrix C_t. Clearly, C_t must be symmetric and positive-definite for all t. Different models rely on different techniques to ensure this positive-definiteness.

Broadly speaking, we can categorize the MSV models into two groups. In the first group, a model is directly built for C_t. In the second group, a variance-covariance decomposition is first carried out and then each component in the decomposition is modeled separately.

[1]We would like to thank Viktor Todorov (the co-editor), an associate editor, two referees, and Peter Hansen for useful discussions and comments. Han Chen (Email: hanchen@hnu.edu.cn) and Yijie Fei (Email: yijiefei@hnu.edu.cn), College of Finance and Statistics, Hunan University, 109 Shijiachong Road, Changsha, China. Jun Yu (Email: junyu@um.edu.mo), Faculty of Business Administration, University of Macau, Avenida da Universidade, Macao, China.

[2]We do not review any models based on the factor structure, as our proposed models are all based on direct modeling of the variance-covariance matrix.

Within the first group of models, four methods have been considered. The first method is based on the matrix exponential. For example, Ishihara et al. (2016) and Asai et al. (2022) assume that

$$C_t = \exp(H_t/2),$$

and propose to model $vech(H_t)$ as a vector autoregressive (VAR) process. By the definition of the matrix exponential, C_t is guaranteed to be positive-definite. The major drawback of this model is that the relationship between latent variables and the original volatilities/correlations is highly nonlinear and hence, hard to interpret.

The second method utilizes the well-known Cholesky decomposition. For instance, Lopes et al. (2010) propose to decompose C_t as

$$C_t = A_t H_t A_t',$$

where H_t is a diagonal matrix and A_t is a lower triangular matrix, and then model all the nonzero elements in A_t and H_t as the autoregressive process. Similarly, Nakajima (2017), Shirota et al. (2017) and Zaharieva et al. (2020) also use this decomposition to set up their MSV models. Unfortunately, in the Cholesky decomposition, order matters. That is, the resulting variance-covariance matrix depends on the ordering of assets. This dependence is highly undesirable as the performance of the model depends on the ordering. See Arias et al. (2023) for a detailed discussion on the macroeconomic forecasting performance of MSV based on this specification. Moreover, the dynamics of the volatilities and the correlations are not separated.

The third method takes advantage of the Wishart distribution, whose support includes only positive-definite matrices. This approach is considered in Gouriéroux et al. (2009), where a Wishart autoregressive (AR) process is used. Specifically, they assume that

$$\begin{aligned} C_t &= \sum_{i=1}^{m} x_{it} x_{it}', \\ x_{it} &= A x_{i,t-1} + \epsilon_{it} \text{ and } \epsilon_{it} \sim N(0, \Sigma), \end{aligned}$$

where (m, A, Σ) are unknown parameters. Alternatively, one can also model C_t using the inverse Wishart as in Philipov and Glickman (2006). In this case, we have

$$C_t^{-1} | v, C_{t-1}^{-1} \sim Wishart\left(v, \frac{1}{v}(A^{1/2})(C_{t-1}^{-1})^d (A^{1/2})'\right),$$

where (v, d, A) are unknown parameters. Clearly, the dynamics of the volatilities and the correlations are not separated.

Dellaportas et al. (2023) point out that Wishart-based MSV models are not able to scale well with p, the number of assets, as the computational complexity is $O(p^3)$. In view of this drawback, they propose a new MSV model assuming Gaussian latent processes for functions of the eigenvalues and rotation angles of C_t. In particular, the spectral decomposition implies that

$$C_t = P_t \Lambda_t P_t',$$

where Λ_t is a diagonal matrix of eigenvalues and P_t is the eigenvector matrix. They further decompose P_t as a product of $p(p-1)/2$ rotation matrices, whose elements are modeled after transformation. This parameterization leads to $O(p^2)$ complexity and thus more scalable. Nevertheless, the modeled variables in this setup are still hard to interpret.

Models in the second group treat the volatilities and the correlation matrix separately and is amenable to easy interpretation of driving factors of volatilities and correlations. Consider the following decomposition

$$C_t = V_t^{1/2} R_t V_t^{1/2}, \tag{A.1}$$

where V_t is a diagonal matrix collecting all the variances, and R_t is the correlation matrix. By construction, the diagonal elements of R_t are ones and the off-diagonal elements of R_t are pair-wise correlation coefficients. For our purpose, the major difference in model designs within this group lies in how R_t is parameterized. The critical issue in this setup is to ensure R_t is a valid correlation matrix, such as the positive-definiteness, symmetry, all the diagonal elements being one, all the off-diagonal elements taking values in $[-1, 1]$. The first and the simplest model in this fashion is the constant correlation MSV in Harvey et al. (1994), where

$$R_t = R, \text{ for all } t. \tag{A.2}$$

A similar assumption is made in Chan et al. (2006), Asai and McAleer (2006), and Ishihara and Omori (2012). Although the assumption of constant correlation makes statistical inference simple, it is too restrictive for financial time series.

To allow for time-varying correlations, Asai and McAleer (2009b) consider two models, both motivated by the dynamic conditional correlation (DCC) model of Engle (2002). The idea is to write the correlation matrix as

$$R_t = \tilde{Q}_t^{-1} Q_t \tilde{Q}_t^{-1},$$

where $\tilde{Q}_t = (diag(diag(Q_t))^{1/2}$. By construction, all diagonal elements of R_t are ones and R_t is a valid correlation matrix as long as Q_t is symmetric positive-definite. The two existing

Wishart-distribution-based models for Q_t are, respectively,

$$Q_{t+1} = (1-\phi)\bar{Q} + \phi Q_t + \Xi_t, \text{ where } \Xi_t \sim Wishart(k, \Lambda),$$

and

$$Q_{t+1}^{-1}|k, Q_t^{-1} \sim Wishart\left(k, \frac{1}{k} Q_t^{-\phi/2} \Lambda Q_t^{-\phi/2}\right),$$

where the unknown parameters are k, ϕ, Λ. Asai and McAleer (2009b) argue that the second one is preferred, which is also used in Arias et al. (2023) for macroeconomic forecasting.

Inspired by the dynamic equicorrelation (DECO) model of Engle and Kelly (2012), Kurose and Omori (2016) propose to model R_t as

$$R_t = (1-\rho_t)I + \rho_t J,$$

where I is an identity matrix, and J is a square matrix with all elements being ones. To ensure that ρ_t is within $(-1, 1)$, Kurose and Omori (2016) model the Fisher z-transformation of ρ_t as an autoregressive process. As in the model of Yamauchi and Omori (2020), a lower bound for ρ_t is needed to guarantee the positive-definiteness of R_t. This lower bound approaches zero as the number of assets goes to infinity.

Following Yu and Meyer (2006), Yamauchi and Omori (2020) propose to model the pairwise correlations by the Fisher z-transformation. That is, letting $R_t = \{\rho_{ij,t}\}$ and

$$g_{ij,t} = \frac{1}{2} \log \frac{1+\rho_{ij,t}}{1-\rho_{ij,t}} := F\left(\rho_{ij,t}\right), \tag{A.3}$$

they assume that $g_{ij,t}$ follows a random walk process for any $i \neq j$. By construction, $|\rho_{ij,t}| < 1$. When $p = 2$, that is, only two returns are modeled, this transformation ensures the positive-definiteness of R_t. However, if $p > 2$, this element-wise operation does not guarantee the positive-definiteness of R_t. Yamauchi and Omori (2020) further derive algebraic bounds for $\rho_{ij,t}$ that ensure the positive-definiteness of R_t. The bounds for one particular $\rho_{ij,t}$ are conditional on all other elements in R_t. Therefore, the restriction is well suited for the single-move Gibbs sampling technique, but hard to be implemented by other estimation methods.

A.2 Standard Bayesian inference for MSV models

Unlike GARCH-type models, which can be estimated straightforwardly by the frequentist maximum likelihood (ML) method, estimating SV models are particularly challenging in terms of likelihood-based estimation and inference. The difficulty mainly arises from the high-dimensional latent variables involved in SV models. To be more specific, to obtain the

likelihood function of SV models, one needs to integrate out all the latent variables from the joint probability density of observables and latent variables, that is,

$$p(r|\theta) = \int p(r, x|\theta)\mathrm{d}z = \int p(r|h, q, \theta)p(h|\theta)p(q|\theta)\mathrm{d}h\mathrm{d}q.$$

Unfortunately, such an integration, being $((p + d) \times T)$-dimensional, does not have the analytical solution.

In the context of MSV, there is an extra difficulty with the ML method. MSV models involve a large number of parameters (i.e. the dimension of θ is large). The ML method requires numerical maximization of $\log p(r|\theta)$ over θ. This often imposes a numerical challenge.

To deal with these two complications, the MSV literature rely on Bayesian methods to conduct statistical inference. A popular choice is standard MCMC methods which conduct the Bayesian posterior analysis based on $p(r|\theta, x)$ which is more tractable than $p(r|\theta)$. In particular, standard MCMC methods consists of alternately updating the component of x conditional on θ and θ conditional on x. A single-move MCMC method draws a single element of x at a time while a multiple-move MCMC method draws a block of elements of x at a time.

Standard MCMC algorithms have been applied to estimate MSV models in the literature. Yu and Meyer (2006) use the single-move algorithm to estimate several bivariate MSV models. Yamauchi and Omori (2020) use the single-move algorithm to estimate the pairwise-Fisher-transformation-based MSV model. However, the single-move algorithm is well known to be inefficient, as it generates highly autocorrelated Markov chains, suggesting a vast amount of random draws are required to achieve a satisfactory accuracy of estimation.

To improve efficiency, Ishihara and Omori (2012), Ishihara et al. (2016), Kurose and Omori (2016) have resorted to the multi-move algorithms to estimate different MSV models. These studies are built on some earlier works by De Jong and Shephard (1995), Pitt and Shephard (1999), Kim et al. (1998), and Watanabe and Omori (2004) in the univariate time series context. The multi-move samplers often require the second-order approximation to the target distribution. In general the derivation of the approximation is complicated and model dependent, making a generic multi-move algorithm not possible. Arias et al. (2023) propose a novel elliptical slice sampling method for estimating a time-varying parameter VAR with innovations following MSV-GFT. Assuming random walk dynamics for all latent variables, they successfully implement this method to quarterly data of four macroeconomic variables. Their approach can be treated as a different type of multi-move sampler. For our application with thousands of daily financial observations, however, slice sampler is impractical due to heavy computational cost.

When latent variables of various degree of persistence co-exist in a model, the single-move sampler and the multi-move sampler may be combined. This idea is exploited in Asai and McAleer (2009b) where the multi-move sampler is applied to the latent variables that determine volatilities and the single-move sampler is applied to the latent variables that determine the correlations.

While standard MCMC algorithms can draw random samples from $p(\theta, x|r)$, additional efforts are needed to compute the likelihood $p(r|\theta)$ and the marginal likelihood $p(r)$ of the model. The marginal likelihood is an important quantity for model comparison.

B Introduction to PG approach

To fix some notations, let $r = (r_1', ..., r_T')'$, $h = (h_1', ..., h_T')'$, $q = (q_1', ..., q_T')'$, $x = (h', q')' := (x_1', ..., x_T')'$ so that $x_t = (h_t', q_t')'$. Vector x contains all latent variables and vector x_t contains all latent variables at period t. We use $x_{1:t}$ to denote $(x_1', ..., x_t')'$ for any $t = 1, \cdots, T$, θ denote the set of parameters in the model, and $p(r|\theta)$ denote the likelihood function of the model.

Consider a general state-space model given by

$$\begin{aligned} r_t|x_t = x, \theta &\sim f(\cdot|x, \theta), \\ x_{t+1}|x_t = x, \theta &\sim g(\cdot|x, \theta), \text{ and } x_1 \sim \mu_\theta(\cdot). \end{aligned} \tag{B.1}$$

where $f(\cdot|x, \theta)$ is the measurement density, $g(\cdot|x, \theta)$ is the transition probability density and $\mu_\theta(\cdot)$ is the initial density.

To sample from $p(\theta, x_{1:T}|r_{1:T})$, a Gibbs sampler draws alternately from the two conditional densities, $p(\theta|x_{1:T}, r_{1:T})$ and $p(x_{1:T}|r_{1:T}, \theta)$. PG draws random samples from $p(x_{1:T}|r_{1:T}, \theta)$ based on the particle filter, which is applicable as long as the measurement density $f(\cdot|x, \theta)$ can be numerically evaluated and the transition density $g(\cdot|x, \theta)$ can be simulated.[3]

The particle filter combines importance sampling and Monte Carlo simulations to approximate the target distribution. The key idea is to represent the distribution by a set of random samples with the corresponding weights and calculate the quantity of interest based on these samples and weights. Let $\{x_{1:t}^{(i)}, w_t^{(i)}\}_{i=1}^N$ be a random measure, where $\{x_{1:t}^{(i)}, i = 1, ..., N\}$ is a set of support points and $\{w_t^{(i)}, i = 1, ..., N\}$ are the associated weights. Each point is called a particle, and N is the number of particles used. The approximate distribution can then

[3]Despite its general applicability, when implementing particle filter for a particular model, many subtle issues must be considered. These include how to choose a proper importance density, how many particles to use, and whether a resampling step should be added. For a thorough discussion, see Arulampalam et al. (2002) and Johansen and Doucet (2008).

be written as

$$\hat{p}_\theta(dx_{1:t}|r_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta_{x_{1:t}^{(i)}}(dx_{1:t}),$$

where $r_{1:t}$ is similarly defined and $\delta(\cdot)$ is the Dirac delta function. $\hat{p}_\theta$ is a discrete weighted approximation to the target distribution p_θ. Apparently, the accuracy of the approximation can be improved if an increasing number of particles are included. Doing so, however, also dramatically raises the computational burden.

To obtain the weights, one resorts to importance sampling. That is, one samples N times from a candidate distribution, say $q_\theta(x_{1:t}|r_{1:t})$, and assign the weight

$$w_t^{(i)} \propto p_\theta(x_{1:t}^{(i)}|r_{1:t}) / q_\theta(x_{1:t}^{(i)}|r_{1:t})$$

to each sample drawn. In practice, it is hard, if not impossible, to pick up a proper importance density for the joint distribution of $x_{1:t}$ conditional on the data when sample size is large. Hence, this approach usually proceeds in a sequential fashion. Specifically, the importance density is chosen to admit the factorization such that

$$q_\theta(x_{1:t}|r_{1:t}) = q_\theta(x_t|x_{t-1}, r_t) q_\theta(x_{1:t-1}|r_{1:t-1}).$$

For any existing weighted sample $\{x_{1:t-1}^{(i)}, w_{1:t-1}^{(i)}\}$ that follows from $p_\theta(x_{1:t-1}|r_{1:t-1})$, we augment it with the new state $x_t^{(i)}$ randomly drawn from $q_\theta(x_t|x_{t-1}, r_t)$. The joint sample, $(x_{t-1}^{(i)}, x_t^{(i)})$ is then a realization from the targeted joint importance density. The corresponding weight for i^{th} sample can easily be updated through

$$\tilde{w}_t^{(i)} \propto w_{t-1}^{(i)} \frac{f_\theta(r_t|x_t^{(i)}) g_\theta(x_t^{(i)}|x_{t-1}^{(i)})}{q_\theta(x_t^{(i)}|x_{t-1}^{(i)}, r_t)},$$

and normalized to be $w_t^{(i)} = \frac{1}{N}\sum_{i=1}^{N} \tilde{w}_t^{(i)}$. An unavoidable problem of this procedure, known as degeneracy, is that after a few iterations, only one particle has a non-negligible weight, which means a large computational cost is spent on particles with almost no contribution. To alleviate this problem, a resampling step is necessary. An important by-product of this filtering strategy is an approximation to $p_\theta(r_{1:t}|r_{1:t-1})$, which has a simple formula $\hat{p}_\theta(r_{1:t}|r_{1:t-1}) = \frac{1}{N}\sum_{i=1}^{N} w_t^{(i)}$. The likelihood can then be easily obtained as $\hat{p}_\theta(r_{1:T}) = \hat{p}_\theta(r_1)\prod_{t=2}^{T} \hat{p}_\theta(r_{1:t}|r_{1:t-1})$.

One subtlety to note is that, to ensure the targeted joint density is indeed the invariant distribution of a Markov chain, we have to modify the particle filter when applying PG. Specifically, one particle trajectory must be specified a priori to serve as a reference. This modified version is known in the literature as conditional particle filter. The intuition is that

this particular path can guide the simulated particles to move within a relevant region of the state space. See Theorem 5 of Andrieu et al. (2010) for more details.

C Details of PGAS algorithm

Consider a state-space model in the form of model (B.1). The output of a PGAS algorithm is a random draw from the joint distribution $p_\theta(x_{1:T}|r_{1:T})$, conditional on one particular set of parameter values. In the following, we omit parameters in all densities with an understanding that they depend on the same θ. The input of this algorithm, except for θ, is a reference trajectory of $x_{1:T}$, which is a sample from the last MCMC iteration. Let us denote that reference trajectory by $x'_{1:T}$. Then, the algorithm proceeds as following:

- Draw $x_1^{(i)}$ from $q_1(x_1|r_1)$, for $i = 1, 2, ..., N-1$.
- Set $x_1^{(N)} = x'_1$.
- Set $w_1^{(i)} = f(r_1|x_1^{(i)})/q_1(x_1^{(i)}|r_1)$, for $i = 1, 2, ..., N$.
- For $t = 2$ to T, do the following:
  - Generate $\{\tilde{x}_{1:t-1}^{(i)}\}_{i=1}^{N-1}$ by sampling with replacement $N-1$ times from $\{x_{1:t-1}^{(i)}\}_{i=1}^{N}$ with probabilities proportional to the importance weights $\{w_{t-1}^{(i)}\}_{i=1}^{N}$.
  - Draw J from $\{1, 2, ..., N\}$ with probabilities proportional to $w_{t-1}^{(i)} g(x'_t|x_{t-1}^{(i)})$ and then set $\tilde{x}_{1:t-1}^{(N)} = x_{1:t-1}^{(J)}$.
  - Simulate $x_t^{(i)}$ from $q_t(x_t|\tilde{x}_{t-1}^{(i)}, r_t)$, for $i = 1, 2, ..., N-1$.
  - Set $x_t^{(N)} = x'_t$.
  - Set $x_{i:t}^{(i)} = (\tilde{x}_{1:t-1}^{(i)}, x_t^{(i)})$.
  - Set weight to be $w_t^{(i)} = f(r_t|x_t^{(i)}) g(x_t^{(i)}|\tilde{x}_{t-1}^{(i)})/q_t(x_t^{(i)}|\tilde{x}_{t-1}^{(i)}, r_t)$, for $i = 1, 2, ..., N$.
- Draw k from $\{1, 2, ..., N\}$ with probabilities proportional to $w_T^{(i)}$ and return $x_{1:T}^* = x_{1:T}^{(k)}$.

Note that this procedure is very similar to the original PG sampler. A major modification is in drawing J, where a new index is drawn and thus the N^{th} trajectory may not be the reference one from the last iteration. In the conditional PG, on the contrary, we fix the last particle to follow the input trajectory $x'_{1:T}$. It is also worth mentioning that the probability of drawing J depends on $g(x'_t|x_{t-1}^{(i)})$ and x'_t is drawn in the last iteration conditional on all observations $r_{1:t}$. Therefore, this step makes the algorithm more like a smoothing instead of filtering.

D Expression of the Jacobian for GFTI

Let $\delta(z) = diag\left(e^{A[z]}\right)$ be the left-hand side of equation (4) in main text. Let $A[z] = Q\Lambda Q'$, where Λ is the diagonal matrix with the eigenvalues, $\lambda_1, \cdots, \lambda_p$, of $A[z]$ and Q is the orthonormal matrix with corresponding eigenvectors. Further define Ξ as a $p^2 \times p^2$ diagonal matrix with elements given by

$$\xi_{ij} = \Xi_{(i-1)p+j,(i-1)p+j} = \begin{cases} e^{\lambda_i} & \text{if } \lambda_i = \lambda_j \\ \frac{e^{\lambda_i} - e^{\lambda_j}}{\lambda_i - \lambda_j} & \text{if } \lambda_i \neq \lambda_j \end{cases}.$$

Following Appendix of Archakov and Hansen (2021), we have

$$J(z) = [D(z)]^{-1} H(z),$$

where $D(z) = diag(\delta_1, \cdots, \delta_p)$ is a $p \times p$ diagonal matrix and $H(z)$ is a $p \times p$ matrix with elements given by

$$[H(z)]_{ij} = (Q_{i,.} \otimes Q_{i,.}) \Xi (Q_{i,.} \otimes Q_{i,.})' = \sum_{k=1}^{p} \sum_{l=1}^{p} q_{ik} q_{jk} q_{il} q_{jl} \xi_{kl},$$

where q_{ij} and $Q_{i,.}$ denote the (i,j)th element and ith row of Q, respectively. From the above expression, it is straightforward to note that the time complexity of computing $J(z)$ is $O(p^4)$.

E Details of Sampling Model Parameters of MSV-GFT

The joint posterior distribution can be written as

$$\begin{aligned} p(\theta, h, q|r) &\propto p(r|\theta, h, q) p(\theta, h, q) \\ &= f(r|h, q) g_\theta(h) g_\theta(q) \pi(\theta) \\ &= f(r_1|h_1, q_1) g_\theta(h_1) g_\theta(q_1) \prod_{t=2}^{T} \left[f(r_t|h_t, q_t) g_\theta(h_t|h_{t-1}) g_\theta(q_t|q_{t-1}) \right] \pi(\theta) \\ &= \prod_{t=1}^{T} \left[\left(\sum_{i=1}^{p} h_{it} \right) |R_t|^{-1/2} \exp\left[-\frac{1}{2} r_t' (V_t^{1/2} R_t V_t^{1/2})^{-1} r_t \right] \right] \\ &\quad \times \prod_{t=2}^{T} \prod_{i=1}^{p} \left[(\sigma_{hi}^2)^{-1/2} \exp\left(-\frac{1}{2\sigma_{hi}^2} \left(h_{it+1} - \mu_{hi} - \phi_{hi}(h_{it} - \mu_{hi}) \right)^2 \right) \right] \\ &\quad \times \prod_{t=2}^{T} \prod_{j=1}^{d} \left[(\sigma_{qj}^2)^{-1/2} \exp\left(-\frac{1}{2\sigma_{qj}^2} \left(q_{jt+1} - \mu_{qj} - \phi_{qj}(q_{jt} - \mu_{qj}) \right)^2 \right) \right] \end{aligned} \tag{E.1}$$

$$\times \prod_{i=1}^{p} \left(\frac{\sigma_{hi}^2}{1-\phi_{hi}^2} \right)^{-1/2} \exp\left(-\frac{(h_{i1}-\mu_{h1})^2}{2\sigma_{hi}^2/(1-\phi_{hi}^2)} \right)$$
$$\times \prod_{j=1}^{d} \left(\frac{\sigma_{qj}^2}{1-\phi_{qj}^2} \right)^{-1/2} \exp\left(-\frac{(q_{j1}-\mu_{q1})^2}{2\sigma_{qj}^2/(1-\phi_{qj}^2)} \right) \pi(\theta).$$

To sample from the posterior distribution of parameters conditional on the realization of latent variables, we can do the following:

1. We can directly sample from the full conditional distribution of μ_{hi} and μ_{qi} which a normal distribution. For $i = 1, ..., p$ and $j = 1, ..., d$,

$$\mu_{hi}|r, h, q, \theta_{/\mu_{hi}} \sim N\left(\tilde{m}_{h\mu}, \tilde{s}_{h\mu}^2\right) \text{ and } \mu_{qj}|r, h, q, \theta_{/\mu_{qj}} \sim N\left(\tilde{m}_{q\mu}, \tilde{s}_{q\mu}^2\right), \tag{E.2}$$

where

$$\tilde{m}_{h\mu} = \tilde{s}_{h\mu}^2 \left\{ \frac{1-\phi_{hi}^2}{\sigma_{hi}^2} h_{i1} + \frac{1-\phi_{hi}}{\sigma_{hi}^2} \sum_{t=1}^{T-1} (h_{it+1} - \phi_{hi} h_{it}) \right\},$$
$$\tilde{m}_{q\mu} = \tilde{s}_{q\mu}^2 \left\{ \frac{1-\phi_{qj}^2}{\sigma_{qj}^2} q_{j1} + \frac{1-\phi_{qj}}{\sigma_{qj}^2} \sum_{t=1}^{T-1} (q_{jt+1} - \phi_{qj} q_{jt}) \right\},$$

and

$$\tilde{s}_{h\mu}^2 = \sigma_{hi}^2 \left[(T-1)(1-\phi_{hi})^2 + (1-\phi_{hi})^2 \right]^{-1},$$
$$\tilde{s}_{q\mu}^2 = \sigma_{qj}^2 \left[(T-1)(1-\phi_{qj})^2 + (1-\phi_{qj})^2 \right]^{-1}.$$

2. To draw random samples from the full conditional distribution of ϕ_{hi} and ϕ_{qi}, one can resort to the Metropolis-Hasting sampler. Since

$$\begin{aligned} \log p(\phi_{hi}|r, h, q, \theta_{/\phi_{hi}}) &\propto \log p(h_i|\phi_{hi}, \theta_{/\phi_{hi}}) + \log \pi(\phi_{hi}) \\ &= \log \pi(\phi_{hi}) - \frac{(h_{i1}-\mu_{hi})^2(1-\phi_{hi}^2)}{2\sigma_{hi}^2} + \frac{1}{2}\log(1+\phi_{hi}^2) \\ &\quad - \frac{\sum_{t=1}^{T-1} \left[(h_{it+1}-\mu_{hi}) - \phi_{hi}(h_{it}-\mu_{hi}) \right]^2}{2\sigma_{hi}^2}, \end{aligned} \tag{E.3}$$

we draw ϕ_{hi}^* from the proposal normal density $N\left(\hat{\phi}_{hi}, V_{\phi_{hi}}\right)$, where

$$\hat{\phi}_{hi} = \frac{\sum_{t=1}^{T-1} (h_{it+1}-\mu_{hi})(h_{it}-\mu_{hi})}{\sum_{t=1}^{T-1} (h_{it}-\mu_{hi})^2},$$

is the ordinary least square estimator of ϕ_{hi} given h_i and

$$V_{\phi_{hi}} = \sigma_{hi}^2 \left[\sum_{t=1}^{T-1} (h_{it} - \mu_{hi})^2 \right]^{-1}.$$

Then, the draw is accepted with probability $\min \left[1, \exp \left\{ g(\phi_{hi}^*)/g(\phi_{hi}^{(i-1)}) \right\} \right]$, where $\phi_{hi}^{(i-1)}$ is the sample from the last MCMC iteration and

$$g(\phi_{hi}) = \log \pi(\phi_{hi}) - \frac{(h_{i1} - \mu_{hi})^2 (1 - \phi_{hi}^2)}{2\sigma_{hi}^2} + \frac{1}{2} \log(1 + \phi_{hi}^2).$$

ϕ_{qi} can be treated in the same fashion.

3. Similar to the case for μ, due to the conjugacy, draws of σ_{hi}^2 can come from an inverse gamma distribution. For $i = 1, ..., p$ and $j = 1, ..., d$,

$$\sigma_{hi}^2 | r, h, q, \theta_{/\sigma_{hi}^2} \sim IG\left(\frac{\tilde{n}_m}{2}, \frac{\tilde{d}_{hm}}{2} \right) \text{ and } \sigma_{qj}^2 | r, h, q, \theta_{/\sigma_{qj}^2} \sim IG\left(\frac{\tilde{n}_m}{2}, \frac{\tilde{d}_{qm}}{2} \right), \tag{E.4}$$

where $\tilde{n}_m = n_{m0} + T$ and

$$\tilde{d}_{hm} = d_{m0} + (h_{i1} - \mu_{hi})^2 (1 - \phi_{hi}^2) + \sum_{t=1}^{T-1} \left[(h_{it+1} - \mu_{hi}) - \phi_{hi}(h_{it} - \mu_{hi}) \right]^2,$$

$$\tilde{d}_{qm} = d_{m0} + (q_{j1} - \mu_{qj})^2 (1 - \phi_{qj}^2) + \sum_{t=1}^{T-1} \left[(q_{jt+1} - \mu_{qj}) - \phi_{qj}(q_{jt} - \mu_{qj}) \right]^2.$$

F Inference for RMSVL-GFT model

Estimating MSV models with realized measures typically is computationally intensive and relies heavily on simulation-based methods. Many studies in this domain employ either Bayesian inference with a single-move MCMC sampler or maximum simulated likelihood. The latter technique, introduced in Koopman and Scharth (2012) and utilized by Asai et al. (2022), is a common choice. In our case, we cannot utilize the maximum simulated likelihood approach because for the relatively modest dimensionality of $p = 6$ considered in our empirical analysis, this method would necessitate an exceedingly large number of iterations before the maximization process converges, rendering it impractical.

Our PGAS-based Bayesian method can be straightforwardly modified to estimate the RMSV-GFT and RMSVL-GFT models. Note that the parameters of RMSVL-GFT model are $\theta = (\mu_h, \mu_q, \phi_h, \phi_q, \sigma_h^2, \sigma_q^2, \rho, \psi_h, \psi_q, \Sigma_h^r, \Sigma_q^r)'$. For additional parameters involved in (7)

and (8), we impose the following normal-inverse-gamma priors:

- $\psi_{hi} \sim N(m_{\psi 0}, s^2_{\psi 0})$ and $\psi_{qj} \sim N(m_{\psi 0}, s^2_{\psi 0})$;
- $\eta^2_{hi} \sim IG(\frac{n_{m0}}{2}, \frac{\eta_{m0}}{2})$ and $\eta^2_{qj} \sim IG(\frac{n_{m0}}{2}, \frac{\eta_{m0}}{2})$.

Let $h^r = (h_1^{r'}, ..., h_T^{r'})'$ and $q^r = (q_1^{r'}, ..., q_T^{r'})'$. The Gibbs sampler for the extended model with realized measures and leverage effect works as follows:

1. Initialize h, q and θ.
2. Draw $h, q|r, h^r, q^r, \theta$.
3. Draw $\mu_h, \mu_q|r, h, q, \theta_{/(\mu_h,\mu_q)}$.
4. Draw $\phi_h, \phi_q|r, h, q, \theta_{/(\phi_h,\phi_q)}$.
5. Draw $\sigma^2_h, \sigma^2_q|r, h, q, \theta_{/(\sigma^2_h,\sigma^2_q)}$.
6. Draw $\rho|r, h, \theta_{/(\rho)}$
7. Draw $\Sigma^r_h, \Sigma^r_q|h, q, h^r, q^r, \theta_{/(\Sigma^r_h,\Sigma^r_q)}$.
8. Draw $\psi_h, \psi_q|h, q, h^r, q^r, \theta_{/(\psi_h,\psi_q)}$.

Iterating over steps 2-8 consists of a complete sweep of MCMC sampling. The PGAS algorithm introduced in Section 3.1 is used to sample the latent variables h and q, conditional on daily returns and realized measures.

Specifically, as (7) and (8) are two extra measurement equations independent of the original ones, the conditional likelihood can be written as

$$p\left(r, h^r, q^r|h, q, \theta\right) = p\left(r|h, q, \theta_{/(\psi_h,\psi_q,\Sigma^r_h,\Sigma^r_q)}\right) p\left(h^r, q^r|h, q, \psi_h, \psi_q, \Sigma^r_h, \Sigma^r_q\right). \tag{F.1}$$

and θ, h, q can then be drawn from the full posterior distribution

$$p\left(\theta, h, q|r, h^r, q^r\right) \propto p\left(r|h, q, \theta_{/(\psi_h,\psi_q,\Sigma^r_h,\Sigma^r_q)}\right) p\left(h^r, q^r|h, q, \psi_h, \psi_q, \Sigma^r_h, \Sigma^r_q\right) p(h, q, \theta). \tag{F.2}$$

All the parameters except for $(\rho, \psi_h, \psi_q, \Sigma^r_h, \Sigma^r_q)$ can be sampled exactly the same as in (E.2), (E.3) and (E.4).

To draw the leverage parameter ρ, note that the conditional distribution of $h_{t+1}|h_t, r_t, \theta$ is

$$h_{t+1}|h_t, r_t, q_t, \theta_{/(\rho)}\theta \sim N\left(\mu_h + \phi_h(h_t - \mu_h)) + \Omega\Sigma_h^{\frac{1}{2}} z_t, \Sigma_h - \Omega\Sigma_h\Omega'\right)$$

where $\Omega = diag(\rho)$. The posterior distribution of ρ is

$$f(\rho|h, r, q, \theta_{/(\rho)}) \propto \prod_{t=1}^{T-1} f(h_{t+1}|h_t, r_t, q_t, \theta_{/(\rho)}) I(\rho \in (-1, 1))$$

and a random walk Metropolis Hasting sampler can be conducted to draw ρ.

To draw ψ_h and Σ_h^r, note that we have following conditional distribution:

$$\psi_{hi}|h, h^r, \theta_{/\psi_{hi}} \sim N\left(\tilde{m}_{hi\psi}, \tilde{s}^2_{hi\psi}\right) \text{ and } \eta^2_{hi}|h, h^r, \theta_{/\eta^2_{hi}} \sim IG\left(\frac{\tilde{n}_m}{2}, \frac{\tilde{\eta}_{hi}}{2}\right), \tag{F.3}$$

where

$$\tilde{m}_{hi\psi} = \tilde{s}^2_{hi\psi}\left(\frac{m_{\psi 0}}{T} + \sum_{t=1}^{T}(h^r_{it} - h_{it})\right), \; \tilde{s}^2_{hi\psi} = \left(\frac{1}{s^2_{\psi 0}} + \frac{T}{\eta^2_{hi}}\right)^{-1},$$

and

$$\tilde{n}_m = n_{m0} + T, \; \tilde{\eta}_{hi} = \frac{\eta_{m0}}{2} + \frac{\sum_{t=1}^{T}(h^r_{it} - h_{it})^2}{2}.$$

The sampling from these distributions are straightforward. Drawing ψ_q and Σ_q^r can be dealt with similarly.

G Comparison with models based on pairwise Fisher transformation

In this section, we compare our MSV-GFT specification with the specification that models each correlations in a pairwise fashion. For the latter approach that is denoted by MSV-P, the volatility of each asset is assumed to follow a univariate SV dynamics. As for the comovement structure among assets, we model each bivariate correlation by the Fisher transformation defined by equation (A.3) following the setup in Yu and Meyer (2006) and assume these correlation sequences are independent. For all latent volatility and (pairwise) correlation sequences, we obtain their posterior distributions by applying MCMC simulation based on PGAS algorithm. The posterior means are then taken as our estimator for those latent variables.

An apparent advantage of estimating correlations in a pairwise manner is that we can avoid inverting GFT operation numerically, as the inverse of bivariate Fisher transformation admits a closed-form expression. Moreover, for a p-dimensional model, we have $p(p-1)/2$ pairwise independent correlation sequences to be modelled and estimated, which can be dealt with by taking advantage of parallel computing. This can dramatically enhance the computational efficiency.

As indicated in the contour plot depicted in Figure 6, the bivariate Fisher transformed

correlation sequences are generally more correlated than those produced by GFT. To measure the information loss due to the pairwise approach, we calculate the total *bivariate* marginal log-likelihood defined by

$$\log ML := \sum_{i=1}^{p} \sum_{j=i+1}^{p} \log p(r^i, r^j),$$

where $r^i = (r_1^i, \cdots, r_T^i)'$ collects all returns of asset i, for both MSV-P and MSV-GFT model, without and with the leverage effect. The first row of Table G.1 reports the values of total bivariate marginal log-likelihood. Clearly our GFT-based specifications outperform their bivariate pairwise counterpart, suggesting an improved in-sample fit of our GFT approach than the bivariate approach.

As an alternative way to compare performance of the two types of models, for each pair of assets together with the sum of all pairs, we calculate the average bivariate predictive log-likelihoods defined by

$$\frac{1}{T-T_0} \sum_{t=T_0+1}^{T} \log p\left(r_t^i, r_t^j | r_{1:t-1}^i, r_{1:t-1}^j\right),$$

for pairwise modeling approach and

$$\frac{1}{T-T_0} \sum_{t=T_0+1}^{T} \log p\left(r_t^i, r_t^j | r_{1:t-1}\right),$$

for our MSV-GFT model. The results can be found in the lower part of Table G.1, where the figures in the parenthesis are the p-values of the MCS test. In almost all cases, our GFT-based specifications significantly outperform their pairwise counterpart.

A serious conceptual problem with the bivariate approach is that the correlation matrix from all bivariate models may not be positive-definite, as mentioned earlier. This problem can hinder possible applications of MSV-P. A solution proposed in Yamauchi and Omori (2020) is to impose conditional restrictions on each correlation given all other correlations. In this paper, following the suggestion from a referee, we consider the projections method proposed in Higham (2002) to ensure positive-definiteness. The idea is to apply a reasonable transformation to obtain a closest counterpart in the space spanned by all valid correlation matrices. The method can be readily implemented using MATLAB built-in function `nearcorr`. Formally, for each period t, let the matrix obtained by inserting all pairwise

Table G.1: Comparison based on MSV-GFT and MSV-P

	MSV-GFT		MSV-P		MSVL-GFT		MSVL-P	
In-sample								
$\log ML$	-124992		-126242		-123826		-124929	
Out-of-sample								
JPM vs. GS	-2.8032	*(0.50)*	-2.8231	*(0.01)*	-2.7995	*(1.00)*	-2.8176	*(0.02)*
JPM vs. HON	-2.9012	*(0.01)*	-2.9037	*(0.00)*	-2.8814	*(1.00)*	-2.8831	*(0.13)*
JPM vs. CAT	-3.2432	*(0.16)*	-3.2447	*(0.16)*	-3.2335	*(1.00)*	-3.2340	*(0.77)*
JPM vs. JNJ	-2.7862	*(0.37)*	-2.7878	*(0.24)*	-2.7786	*(1.00)*	-2.7799	*(0.63)*
JPM vs. AMGN	-3.4987	*(0.90)*	-3.4989	*(0.90)*	-3.4971	*(1.00)*	-3.4971	*(0.99)*
GS vs. HON	-2.9191	*(0.01)*	-2.9205	*(0.01)*	-2.8991	*(1.00)*	-2.9016	*(0.06)*
GS vs. CAT	-3.2293	*(0.51)*	-3.2286	*(0.61)*	-3.2243	*(0.80)*	-3.2240	*(1.00)*
GS vs. JNJ	-2.8144	*(0.64)*	-2.8163	*(0.36)*	-2.8091	*(1.00)*	-2.8115	*(0.64)*
GS vs. AMGN	-3.5146	*(1.00)*	-3.5170	*(0.53)*	-3.5147	*(0.98)*	-3.5176	*(0.53)*
HON vs. CAT	-3.0225	*(0.00)*	-3.0199	*(0.00)*	-2.9939	*(0.08)*	-2.9909	*(1.00)*
HON vs. JNJ	-2.6022	*(0.03)*	-2.6067	*(0.00)*	-2.5799	*(1.00)*	-2.5859	*(0.03)*
HON vs. AMGN	-3.3270	*(0.02)*	-3.3292	*(0.00)*	-3.3073	*(1.00)*	-3.3118	*(0.02)*
CAT vs. JNJ	-2.9506	*(0.29)*	-2.9521	*(0.15)*	-2.9407	*(1.00)*	-2.9419	*(0.56)*
CAT vs. AMGN	-3.6271	*(0.97)*	-3.6271	*(0.97)*	-3.6254	*(1.00)*	-3.6270	*(0.97)*
JNJ vs. AMGN	-3.0687	*(0.35)*	-3.0699	*(0.31)*	-3.0617	*(1.00)*	-3.0636	*(0.35)*
Total	-45.8120	*(0.02)*	-46.2602	*(0.00)*	-45.6552	*(1.00)*	-46.1004	*(0.00)*

predicted correlations be denoted by $\hat{R}_t^{pw}$. We define

$$\tilde{R}_t^{pw} = \begin{cases} \hat{R}_t^{pw}, & \text{if } \hat{R}_t^{pw} \text{ is positive definite} \\ \texttt{nearcorr}(\hat{R}_t^{pw}), & \text{if } \hat{R}_t^{pw} \text{ is not positive definite} \end{cases} \tag{G.1}$$

where the positive-definiteness of $\hat{R}_t^{pw}$ is determined by whether its minimum eigenvalue is positive. We denote the approach MSV(L)-P(C).

To compare the performance of MSV-GFT and MSV-P(C), we consider the construction of the global minimum variance portfolio, which requires a positive-definite forecast of the correlation matrix. Table G.2 reports the average squared return of the global minimum variance portfolio over entire out-of-sample period and three sub-periods, accompanied by the corresponding MCS p-values. The best model for each sample period is highlighted by boldface. We find again that our GFT-based models have superior performance compared with the pairwise counterparts. Such a lead is consistent across different sample periods and statistically significant in terms of the MCS test in many cases. Our comparison once again demonstrates for the superiority of the GFT approach.

Table G.2: Comparison MSV-GFT and MSV-P(C) based on portfolio construction

	Overall		2013		2014		2015	
MSV-GFT	0.767	(*0.302*)	0.548	(*0.241*)	**0.743**	(*1.000*)	1.012	(*0.355*)
MSVL-GFT	**0.756**	(*1.000*)	**0.537**	(*1.000*)	0.741	(*0.654*)	**0.992**	(*1.000*)
MSV-P(C)	0.780	(*0.002*)	0.556	(*0.005*)	0.763	(*0.185*)	1.021	(*0.085*)
MSVL-P(C)	0.772	(*0.011*)	0.544	(*0.226*)	0.761	(*0.185*)	1.012	(*0.355*)

H Extension to high dimensions

In this section, we briefly discuss the possible extensions of our MSV-GFT model to higher-dimensional case. As the computational burden of implementing the model increases exponentially with the number of assets, it may become impractical to use the model without imposing further restrictions.

A natural way to reduce the dimension of our MSV-GFT model is to assume certain factor structure. There are two specifications that can serve for this purpose. The first one stipulates that the $p \times 1$ return vector r_t follows an additive factor structure featuring K factors:

$$r_t = \Lambda f_t + \Omega^{1/2} e_t, \tag{H.1}$$

where Λ is a $p \times K$ matrix of factor loadings, f_t is a K-dimensional vector that is assumed to be generated by the MSV-GFT model (3a)-(3f) introduced in the main paper, Ω is a $p \times p$ (possibly diagonal) covariance matrix and e_t is a vector of p independent standard normal variates without serial dependence. In this model, the covariance matrix of returns $\Sigma_{r,t}$ can be decomposed into two parts:

$$\Sigma_{r,t} = \Lambda \Sigma_{f,t} \Lambda' + \Omega.$$

Note that if Ω is not diagonal, both parts will contribute to the comovement among assets. Such a specification has been widely used in the literature; see, for instance, Asai et al. (2006) and reference therein. In almost all cases, the number of factors K is pre-determined and no more than ten. Note that for identification purpose, constraints must be imposed on Λ. Following Dellaportas et al. (2023) and denoting (i,j)-th element of Λ by λ_{ij}, we may set $\lambda_{ij} = 0$ for $i < j$, $i \leq K$ and $\lambda_{ii} = 1$ for $i \leq K$. The inference of this model is very similar to MSV-GFT model without factor structure. Although additional steps in Gibbs sampler must be conducted to draw extra parameters Λ and Ω conditional on all other parameters and latent processes, they are straightforward and fast. Indeed, conditional on f_t, equation (H.1) is nothing more than a linear regression and it is trivial to draw from the posterior distribution of coefficients Λ and Ω if conjugate priors are assumed. Though empirically attractive, a drawback of this linear additive factor model is that it imposes a low-dimensional factor

structure on both p volatilities and $d = p(p-1)/2$ correlations. It implies that the dynamic volatility of p assets in such a model is assumed to be determined by that of K factors. This additional restriction may not be necessary for high-dimensional implementation as the number of volatilities grows linearly with the dimension, unlike correlations, whose quantity suffers a quadratic increase.[4]

The second specification, proposed in Archakov et al. (2024a), imposes factor structure directly on $p(p-1)/2$ GFT-transformed correlations q_t. In the same spirit, we may assume that

$$q_t = \Lambda f_t, \tag{H.2}$$

where Λ is a $d \times K$ loading matrix with the number of factors K in general much less than d. We may then specify a new model by adding equation (H.2) to (3a)-(3f) in the main paper and replace q_t in (3e) by f_t. This enables us to significantly reduce the number of latent correlation variables from d to K while still maintain p latent volatility variables. The inference for this specification is again similar to the MSV-GFT model without factors, with an extra step to sample from the conditional posterior distribution of Λ. As fewer latent variables are involved, the accuracy of MCMC estimation is expected to be improved. It is worth noting that there is a subtle yet important computational difference from the estimation of first specification above. When estimating the additive factor model (H.1) using PGAS-based MCMC, GFT inverse (denoted by $F^{-1}(\cdot)$ in the main paper) is only operated on $K(K-1)/2 \times 1$ vectors, corresponding to the number of correlations of f_t. For model involves specification (H.2), however, GFT inverse must be implemented to $p(p-1)/2 \times 1$ vectors. This is because in general $F^{-1}(q_t) = F^{-1}(\Lambda f_t)$ can not be expressed as a simple transformation of $F^{-1}(f_t)$. In fact, unless the dimension of f_t is $\frac{1}{2}l(l-1)$ for some positive integer l, $F^{-1}(f_t)$ is not even well-defined. As a result, we still have to compute $F^{-1}(q_t)$ for likelihood evaluation purpose, which is quite time-consuming.

As pointed out by Archakov et al. (2024a), an important special example of the second specification above is when pre-determined block structure is available, in which case the matrix Λ is known and thereby the inference can be further simplified. A block structure arises naturally in applications where the correlation between two variables is defined by their group classification. For example, we may assign stocks from the same sector as one cluster and the number of blocks K will then be equal to that of sectors; see Archakov and Hansen (2024) for an empirical illustration.

For our inference procedure, an important computational merit of making such an assumption is that the complexity of our algorithm can be dramatically reduced. To see this, consider following simple example. Assume that a 5×5 correlation matrix C is known to

[4]Furthermore, each volatility can be dealt with separately and hence parallel computing may be utilized to reduce computational cost. This does not apply to correlations, which must be tackled simultaneously.

have following structure

$$C = \left[\begin{array}{cc:ccc} 1 & \rho_{11} & \rho_{12} & \rho_{12} & \rho_{12} \\ \rho_{11} & 1 & \rho_{12} & \rho_{12} & \rho_{12} \\ \hdashline \rho_{12} & \rho_{12} & 1 & \rho_{22} & \rho_{22} \\ \rho_{12} & \rho_{12} & \rho_{22} & 1 & \rho_{22} \\ \rho_{12} & \rho_{12} & \rho_{22} & \rho_{22} & 1 \end{array}\right].$$

By Corollary 1 of Archakov and Hansen (2024), the block structure will be preserved after logarithm transformation and hence we must have

$$F(C) = vecl(\log C) = (q_{11}, q_{12} * \mathbb{1}_6', q_{22} * \mathbb{1}_3')'$$

for some (q_{11}, q_{12}, q_{22}). Now suppose that $F(C)$ is given and we wish to recover matrix C. Without invoking block structure, GFT inverse of a 10-dimensional vector must be computed. To avoid this calculation, note that GFT inverse is equivalent to solving following nonlinear system of equations

$$diag\left(e^{A[z]}\right) = \mathbb{1}_5,$$

where

$$A[z] = \log C = \left[\begin{array}{cc:ccc} x_1 & q_{11} & q_{12} & q_{12} & q_{12} \\ q_{11} & x_1 & q_{12} & q_{12} & q_{12} \\ \hdashline q_{12} & q_{12} & x_2 & q_{22} & q_{22} \\ q_{12} & q_{12} & q_{22} & x_2 & q_{22} \\ q_{12} & q_{12} & q_{22} & q_{22} & x_2 \end{array}\right].$$

and the unknowns are $z = (z_1, z_2)'$. Define

$$\tilde{A} = \exp\left[\begin{pmatrix} z_1 + q_{11} & \sqrt{6}q_{12} \\ \sqrt{6}q_{12} & z_2 + 2q_{22} \end{pmatrix}\right],$$

again by Corollary 1 of Archakov and Hansen (2024), we can straightforwardly verify that

$$diag\left(e^{A[z]}\right) = \begin{bmatrix} \left(\frac{1}{2}\tilde{A}_{11} + \frac{1}{2}e^{z_1 - q_{11}}\right) \cdot \mathbb{1}_2 \\ \left(\frac{1}{3}\tilde{A}_{22} + \frac{2}{3}e^{z_2 - q_{22}}\right) \cdot \mathbb{1}_3 \end{bmatrix}$$

where $\tilde{A}_{11}$ and $\tilde{A}_{22}$ are two diagonal elements of $\tilde{A}$. Therefore, the problem is reduced to

solving following nonlinear system of equations

$$\begin{cases} \frac{1}{2}\tilde{A}_{11} + \frac{1}{2}e^{z_1 - q_{11}} = 1 \\ \frac{1}{3}\tilde{A}_{22} + \frac{2}{3}e^{z_2 - q_{22}} = 1 \end{cases}$$

which is expected to be much faster to compute.

More generally, if p assets can be categorize into K blocks, we can reduce the problem of inverting a $p(p-1)/2 \times 1$ vector to solving a nonlinear system of equations with K unknowns, which in turn can be tackled by the fixed-point iteration algorithm described in Theorem 3.3 of Archakov et al. (2024b). This type of dimension reduction has also been employed in Tong et al. (2024) to construct high-dimensional score-driven dynamic correlation models with clusters. In fact, for applications with hundreds of assets, the number of blocks involved may still be large and our MSV-GFT model may not be feasible if we directly employ the fixed-point iteration proposed in Archakov et al. (2024b). If that is the case, it is possible to further speed up their algorithm again by taking advantage of Newton or Broyden's method. We leave a detailed investigation in this respect for future endeavors.