Journal of Econometrics 209 (2019) 114–138

Contents lists available at ScienceDirect

Journal of Econometrics

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

Bayesian estimation of dynamic asset pricing models with informative observations☆

Andras Fulop, Junye Li *

ESSEC Business School, Paris-Singapore, France

ARTICLE INFO

Article history:
Received 3 November 2017
Received in revised form 13 November 2018
Accepted 13 November 2018
Available online 23 November 2018

JEL classification:
C11
C13
G12
G13

Keywords:
Non-affineness
Self-exciting jumps
Optimal proposal density
Auxiliary particle filter
Common random numbers
Sequential Monte Carlo sampler

ABSTRACT

In dynamic asset pricing models, when the model structure becomes complex and derivatives data are introduced in estimation, traditional MCMC methods converge slowly, are difficult to design efficient proposals for parameters, and have large computational cost. We propose a two-stage sequential Monte Carlo sampler based on common random numbers and a smooth particle filter. This method is robust to potential model misspecification and can deliver almost full-likelihood-based inference at a much smaller computational cost. It is applied to estimate a class of volatility models that take into account price-volatility co-jumps, non-affineness, and self-excitation. An empirical study using S&P 500 index and variance swap rates shows that both non-affineness and self-excitation need to be introduced in modeling volatility dynamics.

1. Introduction

The literature on financial econometrics has advanced towards incorporating complex patterns in dynamics of stock returns and volatility. Recent empirical evidence shows the need to allow for co-jumps of prices and volatility (Eraker, 2004; Eraker et al., 2003; Jacod and Todorov, 2010; Todorov and Tauchen, 2011), self-exciting jump clustering (Ait-Sahalia et al., 2015; Carr and Wu, 2017; Fulop et al., 2015), and non-affine volatility dynamics (Ait-Sahalia and Kimmel, 2007; Amengual and Xiu, 2018; Christoffersen et al., 2010; Jones, 2003). Furthermore, large panels of derivatives observations are increasingly added to the datasets, as they provide rich information on return and volatility dynamics and are helpful to identify parameters of such complex models.

However, while rich model structure and introduction of derivatives data are useful, they also make model estimation a non-trivial undertaking, as one needs to simultaneously deal with the multiple dynamic latent states following complex non-Gaussian processes and the fixed parameters driving the dynamics that are embedded in the nonlinear pricing functions linking the latent states to the observed asset prices. In a Bayesian context, Gibbs-type Markov chain Monte Carlo (MCMC)

☆ We would like to thank Yacine Ait-Sahalia (the editor), two anonymous referees, Nicolas Chopin and seminar participants at University of Southern California, 2014 Princeton-QUT-SMU Conference in Financial Econometrics, European Seminar on Bayesian Econometrics (ESOBE), The Second ESSEC Workshop on Empirical Finance, 2015 SoFiE Annual Conference, 2015 China International Conference in Finance, 2015 International Conference on Systemic Risk, 2016 Asian Meeting of the Econometric Society, and 2017 China Meeting of the Econometric Society for helpful comments.

* Corresponding author.
E-mail addresses: fulop@essec.fr (A. Fulop), li@essec.edu (J. Li).

https://doi.org/10.1016/j.jeconom.2018.11.014
0304-4076/

methods have been used to estimate jump–diffusion stochastic volatility models where the dynamic latent states and the fixed parameters are iteratively sampled assuming the other quantity is known (Eraker, 2004; Eraker et al., 2003; Johannes and Polson, 2009; Li et al., 2008; Yu et al., 2011). However, such methods can be hard to implement when the model structure becomes complex and derivative data are included. First, the derivatives are very informative on the hidden states given the parameters and vice-versa, hence introducing strong correlation in the MCMC chain. This leads to a very slow convergence. Second, due to the complicated nonlinear derivative pricing functions, sufficient statistics for the conditional distributions of the parameters are hard to obtain and the posteriors of the individual parameters can be highly correlated. Hence, it is hard to design efficient proposal distributions for the parameters. Third, traditional MCMC methods are hard to be parallelized, making them computationally very costly, especially when derivatives data are taken into account.

In this paper, we propose a particle-based method that aims to overcome the above-mentioned issues. The first task is to break the correlation between the latent states and the fixed parameters. We marginalize out the former by estimating the likelihood function using a particle filter and then run a simulation routine targeting the marginal distribution of the fixed parameters. One approach to achieve this aim is the pseudo-marginalization approach pioneered by Andrieu and Roberts (2009) and Andrieu et al. (2010). This approach satisfies the "exact approximation" property, that is, it targets the right posterior despite the Monte Carlo error in the likelihood estimate. However, it is well understood that the key to the successful application of such an approach is the effective control of the noise of the likelihood estimate (Doucet et al., 2015). Unfortunately, we find that with real derivatives data, even a particle filter that uses the locally optimal proposal still provides likelihood estimates that are too noisy for our purposes.

The problem with the standard pseudo-marginalization approach is that it relies on an estimator of the likelihood ratio, $p(y \mid \Theta)/p(y \mid \Theta')$, obtained by dividing two estimators of $p(y \mid \Theta)$ and $p(y \mid \Theta')$, which are independent. By introducing positive correlations between these two estimators relying on common random numbers, it is possible to drastically reduce variance of the estimator of the likelihood ratio. However, it is well-known that fixing the auxiliary random variables used by the particle filter is not sufficient to address this problem as the resampling operation in its original form is such that two particles very far away from each other can be selected even if Θ and Θ' are quite close. To mitigate this problem, we exploit the fact that the observations are informative on the latent states and the filtering/smoothing distribution tends to be localized around its posterior mean. Hence, a normal approximation to the filtering/smoothing density only leads to small approximation errors, allowing us to bypass the resampling step.

The second task is to design efficient proposals targeting the marginal distribution of the parameters. To do so, we employ the density-tempered sequential Monte Carlo (SMC) sampler based on Del Moral et al. (2006). This method represents the target with a population of simulated points and allows one to adapt the proposals to the simulation output in an iterative manner. To profit from the use of common random numbers and to speed up the SMC sampler, we propose a novel two-stage SMC sampler where the first-stage exploration of the parameter space using a coarse likelihood estimate is followed by the second-stage adjustment using the larger number of state particles. The proposed method can be easily parallelized in the parameter dimension and allows us to fully use the computational power of modern graphical processing units (GPUs), resulting in a low computational cost. Importantly, this particle-based method is generic. The only model-specific design effort required from the user is the need to provide a reasonably efficient particle filter for the problem at hand, making it easy to adapt for a wide range of dynamic asset pricing models with informative observations.

To illustrate our methodology, we consider a dynamic asset pricing model that allows for price-volatility co-jumps, self-exciting jump clustering, and non-affine volatility. This model is very general and includes many models found in the literature as special cases. More importantly, even though the model is general, we are able to derive the locally optimal proposal in an analytical form for the dynamic states, where the novelty lies in deriving a fully adapted sampler over the exponentially distributed variance jumps. This approach may be interesting in its own right given the frequent use of such jump specifications in finance.

We implement extensive Monte Carlo studies to check the accuracy, efficiency, and stability of the proposed approach. First, we show that the bias in the log likelihood estimate from the use of our approximate smooth filter compared to the exact particle filter is minor. Second, we investigate what the eventual bias and efficiency loss are due to normal approximation and the use of common random numbers in parameter estimation. To do so, we first run the pseudo-marginalized routine of Duan and Fulop (2015) using the exact particle filter. This delivers draws from the exact posterior and provides a good reference. We then run our two-stage SMC sampler with common random numbers and the smooth particle filter. We find that both mean estimates and root mean squared errors (RMSEs) are very close between the two routines. Hence, we conclude that the small bias in the likelihood estimate due to normal approximation does not lead to any consequential bias or efficiency loss and the asymptotic bias due to the use of common random numbers is not a significant factor for a moderate number of particles. Furthermore, we observe that our two-stage SMC sampler runs at a fraction of the computing time required by the pseudo-marginalized routine.

We implement the empirical investigation using a sample of daily S&P 500 index returns and variance swap rates with maturity 1-, 6-, and 12-month ranging from January 2, 2001 to July 20, 2015. Our analysis uncovers several new empirical patterns. First, an interesting question unexplored so far in literature is whether non-affineness and self-excitation are substitutes to each other given that both are channels to allow for sudden bursts of volatility and extreme events. To investigate this issue, we compare the full model to the nested ones where we switch down non-affineness, self-excitation, or both. Extending most of the extant literature that have used affine specifications, we find support for self-excitation even when non-affineness is allowed for. Second, we find evidence that non-affineness plays a more important role than

self-excitation in capturing volatility dynamics, and that the non-affine specifications decisively beat affine ones, no matter whether self-excitation is taken into account.

The rest of the paper is organized as follows. Section 2 presents the model and discusses the pricing of variance swaps. Section 3 proposes an efficient particle-based method. Section 4 implements Monte Carlo studies. Section 5 presents an empirical application using S&P 500 index returns and variance swap rates. Finally, Section 6 concludes the paper.

2. The model

In this section, we present a dynamic asset pricing model, which takes into account co-jumps of the underlying prices and volatility, non-affineness, and self-excitation in Section 2.1 and discuss the pricing of variance swaps in Section 2.2.

2.1. Stock price and variance dynamics

Under a given probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and the complete filtration $\{\mathcal{F}_t\}_{t\geq 0}$, the underlying stock price, S_t, is assumed to follow the stochastic differential equation of:

$$dS_t/S_t = \mu_t dt + \sqrt{V_t} dW_t + \int_R (e^x - 1)\tilde{\pi}(dx, dt), \tag{1}$$

where μ_t is the instantaneous mean rate, W_t is a standard Brownian motion, V_t captures instantaneous diffusion variance, and the last term accounts for any price jumps with the return jump size, x, defined in the real line, R, through a compensated jump measure, $\tilde{\pi}(dx, dt) = \pi(dx, dt) - \nu_t(dx)dt$, in which $\pi(dx, dt)$ is a random counting measure and $\nu_t dt$ its compensator.

Eq. (1) is a generic model for the stock price dynamics and indicates that the underlying price change consists of two orthogonal martingales: a purely continuous component and a purely discontinuous jump component. Following the literature, we take the jump compensator to be of the form of $\nu_t(x) = \frac{\lambda_t}{\sqrt{2\pi}\sigma_J} \exp\{-\frac{(x-\mu_J)^2}{2\sigma_J^2}\}$, indicating that the jump component follows a Compound Poisson process, where the number of jumps, N_t, arriving at any time interval, t, follows a Poisson process with the intensity λ_t, and the jump size, X_t, is identically and independently normally distributed with mean μ_J and variance σ_J^2, *i.e.*, $X_t \sim N(\mu_J, \sigma_J^2)$.

Return variance is stochastic, clustered, and mean-reverting. Moreover, recent empirical studies find that a big jump in asset prices tends to be associated with an abrupt move in asset volatility, indicating co-jumps of prices and volatility (Eraker, 2004; Eraker et al., 2003; Jacod and Todorov, 2010; Todorov and Tauchen, 2011). It is also found that an extreme movement in markets tends to be followed by another extreme movement, resulting in self-exciting jump clustering (Ait-Sahalia et al., 2015; Carr and Wu, 2017; Fulop et al., 2015). Furthermore, the literature documents that affine processes for volatility are misspecified (Ait-Sahalia and Kimmel, 2007; Amengual and Xiu, 2018; Christoffersen et al., 2010; Jones, 2003). We therefore propose to model the instantaneous diffusion variance, V_t, and the jump intensity, λ_t, as follows,

$$dV_t = \kappa_v(\theta_v - V_t)dt + \sigma_v V_t^{\eta_1/2} dZ_{v,t} + dJ_{v,t}, \tag{2}$$

$$d\lambda_t = \kappa_\lambda(\theta_\lambda - \lambda_t)dt + \sigma_\lambda \lambda_t^{\eta_2/2} dZ_{\lambda,t} + \beta dN_t. \tag{3}$$

Eq. (2) indicates that diffusion variance follows a mean-reversion jump diffusion process, where $Z_{v,t}$ is a standard Brownian motion and is allowed to be correlated with W_t in Eq. (1), $E[dW_t, dZ_t^v] = \rho dt$, to accommodate the diffusion leverage effect, and $J_{v,t}$ is a compound Poisson process whose Lévy density is given by $\nu_{v,t}(x) = \lambda_t \frac{e^{-x/\mu_v}}{\mu_v}$, indicating that diffusion variance jumps at the same time as stock returns with the same jump intensity, λ_t, and its jump size is independently exponentially distributed with mean μ_v. The jump intensity, λ_t, follows a self-exciting process, as indicated in Eq. (3), where $Z_{\lambda,t}$ is an independent standard Brownian motion, and N_t is the same Poisson process as in stock price jumps and diffusion volatility jumps.

The above model (hereafter Model I) has analytical conditional expectations of the variance components. The conditional expectation of the jump intensity can be found as follows:

$$E_0[\lambda_t] = e^{-(\kappa_\lambda - \beta)t}\lambda_0 + (1 - e^{-(\kappa_\lambda - \beta)t})\frac{\kappa_\lambda \theta_\lambda}{\kappa_\lambda - \beta}, \tag{4}$$

from which its long-run mean can be obtained by letting $t \to +\infty$,

$$\bar{\lambda} = \frac{\kappa_\lambda \theta_\lambda}{\kappa_\lambda - \beta}. \tag{5}$$

Solutions (4) and (5) indicate that the conditional expectation of the jump intensity is a weighted average between the current intensity, λ_0, and its long-run mean, $\bar{\lambda}$, and the speed of mean-reversion is controlled by $\kappa_\lambda - \beta$. Using $E_0[\lambda_t]$ in Eq. (4) and $\bar{\lambda}$ in Eq. (5), the conditional expectation of diffusion variance can also be found as follows:

$$E_0[V_t] = e^{-\kappa_v t}V_0 + (1 - e^{-\kappa_v t})\theta_v + \mu_v\Big[\frac{1 - e^{-\kappa_v t}}{\kappa_v}\bar{\lambda} + \frac{e^{-(\kappa_\lambda - \beta)t} - e^{-\kappa_v t}}{\kappa_\lambda - \beta - \kappa_v}(\bar{\lambda} - \lambda_0)\Big], \tag{6}$$

and its long-run mean is given by

$$\bar{V} = \theta_v + \frac{\mu_v}{\kappa_v}\bar{\lambda}. \tag{7}$$

The conditional expectation of diffusion variance consists of two parts, one arising from the continuous part (the first two terms on the right-hand side in Eq. (6)) and the other from the jump component (the last term on the right-hand side in Eq. (6)).

Under the above model specification, quadratic variation of the underlying returns over time $[t, T]$ is given by

$$\begin{aligned} QV_{t,T} &= \int_t^T V_s ds + \int_t^T \int_R x^2 \pi(ds, dx) \\ &= \int_t^T V_s ds + \int_t^T \int_R x^2 \nu_t(dx) ds + \int_t^T \int_R x^2 \tilde{\pi}(ds, dx), \end{aligned} \tag{8}$$

based on which the $\mathbb{P}$-expectation of quadratic variation can be quickly obtained as follows:

$$\begin{aligned} E_t^{\mathbb{P}}[QV(t,T)] &= \int_t^T E_t^{\mathbb{P}}[V_s] ds + \int_R \int_t^T x^2 E_t^{\mathbb{P}}[\nu_t(ds, dx)] \\ &= \int_t^T E_t^{\mathbb{P}}[V_s] ds + Var_J \int_t^T E_t^{\mathbb{P}}[\lambda_s] ds \\ &= [A(T-t) + B(T-t)V_t + C(T-t)\lambda_t](T-t), \end{aligned} \tag{9}$$

where $Var_J = \mu_J^2 + \sigma_J^2$, and $A(T-t)$, $B(T-t)$, and $C(T-t)$ are all in closed-forms (see Appendix A).

To explore the dynamic structure of return variance, we also investigate the following restricted models:

- Model II: both diffusion variance and jump intensity follow non-affine specifications, but there is no self-exciting effect, *i.e.*, $\beta = 0$.
- Model III: both diffusion variance and jump intensity follow affine specifications, and there is a self-exciting effect, *i.e.*, $\eta_1 = \eta_2 = 1$.
- Model IV: both diffusion variance and jump intensity follow affine specifications, and there is no self-exciting effect, *i.e.*, $\eta_1 = \eta_2 = 1$, and $\beta = 0$.

2.2. The risk-neutral dynamics and variance swap pricing

Following Pan (2002), Eraker (2004), and Broadie et al. (2007), we employ a class of models for the stochastic discount factor, M_t, such that the change-of-measure from the objective measure, $\mathbb{P}$, to the risk-neutral measure, $\mathbb{Q}$, does not alter the model structure. Specifically,

$$\begin{aligned} M_t &= \exp\Big(-\int_0^t r_s ds\Big)\mathcal{E}\Big(-\int_0^t \gamma_W(s) dW_s\Big)\mathcal{E}\Big(-\int_0^t \gamma_v(s) dZ_{v,s}\Big)\mathcal{E}\Big(-\int_0^t \gamma_\lambda(s) dZ_{\lambda,s}\Big) \\ &\quad \prod_{j=1}^{N_t} \exp\left\{\frac{\mu_J^2 - (\mu_J^{\mathbb{Q}})^2}{2\sigma_J^2} + \frac{\mu_J^{\mathbb{Q}} - \mu_J}{\sigma_J^2} J_{s,j} + \frac{\mu_v^{\mathbb{Q}} - \mu_v}{\mu_v \mu_v^{\mathbb{Q}}} J_{v,j}\right\}, \end{aligned} \tag{10}$$

where r_t is the risk-free rate of interest, and $\mathcal{E}(\cdot)$ denotes the stochastic (*Doleans-Dade*) exponential operator. The prices for diffusive risks, $Z_{v,t}$ and $Z_{\lambda,t}$, are assumed to take the forms of $\gamma_v(t) = \gamma_v V_t^{1-\eta_1/2}$ and $\gamma_\lambda(t) = \gamma_\lambda \lambda_t^{1-\eta_2/2}$, respectively, where γ_v and γ_λ are constants. For the jump components, the above change of measure indicates that $\nu_t^{\mathbb{Q}}(dx) = \frac{\lambda_t}{\sqrt{2\pi}\sigma_J} \exp\{-\frac{(x-\mu_J^{\mathbb{Q}})^2}{2\sigma_J^2}\}$ and $\nu_{v,t}^{\mathbb{Q}}(x) = \lambda_t \frac{e^{-x/\mu_v^{\mathbb{Q}}}}{\mu_v^{\mathbb{Q}}}$. In contrast, we leave $\gamma_W(t)$ unspecified as our main purpose in this paper is to investigate volatility dynamics.

The pricing kernel, M_t, in Eq. (10) indicates the following risk-neutral model,

$$dS_t/S_t = r_t dt + \sqrt{V_t} dW_t^{\mathbb{Q}} + \int_R (e^x - 1)\tilde{\pi}^{\mathbb{Q}}(dx, dt), \tag{11}$$

$$dV_t = \kappa_v^{\mathbb{Q}}(\theta_v^{\mathbb{Q}} - V_t)dt + \sigma_v V_t^{\eta_1/2} dZ_{v,t}^{\mathbb{Q}} + dJ_{v,t}^{\mathbb{Q}}, \tag{12}$$

$$d\lambda_t = \kappa_\lambda^{\mathbb{Q}}(\theta_\lambda^{\mathbb{Q}} - \lambda_t)dt + \sigma_\lambda V_t^{\eta_2/2} dZ_{\lambda,t}^{\mathbb{Q}} + \beta dN_t, \tag{13}$$

where $\kappa_v^{\mathbb{Q}} = \kappa_v + \sigma_v\gamma_v$, $\theta_v^{\mathbb{Q}} = \kappa_v\theta_v/\kappa_v^{\mathbb{Q}}$, $\kappa_\lambda^{\mathbb{Q}} = \kappa_\lambda + \sigma_\lambda\gamma_\lambda$, and $\theta_\lambda^{\mathbb{Q}} = \kappa_\lambda\theta_\lambda/\kappa_\lambda^{\mathbb{Q}}$. The risk-neutral model has exactly the same structure as the objective one. We therefore can also obtain the risk-neutral conditional expectations of V_t and λ_t in

closed-form similar to Eqs. (6) and (4), and the $\mathbb{Q}$-expectation of quadratic variation has similar form as Eq. (9)

$$\begin{aligned} E_t^{\mathbb{Q}}[QV(t,T)] &= \int_t^T E_t^{\mathbb{Q}}[V_s]ds + Var_J^{\mathbb{Q}} \int_t^T E_t^{\mathbb{Q}}[\lambda_s]ds \\ &= [A^{\mathbb{Q}}(T-t) + B^{\mathbb{Q}}(T-t)V_t + C^{\mathbb{Q}}(T-t)\lambda_t](T-t), \end{aligned} \tag{14}$$

where $A^{\mathbb{Q}}(T-t)$, $B^{\mathbb{Q}}(T-t)$, and $C^{\mathbb{Q}}(T-t)$ are exactly in the same forms as $A(T-t)$, $B(T-t)$, and $C(T-t)$, respectively, except that the corresponding objective parameters are substituted by the risk-neutral parameters.

In fact, the annualized $\mathbb{Q}$-expectation of quadratic variation is the price of a variance swap (VS) with the time to maturity $T-t$,

$$VS_{t,T} = \frac{1}{T-t} E_t^{\mathbb{Q}}[QV_{t,T}]. \tag{15}$$

A variance swap is a type of derivative that allows investors to trade variance as an asset class. At maturity, one leg of the swap pays an amount based upon realized variance, and the other leg pays a fixed amount, which is called the variance swap rate, quoted at the inception. This contract has zero market value at entry and the payoff at maturity of the long position is equal to the difference between realized variance and the variance swap rate. Therefore, under the no-arbitrage condition, the variance swap rate should be the risk-neutral expected value of quadratic variation.

3. An efficient particle-based method

3.1. State-space representation

Our model can be cast into a state-space model framework. In accordance with the existing literature (e.g., Johannes and Polson, 2009; Johannes et al., 2009), we discretize the return process for a small time interval τ using the Euler method and obtain the following observation equation for stock prices

$$\ln S_t = \ln S_{t-1} + \left(\mu - \frac{1}{2}V_{t-1} - k(1)\lambda_{t-1}\right)\tau + X_t \Delta N_t + \sqrt{\tau V_{t-1}}w_t \tag{16}$$

where $k(1) = e^{\mu_J + \sigma_J^2/2} - 1$ is the convexity adjustment for the jump component, w_t is a standard normal noise, X_t is the return jump size, which follows a normal distribution with mean μ_J and variance σ_J^2, and the increment in the jump counter is approximated by a Bernoulli variable, *i.e.*, $\Delta N_t \equiv N_t - N_{t-1} \sim Bernoulli(\lambda_{t-1}\tau)$.

Variance swap rates with n maturities are also available. By assuming that variance swap rates are collected with measurement errors, we have additional observation equations

$$VS_{t,\tilde{\tau}}^{\mathcal{O}} = VS_{t,\tilde{\tau}}^{\mathcal{M}} + \epsilon_t, \tag{17}$$

where $VS_{t,T}^{\mathcal{O}}$ is a vector of market observed variance swap rates at time t with time to maturity $\tilde{\tau}$, and $VS_{t,\tilde{\tau}}^{\mathcal{M}}$ is the corresponding rates implied by the model and computed using the formula (15). The measurement errors, ϵ_t, are assumed to be a multivariate normal with a mean vector of zero and a variance–covariance matrix of Ω, $\epsilon_t \sim N(0, \Omega)$, where $\Omega = \Sigma\Sigma'$ and Σ is a $n \times n$ lower triangular matrix. A special case of Σ, which is often used in literature, is $\Sigma = \sigma_e I_n$, where σ_e is a constant and I_n is a $n \times n$ identity matrix.

We regard diffusion variance, V_t, jump intensity, λ_t, the increment of the jump counter, ΔN_t, and variance jump size, $J_{v,t}$, as state variables. The discretized dynamics of V_t and λ_t are given by

$$V_t = V_{t-1} + \kappa_v(\theta_v - V_{t-1})\tau + \sigma_v V_{t-1}^{\eta_1/2}\sqrt{\tau}z_{v,t} + J_{v,t}\Delta N_t, \tag{18}$$

$$\lambda_t = \lambda_{t-1} + \kappa_\lambda(\theta_\lambda - \lambda_{t-1})\tau + \sigma_\lambda \lambda_{t-1}^{\eta_2/2}\sqrt{\tau}z_{\lambda,t} + \beta\Delta N_t, \tag{19}$$

where $z_{v,t}$ is a standard normal noise, which is correlated to w_t in Eq. (16) with a correlation parameter ρ, $J_{v,t}$ is exponentially distributed with mean parameter μ_v, and $z_{\lambda,t}$ is an independent standard normal noise.

Denote the set of model parameters as Θ and all observations and the latent states as $y_{1:T} = \{\ln S_t, VS_{t,\tilde{\tau}}^{\mathcal{O}}\}_{t=1}^T$, and $x_{1:T} = \{V_t, \lambda_t, \Delta N_t, J_{v,t}\}_{t=1}^T$, respectively. Our aim is to find the joint posterior distribution of the parameters and latent states, $p(\Theta, x_{1:T}|y_{1:T})$, which can be decomposed into

$$p(\Theta, x_{1:T}|y_{1:T}) = p(x_{1:T}|\Theta, y_{1:T})p(\Theta|y_{1:T}), \tag{20}$$

where $p(x_{1:T}|\Theta, y_{1:T})$ solves the state smoothing issue, and $p(\Theta|y_{1:T})$ addresses the parameter inference problem.

Markov chain Monte Carlo (MCMC) methods could be used to estimate the above joint posterior distributions (Eraker, 2004; Eraker et al., 2003; Li et al., 2008; Johannes and Polson, 2009; Yu et al., 2011). Typically, these algorithms use Gibbs sampling to move the latent states conditionally on the fixed parameters and vice-versa. While derivatives data are informative on the latent states, they pose problems to such MCMC methods. In particular, derivative observations introduce a tight link between the dynamic states and the fixed parameters, leading to high autocorrelation in the chain and very slow mixing. Furthermore, the variance swap rate pricing function is a complicated non-linear function of the parameters, causing

a loss of sufficient statistics and extra dependence between parameters. Hence, a generic good proposal over the parameters is hard to design.

In what follows, we come up with a particle-based method, which aims to solve these issues. We first design a fully adapted particle filter and then propose a new two-stage sequential Monte Carlo (SMC) sampler routine that depends on smoothing the particle filter and introducing common random numbers (CRNs). For the class of asset pricing models considered in the paper and for variance swap pricing, we can find the locally optimal particle filter, even though this class of models is very general and includes many models found in literature as special cases. However, we need to emphasize that our two-stage SMC sampler does not necessarily depend on availability of the optimal particle filter and any relatively efficient particle filters can work together with this sampler (see Internet Appendix for an illustration). Furthermore, different from the commonly used MCMC methods, our approach can be easily parallelized, making it computationally efficient and convenient to use in practice.

3.2. A fully adapted auxiliary particle filter

The above state-space model is clearly non-linear and non-Gaussian. State filtering can therefore be efficiently implemented using particle filters given the static parameters. For notational convenience, dependence on Θ is suppressed in most of this subsection. The basic idea is to represent distributions of all random variables with a number of particles drawn directly from the state space and to approximate the posterior density $p(x_t|y_{1:t})$ with the empirical point-mass estimate $\hat{p}(x_t|y_{1:t})$.

Given M samples $\{x_{t-1}^{(i)}; i = 1, 2, \ldots, M\}$ representing the filtering density $p(x_{t-1}|y_{1:t-1})$ at time $t-1$, the recursion

$$p(x_t|y_{1:t}) \propto \int p(y_t|x_t)p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \tag{21}$$

suggests the following importance sampling strategy. First, draw from a known and easily sampled proposal density function, $m_t(x_t|x_{t-1})$. Second, attach importance weights w_t to account for the difference between the target and the proposal,

$$w_t^{(i)} = \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})}{m_t(x_t^{(i)}|x_{t-1}^{(i)})}. \tag{22}$$

The normalized weights are given by $W_t^{(i)} = w_t^{(i)}/\sum_{j=1}^{M} w_t^{(j)}$. Finally, as a solution to the particle degeneracy problem, focus on the most likely particles by resampling the population proportional to W_t if the effective sample size, $ESS = 1/\sum_{i=1}^{M}(W_t^{(i)})^2$, is below some threshold. Let a_t denote the sampled indices. Thus, we obtain the equally weighted sample representing the filtering distribution:

$$\hat{p}(x_t|y_{1:t}) = \frac{1}{M}\sum_{i=1}^{M} \delta_{x_t^{(a_t^{(i)})}}\left(x_t\right). \tag{23}$$

Particle filters provide an estimate of the likelihood of the observations

$$\hat{p}(y_{1:t}|\Theta) = \prod_{l=2}^{t} \hat{p}(y_l|y_{1:l-1}, \Theta)\hat{p}(y_1|\Theta), \tag{24}$$

where

$$\hat{p}(y_l|y_{1:l-1}, \Theta) = \frac{1}{M}\sum_{i=1}^{M} w_l^{(i)}. \tag{25}$$

Importantly, the likelihood estimate in Eq. (24) approximated by particle filters is unbiased, i.e., $E[\hat{p}(y_{1:t}|\Theta)] = p(y_{1:t}|\Theta)$, where the expectation is taken with respect to all random quantities used in particle filters (Del Moral, 2004).

The most commonly used particle filter is the bootstrap filter of Gordon et al. (1993). The bootstrap filter simply takes the state transition law as the proposal, that is, $m_t(x_t|x_{t-1}) = p(x_t|x_{t-1})$, which does not take into account the latest information, leading to poor performance when the observations are informative on the hidden states. Our model has this feature as derivatives contain rich information on volatility and jumps can be pinned down by large moves in asset prices. Doucet et al. (2000) show that the optimal proposal density, which is conditional on the past states and the new observations and minimizes variance of importance weights, has the form of $m_t^*(x_t|x_{t-1}) = p(x_t|x_{t-1}, y_t)$. Fortunately, for the class of asset pricing models considered in the paper and for variance swap pricing, we are able to derive such a (locally) optimal proposal density in an analytical form.

Consider the joint density of the current observations and states conditional on the past states, $p(x_t, y_t|x_{t-1})$, which is a key component in the recursion of Eq. (21):

$$\begin{aligned} p(x_t, y_t|x_{t-1}) &= p(x_t|x_{t-1}, y_t)p(y_t|x_{t-1}) \\ &= p(V_t, \lambda_t, J_{v,t}, \Delta N_t \mid V_{t-1}, \lambda_{t-1}, y_t)p(y_t \mid \lambda_{t-1}, V_{t-1}) \end{aligned}$$

$$= p(V_t, \lambda_t \mid J_{v,t}, \Delta N_t, \lambda_{t-1}, V_{t-1}, y_t)p(J_{v,t} \mid \Delta N_t, \lambda_{t-1}, V_{t-1}, y_t) \times P(\Delta N_t \mid \lambda_{t-1}, V_{t-1}, y_t)p(y_t \mid \lambda_{t-1}, V_{t-1}), \tag{26}$$

which suggests that we need to reweight the past particles using the predictive density of the current observations given the past states, $p(y_t|x_{t-1})$, and then sample from the new states using the optimal proposal density, $p(x_t|x_{t-1}, y_t)$. In what follows, we sketch out how to find each term on the right-hand side of Eq. (26). The detailed derivation is given in Appendix B.1.

To implement, the first object to compute is $p(y_t \mid \lambda_{t-1}, V_{t-1})$. Notice that given the jump indicator, ΔN_t, the variance jumps, $J_{v,t}$, and the past state variables, λ_{t-1} and V_{t-1}, we have a linear observation equation that effectively allows us to temporally marginalize out λ_t and V_t:

$$y_t = C_{t-1} + D_{t-1}J_{v,t} + \eta_t, \qquad \eta_t \sim N(0, \Sigma_{t-1}), \tag{27}$$

where the coefficients C_{t-1}, D_{t-1}, and Σ_{t-1} depend only on λ_{t-1}, V_{t-1}, and ΔN_t. We then write the predictive likelihood of the next observation given the past states as follows:

$$p(y_t \mid \lambda_{t-1}, V_{t-1}) = \sum_{i=0,1} p(y_t \mid \Delta N_t = i, \lambda_{t-1}, V_{t-1})P(\Delta N_t = i \mid \lambda_{t-1}, V_{t-1}), \tag{28}$$

where $P(\Delta N_t = i \mid \lambda_{t-1}, V_{t-1})$ are trivial and $p(y_t \mid \Delta N_t = 0, \lambda_{t-1}, V_{t-1})$ is simply a normal density. Hence the only complication is to compute $p(y_t \mid \Delta N_t = 1, \lambda_{t-1}, V_{t-1})$ where the variance jump, $J_{v,t}$, needs to be integrated out:

$$p(y_t \mid \Delta N_t = 1, \lambda_{t-1}, V_{t-1}) = \int_0^\infty p(y_t \mid J_{v,t}, \Delta N_t = 1, \lambda_{t-1}, V_{t-1}) \times p(J_{v,t} \mid \Delta N_t = 1, \lambda_{t-1}, V_{t-1})dJ_{v,t}, \tag{29}$$

which can be solved analytically. Furthermore, we can also show that the conditional distribution of the variance jumps, $J_{v,t}$, is normal truncated to the positive domain with mean of $\mu_{J_{v,t}}$ and variance of $\sigma^2_{J_{v,t}}$.

Now sampling from the conditional distribution of the jump variables is trivial. First, the jump indicator is a Bernoulli random variable with probabilities:

$$P(\Delta N_t = i \mid y_t, \lambda_{t-1}, V_{t-1}) = \frac{p(y_t|\Delta N_t = i, \lambda_{t-1}, V_{t-1})P(\Delta N_t = i|\lambda_{t-1}, V_{t-1})}{p(y_t \mid \lambda_{t-1}, V_{t-1})}, \tag{30}$$

for $i = 0,1$. Second, the conditional distribution of the variance jumps is a mixture of a Dirac mass on zero and a truncated normal in the positive domain,

$$p(J_{v,t} \mid \Delta N_t, y_t, \lambda_{t-1}, V_{t-1}) = \delta_{\{J_{v,t}=0\}}, \text{ if } \Delta N_t = 0, \tag{31}$$

$$p(J_{v,t} \mid \Delta N_t, y_t, \lambda_{t-1}, V_{t-1}) \sim N_{J_{v,t}>0}(\mu_{J_{v,t}}, \sigma_{J_{v,t}}), \text{ if } \Delta N_t = 1. \tag{32}$$

Finally, given that conditional on $J_{v,t}$, ΔN_t, λ_{t-1}, and V_{t-1}, we have a locally linear Gaussian state space system, the conditional distribution of λ_t and V_t, $p(\lambda_t, V_t \mid J_{v,t}, \Delta N_t, y_t, \lambda_{t-1}, V_{t-1})$, is simply a Gaussian distribution with moments that can be easily obtained by a Kalman recursion.

Now we can implement a fully adapted auxiliary particle filter (APF) as in Pitt and Shephard (1999) using the recursion implied by Eq. (26). In contrast to standard particle filters, to increase efficiency, we first resample the past states given the new observations and then sample the new state variables. The derivation of the fully adapted APF over the exponential variance jumps is new in literature and should be interesting in its own right given the frequent use of similar model specifications in finance literature.[1] We name this particle filter as the exact auxiliary particle filter (Exact-APF), whose detailed algorithm is presented in Appendix B.1.[2]

3.3. A new two-stage SMC sampler

According to Bayes' rule, the posterior distribution of the model parameter is given by

$$p(\Theta|y_{1:T}) \propto p(y_{1:T}|\Theta)p(\Theta), \tag{33}$$

where on the right-hand side, the first term is the likelihood and the second is the prior. The decomposition suggests a hierarchical framework to the problem that targets at the posterior distribution of the fixed parameters: for a given set of model parameters proposed from some proposal, we can run a particle filter, which delivers the estimate of the likelihood, $p(y_{1:T}|\Theta)$, and empirical distribution of the hidden states, $p(x_{1:T}|\Theta, y_{1:T})$.

[1] For a stochastic volatility jump–diffusion model, which excludes self-exciting jump intensity and non-affineness and is a special case of our model, Johannes et al. (2009) propose a partially adapted APF for volatility extraction.

[2] The term "exact" here refers to the fact that this filter provides an unbiased estimate to the likelihood of the data given the fixed parameters.

3.3.1. Pseudo-marginalization revisited

Having built on the decomposition of Eq. (33), Andrieu and Roberts (2009) and Andrieu et al. (2010) have pioneered the pseudo-marginalization approach for parameter inference. The main idea is as follows. Define an auxiliary variable u_t, which includes all the random variables produced by a particle filter at time t such as the state particles and resampling index. Denote the joint posterior distribution of auxiliary variables and the fixed parameters, Θ, as $\tilde{p}(\Theta, u_{1:T}|y_{1:T})$. The unbiasedness property of the likelihood estimate from a particle filter indicates that this joint posterior distribution admits the target, $p(\Theta|y_{1:T})$, as a marginal

$$\int_{u_{1:T}} \tilde{p}(\Theta, u_{1:T}|y_{1:T})du_{1:T} = p(\Theta|y_{1:T}). \tag{34}$$

Now instead of $p(\Theta|y_{1:T})$, we focus on the joint posterior distribution $\tilde{p}(\Theta, u_{1:T}|y_{1:T})$. Building on the tempered sequential Monte Carlo samplers of Del Moral et al. (2006), Duan and Fulop (2015) suggest a new sequential importance sampling algorithm targeting $\tilde{p}(\Theta, u_{1:T}|y_{1:T})$. The key point is to begin with an easy-to-sample distribution and traverse through a sequence of densities to the ultimate target. Construct a sequence of I densities between the extended prior $\pi_1(\Theta, u_{1:T})$ and the posterior $\pi_I = \tilde{p}(\Theta, u_{1:T}|y_{1:T})$ using a tempering sequence ξ_i, $i = 1, 2, \ldots, I$, for $\xi_1 = 0$, $\xi_I = 1$, and

$$\pi_i(\Theta, u_{1:T}) = \frac{\gamma_i(\Theta, u_{1:T})}{Z_i}, \tag{35}$$

$$\gamma_i(\Theta, u_{1:T}) = \hat{p}_i(y_{1:T}|\Theta, u_{1:T})^{\xi_i}\tilde{p}_i(u_{1:T}|\Theta)p(\Theta), \tag{36}$$

where $Z_i = \int \gamma_i(\Theta, u_{1:T})d(\Theta, u_{1:T})$ is a normalizing constant, $\hat{p}_i(y_{1:T}|\Theta, u_{1:T})$ is the estimated likelihood from a particle filter, and $\tilde{p}_i(u_{1:T}|\Theta)$ is the prior distribution of the auxiliary variables.

Moving from $\pi_i(\Theta, u_{1:T})$ to $\pi_{i+1}(\Theta, u_{1:T})$ can be implemented by *reweighting* each parameter particle by $[\hat{p}_i(y_{1:T}|\Theta^{(n)}, u_{1:T}^{(n)})]^{\xi_{i+1}-\xi_i}$, for $n = 1, \ldots, N$. The tempering coefficients, ξ_is, are chosen adaptively to ensure satisfactory particle diversity. With repeatedly reweighting and resampling, the support of the parameter particles would gradually deteriorate, leading to the well-known particle impoverishment problem. To solve this problem, periodically boosting the support becomes a must. Del Moral et al. (2006) suggest to first sample N points from $\gamma_1(\Theta, u_{1:T}) \equiv p(\Theta)$, which is the prior, and then recursively sample from $\gamma_i(\Theta, u_{1:T})$ by *moving* these points using some MCMC kernels with the stationary distribution being $\gamma_i(\Theta, u_{1:T})$. This algorithm provides consistent inference to $p(\Theta|y_{1:T})$ for any given number of state particles, M, as the number of parameter particles, N, goes to infinity.

Since the seminal works of Andrieu and Roberts (2009) and Andrieu et al. (2010), there has been a huge amount of interest in the pseudo-marginalization approach that shows that even in the presence of the Monte Carlo error in the likelihood estimate, the simulation schemes have the right equilibrium distribution for any finite number of the state particles, M, and hence allow us to conduct inference on the exact posterior. Andrieu and Roberts (2009) call this the "exact approximation" property. However, it is well understood that the effective control of the noise in the likelihood estimate is necessary for the adequate mixing of such routines and for their success. In particular, Andrieu et al. (2010) show that M should increase linearly with the sample, T, and Doucet et al. (2015) argue that under a wide set of assumptions, the standard deviation of the log-likelihood estimate should be optimally kept around 1 for the pseudo-marginal MCMC algorithms. Unfortunately, due to the informative derivatives observations and potential model misspecification in the real data, we find that the noise of the likelihood estimate from Exact-APF remains large (see Table 2) even for a large M, making reliable inference over the fixed parameter through the pseudo-marginalized approach hard.

The reason why the pseudo-marginalized algorithms are sensitive to the Monte Carlo error is that the noise in the likelihood estimate can starkly change even when the underlying theoretical likelihood values are close. In particular, assume that the algorithm attempts to move from some fixed parameter, Θ, to a close-by value, $\Theta' = \Theta + \Delta\Theta$. Then, the mixing of the Metropolis–Hastings routine is determined by the estimation noise in the log likelihood difference, $\ln \hat{p}(y_{1:T}|\Theta') - \ln \hat{p}(y_{1:T}|\Theta)$. Given that in the original pseudo-marginal algorithms the random numbers used to estimate the likelihood are independent across the different fixed parameter values, a large error in $\ln \hat{p}(y_{1:T}|\Theta)$ and $\ln \hat{p}(y_{1:T}|\Theta')$ implies a large error in their difference, leading to breakdown of the algorithm. Therefore, to deal with this issue, we need to create strong positive correlation between the Monte Carlo errors of the likelihood estimates across different parameter sets.

3.3.2. Introducing common random numbers and smoothing the likelihood function

The first component in our strategy to ensure strong correlation between the Monte Carlo errors across different parameter sets is the use of common random numbers.[3] In particular, let us assume for now that the likelihood, $\ln \hat{p}(y_{1:T}|\Theta, u)$, is a reasonably smooth function of the model parameters Θ. Here u denotes the set of uniform random numbers entering the likelihood estimation routine. We then propose to use exactly the same random inputs to estimate the likelihood function at all values of Θ, instead of independent draws, and the log-likelihood differences can be written as $\ln \hat{p}(y_{1:T}|\Theta', u) - \ln \hat{p}(y_{1:T}|\Theta, u)$. Provided the smoothness of the likelihood estimate, the use of common random numbers makes $\ln \hat{p}(y_{1:T}|\Theta', u)$ and $\ln \hat{p}(y_{1:T}|\Theta, u)$ highly positively correlated, and therefore drastically reduces the variance of

[3] Common random numbers have been used within particle filtering in a smooth particle filter of Malik and Pitt (2011). For a review on their use in simulated maximum likelihood, see Gourieroux and Monfort (1996).

$\ln \hat{p}(y_{1:T}|\Theta', u) - \ln \hat{p}(y_{1:T}|\Theta, u)$. Obviously, this means that we depart from the pseudo-marginalized framework and our inference will be subject to extra Monte Carlo noise due to the estimation error in the likelihood.

A necessary condition to ensure that the error in $\ln \hat{p}(y_{1:T}|\Theta, u)$ is strongly correlated for close-by values of Θ for fixed u is that the likelihood estimate should be a relatively smooth function of Θ. Unfortunately, it is well known that the resampling step in particle filters precludes such a property. For concreteness, we describe the problem in the context of our auxiliary particle filter, but the issue is analogous for other particle filtering algorithms. To understand the problem, notice that each step of the sequential Monte Carlo routine for $t \geq 1$ amounts sampling from the target distribution

$$G_t^M(x_t, \tilde{x}_{t-1}) = \sum_{i=1}^{M} W_{t-1|t}^{(i)} \delta_{x_{t-1}^{(i)}}\left(\tilde{x}_{t-1}\right) p(x_t \mid y_t, \tilde{x}_{t-1}). \tag{37}$$

Sampling from $G_t^M(x_t, \tilde{x}_{t-1})$ implicitly involves a non-smooth simulator due to resampling even if $p(x_t \mid y_t, \tilde{x}_{t-1})$ can be smoothly sampled. Essentially the problem is that as we vary the inputted uniforms in the resampling algorithms, the output will be jumping between the different $x_{t-1}^{(i)}$ that can be completely different across the different neighboring indices, leading to non-smoothness.

When the state-space is one-dimensional, this problem can be addressed by sorting the particles before resampling them. Malik and Pitt (2011) show that further smoothness can be achieved by using a continuous interpolation instead of the empirical distribution function. There have been recent work on achieving smoother resampling schemes in multi-dimensional state spaces. Gerber and Chopin (2015) generalize the sorting approach to multi-dimensional state-spaces by using space-filling Hilbert curves.[4] Deligiannidis et al. (2018) employ the Hilbert sorting scheme within the context of a correlated pseudo-marginal MCMC algorithm. Jacob et al. (2016) propose maximal coupling and optimal transport methods to provide pairs of particle filters with coupled resampling.

In this paper, we take an alternative approach and resort to a special feature of the class of models we are concerned with. In particular, in our case the observations are very informative on the latent state, indicating that the smoothing density is concentrated on a small domain around the smoothed mean of the state. Therefore, instead of using the target in Eq. (37), we use the following approximate target

$$G_t^A(x_t, \tilde{x}_{t-1}) = N(\tilde{x}_{t-1} \mid \mu_{t-1|t}, V_{t-1|t}) f(x_t \mid y_t, \tilde{x}_{t-1}), \tag{38}$$

where $N(x|\mu_{t-1|t}, V_{t-1|t})$ is a multivariate normal density with mean, $\mu_{t-1|t}$, and variance, $V_{t-1|t}$, equal to the smoothed moments computed from the particles representing the smoothing density $f(x_{t-1} \mid y_{1:t})$. Naturally, there is a bias due to the use of the approximate target in Eq. (38) instead of Eq. (37). However, in models with informative observations, the filtering density is localized around its mean and this bias is likely to be very small. The main advantage of this approximation is that it provides a smooth mapping. The Monte Carlo study in Section 4 suggests that the bias due to this approximation is minor and its effect on parameter estimation is negligible. The detailed algorithm of this smooth particle filter (Smooth-PF) is presented in Appendix B.2.

3.3.3. A new approach

Now we introduce a new tempered SMC sampler that uses common random numbers in the likelihood estimation. Instead of randomizing over the auxiliary variables, fix some random numbers, $u_{1:T}^M = \{u_{1:T}^l; l = 1, \ldots, M\}$, to be used in the filter and denote the likelihood estimate as $\hat{p}(y_{1:T}|\Theta, u_{1:T}^M)$ for the given number of state particles, M. Throughout, we run the SMC sampler over the fixed parameters while keeping the random numbers fixes across the parameter sets. We propose a two-stage algorithm, where the first stage traverses from the prior to a posterior with a likelihood estimated with some number of state particles, M_1, and the second stage builds a bridge of densities to a posterior where the likelihood is estimated with some number of state particles, $M_2 \gg M_1$. Take the corresponding two sets of common random numbers, $u_{1:T}^{M_k}$, $k = 1, 2$, as fixed.

First, we construct a sequence of I_1 densities between the prior $\pi_{1,1} = p(\Theta)$ and the posterior $\pi_{1,I_1} = \hat{p}(y_{1:T}|\Theta, u_{1:T}^{M_1})p(\Theta)$ using a tempering sequence $\xi_{1,i}$, $i = 1, 2, \ldots, I_1$, for $\xi_{1,1} = 0$, $\xi_{1,I_1} = 1$, and

$$\pi_{1,i}(\Theta) = \frac{\gamma_{1,i}(\Theta)}{Z_{1,i}}, \tag{39}$$

$$\gamma_{1,i}(\Theta) = \hat{p}^{\xi_{1,i}}(y_{1:T}|\Theta, u_{1:T}^{M_1})p(\Theta), \tag{40}$$

where $Z_{1,i} = \int \gamma_{1,i}(\Theta)d(\Theta)$. Second, we construct a sequence of I_2 densities between the posterior using M_1 state particles, $\pi_{2,1} = \hat{p}(y_{1:T}|\Theta, u_{1:T}^{M_1})p(\Theta)$ and the posterior using M_2 state particles, $\pi_{2,I_2} = \hat{p}(y_{1:T}|\Theta, u_{1:T}^{M_2})p(\Theta)$ using a tempering sequence $\xi_{2,i}$, $i = 1, 2, \ldots, I_2$, for $\xi_{2,1} = 0$, $\xi_{2,I_2} = 1$, and

$$\pi_{2,i}(\Theta) = \frac{\gamma_{2,i}(\Theta)}{Z_{2,i}}, \tag{41}$$

[4] We would like to thank Mathieu Gerber and Nicolas Chopin for sharing the results of some unpublished experiments on the performance of Hilbert sorting with common random numbers.

$$\gamma_{2,i}(\Theta) = \hat{p}^{\xi_{2,i}}(y_{1:T}|\Theta, u_{1:T}^{M_2}) \times \hat{p}^{1-\xi_{2,i}}(y_{1:T}|\Theta, u_{1:T}^{M_1}) \times p(\Theta), \tag{42}$$

where $Z_{2,i} = \int \gamma_{2,i}(\Theta)d(\Theta)$. Now as before we run an SMC sampler with N parameter particles first through the sequence $\pi_{1,i}(\Theta)$ and then continuing onto $\pi_{2,i}(\Theta)$. In all other respects, we use the same settings as in the pseudo-marginalized routine. When choosing the tempering sequences, $\xi_{1,i}$ and $\xi_{2,i}$, following Del Moral et al. (2012), we set the value of ξ to ensure that the effective sample size (ESS) stays close to some constant. This can be done by a simple grid search, where ESS is evaluated at the grid points of ξ and the one with the ESS closest to this constant is chosen.[5] When moving the parameter particles, to ensure a satisfactory rejuvenation of the support, we keep repeating move step at each density-bridging iteration until the cumulative average acceptance rate across the algorithm achieves a pre-specified constant. See a detailed description of the algorithm in Appendix C.

As a result of this procedure, the algorithm delivers consistent inference over the approximate posterior $\hat{p}(y_{1:T}|\Theta, u_{1:T}^{M_2})$ $p(\Theta)$ as N goes to infinity. Obviously, this depends on the random set $u_{1:T}^{M_2}$, suggesting that there exists an error $\hat{p}(y_{1:T}|\Theta, u_{1:T}^{M_2})$ $p(\Theta) - p(y_{1:T}|\Theta)$ due to the use of common random numbers. This contrasts to the pseudo-marginalized approaches that deliver draws over $p(y_{1:T}|\Theta)$ for a fixed M. However, the crucial advantage of using common random numbers with a smooth likelihood estimator is that the estimation noise in the likelihood does not affect the acceptance rates of the algorithm.

The intuition underlying our two-stage approach is that in the first stage we use a small M_1 for a coarse exploration of the posterior and then in the second stage we decrease the estimation noise by increasing the number of state particles to a much larger value, M_2. Crucially, the role of the second stage is only to correct error in likelihood estimation, mirrored in a much smaller number of necessary density-bridging steps, i.e., we usually have $I_2 \ll I_1$. Hence our two-stage sampler will run much faster compared to an SMC sampler with common random numbers that use M_2 throughout.

Chopin et al. (2013) and Duan and Fulop (2015) also propose multi-stage SMC approaches within the pseudo-marginalization framework that increase the number of state particles, M, as the algorithms advance. There are two important differences between those schemes and ours. First, in the pseudo-marginalization framework, the number of state particles (M) needs to be increased as the sample size grows to keep the acceptance rates away from 0 and make the algorithm stable. In contrast, the stability of the SMC algorithm in our context is not sensitive to the choice of M due to the use of common random numbers. Second, transferring to a larger M is a much simpler task in our case as the use of common random numbers causes strong positive correlation across the weights of the particles in the reweighting steps, leading to a radical decrease of variability of the weights. This is a key reason why in our algorithm the necessary number of density-bridging densities in the second stage is typically very small.

Given the normal approximation in Eq. (38), the true marginal likelihood of the model is unavailable. However, it can be accurately approximated by

$$\begin{aligned} p(y_{1:T}) &= \int p(y_{1:T}|\Theta)p(\Theta)d\Theta \\ &\approx \prod_{i=2}^{I_1}\left[Z_{1,i}/Z_{1,i-1}\right] \times \prod_{i=2}^{I_2}\left[Z_{2,i}/Z_{2,i-1}\right], \end{aligned} \tag{43}$$

which can then be used to construct Bayes factors for model comparison. For any two models, $\mathcal{M}_i$ and $\mathcal{M}_j$, the Bayes factor is given by the ratio of their marginal likelihoods,

$$\mathcal{BF}_{i,j} = \frac{p(y_{1:T}|\mathcal{M}_i)}{p(y_{1:T}|\mathcal{M}_j)}, \tag{44}$$

which, different from the standard classical tests, does not rely on asymptotic distribution theory and provides an intuitive approach to evaluating the relative merits of competing models. It is important to note that the Bayes factor can be used to compare both nested and non-nested models, and furthermore it does not necessarily favor more complex models, as it contains a penalty for using more parameters due to its marginal nature.

4. Monte Carlo studies

4.1. Priors and proposals

Our two-stage SMC sampler is initialized by the prior distributions, hence we need to have proper priors. Given that the likelihood function from particle filters takes a complicated nonlinear function of the fixed parameters in this approach, conjugate priors are not available. Therefore, in this paper, we simply assume normal distributions for the priors, and simulation from the priors becomes straightforward. However, if a parameter under consideration has a bounded support, we take a truncated normal as its prior.

The choice of the hyper-parameters of the prior distributions is based on calibration using data on SP500 index and VIX from January 1997 to December 1999, a period preceding our estimation period, as a training sample to inform locations

[5] In implementation, we generate 1000 equally spaced points between −100 and 0, and their exponentials, which belong to [0, 1], are taken as the grid points.

Table 1
The prior distributions.

Θ	F. Form	Support	(μ_0, σ_0)	Θ	F. Form	Support	(μ_0, σ_0)
μ	Normal	$(-\infty, \infty)$	(0.03, 0.10)	μ_v^Q	Tr. Normal	$(0, \infty)$	(0.05, 0.30)
μ_J	Normal	$(-\infty, \infty)$	(−0.01, 0.15)	κ_λ	Tr. Normal	$(0, \infty)$	(8.00, 15.0)
σ_J	Tr. Normal	$(0, \infty)$	(0.02, 0.15)	θ_λ	Tr. Normal	$(0, \infty)$	(2.00, 8.00)
μ_J^Q	Normal	$(-\infty, \infty)$	(−0.01, 0.15)	σ_λ	Tr. Normal	$(0, \infty)$	(5.00, 15.0)
κ_v	Tr. Normal	$(0, \infty)$	(8.00, 15.0)	η_2	Tr. Normal	$(0, \infty)$	(2.00, 5.00)
θ_v	Tr. Normal	$(0, \infty)$	(0.02, 0.10)	β	Tr. Normal	$(0, \infty)$	(2.00, 5.00)
σ_v	Tr. Normal	$(0, \infty)$	(2.00, 8.00)	γ_v	Tr. Normal	$(-\infty, 0)$	(−6.00, 15.0)
η_1	Tr. Normal	$(0, \infty)$	(2.00, 5.00)	γ_λ	Tr. Normal	$(-\infty, 0)$	(−1.50, 5.00)
ρ	Tr. Normal	$[-1, 1]$	(−0.70, 1.00)	Σ-D	Tr. Normal	$(0, \infty)$	(0.01, 0.10)
μ_v	Tr. Normal	$(0, \infty)$	(0.05, 0.30)	Σ-OD	Normal	$(-\infty, \infty)$	(0, 0.10)

Note: This table presents the exact functional form, the support, and the hyper-parameters for the prior distribution of each parameter. We assume normal distributions for the priors. However, if a parameter under consideration has a bounded support, we take a truncated normal as its prior. In Model I, there are 24 parameters, among which $\sigma_J, \kappa_v, \theta_v, \sigma_v, \eta_1, \mu_v, \kappa_\lambda, \theta_\lambda, \sigma_\lambda, \eta_2, \beta, \mu_v^Q$, and the diagonal elements of Σ need to be positive and ρ needs to be in $[-1, 1]$. Σ-D denotes the diagonal elements of Σ, and Σ-OD denotes the off-diagonal elements of Σ.

Table 2
Monte Carlo on likelihood estimation.

	Panel A: Simulated data				Panel B: Real data			
	512	1024	2048	4096	512	1024	2048	4096
I. The Exact Auxiliary Particle Filter (Exact-APF)								
Mean	33,053.7	33,054.0	33,054.5	33,054.6	59,128.5	59,132.4	59,135.3	59,136.9
Var	2.38	1.79	0.73	0.29	24.88	16.30	14.41	10.82
Time	0.08	0.17	0.36	0.72	0.14	0.30	0.63	1.24
Var($\ln p(y_{1:T}\|\Theta')$-$\ln p(y_{1:T}\|\Theta)$)								
$\Delta = 0.01$	2.30	1.04	0.54	0.29	14.78	10.60	9.35	5.75
$\Delta = 0.001$	1.84	1.33	0.60	0.29	15.77	10.32	9.65	6.47
II. The Smooth Particle Filter (Smooth-PF)								
Mean	33,053.3	33,053.8	33,054.3	33,054.3	59,131.7	59,135.3	59,138.0	59,139.8
Var	2.45	1.14	0.64	0.26	23.32	17.24	9.64	9.00
Time	0.11	0.23	0.49	1.01	0.20	0.42	0.89	1.80
Var($\ln p(y_{1:T}\|\Theta')$-$\ln p(y_{1:T}\|\Theta)$)								
$\Delta = 0.01$	0.03	0.01	0.01	0.00	0.73	0.59	0.60	0.54
$\Delta = 0.001$	0.00	0.00	0.00	0.00	0.01	0.03	0.03	0.02

Note: This table compares the performance of Exact-APF and Smooth-PF proposed in this paper for estimating the log-likelihood. We take the most general model, Model I, as our example. In Panel A, we simulate a sequence of daily observations with the length of 2000 on the underlying stock price and variance swap rates with maturity 1-, 6-, and 12-month at the fixed parameters, which are chosen such that their true values are close to the empirical estimates in Section 5.2. We run 128 independent filters on this simulated sequence of data at different number of particles in both particle filters. Panel B conducts an analogous exercise on the real dataset used in our paper where the filters are evaluated at the posterior mean parameter values in Section 5.2. We report the average estimate of the log-likelihood (Mean), the variance of the log likelihood estimate (Var), the computational time in seconds per filtering (Time), and the variance of the log likelihood ratio, $Var(\ln p(y_{1:T}|\Theta') - \ln p(y_{1:T}|\Theta))$, where $\Theta' = \Theta + \Delta\Theta$.

of parameters. The training-sample approach is widely used to generate the data-based priors. Notably, we find that most parameters are not sensitive to the selection of the priors. Therefore, we give flat priors to most of the model parameters. In Model I, there are 24 parameters, among which $\sigma_J, \kappa_v, \theta_v, \sigma_v, \eta_1, \mu_v, \kappa_\lambda, \theta_\lambda, \sigma_\lambda, \eta_2, \beta, \mu_v^Q$, and the diagonal elements of Σ need to be positive and ρ needs to be in $[-1, 1]$. Table 1 presents the exact functional form, the support, and the hyper-parameters for the prior distribution of each parameter.

We adopt two types of proposals. The first is a Gaussian mixture proposal with a fixed number of components, m, and the second is the random walk proposal. The Gaussian mixture distribution is fitted on the existing particle population using the EM algorithm; the covariance matrix of the random walk proposal is also fitted on the particle proposal and its scale is automatically adjusted to keep the acceptance rate between 0.2 and 0.4. Notice that fitting the proposal distribution using the information in the target is a key to the efficiency in our algorithm and is a key advantage of the sequential Monte Carlo framework. To ensure that the support of the particle set is boosted enough after resampling, we keep moving parameter particles at each density-bridging iteration until the cumulative average acceptance rate across the population reaches a fixed threshold.

4.2. Monte Carlo on likelihood estimation

In this part, we implement Monte Carlo simulations to investigate the likelihood estimate, $\hat{p}(y_{1:T}|\Theta)$, implied by the proposed particle filters. The aim is to investigate the effect of our approximations in Eq. (38) on the bias in the likelihood estimate and to measure the noise in the likelihood estimates. The two particle filters proposed in the paper are: the

Table 3
Monte Carlo on parameter Estimation I.

Θ	True value	Exact-APF-PM		Smooth-PF-CRN	
		Mean	RMSE	Mean	RMSE
μ	0.040	0.042	0.003	0.042	0.003
μ_J	−0.015	−0.016	0.003	−0.016	0.003
σ_J	0.010	0.011	0.002	0.011	0.002
κ_v	6.000	5.989	0.468	5.893	0.484
θ_v	0.010	0.011	0.001	0.011	0.001
σ_v	2.500	2.469	0.061	2.474	0.059
ρ	−0.850	−0.848	0.008	−0.847	0.008
μ_v	0.020	0.024	0.006	0.024	0.006
κ_λ	2.000	1.948	0.231	1.957	0.229
θ_λ	2.500	2.153	0.469	2.200	0.461
σ_λ	0.600	0.621	0.039	0.620	0.037
β	1.500	1.256	0.293	1.292	0.275
μ_J^Q	−0.020	−0.022	0.004	−0.022	0.004
μ_v^Q	0.025	0.030	0.007	0.030	0.007
γ_v	−0.500	−0.512	0.192	−0.472	0.194
γ_λ	−0.700	−1.001	0.499	−0.962	0.556
σ_e	0.002	0.002	0.000	0.002	0.000
Time		6.64 h		1.00 h	

Note: This table presents the results of Monte Carlo studies on parameter estimation. We consider the exact particle filter based pseudo-marginalized approach (Exact-APF-PM) and the two-stage SMC sampler based on the smooth particle filter and CRN (Smooth-PF-CRN). We generate 50 sequences of daily observations with the sample size equal to 2000 on the underlying stock price and variance swap rates with maturity 1-, 6-, and 12-month using the true parameters reported in the table. For each dataset, we run these two Bayesian methods to estimate the fixed parameters. The parameter estimates are taken to be the posterior means. The table reports the mean parameter estimates across the simulation runs together with the root mean square errors of these estimates around the true values. The last row reports the average computational time in hours for running each simulation for each method.

exact auxiliary particle filter (Exact-APF) proposed in Section 3.2, and the smooth particle filter (Smooth-PF) that use approximations in Eq. (38).

We take Model I, with $\Sigma = \sigma_e I_n$, as our example and simulate a sequence of daily observations with the length of 2000 on the underlying stock price and variance swap rates with maturity 1-, 6-, and 12-month at the fixed parameters (see Table 3). We then run 128 independent filterings on the simulated data at some practically feasible numbers of state particles, i.e., $M = 512, 1024, 2048, 4096$, in these two particle filters. All the algorithms described in this paper are massively parallel. This is an important feature as it allows us to fully use the computational power of the modern graphical processing units (GPUs), making the computational cost more manageable. We program in MATLAB and offload the computational bottleneck of the algorithm, the particle filters, to the GPU, coded in CUDA.[6]

Panel A of Table 2 sheds light on the performance of the two particle filters on the simulated data. We have the following findings. First, looking at the mean log-likelihood estimates, we observe that the bias from using the normal approximations in Eq. (38) in Smooth-PF is minor. For example, when the number of state particles equal to 512, the mean log-likelihood estimate from Exact-APF is 33,053.7, whereas it is 33,053.3 in Smooth-PF. When the number of state particles increases to 4096, the performance of the two filters is still similar, 33,054.6 vs. 33,054.3. Second, the variances of the log likelihood estimates in Exact-APF and Smooth-PF are also similar no matter which number of state particles (M) is considered.

Panel B of Table 2 reports the results of an analogous exercise, where both filters are run on the real stock price and variance swap data and the model parameters are set equal to their posterior mean estimates obtained in Section 5.2. The mean log likelihood estimates from Exact-APF and Smooth-PF are again very similar. However, the results on the variances of the likelihood estimates are starkly different when real data are used. For both filters the Monte Carlo noise is an order of magnitude larger. Even with $M = 4096$, the variances are sizeable: it is 10.82 for Exact-APF, and is 9.00 for Smooth-PF. These results could be explained by the possible model misspecification and the presence of outliers in the real data, a prevalent feature of typical financial data. It is well known in particle filtering that outliers tend to increase the Monte Carlo noise of filtering algorithms, potentially making the pseudo-marginalization approach inefficient in real data application.

However, the usefulness of Smooth-PF is at when we embed it within MCMC or SMC, as it leads to a much more regular likelihood surface locally when combined together with common random numbers. To illustrate this point, we also report in Table 2 variances of the log likelihood ratio, $Var(\ln p(y_{1:T}|\Theta') - \ln p(y_{1:T}|\Theta))$ where $\Theta' = \Theta + \Delta\Theta$ and Δ is equal to either 0.01 or 0.001, resulted from Exact-APF and Smooth-PF. For Exact-APF, independent u's are used across parameter sets, whereas for Smooth-PF, u's are fixed across parameter sets. We can easily observe that variance of the log likelihood

[6] A Linux-based computer with i7-CPUs (12G memory) and a Nvidia Titan Black GPU (3G memory) is used to produce the results in the paper.

Table 4
Monte Carlo on marginal likelihood.

	Exact-APF-PM	Smooth-PF-CRN
Mean	32,808.5	32,808.8
STD	2.17	1.81
N. of Bridging		
Stage 1	28.0	28.2
Stage 2	–	2.2

Note: This table presents the results of a Monte Carlo study for the estimation of the marginal log likelihood. We simulate a single dataset of length $T = 2000$ using the parameters reported in Table 3 and then run the two Bayesian methods, Exact-APF-PM and Smooth-PF-CRN, 20 times to estimate the fixed parameters. We use the same set-ups as in Table 3. The first row reports the average of the log marginal likelihood estimates across the 20 independent runs, the second row the standard deviation of the log marginal likelihood estimates, and the third and fourth rows the number of density-bridging iterations taken by these two methods.

Table 5
Monte Carlo on parameter Estimation II.

Θ	True value	Mean	RMSE	Θ	True value	Mean	RMSE
μ	0.040	0.041	0.003	κ_λ	2.000	2.015	0.246
μ_J	−0.015	−0.016	0.002	θ_λ	2.500	2.417	0.396
σ_J	0.010	0.010	0.002	σ_λ	0.600	0.609	0.024
κ_v	6.000	5.927	0.557	β	1.500	1.472	0.237
θ_v	0.010	0.011	0.001	μ_J^Q	−0.020	−0.020	0.003
σ_v	2.500	2.488	0.049	μ_v^Q	0.025	0.027	0.006
ρ	−0.850	−0.848	0.007	γ_v	−0.500	−0.474	0.218
μ_v	0.020	0.023	0.005	γ_λ	−0.700	−0.766	0.552
σ_{e1}	4.7e−3	4.7e−3	8.0e−5	σ_{e21}	−2.0e−3	−2.0e−3	3.9e−5
σ_{e2}	4.8e−4	4.9e−4	3.7e−5	σ_{e31}	−5.0e−4	−5.0e−4	2.1e−5
σ_{e3}	3.7e−5	5.7e−5	5.1e−5	σ_{e32}	1.3e−4	1.3e−4	3.3e−5

Note: This table presents the results of Monte Carlo simulation on the general performance of Smooth-PF-CRN in parameter estimation for Model I with the full lower triangular matrix of Σ. We generate 50 sequences of daily observations with the sample size equal to 2000 on the underlying stock price and variance swap rates with maturity 1-, 6-, and 12-month using the true parameters reported in the table. For each dataset, we run Smooth-PF-CRN to estimate the fixed parameters. The parameter estimates are taken to be the posterior means. The table reports the mean parameter estimates across the simulation runs together with the root mean square errors of these estimates around the true values.

ratio resulted from Smooth-PF is many order of magnitude smaller than that from Exact-APF. Such variance reduction of the log likelihood ratio from Smooth-PF becomes even striking in the real data application, illustrating the benefits of our procedure.

4.3. Monte Carlo on parameter estimation

We now move to the Monte Carlo study on parameter estimation, which aims to show that our proposed particle-based method delivers reliable and efficient parameter estimates. We investigate the exact auxiliary particle filter based pseudo-marginalization approach (Exact-APF-PM) and the two-stage SMC sampler based on the smooth particle filter and CRN (Smooth-PF-CRN). We take the best model in Section 5.2, Model I with $\eta_1 = \eta_2 = 2$, as our example. For the purpose to compare Exact-APF-PM and Smooth-PF-CRN in Tables 3 and 4, we set $\Sigma = \sigma_e I_n$, and for the purpose to evaluate the general performance of Smooth-PF-CRN in Table 5, we set Σ as a full lower triangular matrix.

First, we implement a Monte Carlo simulation study to compare Exact-APF-PM and Smooth-PF-CRN. We generate 50 sequences of daily observations with the sample size of 2000 on the underlying stock price and variance swap rates with maturity 1-, 6-, and 12-month. The true values of model parameters are similar to those obtained in Table 8. We run these two particle-based methods for each simulated dataset. The priors in Table 1 are used to initialize these two algorithms and the Gaussian mixture with 6 components is used as the proposal. We set the number of parameter particles, N, equal to 1024. At each density-bridging iteration, we trigger a resample-move step ESS reaches $N/2$ and we keep moving particles until the cumulative average acceptance rate across the population reaches 2. In Exact-APF-PM, the number of state particles, M, is set to 1024, and in Smooth-PF-CRN, the number of state particles in the first stage is set to $M_1 = 32$ and the number of state particles in the second stage is set to $M_2 = 1024$.

Table 3 presents the results of this simulation study. Both means and RMSEs of the posterior means across the 50 runs are reported. The last row reports the average computational time in hours for running each simulation for each algorithm. For all parameters, both Exact-APF-PM and Smooth-PF-CRN deliver very close mean estimates and RMSEs, which have

important implications: (i) the results show that using the normal approximation to the filtering distribution does not lead to a noteworthy deterioration of the results, and suggest that our smooth particle filter still allows access to essentially the full efficiency of likelihood-based inference; and (ii) the asymptotic bias when using common random numbers mentioned in the simulated maximum likelihood (SML) literature (Gourieroux and Monfort, 1996) does not have a noteworthy effect, and there does not exist any efficiency loss due to the use of a fixed set of common random numbers. Furthermore, we see that Smooth-PF-CRN runs much faster than Exact-APF-PM. The average computing time is about 1.00 h in Smooth-PF-CRN, whereas it is about 6.64 h in Exact-APF-PM. This speed-up is mainly due to our two-stage approach, where the preponderant part of the exploration is done quickly in the first stage with the small number of state particles ($M_1 = 32$). Overall, our proposed two-stage SMC sampler routine using the smooth particle filter and CRNs can deliver almost full-likelihood-based inference at a much smaller computational cost.[7]

In general, the estimate of the marginal likelihood, which is important for Bayesian model comparison, tends to be more sensitive than the ultimate parameter estimates, as the former typically depends on the full path of the algorithm. Table 4 reports the results of a Monte Carlo exercise comparing the two algorithms from this perspective. The only difference compared to the setup in Table 3 is that here we simulate a single dataset and implement 20 independent runs of each algorithm on this dataset.

The marginal log-likelihood estimate from Exact-APF-PM is about 32,808.5 with the standard deviation of 2.17. Smooth-PF-CRN generates a very similar marginal log-likelihood estimate (32,808.8) with a smaller standard deviation (1.81). The last two rows of the table report the number of bridging densities made by the two algorithms. Here the noteworthy feature is the small number of density-bridging steps, 2.2, necessary for the second stage of Smooth-PF-CRN. This is the key to the large speed-up of this routine over the pseudo-marginalized version.[8]

Second, we implement another Monte Carlo study to investigate the general performance of Smooth-PF-CRN in estimating Model I with the full lower triangular matrix of Σ. We have found that the posterior distribution in this specification is demanding as there is a more pronounced multimodality issue than that in the specification of $\Sigma = \sigma_e I_n$. Hence, compared to the simulation setup in Table 3, we find that a more conservative approach is needed for a stable SMC algorithm. In particular, we use a random walk proposal and more bridging densities, i.e., triggering a resample-move step whenever the ESS reaches $0.9 \times N$. Furthermore, we keep moving particles at each density-bridging iteration until the cumulative average acceptance rate across the population reaches 5. The true values of model parameters are the same as those in Table 3 except that the elements of Σ are set to those estimates obtained in Table 9.

Table 5 presents means and RMSEs of posterior means of model parameters across the 50 runs. We find that for nearly all parameters, the means are very close to the true values and RMSEs are relatively small. The only exception is the third diagonal element, σ_{e3}, of Σ, whose mean is distant from the true value and whose RMSE is relatively large. The main reason is that the true value of this parameter is too small, and the algorithm has difficulty to position this parameter. In an unreported experiment, we find that when we use a larger value of σ_{e3} in simulation (3.7e−4 instead of 3.7e−5), the algorithm can accurately identify this parameter (see Internet Appendix).

5. Empirical application

5.1. The data

The data used for estimation are S&P 500 index returns and variance swap rates with fixed maturity at 1-, 6-, and 12-month, ranging from January 2, 2001 to July 20, 2015 for a total of 3523 business days. S&P 500 index values are obtained from *Datastream*, and the variance swap rates are provided by an investment bank. The data cover the market turmoils such as the 2002 dot-com bubble burst, the 2008 global financial crisis, and the recent European debt crisis of 2010–2012.

Table 6 reports the summary statistics of index returns and variance swap rates. We see that the annualized mean of index returns is about 3.3%, and the historical volatility is about 20.2%. The index returns are clearly left-skewed and leptokurtic as the skewness is negative (−0.20) and the kurtosis is by far larger than three (11.67). They are weakly autocorrelated with the first autocorrelation being only about −0.09. The Jarque–Bera test easily rejects the null hypothesis of return normality with a very small p-value.

The mean values of variance swap rates increase with respect to maturity. In volatility measure, they are 20.4% at 1-month maturity, 21.9% at 6-month maturity, and 22.5% at 12-month maturity. In contrast, the standard deviations decrease with respect to maturity. Furthermore, the variance swap rates have positive skewness and excess kurtosis, both of which decrease with respect to maturity. The variance swap rates are highly persistent. The first autocorrelation is 0.983 for 1-month maturity, 0.994 for 6-month maturity, and 0.996 for 12-month maturity.

Fig. 1 presents the time series of index returns and the variance swap rates. The long-maturity variance swap rate is higher than the short-maturity rate when the market is tranquil. However, during the periods of market turmoil, the term structure of variance swap rates reverts: the short-maturity variance swap rate moves up quickly to even higher level than the long-maturity rate.

[7] In Table 3, the true value of σ_e, the standard deviation of the measurement errors, is set to that estimated using the real data. Our two-stage SMC sampler, Smooth-PF-CRN, can still efficiently estimate nearly all parameters when σ_e is set to a larger value, 0.01 instead of 0.002 (see Internet Appendix).

[8] Duan and Fulop (2015) propose a different two-stage scheme to adapt the number of state particles, M, within the pseudo-marginalized context. They do observe improvements compared to the fixed-M setup. However, the gains are much more limited compared to what we observe in the common random number case here.

Table 6
Summary statistics.

	Mean	Std.	Max.	Min.	Skew.	Kurt.	ACF
Returns	0.033	0.202	0.110	−0.095	−0.198	11.67	−0.087
VS1M	0.204	0.091	0.801	0.094	1.997	8.902	0.983
VS6M	0.219	0.072	0.614	0.123	1.482	6.158	0.994
VS1Y	0.225	0.065	0.536	0.130	1.173	4.877	0.996

Note: The table presents summary statistics of the data used for model estimation. The data include S&P 500 index returns and variance swap rates with fixed maturity at 1-, 6-, and 12-month. Variance swap rates are reported in volatility measure. The time period ranges from January 2, 2001 to July 20, 2015 for a total of 3523 business days.

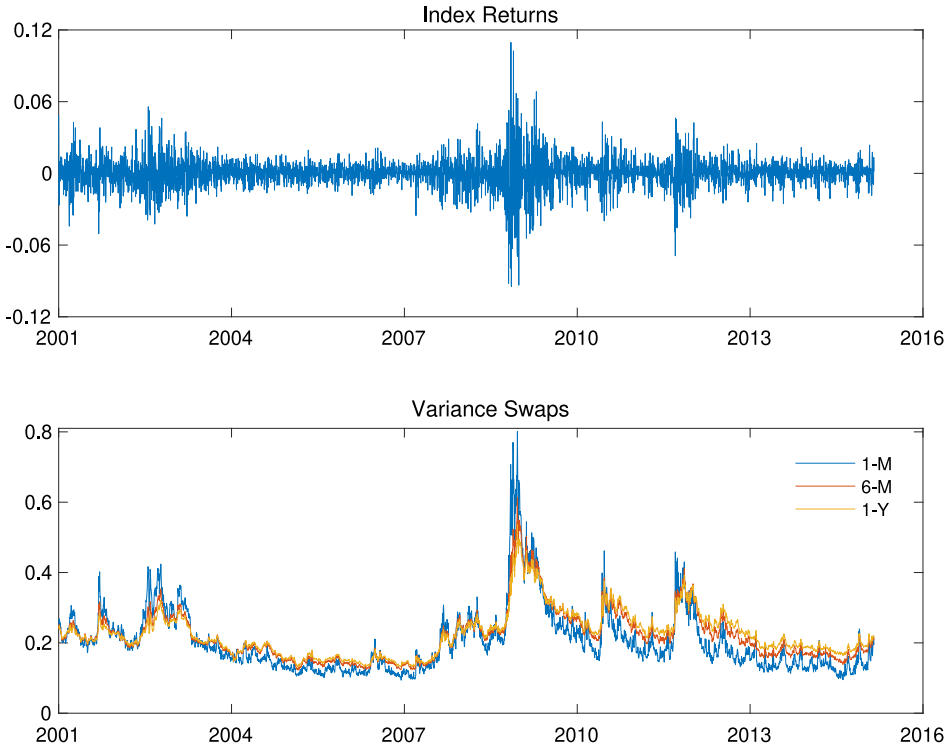

Fig. 1. S&P 500 index returns and variance swap rates. *Note:* The figure plots the S&P500 index returns and variance swap rates with the fixed maturity 1-, 6-, and 12-month. The data range from January 2, 2001 to July 20, 2015, in total, 3523 business days. Variance swap rates are presented in volatility measure.

5.2. Model estimation and empirical results

In this Subsection, we present the empirical estimation results. We first implement statistical analysis and diagnosis and present the estimation results based on models with $\Sigma = \sigma_e I_n$ in Sections 5.2.1 and 5.2.2, respectively. We then introduce correlated variance swap pricing errors and estimate models with the full lower triangular matrix of Σ using Smooth-PF-CRN alone in Section 5.2.3. In empirical implementations, we set the number of parameter particles (N) equal to 2048. For Exact-APF-PM, we set the number of state particles (M) equal to 1024, and for Smooth-PF-CRN, we set the number of state particles at the first stage (M_1) equal to 32, and the number of state particles at the second stage (M_2) equal to 1024. The algorithms are initialized by the priors presented in Table 1. When estimating models with $\Sigma = \sigma_e I_n$, we take the Gaussian mixture with 6 components as the proposal, trigger the resample-move step when the ESS reaches $N/2$, and then keep moving until the cumulative average acceptance rate across the population reaches 2, whereas when estimating models with the full lower triangular matrix of Σ, we take the random walk proposal, trigger the resample-move step when the ESS reaches $0.9 \times N$, and then keep moving until the cumulative average acceptance rate across the population reaches 10.

5.2.1. Statistical analysis and diagnosis

One of the most important statistics in our particle-based methods is the acceptance rate, which measures the efficiency of the move step. We estimate the four models with $\Sigma = \sigma_e I_n$ using Exact-APF-PM and Smooth-PF-CRN, and take a look at acceptance rates. The left panels of Fig. 2 present the last acceptance rate in move steps at each density-bridging iteration

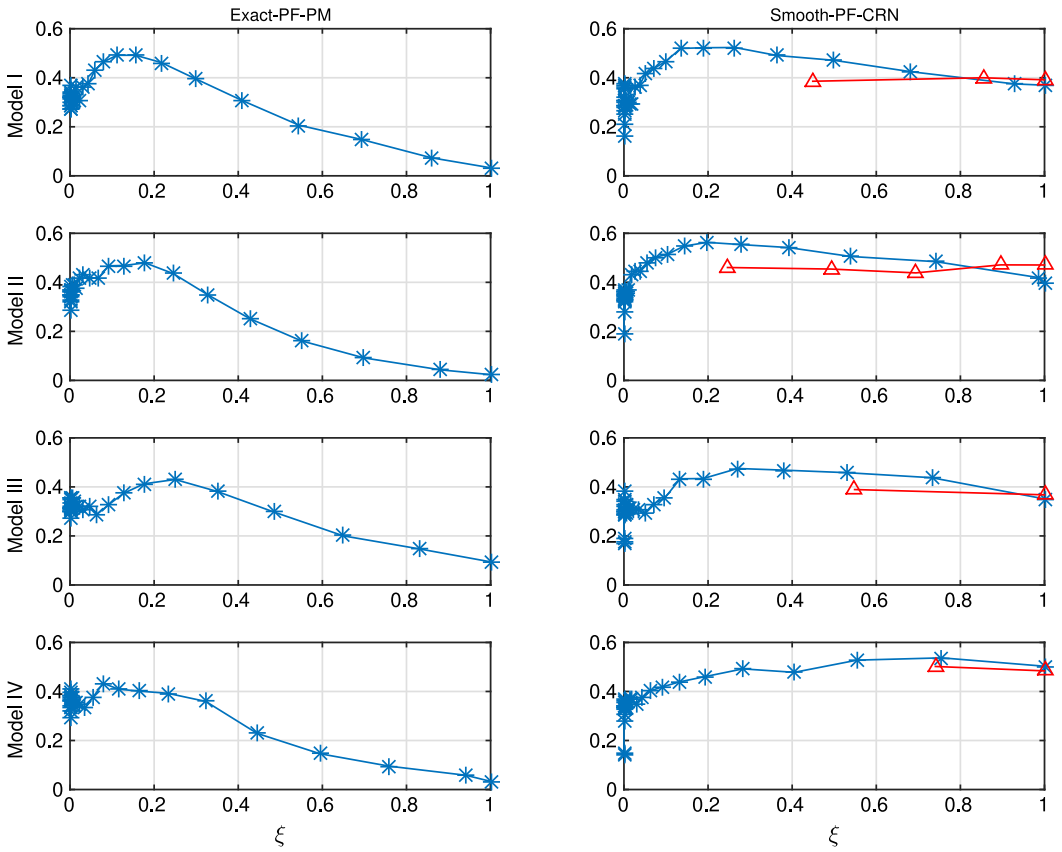

Fig. 2. Acceptance rates. *Note:* The figure plots the last acceptance rate in moving steps at each density-bridging iteration with respect to ξ_i for the four models considered. In our algorithm, ξ_i is automatically selected using a grid search approach. The acceptance rates resulted from Exact-APF-PM are presented in the left panels, and the acceptance rates from Smooth-PF-CRN are presented in the right panels, in which the star-line is for the acceptance rates in the first stage, and the triangular-line is for those in the second stage.

resulting from Exact-APF-PM. We see that for each of the four models, even though the acceptance rate increases and remains high before the tempering coefficient, ξ_i, reaches the level of about 0.20, it declines thereafter and is quickly going down to a low level smaller than 5% when ξ is approaching 1, requiring a large number of move steps (more than 40). The low acceptance rate is because the existence of possible model misspecification and the presence of outliers in the data make variance of the likelihood estimate from Exact-APF large as seen in Table 2.

The left panels of Fig. 2 present the last acceptance rate in move steps at each density-bridging iteration resulting from Smooth-PF-CRN. The star-line is for those from the first stage and the triangular-line for those from the second stage. We find that (i) the acceptance rates in the first stage remain high over the tempering procedure for each of these four models: it is around 20% in the beginning, and then is slowly going up and stays above 40% till the end, suggesting that the number of move steps at each density-bridging iteration ranges from 4 to 10; (ii) The acceptance rates are also quite high in the second stage: they are basically around 40% for the four models, suggesting that the number of move steps at each density-bridging iteration is about 4–5; (iii) There are much less density-bridging iterations in the second stage than in the first stage; and (iv) Consistent with behaviors of the acceptance rate in Fig. 2, Exact-APF-PM requires much larger computational cost than Smooth-PF-CRN does. For example, in estimating Model I, Smooth-PF-CRN takes about 6 h, whereas Exact-APF-PM needs more than 37 h, suggesting that Smooth-PF-CRN is about 5.9 times faster than Exact-APF-PM (see the last row of Table 8).

The above statistical analysis suggests that the use of common random numbers plays a critical role in our 2-stage SMC sampler, which is more robust to outliers and potential model misspecification and much faster than the pseudo-marginalization approach. Therefore, in what follows, we focus on estimation results obtained from Smooth-PF-CRN.

We now move to take a look at the posterior distributions of the fixed parameters. In Fig. 3, the left panels present the posterior distributions of the parameters governing the diffusion volatility dynamics, κ_v, θ_v, σ_v, and μ_v, and the right panels present those governing the jump intensity, κ_λ, θ_λ, σ_λ, and β. All the posteriors seem close to Gaussian distributions except that κ_v is a little right-skewed. This result suggests that asymptotic Gaussianity is in general a good approximation. This is in contrast to the results of Fulop et al. (2015), who show that when only underlying data is used to estimate a similar self-exciting model, statistical learning about the jump dynamics is very slow, and hence the posteriors are very far from normal

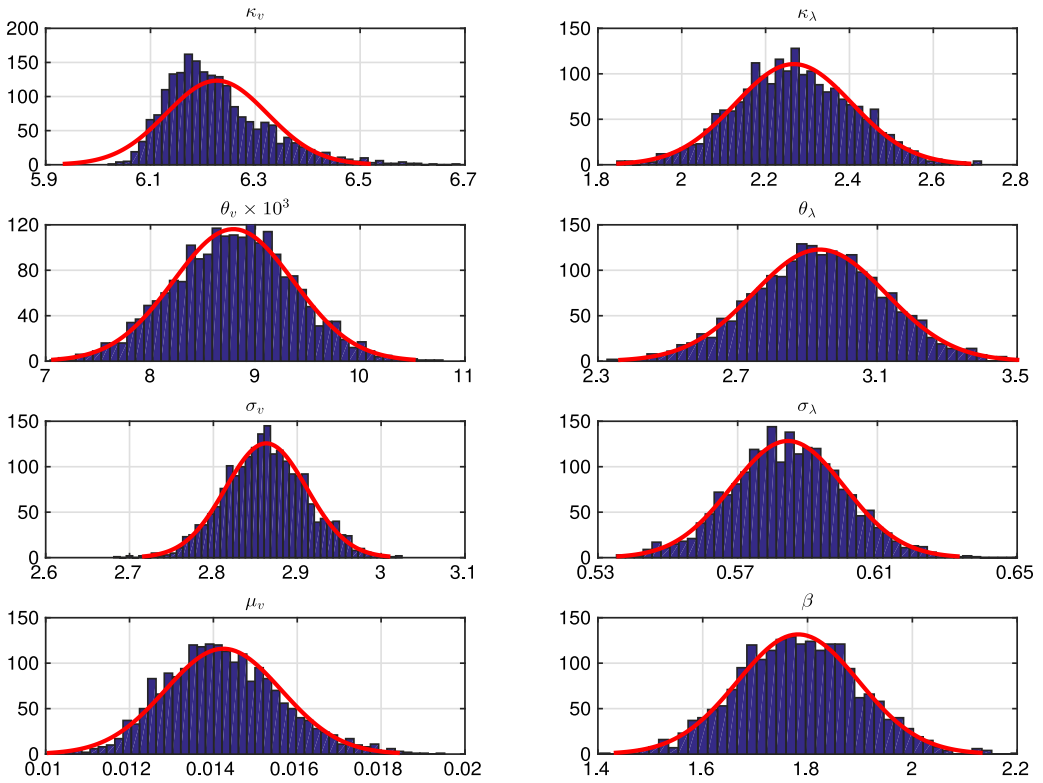

Fig. 3. Posterior distributions. *Note:* The figure presents the posterior distributions of the model parameters resulted from Smooth-PF-CRN. The left panels plot the posterior distributions of the parameters governing the diffusion volatility dynamics, and the right-hand panels plot the posterior distributions of the parameters governing the jump intensity dynamics.

in general. Of course, the crucial difference is the presence of derivatives observations in our case, which are informative on volatility and jump dynamics and hence push the results close to the asymptotic normal regime.

5.2.2. Do we need both non-affineness and self-excitation?

In addition to co-jumps of stock prices and volatility, which have been investigated intensively by the previous literature, our model takes into account both non-affineness and self-excitation in volatility dynamics. Given that both can help generate skewed and fat-tailed stock return distributions, we may wonder whether we need both of them when modeling volatility dynamics.

Table 7 presents the log Bayes factors for model comparison, from which we have the following important findings. First, the non-affine models (Model I and Model II) perform much better than the affine models (Model III and Model IV). For example, the log Bayes factor between Model I and Model III (Model IV) is as large as 818 (989), and the log Bayes factor between Model II and Model III (Model IV) is about 682 (853). This result indicates that the usually used affine models are clearly misspecified, and non-affine specification needs to be used to capture volatility dynamics. With some parsimonious stochastic volatility specifications, which do not include price-volatility co-jumps and self-exciting jump intensity, Jones (2003), Ait-Sahalia and Kimmel (2007), Christoffersen et al. (2010), and Amengual and Xiu (2018) also argue that affine models are misspecified.

Second, the jump intensity follows self-exciting dynamics. No matter whether non-affine specifications or affine specifications are concerned, the model with the self-exciting jump intensity always performs better than that excluding self-excitation. For example, for the non-affine models, the log Bayes factor between Model I and Model II is about 137, and for the affine models, the log Bayes factor between Model III and Model IV is about 171. This result suggests that the self-excitation is an important feature in volatility dynamics. Under affine specifications, Carr and Wu (2017), Ait-Sahalia et al. (2015), and Fulop et al. (2015) find evidence for self-excitation. Our investigation further indicates that though both non-affineness and self-excitation are channels for sudden burst of volatility and extreme events, they are not close substitutes of each other.

Third, non-affineness plays a more important role than self-excitation in volatility specification. Comparing Model II, which allows for non-affineness but shuts down self-excitation, and Model III, which is affine and includes self-exciting effect, we find that Model II performs much better than Model III. The log Bayes factor between Model II and Model III is as large as 682.

Table 7
The log Bayes factors.

	Model I	Model II	Model III	Model IV	Model IR	Model IIR
Model I	0.00	–	–	–	17.9	−134.1
Model II	136.6	0.00	–	–	154.5	2.44
Model III	818.2	681.7	0.00	–	836.2	684.1
Model IV	989.0	852.5	170.8	0.00	1007.0	854.9

Note: The table presents the log Bayes factor of the column model to the row model. For any two given models, $\mathcal{M}_1$ and $\mathcal{M}_2$, if the value of the log Bayes factor is between 0 and 1.1, $\mathcal{M}_1$ is barely worth mentioning; if it is between 1.1 and 2.3, $\mathcal{M}_1$ is substantially better than $\mathcal{M}_2$; if it is between 2.3 and 3.4, $\mathcal{M}_1$ is strongly better than $\mathcal{M}_2$; if it is between 3.4 and 4.6, $\mathcal{M}_1$ is very strongly better than $\mathcal{M}_2$; and if it is larger than 4.6, $\mathcal{M}_1$ is decisively better than $\mathcal{M}_2$. All models are estimated using Smooth-PF-CRN.

Table 8
Parameter estimates for the models with $\Sigma = \sigma_e I_n$.

Θ	Model I		Model II		Model III		Model IV	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
μ	0.046	(0.009)	0.049	(0.009)	0.041	(0.009)	0.043	(0.009)
μ_J	−0.018	(0.001)	−0.018	(0.001)	−0.021	(0.001)	−0.023	(0.002)
σ_J	0.002	(0.001)	0.003	(0.001)	0.002	(0.001)	0.003	(0.002)
κ_v	6.227	(0.097)	6.175	(0.078)	10.93	(0.467)	10.53	(0.525)
θ_v	0.009	(0.001)	0.009	(0.001)	0.007	(0.000)	0.008	(0.000)
σ_v	2.863	(0.049)	2.810	(0.045)	0.607	(0.008)	0.606	(0.008)
η_1	2.000	(−−)	2.000	(−−)	1.000	(−−)	1.000	(−−)
ρ	−0.848	(0.006)	−0.848	(0.006)	−0.765	(0.008)	−0.765	(0.009)
μ_v	0.014	(0.001)	0.020	(0.002)	0.029	(0.003)	0.043	(0.005)
κ_λ	2.266	(0.140)	0.532	(0.101)	2.563	(0.175)	0.656	(0.098)
θ_λ	2.935	(0.190)	10.02	(2.130)	2.140	(0.149)	5.758	(0.841)
σ_λ	0.585	(0.016)	0.703	(0.017)	2.565	(0.097)	2.606	(0.145)
η_2	2.000	(−−)	2.000	(−−)	1.000	(−−)	1.000	(−−)
β	1.782	(0.116)	0.000	(−−)	1.995	(0.144)	0.000	(−−)
μ_J^Q	−0.018	(0.001)	−0.020	(0.001)	−0.021	(0.001)	−0.025	(0.002)
μ_v^Q	0.015	(0.001)	0.020	(0.002)	0.017	(0.001)	0.024	(0.003)
γ_v	−0.038	(0.031)	−0.030	(0.024)	−7.858	(0.769)	−7.366	(0.881)
γ_λ	−0.222	(0.142)	−0.254	(0.143)	−0.083	(0.052)	−0.115	(0.037)
σ_e	0.0017	(0.000)	0.0017	(0.000)	0.0017	(0.000)	0.0017	(0.000)
$\bar{V}$	0.041	(0.006)	0.040	(0.007)	0.035	(0.007)	0.031	(0.004)
$\bar{\lambda}$	14.08	(2.483)	10.02	(2.130)	10.14	(2.424)	5.748	(0.841)
$\bar{V}_{tot}$	0.045	(0.007)	0.043	(0.007)	0.039	(0.008)	0.034	(0.005)
$\sqrt{\bar{V}_{tot}}$	0.213	(0.016)	0.208	(0.017)	0.197	(0.019)	0.185	(0.013)
Time	5.9		5.5		6.8		9.8	

Note: The table presents the parameter estimates resulted from Smooth-PF-CRN for the four models considered in the paper. Model I and Model II refer to non-affine models with $\eta_1 = \eta_2 = 2$. For each parameter, the posterior mean and the posterior standard deviation (in bracket) are reported. $\bar{V}$ is the long-run mean of the diffusion variance and is given by Eq. (7); $\bar{\lambda}$ is the long-run mean of the jump intensity and is given by Eq. (5). $\bar{V}_{tot}$ stands for the long-run mean of total return variance, which is given by $\bar{V}_{tot} = \bar{V} + (\mu_J^2 + \sigma_J^2)\bar{\lambda}$. The last row reports the relative computational time of Exact-APF-PM to Smooth-PF-CRN in estimating these four models. The data used for estimation are S&P 500 index returns and variance swap rates with fixed maturity 1-, 6-, and 12-month, ranging from January 2, 2001 to January 20, 2015, in total, 3523 business days.

Fourth, our model estimation indicates that the estimates of η_1 and η_2 are quite close to 2 in both Model I and Model II. We therefore investigate two restricted non-affine models by setting $\eta_1 = \eta_2 = 2$ in Model IR and Model IIR. We find that the restricted models outperform the corresponding non-restricted models with free η's. For example, the log Bayes factor between Model IR and Model I is about 18, and the log Bayes factor between Model IIR and Model II is about 2.4. Therefore, in what follows, when we refer to Model I (Model II), we actually mean Model IR (Model IIR).

The above model comparison indicates that we need to take into account both non-affineness and self-excitation in order to accurately capture volatility dynamics. Table 8 presents the parameter estimates for the four models. There are some notable findings. First, diffusion volatility, V_t, is more persistent in the non-affine models than in the affine models. For example, the estimate of κ_v is 6.23 (0.10) in Model I and 6.18 (0.08) in Model II, whereas it is 10.9 (0.47) in Model III and 10.5 (0.53) in Model IV. The long-run mean of diffusion variance, $\bar{V}$, which is given by Eq. (7), is about 0.041 (0.006) in Model I, is about 0.040 (0.007) in Model II, is about 0.035 (0.007), and is about 0.031 (0.004) in Model IV, according to the corresponding parameter estimates.

Second, the persistence of the jump intensity is controlled by $\kappa_\lambda - \beta$ in Model I and Model III, whereas it is controlled by κ_λ in Model II and Model IV. It is about 0.49 (0.084) in Model I, is about 0.53 (0.101) in Model II, is about 0.57 (0.132) in Model III, and is about 0.66 (0.098), suggesting that the jump intensity is much more persistent than diffusion volatility in all the models. The long-run mean of the jump intensity is given by formula (5), which is about 14.1 (2.48) in Model I, is about 10.0

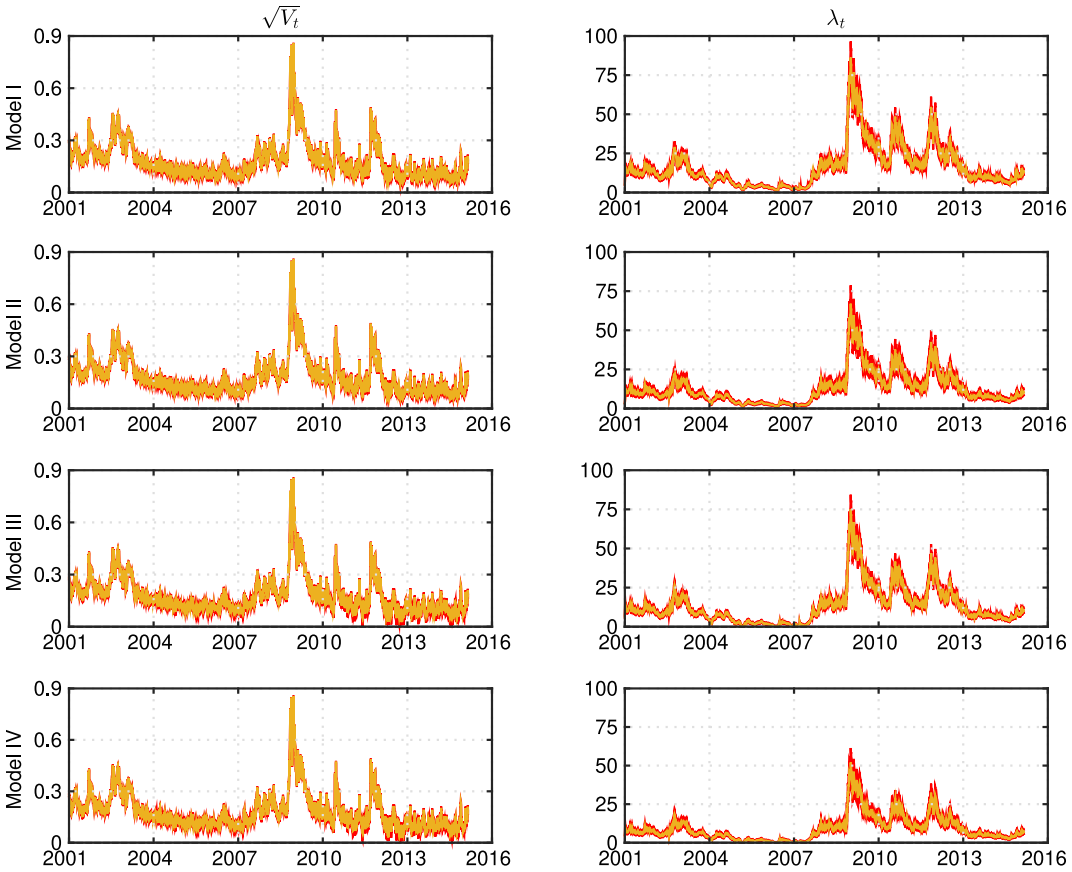

Fig. 4. Diffusion volatility and jump intensity. *Note:* The figure plots filtered diffusion volatility and jump intensity that take into account parameter uncertainty for the four models considered in the paper using the real data on S&P 500 index returns and variance swap rates with the fixed maturity 1-, 6-, and 12-month, ranging from January 2, 2001 to January 20, 2015 for a total of 3.523 business days. The posterior mean and (5, 95)% quantiles are reported at each time point.

(2.13), is about 10.1 (2.42) in Model III, and is about 5.75 (0.84). These values indicate that there are more jumps, on average, in each year in the self-exciting models (Model I and Model III) than in the corresponding models without self-exciting jumps (Model II and Model IV), and that there are more jumps, on average, in each year in the non-affine models (Model I and Model II) than in the corresponding affine models (Model III and Model IV). The model-implied return volatility, $\sqrt{\bar{V}+(\mu_J^2+\sigma_J^2)\bar{\lambda}}$, is about 21.3% (1.6%) in Model I, 20.8% (1.7%) in Model II, 19.7% (1.9%) in Model III, and 18.5% (1.3%) in Model IV.

Third, the parameter β controls the self-excitation of the jump intensity. Its posterior mean is about 1.8 and its posterior standard deviation is about 0.12 in Model I, and its posterior mean is about 2.0 and its posterior standard deviation is about 0.14 in Model III. These values indicate that β is well identified and self-excitation is a key feature of the jump dynamics.

Fourth, the risk-premium parameters, γ_v and γ_λ, are not well identified in the non-affine models (Model I and Model II). However, in the affine models, even though γ_λ is still hard to identify, γ_v is highly statistically significant in both Model III and Model IV.

Last, the mispricing parameter, σ_e, is about 0.0017 and highly statistically significant in all the four models, indicating that even though these models perform very differently in fitting stock price and variance swap rates data, their performance in pricing variance swaps is very similar.

Fig. 4 plots the filtered diffusion volatility and jump intensity that take into account parameter uncertainty for the four models.[9] The posterior means and (5, 95)% quantiles are presented. We can see that both diffusion volatility and jump intensity are well identified, as the (5, 95)% credible intervals are very narrow both in calm periods and in turmoil periods. This is in contrast to the results of Fulop et al. (2015) who find that the jump intensity is hard to pin down using underlying data only. Our more positive results underscore the importance of using derivative data in pinning down the jump dynamics.

[9] We run our smooth particle filter at each parameter set from the posterior distribution of the parameters and sample randomly at each time from this mixture of filtering distributions.

Table 9
Parameter estimates for model I with full Σ.

Θ	Mean	Std	Θ	Mean	Std
μ	0.047	(0.010)	κ_λ	1.440	(0.100)
μ_J	−0.016	(0.001)	θ_λ	2.731	(0.314)
σ_J	0.001	(0.001)	σ_λ	0.527	(0.011)
κ_v	4.616	(0.103)	η_2	2.000	(−−)
θ_v	0.004	(0.001)	β	1.163	(0.088)
σ_v	2.416	(0.037)	μ_J^Q	−0.001	(0.001)
η_1	2.000	(−−)	μ_v^Q	0.013	(0.001)
ρ	−0.849	(0.006)	γ_v	−0.036	(0.034)
μ_v	0.012	(0.001)	γ_λ	−0.113	(0.099)
σ_{e1}	4.68e−3	(2.21e−4)	σ_{e21}	−2.09e−3	(9.24e−5)
σ_{e2}	4.77e−4	(4.30e−5)	σ_{e31}	−5.08e−4	(6.45e−5)
σ_{e3}	3.73e−5	(2.27e−5)	σ_{e32}	1.31e−4	(3.12e−5)
$\bar{V}$	0.042	(0.007)	$\bar{V}_{tot}$	0.046	(0.008)
$\bar{\lambda}$	14.63	(3.000)	$\sqrt{\bar{V}_{tot}}$	0.212	(0.019)
MLLH I	5.981e+4		MLLH II	6.156e+4	

Note: The table presents the parameter estimates resulted from Smooth-PF-CRN for Model I with the full lower triangular matrix of Σ. For each parameter, the posterior mean and the posterior standard deviation (in bracket) are reported. $\bar{V}$ is the long-run mean of the diffusion variance and is given by Eq. (7); $\bar{\lambda}$ is the long-run mean of the jump intensity and is given by Eq. (5). $\bar{V}_{tot}$ stands for the long-run mean of total return variance, which is given by $\bar{V}_{tot} = \bar{V} + (\mu_J^2 + \sigma_J^2)\bar{\lambda}$. The last row reports the marginal log likelihood estimates of Model I with $\Sigma = \sigma_e I_n$ and Model I with the full lower triangular matrix of Σ, MLLH I and MLLH II, respectively. The data used for estimation are S&P 500 index returns and variance swap rates with fixed maturity 1-, 6-, and 12-month, ranging from January 2, 2001 to January 20, 2015, in total, 3523 business days.

Both diffusion volatility and jump intensity go to highest levels during periods of financial crisis. All the four models yield similar diffusion volatility estimates, whereas the levels of jump intensity are sizeably different among these models.

5.2.3. Introducing correlated pricing errors

We now introduce correlated variance swap pricing errors and estimate the models with the full lower triangular matrix of Σ using our two-stage SMC sampler, Smooth-PF-CRN. Table 9 presents the estimates of the model parameters and the marginal likelihood for Model I. For those estimates for the other models, see Internet Appendix.

We find that these parameter estimates deliver very similar implications as those discussed in the previous subsection, even though some of them are a little different. For example, the long-run means of diffusion variance, jump intensity, and total volatility are about 0.042 (0.007), 14.6 (3.00), and 21.2% (1.9%), respectively, in Table 9, whereas they are very similar, 0.041 (0.006), 14.1 (2.48), and 21.3% (1.6%), respectively, in Table 8. Furthermore, we find that all elements in Σ are highly statistically significant except the third diagonal element, σ_{e3}, whose estimate is very small. In general, these estimates suggest that the longer the maturity is, the better the model performs in pricing variance swaps, and that the pricing errors of 1-month variance swaps are negatively correlated with those of 6-month/1-year variance swaps, whereas the pricing errors of 6-month variance swaps are positively correlated with those of 1-year variance swaps.

In the last row of Table 9, we present the marginal log likelihood estimates of Model I with $\Sigma = \sigma_e I_n$ and Model I with the full lower triangular matrix of Σ, MLLH I and MLLH II, respectively. We find that the latter performs much better than the former in fitting stock price and variance swap rates data, as the marginal log likelihood of the former is about 5.98e+4, whereas it is about 6.16e+4 for the latter.

6. Concluding remarks

In this paper, we propose a new particle-based method, which is based on (i) a smooth particle filter and (ii) a two-stage sequential Monte Carlo sampler with common random numbers, for estimating dynamic asset pricing models with informative observations. Through extensive Monte Carlo studies, we document that the new method practically allows almost full-likelihood-based inference at a much small computing cost. Even more importantly, the method is much more robust than the existing pseudo-marginalized methods to the presence of model misspecification, a prevalent feature of real financial and economic data. We illustrate our algorithm on a class of derivative pricing models that allow for co-jumps of prices and volatility, self-excitation, and non-affineness jointly using underlying and derivatives information. The method can effectively control the noise of the likelihood estimate and makes it efficient to estimate such complex models. Using daily S&P 500 index returns and variance swap rates with time to maturity of 1, 6, and 12 months over the period from 2001 to 2015, we show that both non-affineness and self-excitation need to be taken into account in modeling volatility dynamics.

Our method should be useful for a wide range of state-space models in finance and economics, where the observations are highly informative on the latent states. For example, it could facilitate estimation of derivative pricing models using panels of

options, term structure models, or dynamic stochastic general equilibrium models in macroeconomics. Furthermore, while our treatment here is Bayesian, by letting the tempering coefficients increase to a sufficiently large number, the resulting density concentrates around the mode of the estimated likelihood function and our approach yields simulated maximum likelihood point estimates (see, e.g., Johansen et al., 2008).

Appendix A. $\mathbb{P}$-expectation of quadratic variation

The $\mathbb{P}$-expectation of quadratic variation can be obtained from Eq. (9) as follows

$$\begin{aligned} E_t^{\mathbb{P}}[QV(t,T)] &= \int_t^T E_t^{\mathbb{P}}[V_s]ds + Var_J \int_t^T E_t^{\mathbb{P}}[\lambda_s]ds \\ &= [A(T-t) + B(T-t)V_t + C(T-t)\lambda_t](T-t), \end{aligned}$$

where $A(T-t) = A_1(T-t) + A_2(T-t) + A_3(T-t)$, $C(T-t) = C_1(T-t) + C_2(T-t)$, and

$$\begin{aligned} A_1(T-t) &= \Big(1 - B(T-t)\Big)\theta_v, \qquad A_2(T-t) = \Big(\mu_J^2 + \sigma_J^2\Big)\Big(1 - H(T-t)\Big)\bar{\lambda}, \\ A_3(T-t) &= \mu_v \left[\frac{1 - B(T-t)}{\kappa_v} + \frac{H(T-t) - B(T-t)}{\kappa_\lambda - \beta - \kappa_v}\right]\bar{\lambda}, \\ C_1(T-t) &= -\mu_v \frac{H(T-t) - B(T-t)}{\kappa_\lambda - \beta - \kappa_v}, \qquad C_2(T-t) = \Big(\mu_J^2 + \sigma_J^2\Big)H(T-t), \\ B(T-t) &= \frac{1 - e^{-\kappa_v(T-t)}}{\kappa_v(T-t)}, \qquad H(T-t) = \frac{1 - e^{-(\kappa_\lambda - \beta)(T-t)}}{(\kappa_\lambda - \beta)(T-t)}. \end{aligned}$$

Appendix B. Auxiliary and smooth particle filters

B.1. The Exact Auxiliary Particle Filter (Exact-APF)

B.1.1. Derivation details

The observation vector at time t consists of the underlying price and n variance swap rates, i.e., $y_t = [\ln S_t, VS^{\mathcal{O}}_{t,\tilde{\tau}_1}, \ldots, VS^{\mathcal{O}}_{t,\tilde{\tau}_n}]'$. It is easy to derive the coefficients C_{t-1} and D_{t-1} in Eq. (27). The coefficients corresponding to the underlying price are:

$$C_{t-1}^{(1)} = \ln S_{t-1} + \Big(\mu - \frac{1}{2}V_{t-1} - k(1)\lambda_{t-1}\Big)\tau + \mu_J \Delta N_t, \qquad D_{t-1}^{(1)} = 0,$$

whereas the coefficients corresponding to the variance swap observations, $VS^{\mathcal{O}}_{t,\tilde{\tau}_l}$, $l = 1, \ldots, n$, are:

$$\begin{aligned} C_{t-1}^{(l+1)} &= A^{\mathbb{Q}}(\tilde{\tau}_l) + B^{\mathbb{Q}}(\tilde{\tau}_l)\left(V_{t-1} + \kappa_v(\theta_v - V_{t-1})\tau\right) + C^{\mathbb{Q}}(\tilde{\tau}_l)\left(\lambda_{t-1} + \kappa_\lambda(\theta_\lambda - \lambda_{t-1})\tau + \beta\Delta N_t\right), \\ D_{t-1}^{(l+1)} &= B^{\mathbb{Q}}(\tilde{\tau}_l)\Delta N_t. \end{aligned}$$

Furthermore, the covariance matrix Σ_{t-1} in Eq. (27) has the following form:

$$\begin{aligned} \Sigma_{t-1}^{(1,1)} &= \Delta N_t \sigma_J^2 + V_{t-1}\tau, \\ \Sigma_{t-1}^{(1,l+1)} &= \sigma_v V_{t-1}^{(\eta_1+1)/2}\tau B^{\mathbb{Q}}(\tilde{\tau}_{l-1})\rho, \quad \text{for} \quad l = 1, \ldots, n, \\ \Sigma_{t-1}^{(l+1,h+1)} &= B^{\mathbb{Q}}(\tilde{\tau}_l)B^{\mathbb{Q}}(\tilde{\tau}_h)V_{t-1}^{\eta_1}\tau\sigma_v + C^{\mathbb{Q}}(\tilde{\tau}_l)C^{\mathbb{Q}}(\tilde{\tau}_h)\lambda_{t-1}^{\eta_2}\tau\sigma_\lambda + 1_{\{l=h\}}\sigma_\epsilon^2, \\ &\quad \text{for} \quad l = 1, \ldots, n, \quad h = 1, \ldots, n. \end{aligned}$$

Next, we show how to obtain the density $p(y_t \mid \Delta N_t = 1, \lambda_{t-1}, V_{t-1})$. The integral to be solved is:

$$\begin{aligned} &p(y_t \mid \Delta N_t = 1, \lambda_{t-1}, V_{t-1}) \\ &= \int_0^\infty p(y_t \mid J_{v,t}, \Delta N_t = 1, \lambda_{t-1}, V_{t-1}) p(J_{v,t} \mid \Delta N_t = 1, \lambda_{t-1}, V_{t-1}) dJ_{v,t} \\ &= \int_0^\infty |\Sigma_{t-1}|^{-\frac{1}{2}}(2\pi)^{-\frac{n}{2}} e^{-\frac{1}{2}(y_t - C_{t-1} - D_{t-1}J_{v,t})'\Sigma_{t-1}^{-1}(y_t - C_{t-1} - D_{t-1}J_{v,t})} \frac{1}{\mu_v} e^{-\frac{J_{v,t}}{\mu_v}} dJ_{v,t}. \end{aligned}$$

The key to the solution is that the integrand is an exponential quadratic function of $J_{v,t}$. Hence, the conditional distribution of the variance jumps is normal truncated to the positive domain with the following mean and variance parameters:

$$\mu_{J_{v,t}} = \frac{D_{t-1}'\Sigma_{t-1}^{-1}(y_t - C_{t-1}) - 1/\mu_v}{D_{t-1}'\Sigma_{t-1}^{-1}D_{t-1}},$$

$$\sigma^2_{J_{v,t}} = \frac{1}{D'_{t-1}\Sigma^{-1}_{t-1}D_{t-1}},$$

and the above integral can be solved as follows:

$$\begin{aligned}
&p(y_t \mid \Delta N_t = 1, \lambda_{t-1}, V_{t-1}) \\
&= |\Sigma_{t-1}|^{-\frac{1}{2}}(2\pi)^{-\frac{n}{2}} e^{-\frac{1}{2}(y_t - C_{t-1})'\Sigma^{-1}_{t-1}(y_t - C_{t-1})} \frac{1}{\mu_v}\sigma_{J_{v,t}} e^{\frac{1}{2}\frac{\mu^2_{J_{v,t}}}{\sigma^2_{J_{v,t}}}} \\
&\quad \times \sqrt{2\pi}\int_0^\infty \frac{1}{\sigma_{J_{v,t}}\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(J_{v,t}-\mu_{J_{v,t}})^2}{\sigma^2_{J_{v,t}}}} dJ_{v,t} \\
&= |\Sigma_{t-1}|^{-\frac{1}{2}}(2\pi)^{-\frac{n}{2}} e^{-\frac{1}{2}(y_t - C_{t-1})'\Sigma^{-1}_{t-1}(y_t - C_{t-1})} \frac{1}{\mu_v} e^{\frac{1}{2}\frac{\mu^2_{J_{v,t}}}{\sigma^2_{J_{v,t}}}} \sigma_{J_{v,t}}\sqrt{2\pi}\left(1 - \Phi\left(-\frac{\mu_{J_{v,t}}}{\sigma_{J_{v,t}}}\right)\right).
\end{aligned}$$

B.1.2. The algorithm

We now describe the algorithmic steps to our auxiliary particle filter with M particles. As discussed in the main text, the structure of our model allows us to first resample and then move particles. The detailed algorithm consists of the following steps. To simplify notation, when we write a superscript i, we always mean $i = 1, 2, \ldots, M$.

Initialization. At time $t = 0$, set the state variables as $V_0^{(i)} = V_0$ and $\lambda_0^{(i)} = \lambda_0$, respectively, and give each set of particles a weight $W_0^{(i)} = 1/M$;

Time recursion. For $t = 1, 2, \ldots, T$, we have an equally weighted sample, $(V_{t-1}^{(i)}, \lambda_{t-1}^{(i)})$, and $W_{t-1}^{(i)} = 1/M$, representing the filtering distribution at $t-1$, $p(V_{t-1}, \lambda_{t-1} \mid y_{1:t-1})$.

- **Step 1 (Peak Forward):** The predictive density of the new density given the past states, $p(y_t \mid V_{t-1}, \lambda_{t-1})$, is available in closed form. The smoothing distribution over the past states is connected to the filtering density through:

  $$p(V_{t-1}, \lambda_{t-1} \mid y_{1:t}) \propto p(y_t \mid V_{t-1}, \lambda_{t-1}) p(V_{t-1}, \lambda_{t-1} \mid y_{1:t-1}).$$

  To represent the smoothing density by weighted particles, we need then to operationalize this by reweighting our particles:

  $$w^{(i)}_{t-1|t} = p(y_t \mid V^{(i)}_{t-1}, \lambda^{(i)}_{t-1}) \times w^{(i)}_{t-1}.$$

  Further, the likelihood of the new observation can be approximated as follows

  $$\hat{p}(y_t \mid y_{1:t-1}) = \frac{1}{M}\sum_{i=1}^{M} w^{(i)}_{t-1|t}.$$

- **Step 2** (**Resample** V_{t-1}, λ_{t-1}): Resample the particles proportional to the normalized weights $W^{(i)}_{t-1|t} = \frac{w^{(i)}_{t-1|t}}{\sum_{i=1}^M w^{(i)}_{t-1|t}}$ to arrive to an equally weighted distribution of the smoothing distribution, $p(V_{t-1}, \lambda_{t-1} \mid y_{1:t})$. We denote this sample as $(V^{(i)}_{t-1|t}, \lambda^{(i)}_{t-1|t})$.
- **Step 3** (**Generate from the jump counter** ΔN_t): The conditional distribution of the ΔN_t given the new observation is known analytically and can be trivially sampled from a Bernoulli distribution with probabilities:

  $$\begin{aligned}
  \text{For } i = 0, 1 \ : \ & P(\Delta N^{(i)}_t = i \, y_t, \lambda^{(i)}_{t-1|t}, V^{(i)}_{t-1|t}) \\
  &= \frac{f(y_t \mid \Delta N_t = i, \lambda^{(i)}_{t-1|t}, V^{(i)}_{t-1|t}) P(\Delta N_t = i \mid \lambda^{(i)}_{t-1|t}, V^{(i)}_{t-1|t})}{f(y_t \mid \lambda^{(i)}_{t-1|t}, V^{(i)}_{t-1|t})}
  \end{aligned}$$

- **Step 4** (**Generate from the variance jumps** $J_{v,t}$): The conditional distribution of the variance jumps is a truncated normal that can be easily sampled using the inverse CDF method,

$$J_{v,t}^{(i)}|\Delta N_t^{(i)} = 0, \lambda_{t-1|t}^{(i)}, V_{t-1|t}^{(i)} = 0,$$
$$J_{v,t}^{(i)}|\Delta N_t^{(i)} = 1, \lambda_{t-1|t}^{(i)}, V_{t-1|t}^{(i)} \sim N(\mu_{J_{v,t}}, \sigma_{J_{v,t}}), \text{ truncated to } J_{v,t} \geq 0.$$

- **Step 5** (**Generate from** λ_t, V_t): Given Eqs. (18) and (19), the conditional density $p(V_t, \lambda_t \mid V_{t-1}, \lambda_{t-1}, \Delta N_t, J_{v,t})$ is bivariate normal, i.e., we have effectively a conditionally linear Gaussian transition equation. Furthermore, given the stock price dynamics in Eq. (16), and the linear variance swap pricing formula in Eq. (14) together with the assumption of additive Gaussian noise on the variance swap observations in Eq. (17), the measurement equation is also locally linear Gaussian. Hence, conditional on V_{t-1}, λ_{t-1}, $\Delta N_t, J_{v,t}$, the states, (V_t, λ_t), and the observations, y_t, have a joint multivariate normal distribution and the standard Kalman update applies to obtain the conditional distribution of the states conditional on the observations. Following the Kalman update, the conditional distribution of the states, $p(V_t, \lambda_t \mid y_t, V_{t-1|t}^{(i)}, \lambda_{t-1|t}^{(i)}, \Delta N_t^{(i)}, J_{v,t}^{(i)})$, is normal and can be straightforwardly sampled, providing an equally weighted sample of the filtering distribution, $(V_t^{(i)}, \lambda_t^{(i)})$ and $W_t^{(i)} = 1/M$.

B.2. The Smooth Particle Filter (Smooth-PF)

In what follows, we only describe the steps where the smooth approximate particle filter differs from the auxiliary particle filter described in Appendix B.1.

Time recursion. For $t = 1, 2, \ldots, T$, generate M independent 6-dimensional uniform random numbers, $u_t^{(i)}$. Then the steps are as in the auxiliary particle filter, with the following differences:

- **Step 2b** (**Generate from** $V_{t-1|t}, \lambda_{t-1|t}$): The crucial point in the proposed filter is to avoid the irregularity introduced by the resampling steps. Hence, instead of resampling, we fit a multivariate normal distribution on the smoothing distribution and sample from this normal using the inverse CDF method. Denote by $\mu_{t-1|t}$ and $P_{t-1|t}$ the mean and covariance matrix of the weighed sample, $(\lambda_{t-1}^{(i)}, V_{t-1}^{(i)})$ and $W_{t-1|t}^{(i)}$. Then we use these moments and the first two coordinates of the uniform set, $u_{t,1}^{(i)}$ and $u_{t,2}^{(i)}$, to generate $V_{t-1|t}^{(i)}$ and $\lambda_{t-1|t}^{(i)}$:

$$(V_{t-1|t}^{(i)}, \lambda_{t-1|t}^{(i)}) = \mu_{t-1|t} + chol(P_{t-1|t})(\Phi^{-1}(u_{t,1}^{(i)}), \Phi^{-1}(u_{t,2}^{(i)})),$$

  where *chol* denotes the lower diagonal Cholesky decomposition. To decrease the extra variance, it is recommended to add a moment-matching step to ensure that the resulting particle set has the exactly same mean and variance as $\mu_{t-1|t}$ and $P_{t-1|t}$.
- **Step 3b** (**Generate from the jump counter** ΔN_t): The third coordinate of the uniform set, $u_{t,3}^{(i)}$, is used to generate from the jump counter. A straightforward application of the conditional sampler leads to:

$$\Delta N_t^{((i))} = 1_{\{u_{t,3}^{(i)} < P(\Delta N_t = 1 | y_t, V_{t-1|t}^{(i)}, \lambda_{t-1|t}^{(i)})\}}.$$

- **Step 4b** (**Generate from the variance jumps** $J_{v,t}$): The fourth coordinate of the uniform set, $u_{t,4}^{(i)}$, is used to generate from the variance jumps. As the normal CDF and the inverse normal CDF used within simulating the truncated normal are smooth, there are no complications here.
- **Step 5b** (**Generate from** λ_t, V_t): Following the Kalman update, the conditional distribution of the states, $p(V_t, \lambda_t \mid y_t, V_{t-1|t}^{(i)}, \lambda_{t-1|t}^{(i)}, \Delta N_t^{(i)}, J_{v,t}^{(i)})$, is normal and can be straightforwardly sampled. Denote the conditional mean and variance by $\mu_{t|t}^{(i)}$ and $P_{t|t}^{(i)}$. Then we use these moments and the last two coordinates of the uniform set, $u_{t,5}^{(i)}$ and $u_{t,6}^{(i)}$, to generate $V_t^{(i)}$ and $\lambda_t^{(i)}$:

$$(V_t^{(i)}, \lambda_t^{(i)}) = \mu_{t|t} + chol(P_{t|t})(\Phi^{-1}(u_{t,5}^{(i)}), \Phi^{-1}(u_{t,6}^{(i)})).$$

  This provides us with the weighted sample, $(V_t^{(i)}, \lambda_t^{(i)})$ and $W_t^{(i)}$, representing the filtering density at t.

Appendix C. Two-stage SMC sampler with common random numbers

In this appendix, we describe in detail the steps in the two-stage SMC sampler we propose for parameter estimation. Fix some random numbers, $u_{1:T}^{M_k} = \{u_{1:T}^{(l)}; l = 1, \ldots, M_k\}$ for $k = 1, 2$ and $M_1 \ll M_2$, to be used in the particle filter across all fix parameter sets and denote the likelihood estimate at the fixed parameter set, Θ, from running the smooth particle filter in Appendix B.2 as $\hat{p}(y_{1:T}|\Theta, u_{1:T}^{M_k})$ for the given number of state particles, M_k.

C.1. Stage 1

As in Del Moral et al. (2006), we propagate N parameter particles through the sequence of densities defined in Eq. (40) using sequential Monte Carlo sampling. In what follows, the superscript j always means for each j in $(1, \ldots, N)$.

Initialization. At $i = 1$, we need to obtain N samples from $\pi_{1,1}(\Theta) = p(\Theta)$:

- Sample $\Theta^{(j)}(1) \sim p(\Theta)$ from the prior distribution over the fixed parameters.
- Run the smooth particle filter for each $\Theta^{(j)}(1)$ at the random numbers $u_{1:T}^{M_1}$ and record the likelihood estimate $\hat{p}(y_{1:T}|\Theta^{(j)}(i), u_{1:T}^{M_1})$
- Attribute an equal weight, $S^{(j)}(1) = 1/N$, to each particle.

For $i = 2, \ldots, I_1$, the weighted sample at the $(i-1)$th iteration, i.e., $(S^{(j)}(i-1), \Theta^{(j)}(i-1))$ represents $\pi_{1,i-1}(\Theta)$. The algorithm then goes through the following steps to advance to $\pi_{1,i}(\Theta)$.

Reweighting. Moving from $\pi_{1,i-1}(\Theta)$ to $\pi_{1,i}(\Theta)$ can be implemented by reweighting the particles using the ratio of the two densities. This yields the following unnormalized incremental weights:

$$\tilde{s}^{(j)}(i) = \frac{\gamma_{1,i}(\theta^{(j)}(i-1))}{\gamma_{1,i-1}(\theta^{(j)}(i-1))} = \left[\hat{p}(y_{1:T}|\theta^{(j)}(i-1), u_{1:T}^{M_1})\right]^{\xi_{1,i}-\xi_{1,i-1}}.$$

Del Moral et al. (2012) show that the tempering sequence can be adaptively chosen to ensure sufficient particle diversity. We follow their procedure and always set the next value of the tempering sequence $\xi_{1,i}$ automatically to ensure that the effective sample size, $ESS = \frac{1}{\frac{1}{N}\sum_{j=1}^{N}[S^{(j)}(i-1)]^2}$, stays close to some constant B. This is achieved by a simple grid search, where the ESS is evaluated at the grid points of $\xi_{1,i}$ and the one with ESS closest to B is chosen.

Furthermore, the identity $\frac{Z_{1,i}}{Z_{1,i-1}} = \int_{\Theta} \frac{\gamma_i(\Theta)}{\gamma_{i-1}(\Theta)}\pi_{i-1}(\Theta)d(\Theta)$ suggests that the ratio of the normalizing constants can be estimated as

$$\widehat{\left[\frac{Z_{1,i}}{Z_{1,i-1}}\right]} = \sum_{j=1}^{N} S^{(j)}(i-1)\tilde{s}^{(j)}(i).$$

Finally, the weights are normalized and become $S^{(j)}(i) = \frac{S^{(j)}(i-1)\tilde{s}^{(j)}(i)}{\sum_{k=1}^{N} S^{(k)}(i-1)\tilde{s}^{(k)}(i)}$.

Resampling. As the algorithm proceeds, the variability of the importance weights will tend to increase, leading to sample impoverishment. In order to focus the computational effort on areas of high probability, it is common to resample particles proportional to their weights. When $ESS < B$, we resample the particles $\Theta^{(j)}(i-1)$ proportional to $S^{(j)}(i-1)$ and after resampling, set $S^{(j)}(i-1) = \frac{1}{N}$.

Moving. With repeatedly reweighting and resampling, the support of the sample of the parameters gradually deteriorates, leading to a well-known particle depletion situation. Periodically boosting the support is thus a must. For this purpose, we resort to Metropolis–Hastings (MH) moves and keep the target, $\gamma_{1,i}(\Theta)$, unchanged. A new particle Θ^* is proposed using the proposal density $h_i(\Theta^*|\Theta)$. The resulting MH move step looks as follows:

- Draw parameters from $\Theta^* \sim h_i(\cdot|\Theta^{(j)}(i-1))$. Run the smooth particle filter on Θ^* at the random numbers $u_{1:T}^{M_1}$ and compute the likelihood estimate $\hat{p}(y_{1:T}|\Theta^*, u_{1:T}^{M_1})$.
- The MH acceptance rate is $\alpha = 1 \wedge \tilde{\alpha}$, where

$$\tilde{\alpha} = \frac{\left[\hat{p}(y_{1:T}|\Theta^*, u_{1:T}^{M_1})\right]^{\xi_{1,i}} p(\Theta^*)}{\left[\hat{p}(y_{1:T}|\Theta^{(j)}(i-1), u_{1:T}^{M_1})\right]^{\xi_{1,i}} p(\Theta^{(j)}(i-1))} \frac{h_i(\Theta^{(j)}(i-1) \mid \Theta^*)}{h_i(\Theta^* \mid \Theta^{(j)}(i-1))}.$$

That is, $\Theta^{(j)}(i) = \Theta^*$ with probability α and $\Theta^{(j)}(i) = \Theta^{(j)}(i-1)$ with probability of $1-\alpha$.

C.2. Stage 2

At the second stage, we propagate the parameter particles through the density sequence defined in Eq. (42).

Initialization. At $i = 1$, we need to represent $\hat{p}(y_{1:T}|\Theta, u_{1:T}^{M_1})p(\Theta)$, which is exactly the output of the first stage:

- Both the parameter particles $\Theta^{(j)}(1)$ and the normalized weights $S^{(j)}(1)$ are the output of the last iteration of Stage 1.
- We run the smooth particle filter for each $\Theta^{(j)}(1)$ at the random numbers $u_{1:T}^{M_2}$ and record the likelihood estimate $\hat{p}(y_{1:T}|\Theta^{(j)}(1), u_{1:T}^{M_2})$.

For $i = 1, \ldots, I_2$, we have similar reweighting, resampling, and moving steps as in Stage 1 for proceeding from $\pi_{2,i-1}(\Theta)$ to $\pi_{2,i}(\Theta)$. The implementation of Stage 2 differs from Stage 1 only in the following points:

Reweighting. Moving from $\pi_{2,i-1}(\Theta)$ to $\pi_{1,2}(\Theta)$ can be implemented by reweighting the particles by the ratio of the two densities. This yields the following unnormalized incremental weights:

$$\tilde{s}^{(j)}(i) = \frac{\gamma_{2,i}(\Theta^{(j)}(i-1))}{\gamma_{2,i-1}(\Theta^{(j)}(i-1))} = \left[\frac{\hat{p}(y_{1:T}|\Theta^{(j)}(i-1), u_{1:T}^{M_2})}{\hat{p}(y_{1:T}|\Theta^{(j)}(i-1), u_{1:T}^{M_1})}\right]^{\xi_{2,i}-\xi_{2,i-1}}.$$

Notice that compared to the first stage, the key quantity in the incremental weights is the ratio of the likelihood estimates at M_2 and M_1, showing that the role of the second stage is just to decrease the Monte Carlo noise in the likelihood estimate. Further, it is important to notice that the use of common random numbers across the parameter sets means that the Monte Carlo noise in this ratio is highly positively correlated across $\Theta^{(j)}(i-1)$, hence reducing the effect of this noise on the variability of the weights $\tilde{s}^{(j)}(i)$. These two features will result in a much smaller number of bridging densities in the second stage compared to the first one.

Moving. The move steps will consist of the following:

- Draw parameters from $\Theta^* \sim h_i(\cdot|\Theta^{(i)}(i-1))$. Run the smooth particle filter on Θ^* both at the random numbers $u_{1:T}^{M_1}$ and $u_{1:T}^{M_2}$ and compute the corresponding likelihood estimates, $\hat{p}(y_{1:T}|\Theta^*, u_{1:T}^{M_1})$ and $\hat{p}(y_{1:T}|\Theta^*, u_{1:T}^{M_2})$.
- The MH acceptance rate is $\alpha = 1 \wedge \tilde{\alpha}$, where

$$\tilde{\alpha} = \frac{\left[\hat{p}(y_{1:T}|\Theta^*, u_{1:T}^{M_2})\right]^{\xi_{2,i}}\left[\hat{p}(y_{1:T}|\Theta^*, u_{1:T}^{M_1})\right]^{1-\xi_{2,i}} p(\Theta^*)}{\left[\hat{p}(y_{1:T}|\Theta^{(j)}(i-1), u_{1:T}^{M_2})\right]^{\xi_{2,i}}\left[\hat{p}(y_{1:T}|\Theta^{(j)}(i-1), u_{1:T}^{M_1})\right]^{1-\xi_{2,i}} p(\Theta^{(j)}(i-1))} \frac{h_i(\Theta^{(j)}(i-1) \mid \Theta^*)}{h_i(\Theta^* \mid \Theta^{(j)}(i-1))}.$$

That is, $\Theta^{(j)}(i) = \Theta^*$ with probability α and $\Theta^{(j)}(i) = \Theta^{(j)}(i-1)$ with probability of $1-\alpha$.

Appendix D. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.jeconom.2018.11.014.

References

Ait-Sahalia, Y., Cacho-Diaz, J., Laeven, R., 2015. Modeling financial contagion using mutually exciting jump processes. J. Financ. Econ. 117, 585–606.
Ait-Sahalia, Y., Kimmel, R., 2007. Maximum likelihood estimation of stochastic volatility models. J. Financ. Econ. 83, 413–452.
Amengual, D., Xiu, D., 2018. Resolution of policy uncertainty and sudden declines in volatility. J. Econometrics 203, 297–315.
Andrieu, C., Doucet, A., Holenstein, R., 2010. Particle markov chain monte carlo (with discussions). J. R. Stat. Soc. Ser. B Stat. Methodol. 72, 269–342.
Andrieu, C., Roberts, G.O., 2009. The pseudo-marginal approach for efficient computation. Ann. Statist. 37, 697–725.
Broadie, M., Chernov, M., Johannes, M., 2007. Model specification and risk premia: Evidence from futures options. J. Finance 62, 1453–1490.
Carr, P., Wu, L., 2017. Leverage effect, volatility feedback, and self-exciting market disruptions. J. Financ. Quant. Anal. 52, 2119–2156.
Chopin, N., Jacob, P.E., Papaspiliopoulos, O., 2013. SMC^2: A sequential monte carlo algorithm with particle markov chain monte carlo updates. J. Roy. Statist. Soc. Ser. B 75, 397–426.
Christoffersen, P., Jacobs, K., Mimouni, K., 2010. Volatility dynamics for the S&P 500: Evidence from realized volatility, daily returns, and option prices. Rev. Financ. Stud. 23, 3141–3189.
Del Moral, P., 2004. Feyman-Kac Formulae Genealogical and Interacting Particle Systems with Applications. Springer, New York.
Del Moral, P., Doucet, A., Jasra, A., 2006. Sequential monte carlo samplers. J. Roy. Statist. Soc. Ser. B 68, 411–436.
Del Moral, P., Doucet, A., Jasra, A., 2012. An adaptive sequential monte carlo method for approximate bayesian computation. Stat. Comput. 22, 1009–1020.
Deligiannidis, G., Doucet, A., Pitt, K., 2018. The correlated pseudomarginal method. J. Roy. Statist. Soc. Ser. B 80, 839–870.
Doucet, A., Godsill, S., Andrieu, C., 2000. On sequential monte carlo sampling methods for bayesian filtering. Stat. Comput. 10, 197–208.
Doucet, A., Pitt, M.K., Deligiannidis, G., Kohn, R., 2015. Efficient implementation of markov chain monte carlo when using an unbiased likelihood estimator. Biometrika 102, 295–313.
Duan, J., Fulop, A., 2015. Density-tempered marginalized sequential monte carlo samplers. J. Bus. Econom. Statist. 33, 192–202.
Eraker, B., 2004. Do stock prices and volatility jump? Reconciling evidence from spot and option prices. J. Finance 59, 1367–1403.
Eraker, B., Johannes, M., Polson, N., 2003. The impact of jumps in volatility and returns. J. Finance 58, 1269–1300.
Fulop, A., Li, J., Yu, J., 2015. Self-exciting jumps, learning, and asset pricing implications. Rev. Financ. Stud. 28, 876–912.
Gerber, M., Chopin, N., 2015. Sequential quasi monte carlo. J. R. Stat. Soc. Ser. B Stat. Methodol. 77, 509–579.
Gordon, N., Salmond, D., Smith, A., 1993. Novel approach to nonlinear and non-gaussian bayesian state estimation. IEEE Proc.-F 140, 107–113.
Gourieroux, C., Monfort, A., 1996. Simulation-Based Econometric Methods. Oxford University Press, New York.
Jacob, P., Lindsten, F., Schon, T.B., 2016. Coupling of Particle Filters. Working Paper.
Jacod, J., Todorov, V., 2010. Do price and volatility jump together? Ann. Appl. Probab. 20, 1425–1469.
Johannes, M., Polson, N., 2009. MCMC methods for financial econometrics. In: Hansen, L.P., Ait-Sahalia, Y. (Eds.), Handbook of Financial Econometrics. Elsevier, Amsterdam.
Johannes, M., Polson, N., Stroud, J., 2009. Optimal filtering of jump diffusions: Extracting latent states from asset prices. Rev. Financ. Stud. 22, 2559–2599.
Johansen, A.M., Doucet, A., Davy, M., 2008. Particle methods for maximum likelihood estimation in latent variable models. Stat. Comput. 18, 47–57.
Jones, C., 2003. The dynamics of stochastic volatility: Evidence from underlying and options markets. J. Econometrics 116, 181–224.
Li, H., Wells, M.T., Yu, C.L., 2008. A bayesian analysis of return dynamics with lévy jumps. Rev. Financ. Stud. 21, 2345–2378.
Malik, S., Pitt, M., 2011. Particle filters for continuous likelihood evaluation and maximisation. J. Econometrics 165, 190–209.
Pan, J., 2002. The jump-risk premia implicit in options: Evidence from an integrated time-series study. J. Financ. Econ. 63, 3–50.
Pitt, M.K., Shephard, N., 1999. Filtering via simulation: Auxiliary particle filters. J. Amer. Statist. Assoc. 94, 590–591.
Todorov, V., Tauchen, G., 2011. Volatility jumps. J. Bus. Econom. Statist. 29, 356–371.
Yu, C.L., Li, H., Wells, M.T., 2011. MCMC estimation of levy jumps models using stock and option prices. Math. Finance 21, 383–422.