# Stochastic Variational Inference for GARCH Models

Hanwen Xuan[1*], Luca Maestrini[2†], Feng Chen[1,3†] and Clara Grazian[4,5†]

[1*]School of Mathematics and Statistics, University of New South Wales, Anita B. Lawrence Centre, Kensington, 2052, New South Wales, Australia.
[2]Research School of Finance, Actuarial Studies and Statistics, The Australian National University, Building 26C, Kingsley Street, Canberra, 2601, ACT, Australia.
[3]UNSW Data Science Hub, University of New South Wales, Anita B. Lawrence Centre, Kensington, 2052, NSW, Australia.
[4]School of Mathematics and Statistics, University of Sydney, Carslaw Building, Camperdown, 2006, NSW, Australia.
[5]ARC Training Centre in Data Analytics for Resources and Environments, University of Sydney, Biomedical Building, South Eveleigh, 2015, NSW, Australia.

*Corresponding author(s). E-mail(s): t.xuan@unsw.edu.au;
Contributing authors: luca.maestrini@anu.edu.au; feng.chen@unsw.edu.au;
clara.grazian@sydney.edu.au;
†These authors contributed equally.

**Abstract**

Stochastic variational inference algorithms are derived for fitting various heteroskedastic time series models. We examine Gaussian, t, and skewed t response GARCH models and fit these using Gaussian variational approximating densities. We implement efficient stochastic gradient ascent procedures based on the use of control variates or the reparameterization trick and demonstrate that the proposed implementations provide a fast and accurate alternative to Markov chain Monte Carlo sampling. Additionally, we present sequential updating versions of our variational algorithms, which are suitable for efficient portfolio construction and dynamic asset allocation.

**Keywords:** Bayesian inference, financial time series, skewed t innovation, stochastic gradient descent, variational Bayes.

## 1 Introduction

Most financial decisions are based on the trade-off between risk and return (Markowitz, 1952). Modeling volatility, which is often used to measure risk, is therefore crucial in risk management. Traditional econometric models assume that the variance of asset returns is constant, but this assumption may be unrealistic as it overlooks the presence of heteroskedasticity in financial time series (Engle, 2001). To address this issue, Engle (1982) introduced auto-regressive conditional heteroskedastic (ARCH) models, which utilize lagged disturbance terms to explain a time-varying conditional variance process. ARCH models were extended to the generalized ARCH (GARCH) class by Bollerslev (1986) to allow for a more flexible lag structure where the current conditional

variance depends not only on the past conditional variances but also on the past disturbance terms. Currently, GARCH models are widely employed to describe the dynamics of volatility processes; however, conventional ARCH and GARCH models assume Gaussian errors, which fail to capture common features of financial time series such as heavy-tailedness and skewness. To improve both goodness-of-fit and prediction accuracy, Bollerslev (1987) and Lambert and Laurent (2001) respectively studied the use of t and skewed t distributions for modeling error terms (see also Degiannakis, 2004).

Maximum likelihood estimation is typically used to estimate the parameters of GARCH models (González-Rivera and Drost, 1999), as it guarantees ease of implementation and asymptotic efficiency (Silvennoinen and Teräsvirta, 2021). However, uncertainty quantification of maximum likelihood estimates is usually performed by taking advantage of their asymptotic normality (Nakatsuma, 2000), which may lead to underestimating uncertainty and model risk as a consequence. Bayesian inference offers an attractive alternative as it naturally expresses the uncertainty around the estimation process by returning a posterior probability distribution for the model parameters. In a Bayesian context, Nakatsuma (2000) proposed a Markov chain Monte Carlo (MCMC) method with Metropolis-Hasting steps for fitting GARCH models. Ardia et al (2008) used MCMC to fit regime-switching GARCH models with t innovations. A fully automated MCMC procedure for Bayesian estimation in GARCH models with t innovations is available in the `R` package `bayesGARCH` (Ardia and Hoogerheide, 2010). Li et al (2021) propose a sequential Monte Carlo algorithm for Student t GARCH models. While several MCMC methods are available for GARCH models (Gerlach and Tuyl, 2006; Henneke et al, 2011; Virbickaite et al, 2015; Contino and Gerlach, 2017; Xaba et al, 2021), these may suffer from slow convergence and be computationally intensive, in particular for long time series (Shiferaw, 2019; Mozumder et al, 2021; Livingston Jr and Nur, 2022).

Variational Bayes (VB) methods (Jordan et al, 1998) have emerged as a more computationally efficient alternative to MCMC. Early research on variational inference focused on deriving algorithms with closed-form updates for conjugate models – see, for example, Bishop and Nasrabadi (2006), Ormerod and Wand (2010) or Maestrini and Wand (2018) for variational inference through conjugate model representations, including for skewed t regression. Over recent years, stochastic gradient methods for variational inference (e.g., Paisley et al, 2012) have found wider interest, especially for non-conjugate models and as an alternative to avoid lengthy mathematical derivations.

However, variational inference has found limited attention in the literature on GARCH models. This is partially due to the characteristics of time series, which represent an obstacle to scalable algorithm implementations. Hoffman et al (2013) developed a scalable stochastic variational inference algorithm to analyze massive datasets for complex topic models by subsampling the data to form noisy estimates. However, there is no clear and unbiased way to subsample time series data for GARCH models (Politis, 2003; Alonso and Maharaj, 2006). Tran et al (2021a) proposed a manifold-based variational Bayes approach for GARCH models with Gaussian error terms that maps the variational parameters from the Euclidean space to a Riemannian manifold, but its extension to models with non-Gaussian errors is not immediate.

In recent years there have been several attempts to develop sequential variational inference approaches for time series models, i.e., implementations of variational algorithms that update inference as soon as new data points of the series are available. Lambert et al (2022) proposed recursive Gaussian variational approximations for models assuming observations are independent, which are therefore not suitable for GARCH models and time-dependent data. Tomasetti et al (2022) developed an updating VB (UVB) method as a sequential extension of *batch stochastic variational approximations* for streaming data (Broderick et al, 2013; Foti et al, 2014). This method is based on deriving a sequence of posterior distributions that are used as priors for the following steps through updates resulting from Bayes' theorem. Each posterior update only requires data observed in the previous step. While very efficient, UVB targets the approximating pseudo-posterior rather than the true posterior, resulting in reduced accuracy. An alternative sequential implementation is

the one proposed by Gunawan et al (2021), which targets the true posterior distribution instead. Their approach is named sequential stochastic VB (Seq-SVB) and it is used to perform prediction with factor stochastic volatility models. A potential drawback of Seq-SVB is that the likelihood function is computed for all the time series, affecting computational efficiency.

This work includes two main contributions. First, we develop stochastic variational Bayes (SVB) algorithms to approximate the posterior distributions of parameters in GARCH models with Gaussian, t and skewed t errors (also known as innovations in the econometric literature). Our variational approximations are compared to an MCMC target. We assess the performance of variational inference using control variates or the reparametrization trick in terms of computational speed and estimation accuracy. Results demonstrate that our stochastic variational implementations are accurate while being faster than MCMC. Our second contribution is the adaptation of the stochastic VB approaches to allow for sequential updates. We implement the procedures developed by Gunawan et al (2021) and Tomasetti et al (2022) and compare them for the first time, in the specific case of GARCH models. Our results demonstrate that both UVB and Seq-SVB are faster than the base SVB and show the presence of a trade-off between accuracy and computational costs.

The remainder of the paper is organized as follows. Section 2 reviews GARCH models, variational Bayes approximations and their stochastic versions. Section 3 develops VB algorithms for GARCH models with different innovation terms. Section 4 proposes sequential versions of the variational algorithms from the previous section for efficiently updating posterior distributions when new observations are available. Section 5 presents the results of our simulation experiments and Section 6 illustrates the use of VB for real-world financial time series. Section 7 concludes the paper with a discussion.

# 2 Stochastic Variational Bayes for GARCH Models

## 2.1 GARCH Models

Let $y_1, \ldots, y_T$ be a time series, for instance the log-difference of stock prices at two consecutive time points, i.e., $y_t = \log P_t - \log P_{t-1}$. The GARCH$(p, q)$ model for the $t$th observation, $t = 1, \ldots, T$, takes the form

$$\begin{aligned} y_t &= \sigma_t \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0,1), \\ \sigma_t^2 &= \omega + \textstyle\sum_{i=1}^{p} \alpha_i y_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2. \end{aligned} \tag{1}$$

The time-varying volatility process for $\sigma_t^2$ follows a deterministic autoregressive structure and here we implicitly assume the dependence of this structure on the $\alpha_i$ and $\beta_j$ parameters writing $\sigma_t^2$ instead of $\sigma_t^2(\alpha_1, \ldots, \alpha_p, \beta_1, \ldots, \beta_q)$. The success of this class of models lies in the fact that the log-likelihood function $\ell(\theta) = \log L(\theta;\ y)$ of $\theta = (\omega, \alpha_1, \ldots, \alpha_p, \beta_1, \ldots, \beta_q)$ is tractable and has the following expression for model (1):

$$\ell(\theta) = -\frac{1}{2} \sum_{t=1}^{T} \left\{ \log(2\pi) + \log(\sigma_t^2) + \frac{y_t^2}{\sigma_t^2} \right\}. \tag{2}$$

In real applications it has been widely recognized that financial returns exhibit significant deviations from normality, and typically have heavier tails than the normal distribution (Shirvani et al, 2020). To address this issue, GARCH models have been extended to include t error terms (Bollerslev, 1987; Baillie and Bollerslev, 1989; Beine et al, 2002). This is achieved by replacing the normal distribution assumption in (1) with $\varepsilon_t \sim \mathcal{T}(0, 1;\ \nu)$, where $\mathcal{T}(0, 1;\ \nu)$ represents a standardized t distribution with degrees of freedom $\nu$. A random variable $x$ has a standardized t distribution $\mathcal{T}(0, 1;\ \nu)$ with degrees of freedom $\nu > 2$ if its probability density function is

$$f(x) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi(\nu-2)}\,\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{x^2}{\nu-2}\right)^{-\frac{\nu+1}{2}}.$$

The log-likelihood function for the t GARCH$(p, q)$ model defined by the parameter vector $\theta =$

$(\nu, \omega, \alpha_1, \ldots, \alpha_p, \beta_1, \ldots, \beta_q)$ is

$$\ell(\theta) = T\Bigg\{ \log\Gamma\Big(\frac{\nu+1}{2}\Big) - \log\Gamma\Big(\frac{\nu}{2}\Big) - \frac{1}{2}\log\big\{\pi(\nu-2)\big\}\Bigg\} - \frac{1}{2}\sum_{t=1}^{T}\Bigg[\log(\sigma_t^2) + (\nu+1)\log\Bigg\{1 + \frac{y_t^2}{(\nu-2)\sigma_t^2}\Bigg\}\Bigg]. \quad (3)$$

Skewness is a pervasive feature observed in empirical data which can be taken into account in financial time series modeling and inference (Jondeau et al, 2019). Lambert and Laurent (2001) incorporated the skewed t distribution introduced by Fernández and Steel (1998) in GARCH models to account for skewness. Skewed t distributions are highly flexible and introduce asymmetry in the data model by adding only one parameter to the t distribution. In the corresponding model, the normal distribution assumption in (1) is replaced by $\varepsilon_t \sim \mathcal{ST}(0, 1;\ \nu, \xi)$, where the skewness parameter $\xi$ is a positive real number. A value of $\xi < 1$ indicates negative skewness, while a value of $\xi > 1$ indicates positive skewness. We follow the framework and definition provided by Fernández and Steel (1998): a random variable $x$ has a skewed t distribution $\mathcal{ST}(0, 1;\ \nu, \xi)$ with degrees of freedom $\nu > 2$ and skewness parameter $\xi > 0$ if its probability density function is

$$f(x) = \frac{2s\Gamma\left(\frac{\nu+1}{2}\right)}{\left(\xi + \frac{1}{\xi}\right)\sqrt{\pi(\nu-2)}\Gamma\left(\frac{\nu}{2}\right)} \times \left\{1 + \frac{(sx+m)^2}{\nu-2}\xi^{-2I_t}\right\}^{-\frac{\nu+1}{2}},$$

where

$$I_t = \begin{cases} 1, & \text{if} \quad x \geq -m/s, \\ -1, & \text{if} \quad x < -m/s, \end{cases}$$

$$m = \Gamma\left(\frac{\nu-1}{2}\right)\left\{\Gamma\left(\frac{\nu}{2}\right)\right\}^{-1}\frac{\sqrt{\nu-2}}{\sqrt{\pi}}\left(\xi - 1/\xi\right)$$

and $s = \sqrt{(\xi^2 + 1/\xi^2 - 1) - m^2}$.

Note that the density function has been standardized to have a zero mean and unit variance. Defining the vector of parameters as $\theta = (\nu, \xi, \omega, \alpha_1, \ldots, \alpha_p, \beta_1, \ldots, \beta_q)$, the log-likelihood for the skewed t GARCH model with $T$ observations is given by

$$\ell(\theta) = T\Bigg\{ \log\Gamma\Big(\frac{\nu+1}{2}\Big) - \log\Gamma\Big(\frac{\nu}{2}\Big) - \frac{1}{2}\log\big\{\pi(\nu-2)\big\} + \log\left(\frac{2}{\xi+1/\xi}\right) + \log(s)\Bigg\} - \frac{1}{2}\sum_{t=1}^{T}\Bigg[\log(\sigma_t^2) + (1+\nu) \log\Bigg\{1 + \frac{(sy_t/\sigma_t + m)^2}{\nu-2}\xi^{-2I_t}\Bigg\}\Bigg]. \quad (4)$$

While the likelihood is still tractable, inference based on MCMC may be computationally costly in this case, and convergence slow (Iqbal and Triantafyllopoulos, 2021). Alternative skewed t GARCH-type models have been presented, e.g. in Hansen (1994) and Liu et al (2015). We anticipate the implementation of stochastic variational inference for these alternative skewed t GARCH models is similar.

## 2.2 Variational Inference

Let $y$ denote a vector of observed data, and let $\theta$ be a vector of parameters. In a Bayesian setting, given a prior distribution $p(\theta)$ and a likelihood function $L(\theta;\ y)$, the prior beliefs about $\theta$ are updated after collecting the data to obtain the posterior distribution:

$$p(\theta|y) = \frac{L(\theta;\ y)p(\theta)}{p(y)} \propto L(\theta;\ y)p(\theta),$$

where $p(y)$ is the marginal likelihood, or evidence. In practice, the posterior distribution in Bayesian inference may be intractable, as its normalizing factor, or evidence, $p(y)$, is often not available in closed form. Consequently, simulation-based Markov chain Monte Carlo (MCMC) methods are typically employed to estimate the posterior distribution through sampling (Robert et al, 1999). However, MCMC sampling can be computationally expensive or impractical for complex models. An alternative approach is provided by variational Bayes (VB), which aims to approximate the intractable posterior distribution using a variational approximating density $q(\theta)$ selected from a family of densities $\mathcal{Q}$. In VB, the approximating density is chosen to minimize the Kullback-Leibler

(KL) divergence between the variational approximating density and the true posterior density.

$$q_\lambda^*(\theta) = \underset{q_\lambda \in \mathcal{Q}, \lambda \in \mathcal{M}}{\arg\min}\ \mathrm{KL}\big\{q_\lambda(\theta) \| p(\theta|y)\big\} = \underset{q_\lambda \in \mathcal{Q}, \lambda \in \mathcal{M}}{\arg\min} \int q_\lambda(\theta) \log \frac{q_\lambda(\theta)}{p(\theta|y)} d\theta,$$

where $\lambda$ denotes a vector of parameters of the variational approximating density in the parameter space $\mathcal{M}$. Minimization of the KL divergence is equivalent to maximizing the evidence lower bound (ELBO) $\mathcal{L}(\lambda)$ (see, e.g., Bishop and Nasrabadi, 2006 or Ranganath et al, 2014), which is defined as

$$\mathcal{L}(\lambda) = \int q_\lambda(\theta) \log \frac{L(\theta;\ y) p(\theta)}{q_\lambda(\theta)} d\theta. \qquad (5)$$

In general, $\mathcal{L}(\lambda)$ does not have a closed-form solution in non-conjugate models. Stochastic variational inference approaches based on stochastic gradient methods (Paisley et al, 2012; Nott et al, 2012; Hoffman et al, 2013; Kingma and Welling, 2013; Ranganath et al, 2014; Rezende et al, 2014) allow for efficient optimization of $\mathcal{L}(\lambda)$, even when the ELBO cannot be evaluated analytically. In stochastic variational inference an initial value of the variational parameters $\lambda^{(0)}$ is stated and iteratively updated through

$$\lambda^{(i+1)} = \lambda^{(i)} + a_i \widehat{\nabla_\lambda \mathcal{L}(\lambda^{(i)})}$$

until a stopping rule is satisfied. Here $\widehat{\nabla_\lambda \mathcal{L}(\lambda^{(i)})}$ denotes an unbiased estimator of the gradient vector of $\mathcal{L}(\lambda)$ computed with respect to $\lambda$ and $\{a_i, i \geq 1\}$ is a sequence of vector-valued learning rates satisfying the Robbins-Monro conditions $\sum_i a_i = \infty$ and $\sum_i a_i^2 < \infty$ (Robbins and Monro, 1951). The topic of adaptive learning rates has been widely discussed in the literature (e.g., Polyak, 1964; Duchi et al, 2011; Tieleman et al, 2012; Zeiler, 2012). In this work, we adopt ADAM adaptive learning rates (Kingma and Ba, 2014) and a moving average stopping criterion (see Tran et al, 2021b for more details).

It is possible to show that the gradient vector can be written in an expectation form computed with respect to the variational density $q_\lambda(\theta)$ as

$$\nabla_\lambda \mathcal{L}(\lambda) = \mathbb{E}_{q_\lambda}\left\{h_\lambda(\theta) \nabla_\lambda \log q_\lambda(\theta)\right\}, \qquad (6)$$

where $h_\lambda(\theta) = \log\left\{p(\theta) L(\theta;\ y) / q_\lambda(\theta)\right\}$ (e.g., see Tran et al, 2021b). A simple estimator of the $j$-th element of the ELBO gradient is then given by

$$\widehat{\nabla_{\lambda_j} \mathcal{L}(\lambda)} = \frac{1}{S} \sum_{s=1}^{S} h_\lambda(\theta_s) \nabla_{\lambda_j} \log q_\lambda(\theta_s), \qquad (7)$$

where $s = 1, \ldots, S$ are samples from the variational distribution $q_\lambda(\theta)$.

In this work, we consider a popular form of VB known as Gaussian variational approximation, which assumes $q_\lambda(\theta)$ to be a multivariate Gaussian distribution with mean $\mu$ and covariance matrix $\Sigma$. Several authors have studied and discussed the parameterization of Gaussian variational approximations; see, for example, Opper and Archambeau, 2009; Ormerod and Wand, 2012; Ong et al, 2018b; Quiroz et al, 2022. We adopt the Cholesky factorization suggested by Titsias and Lázaro-Gredilla (2014) and factorize the covariance matrix as $\Sigma = LL^\top$ where $L$ is a lower triangular matrix. Consequently, the true posterior distribution $p(\theta|y)$ can be approximated by a $\mathcal{N}_d(\mu, LL^\top)$ distribution; in this case $\lambda = (\mu^\top, \mathrm{vech}(L)^\top)^\top$, where $\mathrm{vech}(\cdot)$ is the half-vectorization operator. Introducing this factorization reduces the computational cost to reach convergence while preserving the positive definiteness of the covariance matrix $\Sigma$.

The performance of stochastic optimization algorithms, including stochastic VB greatly depends on the variance of the gradient estimator (Liu et al, 2019) and this has led to the development of computational solutions to reduce it. Paisley et al (2012) and Ranganath et al (2014) adopt a variance reduction technique based on the use of control variates to improve the convergence of stochastic VB, where the gradient estimator function is replaced by another function with equivalent expectation but smaller variance. For any arbitrary number $c_j$, the $j$-th element of the estimator version of (7) can be slightly modified to

$$\widehat{\nabla_{\lambda_j} \mathcal{L}(\lambda)} = \frac{1}{S} \sum_{s=1}^{S} \left(h_\lambda(\theta_s) - c_j\right) \nabla_{\lambda_j} \log q_\lambda(\theta_s), \qquad (8)$$

which is still an unbiased estimator of the gradient of $\mathcal{L}(\lambda)$ (Ranganath et al, 2014; Tran et al, 2021b). The optimal control variate $c_j$ that minimizes the variance can be calculated by taking the derivatives of the variance of the gradient estimator with

respect to $c_j$:

$$c_j = \frac{\mathbb{C}\text{ov}\left\{\nabla_{\lambda_j} \log q_\lambda(\theta), h_\lambda(\theta)\nabla_{\lambda_j} \log q_\lambda(\theta)\right\}}{\mathbb{V}\text{ar}\left\{\nabla_{\lambda_j} \log q_\lambda(\theta)\right\}}, \quad (9)$$

where $j = 1, \ldots, d$. This method requires minimal analytic calculations and can be applied to a variety of models.

The reparametrization trick proposed by Kingma and Welling (2013) (see also Kingma and Ba, 2014; Rezende et al, 2014; Titsias and Lázaro-Gredilla, 2014; Tran et al, 2021b) offers an attractive alternative to the control variates approach, which consists in rewriting an expectation so that the distribution with respect to which the gradient is taken is independent of the model parameters. Suppose samples are meant to be generated from $\theta \sim q_\lambda(\theta)$, then it is often possible to write $\theta = g_\lambda(\epsilon)$, where $\epsilon$ is an auxiliary variable with an independent marginal distribution $p(\epsilon)$ and $g_\lambda(\epsilon)$ is a deterministic function parameterized by $\lambda$. Using this deterministic transformation, it is possible to rewrite $\mathcal{L}(\lambda)$ in (5) as an expectation with respect to the alternative measure $p(\epsilon)$, i.e. as

$$\mathcal{L}(\lambda) = \mathbb{E}_{q_\lambda(\theta)}\left\{h_\lambda(\theta)\right\} = \mathbb{E}_{p(\epsilon)}\left[h_\lambda\{g_\lambda(\epsilon)\}\right].$$

Then the gradient of the ELBO, $\nabla_\lambda \mathcal{L}(\lambda)$ can be calculated with respect to the measure $p(\epsilon)$

$$\nabla_\lambda \mathcal{L}(\lambda) = \mathbb{E}_{p(\epsilon)}\left\{\nabla_\lambda g_\lambda(\epsilon)^\top \nabla_\theta h_\lambda(\theta)\right\} \quad (10)$$

(see Tran et al, 2021b for details). The reparametrization trick is used to express the gradient of the expectation as an expectation of a gradient. Provided that $g_\lambda$ is differentiable, then samples can be generated from $p(\varepsilon)$ to estimate $\nabla_\lambda \mathcal{L}(\lambda)$ via Monte Carlo.

When using the reparametrization trick and Gaussian variational approximations with a Cholesky factorization, the estimates of the $\mathcal{L}(\lambda)$ gradient depend on the gradient of the deterministic transformation function $\nabla_\lambda g_\lambda(\epsilon)$, as shown in (10), and are computed with respect to the variational parameters $\mu$ and $L$. The gradient of $\mathcal{L}(\lambda)$ with respect to the mean vector $\mu$ can be estimated via Monte Carlo sampling from the alternative probability measure $p(\epsilon)$ by noting that

$$\nabla_\mu \mathcal{L}(\lambda) = \mathbb{E}_{p(\epsilon)}\left\{\nabla_\theta h_\lambda(\theta)\right\}. \quad (11)$$

Let $\text{vech}(A)$ denote the half-vectorization of a matrix $A$ of size $d \times d$, i.e. the vector of length $d(d+1)/2$ obtained by stacking the elements in the matrix lower triangle, including the elements along the diagonal of $A$, from left to right and top to bottom. The gradient of the half-vectorization of $L$ under $p(\epsilon)$ is

$$\nabla_{\text{vech}(L)} \mathcal{L}(\lambda) = \mathbb{E}_{p(\epsilon)}\left[\text{vech}\left\{\nabla_\theta h_\lambda(\theta)\epsilon^T\right\}\right], \quad (12)$$

with $\theta = \mu + L\epsilon$.

# 3 Batch Stochastic Variational Bayes for GARCH Models

In this section, we derive an algorithm for fitting GARCH models via SVB (Stochastic Variational Bayes). The algorithm provides approximate posterior densities for the parameters of interest in models with the distributional assumptions on the innovations of Section 2.

We adopt the notation of Wand (2014) and Ong et al (2018a) and start by deriving the algorithm using the control variates approach first, followed by the reparametrization trick.

To define an SVB (Stochastic Variational Bayes) algorithm, it is necessary to have an unbiased estimator of the gradient vector of $\mathcal{L}(\lambda)$ with respect to the variational parameter $\lambda$, denoted as $\widehat{\nabla_\lambda \mathcal{L}(\lambda)}$.

Considering Equation (7), this estimate depends on:

$$\begin{aligned} h_\lambda(\theta) &= \log \frac{L(\theta;\, y)p(\theta)}{q_\lambda(\theta)} \\ &= \ell(\theta) + \log p(\theta) - \log q_\lambda(\theta), \end{aligned} \quad (13)$$

where $\ell(\theta)$ is the log-likelihood function, $p(\theta)$ is the prior distribution and $q_\lambda(\theta)$ is the variational distribution. It is easy to notice that $\log q_\lambda(\theta)$ and, consequently, $\nabla_\lambda \log q_\lambda(\theta)$ depend only on the variational density $q_\lambda(\theta)$, and they do not rely on the specific model under examination. Here, we take $q_\lambda(\theta)$ to be a multivariate Gaussian distribution. The simplest approximation assumes the covariance matrix $\Sigma$ in the Gaussian variational density $q_\lambda(\theta)$ has a fully independent structure, meaning it is diagonal (mean-field approximation). However, this approximation lacks the ability to capture the dependence structure of the

target posterior density and tends to underestimate the uncertainties of the model parameters (Ong et al, 2018b). We will compare results using a dense or a diagonal covariance matrix.

According to Tan and Nott (2018) and Ong et al (2018b), faster convergence in variational approximations for state-space models can be achieved by parametrizing the precision matrix instead of the covariance matrix. However, these works usually take advantage of sparse structures. In our simulations, we have observed that, for GARCH models, variational algorithms using control variates converge faster when the approximation is based on a parameterization of the precision matrix. However, the opposite holds true when using the reparametrization trick. We define the precision matrix as $\Omega = \Sigma^{-1} = CC^\top$, where $C$ is a lower triangular matrix. We believe that this behaviour, and a preference for using the covariance matrix when using the reparametrization trick, derive from the absence of a sparse structure in the models under consideration.

To implement an algorithm based on control variates, the variational parameters to be optimized are $\lambda = (\mu^\top, \text{vech}(C)^\top)^\top$, where $\mu$ is a $d$-dimensional vector, and $\text{vech}(C)$ is the half-vectorization of $C$, i.e. a vector of $d(d+1)/2$-dimensional elements obtained by vectorizing the lower triangular part of $C$. The log-density of the multivariate Gaussian distribution is

$$\begin{aligned}\log q_\lambda(\theta) = &-\frac{d}{2}\log 2\pi + \log|C| \\ &-\frac{1}{2}(\theta-\mu)^\top CC^\top(\theta-\mu)\end{aligned} \tag{14}$$

and the gradient of the log-multivariate Gaussian density with respect to $\mu$ and $\text{vech}(C)$ is

$$\nabla_\lambda \log q_\lambda(\theta) = (\nabla_\mu \log q_\lambda(\theta)^\top, \nabla_{\text{vech}\,(C)} \log q_\lambda(\theta)^\top)^\top,$$

where

$$\begin{aligned}\nabla_\mu \log q_\lambda(\theta) = &\ CC^\top(\theta-\mu) \\ \text{and } \nabla_{\text{vech}\,(C)} \log q_\lambda(\theta) = &\ \text{vech}\big(\text{diag}(1/C) \\ &-(\theta-\mu)(\theta-\mu)^\top C\big).\end{aligned}$$

Here $\text{diag}(1/C)$ denotes a diagonal matrix whose $i$-th entry is $1/C_{ii}$ and all the elements of $\mu$ and $\text{vech}\,(C)$ are not restricted. To ensure that $C$ is positive-definite, we parameterize the optimization problem in terms of the lower triangular matrix $C'$, where $C'_{ij} = C_{ij}$ if $i \neq j$ and $C'_{ii} = \exp(C_{ii})$.

While producing an unbiased estimator of the ELBO with smaller variance, the use of control variates can slow down the variational algorithm implementation (Tran et al, 2021b) and the reparametrization trick can be used as an alternative to derive faster algorithms. This approach uses a reparametrized gradient estimator that is adapted to the specific model. From (13), the generic expression for the gradient is

$$\begin{aligned}\nabla_\theta h_\lambda(\theta) &= \nabla_\theta \log \frac{L(\theta;\, y)p(\theta)}{q_\lambda(\theta)} \\ &= \nabla_\theta \ell(\theta) + \nabla_\theta \log p(\theta) - \nabla_\theta \log q_\lambda(\theta).\end{aligned} \tag{15}$$

Finally, the gradient of the log multivariate Gaussian variational density with respect to $\theta$ is

$$\nabla_\theta \log q_\lambda(\theta) = -\Sigma^{-1}(\theta-\mu).$$

## 3.1 GARCH Model with Gaussian Innovations

For simplicity, we will restrict our attention to the GARCH(1,1) model. Extensions to the general GARCH(p,q) as given in (1) are straightforward. The parameters of a GARCH(1,1) with Gaussian innovations are $\phi = (\omega, \alpha, \beta)^\top$, with $\omega > 0$, $\alpha > 0$, $\beta > 0$ and $\alpha + \beta < 1$. Tran et al (2021a) suggest the following transformations to work with unconstrained parameters: $\alpha = \psi_1\psi_2$, $\beta = \psi_1(1-\psi_2)$ with $0 < \psi_i < 1$, $i = 1, 2$, $\theta_\omega = \log(\omega)$, $\theta_{\psi_1} = \log\{\psi_1/(1-\psi_1)\}$ and $\theta_{\psi_2} = \log\{\psi_2/(1-\psi_2)\}$, therefore $\theta_\phi = (\theta_\omega, \theta_{\psi_1}, \theta_{\psi_2})^\top$, and $\theta = \theta_\phi$.

An inverse gamma prior distribution $IG(1,1)$ for $\omega$ and independent uniform prior distributions $U(0,1)$ for $\psi_1$ and $\psi_2$ are assumed. These prior distributions induces a negative log-gamma density $NLG(1,1)$ for $\theta_\omega$ and an independent standard logistic distribution for $\theta_{\psi_1}$ and $\theta_{\psi_2}$, i.e.

$$\log p(\theta_\omega) = -\theta_\omega - \exp(-\theta_\omega), \tag{16}$$

$$\log p(\theta_{\psi_1}) = -\theta_{\psi_1} - 2\log(1+\exp(-\theta_{\psi_1})) \tag{17}$$

and

$$\log p(\theta_{\psi_2}) = -\theta_{\psi_2} - 2\log(1+\exp(-\theta_{\psi_2})). \tag{18}$$

The log-likelihood function of the GARCH(1,1) model with Gaussian innovations is defined in (2) and using (13) this gives rise to

$$\begin{aligned} h_\lambda(\theta) = &-\frac{1}{2}\sum_{t=1}^{T}\left\{\log(2\pi)+\log(\sigma_t^2)+\frac{y_t^2}{\sigma_t^2}\right\} \\ &+\log p(\theta)+\frac{d}{2}\log 2\pi-\log|C| \\ &+\frac{1}{2}(\theta-\mu)^\top CC^\top(\theta-\mu). \end{aligned} \tag{19}$$

Using (14), it is then possible to implement the control variates approach.

To implement a variational algorithm using the reparametrization trick, the gradients $\nabla_\theta \ell(\theta)$ and $\nabla_\theta \log p(\theta) = \left(\nabla_{\theta_\omega}\log p(\theta_\omega), \nabla_{\theta_{\psi_1}}\log p(\theta_{\psi_1}), \nabla_{\theta_{\psi_2}}\log p(\theta_{\psi_2})\right)^\top$ for the GARCH(1,1) model with Gaussian innovations are needed. The three components of the second gradient are

$$\nabla_{\theta_\omega}\log p(\theta_\omega) = -1+\exp(-\theta_\omega), \tag{20}$$

$$\nabla_{\theta_{\psi_1}}\log p(\theta_{\psi_1}) = -1+\frac{2\exp(-\theta_{\psi_1})}{1+\exp(-\theta_{\psi_1})} \tag{21}$$

and

$$\nabla_{\theta_{\psi_2}}\log p(\theta_{\psi_2}) = -1+\frac{2\exp(-\theta_{\psi_2})}{1+\exp(-\theta_{\psi_2})}. \tag{22}$$

The gradient of the log-likelihood function for the Gaussian GARCH(1,1) model (see also Fiorentini et al, 1996) in (2) with respect to the unconstrained parameters is

$$\nabla_\theta \ell(\theta) = \frac{1}{2}\sum_{t=1}^{T}\frac{1}{\sigma_t^2}\frac{\partial\sigma_t^2}{\partial\theta}\left(\frac{y_t^2}{\sigma_t^2}-1\right) \tag{23}$$

and the partial derivatives of $\sigma_t^2$ with respect to the unconstrained parameters are given in Appendix A.1. Finally, $\nabla_\theta h_\lambda(\theta)$ defined in (15) takes the following form:

$$\begin{aligned} \nabla_\theta h_\lambda(\theta_\phi) = &\frac{1}{2}\sum_{t=1}^{T}\frac{1}{\sigma_t^2}\frac{\partial\sigma_t^2}{\partial\theta_\phi}\left(\frac{y_t^2}{\sigma_t^2}-1\right) \\ &+\nabla_\theta\log p(\theta_\phi)+\Sigma^{-1}(\theta_\phi-\mu), \end{aligned} \tag{24}$$

where $\nabla_\theta \log p(\theta_\phi)$ is stated in (20)–(22).

## 3.2 GARCH Model with t Innovations

To adapt SVB with control variates to the GARCH model with t innovations, it is necessary to derive the model-specific $\log p(\theta)$ and $\ell(\theta)$. The log-likelihood function for the t model is given in Equation (3). The prior distributions for $\omega$, $\alpha$, and $\beta$ in the time-varying variance equation for $\sigma_t^2$ remain the same as in the Gaussian case. As for the degrees of freedom parameter $\nu$, we choose a translated exponential prior density (Ardia et al, 2008).

$$p(\nu) = \exp\big\{-(\nu-2)\big\}\mathbb{1}_{\{\nu>2\}}.$$

Similarly to $\omega$, $\alpha$, and $\beta$, $\nu$ can also be mapped to an unrestricted space by setting $\nu = \log\{\exp(\theta_\nu)+1\}+2$. According to Tan and Nott (2018), this specific transformation ensures greater numerical stability compared to the transformation $\nu = \exp(\theta_\nu)+2$. The resulting log-prior density of $\theta_\nu$ is then

$$\begin{aligned} \log p(\theta_\nu) = &-\log\{\exp(\theta_\nu)+1\} \\ &-\log\{1+\exp(-\theta_\nu)\}. \end{aligned} \tag{25}$$

The parameters to be optimized are then the unconstrained ones, $\theta = (\theta_\phi, \theta_\nu)$, where $\theta_\phi = (\theta_\omega, \theta_{\psi_1}, \theta_{\psi_2})^T$, and $\phi = (\omega, \alpha, \beta)^T$.

The expression of $h_\lambda(\theta)$ defined in (13) takes the following form for the t GARCH(1,1) model:

$$\begin{aligned} h_\lambda(\theta) = T\Bigg\{ &\log\Gamma\Big(\frac{\log\{\exp(\theta_\nu)+1\}+3}{2}\Big) \\ &-\log\Gamma\Big(\frac{\log\{\exp(\theta_\nu)+1\}+2}{2}\Big) \\ &-\frac{1}{2}\log\big\{\pi(\log\{\exp(\theta_\nu)+1\})\big\}\Bigg\} \\ &-\frac{1}{2}\sum_{t=1}^{T}\Bigg[\log(\sigma_t^2)+(\log\{\exp(\theta_\nu)+1\}+3) \\ &\times\log\Bigg\{1+\frac{y_t^2}{\log\{\exp(\theta_\nu)+1\}\sigma_t^2}\Bigg\}\Bigg]+\log p(\theta) \\ &+\frac{d}{2}\log 2\pi-\log|C| \\ &+\frac{1}{2}(\theta-\mu)^\top CC^\top(\theta-\mu), \end{aligned} \tag{26}$$

where $\log p(\theta)$ is given by the sum of (16)–(18) and (25).

To implement SVB with the reparametrization trick, the gradients of the log-prior densities for the $\theta_\phi$ components remain the same as in Equations (20)–(22). Additionally, the gradient of the log-prior density for $\theta_\nu$ is given by

$$\nabla_{\theta_\nu} \log p(\theta_\nu) = \frac{\exp(-\theta_\nu)}{1+\exp(-\theta_\nu)} - \frac{\exp(\theta_\nu)}{1+\exp(\theta_\nu)}. \tag{27}$$

The derivational steps for the gradient of the log-likelihood function from the t GARCH model are provided in Appendix A.2 (also refer to Levy, 2003). Therefore, the expression for $\nabla_\theta h_\lambda(\theta)$, as defined in Equation (15) for the t GARCH(1,1) model, takes the following form:

$$\nabla_\theta h_\lambda(\theta) = \nabla_\theta \ell(\theta) + \nabla_\theta \log p(\theta) + \Sigma^{-1}(\theta-\mu), \tag{28}$$

where the expression of $\nabla_\theta \ell(\theta) = \left(\frac{\partial \ell(\theta)}{\partial \theta_\phi}, \frac{\partial \ell(\theta)}{\partial \theta_\nu}\right)^\top$ can be found in the appendix A.2 and $\nabla_\theta \log p(\theta)$ is stated in (20)–(22) and (27).

### 3.3 GARCH Model with Skewed t Innovations

The log-likelihood for the GARCH(1,1) model with standardized skewed t innovations is presented in Equation (4). Additionally, the skewness parameter $\xi$ is assumed to follow an inverse gamma $IG(1,1)$ prior distribution, according to the same reasoning in the definition of the prior distribution for $\omega$:

$$p(\xi) = \xi^{-2}\exp(-\xi^{-1})\mathbb{1}_{\{\xi>0\}}.$$

Once again, the skewness parameter $\xi$ can be transformed into an unconstrained parameter using the transformation $\xi = \log\exp(\theta_\xi)+1$. As a result, the log-prior density of $\theta_\xi$ is given by

$$\begin{aligned}\log p(\theta_\xi) = &-2\log\left(\log(\exp(\theta_\xi)+1)\right)\\ &-\log(\exp(\theta_\xi)+1)^{-1}\\ &-\log\left(1+\exp(-\theta_\xi)\right).\end{aligned} \tag{29}$$

The original set of parameters $(\phi,\nu,\xi)^\top$ with $\phi=(\omega,\alpha,\beta)^\top$ is replaced by the unrestricted parameters $\theta=(\theta_\phi,\theta_\nu,\theta_\xi)^\top$, where $\theta_\phi=(\theta_\omega,\theta_{\psi_1},\theta_{\psi_2})$. The prior densities for the constrained parameters $\omega$, $\alpha$, $\beta$, and $\nu$ remain the same as those of the t GARCH model described in Section 3.2.

The expression for $h_\lambda(\theta)$ defined in Equation (13) takes the following form for the skewed t GARCH(1,1) model:

$$\begin{aligned}h_\lambda(\theta) = T\Bigg\{ &\log\Gamma\left(\frac{\log\{\exp(\theta_\nu)+1\}+3}{2}\right)\\ &-\log\Gamma\left(\frac{\log\{\exp(\theta_\nu)+1\}+2}{2}\right)\\ &-\frac{1}{2}\log\left\{\pi(\log\{\exp(\theta_\nu)+1\})\right\}\\ &+\log\left(\frac{2}{\log\{\exp(\theta_\xi)+1\}+1/\log\{\exp(\theta_\xi)+1\}}\right)\\ &+\log(s)\Bigg\} - \frac{1}{2}\sum_{t=1}^{T}\Bigg[\log(\sigma_t^2)+(\log\{\exp(\theta_\nu)+1\}\\ &+3)\log\left\{1+\frac{(sy_t/\sigma_t+m)^2}{\log\{\exp(\theta_\nu)+1\}}\xi^{-2I_t}\right\}\Bigg]\\ &+\log p(\theta)+\frac{d}{2}\log 2\pi-\log|C|\\ &+\frac{1}{2}(\theta-\mu)^\top CC^\top(\theta-\mu),\end{aligned} \tag{30}$$

where $\log p(\theta)$ is given by the sum of (16)–(18), (25) and (29).

To implement SVB with the reparametrization trick, the gradients of the log-prior densities with respect to $\theta_\omega,\theta_\alpha,\theta_\beta,\theta_\nu$ are required. These gradients can be found in Appendix A.2. Additionally, the gradient of the log-prior density for $\theta_\xi$ is

$$\begin{aligned}\nabla_{\theta_\xi}\log p(\theta_\xi) = &-\frac{2\exp(\theta_\xi)}{\left(1+\exp(\theta_\xi)\right)\log\left(\exp(\theta_\xi)+1\right)}\\ &+\frac{\exp(\theta_\xi)}{1+\exp(\theta_\xi)}+\frac{\exp(-\theta_\xi)}{1+\exp(-\theta_\xi)}.\end{aligned} \tag{31}$$

The gradient for the log-likelihood function of the skewed t GARCH model is given in Appendix A.3. From (15),

$$\nabla_\theta h_\lambda(\theta) = \nabla_\theta \log p(y|\theta) + \nabla_\theta \log p(\theta) + \Sigma^{-1}(\theta-\mu), \tag{32}$$

where $\nabla_\theta \log p(y|\theta)$ is provided in Appendix A.3 and $\nabla_\theta \log p(\theta)$ is given by (20)–(22), (27) and (31).

Algorithms 1 and 2 outline SVB procedures for GARCH-type models with different types of innovations, namely Gaussian, t, and skewed t. These algorithms utilize the Gaussian Cholesky factorization and incorporate the use of control variates and the reparametrization trick, respectively.

# 4 Sequential Variational Inference

In the previous section, we introduced SVB approximations for GARCH models. One characteristic of time series data, such as those described by GARCH models, is that data points may become available sequentially. Ideally, inference should be updated sequentially as new information becomes available, avoiding the inefficient re-evaluation of the likelihood function for the entire dataset. In this section, we present implementations of the variational approximations from the previous section that allow for sequential updating. We adapt the strategies of Gunawan et al (2021) and Tomasetti et al (2022) for the sequential updating of variational approximations for GARCH models for the first time in the literature and compare their performance.

## 4.1 Updating Variational Bayes

In their work, Tomasetti et al (2022) propose a procedure called updating variational Bayes (UVB), which enables sequential updates of posterior distributions without the need to re-compute the likelihood function at each step.

Consider a sequence of time points $T_1, T_2, \ldots$, and suppose we want to update the posterior density $p(\theta|y_{1:T_n})$ after observing new data points in the interval $[T_{n+1}, T_{n+h}]$ to derive $p(\theta|y_{1:T_{n+h}})$. The posterior distribution computed up to time $T_n$ can then be used as the prior distribution for the subsequent observation interval. As a result, the likelihood function for the interval $[T_{n+1}, T_{n+h}]$ is based solely on $h$ data points:

$$p(\theta|y_{1:T_{n+h}}) \propto L(\theta;\ y_{T_{n+1}:T_{n+h}})p(\theta|y_{1:T_n}). \quad (33)$$

Hence, the posteriors for new observations can be updated sequentially by recursively applying Bayes' theorem.

We now define the UVB procedure for GARCH-type models. To initialize the UVB algorithm, we can employ the batch stochastic variational approach described in Section 3 and compute the first variational approximation $q_{\lambda_n}(\theta|y_{1:T_n})$ of the posterior density $p(\theta|y_{1:T_n})$. After obtaining the initial variational approximation and observing new data points in the interval $[T_{n+1}, T_{n+h}]$, we can approximate the posterior distribution defined in Equation (33) with a pseudo-posterior obtained using the variational approximation of the posterior distribution after the first $T_n$ points:

$$\tilde{p}(\theta|y_{1:T_{n+h}}) \propto L(\theta;\ y_{T_{n+1}:T_{n+h}}|\theta)q_{\lambda_n}(\theta|y_{T_1:T_n}). \quad (34)$$

Instead of minimizing the Kullback-Leibler divergence between the variational density $q_{\lambda_{n+h}}(\theta|y_{1:T_{n+h}})$ and the exact posterior density $p(\theta|y_{1:T_{n+h}})$, UVB modifies the objective function by substituting the exact posterior $p(\theta|y_{1:T_{n+h}})$ with the pseudo-posterior density $\tilde{p}(\theta|y_{1:T_{n+h}})$. For each $n = 1, 2, \ldots$, the variational parameter $\lambda_{n+h}$ is optimized by maximizing the evidence lower bound (ELBO) using a batch stochastic variational approximation. This modification enables UVB to update the posterior density sequentially and efficiently. It is worth noting that in this article, we employ the faster reparametrization trick approach for UVB, whereas the original paper only considers the control variates case, which was found to be slower than the reparametrization trick approach in our numerical experiments. Algorithm 3 outlines the steps involved in implementing the UVB approach.

## 4.2 Sequential Variational Bayes Approximation

The alternative approach proposed by Gunawan et al (2021), called sequential SVB (Seq-SVB), differs from UVB in that it targets the true posterior distribution instead of the pseudo-posterior distribution (34).

Instead of sequentially using variational approximations to the posterior distribution, Seq-SVB optimizes the variational parameters $\lambda_n$ at

**Algorithm 1** SVB for GARCH(1,1) models with Gaussian, t and skewed t innovations using a Gaussian approximating density and the control variates.

**Require:**

- initial $\lambda^{(0)} = ({\mu^{(0)}}^\top, \text{vech}(C^{(0)})^\top)^\top$;
- adaptive learning weights $\beta_1$,$\beta_2 \in (0, 1)$
- fixed learning rate $\eta_0$
- threshold $\tau$
- rolling window size $t_W$
- maximum patience parameter $P$
- prior distribution $p(\theta)$
- likelihood function $L(\theta;\ y)$
- Monte Carlo Samples $S$.

**Initialization:**

- Set $C'^{(0)} = C^{(0)}$.
- Update $\text{diag}(C'^{(0)}) = \exp\big(\text{diag}(C'^{(0)})\big)$.
- Generate $\theta_{(s)} \sim \mathcal{N}\big(\mu^{(0)}, (C'^{(0)}{C'^{(0)}}^\top)^{-1}\big)$.
- Compute $\widehat{\nabla_\lambda \mathcal{L}(\lambda^{(0)})}$ in (7) where $h_\lambda(\theta)$ is given in (19) for Gaussian, (26) for t, and (30) for skewed t innovations.
- Calculate the $j$-th element of the control variate $c_j$ in (9) using $\theta_s$.
- Set $g_0 := \widehat{\nabla_\lambda \mathcal{L}(\lambda^{(0)})}, v_0 := (g_0)^2, \bar{g} := g_0, \bar{v} := v_0$, $t = 0$, patience $= 0$.

**End Initialization**

**while stop = FALSE do**

- Set $C'^{(t)} = C^{(t)}$.
- Update $\text{diag}(C'^{(t)}) = \exp\big(\text{diag}(C'^{(t)})\big)$.
- Generate $\theta_s \sim \mathcal{N}\big(\mu^{(t)}, (C'^{(t)}{C'^{(t)}}^\top)^{-1}\big)$.
- Compute the unbiased estimator $g_t := \nabla_\lambda \widehat{\mathcal{L}(\lambda^{(t)})}$.
- Estimate the new $j$-th entry of the control variate $c_j$ in (9).
- Calculate the coordinate-wise operator $v_t = (g_t)^2$ and set $\bar{g} = \beta_1 \bar{g} + (1 - \beta_1) g_t$, $\bar{v} = \beta_2 \bar{v} + (1 - \beta_2) v_t$.
- Compute the learning rate $\alpha_t = \min(\eta_0, \eta_0 \frac{\tau}{t})$.
- Update $\lambda^{(t+1)} = \lambda^{(t)} + \alpha_t \frac{\bar{g}}{\sqrt{\bar{v}}}$.
- Compute the ELBO estimator $\widehat{\mathcal{L}(\lambda^{(t)})}$ in (5) using Monte Carlo samples $\theta_s$.

**if** $t \geq t_W$ **then**
  Compute the moving average lower bound $\overline{\mathcal{L}}_{t-t_W+1}(\lambda) = \frac{1}{t_W} \sum_{k=1}^{t_W} \widehat{\mathcal{L}(\lambda^{(t-k+1)})}$,
  **and if** $\overline{\mathcal{L}}_{t-t_W+1}(\lambda) \geq \max(\overline{\mathcal{L}}(\lambda))$ **then** patience $= 0$,
  **otherwise** patience = patience + 1.
**end if**
**if** patience $\geq P$ **then stop = TRUE**.
**else**
  Set t = t+1.
**end if**

**end while**

**Algorithm 2** SVB for GARCH(1,1) models with Gaussian, t and skewed t innovations using a Gaussian approximating density and the reparametrization trick.

**Require:**

- initial $\lambda^{(0)} = (\mu^{(0)^\top}, \text{vech}(C^{(0)})^\top)^\top$;
- adaptive learning weights $\beta_1$,$\beta_2 \in (0, 1)$
- fixed learning rate $\eta_0$
- threshold $\tau$
- rolling window size $t_W$
- maximum patience parameter $P$
- prior distribution $p(\theta)$
- likelihood function $L(\theta;\ y)$
- gradients of log-likelihood function $\nabla \log L(\theta;\ y)$
- Monte Carlo Samples $S$.

**Initialization:**

- Generate $\epsilon_s \sim p_\epsilon(.) = \mathcal{N}_d(0, I), \quad s = 1, \ldots, S$
- Compute the unbiased gradient estimator of the ELBO in (11) and (12) where $\nabla_\theta h_\lambda(\theta)$ are stated in (24), (28) and (32) for Gaussian, t and skewed t innovations respectively, $\nabla_\lambda \widehat{\mathcal{L}(\lambda^{(0)})} = \left(\nabla_\mu \widehat{\mathcal{L}(\lambda^{(0)})^T}, \nabla_{\text{vech}(L)} \widehat{\mathcal{L}(\lambda^{(0)})^T}\right)^T$, with Monte Carlo samples $\theta_s = \mu^{(0)} + L^{(0)}\epsilon_s$.
- Set $g_0 := \nabla_\lambda \widehat{\mathcal{L}(\lambda^{(0)})}, v_0 := (g_0)^2, \bar{g} := g_0, \bar{v} := v_0$, $t = 0$, patience $= 0$ and **stop = FALSE**.

**End Initialization**

**while stop = FALSE do**

- Generate $\epsilon_s \sim p_\epsilon(.) = \mathcal{N}_d(0, I), s = 1, \ldots, S$.
- Recalculate $\mu^{(t)}, L^{(t)}$ from the updated $\lambda^{(t)}$.
- Update the unbiased gradient estimator $\nabla_\lambda \widehat{\mathcal{L}(\lambda^{(t)})} = \left(\nabla_\mu \widehat{\mathcal{L}(\lambda^{(t)})^T}, \nabla_{\text{vech}(L)} \widehat{\mathcal{L}(\lambda^{(t)})^T}\right)^T$ in (11) and (12) where $\nabla_\theta h_\lambda(\theta)$ are stated in (24), (28) and (32) for Gaussian, t and skewed t innovations respectively, with new Monte Carlo samples $\theta_s = \mu^{(t)} + L^{(t)}\epsilon_s$.
- Calculate the coordinate-wise operator $v_t = (g_t)^2$ and set $\bar{g} = \beta_1 \bar{g} + (1 - \beta_1) g_t$, $\bar{v} = \beta_2 \bar{v} + (1 - \beta_2) v_t$.
- Compute the learning rate $\alpha_t = \min(\eta_0, \eta_0 \frac{\tau}{t})$ and update

$$\lambda^{(t+1)} = \lambda^{(t)} + \alpha_t \frac{\bar{g}}{\sqrt{\bar{v}}}.$$

- Compute the ELBO estimator $\widehat{\mathcal{L}(\lambda^{(t)})}$ in (5) using Monte Carlo samples $\theta_s$.

**if** $t \geq t_W$ **then**
Compute the moving average lower bound $\overline{\mathcal{L}}_{t-t_W+1}(\lambda) = \frac{1}{t_W} \sum_{k=1}^{t_W} \mathcal{L}(\widehat{\lambda^{(t-k+1)}})$,
**and if** $\overline{\mathcal{L}}_{t-t_W+1}(\lambda) \geq \max(\overline{\mathcal{L}}(\lambda))$ **then** patience $= 0$,
**otherwise** patience $=$ patience $+ 1$.
**end if**
**if** patience $\geq P$ **then stop = TRUE**.
**else**
Set t = t+1.
**end if**

**end while**

**Algorithm 3** Updating Variational Bayes

**Require:** Prior, likelihood

**Initialization:**
Observe $y_{T_1:T_n}$.
Compute the optimized variational density $q_{\lambda_n}(\theta|y_{T_1:T_n})$ using SVB via Algorithm 1 or 2.
**End Initialization**

**for** $j = 1, \ldots, h$ **do**
Collect new data points $y_{T_{n+j-1}+1:T_{n+j}}$.
Use $q_{\lambda_{n+j-1}}$ and (34) to construct the UVB pseudo-posterior up $\tilde{p}(\theta|y_{T_1:T_{n+j}})$ up to proportionality.
Minimize $\mathrm{KL}\{q_{\lambda_{n+j}}(\theta|y_{T_1:T_{n+j}})\|\tilde{p}(\theta|y_{T_1:T_{n+j}})\}$ using SVB via Algorithm 1 or 2.
**end for**

**Algorithm 4** Sequential Stochastic Variational Bayes

**Require:** Prior, likelihood

**Initialization:**
Observe $y_{T_1:T_n}$.
Compute $q_{\lambda_n}(\theta|y_{T_1:T_n})$ using SVB Algorithm 1 or 2.
**End Initialization**

**for** $j = 1, 2, \ldots, h$ **do**
Define the new vector of observations $y_{T_1:T_{n+j}}$.
Initialise SVB using $\lambda_{n+j-1}$
Compute $q_{\lambda_{n+j}}(\theta|y_{T_1:T_{n+j}})$ using SVB via Algorithm 1 or 2.
**end for**

time $T_n$ and employs them as initial values for the stochastic variational approximations of the variational parameter $\lambda_{n+1}$ at time $T_{n+1}$. Seq-SVB achieves greater accuracy than UVB as it directly targets the true posterior distribution. However, it comes at a higher computational cost since it requires evaluating the likelihood function at each time in the interval $[T_1, T_{n+1}]$.

Now, we define the Seq-SVB algorithm for GARCH models. In the initialization step, batch SVB, as outlined in Algorithms 1 and 2, is utilized for the data up to time $T_n$ to obtain the optimal variational parameters $\lambda$ up to time $T_n$. Subsequently, the first variational approximation $q_{\lambda_n}(\theta|y_{T_1:T_n})$ of the true posterior $p(\theta|y_{T_1:T_n})$ is obtained. After observing new data points up to time $T_{n+1}$, another SVB algorithm can be implemented using all the available data in the interval $[T_1, T_{n+1}]$, with the optimal $\lambda_n$ from the initialization serving as the initial value for the variational parameter. This procedure is repeated for the instances $(n+1), (n+2), \ldots$. The implementation of Seq-SVB is summarized in Algorithm 4.

Note that the work by Gunawan et al (2021) addresses volatility models, with a specific focus on multivariate factor volatility models framed in a state space form. The setting of this work is simplified, because the parameter dimension does not depend on the series length and we focus on univariate time-series.

# 5 Simulation Study

## 5.1 Batch Stochastic Variational Bayes

We first assess the accuracy of the batch SVB approximation discussed in Section 3 by applying Algorithms 1 and 2 to GARCH models with the innovation types under examination. All computations were performed in `R` (R Core Team, 2023) on the Katana computational cluster with 8-CPU cores and 4GB memory per core of the University of New South Wales (https://docs.restech.unsw.edu.au/using_katana/about_katana/).

We generated time series of lengths 1,000 and 5,000 from GARCH(1,1) models with Gaussian, t, and skewed t innovations using the `R` package `fGarch` (Wuertz et al, 2022). For each setting, we simulated 1,000 replicates. The GARCH models were generated using parameter values similar to those estimated from the S&P500 return index time series analyzed in Section 6 with MCMC. Specifically, we used $\omega = 0.1$, $\alpha = 0.2$, and $\beta = 0.75$ for Gaussian, t, and skewed t innovations. We set $\nu = 4$ for the t and skewed t degrees of freedom, and $\xi = 0.8$ as the skewed t asymmetry parameter.

The algorithm parameters of Algorithms 1 and 2 were set to the default values employed in Tran et al (2021b), with some slight adjustments made to improve convergence. Specifically, the maximum patience parameter was set to $P = 100$, and the moving average window size was set to $t_W = 25$. Both adaptive learning weights were set

to $\beta_1 = \beta_2 = 0.9$, and the adaptive learning rate was set to $\eta_0 = 0.02$. We used $\tau = 1,000$ as the threshold. In order to investigate the impact of the number of Monte Carlo samples on the efficiency and computational time, we conducted experiments using different values of $S$ for both the control variates and reparametrization methods. Specifically, we fixed the number of MC samples at $S = 5, 10$, and 50 for both methods. By comparing the results obtained with these different values of $S$, we can assess the influence of the number of MC samples on the estimation accuracy and computational efficiency of the two approaches. In this paper, we focus on presenting the results obtained using the control variates method with $S = 10$, and the reparametrization trick method with $S = 5$. These particular choices for the number of MC samples were selected based on similar computational time, ensuring a fair comparison between the two approaches (See table 2). The additional results obtained using different values of $S$ can be found in the Appendix B, providing a comprehensive analysis of the sensitivity of the results to the choice of MC sample size.

Figure 1 shows the ELBO values with respect to the number of iterations in implementations based on control variates and the reparametrization trick, using both the Cholesky factorization of the precision matrix $\Omega$ and the covariance matrix $\Sigma$. Interestingly, convergence appears to be faster when using the precision matrix with control variates, and when using the covariance matrix with the reparametrization trick. In some cases SVB failed to reach convergence when using the covariance matrix with control variates.

The accuracy of the approximation is analyzed by comparing it to the posterior distribution obtained through MCMC with 1,000,000 iterations, after discarding the first 100,000 samples as burn-in. MCMC was implemented using random walk Metropolis-Hastings with Gaussian proposal distributions. The mean and variance of the proposal distributions were set to the maximum likelihood estimates, following Chan and Grant (2016). For comparison with the MCMC target, we computed the variational approximation accuracy with respect to the posterior density as

$$\text{accuracy} \equiv 100 \left(1 - \frac{1}{2} \int_{-\infty}^{\infty} \left|q_\lambda(\theta) - p(\theta|y)\right| d\theta\right) \%,$$

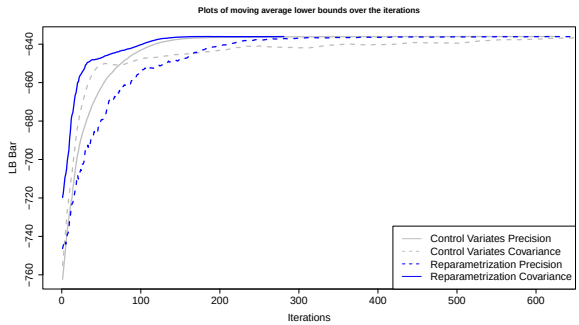


**Fig. 1**: Moving average lower bounds over iterations for the control variates and reparametrization trick methods with different Cholesky factorizations (i.e., precision matrix or covariance matrix). Results refer to a single replicate.

with 100% indicating perfect matching between the variational density $q_\lambda(\theta)$ and the posterior distribution $p(\theta|y)$ obtained via MCMC.

Figure 2 compares the accuracy of the variational approximations obtained when using SVB (on the left) and mean-field SVB (on the right) with control variates for time series of length 1,000 from a GARCH(1,1) model with Gaussian innovations. As expected, the mean-field SVB approach, which uses a fully diagonal covariance matrix for the variational approximating density, results in significantly lower accuracy for all the parameters. On the other hand, the implementation based on a full covariance matrix allows to achieve high accuracy compared to the MCMC target.

Figure 3 presents a comparison of the approximated posterior distributions obtained using SVB with a diagonal (mean-field) and non-diagonal covariance matrix, with control variates, or with the reparametrization trick and a non-diagonal covariance matrix, along with the MCMC posterior. On the right, the ELBO is illustrated as a function of the number of iterations. The results are based on time series of length 1,000 with Gaussian innovations. Once again, it is evident that the mean-field SVB approach tends to underestimate the posterior uncertainty. On the other hand, both control variates and the reparametrization trick produce highly accurate approximations. Control variates and the reparametrization trick lead to a higher average ELBO ($-781.57$ for the control variates approach and $-781.56$ for the reparametrization trick implementation) than

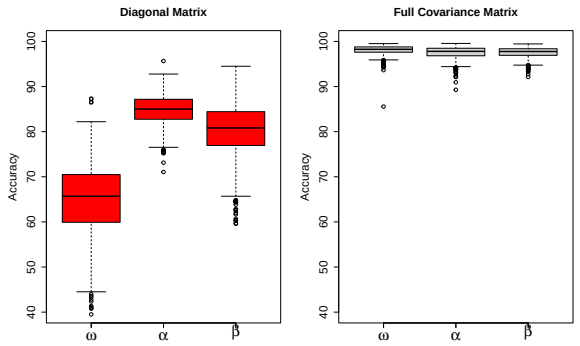


**Fig. 2**: Boxplots of the accuracy of the SVB approximate posterior densities implemented with control variates for the parameters of the GARCH model with Gaussian innovations. The boxplots for SVB using a full covariance matrix of the Gaussian approximating density are shown in the left panel; boxplots for SVB using a diagonal covariance matrix of the Gaussian approximating density are shown in the right panel.

the mean-field approximation (−782.26), and the reparametrization trick algorithm achieved faster convergence in general.

Figure 4 shows the efficiency of our SVB method in approximating the posterior densities of the parameters for a randomly selected T GARCH-type dataset. Both the control variates and reparametrization trick approaches exhibit an impressive level of agreement with the MCMC approximation, which utilized one million Monte Carlo samples. The overlapping of the posterior densities suggests that our SVB methods provide accurate and reliable estimates of the model parameters.

Figure 5 illustrates the approximate posterior densities for SVB applied to one replicate of the simulation study generated from the GARCH model with skewed t innovations. The figure also includes a plot showing the SVB lower bound as a function of the number of iterations for both the control variates and reparametrization trick implementations. All approximations appear to provide accurate results. The convergence rate suggests that the reparametrization trick reaches convergence faster compared to the algorithm using control variates.

Figure B1 - B3 (in appendix B) present boxplots comparing the accuracy results of SVB algorithms for a GARCH(1,1) model with different innovations. For shorter time series, we did not observe a clear advantage in terms of accuracy between the control variates and reparametrization trick approaches. However, as the length of the time series increases, the reparametrization trick method consistently outperforms the control variates method in terms of accuracy. This is evident from the boxplots, where the accuracy results obtained using the control variates method are consistently lower and exhibit a wider spread compared to the reparametrization trick method.

Table 1 provides a summary of the average accuracy levels of the posterior approximations for all the parameters in the three considered models. The table also includes a comparison between time series with 1,000 observations and the longer ones with 5,000 observations. It is evident that the mean-field approximation underestimates the uncertainty of the posterior distribution and exhibits the lowest accuracy. Across all datasets, the accuracy is consistently below 89%, with $\omega$ being the most challenging parameter to estimate. This issue persists across datasets of different lengths.

On the other hand, both the control variates and reparametrization trick methods achieve higher accuracy results when using an unrestricted covariance matrix. For both methods, the accuracy results for most parameters are consistently above 88% across different time series lengths except for the skewness parameter $\xi$ in the longer time series with the reparametrization trick approach. Interestingly, the performance of the control variates and reparametrization trick methods varies depending on the length of the time series. The control variates approach demonstrates higher accuracy for shorter time series, while the reparametrization trick method performs better for longer time series. This finding suggests that the choice of method may depend on the specific characteristics and length of the time series under consideration.

Table 2 presents the computational running times of all the considered implementations, including a comparison with standard MCMC. The MCMC was stopped after 50,000 iterations, as suggested in Tan and Nott (2018) and Ong et al (2018b), and convergence was also checked

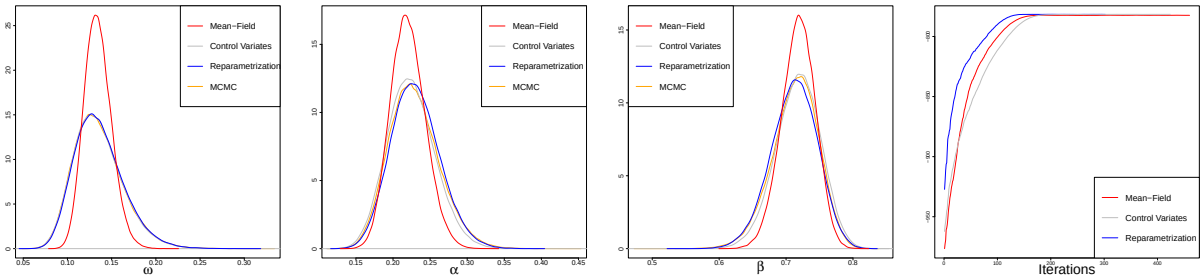


**Fig. 3**: Posteriors estimations using mean-field VB (with control variates), SVB (with control variates), and SVB (with reparametrization trick) compared with the posterior distributions obtained via MCMC. The last plot on the right shows the moving average ELBO over iterations for a randomly selected simulated Gaussian GARCH-type dataset.

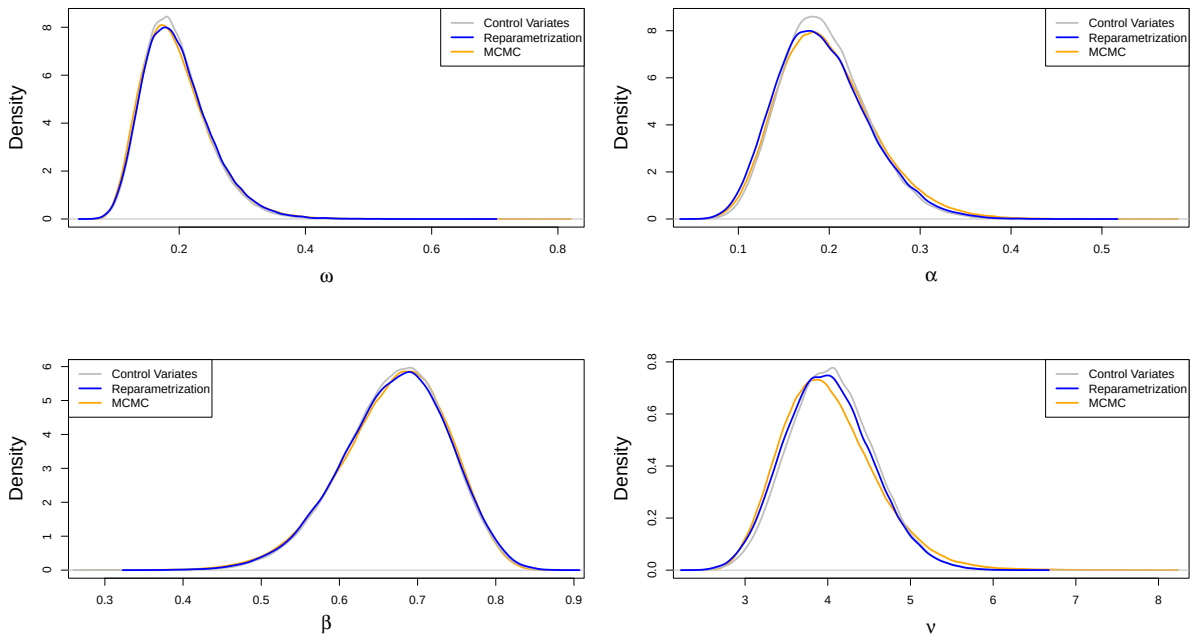


**Fig. 4**: Approximated posterior densities from the SVB methods and MCMC estimates, as well as the moving average ELBO values over iterations, for a randomly selected t GARCH-type dataset.

using the Gelman-Rubin diagnostics and autocorrelations (Brooks and Gelman, 1998).

The table presents the average computational times for various approximation methods applied to 1,000 datasets with different numbers of Monte Carlo samples $S$. It is observed that the computational time increases as the number of MC samples increases. Comparing the same number of MC samples, the control variates approach exhibits faster computational times than the reparametrization trick method. However, it is important to note that the accuracy results of the control variates approach may not meet the same standards, particularly for longer time series. By selecting MC samples of $S = 10$ for the control variates approach and $S = 5$ for the

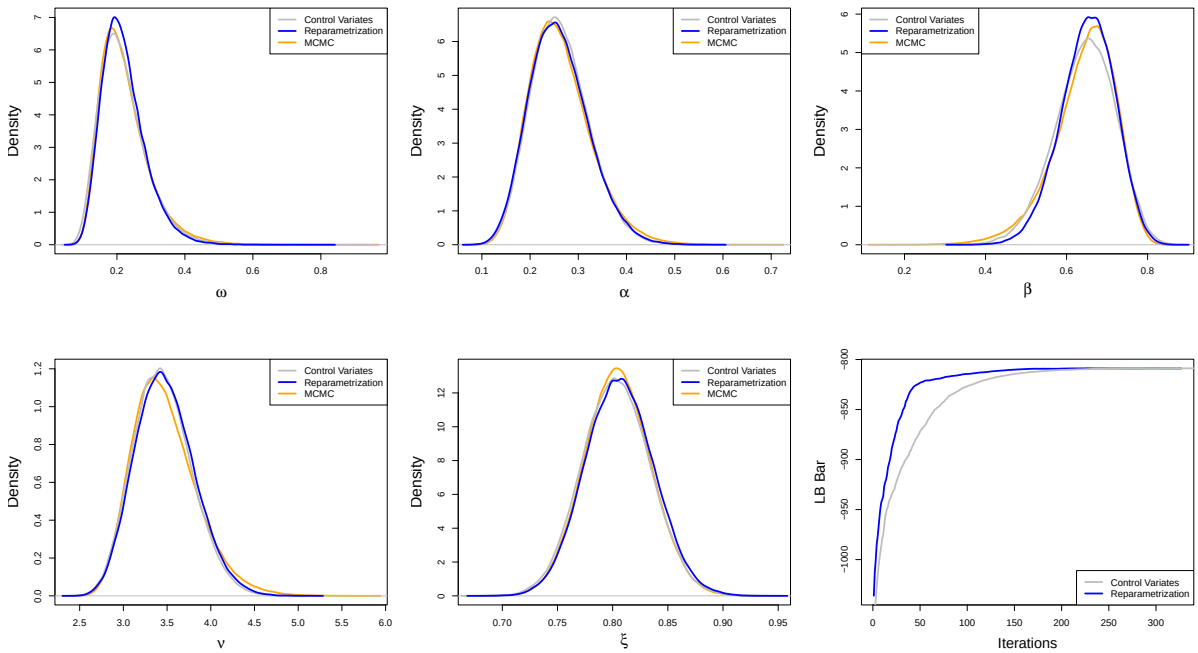


**Fig. 5**: Approximated posterior densities from the SVB methods and MCMC estimates, as well as the moving average ELBO values over iterations, for a randomly selected skewed t GARCH-type dataset.

**Table 1**: Average accuracy for the posterior distributions of each parameter of different GARCH models and methods with 1000 and 5000 observations across 1000 simulated datasets.

| | 1000 observations | | | | | 5000 observations | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\omega$ | $\alpha$ | $\beta$ | $\nu$ | $\xi$ | $\omega$ | $\alpha$ | $\beta$ | $\nu$ | $\xi$ |
| Gaussian GARCH MF | 66.19 | 88.08 | 85.75 | | | 58.81 | 84.24 | 78.30 | | |
| Gaussian GARCH CV | 96.62 | 96.35 | 96.52 | | | 91.04 | 94.08 | 92.99 | | |
| Gaussian GARCH RT | 95.93 | 94.76 | 95.00 | | | 95.35 | 94.64 | 94.61 | | |
| t GARCH CV | 95.57 | 95.12 | 95.59 | 92.23 | | 90.88 | 91.79 | 93.05 | 94.07 | |
| t GARCH RT | 95.92 | 93.97 | 94.65 | 91.81 | | 95.64 | 94.35 | 94.79 | 94.42 | |
| Skewed t GARCH CV | 95.20 | 94.42 | 95.40 | 91.22 | 93.04 | 88.80 | 89.49 | 91.69 | 92.78 | 88.70 |
| Skewed t GARCH RT | 95.73 | 94.19 | 94.87 | 90.35 | 91.32 | 94.41 | 93.65 | 93.79 | 92.99 | 82.34 |

**Table 2**: Average computational running time (in seconds) of SVB algorithms for various GARCH models, with time series of length 1,000 and 5,000 across 1000 simulated datasets.

| | Gaussian GARCH | | t GARCH | | Skewed t GARCH | |
|---|---|---|---|---|---|---|
| | 1000 obs | 5000 obs | 1000 obs | 5000 obs | 1000 obs | 5000 obs |
| CV $S = 5$ | 2.83 | 6.08 | 3.23 | 7.13 | 5.93 | 17.34 |
| CV $S = 10$ | 4.40 | 11.90 | 4.75 | 13.03 | 9.13 | 31.10 |
| CV $S = 50$ | 17.35 | 60.70 | 17.95 | 64.44 | 35.46 | 167.56 |
| RT $S = 5$ | 4.26 | 16.93 | 4.69 | 17.81 | 8.42 | 32.34 |
| RT $S = 10$ | 8.03 | 33.26 | 8.29 | 35.11 | 15.32 | 63.66 |
| RT $S = 50$ | 38.78 | 165.92 | 39.55 | 175.37 | 74.46 | 324.71 |
| MCMC | 23.66 | 99.28 | 26.47 | 109.39 | 75.23 | 326.98 |

reparametrization trick approach, we are able to achieve reasonable accuracy results while significantly reducing the computational time compared to the traditional MCMC sampling approximation method.

For time series with 1,000 observations, SVB with control variates is 4.37, 4.57, and 7.24 times faster than the traditional MCMC method for Gaussian, t, and skewed t GARCH models, respectively. The reparametrization trick method performs even faster, precisely 4.55, 4.64, and 7.93 times faster than MCMC. For the longer time series with 5,000 observations, SVB with control variates is 7.34, 7.40, and 9.51 times faster than MCMC in terms of average computational time, while SVB with the reparametrization trick is 4.86, 5.14, and 9.11 times faster for Gaussian, t, and skewed t GARCH models, respectively.

While offering a notable reduction in computational time compared to traditional MCMC methods, our SVB methods maintain reasonable accuracy in approximating the posterior densities in GARCH-type models. This combination of accuracy and computational speed makes SVB an attractive alternative for analyzing GARCH models in various applications.

## 5.2 Sequential Variational Bayes

We now investigate the performance of the sequential alternatives to SVB described in Section 4 and compare the accuracy and computational times of UVB and Seq-SVB. We generate 1,000 datasets from the GARCH model with skewed t innovations and 1,000 observations and an equal number of replicates from the same model but with 5,000 observations. We perform this comparison focusing on the skewed t case, which comprises both Gaussian and t GARCH models as special cases.

For the sequential VB method, we exclusively employ the reparametrization trick approach with a full covariance matrix. This choice is motivated by the fact that the mean-field batch version offers significantly lower accuracy compared to the unrestricted case, as shown in the previous simulated experiments. The decision to choose the reparametrization trick approach over the control variates method for our sequential variational Bayes implementation was based on the availability of prior discussions and implementations as the control variates method has already been extensively discussed and implemented in Tomasetti et al (2022). Furthermore, the reparametrization trick has not been explored for UVB in previous work. Therefore, to expand the scope of our study and explore alternative approaches, we opted to implement the reparametrization trick for the sequential version.

For the time series with 1,000 observations, i.e., $T_N = 1,000$, we choose ten different starting points $T_n = 500, 550, 600, \ldots, 950$ and perform sequential updates every $(T_N - T_n)/c$ observations, with $c = 1, 2, 5, 10$. For example, if $T_N = 1,000$, $T_n = 600$ and $c = 2$, the posterior distribution is first approximated using batch SVB on the first 600 observations, and then the posterior distribution is updated twice as new data in blocks of 200 observations arrive (see Figure 6).

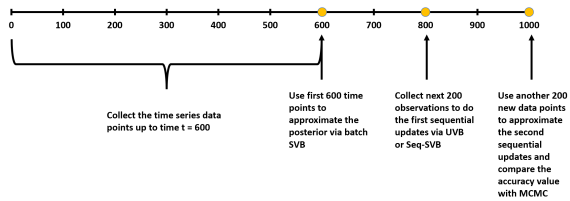


**Fig. 6**: Demonstration of sequential SVB method, i.e., $T_n = 600$ and $c = 2$, the posterior is estimated for the first 600 time points via batch SVB, and the posteriors are updated twice as new observations arrive up to $T_N = 1000$.

Figure 7 shows the average accuracy obtained across 1,000 simulated skewed t GARCH datasets, each containing 1,000 observations. The accuracy is measured for different starting values and various numbers of sequential updates. Seq-SVB methods consistently outperform UVB in terms of accuracy. This can be attributed to the fact that Seq-SVB directly targets the true posteriors instead of aiming to approximate the pseudo-posterior distributions. Remarkably, Seq-SVB methods exhibit nearly identical accuracy regardless of the number of updates or the choice of starting points. This result is expected, as the only difference among Seq-SVB approaches is the initial value of the variational parameters for the SVB algorithm, which ultimately should converge to the same value. On the other hand, UVB

approaches produce accuracy results that depend on the number of observations in the updates. As the number of sequential updates increases, the accuracy tends to decrease since the posterior approximation at each update is based on fewer observations, and each sequential update introduces a slight deviation from the true posterior. If the initial SVB approximation is based on more observations, the overall approximation is more accurate.

Figure 8 shows the average computational time with different starting points for a single sequential update. Both UVB and Seq-SVB outperform batch SVB in terms of computational speed. Seq-SVB exhibits a small downward slope, indicating a slight decrease in computation time as the number of new data points decreases. UVB, on the other hand, shows a more noticeable downward slope, indicating a more significant reduction in computational time with fewer newly arrived data points.

In Seq-SVB, the initial values used in the optimization algorithm can have an impact on convergence and computational time. When fewer new data points are available, the initial values might already be relatively close to the final converged values at time $T_N$. As a result, the optimization algorithm requires fewer iterations to reach convergence, leading to a slight decrease in computational time.

Nevertheless, the significant reduction in computation time observed in the UVB approach can be attributed to the lower number of observations that need to be recomputed in the log-likelihood calculations. UVB only needs to incorporate the likelihood of the newly arrived data points rather than the entire dataset, resulting in faster computations.

Table 3 presents the average computational time for a single update using both UVB and Seq-SVB with 1,000 and 5,000 observations. Regardless of the time series length, the trends remain consistent. UVB consistently outperforms Seq-SVB in terms of computational speed per update, and the computational gains are more significant when there are fewer newly arrived data points.

This advantage of UVB becomes more evident with longer time series lengths as Seq-SVB requires processing the entire data series for each update, while UVB only needs to incorporate the incremental time series.

We conclude this section by noticing that the parameter dimension in the models under consideration is limited. If the dimension of the parameter increases, for example because it depends on the series length or because of the presence of a latent structure, we anticipate that UVB will lead to increased computational saving with respect to Seq-SVB, since UVB evaluates a more costly likelihood function based on less observations for each update; on the other hand, UVB will likely be associated to increased accuracy loss, since at every update there will be an increased approximation error and this will propagate at each update.

# 6 Standard & Poor Data

We apply our batch and sequential variational Bayes methods to a real financial stock return dataset, specifically a time series of the Standard & Poor 500 (S&P500) stock index. After performing the data clearance procedure, which involved removing any missing or incomplete data, as well as calculating the log-returns of the time series, we obtained a dataset comprising 1,000 observations. These observations were recorded over a period spanning from 2nd January 2015 to 4th January 2019. The log-returns of the time series exhibit a range between -4.18% and 5.69%, with a median log return of 0.03%. The distribution of the log returns is negatively skewed, as evidenced by a skewness value of -0.33. Additionally, the kurtosis of the distribution is higher than that of a normal distribution, indicating excess kurtosis, with a kurtosis value of 4.59. We implemented the SVB approaches using Gaussian, t, and skewed t models on the provided dataset. Model selection criteria, namely the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC), were used to assess the performance of the models. The AIC and BIC values for each model are reported in Table 4. (Notice that these values were calculated using maximum likelihood estimates). The skewed t GARCH model returns the lowest values for both AIC and BIC. With an AIC value of 2165.44 and a BIC value of 2189.98, the skewed t GARCH model is selected as the best-fitting model for the dataset.

Figure 9 showcases the estimated marginal posterior density plots for the parameters $\omega$, $\alpha$, $\beta$, $\nu$ and $\xi$ of the skewed t GARCH model,

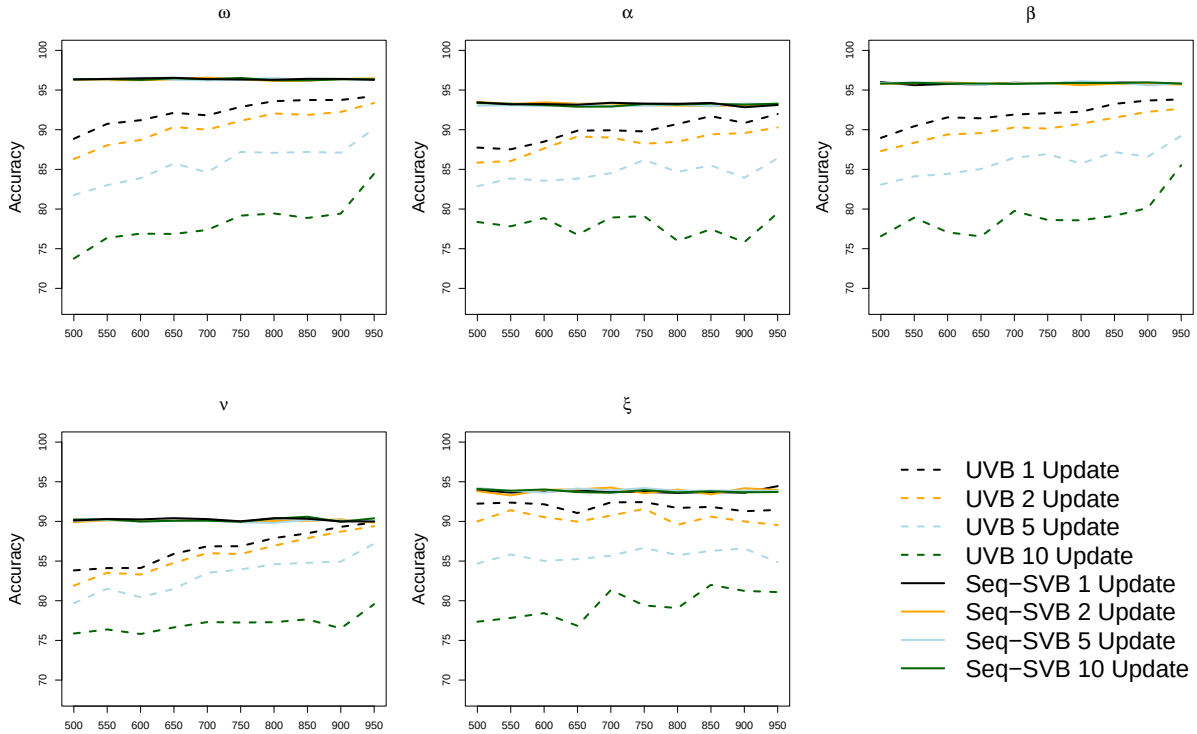


**Fig. 7**: Average accuracy over 1,000 simulated skewed t GARCH datasets with 1,000 observations for different starting values and different numbers of sequential updates.

| | $t = 500$ | $t = 550$ | $t = 600$ | $t = 650$ | $t = 700$ | $t = 750$ | $t = 800$ | $t = 850$ | $t = 900$ | $t = 950$ |
|---|---|---|---|---|---|---|---|---|---|---|
| UVB 1000 observations | 13.32 | 13.02 | 12.17 | 11.11 | 10.11 | 9.41 | 8.17 | 7.71 | 7.15 | 6.27 |
| Seq-SVB 1000 observations | 20.14 | 19.93 | 19.63 | 19.19 | 18.91 | 19.04 | 19.65 | 18.35 | 18.78 | 18.65 |

| | $t = 2500$ | $t = 2750$ | $t = 3000$ | $t = 3250$ | $t = 3500$ | $t = 3750$ | $t = 4000$ | $t = 4250$ | $t = 4500$ | $t = 4750$ |
|---|---|---|---|---|---|---|---|---|---|---|
| UVB 5000 observations | 46.16 | 42.81 | 38.33 | 33.00 | 30.14 | 26.30 | 22.59 | 17.27 | 13.81 | 9.06 |
| Seq-SVB 5000 observations | 80.31 | 84.80 | 83.74 | 83.16 | 83.36 | 78.27 | 85.03 | 79.21 | 81.60 | 78.99 |

**Table 3**: Average computational time for one single update will decrease as the number of new observations decreases. UVB is caused by the greater number of the likelihood function computed. Seq-SVB is due to worse starting values for SVB.

| Model | AIC | BIC |
|---|---|---|
| Gaussian GARCH | 2262.46 | 2277.18 |
| t GARCH | 2170.88 | 2190.51 |
| Skewed t GARCH | 2165.44 | 2189.98 |

**Table 4**: AIC and BIC values for GARCH models

using MCMC and batch SVB with control variates and the reparametrization trick. The density plots obtained from both the control variates method and the reparametrization trick closely resemble the estimates obtained from the MCMC method, indicating a high level of accuracy in approximating the posterior densities.

The accuracy results for the control variates method are 97.56% for $\omega$, 95.86% for $\alpha$, 97.42% for $\beta$, 93.23% for $\nu$ and 93.44% for $\xi$. The reparametrization trick implementation yields similar accuracy results: 94.06% for $\omega$, 98.13% for $\alpha$, 95.42% for $\beta$, 90.49% for $\nu$ and 92.80% for $\xi$. These accuracy percentages demonstrate the effectiveness of both implementations in

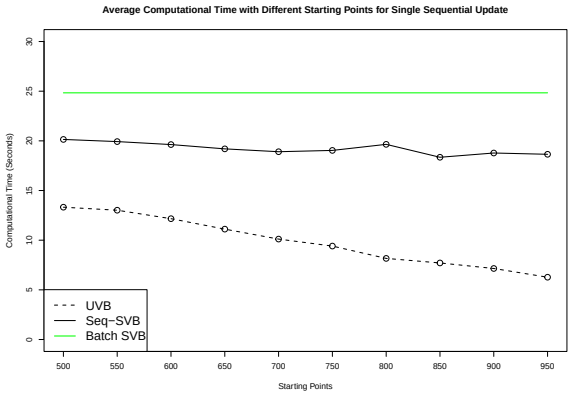


**Fig. 8**: Both UVB and Seq-SVB outperform batch SVB in terms of computational speed. Seq-SVB shows a small downward slope and UVB exhibits a more noticeable downward slope indicating different computational gain.

providing reasonable approximations of the posterior densities from the skewed t GARCH model when applied to a real S&P500 dataset.

Figure 10 illustrates the accuracy results obtained from applying the skewed t GARCH model to the real S&P 500 dataset using the Seq-SVB and UVB methods. Different starting values and varying numbers of sequential updates are considered.

Consistent with previous findings, the Seq-SVB approaches consistently outperform the UVB methods in terms of accuracy. The accuracy results in the UVB methods are significantly influenced by the choice of starting values and the number of updates. On the other hand, the Seq-SVB approach demonstrates robust accuracy regardless of the starting values and the number of updates.

The upward slope observed in the plot indicates that using fewer newly arrived data points leads to more accurate approximations. This suggests that incorporating a smaller portion of the newly observed data in each update results in improved accuracy. The Seq-SVB approach benefits from this property, as it only updates the variational parameters with a subset of the newly arrived data, leading to more accurate posterior approximations.

These results provide further evidence that the Seq-SVB methods are well-suited for sequential inference tasks, such as analyzing real financial datasets. They offer accurate approximations of the posterior distributions, and the accuracy is relatively insensitive to the choice of starting values and the number of updates.

| | SVB | 1 update | 2 updates | 5 updates | 10 updates |
|---|---|---|---|---|---|
| $\omega$ | 96.06 | 88.82 | 81.29 | 72.91 | 67.33 |
| $\alpha$ | 99.18 | 92.39 | 92.29 | 79.19 | 69.45 |
| $\beta$ | 97.80 | 91.51 | 89.39 | 79.18 | 71.52 |
| $\nu$ | 90.60 | 82.96 | 72.12 | 70.66 | 71.51 |
| $\xi$ | 96.26 | 94.98 | 92.48 | 87.45 | 81.35 |

**Table 5**: Comparison of the average accuracy results for the skewed t GARCH model parameters using batch SVB and UVB with ten different starting values, $T_n = \{500, 550, 600, \ldots, 950\}$.

| | SVB | 1 update | 2 updates | 5 updates | 10 updates |
|---|---|---|---|---|---|
| $\omega$ | 96.06 | 97.50 | 97.49 | 96.40 | 97.40 |
| $\alpha$ | 99.18 | 96.37 | 96.20 | 95.00 | 96.17 |
| $\beta$ | 97.80 | 96.56 | 97.29 | 96.25 | 97.05 |
| $\nu$ | 90.60 | 90.13 | 91.12 | 89.80 | 90.24 |
| $\xi$ | 96.26 | 96.65 | 95.63 | 95.98 | 96.05 |

**Table 6**: Comparison of the average accuracy results for the skewed t GARCH model parameters using batch SVB and Seq-SVB with ten different starting values, $T_n = \{500, 550, 600, \ldots, 950\}$.

Table 5 and 6 summarize the mean accuracy results for the parameters of the skewed t GARCH model using UVB and Seq-SVB approaches, compared to the batch SVB method. The accuracy results are obtained by considering ten different starting values ranging from $T_n = 500$ to $T_n = 950$.

The findings indicate that the accuracy of the UVB method decreases as the number of updates increases. This suggests that the UVB method may struggle to provide accurate results when a larger number of updates are applied. On the other hand, the Seq-SVB method exhibits more consistent accuracy results across different starting values, indicating its robustness and insensitivity to the number of updates.

However, it is important to note that the Seq-SVB method requires a longer computational time compared to UVB. The mean computational time for Seq-SVB, considering ten different starting values, is 47.62 seconds for a single update, while UVB only requires 17.11 seconds. Therefore, the

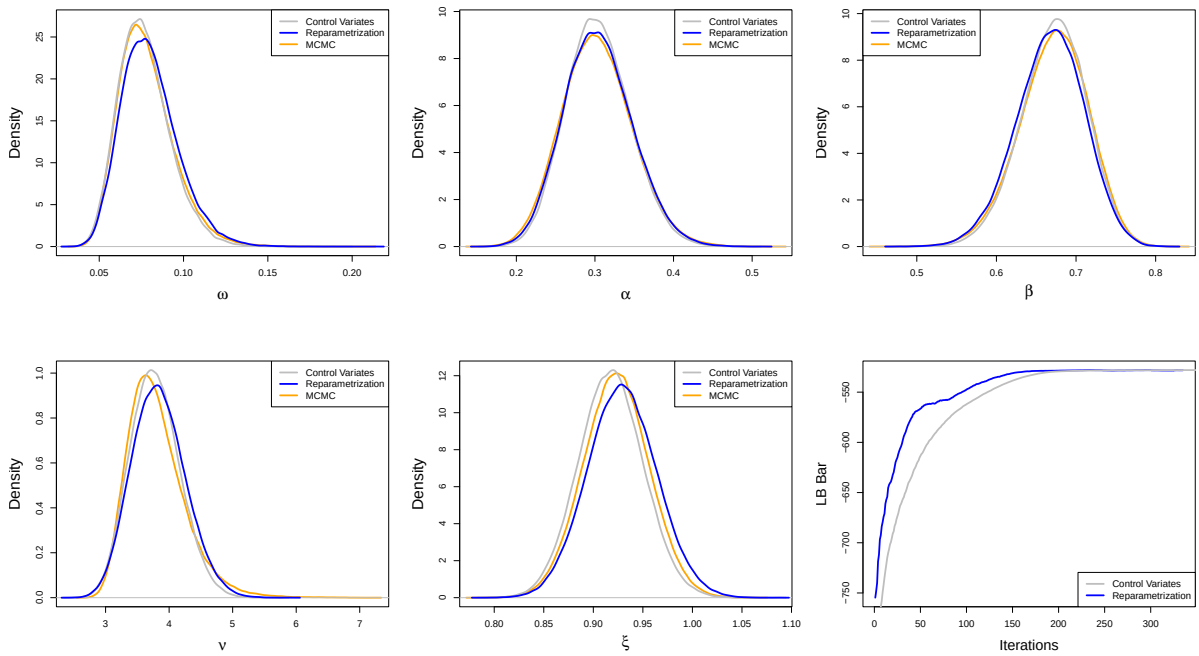


**Fig. 9**: S&P500 dataset. The posterior density plots of the skewed t GARCH model parameters $\omega, \alpha, \beta, \nu, \xi$ using MCMC and batch SVB with control variates and reparametrization trick methods.

choice between Seq-SVB and UVB should consider the trade-off between computational time and accuracy. Seq-SVB may provide higher accuracy but requires more computational resources, while UVB offers a more time-efficient solution at the expense of slightly lower accuracy.

In summary, the Seq-SVB method demonstrates robust accuracy across different starting values and update frequencies, while UVB is more sensitive to the number of updates but computationally faster. Researchers should consider these trade-offs based on their specific requirements and constraints when selecting between Seq-SVB and UVB for their applications.

# 7 Conclusion

This paper introduces and compares batch stochastic variational Bayes (SVB) algorithms for approximating posterior densities in GARCH-type models, including Gaussian, t, and skewed t GARCH models. The control variates and reparametrization trick methods are implemented and compared with traditional MCMC estimation. Both SVB methods demonstrate high accuracy in approximating the posterior densities of GARCH time series models.

For the models examined in this study, both the control variates and reparametrization trick approaches demonstrate notable improvements in computational efficiency compared to traditional MCMC methods. The computational time is influenced by the number of Monte Carlo samples $S$. In general, the control variates method exhibits faster computation times compared to the reparametrization trick method when the same number of MC samples is employed. However, the control variates method may struggle to maintain the same level of accuracy when using a small number of MC samples, especially for the longer time series.

We have provided sequential versions of the SVB approaches, namely UVB and Seq-SVB, to handle streaming data or scenarios with frequent updates. Both UVB and Seq-SVB outperform the batch SVB methods in terms of computational speed. UVB achieves significant time savings by only evaluating the likelihood of new observations in each update. However, UVB may sacrifice some accuracy compared to Seq-SVB, especially with a

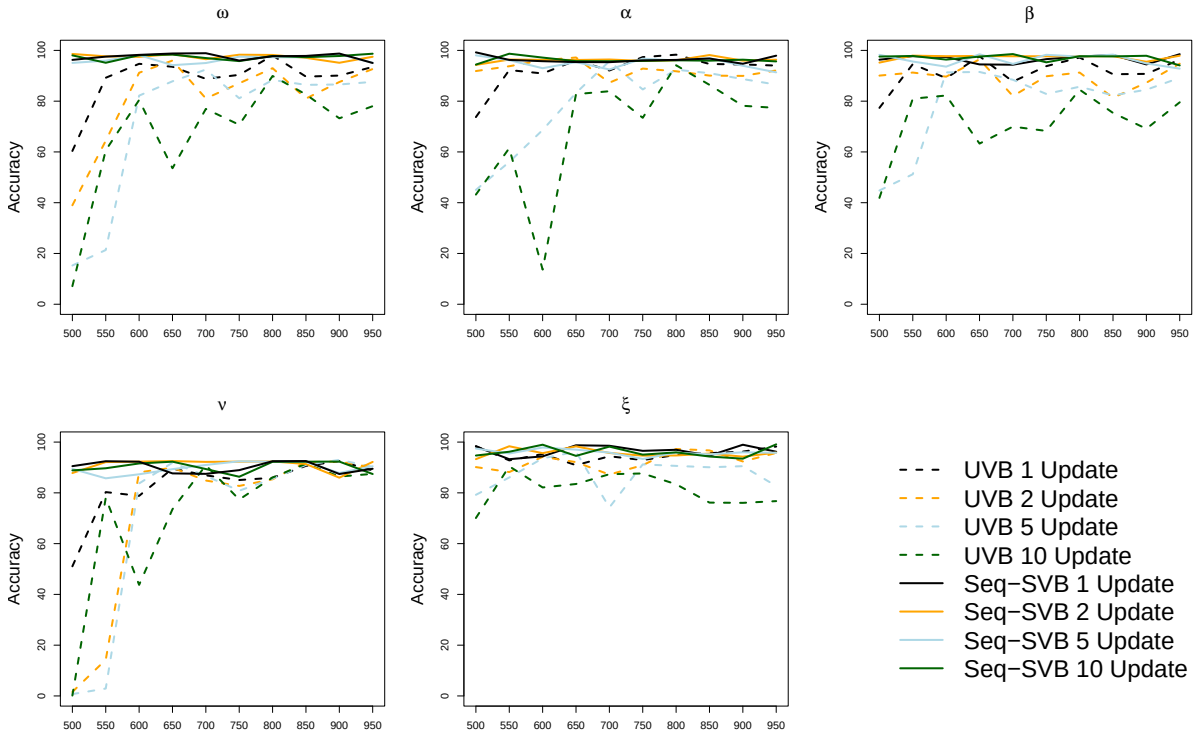


**Fig. 10**: Accuracy results of a real S&P 500 dataset with different starting values and different numbers of sequential updates under the skewed t GARCH model.

high number of sequential updates or a substantial number of new observations.

A promising approach is to run UVB and Seq-SVB in parallel, leveraging the strengths of both methods. Additionally, refreshing the pseudo-posterior in UVB using the posterior obtained from Seq-SVB can potentially improve accuracy while still benefiting from the computational efficiency of UVB.

In summary, stochastic variational inference methods, particularly the reparametrization trick approach, provide an effective and efficient framework for posterior approximation in GARCH time series models. The choice between UVB and Seq-SVB depends on the specific trade-off between computational speed and accuracy requirements, making them valuable tools for researchers and practitioners in various applications involving time series analysis.

**Acknowledgments.** This research includes computations using the computational cluster Katana supported by Research Technology Services at University of New South Wales, Sydney.

# Appendix A

## A.1 Analytical Derivatives for the Time-varying Variance Series in the Gaussian GARCH Model

By using the chain rule,

$$\frac{\partial \sigma_t^2}{\partial \theta_\phi} = \frac{\partial \sigma_t^2}{\partial \phi} \frac{\partial \phi}{\partial \theta_\phi}, \tag{A1}$$

where $\phi = (\omega, \alpha, \beta)^\top$ is a vector containing the original parameters for the classical GARCH(1,1) model. Taking the partial derivatives with respect to $\phi$, we can derive the following recursive formulas:

$$\frac{\partial \sigma_t^2}{\partial \omega} = 1 + \beta \frac{\partial \sigma_{t-1}^2}{\partial \omega},$$

$$\frac{\partial \sigma_t^2}{\partial \alpha} = y_{t-1}^2 + \beta \frac{\partial \sigma_{t-1}^2}{\partial \alpha},$$
$$\frac{\partial \sigma_t^2}{\partial \beta} = \sigma_{t-1}^2 + \beta \frac{\partial \sigma_{t-1}^2}{\partial \beta}.$$

The derivatives of the original parameters $\phi$ with respect to the unconstrained parameters $\theta_\phi = (\theta_\omega, \theta_{\psi_1}, \theta_{\psi_2})^\top$, $\frac{\partial \phi}{\partial \theta_\phi}$, are:

$$\begin{aligned}
\frac{\partial \omega}{\partial \theta_\omega} &= \exp(\theta_\omega), \\
\frac{\partial \alpha}{\partial \theta_{\psi_1}} &= \frac{\exp(-\theta_{\psi_1})}{\{1+\exp(-\theta_{\psi_1})\}^2\{1+\exp(-\theta_{\psi_2})\}}, \\
\frac{\partial \alpha}{\partial \theta_{\psi_2}} &= \frac{\exp(-\theta_{\psi_2})}{\{1+\exp(-\theta_{\psi_1})\}\{1+\exp(-\theta_{\psi_2})\}^2}, \\
\frac{\partial \beta}{\partial \theta_{\psi_1}} &= \frac{\exp(-\theta_{\psi_1}-\theta_{\psi_2})}{\{1+\exp(-\theta_{\psi_1})\}^2\{1+\exp(-\theta_{\psi_2})\}}, \\
\frac{\partial \beta}{\partial \theta_{\psi_2}} &= -\frac{\exp(\theta_{\psi_2})}{\{1+\exp(-\theta_{\psi_1})\}\{1+\exp(\theta_{\psi_2})\}^2} \\
\frac{\partial \omega}{\partial \theta_{\psi_1}} &= \frac{\partial \omega}{\partial \theta_{\psi_2}} = \frac{\partial \alpha}{\partial \theta_\omega} = \frac{\partial \beta}{\partial \theta_\omega} = 0.
\end{aligned}$$

Combining all the parts together, we can derive an analytical form for Equation (A1):

$$\begin{aligned}
\frac{\partial \sigma_t^2}{\partial \theta_\omega} &= \frac{\partial \sigma_t^2}{\partial \omega}\frac{\partial \omega}{\partial \theta_\omega} = \omega + \beta \frac{\partial \sigma_{t-1}^2}{\partial \theta_\omega}, \\
\frac{\partial \sigma_t^2}{\partial \theta_{\psi_1}} &= \frac{\partial \sigma_t^2}{\partial \alpha}\frac{\partial \alpha}{\partial \theta_{\psi_1}} + \frac{\partial \sigma_t^2}{\partial \beta}\frac{\partial \beta}{\partial \theta_{\psi_1}} \\
&= \mathcal{D}_1 y_{t-1}^2 + \mathcal{D}_2 \sigma_{t-1}^2 + \beta \frac{\partial \sigma_{t-1}^2}{\partial \theta_{\psi_1}} \\
\frac{\partial \sigma_t^2}{\partial \theta_{\psi_2}} &= \frac{\partial \sigma_t^2}{\partial \alpha}\frac{\partial \alpha}{\partial \theta_{\psi_2}} + \frac{\partial \sigma_t^2}{\partial \beta}\frac{\partial \beta}{\partial \theta_{\psi_2}} \\
&= \mathcal{D}_3 y_{t-1}^2 + \mathcal{D}_4 \sigma_{t-1}^2 + \beta \frac{\partial \sigma_{t-1}^2}{\partial \theta_{\psi_2}},
\end{aligned}$$

where

$$\begin{aligned}
\mathcal{D}_1 &= \alpha \frac{\exp(-\theta_{\psi_1})}{1+\exp(-\theta_{\psi_1})}, \\
\mathcal{D}_2 &= \beta \frac{\exp(-\theta_{\psi_1})}{1+\exp(-\theta_{\psi_1})}, \\
\mathcal{D}_3 &= \alpha \frac{\exp(-\theta_{\psi_2})}{1+\exp(-\theta_{\psi_2})} \\
\mathcal{D}_4 &= -\beta \frac{\exp(\theta_{\psi_2})}{1+\exp(\theta_{\psi_2})}.
\end{aligned}$$

## A.2 Analytical Derivatives of the Log-likelihood Function of the t GARCH Model

The gradient of the log-likelihood function of the t GARCH model is given by

$$\nabla_\theta \ell(\theta) = \frac{\partial \ell(\theta)}{\partial \theta} = \left( \frac{\partial \ell(\theta)}{\partial \theta_\phi}^\top, \frac{\partial \ell(\theta)}{\partial \theta_\nu}^\top \right)^\top. \quad \text{(A2)}$$

Differentiating the log-likelihood function with respect to $\theta_\phi$, we obtain

$$\frac{\partial \ell(\theta_\phi, \theta_\nu)}{\partial \theta_\phi} = \frac{1}{2}\sum_{t=1}^{T} \frac{1}{\sigma_t^2}\frac{\partial \sigma_t^2}{\partial \theta_\phi}\left[\frac{y_t^2}{\sigma_t^2}\mathcal{A} - 1\right],$$

where

$$\mathcal{A} = \frac{\sigma_t^2(\nu+1)}{(v-2)\sigma_t^2 + y_t^2}.$$

Note that the recursive gradient $\frac{\partial \sigma_t^2}{\partial \theta_\phi}$ can be found as in Appendix A.1.

Differentiating the log-likelihood function with respect to $\theta_\nu$, we obtain

$$\begin{aligned}
\frac{\partial \ell(\theta_\phi, \theta_\nu)}{\partial \theta_\nu} &= \frac{\partial \ell(\theta_\phi, \theta_\nu)}{\partial \nu}\frac{\partial \nu}{\partial \theta_\nu} \\
\frac{\partial \ell(\theta_\phi, \theta_\nu)}{\partial \nu} &= \sum_{t=1}^{T}\Big\{\mathcal{B} + \mathcal{A}\frac{y_t^2}{2(\nu-2)\sigma_t^2} \\
&\quad - \frac{1}{2}\log\left(1 + \frac{y_t^2}{(\nu-2)\sigma_t^2}\right)\Big\} \\
\frac{\partial \nu}{\partial \theta_\nu} &= \frac{\exp(\theta_\nu)}{1+\exp(\theta_\nu)},
\end{aligned}$$

where

$$\mathcal{B} = \frac{1}{2}\Psi\left(\frac{\nu+1}{2}\right) - \frac{1}{2}\Psi\left(\frac{\nu}{2}\right) - \frac{1}{2(\nu-2)},$$

and $\Psi(\cdot)$ is the derivative of the logarithmic gamma function.

## A.3 Analytical Derivatives of the Log-likelihood Function of Skewed t GARCH Model

Similarly to Section A.2, the analytical derivatives for the log-likelihood function $\nabla_\theta \log p(y|\theta)$ of the skewed t GARCH model can be decomposed as

follows, recalling that $\phi = (\omega, \alpha, \beta)^\top$ and $\theta_\phi = (\theta_\omega, \theta_\alpha, \theta_\beta)^\top$:

$$\nabla_\theta \log p(y|\theta) = \frac{\partial \ell(\theta)}{\partial \theta} = \left( \frac{\partial \ell(\theta)}{\partial \theta_\phi}^\top, \frac{\partial \ell(\theta)}{\partial \theta_\nu}^\top, \frac{\partial \ell(\theta)}{\partial \theta_\xi}^\top \right)^\top. \quad \text{(A3)}$$

Differentiating the log-likelihood function with respect to the first part of the unconstrained parameters $\theta_\phi$ yields

$$\frac{\partial l(\theta)}{\partial \theta_\phi} = \frac{1}{2} \sum_{t=1}^{T} \frac{1}{\sigma_t^2} \frac{\partial \sigma_t^2}{\partial \theta_\phi} (\mathcal{C} - 1),$$

where

$$\mathcal{C} = \frac{s(\nu+1)\left(\frac{s y_t}{\sigma_t} + m\right) y_t}{(\nu-2)\xi^{2I_t}\left(1 + \frac{(s z_t + m)^2}{\nu-2}\xi^{-2I_t}\right)\sigma_t}.$$

The recursive sequence $\frac{\partial \sigma_t^2}{\partial \theta_\phi}$ has been computed before and can be found in the Gaussian GARCH model reparametrization trick part of Section A.1. We can use the chain rule to derive the gradient of the log-likelihood function with respect to the transformed parameter $\theta_\nu$. We first let

$$h = (1+\nu) \log \left\{ 1 + \xi^{-2I_t} \frac{\left(\frac{s y_t}{\sigma_t + m}\right)^2}{\nu - 2} \right\}$$

so that

$$\begin{aligned} \frac{\partial \ell(\theta)}{\partial \theta_\nu} &= \frac{\partial \ell(\theta)}{\partial \nu} \frac{\partial \nu}{\partial \theta_\nu} \\ &= T\left( \frac{1}{2}\Psi(\frac{\nu+1}{2}) \frac{\partial \nu}{\partial \theta_\nu} - \frac{1}{2}\Psi(\frac{\nu}{2}) \frac{\partial \nu}{\partial \theta_\nu} \right. \\ &\quad \left. - \frac{1}{2(\nu-2)} \frac{\partial \nu}{\partial \theta_\nu} + \frac{\partial \log(s)}{\partial \theta_\nu} \right) - \frac{1}{2} \sum_{t=1}^{T} \frac{\partial h}{\partial \theta_\nu}, \end{aligned}$$

where

$$\frac{\partial \nu}{\partial \theta_\nu} = \frac{\exp(\theta_\nu)}{1 + \exp(\theta_\nu)}.$$

Next, we compute the partial derivatives of $\log(s)$ and of the function $h$ both with respect to the unconstrained parameter $\theta_\nu$. We have

$$\begin{aligned} \frac{\partial \log(s)}{\partial \theta_\nu} &= \frac{\partial \log(s)}{\partial m} \frac{\partial m}{\partial \nu} \frac{\partial \nu}{\partial \theta_\nu} \\ &= \frac{1}{2} \frac{\partial \log(s^2)}{\partial m} \frac{\partial m}{\partial \nu} \frac{\partial \nu}{\partial \theta_\nu}, \end{aligned}$$

where

$$\begin{aligned} \frac{\partial \log(s^2)}{\partial m} &= \frac{-2m}{(\xi^2 + \frac{1}{\xi^2} - 1) - m^2} \\ \frac{\partial m}{\partial \nu} &= -\left(\xi - \frac{1}{\xi}\right) \frac{1}{\sqrt{\pi}} \Gamma(\frac{\nu-1}{2}) \\ &\quad \frac{\left(\Psi(\frac{\nu}{2})(\nu-2) + \Psi(\frac{\nu-1}{2})(2-\nu) - 1\right)}{2\Gamma(\frac{\nu}{2})\sqrt{\nu-2}} \\ \frac{\partial \nu}{\partial \theta_\nu} &= \frac{\exp(\theta_\nu)}{1 + \exp(\theta_\nu)}. \end{aligned}$$

Let $f = \log\left(1 + \frac{\left(s\frac{y_t}{\sigma_t} + m\right)^2}{\nu-2}\xi^{-2I_t}\right)$ and $g = \frac{\left(s\frac{y_t}{\sigma_t} + m\right)^2}{\nu-2}$, the partial derivative of function $f$ with respect to $\nu$ is

$$\frac{\partial f}{\partial \nu} = \frac{\xi^{-2I_t} \frac{\partial g}{\partial \nu}}{1 + \frac{\left(s\frac{y_t}{\sigma_t} + m\right)^2}{\nu-2}\xi^{-2I_t}}.$$

The gradient of function $g$ with respect to $\nu$ is

$$\begin{aligned} \frac{\partial g}{\partial \nu} &= (\nu-2)^{-1} \frac{\partial\left((s\frac{y_t}{\sigma_t} + m^2)\right)}{\partial \nu} \\ &\quad + \left(s\frac{y_t}{\sigma_t} + m\right)^2 \frac{\partial\left((\nu-2)^{-1}\right)}{\partial \nu}. \end{aligned}$$

We also have

$$\begin{aligned} \frac{\partial\left((\nu-2)^{-1}\right)}{\partial \nu} &= -\frac{1}{(\nu-2)^2} \\ \frac{\partial\left((s\frac{y_t}{\sigma_t} + m^2)\right)}{\partial \nu} &= 2\left(s\frac{y_t}{\sigma_t} + m\right) \frac{\partial\left(s\frac{y_t}{\sigma_t} + m\right)}{\partial \nu} \\ &= 2\left(s\frac{y_t}{\sigma_t} + m\right)\left(\frac{y_t}{\sigma_t}\frac{\partial s}{\partial \nu} + \frac{\partial m}{\partial \nu}\right). \end{aligned}$$

Since $\frac{\partial m}{\partial \nu}$ is stated in Equation A4, we are only required to find $\frac{\partial s}{\partial \nu}$ to obtain an analytical expression for the above. This is

$$\begin{aligned}\frac{\partial s}{\partial \nu} &= \frac{1}{2\sqrt{\left(\xi^2+\frac{1}{\xi^2}-1\right)-m^2}}\frac{\partial(\xi^2+\frac{1}{\xi^2}-1-m^2)}{\partial \nu}\\ &= \frac{1}{2\sqrt{\left(\xi^2+\frac{1}{\xi^2}-1\right)-m^2}}\left(-\frac{\partial(m^2)}{\partial \nu}\right)\\ &= \frac{1}{2\sqrt{\left(\xi^2+\frac{1}{\xi^2}-1\right)-m^2}}\left(-2m\frac{\partial m}{\partial \nu}\right)\\ &= -\frac{1}{\sqrt{\left(\xi^2+\frac{1}{\xi^2}-1\right)-m^2}}m\frac{\partial m}{\partial \nu}.\end{aligned}$$

Given function $h$, its first derivative with respect to $\nu$ can be computed using the chain rule:

$$\frac{\partial h}{\partial \nu} = (1+\nu)\frac{\partial f}{\partial \nu} + f$$

and the first derivative with respect to the transformed parameter $\theta_\nu$ is

$$\frac{\partial h}{\partial \theta_\nu} = \frac{\partial h}{\partial \nu}\frac{\partial \nu}{\partial \theta_\nu},$$

where each component is provided above.

The gradients of the log-likelihood function with respect to the transformed parameter $\theta_\xi$ can be calculated in a similar way:

$$\begin{aligned}\frac{\partial \ell(\theta)}{\partial \theta_\xi} &= \frac{\partial \ell(\theta)}{\partial \xi}\frac{\partial \xi}{\partial \theta_\xi}\\ &= T\left(-\frac{\xi^2-1}{\xi^3+\xi}\frac{\partial \xi}{\partial \theta_\xi}+\frac{\partial \log(s)}{\partial \theta_\xi}\right)-\frac{1}{2}\sum_{t=1}^{T}\frac{\partial h}{\partial \theta_\xi},\end{aligned}$$

where

$$\frac{\partial \xi}{\partial \theta_\xi} = \frac{\exp(\theta_\xi)}{1+\exp(\theta_\xi)}.$$

Differentiating $\log(s)$ with the unrestricted skewness parameters $\theta_\xi$ we have

$$\begin{aligned}\frac{\partial \log(s)}{\partial \theta_\xi} &= \frac{\partial \log(s)}{\partial \xi}\frac{\partial \xi}{\partial \theta_\xi}\\ &= \frac{1}{2}\frac{\partial \log(s^2)}{\partial \xi}\frac{\partial \xi}{\partial \theta_\xi}\\ &= \frac{1}{2}\frac{\partial \log\left((\xi^2+\frac{1}{\xi^2}-1)-m^2\right)}{\partial \xi}\frac{\partial \xi}{\partial \theta_\xi}\\ &= \frac{1}{2}\left(\frac{2\xi-\frac{2}{\xi^3}-\frac{\partial(m^2)}{\partial \xi}}{(\xi^2+\frac{1}{\xi^2}-1)-m^2}\right)\frac{\partial \xi}{\partial \theta_\xi}.\end{aligned}$$

With some further differentiation, we can get

$$\frac{\partial(m^2)}{\partial \xi} = 2m\frac{\partial m}{\partial \xi} \tag{A4}$$

$$\frac{\partial m}{\partial \xi} = \left(1+\frac{1}{\xi^2}\right)\frac{\Gamma(\frac{\nu-1}{2})\sqrt{\nu-2}}{\sqrt{\pi}\Gamma(\frac{\nu}{2})}. \tag{A5}$$

The partial derivative of the function $f$ with respect to $\theta_\xi$ is

$$\frac{\partial f}{\partial \theta_\xi} = \frac{\partial f}{\partial \xi}\frac{\partial \xi}{\partial \theta_\xi}, \tag{A6}$$

where

$$\begin{aligned}\frac{\partial \xi}{\partial \theta_\xi} &= \frac{\exp(\theta_\xi)}{1+\exp(\theta_\xi)}\\ \frac{\partial f}{\partial \xi} &= \frac{(\nu-2)^{-1}\frac{\partial}{\partial \xi}\left(\left(s\frac{y_t}{\sigma_t}+m\right)^2\xi^{-2I_t}\right)}{1+\frac{\left(s\frac{y_t}{\sigma_t}+m\right)^2}{\nu-2}\xi^{-2I_t}}\end{aligned}$$

We first let $r = \left(s\frac{y_t}{\sigma_t}+m\right)^2\xi^{-2I_t}$, hence,

$$\begin{aligned}\frac{\partial r}{\partial \xi} &= \frac{\partial\left(s\frac{y_t}{\sigma_t}+m\right)^2}{\partial \xi}\xi^{-2I_t}+\left(s\frac{y_t}{\sigma_t}+m\right)^2\frac{\partial(\xi^{-2I_t})}{\partial \xi}\\ &= 2\left(s\frac{y_t}{\sigma_t}+m\right)\left(\frac{y_t}{\sigma_t}\frac{\partial s}{\partial \xi}+\frac{\partial m}{\partial \xi}\right)\xi^{-2I_t}\\ &\quad+\left(s\frac{y_t}{\sigma_t}+m\right)^2\frac{\partial(\xi^{-2I_t})}{\partial \xi}\\ &= 2\left(s\frac{y_t}{\sigma_t}+m\right)\left(\frac{y_t}{\sigma_t}\frac{\partial s}{\partial \xi}+\frac{\partial m}{\partial \xi}\right)\xi^{-2I_t}\\ &\quad-\left(s\frac{y_t}{\sigma_t}+m\right)^2\left(2I_t\xi^{-2I_t-1}\right).\end{aligned}$$

The derivation of $\frac{\partial m}{\partial \xi}$ is stated in Equation A5, thus, it is only necessary to find the partial

derivatives for $\frac{\partial s}{\partial \xi}$:

$$\frac{\partial s}{\partial \xi} = \frac{1}{2\sqrt{\left(\xi^2 + \frac{1}{\xi^2} - 1\right) - m^2}} \frac{\partial(\xi^2 + \frac{1}{\xi^2} - 1 - m^2)}{\partial \xi}$$
$$= \frac{1}{2\sqrt{\left(\xi^2 + \frac{1}{\xi^2} - 1\right) - m^2}} \left(2\xi - \frac{2}{\xi^3} - 2m\frac{\partial m}{\partial \xi}\right),$$

which is the last part needed to compute the partial derivative of function $f$ with respect to $\theta_\xi$ in Equation A6.

Given the definition of $h$, the first derivative with respect to $\xi$ is

$$\frac{\partial h}{\partial \xi} = (1+\nu)\frac{\partial f}{\partial \xi},$$

and the first derivative with respect to the transformed parameter $\theta_\xi$ is

$$\frac{\partial h}{\partial \theta_\xi} = \frac{\partial h}{\partial \xi}\frac{\partial \xi}{\partial \theta_\xi}.$$

Therefore, the model-specific gradient with respect to the reparametrized parameters,

$$\nabla_\theta h_\lambda(\theta) = \nabla_\theta \log p(\theta) + \nabla_\theta \log p(y|\theta) - \nabla_\theta \log q_\lambda(\theta),$$

can be fully computed now.

# Appendix B Additional Figures & Tables

Tables B1-B3 present the accuracy results for the parameters of the Gaussian, t, and skewed t GARCH models, respectively. The accuracy is measured as a percentage and higher values indicate a closer approximation to the true values, which are reported for both the control variates and reparametrization trick approaches with different numbers of Monte Carlo samples. The tables are divided into two sections based on the number of observations, 1000 and 5000. The accuracy results in the tables represent the mean accuracy over 1,000 replicates.

Figure B1 - B3 present boxplots comparing the accuracy results of SVB algorithms for a GARCH(1,1) model with different innovations.

| | $\omega$ | $\alpha$ | $\beta$ |
|---|---|---|---|
| | 1000 Observations | | |
| CV S=5 | 94.67 | 94.82 | 94.92 |
| CV S=10 | 96.62 | 96.35 | 96.52 |
| CV S=50 | 98.09 | 97.53 | 97.57 |
| RT S=5 | 95.94 | 94.75 | 94.99 |
| RT S=10 | 96.43 | 95.31 | 95.63 |
| RT S=50 | 97.15 | 95.98 | 96.24 |
| | 5000 Observations | | |
| CV S=5 | 83.58 | 90.57 | 88.66 |
| CV S=10 | 91.04 | 94.08 | 92.99 |
| CV S=50 | 98.08 | 97.72 | 97.97 |
| RT S=5 | 95.34 | 94.65 | 94.61 |
| RT S=10 | 96.46 | 95.77 | 95.86 |
| RT S=50 | 97.68 | 97.38 | 97.24 |

**Table B1**: Average accuracy for the parameters of the Gaussian GARCH model. The accuracy results are reported for both the control variates and reparametrization trick approaches with different numbers of Monte Carlo samples and different time lengths.

| | $\omega$ | $\alpha$ | $\beta$ | $\nu$ |
|---|---|---|---|---|
| | 1000 Observations | | | |
| CV $S=5$ | 93.73 | 93.73 | 94.24 | 91.70 |
| CV $S=10$ | 95.57 | 95.12 | 95.59 | 92.23 |
| CV $S=50$ | 97.35 | 96.21 | 96.78 | 92.40 |
| RT $S=5$ | 95.92 | 93.98 | 94.67 | 91.81 |
| RT $S=10$ | 96.15 | 94.14 | 95.03 | 91.71 |
| RT $S=50$ | 96.50 | 93.81 | 95.54 | 91.32 |
| | 5000 Observations | | | |
| CV $S=5$ | 86.32 | 87.21 | 89.92 | 91.73 |
| CV $S=10$ | 90.88 | 91.79 | 93.05 | 94.07 |
| CV $S=50$ | 97.68 | 96.54 | 97.57 | 95.48 |
| RT $S=5$ | 95.64 | 94.36 | 94.81 | 94.42 |
| RT $S=10$ | 96.88 | 94.80 | 96.15 | 94.67 |
| RT $S=50$ | 98.05 | 95.75 | 97.36 | 94.91 |

**Table B2**: Average accuracy for the parameters of the t GARCH model. The accuracy results are reported for both the control variates and reparametrization trick approaches with different numbers of Monte Carlo samples and different time lengths.

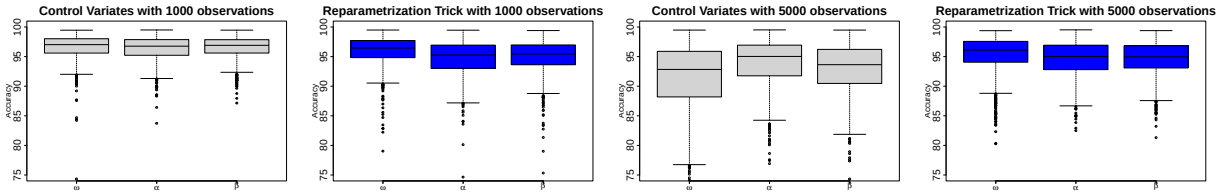


**Fig. B1**: Boxplots for the accuracy results of the control variates and reparametrization trick methods with $S = 10$ for the Gaussian GARCH model, with 1,000 and 5,000 observations, based on 1,000 replicates.

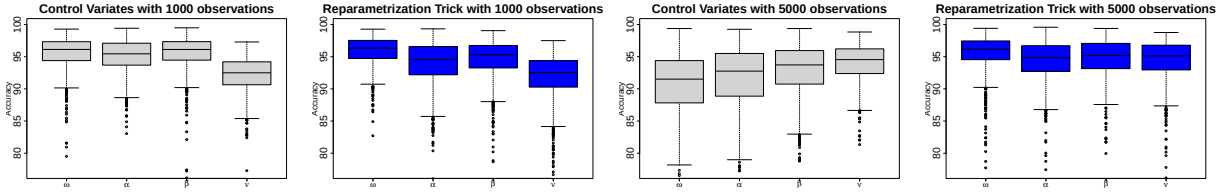


**Fig. B2**: Boxplots for the accuracy results of the control variates and reparametrization trick methods with $S = 10$ for the t GARCH model, with 1,000 and 5,000 observations, based on 1,000 replicates.

| | $\omega$ | $\alpha$ | $\beta$ | $\nu$ | $\xi$ |
|---|---|---|---|---|---|
| | | | 1000 Observations | | |
| CV $S = 5$ | 92.90 | 92.71 | 93.82 | 90.60 | 90.76 |
| CV $S = 10$ | 95.20 | 94.42 | 95.40 | 91.22 | 93.04 |
| CV $S = 50$ | 97.35 | 95.88 | 96.76 | 91.71 | 96.02 |
| RT $S = 5$ | 95.73 | 94.17 | 94.86 | 90.31 | 91.32 |
| RT $S = 10$ | 96.19 | 94.19 | 95.52 | 90.07 | 94.12 |
| RT $S = 50$ | 96.70 | 93.97 | 96.02 | 90.10 | 96.94 |
| | | | 5000 Observations | | |
| CV $S = 5$ | 84.62 | 85.21 | 88.56 | 90.41 | 84.13 |
| CV $S = 10$ | 88.80 | 89.49 | 91.69 | 92.78 | 88.70 |
| CV $S = 50$ | 97.52 | 96.28 | 97.38 | 94.91 | 95.13 |
| RT $S = 5$ | 94.41 | 93.65 | 93.77 | 93.02 | 82.34 |
| RT $S = 10$ | 96.32 | 94.64 | 95.57 | 93.85 | 88.77 |
| RT $S = 50$ | 98.03 | 95.76 | 97.57 | 94.29 | 95.76 |

**Table B3**: Average accuracy for the parameters of the skewed t GARCH model. The accuracy results are reported for both the control variates and reparametrization trick approaches with different numbers of Monte Carlo samples and different time lengths.

# References


Alonso AM, Maharaj EA (2006) Comparison of time series using subsampling. Computational statistics & data analysis 50(10):2589–2599

Ardia D, Hoogerheide LF (2010) Bayesian estimation of the GARCH (1, 1) model with student-t innovations. The R Journal 2(2):41–47

Ardia D, et al (2008) Financial risk management with Bayesian estimation of GARCH models, vol 612. Springer

Baillie RT, Bollerslev T (1989) Common stochastic trends in a system of exchange rates. the Journal of Finance 44(1):167–181

Beine M, Bénassy-Quéré A, Lecourt C (2002) Central bank intervention and foreign exchange rates: new evidence from FIGARCH estimations. Journal of International Money and Finance 21(1):115–144

Bishop CM, Nasrabadi NM (2006) Pattern Recognition and Machine Learning, vol 4. Springer

Bollerslev T (1986) Generalized autoregressive conditional heteroskedasticity. Journal of econometrics 31(3):307–327

Bollerslev T (1987) A conditionally heteroskedastic time series model for speculative prices and rates of return. The review of economics and statistics pp 542–547

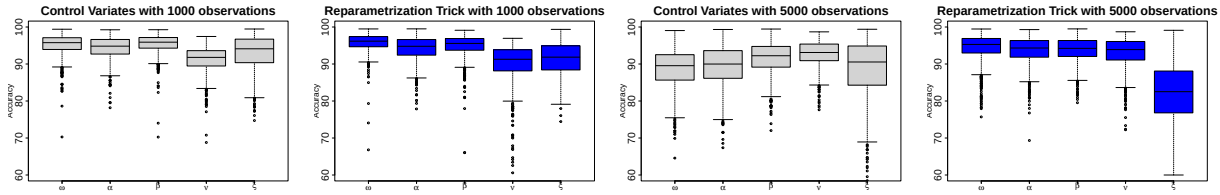


**Fig. B3**: Boxplots for the accuracy results of the control variates and reparametrization trick methods with $S = 10$ for the skewed t GARCH model, with 1,000 and 5,000 observations, based on 1,000 replicates.

Broderick T, Boyd N, Wibisono A, et al (2013) Streaming variational Bayes. Advances in neural information processing systems 26

Brooks SP, Gelman A (1998) General methods for monitoring convergence of iterative simulations. Journal of computational and graphical statistics 7(4):434–455

Chan JC, Grant AL (2016) Modeling energy price dynamics: GARCH versus stochastic volatility. Energy Economics 54:182–189

Contino C, Gerlach RH (2017) Bayesian tail-risk forecasting using realized GARCH. Applied Stochastic Models in Business and Industry 33(2):213–236

Degiannakis S (2004) Volatility forecasting: evidence from a fractional integrated asymmetric power ARCH skewed-t model. Applied Financial Economics 14(18):1333–1342

Duchi J, Hazan E, Singer Y (2011) Adaptive subgradient methods for online learning and stochastic optimization. Journal of machine learning research 12(7)

Engle R (2001) GARCH 101: The use of ARCH/-GARCH models in applied econometrics. Journal of economic perspectives 15(4):157–168

Engle RF (1982) Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. Econometrica: Journal of the econometric society pp 987–1007

Fernández C, Steel MF (1998) On bayesian modeling of fat tails and skewness. Journal of the american statistical association 93(441):359–371

Fiorentini G, Calzolari G, Panattoni L (1996) Analytic derivatives and the computation of GARCH estimates. Journal of applied econometrics 11(4):399–417

Foti N, Xu J, Laird D, et al (2014) Stochastic variational inference for hidden Markov models. Advances in neural information processing systems 27

Gerlach R, Tuyl F (2006) MCMC methods for comparing stochastic volatility and GARCH models. International Journal of Forecasting 22(1):91–107

González-Rivera G, Drost FC (1999) Efficiency comparisons of maximum-likelihood-based estimators in GARCH models. Journal of Econometrics 93(1):93–111

Gunawan D, Kohn R, Nott D (2021) Variational Bayes approximation of factor stochastic volatility models. International Journal of Forecasting 37(4):1355–1375

Hansen BE (1994) Autoregressive conditional density estimation. International Economic Review pp 705–730

Henneke JS, Rachev ST, Fabozzi FJ, et al (2011) MCMC-based estimation of Markov switching ARMA-GARCH models. Applied Economics 43(3):259–271

Hoffman MD, Blei DM, Wang C, et al (2013) Stochastic variational inference. Journal of Machine Learning Research

Iqbal F, Triantafyllopoulos K (2021) Bayesian inference of multivariate rotated GARCH models with skew returns. Communications

in Statistics-Simulation and Computation 50(10):3105–3123

Jondeau E, Zhang Q, Zhu X (2019) Average skewness matters. Journal of Financial Economics 134(1):29–47

Jordan MI, Ghahramani Z, Jaakkola TS, et al (1998) An introduction to variational methods for graphical models. Learning in graphical models pp 105–161

Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980

Kingma DP, Welling M (2013) Auto-encoding variational Bayes. arXiv preprint arXiv:13126114

Lambert M, Bonnabel S, Bach F (2022) The recursive variational Gaussian approximation (R-VGA). Statistics and Computing 32(1):10

Lambert P, Laurent S (2001) Modelling financial time series using GARCH-type models with a skewed student distribution for the innovations. Tech. rep.

Levy GF (2003) Analytic derivatives of asymmetric Garch models. Journal of Computational Finance 6(3):21–64

Li D, Clements A, Drovandi C (2021) Efficient Bayesian estimation for GARCH-type models via Sequential Monte carlo. Econometrics and Statistics 19:22–46

Liu L, Jiang H, He P, et al (2019) On the variance of the adaptive learning rate and beyond. arXiv preprint arXiv:190803265

Liu Y, Li JSH, Ng ACY (2015) Option pricing under garch models with hansen's skewed-t distributed innovations. The North American Journal of Economics and Finance 31:108–125

Livingston Jr G, Nur D (2022) Bayesian inference of multivariate-GARCH-BEKK models. Statistical Papers pp 1–26

Maestrini L, Wand MP (2018) Variational message passing for skew t regression. Stat 7(1):e196

Markowitz H (1952) Portfolio selection. The Journal of Finance 7(1):77–91. URL http://www.jstor.org/stable/2975974

Mozumder S, Choudhry T, Dempsey M (2021) Option pricing model biases: Bayesian and Markov Chain Monte Carlo regression analysis. Computational Economics 57:1287–1305

Nakatsuma T (2000) Bayesian analysis of ARMA-GARCH models: A Markov chain sampling approach. Journal of Econometrics 95(1):57–69

Nott DJ, Tan SL, Villani M, et al (2012) Regression density estimation with variational methods and stochastic approximation. Journal of Computational and Graphical Statistics 21(3):797–820

Ong VM, Nott DJ, Tran MN, et al (2018a) Variational Bayes with synthetic likelihood. Statistics and Computing 28:971–988

Ong VMH, Nott DJ, Smith MS (2018b) Gaussian variational approximation with a factor covariance structure. Journal of Computational and Graphical Statistics 27(3):465–478

Opper M, Archambeau C (2009) The variational gaussian approximation revisited. Neural computation 21(3):786–792

Ormerod JT, Wand MP (2010) Explaining variational approximations. The American Statistician 64(2):140–153

Ormerod JT, Wand MP (2012) Gaussian variational approximate inference for generalized linear mixed models. Journal of Computational and Graphical Statistics 21(1):2–17

Paisley J, Blei D, Jordan M (2012) Variational Bayesian inference with stochastic search. arXiv preprint arXiv:12066430

Politis DN (2003) The impact of bootstrap methods on time series analysis. Statistical science pp 219–230

Polyak BT (1964) Some methods of speeding up the convergence of iteration methods. Ussr

computational mathematics and mathematical physics 4(5):1–17

Quiroz M, Nott DJ, Kohn R (2022) Gaussian variational approximations for high-dimensional state space models. Bayesian Analysis 1(1):1–28

R Core Team (2023) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, URL https://www.R-project.org/

Ranganath R, Gerrish S, Blei D (2014) Black box variational inference. In: Artificial intelligence and statistics, PMLR, pp 814–822

Rezende DJ, Mohamed S, Wierstra D (2014) Stochastic backpropagation and approximate inference in deep generative models. In: International conference on machine learning, PMLR, pp 1278–1286

Robbins H, Monro S (1951) A stochastic approximation method. The annals of mathematical statistics pp 400–407

Robert CP, Casella G, Casella G (1999) Monte Carlo statistical methods, vol 2. Springer

Shiferaw YA (2019) Time-varying correlation between agricultural commodity and energy price dynamics with Bayesian multivariate DCC-GARCH models. Physica A: Statistical Mechanics and Its Applications 526:120,807

Shirvani A, et al (2020) Stock returns and roughness extreme variations: A new model for monitoring 2008 market crash and 2015 flash crash. Applied Economics and Finance 7(3):78–95

Silvennoinen A, Teräsvirta T (2021) Consistency and asymptotic normality of maximum likelihood estimators of a multiplicative time-varying smooth transition correlation GARCH model. Econometrics and Statistics

Tan LS, Nott DJ (2018) Gaussian variational approximation with sparse precision matrices. Statistics and Computing 28:259–275

Tieleman T, Hinton G, et al (2012) Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning 4(2):26–31

Titsias M, Lázaro-Gredilla M (2014) Doubly stochastic variational Bayes for non-conjugate inference. In: International conference on machine learning, PMLR, pp 1971–1979

Tomasetti N, Forbes C, Panagiotelis A (2022) Updating variational Bayes: fast sequential posterior inference. Statistics and Computing 32(1):4

Tran MN, Nguyen DH, Nguyen D (2021a) Variational Bayes on manifolds. Statistics and Computing 31:1–17

Tran MN, Nguyen TN, Dao VH (2021b) A practical tutorial on variational Bayes. arXiv preprint arXiv:210301327

Virbickaite A, Ausín MC, Galeano P (2015) Bayesian inference methods for univariate and multivariate GARCH models: A survey. Journal of Economic Surveys 29(1):76–96

Wand MP (2014) Fully simplified multivariate normal updates in non-conjugate variational message passing. Journal of Machine Learning Research

Wuertz D, RUnit S, Chalabi MY (2022) Package 'fGarch'. Tech. rep., working paper/manual, 09.11. 2009. URL http://cran. r-project. org/web ...

Xaba LD, Moroke ND, Metsileng LD (2021) Performance of MS-GARCH models: Bayesian MCMC-based estimation. Handbook of Research on Emerging Theories, Models, and Applications of Financial Econometrics pp 323–356

Zeiler MD (2012) Adadelta: an adaptive learning rate method. arXiv preprint arXiv:12125701