Deep Estimation for Volatility Forecasting

Léo Parent

Deep Estimation for Volatility Forecasting

Léo Parent
PRISM Sorbonne
Paris 1 Panthéon-Sorbonne University
leo.parent@etu.univ-paris1.fr

June 6, 2023

Abstract

The use of deep neural networks (DNNs) for the calibration of volatility models applied to pricing and hedging issues has led to abundant academic literature. In contrast, few works utilize these tools for model estimation with a focus on volatility forecasting. Based on this observation, this article introduces an innovative deep estimation method using historical data, specifically designed for volatility forecasting. To illustrate this method, the article focuses on estimating a version of the rough path-dependent volatility (RPDV) models (Parent, 2022), which is well-suited to the prediction objective and very complex to estimate using standard approaches. After formalizing the estimation problem within the framework of Bayesian decision theory, the article details the methodology for constructing the estimator function. Finally, a comprehensive evaluation of the estimation approach is conducted using both synthetic and market data to assess its performance.

Keywords: PDV model, volatility forecasting, rough volatility, rough path-dependent volatility model, volatility forecasting, deep learning, deep calibration, Bayesian decision theory.

JEL classification: C11, C13, C15, C19, C22, C51, C53, C58.

1 Introduction

The use of machine learning methods in finance has experienced a significant boom in recent years. Among the most important use cases is the deep calibration of volatility models which consists in using artificial neural networks (NNs) to determine a set of parameters for a certain model in order to best meet given pricing or hedging objectives. If this topic resulted in abundant literature (Bayer *et al.*, 2019; Barra *et al.*, 2020; Horvath *et al.* 2021; Rosenbaum and Zhang, 2021), research on the use of NNs for model estimation in the context of volatility forecasting is much scarcer. However, this subject also represents an important issue within quantitative finance.

The potential interest in deep estimation within this framework is manifold. Firstly, the last decade has seen the emergence of numerous new volatility models that could serve as powerful tools for volatility forecasting. However, due to their complexity, some of these models are practically difficult to estimate using standard approaches such as maximum likelihood-like methods. In addition, as decision theory demonstrates, the most probable set of parameters is not necessarily the optimal choice from a consequentialist perspective (Parmigian and Inoue, 2009; Berger, 2013). In fact, the optimality of a parameter set depends not only on the parameter distribution but also on the model's intended use for estimation (Hernandez, 2016). For instance,

the optimality criterion for the parameters of a model used in an options pricing perspective is typically defined by a fitting criterion, either of an option price map as in the article by Horvath *et al.* (2021), or of an implied volatility surface as in the article by Rosenbaum and Zhang (2021). Analogously, the optimality criterion used for parameter estimation in the context of volatility forecasting must reflect this objective to be fully consistent. However, the associated optimization problem can be very difficult, if not impossible, to solve analytically.

Accordingly, the present article aims to introduce a deep estimation method for volatility models based on historical data, grounded in the theoretical framework of Bayesian decision theory (BDT). The core principle of this method involves estimating a volatility model through the interaction of two neural networks (NNs): the first NN associates a historical dataset with a vector of parameters and state variables of the considered model, while the second NN uses this vector to estimate different moments of the associated volatility. This second NN compensates for the lack of analytical formulas for the volatility moments of the model, playing a role similar to the NN pricing map approximator proposed by Horvath *et al.* (2021). The volatility model under consideration for estimation is a specific version of the rough path-dependent volatility (RPDV) models (Paren,t 2022). It is a good candidate for the proposed approach since, on the one hand, it provides a framework for capturing the main empirical features that characterize volatility dynamics, making it a potentially suitable model for volatility forecasting, and on the other hand, it is very complex to estimate using traditional approaches. Obviously, although the article focuses on this model, the general principle of the proposed estimation method can be applied to other volatility models as well.

The paper is organized as follows. Section 2 provides the definition of the RPDV model intended for estimation, along with an explanation of its role in forecasting. This leads to the formalization of the estimation problem as an optimization issue within the BDT framework. In section 3, a method for constructing an estimator function is presented, aiming to address this problem by utilizing two deep neural networks within a collaborative game framework. Lastly, in section 4, a comprehensive evaluation of the resulting estimator function is conducted from various perspectives, using both synthetic and market data.

2 Exposition of the estimation problem

2.1 The model to be estimated

2.1.1 The considered rough path-dependent volatility model

In the present paper, we aim to estimate the following RPDV model:

$$\begin{cases} \dfrac{\mathrm{d}P_t}{P_t} &= \big(\lambda_1\sigma_t + \lambda_2(\sigma_t)^2\big)\mathrm{d}t + \sigma_t \mathrm{d}B_t, \\ \sigma_t &= \beta_0 + \beta_1 R_{1,t} + \beta_2\sqrt{R_{2,t}}, \\ R_{1,t} &= \displaystyle\int_{-\infty}^{t-\epsilon} (t-u)^{-\alpha_1}\left(\frac{\mathrm{d}P_u}{P_u} - \kappa_1 \cdot R_{1,u}\mathrm{d}u\right), \\ R_{2,t} &= \displaystyle\int_{-\infty}^{t-\epsilon} (t-u)^{-\alpha_2}\big(\sigma_u^2 - \kappa_2 \cdot R_{2,u}\big)\mathrm{d}u \end{cases}. \tag{1}$$

Here, the asset price thus depends on λ_1 and λ_2, which are positive risk premia, B is a Brownian motion that constitutes the unique source of randomness, and σ is the volatility process. This volatility process is a multilinear function with β_0 a positive constant, $\beta_1 \leq 0$ a sensitivity parameter to R_1 that can be viewed as an

asset price trend variable, and $\beta_2 \geq 0$ a sensitivity parameter to R_2 that can be viewed as a variable measuring recent market price activity regardless of the sign of the trend. ϵ is a positive parameter close to zero that encodes a latency of the impact of price dynamics on the volatility process. Technically, this parameter allows for values of α_j greater than 0.5 to be given without causing divergence issues. Furthermore, the memory of the processes R_1 and R_2 depends on the respective positive parameters α_1, κ_1 and α_2, κ_2.

This model has several remarkable properties that make it highly suitable for volatility forecasting issues. First, it is structurally adapted to jointly capture two important empirical features, which are the rough behavior and the path-dependence of the volatility process. The rough volatility dynamics are determined by the rough kernels $K_1(\tau) = \tau^{-\alpha_1}$ and $K_2(\tau) = \tau^{-\alpha_2}$, while the model incorporates path-dependency through the processes R_1 and R_2. Additionally, this version of the RPDV model shares a similar structure with the PDV model introduced by Guyon and Lekeufack (2022), which has demonstrated strong predictive capabilities in volatility forecasting.

2.1.2 The Markovian approximation of the model and its discretization scheme

Like other rough volatility models, the RPDV model is non-Markovian, which makes it difficult to simulate efficiently (Parent, 2022). However, as shown by Parent, we can approximate model 1 by the following Markovian model (see Appendix A.1), which will be referred to as the M-RPDV model. This model substitutes rough kernels $t^{-\alpha_j}$ with kernels of the form $\tilde{K}_j(\tau) = \sum_{i=1}^{n} w_{j,i}\gamma_{j,i}e^{-\gamma_{j,i}\tau}$:

$$\begin{cases} \dfrac{\mathrm{d}P_t}{P_t} & = \big(\lambda_1\sigma_t + \lambda_2(\sigma_t)^2\big)\mathrm{d}t + \sigma_t \mathrm{d}B_t, \\ \sigma_t & = \beta_0 + \beta_1 R_{1,t} + \beta_2\sqrt{R_{2,t}}, \\ \mathrm{dR}_{1,t} & = \Gamma_1 \cdot \left(\dfrac{\mathrm{d}P_t}{P_t} - \kappa_1 R_{1,t}\mathrm{d}t\right) - \Gamma_1 \odot \mathrm{R}_{1,t}\mathrm{d}t, \\ \mathrm{dR}_{2,t} & = \Gamma_2 \cdot \left(\sigma_t^2 - \kappa_2 R_{2,t}\right)\mathrm{d}t - \Gamma_2 \odot \mathrm{R}_{2,t}\mathrm{d}t, \\ R_{1,t} & = \mathrm{W}_1^\top \mathrm{R}_{1,t}, \\ R_{2,t} & = \mathrm{W}_2^\top \mathrm{R}_{2,t}, \end{cases} \tag{2}$$

where W_j the vector of weights $(w_{j,i})_{1\leq i\leq n}$ and Γ_j the vector of discount coefficients $(\gamma_{j,i})_{1\leq i\leq n}$, such that

$$\mathrm{W}_j = \begin{bmatrix} w_{j,1} \\ ... \\ w_{j,n} \end{bmatrix}, \quad \Gamma_j = \begin{bmatrix} \gamma_{j,1} \\ ... \\ \gamma_{j,n} \end{bmatrix}.$$

The method used to obtain these vectors is presented in Appendix A.2. It should be noted that model 2 depends on the parameter vector

$$\phi = \big(\lambda_1, \lambda_2, \beta_0, \beta_1, \beta_2, \alpha_1, \alpha_2, \kappa_1, \kappa_2\big),$$

and that all relevant information at time T for the volatility dynamics is aggregated into the following vector of state variables:

$$\mathrm{R}_T = \left(R_{1,T}^{(1)}, ..., R_{1,T}^{(n)}, R_{2,T}^{(1)}, ..., R_{2,T}^{(n)}\right).$$

Consequently, the estimation procedure will consist of estimating the $2n+9$ vector

$$\theta_T = \Big(\lambda_1, \lambda_2, \beta_0, \beta_1, \beta_2, \alpha_1, \alpha_2, \kappa_1, \kappa_2, R_{1,T}^{(1)}, ..., R_{1,T}^{(n)}, R_{2,T}^{(1)}, ..., R_{2,T}^{(n)}\Big), \tag{3}$$

with T a given period. This vector is therefore composed of 9 parameters and $2n$ state variables. To perform simulations from the model required by the estimation procedure, we use the following explicit Euler discretization scheme:

$$\begin{cases} P_{t+\Delta t} & = P_t\Big(1 + \big(\lambda_1\sigma_t + \lambda_2(\sigma_t)^2\big)\Delta t + \sigma_t\big(B_{t+\Delta t} - B_t\big)\Big), \\ \mathrm{R}_{1,t+\Delta t} & = \mathrm{R}_{1,t} \odot \big(\mathbf{1}_n - \Gamma_1 \cdot \Delta t\big) + \Gamma_1 \cdot \left(\dfrac{P_{t+\Delta t} - P_t}{P_t} - \kappa_1 R_{1,t}\Delta t\right), \\ \mathrm{R}_{2,t+\Delta t} & = \mathrm{R}_{2,t} \odot \big(\mathbf{1}_n - \Gamma_2 \cdot \Delta t\big) + \Gamma_2 \cdot \big(\sigma_t^2 - R_{2,t}\big)\Delta t, \\ R_{1,t+\Delta t} & = \mathrm{W}_1^\top \mathrm{R}_{1,t+\Delta t}, \\ R_{2,t+\Delta t} & = \mathrm{W}_1^\top \mathrm{R}_{1,t+\Delta t} \\ \sigma_{t+\Delta t} & = \beta_0 + \beta_1 R_{1,t+\Delta t} + \beta_2\sqrt{R_{2,t}}, \end{cases} \tag{4}$$

with Δt being the time step of simulations and $(B_{t+\Delta t} - B_t) \sim \mathcal{N}(0, \Delta t)$. It is important to note that in order to ensure the stability of the scheme, all coordinates of Γ_j must be lower than $\frac{1}{\Delta t}$. If one wishes to eliminate this condition, an alternative is to opt for an implicit-explicit scheme analogous to the scheme proposed by Rosenbaum and Zhang (2021) for the quadratic rough Heston model. The time step Δt used in this article is $\frac{1}{19656}$[1] year, and the larger discount factor is equal to 10000 (expressed in years). Therefore, because $\forall i, j, \gamma_{j,i}\Delta t < 1$, this stability issue does not arise.

2.2 The Bayesian estimation problem to solve: a forecasting objective-based estimation problem

2.2.1 The forecasting issue

The estimation method presented in this article for model 4 is specifically designed to address a particular forecasting problem. More precisely, we place in a context in which we have a data matrix D of the form

$$\mathrm{D} = \begin{pmatrix} P_{t_1} & \tilde{\sigma}_{t_1} \\ ... & ... \\ P_{t_N} & \tilde{\sigma}_{t_N} \end{pmatrix}, \tag{5}$$

where $t_1 < \ldots < t_N = T$, P represents the price of a financial asset, and $\tilde{\sigma}$ is a proxy of realized volatility defined as the square root of the sum of squares of a sample of 78 observations of logarithmic returns over the considered period[2]. From this $N \times 2$ data matrix, we want to get an estimator as accurate as possible of the following set of conditional moment vectors:

$$\Omega_{\mathrm{M}} = \left\{\Big(\mathbb{E}\big[\sigma_{T+\delta_k}|\mathrm{D}\big], \mathbb{S}\mathrm{td}\big[\sigma_{T+\delta_k}|\mathrm{D}\big]\Big)\right\}_{k=1}^{p}, \tag{6}$$

[1]The reason for choosing this discretization time step is that a trading day is approximately equal to $\frac{1}{252}$ of a year, and the realized volatility estimator used in this article is calculated using 78 price observations per trading day: $\frac{1}{252} \times \frac{1}{78} = \frac{1}{19656}$.

[2]Regarding the simulated data, this proxy is calculated from 78 log-returns evenly distributed over a period of $\frac{1}{252}$ year. For the real data used in section 4.2, $\tilde{\sigma}$ is calculated from a sample of 78 5-minute log-returns.

where $\delta_{k1\leq k\leq p}$ represents different time horizons. In this article, we will consider the horizons of 1, 5, 21, 42, and 63 trading days, which are defined here as $\frac{1}{252}$ year. The RPDV model will therefore serves as a tool to estimate these moments. Consequently, contrary to standard statistical approaches like maximum likelihood estimation, the estimation procedure will not consist of determining the most likely vector θ_T, but rather the vector θ_T that serves this forecasting goal the best.

2.2.2 The Bayesian estimation problem

To propose an appropriate estimation method for the RPDV model that aligns with the forecasting objective defined in section 2.2.1, we adopt the theoretical framework of Bayesian decision theory (Berger, 2013, Bickel and Doksum, 2015). As a result, we assume that the dynamics of (P, σ) follow a model given by 4, and $\boldsymbol{\theta}_t$ is considered as a random vector with a prior distribution $\boldsymbol{\pi}$ (i.e., $\boldsymbol{\theta}_t \sim \boldsymbol{\pi}$), $\forall\ t$. Under these assumptions and following the principles of BDT, an estimator $\hat{\boldsymbol{\theta}}_T$ of $\boldsymbol{\theta}_T$ is optimal given D and a loss function L if it minimizes the expected posterior loss defined as follows:

$$\mathbb{E}_{\boldsymbol{\pi}_{\mathrm{D}}}\Big[L\big(\boldsymbol{\theta}_T, \hat{\theta}_T\big)\Big] = \int_{\mathbb{R}^{2n+9}} L\big(\theta_T, \hat{\theta}_T\big)\mathrm{d}\boldsymbol{\pi}_{\mathrm{D}}\left(\theta_T\right), \tag{7}$$

where $\boldsymbol{\pi}_{\mathrm{D}}$ represents the posterior distribution for $\boldsymbol{\theta}_T$ given D[3]. Regarding the loss function L, its purpose is to capture the objective of the estimation, which is to obtain an estimator of the conditional moments in Ω_{M}. This function L is defined as follows:

$$L\big(\theta_T, \hat{\theta}_T\big) = \sum_{k=1}^{p} c_k \cdot C\left(M(\theta_T, \delta_k), M(\hat{\theta}_T, \delta_k)\right), \tag{8}$$

where $\{c_k\}_{k=1}^{p}$ are positive weights, C is another loss function, and M is a function defined as:

$$M(\theta_T, \delta_k) = \Big(\mathbb{E}\big[\sigma_{T+\delta_k}|\theta_T\big], \mathbb{S}\mathrm{td}\big[\sigma_{T+\delta_k}|\theta_T\big]\Big), \quad \forall\ \boldsymbol{\pi}(\theta_T) \neq 0\ \delta_k \in \mathbb{R}_+, \tag{9}$$

where $\mathbb{E}\big[\sigma_{T+\delta_k}|\theta_T\big]$ and $\mathbb{S}\mathrm{td}\big[\sigma_{T+\delta_k}|\theta_T\big]$ represent the conditional mean and standard deviation of volatility at horizon δ_k given θ_T. In other words, the cost associated with an estimator $\hat{\theta}_T$ given the true θ-vector θ_T is a function of the prediction error in the mean and standard deviation of volatility for time horizons $T+\delta_1, ..., T+\delta_p$ induced by this choice of θ-estimator. This cost is influenced by the form of C, which will be specified in section 2.3. Irrespective of the specific form of C and within the previously established framework, the Bayes estimator of θ_T under the posterior measure $\boldsymbol{\pi}_{\mathrm{D}}$ is a solution to the following optimization program:

$$\underset{\hat{\theta}_T \in \mathbb{R}^{2n+9}}{\arg\min} \sum_{k=1}^{p} c_k \cdot \mathbb{E}_{\boldsymbol{\pi}_{\mathrm{D}}}\Big[C\left(M(\theta_T, \delta_k), M(\hat{\theta}_T, \delta_k)\right)\Big]. \tag{10}$$

The objective of the estimation method introduced in this article, which will be presented in section 3, is to find an approximate solution to this optimization problem.

2.3 The loss function: a sum of proxy divergence measures

As mentioned in section 2.2, the choice of the loss function is crucial as it implicitly encodes preferences regarding estimation errors. The mean squared error (MSE) is commonly used as a loss function in forecasting problems due to its simplicity. However, although the MSE has certain advantages, it may not be the most suitable loss function for the forecasting objective. In this case, using the MSE would give excessive weight

[3]In practice, updating $\boldsymbol{\pi}$ with the information contained in D (i.e., determining $\boldsymbol{\pi}_{\mathrm{D}}$) is not a trivial task. The estimation procedure presented in section 3 does not require directly computing this posterior measure.

to situations where the expected volatility and volatility of volatility are high compared to cases where these quantities are low. Therefore, we will employ an ad-hoc loss function that can be interpreted as a sum of proxy divergence measures.

The starting point is the empirical observation that log-volatility increments closely follow a Gaussian distribution (Gatheral *et al.* 2018), and empirical volatility distributions closely resemble log-normal distributions (Tegnér and Poulsen 2018). Additionally, the conditional volatility distributions generated by the RPDV model also exhibit a similar log-normal behavior. Based on these observations, if we assume $\boldsymbol{\theta}_T = \theta_T$, we can approximate the distribution of $\sigma_{T+\delta}$ by a log-normal distribution with parameters $\mathrm{m}(\theta_{T+\delta})$ and $\mathrm{s}(\theta_{T+\delta})$. Leveraging the analytical expressions for the expectation and variance of the log-normal distribution, we can express them as follows:

$$\mathbb{E}\big[\sigma_{T+\delta}|\theta_T\big] \approx e^{\mathrm{m}(\theta_{T+\delta_k})+\mathrm{s}(\theta_{T+\delta_k})^2/2}, \quad \mathbb{V}\mathrm{ar}\big[\sigma_{T+\delta}|\theta_T\big] \approx \left(e^{\mathrm{s}(\theta_{T+\delta_k})^2}-1\right) e^{2\mathrm{m}(\theta_{T+\delta_k})+\mathrm{s}(\theta_{T+\delta_k})^2}.$$

Equivalently, we can write (see details in appendix B):

$$\begin{aligned} \tilde{\mathrm{m}}\big(\theta_{T+\delta_k}\big) &= \log\left(\mathbb{E}\big[\sigma_{T+\delta}|\theta_T\big]\right) - 0.5\log\left(\frac{\mathbb{V}\mathrm{ar}\left[\sigma_{T+\delta} \mid \theta_T\right]}{\mathbb{E}\left[\sigma_{T+\delta} \mid \theta_T\right]^2}+1\right), \\ \tilde{\mathrm{s}}\big(\theta_{T+\delta_k}\big)^2 &= \log\left(\frac{\mathbb{V}\mathrm{ar}\left[\sigma_{T+\delta} \mid \theta_T\right]}{\mathbb{E}\left[\sigma_{T+\delta} \mid \theta_T\right]^2}+1\right), \end{aligned} \tag{11}$$

where $\tilde{\mathrm{m}}\big(\theta_{T+\delta_k}\big)$ and $\tilde{\mathrm{s}}\big(\theta_{T+\delta_k}\big)^2$ are approximations of $\mathrm{m}(\theta_{T+\delta})$ and $\mathrm{s}(\theta_{T+\delta})$, respectively. Furthermore, the divergence between two log-normal distributions, $\mathcal{LN}\big(m_1,(s_1)^2\big)$ and $\mathcal{LN}\big(m_2,(s_2)^2\big)$, can be expressed as an analytical function of m_1, s_1, m_2, and s_2. Specifically, for the case of Kullback-Leibler (KL) divergence, we have (Gil *et al.*, 2013):

$$\mathcal{D}_{KL}\left(\mathbb{P}_1,\mathbb{P}_2\right) = \frac{(m_1-m_2)^2+(s_1)^2-(s_2)^2}{2(s_2)^2}+\log\left(\frac{s_2}{s_1}\right),$$

where $\mathbb{P}_1 = \mathcal{LN}\big(m_1,(s_1)^2\big)\big)$ and $\mathbb{P}_2 = \mathcal{LN}\big(m_2,(s_2)^2\big)$.

Considering these elements, we specify the loss function as follows[4]:

$$L\left(\theta_T,\hat{\theta}_T\right) = \sum_{k=1}^{p} \underbrace{\left(\frac{\left(\tilde{\mathrm{m}}\big(\hat{\theta}_{T+\delta_k}\big)-\tilde{\mathrm{m}}\big(\theta_{T+\delta_k}\big)\right)^2+\tilde{\mathrm{s}}\big(\hat{\theta}_{T+\delta_k}\big)^2-\tilde{\mathrm{s}}\big(\theta_{T+\delta_k}\big)^2}{2\tilde{\mathrm{s}}\big(\theta_{T+\delta_k}\big)^2}+\log\left(\frac{\tilde{\mathrm{s}}\big(\theta_{T+\delta_k}\big)}{\tilde{\mathrm{s}}\big(\hat{\theta}_{T+\delta_k}\big)}\right)\right)}_{\mathcal{D}_{KL}\left(\tilde{\mathbb{P}}_i,\mathbb{P}_i\right)},$$

with $\mathbb{P}_i = \mathcal{LN}\Big(\tilde{\mathrm{m}}\big(\theta_{T+\delta_k}\big),\tilde{\mathrm{s}}\big(\theta_{T+\delta_k}\big)^2\Big)$ and $\hat{\mathbb{P}}_i = \mathcal{LN}\Big(\tilde{\mathrm{m}}\big(\hat{\theta}_{T+\delta_k}\big),\tilde{\mathrm{s}}\big(\hat{\theta}_{T+\delta_k}\big)^2\Big)$.

Therefore, the loss function L can be understood as the summation of estimated KL divergences between the predicted volatility distribution and the true volatility distribution at various time horizons. It quantifies the discrepancy between these distributions. It is worth noting that although the log-normal distribution is an approximation of the distribution of $\sigma_{T+\delta}$ given θ_T, the KL divergence $\mathcal{D}_{KL}\left(\hat{\mathbb{P}}_i,\mathbb{P}_i\right)$ achieves its minimum value (which is 0) when $\mathbb{E}\big[\sigma_{T+\delta}|\theta_T\big] = \mathbb{E}\big[\sigma_{T+\delta}|\hat{\theta}_T\big]$ and $\mathbb{V}\mathrm{ar}\big[\sigma_{T+\delta}|\theta_T\big] = \mathbb{V}\mathrm{ar}\big[\sigma_{T+\delta}|\hat{\theta}_T\big]$.

[4]Note that L is a loss function of the form 8.

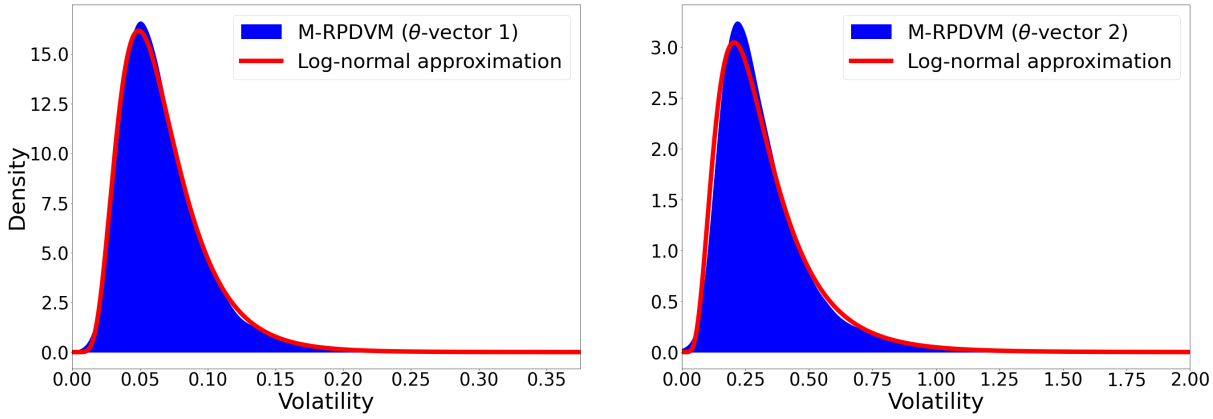

Figure 1: Example of two volatility distributions at a one-week horizon generated from the M-RPDV model using two different θ-vectors and their respective log-normal approximations. The KL-divergence between the two associated distributions is equal to 4.66, while the KL-divergence between the log-normal approximations of these distributions is equal to 3.92.

3 Construction of the Bayesian estimator function

Section 2.2 outlined the estimation problem we seek to solve, defined by the optimization program 10. However, two main obstacles need to be overcome: firstly, an analytical formula for the function M is not available, and secondly, updating the prior distribution $\boldsymbol{\pi}$ to obtain the posterior distribution $\boldsymbol{\pi}_{\mathrm{D}}$ is a highly complex task. To address the first issue, we will adopt a strategy similar to that of Horvath *et al.* (2021), which compensates for the lack of an analytical formula for the option price by using "a neural network that maps parameters of a stochastic model to pricing functions" in their calibration process. In a comparable fashion, we introduce in section 3.1 an estimator for the function M in the form of an NN that maps (θ_T, δ) to the conditional mean and standard deviation $\mathbb{E}\big[\sigma_{T+\delta}|\theta_T\big]$ and $\mathbb{S}\mathrm{td}\big[\sigma_{T+\delta}|\theta_T\big]$. We will use this proxy of M to calibrate a second NN that will play the role of a Bayesian estimator function. Therefore, the objective of this second NN, the architecture of which will be detailed in section 3.2, is to provide, for any D, a proxy for the Bayesian estimator of θ_T, i.e., an approximate solution to the optimization program 10. Section 3.3 will detail an estimation procedure followed to achieve this situation, a method that circumvents the challenge of estimating the posterior distribution $\boldsymbol{\pi}_{\mathrm{D}}$ directly by indirectly addressing the original problem, solving a related problem under the prior measure $\boldsymbol{\pi}$.

3.1 The neural network as a proxy for function M

The initial stage in constructing the estimator function involves developing an estimator for the function M, denoted as $\mathcal{M}$. This estimator maps (θ, δ) to the conditional moments $\mathbb{E}\big[\sigma_{T+\delta}|\theta_T\big]$ and $\mathbb{S}\mathrm{td}\big[\sigma_{T+\delta}|\theta_T\big]$. The objective is for $\mathcal{M}$ to approximate M for all $\theta : \boldsymbol{\pi}(\theta_T) \neq 0$ and $\delta \in \delta_1, ..., \delta_p$, as expressed by the approximation

$$\mathcal{M}\left(\theta_T, \delta\right) \approx M\left(\theta_T, \delta\right) \quad \text{and} \quad \nabla\mathcal{M}\left(\theta_T, \delta\right) \approx \nabla M\left(\theta_T, \delta\right). \tag{12}$$

The objective is for $\mathcal{M}$ not only to be a good approximation of the function M, but also for the gradient of $\mathcal{M}$ to be approximately equal to the gradient of M for all $\theta : \boldsymbol{\pi}(\theta_T) \neq 0$ and $\delta \in \delta_1, ..., \delta_p$. This property $(\nabla\mathcal{M}(\theta_T, \delta) \approx \nabla M(\theta_T, \delta))$ is crucial in the role that $\mathcal{M}$ will play in learning the estimator of the θ-vector. To achieve this, $\mathcal{M}$ will be implemented as a neural network (NN) with a specialized and tailored architecture designed for this task. In this section, we will provide a detailed description of the adopted network architecture.

3.1.1 General structure of the network

As mentioned earlier, the architecture of $\mathcal{M}$ is specifically adapted to ensure that the gradient passed to the estimator neural network contains informative information about the implications of the chosen θ-vector in moment predictions. This choice is motivated by the observation that a more standard network structure often results in a fitted NN that is primarily sensitive to state vectors R_1 and R_2, thus missing the main purpose of $\mathcal{M}$. In order to address this issue and satisfy the requirements described in equations 12, $\mathcal{M}$ adopts an architecture schematized in figure 2.

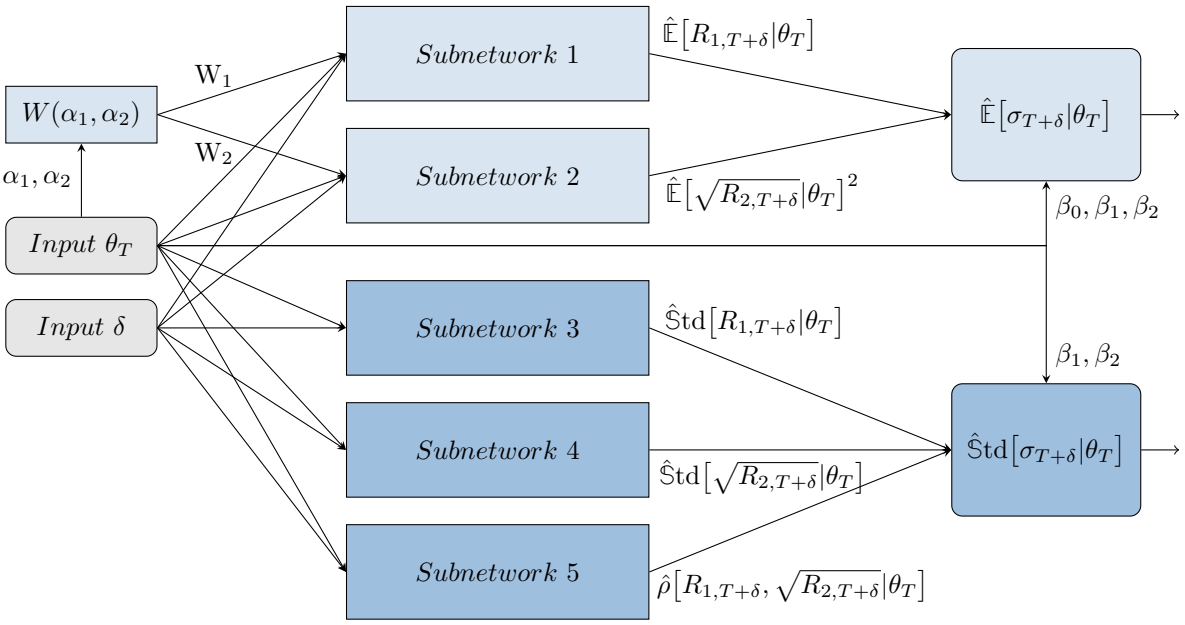

Figure 2: The architecture of the neural network $\mathcal{M}$.

Firstly, the neural network (NN) consists of 2 distinct input layers: the first layer receives the θ-vectors and thus has $9 + 2n$ input neurons, while the second layer has a dimension of 1 to specify the temporal horizon δ for which the conditional moments are to be computed. These 2 input layers feed several subregions of the network, which can be segmented into 2 main components. The first component is responsible for estimating $\mathbb{E}[\sigma_{T+\delta}|\theta_T]$, while the second component is tasked with estimating $\mathbb{S}\mathrm{td}[\sigma_{T+\delta}|\theta_T]$.

3.1.2 The part of the network responsible for estimating $\mathbb{E}[\sigma_{T+\delta}|\theta_T]$

The part of the network responsible for predicting $\mathbb{E}[\sigma_{T+\delta}|\theta_T]$ utilizes the fact that this expectation can be expressed as follows:

$$\mathbb{E}[\sigma_{T+\delta}|\theta_T] = \beta_0 + \beta_1 \mathbb{E}\big[R_{1,T+\delta}|\theta_T\big] + \beta_2 \mathbb{E}\big[\sqrt{R_{2,T+\delta}}|\theta_T\big] \tag{13}$$

The approach is to estimate $\mathbb{E}\big[R_{1,T+\delta}|\theta_T\big]$ and $\mathbb{E}\big[\sqrt{R_{2,T+\delta}}|\theta_T\big]$ $\Big(\text{more precisely, } \mathbb{E}\big[\sqrt{R_{2,T+\delta}}|\theta_T\big]^2\Big)$ separately using two parallel subnetworks: subnetwork 1 and subnetwork 2 as depicted in figure 2. Equation 13 is then used to calculate an estimation of $\mathbb{E}[\sigma_{T+\delta}|\theta_T]$. These subnetworks each consist of 6 hidden layers, with the first

5 layers containing 100 ReLU neurons each, and the final layer consisting of 20 linear neurons. The last hidden layer connects to an output layer with a single neuron (each subnetwork has its own output neuron/layer), which receives input from the two input layers associated with θ_T and δ, as well as from a function that computes the vectors W_1 and W_2 from θ_T.

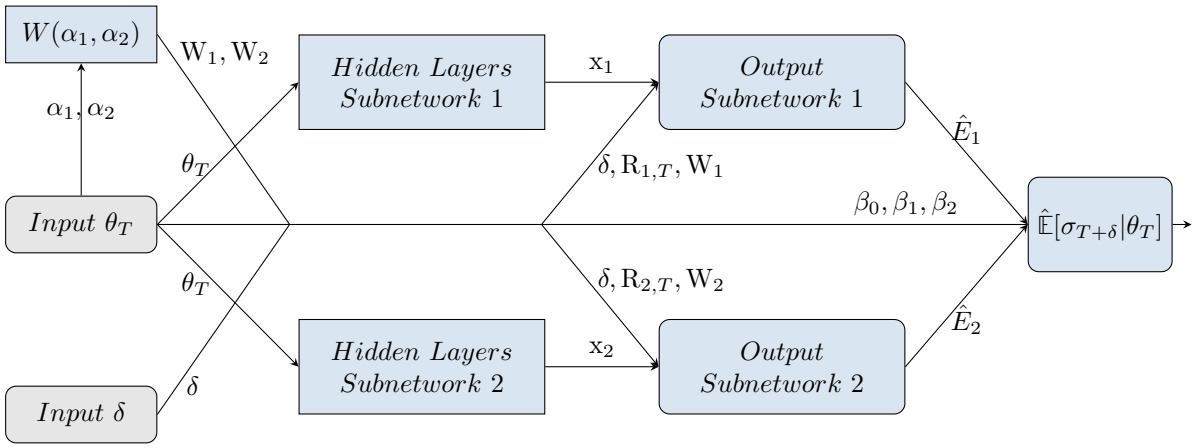

Figure 3: The part of the network $\mathcal{M}$ responsible for estimating $\mathbb{E}[\sigma_{T+\delta}|\theta_T]$.

The activation function associated with this output neuron is of the form:

$$\hat{E}_j\big(\delta, \mathrm{R}_{j,T}, \mathrm{W}_j, \mathrm{x}_j\big) = \underbrace{\sum_{i=1}^{n} w_{j,i} e^{-\gamma_i \delta} \mathrm{R}_{j,i,T}}_{(1)} + \underbrace{\sum_{k=1}^{20} \big(1 - e^{-g_k \delta}\big)\, \mathrm{x}_{j,k}}_{(2)}, \tag{14}$$

where x_j is the output vector of the last hidden layer associated with subnetwork j. The exponential weights $\big(g_k\big)_{1 \le k \le 20}$ are defined as follows:

$$g_k = \exp\left(\log\left(\frac{1}{1000}\right) + \frac{\log(100) - \log\left(\frac{1}{1000}\right)}{20 - 1}(k-1)\right)^{-1}. \tag{15}$$

The specific form of this activation function is particularly suited to its objective in view of the analytical expressions of $\mathbb{E}\big[R_{1,T+\delta}|\theta_T\big]$ and $\mathbb{E}\big[\sqrt{R_{2,T+\delta}}|\theta_T\big]$. First of all, when $\delta = 0$,

$$\hat{E}_1\big(0, \mathrm{R}_{1,t}, \mathrm{W}_1, \mathrm{x}_1\big) = R_{1,t} \quad \text{and} \quad \hat{E}_2\big(0, \mathrm{R}_{2,t}, \mathrm{W}_2, \mathrm{x}_2\big) = R_{2,t}.$$

Thus, vectors x_j have no impact on the calculation since we use the analytical formulas for $R_{1,t}$ and $R_{2,t}$ which are θ-measurable. On the other hand, when $\delta = \infty$, the term (1) in equation 13 becomes zero, and we have

$$\hat{E}_1\big(\infty, \mathrm{R}_{1,t}, \mathrm{W}_1, \mathrm{x}_1\big) = \sum_{k=1}^{20} \mathrm{x}_{1,k} \quad \text{and} \quad \hat{E}_2\big(\infty, \mathrm{R}_{2,t}, \mathrm{W}_2, \mathrm{x}_2\big) = \sum_{k=1}^{20} \mathrm{x}_{2,k},$$

which can be respectively interpreted as the estimated asymptotic value of the expectation of R_1 and the

squared estimated asymptotic value of the expectation of $\sqrt{R_2}$. Besides these polar cases, component (2) aims to estimate clearly identified variables. For the network responsible for estimating $\mathbb{E}\big[R_{1,T+\delta}|\theta_T\big]$, the objective is that for all δ:

$$\begin{aligned}\sum_{k=1}^{20}\left(1-e^{-g_k\delta}\right)\mathrm{x}_{1,k} \approx\ & \mathbb{E}[R_{1,T+\delta}|\theta_T] - \sum_{i=1}^{n}\gamma_{1,i}w_{1,i}e^{-\gamma_i\delta}\mathrm{R}_{1,i,T} \\ &= \mathbb{E}\left[\int_T^{T+\delta}\sum_{i=1}^{n}\gamma_{1,i}w_{1,i}e^{-\gamma_{1,i}(T+\delta-u)}\left(\frac{\mathrm{d}P_u}{P_u}-\kappa_1\mathrm{R}_{1,i,u}\mathrm{d}u\right)\Big|\theta_T\right] \\ &= \mathbb{E}\left[\int_T^{T+\delta}\hat{K}_1(T+\delta-u)\left(\frac{\mathrm{d}P_u}{P_u}-\kappa_1 R_{1,u}\mathrm{d}u\right)\Big|\theta_T\right].\end{aligned}$$

For the network responsible for estimating $\mathbb{E}\big[\sqrt{R_{2,T+\delta}}|\theta_T\big]$, the objective is that for all δ[5]:

$$\begin{aligned}\sum_{k=1}^{20}\left(1-e^{-g_k\delta}\right)\mathrm{x}_{2,k} \approx\ & \mathbb{E}\big[R_{2,T+\delta}|\theta\big] - \mathbb{V}\mathrm{ar}\big[\sqrt{R_{2,T+\delta}}|\theta\big] - \sum_{i=1}^{n}\gamma_{2,i}w_{2,i}e^{-\gamma_i\delta}\mathrm{R}_{2,i,T} \\ &= \mathbb{E}\left[\int_T^{T+\delta}\sum_{i=1}^{n}\gamma_{2,i}w_{2,i}e^{-\gamma_{2,i}(T+\delta-u)}\left(\sigma_u^2\mathrm{d}u-\kappa_2\mathrm{R}_{2,i,u}\mathrm{d}u\right)\Big|\theta_T\right] - \mathbb{V}\mathrm{ar}\big[\sqrt{R_{2,T+\delta}}|\theta_T\big]. \\ &= \mathbb{E}\left[\int_T^{T+\delta}\hat{K}_2(T+\delta-u)\left(\sigma_u^2\mathrm{d}u-\kappa_2\mathrm{R}_{2,i,u}\mathrm{d}u\right)\Big|\theta_T\right] - \mathbb{V}\mathrm{ar}\big[\sqrt{R_{2,T+\delta}}|\theta_T\big].\end{aligned}$$

These output layers of the two sub-networks thus produce an estimation of $\mathbb{E}\big[R_{1,T+\delta}|\theta_T\big]$ and $\mathbb{E}\big[\sqrt{R_{2,T+\delta}}|\theta_T\big]$, respectively, which feed into a global output neuron of the network $\mathcal{M}$: the neuron whose output is the estimator $\hat{\mathbb{E}}[\sigma_{T+\delta}|\theta_T]$. This neuron, also fed by the input layer associated with θ_T, then computes the estimator of $\mathbb{E}[\sigma_{T+\delta}|\theta_T]$ using the analytical formula 13:

$$\hat{\mathbb{E}}[\sigma_{t+\delta t}|\theta] = \beta_0 + \beta_1\hat{E}_1 + \beta_2\sqrt{\left(\hat{E}_2\right)_+}, \tag{16}$$

with $\hat{E}_1$ the output of the sub-network 1 and $\hat{E}_2$ the output of the sub-network 2.

3.1.3 The part of the network responsible for estimating $\mathbb{S}\mathrm{td}[\sigma_{T+\delta}|\theta_T]$

The part of the network responsible for predicting $\mathbb{S}\mathrm{td}\big[R_{1,T+\delta}|\theta_T\big], \mathbb{S}\mathrm{td}[\sigma_{T+\delta}|\theta_T]$ uses the fact that the variance is equal to (see appendix D.3):

$$\begin{aligned}\mathbb{V}\mathrm{ar}[\sigma_{T+\delta}|\theta_T] = (\beta_1\mathbb{S}\mathrm{td}[R_{1,T+\delta}|\theta_T])^2 &+ \left(\beta_2\mathbb{S}\mathrm{td}\big[\sqrt{R_{2,T+\delta}}\big|\theta_T\big]\right)^2 \\ &+ 2\beta_1\beta_2\mathbb{S}\mathrm{td}[R_{1,T+\delta}|\theta_T]\mathbb{S}\mathrm{td}\big[\sqrt{R_{2,T+\delta}}\big|\theta_T\big]\rho\big[R_{1,T+\delta},\sqrt{R_{2,T+\delta}}\big|\theta_T\big],\end{aligned} \tag{17}$$

where $\rho\big[R_{1,T+\delta},\sqrt{R_{2,T+\delta}}\big|\theta_T\big]$ is the correlation between $R_{1,T+\delta}$ and $\sqrt{R_{2,T+\delta}}$ given θ_T. Similar to the branch of $\mathcal{M}$ responsible for estimating $\mathbb{E}[\sigma_{T+\delta}|\theta_T]$, the approach is to estimate $\mathbb{S}\mathrm{td}[R_{1,T+\delta}|\theta_T], \mathbb{S}\mathrm{td}\big[\sqrt{R_{2,T+\delta}}\big|\theta_T\big]$, and $\rho\big[R_{1,T+\delta},\sqrt{R_{2,T+\delta}}\big|\theta_T\big]$ using 3 parallel sub-networks assigned to each of these components. These 3 sub-networks are each composed of 6 hidden layers, with the first layer being fed by the input layer θ_T. The first 5 layers of each sub-network consist of 100 ReLU neurons. The last hidden layer is composed of linear neurons, with 3 neurons for sub-networks 3 and 4, which are responsible for estimating $\mathbb{S}\mathrm{td}[R_{1,T+\delta}|\theta_T]$

[5]It arises from the relationship:

$$\mathbb{V}\mathrm{ar}\Big[\sqrt{R_{2,T+\delta}}|\theta_T\Big] = \mathbb{E}\big[R_{2,T+\delta}|\theta_T\big] - \mathbb{E}\Big[\sqrt{R_{2,T+\delta}}|\theta_T\Big]^2 \Leftrightarrow \mathbb{E}\Big[\sqrt{R_{2,T+\delta}}|\theta_T\Big]^2 = \mathbb{E}\big[R_{2,T+\delta}|\theta_T\big] - \mathbb{V}\mathrm{ar}\Big[\sqrt{R_{2,T+\delta}}|\theta_T\Big].$$

and $\mathbb{S}\mathrm{td}\big[\sqrt{R_{2,T+\delta}}\big|\theta_T\big]$, respectively, and 21 neurons for the 5th sub-network responsible for estimating the correlation $\rho\big[R_{1,T+\delta}, \sqrt{R_{2,T+\delta}}\big|\theta_T\big]$. The last hidden layer of each sub-network feeds an output neuron, which also receives input from the δ layer.

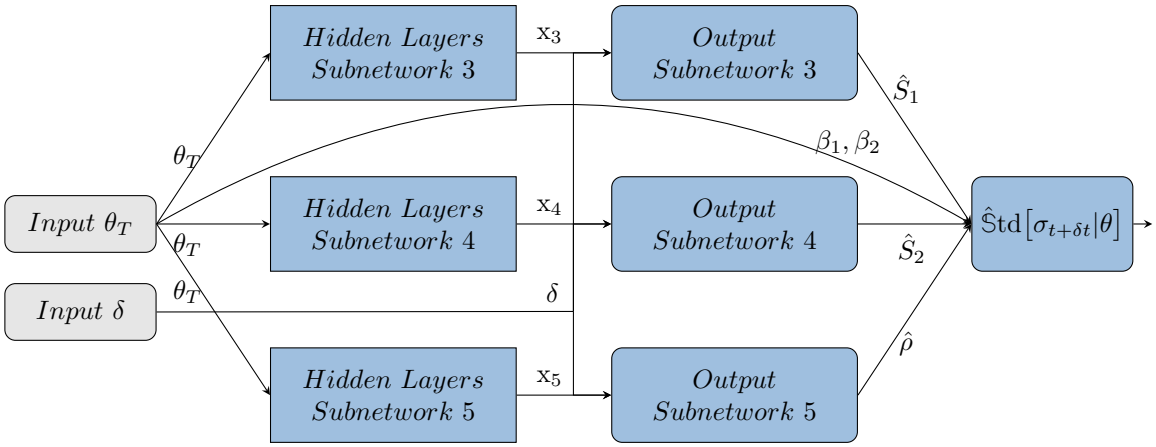

Figure 4: The part of the network $\mathcal{M}$ responsible for estimating $\mathbb{S}\mathrm{td}[\sigma_{t+\delta}|\theta_T]$.

For sub-networks 3 and 4, the output neuron is associated with an activation function of the following form:

$$\hat{S}_j(\delta, \mathrm{x}_j) = \exp\left(\mathrm{x}_{j,1} + \max(\mathrm{x}_{j,2}, \epsilon) \cdot \log\left(\frac{\delta}{1 + (\mathrm{x}_{j,3})_+ \cdot \delta}\right)\right)$$

where ϵ is a positive constant close to zero. The choice of this activation function is motivated by the fact that, due to the properties of the RPDV model, the logarithms of the respective standard deviations of R_1 and $\sqrt{R_2}$ for the considered time horizons approximately follow a relationship of the form:

$$\hat{\mathbb{S}}\mathrm{td}\big[(R_{j,T+\delta})^{1/j}|\theta_T\big] \approx \exp\big(a + b \cdot \log(\delta)\big),$$

where b is positive. In the chosen activation function, $\log(\delta)$ is replaced with $\log\big(\delta/(1+\mathrm{x}_{j,3}\cdot\delta)\big)$ to potentially capture concavity of the relationship for certain θ-vectors. Additionally, it is also interesting to note that this function ensures that the estimated standard deviations are zero when $\delta = 0$, consistent with the fact that $\mathrm{R}_{1,T}$ and $\mathrm{R}_{2,T}$ are θ-measurable.

For subnetwork 5 responsible for estimating the correlation between $R_{1,T+\delta}$ and $\sqrt{R_{2,T+\delta}}$ given θ_T, the output neuron is associated with the following activation function:

$$\hat{\rho}(\delta, y) = \min\left(2;\ \mathrm{x}_{5,1} + \sum_{k=2}^{21}\left(1 - e^{-g_k\delta}\right)\mathrm{x}_{5,k}\right) - 1.$$

This activation function ensures that the output range is limited to the interval $[-1, 1]$, which is suitable for estimating a correlation. Furthermore, the instantaneous correlation and the asymptotic correlation between

R_1 and $\sqrt{R_2}$ conditioned on θ_T are given by:

$$\hat{\rho}(0, \mathrm{x}_5) = \min\left(2;\ \mathrm{x}_{5,1}\right) - 1 \quad \text{and} \quad \hat{\rho}(\infty, \mathrm{x}_5) = \min\left(2;\ \mathrm{x}_{5,1} + \sum_{k=2}^{21} \mathrm{x}_{5,k}\right) - 1.$$

Each output layer of subnetworks 3, 4, and 5 feeds the output neuron of $\mathcal{M}$ responsible for estimating $\mathbb{S}\mathrm{td}[\sigma_{T+\delta}|\theta_T]$. This output neuron, also fed by the input layer θ_T, is associated with the following activation function using the analytical expression 17:

$$\hat{\mathbb{S}}\mathrm{td}[\sigma_{T+\delta}|\theta_T] = \sqrt{\left(\beta_1\hat{S}_1\right)^2 + \left(\beta_2\hat{S}_2\right)^2 + 2\beta_1\beta_2\hat{S}_1\hat{S}_2\hat{\rho}},$$

with $\hat{S}1, \hat{S}2$ and $\hat{\rho}$ being the respective outputs of sub-networks 1, 2, and 3.

3.2 The estimator function

As mentioned in the introduction of section 3, the purpose of the function $\mathcal{M}$ defined in section 3.1 is to assist in the training of a second neural network, the estimator function Θ, which is responsible for estimating the Bayes estimator of θ_T from a data matrix D under the posterior measure $\boldsymbol{\pi}_{\mathrm{D}}$. The objective is to construct an estimator function Θ that satisfies the following criterion:

$$\mathbb{E}_{\boldsymbol{\pi}_{\mathrm{D}}}\Big[L\big(\boldsymbol{\theta}_T, \Theta(\mathrm{D})\big)\Big] \approx \min_{\hat{\theta}_T}\ \mathbb{E}_{\boldsymbol{\pi}_{\mathrm{D}}}\Big[L\big(\boldsymbol{\theta}_T, \hat{\theta}_T\big)\Big], \quad \forall\ \mathrm{D} : \boldsymbol{\pi}(\mathrm{D}) \neq 0. \tag{18}$$

Therefore, the architecture of Θ should be designed to extract all relevant information contained in D in order to achieve the stated objective. To this end, Θ take the following general form:

$$\Theta\Big(\mathrm{D}\Big) = \mathcal{NN}\Big(\mathcal{E}\big(\mathrm{D}\big)\Big),$$

where $\mathcal{NN}$ is a neural network and $\mathcal{E}$ is a time-series encoder generally defined by:

$$\mathcal{E}\big(\mathrm{D}\big) = \begin{pmatrix} z_1 \\ ... \\ z_{n_\mathcal{E}} \end{pmatrix}, \tag{19}$$

with $n_\mathcal{E}$ the number of features extract by $\mathcal{E}$. The estimator function first encodes with $\mathcal{E}$ the raw data matrix D into a feature vector that is used as input for a neural network $\mathcal{NN}$ to predict the θ-vector. The sections 3.2.1 and 3.2.2 provide a detailed description of the structure of these two components that form Θ.

3.2.1 Dual encoder structure: combining non-trainable and trainable methods

The role of the encoder $\mathcal{E}$ is to extract informative features from D for estimating θ_T. Given the variety of methods available for encoding time series, $\mathcal{E}$ can take different forms. These methods, found in the academic literature, include transforming time series into pattern variables (Kimoto et al., 1990; Usmani *et al.*, 2016), imaging time series (Wang and Oates, 2015; Barra *et al.*, 2020), or using signature methods (Morill *et al.*, 2020). Depending on the chosen method, the encoding can be predetermined, meaning that features are extracted using a fixed method determined in advance, or it can be learned during the training process, allowing the encoder to adapt to the specific data characteristics. In this work, the encoder $\mathcal{E}$ combines both approaches by incorporating a non-trainable component and a trainable component. The aim is to leverage prior knowledge of the process by extracting informative metrics using the non-trainable component, while complementing them with the trainable component of the encoder.

3.2.1.1 The non-trainable encoder

The non-trainable component of the encoder $\mathcal{E}$, denoted as $\mathcal{E}_1$, is a pre-determined method that extracts informative metrics from D for determining the θ-vector. Specifically, $\mathcal{E}_1$ computes the following features from D:

- The serial correlation of the log realized volatility for the following lag times expressed in trading days: 1, 2, 3, 4, 5, 10, 20, 60, 125, 252.
- The mean of the absolute value of realized log-volatility increments over the following time intervals in trading days: 1, 2, 3, 4, 5, 10, 20, 60, 125, 252.
- The first four moments of the distribution of returns and realized volatility for the time horizons of 1, 5, 21, 63, 252 expressed in trading days.
- The 20 percentiles of the distribution of returns and of the realized volatility for the time horizons of 1, 5, 21, 63, 252 expressed in trading days.
- The linear regression coefficient between the volatility increments and the returns for the following lag times expressed in trading days: 1, 2, 3, 4, 5, 10, 20, 60, 125, 252.
- The standardized exponential moving averages of returns, realized volatility, and realized variance, which are defined respectively as

$$m_{1,j} = \frac{\sum_{i=1}^{N} r_{t_i} e^{(t_i - t) g_j}}{\sum_{i=1}^{N} e^{(t_i - t) g_j}}, \quad m_{2,j} = \frac{\sum_{i=1}^{N} \tilde{\sigma}_{t_i} e^{(t_i - t) g_j}}{\sum_{i=1}^{N} e^{(t_i - t) g_j}}, \quad m_{3,j} = \frac{\sum_{i=1}^{N} \tilde{\sigma}_{t_i}^2 e^{(t_i - t) g_j}}{\sum_{i=1}^{N} e^{(t_i - t) g_j}},$$

  where $g_j \in \big(g_k\big)_{1 \leq k \leq 20}$ and their values are defined in equation 15 (section 3.1.2).

The metrics computed by $\mathcal{E}_1$ are diverse, allowing for a multifaceted approach to the data in D. These metrics, along with those from the trainable component of the encoder $\mathcal{E}$, will be used as inputs to the $\mathcal{NN}$ network.

3.2.1.2 The trainable encoder

The trainable component of $\mathcal{E}$, denoted as $\mathcal{E}_2$, aims to complement the metrics calculated by $\mathcal{E}_1$, adopting a more agnostic approach. It consists of a convolutional neural network (CNN) with a structure similar to that of a multi-scale CNN (MCNN) proposed by Cui *et al.* (2016). The input layer of $\mathcal{E}_2$ takes the raw data matrix D as input and feeds it into four branches. The first layer of each branch is associated with a function defined as:

$$\mathcal{A}\big(\mathrm{D}, l\big) = \begin{pmatrix} \frac{P_{t_{1+l}}}{P_{t_1}} & \tilde{\sigma}_{[t_1:t_{1+l}]} & r_{[t_1:t_{1+l}]} & \tilde{\sigma}_{t_{1+l}} - \tilde{\sigma}_{t_1} \\ \dots & \dots & & \\ \frac{P_{t_N}}{P_{t_1}} & \tilde{\sigma}_{[t_1:t_{1+l}]} & r_{[t_{N-l}:t_N]} & \tilde{\sigma}_{t_N} - \tilde{\sigma}_{t_{N-l}} \end{pmatrix}, \tag{20}$$

where

$$\tilde{\sigma}_{[t_i:t_{i+l}]} = \sqrt{\frac{1}{l} \sum_{k=1}^{l} \tilde{\sigma}_{t_{i+k}}^2} \quad \text{and} \quad r_{[t_i:t_{i+l}]} = \frac{P_{t_{i+l}} - P_{t_i}}{P_{t_i}}.$$

This function transforms the original $N \times 2$ matrix into a $(N - l) \times 4$ matrix. The first column represents the normalized price with respect to the date t_1. The second column corresponds to the integrated volatility over a time window of l trading days, while the third column denotes the asset return over the same time window.

The fourth column captures the variation of volatility over the l-day period. For branches 1, 2, 3, and 4, the time window parameter l is fixed at 1, 5, 21, and 63, respectively. This choice allows for the augmentation of the original matrix D with two additional informative columns and captures important information at different time scales using multiple l values. Each augmented matrix is processed by a convolutional layer followed by an average pooling layer for each of the four branches. Each convolutional layer consists of 50 filters of size 5×4, with a stride of 1. The pooling layers perform global average pooling for each filtered time series. The outputs of the four pooling layers associated with the branches of $\mathcal{E}_2$ are finally flattened and concatenated with the output of $\mathcal{E}_1$. This combined output is then used as input to the network $\mathcal{NN}$.

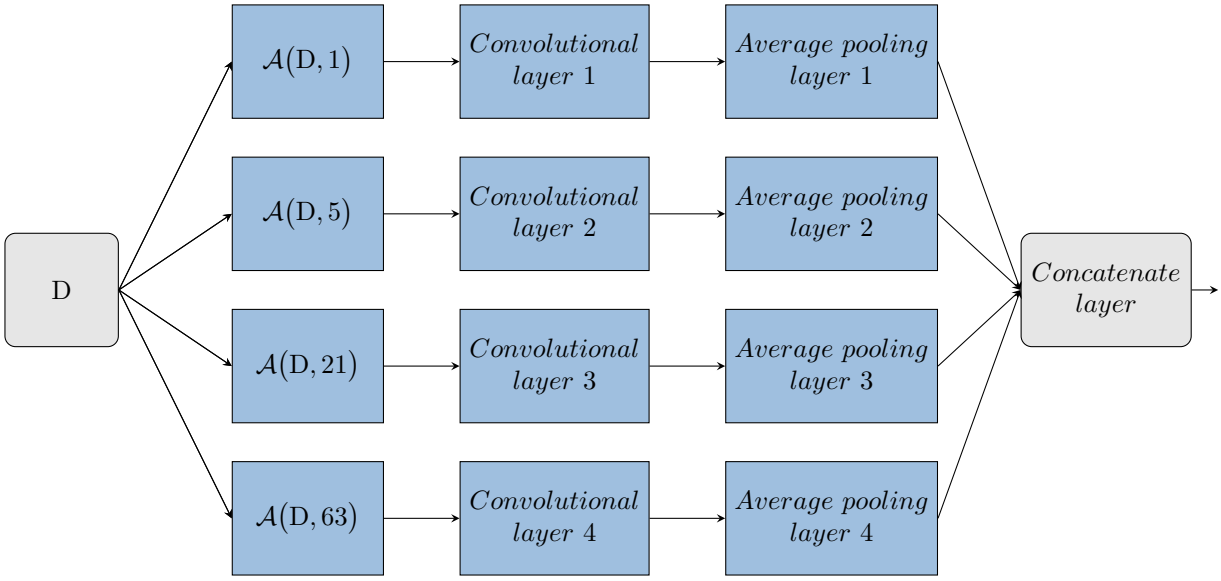

Figure 5: The architecture of the encoder $\mathcal{E}_2$.

3.2.2 The network $\mathcal{NN}$: from encoded data to θ-vector

The $\mathcal{NN}$ network is responsible for predicting the Bayesian estimator of the θ-vector using the feature vector provided by the encoder $\mathcal{E}$. To accomplish this, $\mathcal{NN}$ proceeds sequentially by first estimating the 9 parameters of the model and then, in a second step, estimating the state variables. These two operations are performed by two separate multilayer perceptrons (MLPs): $\mathcal{NN}_1$, which is responsible for estimating ϕ, and $\mathcal{NN}_2$, which is responsible for estimating R_T.

To begin with, the input layer of $\mathcal{NN}_1$ is fed by the output of $\mathcal{E}$, which results in $n_{\mathcal{E}}$ input neurons. This is followed by 6 ReLU layers, each with 100 neurons. The last ReLU layer feeds into the output layer, which consists of 9 neurons that correspond to the parameters to be predicted. The output neuron responsible for estimating β_1 is associated with an inverted ReLU activation function, while the other output neurons are associated with a standard ReLU. The output layer of $\mathcal{NN}_1$ is then fed into the network $\mathcal{NN}_2$, as well as into the output layer of $\mathcal{NN}$ (and therefore the output layer of Θ), where it is concatenated with the output of the $\mathcal{NN}_2$ network.

The MLP $\mathcal{NN}_2$ receives as input both the output of $\mathcal{E}$ and the output of $\mathcal{NN}_1$, which results in $n_{\mathcal{E}} + 9$ input

neurons. Like the input layer of $\mathcal{NN}_1$, the input layer of $\mathcal{NN}_2$ is followed by 6 ReLU layers, each with 100 neurons. Finally, the output layer of $\mathcal{NN}_1$ consists of $2n$ linear neurons, representing the $2n$ state variables. This output layer feeds into the output layer of $\mathcal{NN}$, where it is concatenated with the output of the $\mathcal{NN}_2$ network.

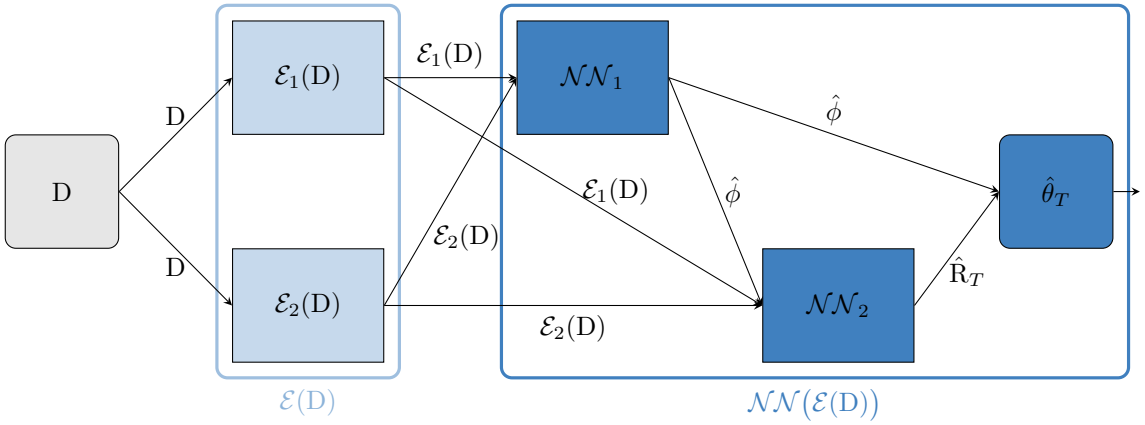

Figure 6: The architecture of the estimator function Θ.

3.3 The estimation procedure

This section presents an estimation approach for the RPDV model, utilizing the functions $\mathcal{M}$ and Θ introduced in sections 3.1 and 3.2, respectively. This method can be divided into three phases: the generation of training data, the training of $\mathcal{M}$, and the training of Θ. These three steps are subsequently described in this section. At the end of this procedure, Θ can be used as an estimator function for the θ-vectors.

3.3.1 Generation of initial data

To calibrate the functions $\mathcal{M}$ and Θ, it is necessary to generate training sets. This process involves defining the method for generating the θ-vectors, which is equivalent to establishing the prior measure $\boldsymbol{\pi}$. We will then outline the different steps involved in constructing the training sets for $\mathcal{M}$ and Θ.

3.3.1.1 Defining the prior measure $\boldsymbol{\pi}$

The estimation procedure requires defining the prior measure $\boldsymbol{\pi}$. The specification of this measure incorporates prior knowledge about the parameters, even if this knowledge is vague. By constraining the parameter space, it is likely to increase the estimation quality without overly restricting the range of possible values. The idea is therefore to propose a generation procedure for parameter θ-vectors that exploits prior knowledge about price and volatility dynamics, without excessively constraining the parameter space.

In this context, certain coordinates of the random vector $\boldsymbol{\theta}_T$ are assumed to be independent random variables, while others exhibit a correlation structure. Specifically, the parameters $\beta_0, \alpha_1, \alpha_2, \kappa_1, \kappa_2$ are distributed independently, as follows:

$$(\beta_0, \alpha_1, \alpha_2, \kappa_1, \kappa_2) \sim \mathcal{U}(0, 0.25) \times \mathcal{U}(0, 1) \times \mathcal{U}(0, 1) \times \mathcal{U}(0, 5) \times \mathcal{U}(0, 5) \times \mathcal{U}(0, 0.15).$$

Regarding the risk premia, their generation is slightly more complex. Firstly, the value of the price drift $\lambda_1\sigma_t + \lambda_2(\sigma_t)^2$ is generated when the volatility is equal to 15% ($\sigma_t = 0.15$) as follows:

$$\bar{\mu} \sim \mathcal{U}(0, 0.1).$$

In other words, under the measure $\boldsymbol{\pi}$, the price drift ranges from 0 to 10% when the volatility level is 15%. Then, there is a $\frac{1}{3}$ probability that $\lambda_1 = \frac{\bar{\mu}}{0.15}$ and $\lambda_2 = 0$, a $\frac{1}{3}$ probability that $\lambda_1 = 0$ and $\lambda_2 = \frac{\bar{\mu}}{0.15^2}$, and a $\frac{1}{3}$ probability that

$$\lambda_1 \sim \mathcal{U}(0,1) \cdot \frac{\bar{\mu}}{0.15}, \quad \lambda_2 = \frac{\bar{\mu} - 0.15\lambda_1}{0.15^2}.$$

Therefore, there is an equal probability of having a pure volatility premium, a pure variance premium, or a mixture of both. The last two remaining parameters are β_1 and β_2. In both cases, it makes sense to consider the specificity of the kernel associated with the variables for which β_1 and β_2 determine the volatility sensitivity. With respect to β_1, it is generated as follows:

$$\beta_1 \sim \mathcal{U}(-1.5, 0) \cdot \left(\sum_{i=1}^{n} \sum_{k=1}^{n} \frac{w_{1,i} w_{1,k} \gamma_{1,i} \gamma_{1,k}}{\gamma_{1,i} + \gamma_{1,k}} e^{-(\gamma_{1,i} + \gamma_{1,k})} \right)^{-0.5}.$$

The term that weights $\mathcal{U}(-1.5, 0)$ is the inverse of the asymptotic standard deviation of the BSS process associated with the kernel $\hat{K}_1$ (see appendix D.2). This weighting allows for the generation of reasonable values of β_1 given α_1. Similarly, the parameter β_2 is generated as follows:

$$\beta_2 \sim \frac{\mathcal{U}(0,1)}{\sum_{i=1}^{n} w_{2,i}}.$$

The term that weights $\mathcal{U}(0,1)$ corresponds to the inverse of the integral over $\mathbb{R}_+$ of the kernel $\hat{K}_2$ (see appendix D.1). This weighting prevents volatility from exploding regardless of the value of κ_2.

Next, we define how the state variables $\{(\mathrm{R}_1)_j\}_{j=1}^n$ and $\{(\mathrm{R}_2)_j\}_{j=1}^n$ are generated. The approach consists of three steps. The first step is the initialization of the state variables. The state variables $\{(\mathrm{R}_1)_j\}_{j=1}^n$ are simply initialized to zero, and the state variables $\{(\mathrm{R}_2)_j\}_{j=1}^n$ are initialized as follows:

$$(\mathrm{R}_{2,0})_i \sim (\beta_0)^2 \cdot \mathcal{U}(0.9, 1.1).$$

From these initial values and the associated parameter vector, a simulation of the volatility dynamics is performed over a period of 5 years. Finally, the values of the state variables at the end of this simulation are retained.

The θ-vector is retained if and only if the initial value of the volatility is positive and lower than 300%, i.e., if $0 < \beta_0 + \beta_1 R_{1,T} + \beta_2 \sqrt{R_{2,T}} < 3$.

3.3.1.2 Constructing training sets

The objective is to generate data samples

$$\left\{ \mathrm{D}, \theta_T, \left\{ \bar{E}_{1,T+\delta_k}, \bar{E}_{2,T+\delta_k}, \bar{S}_{1,T+\delta_k}, \bar{S}_{2,T+\delta_k}, \bar{\rho}_{T+\delta_k} \right\}_{k=1}^{p} \right\},$$

where D is a data matrix of the form 5 (section 2.2.1) generated from the generator 4 (section 2.1.2), θ_T is the value taken by the θ-vector at the end of the simulation of D, and where $\bar{E}_{k,T+\delta_j}, \bar{S}_{1,T+\delta_k}, \bar{\rho}_{T+\delta_k}$ are unbiased

estimators of $\mathbb{E}[R_{k,T+\delta}|\theta_T], \mathbb{S}\text{td}[R_{k,T+\delta}|\theta_T]$ and $\rho\big[R_{1,T+\delta}, \sqrt{R_{2,T+\delta}}|\theta_T\big]$, respectively. To do this, we use the following algorithm.

Algorithm 1 Procedure for generating training sets.

Require: $\boldsymbol{\pi}, n_1, n_2$

1. Generate an i.i.d. sample $\Omega_{\theta_0} = \left\{\theta_{t_0}^{(i)}\right\}_{1\leq i\leq n_1}$ from the distribution $\boldsymbol{\pi}$.

2. Generate from Ω_{θ_0} and model 4 the pairs $\left\{\mathrm{D}^{(i)}, \theta_T^{(i)}\right\}_{1\leq i\leq n_1}$.

3. Generate n_2 time series over the periods $T+\delta_1, ..., T+\delta_p$ using model 4 for each $\theta \in \left\{\theta_T^{(i)}\right\}_{1\leq i\leq n_1}$, and extract from each series the sets $\Omega_R = \left\{ \left\{R_{1,T+\delta_k}^{(i,k)}, R_{2,T+\delta_k}^{(i,j)}\right\}_{k=1}^{p} \right\}_{\substack{1\leq i\leq n_1 \\ 1\leq j\leq n_2}}$.

4. Compute set $\left\{\left\{\bar{E}_{1,T+\delta_k}^{(i)}, \bar{E}_{2,T+\delta_k}^{(i)}, \bar{S}_{1,T+\delta_k}^{(i)}, \bar{S}_{2,T+\delta_k}^{(i)}, \bar{\rho}_{T+\delta_k}^{(i)}\right\}_{k=1}^{p}\right\}_{1\leq i\leq n_1}$ from sample set Ω_R.

return $\left\{\mathrm{D}^{(i)}, \theta_T^{(i)}, \left\{\bar{E}_{1,T+\delta_k}^{(i)}, \bar{E}_{2,T+\delta_k}^{(i)}, \bar{S}_{1,T+\delta_k}^{(i)}, \bar{S}_{2,T+\delta_k}^{(i)}, \bar{\rho}_{T+\delta_k}^{(i)}\right\}_{1\leq k\leq p}\right\}_{1\leq i\leq n_1}$

This data generation procedure aligns with the adopted Bayesian approach. The set of i.i.d. matrices $\left\{\mathrm{D}^{(i)}\right\}_{1\leq i\leq n_1}$ is generated from initial vectors $\left\{\theta_{t_0}^{(i)}\right\}_{1\leq i\leq n_1}$ sampled from the prior distribution $\boldsymbol{\pi}$, which incorporates vague knowledge about the model parameters. In this article, we specify $n_1 = 200\ 000$ and $n_2 = 200$. The large value chosen for n_1 ensures good coverage of the parameter and state variable space. As for n_2, its value allows for empirical moment estimators with reasonable variance on average.

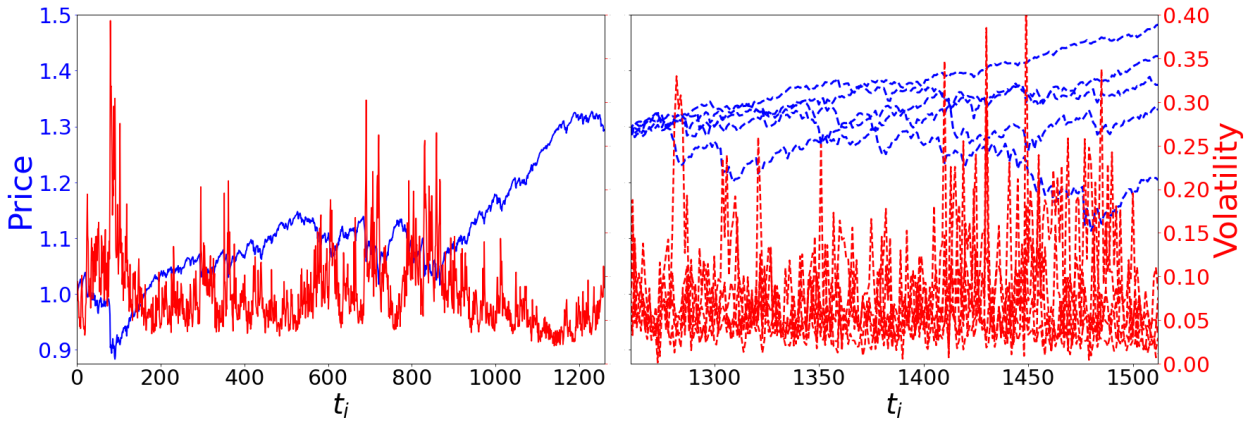

Figure 7: The left figure plots a historical trajectory of realized joint price and volatility of an asset over 1260 trading days contained in a matrix $\mathrm{D}^{(i)}$, generated from the initial θ-vector value $\theta_0^{(i)}$. The right figure represents 5 continuations of this trajectory over 252 trading days generated from $\theta_{1260}^{(i)}$.

3.3.2 The training of $\mathcal{M}$

The second phase of the general procedure involves training $\mathcal{M}$ using the datasets generated in section 3.3.1. The method to be proposed for this purpose is derived from the following proposition proved in appendix C.1.

Proposition 1 *Let be* $\theta_T^{(1)}, ..., \theta_T^{(n_1)}$ *a sequence of i.i.d. random variable following* $\boldsymbol{\pi}$*, and* $\left\{\bar{M}_{T+\delta_k}^{(1)}\right\}_{1\leq j\leq p}, ...,$ $\left\{\bar{M}_{T+\delta_k}^{(n_1)}\right\}_{1\leq j\leq p}$ *a sequence of sets such as* $\forall\ i,k, \bar{M}_{T+\delta_k}^{(i)}$ *is an unbiased estimator of* $M\left(\theta_T^{(i)}, \delta_k\right)$ *calculated from a sample of size* n_2*. If it exists* $\mathcal{M}^\star$*, such as* $\mathcal{M}^\star\left(\theta_T, \delta_k\right) = M\left(\theta_T, \delta_k\right),\ \forall\ \theta_T : \boldsymbol{\pi}\left(\theta_T\right) \neq 0$ *and* $\delta_k \in \{\delta_1, ..., \delta_p\}$*, thus* $\forall\ \hat{\mathcal{M}}^\star$ *solution to*

$$\arg\min_{\mathcal{M}}\ \frac{1}{n_1}\sum_{i=1}^{n_1}\sum_{k=1}^{p}\left\|\mathcal{M}\left(\theta_T^{(i)}, \delta_k\right) - \bar{M}_{T+\delta_k}^{(i)}\right\|_2^2,$$

$\hat{\mathcal{M}}^\star\left(\theta, \delta_k\right) = M\left(\theta_T, \delta_k\right), \forall\ \theta_T : \boldsymbol{\pi}\left(\theta_T\right) \neq 0$ *and* $\delta_k \in \{\delta_1, ..., \delta_p\}$.

Therefore, this proposition implies a way to make $\mathcal{M}$ a convergent estimator of M, under suitable conditions of existence, for the time horizons $\delta_1, ..., \delta_p$ and the θ-vectors associated with a non-zero probability under the measure $\boldsymbol{\pi}$. It simply involves minimizing the mean squared difference between the estimators returned by $\mathcal{M}$ and the unbiased estimator $\bar{M}$ for each pair $\left(\delta^{(i)}, \theta_T^{(i)}\right)$ in the training set. This minimization constitutes the second step of algorithm 2, which we propose for training $\mathcal{M}$.

Algorithm 2 Training procedure for $\mathcal{M}$

Require: $\left\{\theta_T^{(i)}, \left\{\bar{E}_{1,T+\delta_j}^{(i)}, \bar{E}_{2,T+\delta_j}^{(i)}, \bar{S}_{1,T+\delta_j}^{(i)}, \bar{S}_{2,T+\delta_j}^{(i)}, \bar{\rho}_{T+\delta_j}^{(i)}\right\}_{1\leq k\leq p}\right\}_{1\leq i\leq n_1}$

1. Optimize

$$\arg\min_{\mathcal{M}}\ \frac{1}{n_1}\sum_{i=1}^{n_1}\sum_{k=1}^{p}\left(\hat{\rho}_{T+\delta_k}^{(i)} - \bar{\rho}_{T+\delta_k}^{(i)}\right)^2 + \sum_{j=1}^{2}\left(\left(\hat{E}_{j,T+\delta_k}^{(i)} - \bar{E}_{j,T+\delta_k}^{(i)}\right)^2 + \left(\hat{S}_{j,T+\delta_k}^{(i)} - \bar{S}_{j,T+\delta_k}^{(i)}\right)^2\right).$$

2. Optimize

$$\arg\min_{\mathcal{M}}\ \frac{1}{n_1}\sum_{i=1}^{n_1}\sum_{k=1}^{p}\left\|\mathcal{M}\left(\theta_T^{(i)}, \delta_k\right) - \bar{M}_{T+\delta_k}^{(i)}\right\|_2^2,$$

starting from the training of $\mathcal{M}$ obtained at the end of step 1.

return $\hat{\mathcal{M}}^\star$

The first step of the procedure serves only to prepare for the second (and final) step of the training of $\mathcal{M}$. It involves independently training the five subnetworks that compose $\mathcal{M}$ (see figure 2). The objective by the end of step 1 is as follows:

$$\hat{E}_{j,T+\delta_k}^{(i)} \approx \mathbb{E}\left[\left(R_{j,T+\delta_k}\right)^{1/p}\middle|\theta_T^{(i)}\right]^p, \quad \hat{S}_{j,T+\delta_k}^{(i)} \approx \mathbb{S}\mathrm{td}\left[R_{j,T+\delta_k}\middle|\theta_T^{(i)}\right] \quad \text{and} \quad \hat{\rho}_{T+\delta_k}^{(i)} \approx \rho\left[R_{1,T+\delta_k}, \sqrt{R_{2,T+\delta_k}}\middle|\theta_T^{(i)}\right],$$

$\forall\ \{i,k\}$. In this first phase, the training of $\mathcal{M}$ is conducted to align with the specific role of each of the 5 subnetworks within it. This first phase, which frames the outputs of the subnetworks in $\mathcal{M}$, is followed by step 2, which is aimed at achieving the objective 12 stated in the introduction of section 3.1. As previously mentioned, according to proposition 3.3.2, under suitable conditions of existence, as n_1 and n_2 approach

infinity, solving the optimization problem associated with step 2 involves finding $\hat{\mathcal{M}}^\star$ such that:

$$\hat{\mathcal{M}}^\star(\theta, \delta_k) = M(\theta_T, \delta_k), \forall\ \theta_T : \boldsymbol{\pi}(\theta_T) \neq 0 \ \text{ and } \ \delta_k \in \{\delta_1, ..., \delta_p\}.$$

The assumption of the existence of $\mathcal{M}^\star$ is related to the flexibility of the network $\mathcal{M}$ in approximating the function M. Moreover, the reasonableness of this assumption is guaranteed by the structure of the subnetworks that constitute $\mathcal{M}$, which can approximate any continuous function due to the universal approximation theorem (Hornik *et al.* 1989).

3.3.3 The training of Θ

The third and final step of the estimation process involves training Θ to achieve the desired situation described in 18 at the beginning of section 3.2. The consistency of the method to be proposed in this section for this purpose stems from the following proposition demonstrated in appendix C.2.

Proposition 2 *Let $\theta_{t_0}^{(1,1)}, ..., \theta_{t_0}^{(n_1,1)}$ be a sequence of i.i.d. random variables following $\boldsymbol{\pi}$, $\mathrm{D}^{(1)}, ..., \mathrm{D}^{(n_1)}$ a set of time-series such that $\mathrm{D}^{(i)}$ is generated from the M-RPDV associated with the θ-vector $\theta_{t_0}^{(i)}$, and $\theta_T^{(1,2)}, ..., \theta_T^{(n_1,2)}$ the set of values taken by θ at time t_N for each time series $\mathrm{D}^{(i)}$. If there exists $\Theta^\star$ such that for all $\mathrm{D} : \mathbb{P}_{\boldsymbol{\pi}}(\mathrm{D}) \neq 0$[6], $\Theta^\star(\mathrm{D})$ is a Bayes estimator of θ_T under the posterior measure $\boldsymbol{\pi}$, then for any $\hat{\Theta}^\star$ solution to the optimization problem*

$$\arg\min_{\Theta} \lim_{n_1 \to +\infty} \frac{1}{n_1} \sum_{i=1}^{n_1} L\left(\theta_T^{(1,i)}, \Theta\left(\mathrm{D}^{(i)}\right)\right),$$

$\hat{\Theta}^\star(\mathrm{D})$ is a Bayes estimator of θ_T under the posterior measure $\boldsymbol{\pi}_{\mathrm{D}}, \forall\ \mathrm{D} : \mathbb{P}_{\boldsymbol{\pi}}(\mathrm{D}) \neq 0$.

This proposition, therefore, has significant implications as it provides a way to calibrate Θ, such that asymptotically and under a suitable existence condition, $\Theta(\mathrm{D})$ becomes a Bayesian estimator of θ_T under the posterior measure π_{D}, for all $\mathrm{D} : \mathbb{P}_{\boldsymbol{\pi}}(\mathrm{D}) \neq 0$. This result is even more remarkable considering that the estimation method does not require explicit calculation of the posterior measures (i.e., the measure $\boldsymbol{\pi}$ updated by a matrix D) at any point. Indeed, it simply involves following steps 1 and 2 of algorithm 4.1.1) to generate a set of pairs $\left\{\mathrm{D}^{(i)}, \theta_T^{(i)}\right\}_{1 \leq i \leq n_1}$, and then minimizing the average losses measured by L between the θ-vectors predicted by Θ (i.e. $\mathrm{D}^{(i)}$) and the true θ-vectors (i.e. $\theta_T^{(i)}$). However, since L depends on the function M, which is not known, we use the following proxy that replaces M with $\mathcal{M}$:

$$\hat{L}\left(\theta_T^{(i)}, \Theta(\mathrm{D}^{(i)})\right) = \sum_{k=1}^{p} C\left(\mathcal{M}\left(\theta_T^{(i)}, \delta_k\right), \mathcal{M}\left(\Theta\left(\mathrm{D}^{(i)}\right), \delta_k\right)\right).$$

The idea is that after training $\mathcal{M}$ (algorithm 2), $\mathcal{M}(\theta_T, \delta) \approx M(\theta_T, \delta)$ and $\nabla\mathcal{M}(\theta_T, \delta) \approx \nabla M(\theta_T, \delta)$, and therefore:

$$\hat{L}\left(\theta_T^{(i)}, \Theta(\mathrm{D}^{(i)})\right) \approx L\left(\theta_T^{(i)}, \Theta(\mathrm{D}^{(i)})\right) \quad \text{and} \quad \nabla\hat{L}\left(\theta_T^{(i)}, \Theta(\mathrm{D}^{(i)})\right) \approx \nabla L\left(\theta_T^{(i)}, \Theta(\mathrm{D}^{(i)})\right).$$

Consequently, the quality of the approximation of the function M by $\mathcal{M}$ is crucial in the training of Θ. Regarding the assumption of the existence of $\Theta^\star$ on which proposition 3.3.3 relies, its reasonableness depends on both the encoder's ability to extract all relevant information contained in the time series D and the plasticity of the networks $\mathcal{NN}_1$ and $\mathcal{NN}_2$.

[6] $\mathbb{P}_{\boldsymbol{\pi}}$ denotes the distribution of $\mathbf{D}$ induced by $\boldsymbol{\pi}$.

Based on these elements, the training of Θ is carried out using the following algorithm.

Algorithm 3 Learning procedure for the estimator function Θ

Require: $\left\{\mathrm{D}^{(i)}, \theta_T^{(i)}\right\}_{1\leq i\leq n_1}$

1. Optimize

$$\underset{\mathcal{E}_2,\ \mathcal{NN}_1}{\arg\min}\ \frac{1}{n_1}\sum_{i=1}^{n_1}\left\|\mathcal{NN}_1\Big(\mathcal{E}\Big(\mathrm{D}^{(i)}\Big)\Big)-\phi^{(i)}\right\|_2^2.$$

2. Optimize

$$\underset{\mathcal{NN}_2}{\arg\min}\ \frac{1}{n_1}\sum_{i=1}^{n_1}\left(\mathcal{NN}_2\Big(\mathcal{E}\Big(\mathrm{D}^{(i)}\Big),\phi^{(i)}\Big)-\mathrm{R}_T^{(i)}\right)^2.$$

3. Optimize

$$\underset{\mathcal{NN}_2}{\arg\min}\ \frac{1}{n_1}\sum_{i=1}^{n_1}\left(\mathcal{M}\Big(\Big(\phi^{(i)},\mathcal{NN}_2\Big(\mathrm{D}^{(i)}\Big)\Big),0\Big)-\mathcal{M}\Big(\theta^{(i)},0\Big)\right)^2.$$

4. Optimize

$$\underset{\Theta}{\arg\min}\ \frac{1}{n_1}\sum_{i=1}^{n_1}\sum_{k=1}^{p}C\left(\mathcal{M}\left(\theta_T^{(i)},\delta_k\right),\mathcal{M}\left(\Theta\left(\mathrm{D}^{(i)}\right),\delta_k\right)\right).$$

return $\hat{\Theta}^\star$

The first three learning steps for sub-regions of the network that constitutes Θ in practice serve only to prepare for the fourth and final step, which is directly derived from proposition 3.3.3. In the first phase, only the components of the network responsible for predicting the parameter vector ϕ, i.e., the trainable encoder $\mathcal{E}_2$ and the neural network $\mathcal{NN}_1$, are trained. The aim is to guide the learning of Θ initially by minimizing the sum of squared differences between predicted and actual parameter vectors, thereby obtaining parameters consistent with the prior measure $\boldsymbol{\pi}$. Next, the neural network $\mathcal{NN}_2$ responsible for predicting the state variable vector R is trained. The calibration of $\mathcal{NN}_2$ uses the encoder $\mathcal{E}$ fitted during phase 1 and actual parameter vectors, rather than those estimated by $\mathcal{NN}_1$. This approach allows for a more focused learning of the relationship between data and the vector of state variables to be predicted, without introducing any bias caused by estimation errors in $\mathcal{NN}_1$. The first three steps of the algorithm described previously are not important in themselves, but serve only to prepare for the final learning step of Θ. This final step aims to achieve the objective defined in the introduction of section 3.2, namely that Θ returns an estimator of θ_T that is close (in terms of expected loss measured by L) to its Bayesian estimator under the posterior measure. The consistency of the optimization program solved in this step with this objective is established by the following proposition proved in the appendix C.2.

4 Assessment of the estimation method

The purpose of this section is to evaluate the estimator function defined in section 3. For this purpose, we perform various tests, starting with synthetic data and then moving on to market data.

4.1 Evaluation of estimation method using synthetic data

In this section, the objective is to evaluate the estimation method presented in section 3 using synthetic data. We start by assessing the accuracy of the moment estimator $\mathcal{M}$ in approximating the function M. Next, we evaluate the effectiveness of the estimator function Θ in providing consistent estimates of the θ-vector that align with our forecasting objectives.

4.1.1 Test dataset and evaluation metrics

The test dataset is generated using algorithm , as introduced in section 3.3.1.2, with parameters $n_1 = 10000$ and $n_2 = 1000$. Hence, we have the following elements that will be used to construct the test datasets:

$$\left\{ \mathrm{D}^{(i)}, \theta_T^{(i)}, \left\{ \bar{M}_{T+\delta_k}^{(i)} \right\}_{1 \leq k \leq p} \right\}_{1 \leq i \leq 10000}.$$

The choice of $n_2 = 1000$ in Algorithm 1 allows us to consider $\bar{M}_{T+\delta_k}^{(i)}$ as reasonably accurate estimators of $M\big(\theta_T^{(i)}, \delta\big)$. This consistency enables us to use them as targeted values for comparison with the predicted values generated by $\mathcal{M}\big(\theta_T^{(i)}, \delta\big)$ and $\mathcal{M}\big(\mathrm{D}^{(i)}, \delta\big)$.

The evaluation of the estimated conditional moments $\hat{y}_i$ with the targeted conditional moment values y_i will be conducted using the following metrics:

- The root mean squared error(RMSE)

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n_1} \big(y_i - \hat{y}_i\big)^2}{N}}.$$

- The mean absolute error (MAE)

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n_1} \big|y_i - \hat{y}_i\big|}{N}.$$

- The mean absolute percentage error (MAPE)

$$\mathrm{MAPE} = \frac{1}{n_1} \frac{\sum_{i=1}^{n_1} \big|y_i - \hat{y}_i\big|}{y_i}$$

- The coefficient of determination

$$r^2 = 1 - \frac{\sum_{i=1}^{n_1} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n_1} \left(y_i - \frac{1}{n_1} \sum_{i=1} y_i\right)^2}.$$

The use of these different metrics allows for evaluating, from different angles, the moment estimators. RMSE is a classic metric that measures the difference between predicted and actual values, while taking into account the variance of errors. MAE, on the other hand, measures the average of absolute errors, providing an indication of the overall accuracy of the estimation. MAPE has the advantage of comparing the performance of the estimation, taking into account the heterogeneity of the magnitudes of the moments. Finally, the coefficient of determination r^2 measures the overall adequacy of the model by providing an indication of the proportion of variance explained by the model.

4.1.2 Evaluation of the ability of $\mathcal{M}$ to approximate M

We aim to evaluate how closely $\mathcal{M}$ approximates M. To do so, we compare the estimators

$$\mathcal{M}(\theta_T, \delta)_1 = \hat{\mathbb{E}}\big[\sigma_{T+\delta}|\theta_T\big] \quad \text{and} \quad \mathcal{M}(\theta_T, \delta)_2 = \hat{\mathbb{S}}\text{td}\big[\sigma_{T+\delta}|\theta_T\big],$$

with the corresponding empirical estimators calculated from a sample of volatility trajectories using θ_T. This comparison is performed using the test dataset

$$\left\{ \left\{ \theta_T^{(i)}, \bar{M}_{T+\delta_k}^{(i)} \right\}_{1 \leq i \leq 10000} \right\}_{1 \leq k \leq 5},$$

which is extracted from the synthetic data defined in section 4.1.1. The results obtained are reported in tables 1 and 2.

	$\delta = 1$	$\delta = 5$	$\delta = 21$	$\delta = 42$	$\delta = 63$
RMSE	0.0055	0.0069	0.0095	0.0090	0.0100
MAE	0.0029	0.0035	0.0043	0.0045	0.0049
MAPE	0.0270	0.0325	0.0391	0.0421	0.0453
R-Squared	0.9958	0.9920	0.9832	0.9842	0.9800

Table 1: Evaluation metrics for the estimation of $\mathbb{E}[\sigma_{T+\delta}|\theta_T]$ by $\mathcal{M}$.

	$\delta = 1$	$\delta = 5$	$\delta = 21$	$\delta = 42$	$\delta = 63$
RMSE	0.0258	0.0279	0.0297	0.0310	0.0353
MAE	0.0107	0.0105	0.0122	0.0131	0.0141
MAPE	0.1081	0.0965	0.1046	0.1102	0.1143
R-Squared	0.9010	0.8887	0.8742	0.8675	0.8359

Table 2: Evaluation metrics for the estimation of $\mathbb{S}\text{td}[\sigma_{T+\delta}|\theta_T]$ by $\mathcal{M}$.

The obtained results demonstrate that $\mathcal{M}$ provides a reliable approximation of M across different time horizons. When estimating $\mathbb{E}[\sigma_{T+\delta}|\theta_T]$, the evaluation metrics indicate a significant agreement with the empirical estimators in terms of both absolute and relative deviation. For example, the MAE falls within the range of 0.0029 to 0.0049 for various time horizons δ, indicating that, on average, the estimated values of $\mathbb{E}[\sigma_{T+\delta}|\theta_T]$ provided by $\mathcal{M}$ deviate from the empirical estimator by less than 0.005 units. Additionally, the MAPE metric reveals that the average absolute deviation between these two estimators ranges from 2.7% to 4.5% in relative terms, which is notably low. Furthermore, the consistently high R-squared values exceeding 98% confirm the excellent quality of approximation for the conditional expectation by $\mathcal{M}$.

The same observation applies to the estimation of the conditional standard deviations $\mathbb{S}\text{td}[\sigma_{T+\delta}|\theta_T]$, albeit with some nuances. The evaluation metrics demonstrate a significant agreement between $\mathcal{M}$ and the empirical estimators, indicating a reliable approximation of $\mathbb{S}\text{td}[\sigma_{T+\delta}|\theta_T]$ for the different time horizons. However, it is worth noting that the deviation between these estimators is significantly larger compared to the estimation of $\mathbb{E}[\sigma_{T+\delta t}|\theta_T]$. Specifically, we observe that the MAE ranges from 0.0107 to 0.0141 for different values of δ, suggesting that, on average, the estimated values provided by $\mathcal{M}$ deviate from the empirical estimator by less than 0.015 units. In terms of the MAPE, the average absolute deviation between the estimators ranges from 9.65% to 11.43% in relative terms. Moreover, the R-squared values for $\mathbb{S}\text{td}[\sigma_{T+\delta}|\theta_T]$ range from 83.59% to 90.10%, indicating a relatively high concordance between $\mathcal{M}$ and the empirical estimators, but lower compared to the estimation of $\mathbb{E}[\sigma_{T+\delta}|\theta_T]$.

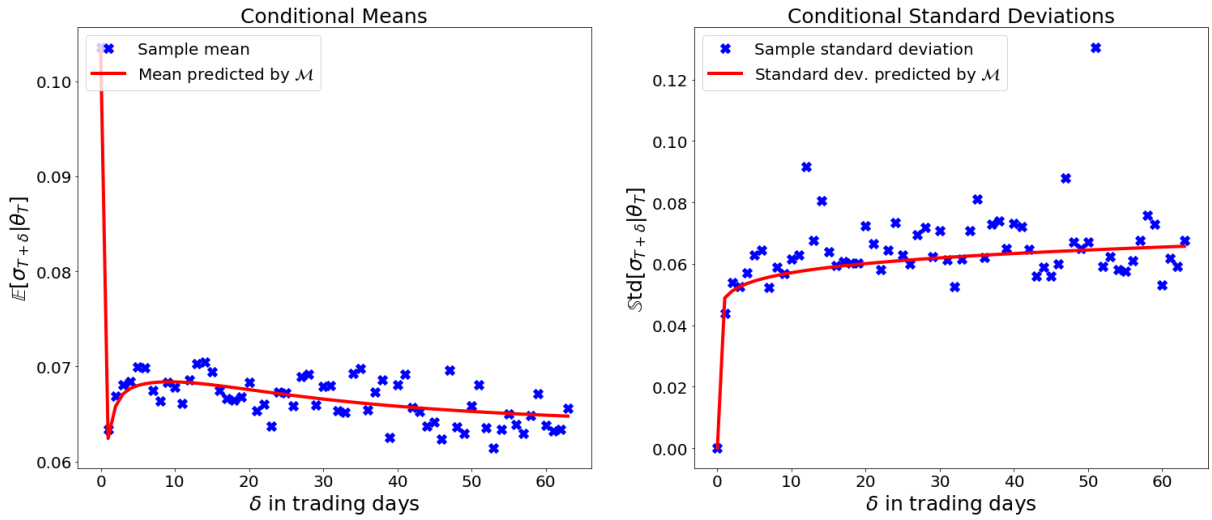

Figure 8: Example of estimated conditional means and standard deviations by $\mathcal{M}$ compared to sample conditional estimators.

However, this discrepancy does not necessarily imply a lower quality of estimation for $\mathbb{S}\text{td}[\sigma_{T+\delta t}|\theta_T]$. In fact, it can be mainly attributed to the higher variance of the empirical estimator used for estimating the standard deviations. This higher variance is illustrated by the example exhibited in Figure 8, where the absolute percentage error for the 51 trading days horizon is greater than 50%. However, by replicating the experiment with 20 000 simulations, this absolute error decreases significantly to less than 2%. This suggests that the observed residuals are primarily due to the higher variance of the empirical estimators used rather than a misestimation of $\mathcal{M}$.

4.1.3 Assessment of the estimator function Θ

4.1.3.1 Evaluation based on conditional moments

To evaluate the performance of the estimator function Θ, we compare in this section the following estimators:

$$\mathcal{M}\big(\Theta\big(\mathrm{D}^{(i)}\big),\delta\big)_1 = \hat{\mathbb{E}}\big[\sigma_{t+\delta}|\mathrm{D}^{(i)}\big] \quad \text{and} \quad \mathcal{M}\big(\Theta\big(\mathrm{D}^{(i)}\big),\delta\big)_2 = \hat{\mathbb{S}}\text{td}\big[\sigma_{t+\delta}|\mathrm{D}^{(i)}\big],$$

using two types of moment estimators: the empirical estimators already used in section 4.1.2, and the estimators computed from $\mathcal{M}$ using the actual θ-vectors.

To begin the evaluation, we use the following test dataset, where the targeted values are the sample moments:

$$\left\{\left\{\mathrm{D}^{(i)}, \bar{M}^{(i)}_{T+\delta_k}\right\}_{1\leq i\leq 10000}\right\}_{1\leq k\leq 5}.$$

The results obtained from this evaluation are presented in tables 3 and 4.

	$\delta = 1$	$\delta = 5$	$\delta = 21$	$\delta = 42$	$\delta = 63$
RMSE	0.0147	0.0129	0.0128	0.0122	0.0129
MAE	0.0083	0.0076	0.0074	0.0075	0.0077
MAPE	0.0904	0.0799	0.0753	0.0747	0.0761
R-Squared	0.9705	0.9725	0.9696	0.9705	0.9662

Table 3: Evaluation metrics comparing $\mathcal{M}\big(\Theta\big(\mathrm{D}\big),\delta\big)_1$ and $\big(\bar{M}_{T+\delta_k}\big)_1$.

	$\delta = 1$	$\delta = 5$	$\delta = 21$	$\delta = 42$	$\delta = 63$
RMSE	0.0355	0.0368	0.0373	0.0400	0.0476
MAE	0.0156	0.0155	0.0165	0.0175	0.0185
MAPE	0.1857	0.1655	0.1610	0.1634	0.1655
R-Squared	0.8007	0.7944	0.7891	0.7688	0.7048

Table 4: Evaluation metrics comparing $\mathcal{M}\big(\Theta\big(\mathrm{D}\big), \delta\big)_2$ and $\big(\bar{M}_{T+\delta_k}\big)_2$.

A first observation is that, consistently, the cost metrics are higher and the R-squared values are lower compared to the case studied in section 4.1.2, where the θ-vectors are known. However, the observed difference, although significant, remains relatively moderate, which suggests the quality of the estimation of the θ-vectors produced by Θ. However, to better interpret these results, it is important to note that this difference tends to decrease relative to the temporal horizon. Thus, while the MAPE between $\mathcal{M}\big(\theta_T, \delta\big)_1$ and $\big(\bar{M}_{T+\delta}\big)_1$ increases from 2.7% for a 1-day trading horizon to 4.5% for a 3-month horizon (63 trading days), the MAPE between $\mathcal{M}\big(\Theta\big(\mathrm{D}^{(i)}\big), \delta\big)_1$ and $\big(\bar{M}_{T+\delta}\big)_1$ decreases for the same periods from 9% to 7.6%. Similarly, the MAPE between $\mathcal{M}\big(\theta_T, \delta\big)_2$ and $\big(\bar{M}_{T+\delta}\big)_2$ decreases from 10.8% for a 1-day trading horizon to 11.4% for a 3-month horizon, compared to a decrease from 18.6% to 16.5% for the same horizons when comparing $\mathcal{M}\big(\theta_T, \delta\big)_2$ and $\big(\bar{M}_{T+\delta}\big)_2$. A first explanation for this phenomenon could be the decreasing significance of short-term information contained in the state variables associated with higher discount factors, as it has a diminishing impact on the conditional moments. However, these state variables are particularly challenging to estimate due to the daily observation frequency, which explains the reduction in the cost gaps between $\mathcal{M}\big(\Theta\big(\mathrm{D}\big), \delta\big)$ and $\mathcal{M}\big(\theta_T, \delta\big)$ as δ increases. This phenomenon may also be partly caused by the variance of the empirical estimator $\bar{M}_{T+\delta}$. To isolate the impact of estimating the θ-vectors using Θ, it is valuable to directly compare the estimator $\mathcal{M}\big(\Theta\big(\mathrm{D}\big), \delta\big)$ with the estimator $\mathcal{M}\big(\theta_T, \delta\big)$. To investigate this further, we employ the same evaluation procedure for Θ as discussed previously, but on the following test dataset:

$$\left\{ \left\{ \mathrm{D}^{(i)}, \mathcal{M}\Big(\theta_T^{(i)}, \delta_k\Big) \right\}_{1 \leq i \leq 10{,}000} \right\}_{1 \leq k \leq 5}.$$

The resulting outcomes from this evaluation are presented in the following tables.

	$\delta = 1$	$\delta = 5$	$\delta = 21$	$\delta = 42$	$\delta = 63$
RMSE	0.0128	0.0114	0.0113	0.0115	0.0118
MAE	0.0078	0.0072	0.0072	0.0074	0.0077
MAPE	0.0851	0.0763	0.0757	0.0797	0.0840
R-Squared	0.9744	0.9767	0.9750	0.9729	0.9711

Table 5: Evaluation metrics comparing $\mathcal{M}\big(\Theta\big(\mathrm{D}\big), \delta\big)_1$ and $\mathcal{M}\big(\theta_T, \delta\big)_1$.

	$\delta = 1$	$\delta = 5$	$\delta = 21$	$\delta = 42$	$\delta = 63$
RMSE	0.0209	0.0214	0.0222	0.0229	0.0235
MAE	0.0108	0.0111	0.0116	0.0121	0.0125
MAPE	0.1603	0.1492	0.1431	0.1421	0.1423
R-Squared	0.9031	0.9034	0.9029	0.9014	0.8997

Table 6: Evaluation metrics comparing $\mathcal{M}\big(\Theta\big(\mathrm{D}\big), \delta\big)_2$ and $\mathcal{M}\big(\theta_T, \delta\big)_2$.

In a consistent manner, the majority of cost metrics demonstrate lower values, and the R-squared value consistently shows higher values when using $\mathcal{M}\big(\theta_T, \delta\big)$ as targeted values instead of empirical moment estimators.

Although the difference is relatively small for conditional expectation estimators, it becomes more significant for conditional standard deviation estimators. Specifically, the R-squared is 10 to 20 points lower when using $\mathcal{M}\big(\theta_T, \delta\big)_2$ as the targeted value compared to the sample standard deviation. This suggests that a significant portion of the discrepancy between the estimator $\mathcal{M}\big(\theta_T, \delta\big)_2$ and the empirical estimator of the conditional standard deviation can be attributed to the variance of the latter. In practical terms, this finding further strengthens the notion that the future volatility distributions associated with the θ-vectors estimated by Θ closely align with the actual future volatility distributions, not only in terms of the mean but also in terms of the standard deviation.

4.1.3.2 Evaluation based on conditional distributions using the Kolmogorov-Smirnov test

In addition to evaluating the estimator function through conditional moments, it is important to examine the consistency between the estimated θ-vectors and the true θ-vectors in terms of the associated conditional distributions. To address this aspect, we conduct a statistical experiment to assess the adequacy of the estimated θ-vectors by Θ.

The first step of this experiment involves estimating each θ-vector associated with each matrix $\mathrm{D}^{(i)}$ using the Θ method. Subsequently, we generate 100 simulations for each estimated vector, consideringhe the p time horizons of interest to us: 1, 5, 21, 42, and 63 trading days. For each combination of $\left(\theta_T^{(i)}, \hat{\theta}_T^{(i)}\right)$ and for each time horizon, we employ the Kolmogorov-Smirnov (KS) test to calculate the p-value between the simulated volatility sample generated from the estimated θ-vector by Θ and the sample generated from the true θ-vector. The p-value indicates the likelihood of observing a discrepancy as large as or larger than the one observed, assuming both samples are drawn from the same distribution. By computing the proportion of non-rejection of the null hypothesis of the KS test at different significance levels, we can evaluate the agreement between the estimated and true θ-vectors regarding the underlying conditional distributions. The results of this analysis are presented in the following table, providing valuable insights into the robustness and reliability of the estimation procedure conducted by Θ.

	$\delta = 1$	$\delta = 5$	$\delta = 21$	$\delta = 42$	$\delta = 63$
Proportion (%) at a significance level of 0.1%	93.0	94.8	96.4	97.1	97.2
Proportion (%) at a significance level of 1%	84.7	88.4	90.2	91.9	91.9
Proportion (%) at a significance level of 5%	76.1	80.9	83.5	84.9	85.2
Proportion (%) at a significance level of 10%	68.5	73.2	76.4	77.8	77.5

Table 7: Proportion of non-rejection of the null hypothesis of the KS test.

The results obtained demonstrate a high level of consistency between the conditional distributions generated from the estimated θ-vectors by Θ and those generated from the true θ-vectors. This consistency is evident through significant proportions of non-rejection observed across different significance levels for the various time horizons examined. Notably, even with a relatively high significance level of 10%, a substantial portion of the sample (ranging from 68.5% to 77.5% depending on the time horizon) does not reject the null hypothesis, indicating a strong agreement between the estimated and true θ-vectors concerning the associated conditional distributions. These findings emphasize the robustness and reliability of the estimation procedure performed by Θ in capturing the underlying future volatility distributions accurately.

Furthermore, it is interesting to note that the proportion of non-rejection of the null hypothesis increases with the time horizon. This phenomenon can be explained by two main factors already discussed in section 4.1.2. Firstly, as the time horizon increases, the data variability also increases, leading to conditional distributions with a larger standard deviation and, consequently, a greater acceptance of the null hypothesis. Secondly, as

the time horizon grows, the significance of short-term information contained in the state variables associated with higher discount factors diminishes in its impact on the conditional distributions. However, these state variables are particularly challenging to estimate due to the daily observation frequency.

4.2 Evaluation of estimation procedure using market data

The objective of the estimation procedure presented in this article is to make the RPDV model a robust model for volatility prediction. The purpose of this section is therefore to evaluate the performance of the RPDV model on real data according to this objective.

4.2.1 Market data sets

To evaluate the performance of Θ on real data, the tests conducted in this section utilize historical data from 2000 to 2022 for five stock indices: S&P500, Nasdaq, FTSE, DAX, and Euro Stoxx 50. These historical datasets consist of daily observations for each index, including its corresponding value and the realized volatility over the day, annualized. To create the input matrices D for the estimator Θ, a rolling window approach is employed. Specifically, a window of size 1260×2 is slid with a step of 1 trading day over the 22-year historical period. This process generates the matrices D of dimension 1260×2 that are used as inputs for the estimator Θ. Furthermore, the prediction horizons considered are 1, 5, 21, 42, and 63 trading days. Therefore, from a historical period of 5544 trading days (approximately 22 years), a total of 4222 (5544 - 1260 + 1 - 63) test pairs are obtained: $\left\{\mathrm{D}^{(i)}, \left\{\tilde{\sigma}_{i+\delta_k}\right\}_{k=1}^{5}\right\}_{i=1}^{4222}$.

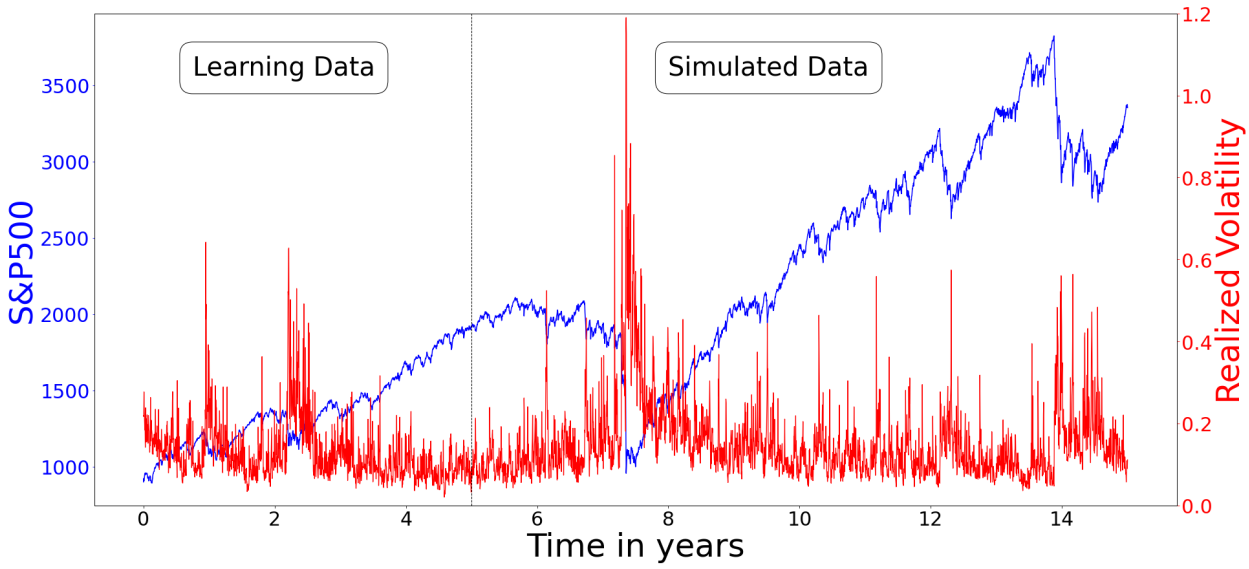

Figure 9: Example of joint evolution of the S&P500 and its realized volatility: the first 5 years are real data used to estimate θ_T from Θ, followed by 10 simulated years using this θ-vector.

4.2.2 Comparison of model forecasts with benchmark volatility models

The objective of this section is to assess the performance of the RPDV model on market data presented in section 4.2.1. For this purpose, we use as volatility forecaster, the estimator of $\mathbb{E}\left[\sigma_{T+\delta}|\mathrm{D}\right]$:

$$\hat{\sigma}_{T+\delta} = \mathcal{M}\big(\Theta(\mathrm{D}), \delta\big)\big)_1,$$

which we then compare with the following benchmark models from academic literature (Gatheral *et al.* 2019, Rosenbaum and Zhang 2022):

- The autoregressive (AR) models of order 5 and 21 (with trading days frequency), which take the following form for an order p model:

$$\sigma_t = a + \sum_{k=1}^{p} b_k \cdot \sigma_{t-k}.$$

- The heterogeneous autoregressive (HAR) models introduced by Corsi (2002) HAR

$$\hat{\sigma}_t = a + b_1\sigma_{t-1} + b_2 \sum_{k=1}^{5} \sigma_{t-k} + b_3 \sum_{k=1}^{21} \sigma_{t-k}.$$

- The rough fractional stochastic volatility (RFSV) model introduced by Gatheral *et al.* (2019)

$$\hat{\sigma}_t = \exp\left(\frac{\cos(H\pi)}{\pi} \int_{-\infty}^{t-1} \frac{\sigma_s}{(t-s+1)(t-s)^{H+0.5}} \mathrm{d}s + \frac{\mathrm{G}(1.5-H)\nu^2}{2\mathrm{G}(H+0.5)\mathrm{G}(2-2H)}\right).$$

  where G(.) denotes the gamma function. In practice, we truncate the integral to 1260 trading days and approximate it using a Riemann sum.

Each of these models is estimated using the data contained in the first column of the matrices D, which represents the historical realized volatility over the past 1260 trading days (approximately 5 years). The parameters of the AR and HAR models are estimated using the ordinary least squares method, while the FSVM is estimated using the method proposed by Gatheral. The accuracy of these different forecasters is evaluated by calculating the MSE between their respective forecasts and realized volatility, using the market data considered in section 4.2.1.

The results presented in table 8 demonstrate that, in most of the cases examined, the RPDV predictions outperform alternative models in terms of forecast accuracy. However, it is important to note that the predictive ability of RPDV heavily relies on the specific time horizon being considered. Specifically, for all 1-day horizon forecasts, the MSE-based performance of RPDV is inferior to that of the AR models or RFSV. Nevertheless, its relative performance significantly improves for the 5 trading days horizon, where it becomes the most accurate model in 3 out of 5 cases. Moreover, for longer time horizons such as 1, 2, and 3 months, RPDV consistently outperforms other volatility forecasters. In general, as the time horizon increases, RFSV forecasts tend to outperform those of other models.

The differential performance based on the considered time horizons could be attributed to two different reasons. The first one is the chosen model. This one is purely path-dependent, whereas empirical data suggests that volatility dynamics are also driven by exogenous frictions (Guyon and Lekeufack 2023/Parent 2023). Additionally, the kernels associated with the variables R_1 and R_2 follow a power law, while shifted power law kernels better fit market data according to Guyon and Lekeufack (2023). Therefore, using a model with more flexible kernels and incorporating an exogenous component of volatility could improve the performance of the approach. The second possible cause could be the estimation method itself. Indeed, unlike the other models considered, Θ is trained only using synthetic data with a constant observation time step of $\frac{1}{252}$ of a year, whereas the real data has an uneven observation frequency due to factors such as the presence of weekends. Thus, the empirical observation gap between two consecutive weekdays is $\frac{1}{365}$ of a year, and $\frac{3}{365}$ of a year between Friday and the following Monday. This bias can have a relatively strong impact on the

estimation of volatility at a short time horizon and tends to be smoothed out as the time horizon increases. Nevertheless, this is currently a limitation of the proposed estimation method that is worth highlighting. To address this limitation, one potential approach is to incorporate a combination of synthetic and real data in the training of Θ. By including real data with varying observation frequencies, the model can better adapt to the characteristics of empirical data. Additionally, introducing noise or biases in the training data can enhance the robustness of the θ-vector estimation and help reduce potential biases.

	AR(5)	AR(21)	HAR	RFSV	RPDV
SPX $\delta = 1$	0.0030	0.0031	0.0036	0.0031	0.0033
SPX $\delta = 5$	0.0052	0.0052	0.0052	0.0051	0.0050
SPX $\delta = 21$	0.0087	0.0083	0.0081	0.0080	0.0076
SPX $\delta = 42$	0.0106	0.0100	0.0098	0.0094	0.0087
SPX $\delta = 63$	0.0111	0.0103	0.0103	0.0099	0.0089
Nasdaq $\delta = 1$	0.0021	0.0022	0.0027	0.0022	0.0028
Nasdaq $\delta = 5$	0.0039	0.0040	0.0039	0.0038	0.0040
Nasdaq $\delta = 21$	0.0064	0.0063	0.0059	0.0057	0.0058
Nasdaq $\delta = 42$	0.0076	0.0072	0.0069	0.0066	0.0064
Nasdaq $\delta = 63$	0.0080	0.0075	0.0073	0.0069	0.0065
FTSE $\delta = 1$	0.0038	0.0039	0.0042	0.0038	0.0040
FTSE $\delta = 5$	0.0055	0.0053	0.0053	0.0052	0.0053
FTSE $\delta = 21$	0.0086	0.0077	0.0076	0.0073	0.0074
FTSE $\delta = 42$	0.0101	0.0090	0.0088	0.0083	0.0083
FTSE $\delta = 63$	0.0107	0.0094	0.0093	0.0088	0.0086
DAX $\delta = 1$	0.0026	0.0026	0.0031	0.0026	0.0029
DAX $\delta = 5$	0.0045	0.0044	0.0042	0.0042	0.0041
DAX $\delta = 21$	0.0075	0.0067	0.0064	0.0062	0.0060
DAX $\delta = 42$	0.0090	0.0082	0.0076	0.0073	0.0068
DAX $\delta = 63$	0.0096	0.0086	0.0081	0.0078	0.0072
Stox50 $\delta = 1$	0.0039	0.0040	0.0046	0.0039	0.0042
Stox50 $\delta = 5$	0.0062	0.0061	0.0061	0.0060	0.0058
Stox50 $\delta = 21$	0.0099	0.0091	0.0088	0.0084	0.0081
Stox50 $\delta = 42$	0.0110	0.0105	0.0101	0.0095	0.0090
Stox50 $\delta = 63$	0.0114	0.0106	0.0105	0.0099	0.0092

Table 8: MSE for the AR, HAR, RFSV and RPDV predictors.

4.2.3 Evaluation by density

The evaluation of standard deviations for conditional volatility distributions is not directly possible from historical data since, by definition, we only have a single realization for each date. Therefore, we proceed indirectly by using the approximation of volatility distributions with the log-normal distribution introduced in section 2.3. In this framework, our estimator for the volatility distribution at horizon $T + \delta$ at time T is the log-normal distribution $\mathcal{LN}\left(\tilde{\mathrm{m}}_{T+\delta}, (\tilde{\mathrm{s}}_{T+\delta})^2\right)$, where:

$$\tilde{\mathrm{m}}_{T+\delta} = \log\big(\mathcal{M}\big(\Theta(\mathrm{D}), \delta\big)_1\big) - 0.5\big(\tilde{\mathrm{s}}_{T+\delta}\big)^2 \quad \text{and} \quad \big(\tilde{\mathrm{s}}_{T+\delta}\big)^2 = \log\left(\left(\frac{\mathcal{M}\big(\Theta(\mathrm{D}), \delta\big)_2}{\mathcal{M}\big(\Theta(\mathrm{D}), \delta\big)_1}\right)^2 + 1\right).$$

Using the properties of the log-normal distribution, we define the estimator of the cumulative distribution function (CDF) of $\sigma_{T+\delta}$ at time T as follows:

$$\hat{F}_{T+\delta}(\sigma) = 0.5 + 0.5 \cdot \operatorname{erf}\left(\frac{\log(\sigma) - \tilde{\mathrm{m}}_{T+\delta}}{\tilde{\mathrm{s}}_{T+\delta}\sqrt{2}}\right). \tag{21}$$

We then proceed to calculate the proportion of observed realized volatility values that fall within different confidence intervals constructed based on this estimated CDF. Specifically, we compute the included proportion of the sample that falls within:

- the bilateral confidence interval $[\alpha/2 : 1 - \alpha/2]$:

$$p_{1-\alpha} = \frac{\sum_{k=1}^{N} \mathbb{1}\left\{\frac{\alpha}{2} \leq \hat{F}_{T_k+\delta}\left(\sigma_{T_k+\delta}\right) \leq 1 - \frac{\alpha}{2}\right\}}{N},$$

- the upper unilateral confidence interval $[\alpha : 1]$:

$$p^{(+)}_{1-\alpha} = \frac{\sum_{k=1}^{N} \mathbb{1}\left\{\alpha \leq \hat{F}_{T_k+\delta}\left(\sigma_{T_k+\delta}\right)\right\}}{N},$$

- the lower unilateral confidence interval $[0 : 1 - \alpha]$:

$$p^{(-)}_{1-\alpha} = \frac{\sum_{k=1}^{N} \mathbb{1}\left\{\hat{F}_{T_k+\delta}\left(\sigma_{T_k+\delta}\right) \leq 1 - \alpha\right\}}{N}.$$

These calculations are performed for the following values of α: 0.05, 0.1, 0.25, 0.5. The idea is then to compare the theoretical proportion, which should be $1 - \alpha$, with the calculated proportions $p_{1-\alpha}$, $p^{(+)}_{1-\alpha}$, and $p^{(-)}_{1-\alpha}$. Indeed, the closer the calculated proportions $p_{1-\alpha}$, $p^{(+)}_{1-\alpha}$, and $p^{(-)}_{1-\alpha}$ are to $1 - \alpha$, the stronger the indication that $\hat{F}_{T_k+\delta}$ is a reliable estimator of the conditional distributions of volatility at horizon δ.

The figures reported in the table 9 demonstrate that our estimator of conditional volatility distributions generally provides a good approximation of the actual conditional distributions. The proportions of realized volatility included in the estimated confidence intervals are typically close to the theoretical proportions (i.e., $1 - \alpha$), indicating that the estimator effectively captures the characteristics of the conditional distributions. This, in turn, suggests that $\mathcal{M}\big(\Theta\big(\mathrm{D}\big), \delta\big)_2$ produces a good estimation of conditional standard deviations. However, it should be noted that, as already mentioned in section 4.2.2, the quality of the model estimations is quite sensitive to the considered time horizon. Thus, while the difference between the theoretical proportion and the observed proportion inside the confidence intervals is around 10 points in most cases for a 1-day horizon, this difference is almost always less than 5 points for horizons equal to or greater than 1 month.

Another interesting point is that the narrower the confidence interval, the more accurate the model estimation, in the sense that the empirical proportions approach the theoretical proportions. Moreover, the empirical proportions are generally lower than the theoretical proportions, especially when considering wider confidence intervals. This phenomenon can be explained by several factors. The first, which is certainly the most important, is that observations outside the confidence intervals are simply the result of a poor model prediction for a part of the sample. This hypothesis is supported by the fact that these discrepancies are strongly correlated with the model's relative performance reported in table 8. Another factor explaining this discrepancy is an underestimation by the model of the conditional standard deviations. In fact, even if the model perfectly predicted the conditional means, such an underestimation of the standard deviations would lead

to $p_{1-\alpha} < 1-\alpha$. Finally, part of this difference could be also explained by the use of the log-normal approximation. Indeed, with constant mean and standard deviation, the kurtosis of the volatility distributions directly generated from RPDV model tends to be slightly higher than their log-normal approximations (it can be seen in figure 1).

	$p_{0.95}$	$p^{(+)}_{0.95}$	$p^{(-)}_{0.95}$	$p_{0.90}$	$p^{(+)}_{0.90}$	$p^{(-)}_{0.90}$	$p_{0.75}$	$p^{(+)}_{0.75}$	$p^{(-)}_{0.75}$	$p_{0.50}$	$p^{(+)}_{0.50}$	$p^{(-)}_{0.50}$
SPX $\delta = 1$	0.86	0.90	0.91	0.80	0.85	0.85	0.65	0.72	0.72	0.44	0.50	0.50
SPX $\delta = 5$	0.89	0.92	0.91	0.83	0.88	0.85	0.67	0.74	0.69	0.44	0.53	0.47
SPX $\delta = 21$	0.90	0.92	0.93	0.85	0.88	0.88	0.70	0.74	0.71	0.46	0.52	0.48
SPX $\delta = 42$	0.90	0.92	0.94	0.86	0.87	0.89	0.71	0.75	0.73	0.48	0.52	0.48
SPX $\delta = 63$	0.91	0.92	0.94	0.86	0.88	0.89	0.73	0.75	0.74	0.49	0.52	0.49
Nasdaq $\delta = 1$	0.87	0.90	0.91	0.81	0.85	0.86	0.65	0.70	0.73	0.43	0.47	0.53
Nasdaq $\delta = 5$	0.89	0.93	0.89	0.82	0.88	0.83	0.68	0.75	0.70	0.45	0.51	0.49
Nasdaq $\delta = 21$	0.89	0.93	0.92	0.84	0.88	0.86	0.69	0.75	0.72	0.46	0.50	0.50
Nasdaq $\delta = 42$	0.90	0.92	0.94	0.85	0.88	0.88	0.71	0.75	0.73	0.48	0.52	0.48
Nasdaq $\delta = 63$	0.90	0.92	0.94	0.86	0.88	0.88	0.72	0.75	0.74	0.49	0.51	0.49
FTSE $\delta = 1$	0.86	0.88	0.89	0.79	0.82	0.82	0.63	0.68	0.68	0.42	0.47	0.52
FTSE $\delta = 5$	0.88	0.89	0.92	0.82	0.83	0.87	0.65	0.71	0.73	0.43	0.49	0.51
FTSE $\delta = 21$	0.90	0.91	0.93	0.84	0.86	0.86	0.68	0.72	0.74	0.46	0.49	0.51
FTSE $\delta = 42$	0.90	0.92	0.94	0.85	0.86	0.88	0.69	0.72	0.75	0.47	0.50	0.50
FTSE $\delta = 63$	0.91	0.91	0.94	0.85	0.86	0.88	0.70	0.72	0.76	0.48	0.50	0.50
DAX $\delta = 1$	0.88	0.89	0.91	0.83	0.83	0.87	0.66	0.71	0.73	0.45	0.50	0.50
DAX $\delta = 5$	0.89	0.90	0.94	0.84	0.85	0.89	0.68	0.73	0.75	0.47	0.51	0.49
DAX $\delta = 21$	0.90	0.91	0.94	0.86	0.86	0.90	0.69	0.73	0.76	0.49	0.50	0.50
DAX $\delta = 42$	0.91	0.91	0.94	0.86	0.87	0.91	0.70	0.73	0.77	0.49	0.50	0.50
DAX $\delta = 63$	0.91	0.91	0.94	0.87	0.87	0.92	0.71	0.73	0.78	0.50	0.50	0.50
Stox $\delta = 1$	0.87	0.89	0.91	0.82	0.84	0.83	0.65	0.70	0.70	0.43	0.49	0.51
Stox $\delta = 5$	0.89	0.91	0.92	0.84	0.86	0.85	0.67	0.71	0.71	0.45	0.50	0.50
Stox $\delta = 21$	0.90	0.92	0.92	0.85	0.87	0.87	0.69	0.73	0.72	0.48	0.51	0.49
Stox $\delta = 42$	0.91	0.92	0.94	0.86	0.88	0.88	0.70	0.73	0.73	0.49	0.51	0.49
Stox $\delta = 63$	0.91	0.92	0.94	0.87	0.87	0.88	0.71	0.73	0.74	0.50	0.50	0.50

Table 9: Proportions of realized volatility samples included in estimated confidence intervals.

5 Conclusion

The present article has introduced an innovative deep estimation method for volatility models, specifically designed for volatility forecasting within the theoretical framework of Bayesian decision theory. To illustrate this method, the article focused on estimating a version of the RPDV model.

The objective of the proposed approach was formally outlined in Section 2. It involves constructing an estimator function for the considered model that, from a historical matrix of price and realized volatility data, returns optimal parameters and state variables according to Bayesian decision theory principles and based on the criterion defined in this section. This criterion, arising from a forecasting objective across different horizons, has been defined as a function of the first two conditional moments of volatility at various time horizons.

The estimation method itself has been exposed in section 3. It involves 2 NNs. The first one, denoted as Θ, serves as the estimator function. It takes a historical matrix of price and realized volatility data as input

and returns a θ-vector containing the parameters and state variables defining a Markovian approximation of the RPDV model at a specific time instant. The second neural network, denoted as $\mathcal{M}$, estimates the mean and standard deviation of the volatility under the considered RPDV model for a given pair of θ-vector and time horizon. This NN thus addresses the absence of an analytical formula for these moments. The proposed estimation method first involves training this neural network $\mathcal{M}$, and then, in a second step, training Θ through interaction with $\mathcal{M}$. In this approach, $\mathcal{M}$ is used to compute the cost of the θ-vectors predicted by Θ, thereby adjusting the parameters of the neural network Θ accordingly. Importantly, it has been demonstrated that under certain conditions, following the proposed estimation procedure, Θ behaves asymptotically as a Bayesian estimator aligned with the volatility prediction objective outlined in section 2. Consequently, the outputs of Θ offer estimations of the optimal θ-vectors tailored to the specified forecasting goal.

Section 4 presents a comprehensive evaluation of the practical effectiveness of the estimator function Θ using both synthetic and market data. The evaluation on synthetic data demonstrates that the estimated θ-vectors by Θ yield volatility distribution estimates that closely align with the real distributions at different time horizons. These results highlight the efficacy of the proposed estimation method within the analytical framework, where the estimation data are noise-free and generated from the model being estimated. The evaluation using market data, spanning 22 years of data from 5 stock indices, provided insights under less favorable conditions. The results showed a generally positive outcome, although with more mixed findings compared to the tests conducted on synthetic data. Notably, the model's performance as a volatility forecaster varied depending on the chosen time horizon. For the 1 trading day volatility forecast, the model exhibited lower performance compared to benchmark forecasters such as AR and RSFV. However, for a 1-week horizon (5 trading days), the model's performance became comparable to, or even slightly better than, the benchmark models. Moreover, the model consistently outperformed other models for longer horizons of 1 month or more, including the HAR and RSFV models that are known for their effectiveness in volatility prediction over longer timeframes.

The reasons put forward to explain the differential performance based on the considered time horizon are of two kinds: the first is related to the chosen volatility model, the second to the estimation procedure itself. Regarding the first reason, a model allowing more flexible kernels and enabling the incorporation of an exogenous component of volatility could be better adapted to capture the empirical dynamics of volatility, thereby improving short-term forecasting. Regarding the estimation procedure itself, the fact that Θ is trained solely on synthetic data with a constant observation time step, while the observation frequency varies for empirical data, could introduce a bias in the prediction of state variables, which diminishes as the prediction time horizon decreases. To address this limitation, one potential approach could be to incorporate a combination of synthetic and real data in the training of Θ. Additionally, introducing noise or biases in the training data may also enhance the robustness of the θ-vector estimation and help reduce potential biases. These avenues for improvement could refine the estimation framework presented in this article, which already demonstrates promising results, particularly in utilizing RPDV as a volatility predictor for medium to long horizons.

6 Acknowledgments

I would like to express my gratitude to my thesis supervisor, Jean-Paul Laurent, for his invaluable support and insightful advice throughout the preparation of this paper.

References

[1] Abi Jaber E., and El Euch O. (2019). Multifactor approximation of rough volatility models. *SIAM Journal on Financial Mathematics*, 10(2), 309-349.

[2] Abi Jaber E. (2019). Lifting the Heston model. *Quantitative Finance*, 19(12), 1995-2013.

[3] Aït-Sahalia Y., and Kimmel R. (2007). Maximum likelihood estimation of stochastic volatility models. *Journal of financial economics*, 83(2), 413-452.

[4] Alizadeh S., Brandt M. W., and Diebold F.X. (2002). Rangebased estimation of stochastic volatility models. *The Journal of Finance*, 57(3), 1047-1091.

[5] Andersen T. G., Bollerslev T., Diebold F. X., and Labys P. (2003). Modeling and forecasting realized volatility. *Econometrica*, 71(2), 579-625.

[6] Barra S., Carta S.M., Corriga A., Podda A.S., and Recupero D.R. (2020). Deep learning and time series-to-image encoding for financial forecasting. *IEEE/CAA Journal of Automatica Sinica*, 7(3), 683-692.

[7] Bayer C., Horvath B., Muguruza A., Stemper B., and Tomas M. (2019). On deep calibration of (rough) stochastic volatility models. *arXiv* preprint arXiv:1908.08806.

[8] Berger J.O. (2013). *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media.

[9] Bickel P.J., and Doksum K.A. (2015). *Mathematical statistics: basic ideas and selected topics, volumes I* CRC Press.

[10] Bildirici M., and Ersin Ö. (2014). Asymmetric power and fractionally integrated support vector and neural network GARCH models with an application to forecasting financial returns in ise100 stock index. *Economic Computation and Economic Cybernetics Studies and Research*, 48, 1-22.

[11] Bollerslev T., Chou, R.Y., and Kroner K. F. (1992). ARCH modeling in finance: A review of the theory and empirical evidence. *Journal of econometrics*, 52(1-2), 5-59.

[12] Brooks C., and Persand G. (2001). Volatility forecasting for risk management. *Journal of Forecasting*, 20(5), 341-356.

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

[14] Csáji B.C. (2001). Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, 24(48), 7.

[15] Cui Z., Chen W., and Chen Y. (2016). Multi-scale convolutional neural networks for time series classification. *arXiv* preprint arXiv:1603.06995.

[16] Diebold F.X. (1998). Elements of forecasting. Cincinnati, OH, USA: South-Western College Pub.

[17] Franses P.H., and Van Dijk D. (1996). Forecasting stock market volatility using (nonlinear) Garch models. *Journal of forecasting*, 15(3), 229-235.

[18] Gatheral J., Jaisson T., and Rosenbaum M. (2018). Volatility is Rough. *Quantitative Finance*, 18(6), 933-949.

[19] Gatheral J., Jusselin P., and Rosenbaum M. (2020). The Quadratic Rough Heston Model and the Joint S&P500/VIX Smile Calibration Problem. *arXiv* preprint arXiv:2001.01789.

[20] Gil M., Alajaji F., and Linder T. (2013). Rényi divergence measures for commonly used univariate continuous distributions. *Information Sciences*, 249, 124-131.

[21] Guyon J., and Lekeufack J. (2023). Volatility is (mostly) path-dependent. *Quantitative Finance*, 23(9), 1221-1258.

[22] Hernandez A. (2016). Model calibration with neural networks. Available at *SSRN* 2812140.

[23] Hornik K., Stinchcombe M., and White H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5), 359-366.

[24] Horvath B., Muguruza A., and Tomas M. (2021). Deep learning volatility: a deep neural network perspective on pricing and calibration in (rough) volatility models. *Quantitative Finance*, 21(1), 11-27.

[25] Kimoto T., Asakawa K., Yoda M., and Takeoka M. (1990). Stock market prediction system with modular neural networks. In 1990 *IJCNN international joint conference on neural networks* (pp. 1-6). IEEE.

[26] Kim S., Shephard N., and Chib S. (1998). Stochastic volatility: likelihood inference and comparison with ARCH models. *The review of economic studies*, 65(3), 361-393.

[27] Morrill J., Fermanian A., Kidger P., and Lyons T. (2020). A generalised signature method for multivariate time series feature extraction. *arXiv* preprint arXiv:2006.00873.

[28] Nelson D. B. (1991). Conditional heteroskedasticity in asset returns: A new approach. *Econometrica: Journal of the econometric society*, 347-370.

[29] Parent L. (2022). Rough Path-Dependent Volatility Models. Available at *SSRN* 4270481.

[30] Parmigiani G., and Inoue L. (2009). *Decision theory: Principles and approaches*. John Wiley & Sons.

[31] Poon S. H., and Granger C.W.J. (2003). Forecasting volatility in financial markets: A review. *Journal of economic literature*, 41(2), 478-539.

[32] Rosenbaum M., and Zhang J. (2021). Deep calibration of the quadratic rough Heston model. *arXiv* preprint arXiv:2107.01611.

[33] Rosenbaum M., and Zhang J. (2022). On the universality of the volatility formation process: when machine learning and rough volatility agree. *arXiv* preprint arXiv:2206.14114.

[34] Tegnér M., Poulsen R. (2018). Volatility is log-normalBut not for the reason you think. *Risks*, 6(2), 46.

[35] Usmani M., Adi S.H., Raza K., and Ali S.S.A. (2016). Stock market prediction using machine learning techniques. In 2016 *3rd international conference on computer and information sciences* (ICCOINS) (pp. 322-327). IEEE.

[36] Wang Z., and Oates T. (2015). Imaging time-series to improve classification and imputation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

[37] Wei Y., Wang Y., and Huang D. (2010). Forecasting crude oil market volatility: Further evidence using GARCH-class models. *Energy Economics*, 32(6), 1477-1484.

[38] Zakoian J. M. (1994). Threshold heteroskedastic models. *Journal of Economic Dynamics and control*, 18(5), 931-955.

Appendix A Approximation of the RPDV model

A.1 Stochastic differential equations for Markovian approximation of the RPDV model

We aim to solve the following SDE:

$$\mathrm{dR}_{1,i,t} = \gamma_i \left(\frac{\mathrm{d}P_t}{P_t} - (\kappa_1 R_{1,t} + \mathrm{R}_{1,i,t})\,\mathrm{d}t \right),$$

To consider the dynamics of $\mathrm{R}_{1,i,t}$, we set $g\left(\mathrm{R}_{1,i,t}, t\right) = e^{\gamma_i t}\mathrm{R}_{1,i,t}$ and apply the Itô lemma:

$$\mathrm{d}e^{-\gamma_i t}\mathrm{R}_{1,i,t} = \gamma_i e^{\gamma_i t}\mathrm{R}_{1,i,t}\mathrm{d}t + e^{\gamma_i t}\mathrm{dR}_{1,i,t} = \gamma_i e^{\gamma_i t}\left(\frac{\mathrm{d}P_t}{P_t} - \kappa_1 R_{1,t}\mathrm{d}t \right).$$

Consequently:

$$\mathrm{R}_{1,i,t} = \mathrm{R}_{1,i,0}e^{-\gamma_i t} + \gamma_i \int_0^t e^{-\gamma_i (t-u)} \left(\frac{\mathrm{d}P_u}{P_u} - \kappa_1 R_{1,u}\mathrm{d}u \right).$$

Thus:

$$\sum_{i=1}^{n} w_{1,i}\mathrm{R}_{1,i,t} = R_{1,t} = \sum_{i=1}^{n} \gamma_i w_{1,i} e^{-\gamma_{1,i} t}\mathrm{R}_{1,i,0} + \int_0^t \underbrace{\sum_{i=1}^{n} \gamma_i w_{1,i} e^{-\gamma_{1,i}(t-u)}}_{\tilde{K}_1(t-u)} \left(\frac{\mathrm{d}P_u}{P_u} - \kappa_1 R_{1,u}\mathrm{d}u \right),$$

If follows that

$$\lim_{t\to+\infty} R_{1,t} = \int_0^t \hat{K}(t-u) \left(\frac{\mathrm{d}P_u}{P_u} - \kappa_1 R_{1,u}\mathrm{d}u \right).$$

Analogously, with

$$\mathrm{dR}_{2,i,t} = \left((\sigma_t)^2 - \kappa_2 R_{2,t} - \gamma_i \mathrm{R}_{2,i,t} \right)\mathrm{d}t,$$

by applying same steps, we obtain:

$$\sum_{i=1}^{n} w_{2,i}\mathrm{R}_{2,i,t} = R_{2,t} = \sum_{i=1}^{n} \gamma_i w_{2,i} e^{-\gamma_{2,i} t}\mathrm{R}_{2,i,0} + \int_0^t \sum_{i=1}^{n} w_{2,i} e^{-\gamma_{2,i}(t-u)} \Big((\sigma_u)^2 - \kappa_2 R_{2,t} \Big)\mathrm{d}u.$$

and therefore

$$\lim_{t\to+\infty} R_{2,t} = \int_0^t \hat{K}(t-u) \Big((\sigma_u)^2 - \kappa_2 R_{2,t} \Big)\mathrm{d}u.$$

A.2 Approximation of the power law kernel

In the original article (Parent 2022), it is shown that vectors W_j and Λ_j can be determined using the work of Abi Jaber (2019) based on the expression of the kernel $K_j(\tau_j) = \tau^{-\alpha_j}$ as the Laplace transform of a positive measure. However, this method has several drawbacks. The first is that the convergence between K_j and $\hat{K}_j$

is relatively slow with respect to n, the number of exponential kernels that make up $\hat{K}_j$. Consequently, when n is small ($n \leq 10$), there exists kernels of the same form (and with the same n) that better approximate the power kernel in the L^2 sense . Furthermore, the discount coefficients obtained through this method depend on α. While this is not inherently a problem, it complexify the estimation problem in practice. For these reasons, an alternative approximation of K_j is used here, in which the discount coefficients are constant (they do not depend on α_j) and only the weight vector W_j vary.

In order to fix Λ_j, we start to remark that the inverse of the discount coefficients $\gamma_{j,i}$ corresponds to the duration of $(\mathrm{R}_{j,t})_i$. Based on that, we start by defining the shortest and longest durations as $\tau_- = \gamma_{j,1}^{-1}$ and $\tau_+ = \gamma_{j,n}^{-1}$, respectively. In this case, we set $\tau_- = \frac{1}{10000}$ and $\tau_+ = 1000$ expressed in years. Subsequently, we perform a uniform logarithmic discretization between these two bounds to determine the values of the remaining $n-2$ discounting coefficients $(\gamma_{j,i})_{2 \leq i \leq n-1}$, as follows:

$$\gamma_{j,i} = \exp\left(\log(\tau_-) + \frac{\log(\tau_+) - \log(t_-)}{n-1}(i-1)\right)^{-1}.$$

The idea is to have a set of exponential kernels with durations distributed in such a way as to be able to well approximate any power law kernel $K(\tau) = \tau^{-\alpha}$ with $\alpha \in]0:1[$. With the value of Λ_j fixed, we then solve the following least-square problem:

$$\underset{\mathrm{W}_j \geq \mathbf{0_n}}{\arg\min} \left\| \mathrm{y}_j - \mathrm{A}_j \mathrm{W}_j \right\|^2$$

with $\mathbf{0_n}$ a n-dimensional vector of zeros,

$$\mathrm{A}_j = \begin{bmatrix} \gamma_{j,1} e^{-\gamma_{j,1}\tau_1} & \dots & \gamma_n e^{-\gamma_{j,n}\tau_1} \\ \vdots & \ddots & \vdots \\ \gamma_{j,1} e^{-\gamma_{j,1} N \tau_N} & \dots & \gamma_n e^{-\gamma_{j,n}\tau_N} \end{bmatrix}, \quad \mathrm{y}_j = \begin{bmatrix} {\tau_1}^{-\alpha} \\ \dots \\ {\tau_N}^{-\alpha} \end{bmatrix}.$$

In order to evaluate the quality of the approximation obtained by this method, we will compare it with the method proposed by Abi Jaber (2019), using $n = 10$ in both cases. Accordingly, we compute the L^1 and L^2 norms of the difference over the time interval $[\frac{1}{10000} : 10]$ between power-law kernels and their associated approximations using each of these 2 methods. Table 10 reports the results obtained.

α	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
$\left\|\hat{K}_1 - K\right\|_{L^1(T)}$	2.443	1.475	0.873	0.586	0.518	0.489	0.563	0.844	1.604
$\left\|\hat{K}_2 - K\right\|_{L^1(T)}$	0.016	0.026	0.041	0.065	0.107	0.188	0.355	0.449	0.970
$\left\|\hat{K}_1 - K\right\|_{L^2(T)}$	0.618	0.22	0.105	0.218	1.391	9.51	65.5	436.14	2809.4
$\left\|\hat{K}_2 - K\right\|_{L^2(T)}$	0.0004	0.004	0.033	0.228	1.412	8.11	43.8	66.51	470.8

Table 10: Comparison of two power-law kernel approximation methods based on L^1 and L^2 norms evaluated over the time interval $[\frac{1}{10000} : 10]$. $\hat{K}_1$ is the approximation method introduced by Abi Jaber, while $\hat{K}_2$ uses the method proposed in this section.

Based on the metrics considered, the method proposed here generally provides a better approximation of the power-law kernel than Abi Jaber's method for most of the considered α values. If we focus on the L^1 norm criterion, this method produces systematically a better approximation for all the considered α values. When we consider the L^2 norm criterion, Abi Jaber's approximation outperforms the method introduced in the present section for the cases where α is equal to 0.5 and 0.6. However, even in these two cases, the difference

in performance is very small.

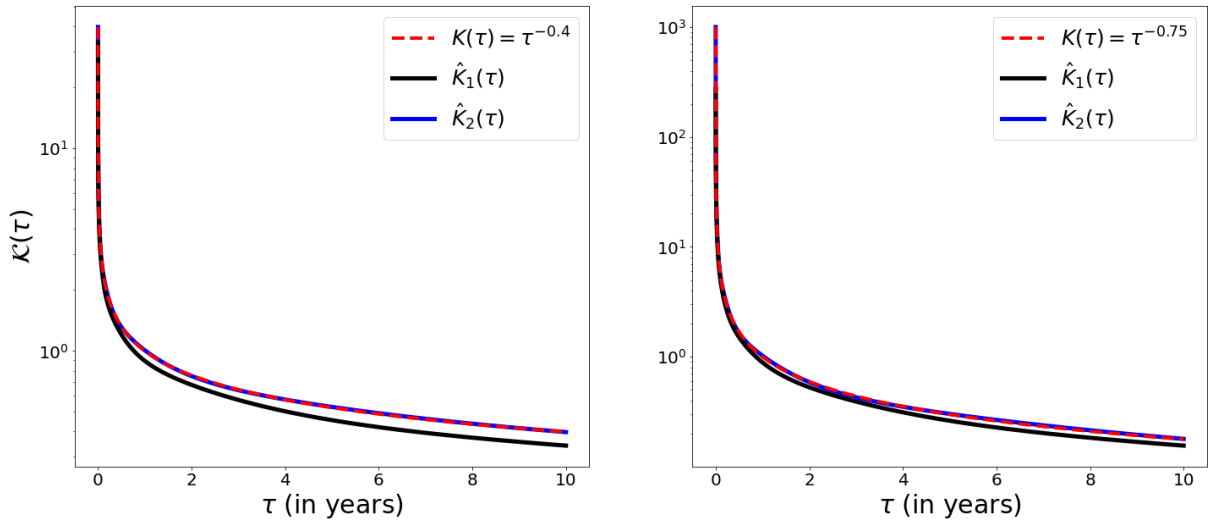

Figure 10: Examples of approximations of two power-law kernels using 2 different approximation methods. Kernel $\hat{K}_1$ is obtained using the Abi Jaber approximation, while kernel $\hat{K}_2$ is obtained using the approximation described in this section.

Appendix B Parameters of the log-normal approximation to the conditional volatility distribution

We want to express m and s in terms of $\mathbb{E}\left[\sigma_{T+\delta}|\theta_T\right]$ and $\mathbb{V}\mathrm{ar}\left[\sigma_{T+\delta}|\theta_T\right]$ given:

$$\mathbb{E}\left[\sigma_{T+\delta}|\theta_T\right] = e^{\mathrm{m}+\frac{\mathrm{s}^2}{2}}, \quad \mathbb{V}\mathrm{ar}\left[\sigma_{T+\delta}|\theta_T\right] = \left(e^{\mathrm{s}^2}-1\right)e^{2\mathrm{m}+\mathrm{s}^2}.$$

It is clear that the first equation can be rewrite as

$$\mathrm{m} = \log\left(\mathbb{E}\left[\sigma_{T+\delta}|\theta_T\right]\right) - \frac{\mathrm{s}^2}{2}.$$

Injecting this result in the second equation, we obtain:

$$\begin{aligned}
\mathbb{V}\mathrm{ar}\left[\sigma_{T+\delta}|\theta_T\right] &= \left(e^{\mathrm{s}^2}-1\right)e^{2\log\left(\mathbb{E}\left[\sigma_{T+\delta}|\theta_T\right]\right)} \\
\mathbb{V}\mathrm{ar}\left[\sigma_{T+\delta}|\theta_T\right] &= \left(e^{\mathrm{s}^2}-1\right)\mathbb{E}\left[\sigma_{T+\delta}|\theta_T\right]^2 \\
\mathrm{s}^2 &= \log\left(\frac{\mathbb{V}\mathrm{ar}\left[\sigma_{T+\delta} \mid \theta_T\right]}{\mathbb{E}\left[\sigma_{T+\delta} \mid \theta_T\right]^2}+1\right)
\end{aligned}$$

Therefore,

$$\mathrm{m} = \log\left(\mathbb{E}\left[\sigma_{T+\delta}|\theta_T\right]\right) - 0.5\log\left(\frac{\mathbb{V}\mathrm{ar}\left[\sigma_{T+\delta} \mid \theta_T\right]}{\mathbb{E}\left[\sigma_{T+\delta} \mid \theta_T\right]^2}+1\right).$$

Appendix C Proofs of convergence results

C.1 Convergence of the estimation procedure of the function $\mathcal{M}$

Proposition 1 *Let be* $\theta_T^{(1)}, ..., \theta_T^{(n_1)}$ *a sequence of i.i.d. random variable following* $\boldsymbol{\pi}$*, and* $\left\{\bar{M}_{T+\delta_k}^{(1)}\right\}_{1\leq j\leq p}, ...,$ $\left\{\bar{M}_{T+\delta_k}^{(n_1)}\right\}_{1\leq j\leq p}$ *a sequence of sets such as* $\forall\ i,k, \bar{M}_{T+\delta_k}^{(i)}$ *is an unbiased estimator of* $M\left(\theta_T^{(i)}, \delta_k\right)$ *calculated from a sample of size* n_2*. If it exists* $\mathcal{M}^\star$*, such as* $\mathcal{M}^\star\left(\theta_T, \delta_k\right) = M\left(\theta_T, \delta_k\right),\ \forall\ \theta_T : \boldsymbol{\pi}\left(\theta_T\right) \neq 0$ *and* $\delta_k \in \{\delta_1, ..., \delta_p\}$*, thus* $\forall\ \hat{\mathcal{M}}^\star$ *solution to*

$$\arg\min_{\mathcal{M}}\ \frac{1}{n_1}\sum_{i=1}^{n_1}\sum_{k=1}^{p}\left\|\mathcal{M}\left(\theta_T^{(i)}, \delta_k\right) - \bar{M}_{T+\delta_k}^{(i)}\right\|_2^2,$$

$\hat{\mathcal{M}}^\star\left(\theta, \delta_k\right) = M\left(\theta_T, \delta_k\right), \forall\ \theta_T : \boldsymbol{\pi}\left(\theta_T\right) \neq 0$ *and* $\delta_k \in \{\delta_1, ..., \delta_p\}$.

Proof of proposition 1. The density being by definition positive or zero, we have the following inequality:

$$\int_{\mathbb{R}^{2n+9}}\left(\min_{\mathcal{M}}\ \sum_{k=1}^{p}\left\|\mathcal{M}\left(\theta_T, \delta_k\right) - M\left(\theta_T, \delta_k\right)\right\|_2^2\right)\mathrm{d}\boldsymbol{\pi}(\theta_T) \leq \min_{\mathcal{M}}\int_{\mathbb{R}^{2n+9}}\ \sum_{k=1}^{p}\left\|\mathcal{M}\left(\theta_T, \delta_k\right) - M\left(\theta_T, \delta_k\right)\right\|_2^2\mathrm{d}\boldsymbol{\pi}(\theta_T).$$

In addition, $\forall\ \mathrm{M} \in \mathbb{R}^2$, M is the unique solution to

$$\arg\min_{\hat{\mathrm{M}}}\ \left\|\mathrm{M} - \hat{\mathrm{M}}\right\|_2^2.$$

It follows that if it exists $\mathcal{M}^\star$ such as $\mathcal{M}^\star(\theta_T, \delta_k) = M(\theta_T, \delta_k),\ \forall\ \theta : \boldsymbol{\pi}\left(\theta_T\right) \neq 0$, and $\delta_k \in \{\delta_1, ..., \delta_p\}$, thus $\forall\ \hat{\mathcal{M}}^\star$ solution to

$$\arg\min_{\mathcal{M}}\int_{\mathbb{R}^{2n+9}}\ \sum_{k=1}^{p}\left\|\mathcal{M}\left(\theta_T, \delta_k\right) - M\left(\theta_T, \delta_k\right)\right\|_2^2\mathrm{d}\boldsymbol{\pi}(\theta),$$

$\hat{\mathcal{M}}^\star(\theta_T, \delta_k) = M(\theta_T, \delta_k),\ \forall\ \theta : \boldsymbol{\pi}\left(\theta\right) \neq 0$. Moreover, because $\bar{M}_{T+\delta_k}^{(i)}$ is an unbiased estimator of $M\left(\theta_T^{(i)}, \delta_k\right)$ calculated from a sample of size n_2 [7] :

$$\lim_{n_2\to+\infty}\bar{M}_{T+\delta_k}^{(i)} = M\left(\theta_T^{(i)}, \delta_k\right).$$

Similarly, by the law the of large numbers

$$\lim_{\substack{n_1\to+\infty\\ n_2\to+\infty}}\frac{1}{n_1}\sum_{i=1}^{n_1}\sum_{k=1}^{p}\left\|\mathcal{M}\left(\theta_T^{(i)}, \delta_k\right) - \bar{M}_{T+\delta_k}^{(i)}\right\|_2^2 = \int_{\mathbb{R}^{2n+9}}\ \sum_{k=1}^{p}\left\|\mathcal{M}\left(\theta_T, \delta_k\right) - M\left(\theta_T, \delta_k\right)\right\|_2^2\mathrm{d}\boldsymbol{\pi}(\theta_T).$$

Therefore, using previous results, under the existence condition of $\mathcal{M}^\star$, $\forall\ \hat{\mathcal{M}}^\star$ solution to

$$\arg\min_{\mathcal{M}}\lim_{\substack{n_1\to+\infty\\ n_2\to+\infty}}\frac{1}{n_1}\sum_{i=1}^{n_1}\sum_{k=1}^{p}\left(\mathcal{M}\left(\theta_T^{(i)}, \delta_k\right) - \bar{M}_{T+\delta_k}^{(i)}\right)^2,$$

$\hat{\mathcal{M}}^\star\left(\theta_T, \delta_k\right) = M\left(\theta_T, \delta_k\right), \forall\ \theta_T : \boldsymbol{\pi}\left(\theta_T\right) \neq 0$ and $\delta_k \in \{\delta_1, ..., \delta_p\}$. QED.

[7] We assume here that, $\forall\ \theta_T : \boldsymbol{\pi}\left(\theta_T\right) \neq 0$ and $[0 : \delta_p]$, $\sigma_{T+\delta_k}$ has finite variance.

C.2 Convergence of the estimation procedure of the estimator function Θ

Proposition 2 *Let $\theta_{t_0}^{(1,1)},...,\theta_{t_0}^{(n_1,1)}$ be a sequence of i.i.d. random variables following $\boldsymbol{\pi}$, $\mathrm{D}^{(1)},...,\mathrm{D}^{(n_1)}$ a set of time-series such that $\mathrm{D}^{(i)}$ is generated from the M-RPDV associated with the θ-vector $\theta_{t_0}^{(i)}$, and $\theta_T^{(1,2)},...,\theta_T^{(n_1,2)}$ the set of values taken by θ at time t_N for each time series $\mathrm{D}^{(i)}$. If there exists $\Theta^\star$ such that for all $\mathrm{D}:\mathbb{P}_{\boldsymbol{\pi}}(\mathrm{D})\neq 0$, $\Theta^\star(\mathrm{D})$ is a Bayes estimator of θ_T under the posterior measure $\boldsymbol{\pi}$, then for any $\hat{\Theta}^\star$ solution to the optimization problem*

$$\arg\min_{\Theta}\lim_{n_1\to+\infty}\frac{1}{n_1}\sum_{i=1}^{n_1}L\bigg(\theta_T^{(i)},\Theta\left(\mathrm{D}^{(i)}\right)\bigg),$$

$\hat{\Theta}^\star(\mathrm{D})$ is a Bayes estimator of θ_T under the posterior measure $\boldsymbol{\pi}_{\mathrm{D}},\forall\ \mathrm{D}:\mathbb{P}_{\boldsymbol{\pi}}(\mathrm{D})\neq 0$.

Proof of proposition 2. The expectation of the cost under the prior measure $\boldsymbol{\pi}$ is defined by:

$$\mathbb{E}_{\boldsymbol{\pi}}\Big[L\big(\boldsymbol{\theta}_T,\Theta\left(\mathbf{D}\right)\big)\Big]=\int_{\mathbb{R}_+^{N\times 2}}\mathbb{E}_{\boldsymbol{\pi}_{\mathrm{D}}}\left[L\big(\boldsymbol{\theta}_T,\Theta\left(\mathrm{D}\right)\big)\right]\mathrm{d}\mathbb{P}_{\boldsymbol{\pi}}(\mathrm{D}).$$

Using this expression, and given density being by definition positive or zero, we have the following inequality:

$$\int_{\mathbb{R}_+^{N\times 2}}\Big(\min_{\Theta}\ \mathbb{E}_{\boldsymbol{\pi}_{\mathrm{D}}}\left[L\big(\boldsymbol{\theta}_T,\Theta\left(\mathrm{D}\right)\big)\right]\Big)\mathrm{d}\mathbb{P}_{\boldsymbol{\pi}}(\mathrm{D})\leq\min_{\Theta}\ \mathbb{E}_{\boldsymbol{\pi}}\Big[L\big(\boldsymbol{\theta}_T,\Theta\left(\mathbf{D}\right)\big)\Big].$$

It follows that if it exists $\Theta^\star$ such as $\forall\ \mathrm{D}:\mathbb{P}_{\boldsymbol{\pi}}(\mathrm{D})\neq 0$, $\Theta^\star(\mathrm{D})$ is a Bayes estimator of θ_T under the posterior measure $\boldsymbol{\pi}_{\mathrm{D}}$, if $\hat{\Theta}$ is solution to

$$\min_{\Theta}\ \mathbb{E}_{\boldsymbol{\pi}}\Big[L\big(\boldsymbol{\theta}_T,\Theta\big(\mathbf{D}\big)\Big],$$

$\hat{\Theta}(\mathrm{D})$ is a Bayes estimator of θ under the posterior measure $\boldsymbol{\pi}_{\mathrm{D}}$, $\forall\ \mathrm{D}:\mathbb{P}_{\boldsymbol{\pi}}(\mathrm{D})\neq 0$.

In addition, if $\mathrm{D}^{(1)},...,\mathrm{D}^{(n)}$ is a set of i.i.d. of time-series such as $\mathrm{D}^{(i)}\sim\mathbb{P}_{\boldsymbol{\pi}}$, by the law of large numbers

$$\lim_{n_3\to+\infty}\frac{1}{n_1}\sum_{i=1}^{n_1}L\bigg(\theta_T^{(i)},\Theta\left(\mathrm{D}^{(i)}\right)\bigg)=\mathbb{E}_{\boldsymbol{\pi}}\Big[L\big(\boldsymbol{\theta}_T,\Theta\left(\mathbf{D}\right)\big)\Big].$$

Combining the above propositions, if there exists $\Theta^\star$ such that for all $\mathrm{D}:\mathbb{P}_{\boldsymbol{\pi}}(\mathrm{D})\neq 0$, $\Theta^\star(\mathrm{D})$ is a Bayes estimator of θ_T under the posterior measure $\boldsymbol{\pi}$, then for any $\hat{\Theta}^\star$ solution to the optimization problem

$$\arg\min_{\Theta}\lim_{n_1\to+\infty}\frac{1}{n_1}\sum_{i=1}^{n_1}L\bigg(\theta_T^{(i)},\Theta\left(\mathrm{D}^{(i)}\right)\bigg),$$

$\hat{\Theta}^\star(\mathrm{D})$ is a Bayes estimator of θ_T under the posterior measure $\boldsymbol{\pi}_{\mathrm{D}}$, $\forall\ \mathrm{D}:\mathbb{P}_{\boldsymbol{\pi}}(\mathrm{D})\neq 0$. QED.

Appendix D Annex results

D.1 Value of the integral of $\hat{K}$ over $\mathbb{R}_+$

We compute the integral of $\hat{K}$ over $\mathbb{R}_+$:

$$\begin{aligned}\int_0^\infty \hat{K}(u)\mathrm{d}u &= \int_0^\infty \sum_{i=1}^n w_i\gamma_i e^{-\gamma_i(t-u)}\mathrm{d}u \\ &= \sum_{i=1}^n \left[w_i e^{-\gamma_i(t-u)}\right]_0^\infty \\ &= \sum_{i=1}^n w_i.\end{aligned}$$

D.2 Standard deviation of a BSS process

The avriance of an integral of the form $\int_0^\infty \hat{K}(u)\mathrm{d}W_u$ can be computed as follows:

$$\begin{aligned}\mathrm{Var}\left(\sum_{i=1}^n \int_0^\infty w_i\gamma_i e^{-\gamma_i u}\mathrm{d}W_u\right) &= \int_0^\infty \left(\sum_{i=1}^n w_i\gamma_i e^{-\gamma_i(t-u)}\right)^2 \mathrm{d}u \\ &= \int_0^\infty \sum_{i=1}^n\sum_{j=1}^n w_i w_j\gamma_i\gamma_j e^{-(\gamma_i+\gamma_j)(t-u)}\mathrm{d}u \\ &= \sum_{i=1}^n\sum_{j=1}^n \frac{w_i w_j\gamma_i\gamma_j}{\gamma_i+\gamma_j}.\end{aligned}$$

It follows that:

$$\mathrm{Std}\left(\sum_{i=1}^n \int_0^\infty w_i\gamma_i e^{-\gamma_i u}\mathrm{d}W_u\right) = \sqrt{\sum_{i=1}^n\sum_{j=1}^n \frac{w_i w_j\gamma_i\gamma_j}{\gamma_i+\gamma_j}e^{-(\gamma_i+\gamma_j)}}.$$

D.3 The variance of the volatility process

The variance of the volatility process is equal to:

$$\begin{aligned}\mathrm{Var}\left(\sigma_{T+\delta}\right) &= \mathrm{Var}\left(\beta_0 + \beta_1 R_{1,T+\delta} + \beta_2\sqrt{R_{2,T+\delta}}\right) \\ &= (\beta_1)^2\mathrm{Var}\left(R_{1,T+\delta}\right) + (\beta_2)^2\mathrm{Var}\left(\sqrt{R_{2,T+\delta}}\right) + 2\beta_1\beta_2\rho_{T+\delta}\sqrt{\mathrm{Var}\left(R_{1,T+\delta}\right)\mathrm{Var}\left(\sqrt{R_{2,T+\delta}}\right)}.\end{aligned}$$