risks

MDPI

Article

Deep Hedging under Rough Volatility

Blanka Horvath [1], Josef Teichmann [2] and Žan Žurič [3,*]

[1] Technische Universität München, King's College London and The Alan Turing Institute, London WC2R 2LS, UK; Blanka.Horvath@kcl.ac.uk
[2] ETH Zürich, 8092 Zürich, Switzerland; josef.teichmann@math.ethz.ch
[3] Faculty of Natural Science, Imperial College London, London SW7 2AZ, UK
[*] Correspondence: z.zuric19@imperial.ac.uk

Abstract: We investigate the performance of the Deep Hedging framework under training paths beyond the (finite dimensional) Markovian setup. In particular, we analyse the hedging performance of the original architecture under rough volatility models in view of existing theoretical results for those. Furthermore, we suggest parsimonious but suitable network architectures capable of capturing the non-Markoviantity of time-series. We also analyse the hedging behaviour in these models in terms of Profit and Loss (P&L) distributions and draw comparisons to jump diffusion models if the rebalancing frequency is realistically small.

Keywords: deep learning; rough volatility; hedging

check for updates

Citation: Horvath, Blanka, Josef Teichmann, and Žan Žurič. 2021. Deep Hedging under Rough Volatility. *Risks* 9: 138. https://doi.org/10.3390/risks9070138

Academic Editor: Alexander Szimayer

Received: 25 February 2021
Accepted: 8 June 2021
Published: 20 July 2021

1. Introduction

Deep learning has undoubtedly had a major impact on financial modelling in the past years and has pushed the boundaries of the challenges that can be tackled: not only can existing problems be solved faster and more efficiently (Bayer et al. 2019; Benth et al. 2020; Cuchiero et al. 2020; Gierjatowicz et al. 2020; Hernandez 2016; Horvath et al. 2021; Liu et al. 2019; Ruf and Wang 2020), but deep learning also allows us to derive (approximative) solutions to optimisations problems (Buehler et al. 2019), where classical solutions have so far been limited in scope and generality. Additionally these approaches are fundamentally data driven, which makes them particularly attractive from business perspectives.

It comes as no surprise that the more similar (or "representative") the data presented to the network in the training phase is to the (unseen) test data that the network is later applied to, the better is the performance of the hedging network on real data in terms of Profit and Loss (P&L). It is also unsurprising that, as markets shift sufficiently far away from a presented regime into new, previously unseen territories, the hedging networks may have to be retrained to adapt to the new environment.

In the current paper we go a step further than just presenting an ad hoc well-chosen market simulator (see Buehler et al. 2020b, 2020a; Henry-Labordere 2019; Wiese et al. 2019, 2020; Cuchiero et al. 2020; Kondratyev and Schwarz 2019; Xu et al. 2020): we investigate a situation where the relevant data are *structurally* so different from the original Markovian setup that it calls for an adjustment of the model architecture itself. In a well-controlled synthetic data environment, we study the behaviour of the hedging engine as relevant properties of the data change.

A good candidate class of models comprises the rough volatility models, which have experienced a surge of research interest in recent years starting with the seminal papers of (Alòs et al. 2007; Bayer et al. 2015; Fukasawa 2010; Gatheral et al. 2018), where a crucial difference was introduced in comparison to classical stochastic models, due to their non-Markovianity. The main difference is parametrized by the so-called Hurst parameter $H \in (0,1)$ in the volatility process, which models the "*memoryness*" and the roughness of the driving fractional Brownian motion. These models reflect much more closely the stylized facts and essential properties of the financial markets. Therefore, now, after several years of research of their asymptotic and numerical behaviour, they provide ideal synthetic data sets to test different machine learning models.

More specifically, in our case, we used synthetic data generated from a rough volatility model with varying levels of the Hurst parameter. In its initial setup, we set the Hurst parameter to $H = 1/2$, which reflects a classical (finite dimensional) Markovian case, which is well-aligned with the majority of the most popular classical financial market models, such as, e.g., the Heston model, which the initial version of the deep hedging results were demonstrated on. We then gradually altered the level of the Hurst parameter to (rough) levels around $H \approx 0.1$, which more realistically reflects market reality as observed in (Alòs et al. 2007; Bolko et al. 2020; Fukasawa 2010; Gatheral et al. 2018; Livieri et al. 2018), thereby introducing a non-Markovian *memory* into the volatility process.

Since rough volatility models are known to reflect the reality of financial markets (as well as the stylised statistical facts) better than classical, finite-dimensional Markovian models do, our findings also give an indication of how a naive application of model architectures to real data could lead to substantial errors. With this, our study allows us to make a number of interesting observations about deep hedging and the data that it is applied to: apart from drawing parallels between discretely observed rough volatility models and jump processes, our findings highlight the need to rethink (or carefully design) risk management frameworks of deep learning models as significant structural shifts in the data occur.

To recap, we compare different methods for hedging under rough volatility models. More precisely, we analyse the perfect hedge for the rBergomi model from (Viens and Zhang 2019) and its performance against the deep hedging scheme in (Buehler et al. 2019), which had to be adapted to a non-Markovian framework. We are particularly interested in the dependence of the P&L on the Hurst parameter and conclude that the deep hedge with our fRNN architecture performs better than the discretised perfect hedge for all H. Moreover, we find that the hedging P&L distributions for low H are highly left-skewed and have significant mass in the left tail under the model hedge as well as the deep hedge. Increasing hedging frequency was explored to mitigate the heavy losses in cases when H is close to zero. Intriguingly, slow response to increased hedging frequency and left-skewed P&L distribution under the rough Bergomi are also characteristic for delta hedges under jump diffusion models (Sepp 2012).

The paper is organised as follows: Section 2 recalls the setup of the original deep hedging framework used in (Buehler et al. 2019). Section 3 gives a brief reminder on hedging under rough volatility models and compares the performance of (feed-forward) hedging network on a rough Bergomi model compared to a theoretically derived model hedge. In Sections 3.3 and 3.4, we draw conclusions with respect to the model architecture, and in Section 3.5, we propose a new architecture that is better suited to the data. Section 4 lays out the hedging under the new architecture and draws conclusions to the existing literature, which outlines some parallels between (continuous) rough volatility models and jump processes in this setting, while Section 5 summarizes our conclusions.

2. Setup and Notation

We adopt the setting in (Buehler et al. 2019) and consider a discrete finite-time financial market with time horizon $[0, T]$ for some $T \in (0, \infty)$ and a finite number of trading dates $0 = t_0 < t_1 < \cdots < t_n = T$, $n \in \mathbb{N}$. We work on a discrete probability space $(\overline{\Omega}, \overline{\mathcal{F}}, \mathbb{P})$, with $\overline{\Omega} = \{\omega_1, \ldots, \omega_N\}$ and a probability measure $\mathbb{P}$ for which $\mathbb{P}[\{\omega_i\}] > 0$ for all $i \in \{1, \ldots, N\}$ and $N \in \mathbb{N}$. Additionally, we fix the notation $\mathcal{X} := \{X : \overline{\Omega} \to \mathbb{R}\}$ for the set of all $\mathbb{R}$-valued random variables on $\overline{\Omega}$. Later we shall consider a continuous setup $(\Omega, \mathcal{F}, \mathbb{P})$, of which the discrete setup will be a precise approximation via Monte Carlo sampling and time discretisation.

The filtration $\overline{\mathbb{F}} = \left(\overline{\mathcal{F}}_k\right)_{k=0,\ldots,n}$ is generated by the $\mathbb{R}^r$-valued information process $(I_k)_{k=0,\ldots,n}$ for some $n, r \in \mathbb{N}$. For any $k \in \{0, \ldots, n\}$, the variable I_k denotes all available new market information at time t_k and $\overline{\mathcal{F}}_k$ represents all available market information up to time t_k.

The market contains $d \in \mathbb{N}$ financial instruments, which can be used for hedging, with mid-prices given by an $\mathbb{R}^d$-valued $\overline{\mathbb{F}}$-adapted stochastic process $S = (S_k)_{k=0,\ldots,n}$. In order to hedge a claim $Z : \overline{\Omega} \to \mathbb{R}$, we may trade in S according to $\mathbb{R}^d$-valued $\overline{\mathbb{F}}$-adapted processes (strategies), which we denote by $\delta := (\delta_k)_{k=0,\ldots,n-1}$ with $\delta_{-1} = \delta_n := 0$ for notational convenience, where $\delta_k = (\delta_k^1, \ldots, \delta_k^d)$. Here, δ_k^i denotes the agent's holdings of the i-th asset at time t_k. We denote the initial cash injected at time t_0 by $p_0 > 0$.

Furthermore, in order to allow for proportional trading costs, for every time t_k and change in position $s \in \mathbb{R}^d$ we consider costs $c_k : s \mapsto c \in [0, \infty)$, where c_k is $\overline{\mathcal{F}}_k$-adapted, upper semi continuous and for which $c_k(0) = 0$ for all $k \in \{0, \ldots, n\}$. The total costs up to time T, when trading according to a trading strategy δ are denoted by $C_T(\delta) := \sum_{k=0}^{n} c_k s_{k-1}(\delta_k - \delta_{k-1})$ again with the notation $s_{-1} := 0$. Finally, we denote by $\mathcal{H}$ a set of all admissible trading strategies.

We consider optimality of hedging under *convex risk measures* (see, e.g., Föllmer and Schied (2016) for the definition) as in (Buehler et al. 2019; Ilhan et al. 2009; Xu 2005). Let us for a moment consider the following problem. Say the agent's terminal portfolio value for a contingent claim $Z \in \mathcal{X}$ at T is given as

$$\mathrm{P\&L}(p_0, \delta, Z; T) := -Z + p_0 + (\delta \cdot S)_T - C_T(\delta), \tag{1}$$

where $(\delta \cdot S)_T := \sum_{k=0}^{n-1} \delta_k \cdot (S_{k+1} - S_k)$ is the discrete stochastic integral and $p_0 > 0$ denotes the expectation of Z with respect to $\mathbb{Q}$, i.e., the given risk-neutral price. The price is exogenously given, meaning either quoted in the market or calculated using some other pricing method, separate from our methodology. In complete markets, there exists $\delta \in \mathcal{H}$ such that $\mathrm{P\&L}(p_0, \delta, Z; T) = 0$ for any $Z \in \mathcal{X}$; however, this cannot be said for incomplete markets with frictions. In this setting, the agent has to accept some risk and specify an optimality criterion. Now in the case of no trading costs, there is an alternative view point of variance optimal hedging (Schweizer 1995), which will be taken in this paper. Consider an equivalent pricing measure $\mathbb{Q}$ of our financial market; then, we can also minimise the variance

$$\pi(Z) = \inf_{\delta \in \mathcal{H}} \mathbb{E}\left[(-Z + (\delta \cdot S)_T + p_0)^2\right], \tag{2}$$

In the rest of this paper, the above optimisation (2) problem and corresponding optimisers are considered in terms of their numerical approximation in the framework of *hedging in a neural network setting* as formulated in (Buehler et al. 2019). In the remainder of this section, we recall the notation and definitions to formulate this approximation property and the conditions that ensure its validity.

Definition 1 (Set of Neural Networks (NNs) with a fixed activation function). *We denote by* $\mathcal{NN}^{\sigma}_{\infty,d_0,d_1}$ *the set of all NNs mapping from* $\mathbb{R}^{d_0} \to \mathbb{R}^{d_1}$ *with a fixed activation function* σ. *The set* $\{\mathcal{NN}^{\sigma}_{M,d_0,d_1}\}_{M\in\mathbb{N}}$ *is then a sequence of subsets in* $\mathcal{NN}_{\infty,d_0,d_1}$ *for which* $\mathcal{NN}^{\sigma}_{M,d_0,d_1} = \{F^{\theta} : \theta \in \Theta_{M,d_0,d_1}\}$ *with* $\Theta_{M,d_0,d_1} \subset \mathbb{R}^{q}$ *for some* $q(M), M \in \mathbb{N}$.

Definition 2. *We call* $\mathcal{H}_M \subset \mathcal{H}$ *the set of unconstrained neural network trading strategies:*

$$\begin{aligned}\mathcal{H}_M &= \left\{(\delta_k)_{k=0,\ldots,n-1} \in \mathcal{H} : \delta_k = F_k(I_0, \ldots, \delta_{k-1}), F_k \in \mathcal{NN}_{M,r(k+1)+d,d}\right\} \\ &= \left\{(\delta_k)_{k=0,\ldots,n-1} \in \mathcal{H} : \delta_k = F_k^{\theta_k}(I_0, \ldots, \delta_{k-1}), \theta_k \in \Theta_{M,r(k+1)+d,d}\right\}\end{aligned} \tag{3}$$

We now replace the set $\mathcal{H}$ in (2) by the finite subset $\mathcal{H}_M \subset \mathcal{H}$. The optimisation problem then becomes

$$\begin{aligned} \pi^M(Z) &:= \inf_{\delta \in \mathcal{H}_M} \mathbb{E}\Big[(-Z + (\delta \cdot S)_T + p_0)^2\Big] \\ &= \inf_{\theta \in \Theta_M} \mathbb{E}\Big[(-Z + (\delta^\theta \cdot S)_T + p_0)^2\Big] \end{aligned} \tag{4}$$

where $\Theta_M = \prod_{k=0}^{n-1} \Theta_{M,r(k+1)+d,d}$ denotes the network parameters from Definition 2. With (3), (4) and Remark 1, the potentially infinite-dimensional problem of finding an optimal hedging strategy is therefore reduced to a finite-dimension and corresponds to finding the optimal NN parameters for the problem (4). Notice that in general, i.e., before time and space discretisation, the problem is inherently infinitely dimensional. We shall always consider appropriate discretisations, which are sufficiently close to the continuous time model by results from numerical analysis, on which we apply the above theory.

Remark 1. *Note that in the above, we do not assume that S is an $(\overline{\mathbb{F}}, \mathbb{P})$-Markov process and that the contingent claim is of the form $Z := g(S_T)$ for a pay-off function $g : \mathbb{R}^d \to \mathbb{R}$. This would allow us to write the optimal strategy $\delta_k = f_k(I_k, \delta_{k-1})$ for some $f_k : \mathbb{R}^{r+d} \to \mathbb{R}^d$.*

The next proposition, which is a direct use of Lemma 4.4 in (Buehler et al. 2019), recalls the central approximation property, which states that the optimal trading strategy $\delta^* \in \mathcal{H}$ that minimizes (2) can be approximated by a semi-recurrent neural network of the form Figure 1 in the sense that the functional $\pi^M(Z)$ converges to $\pi(Z)$ as M becomes large.

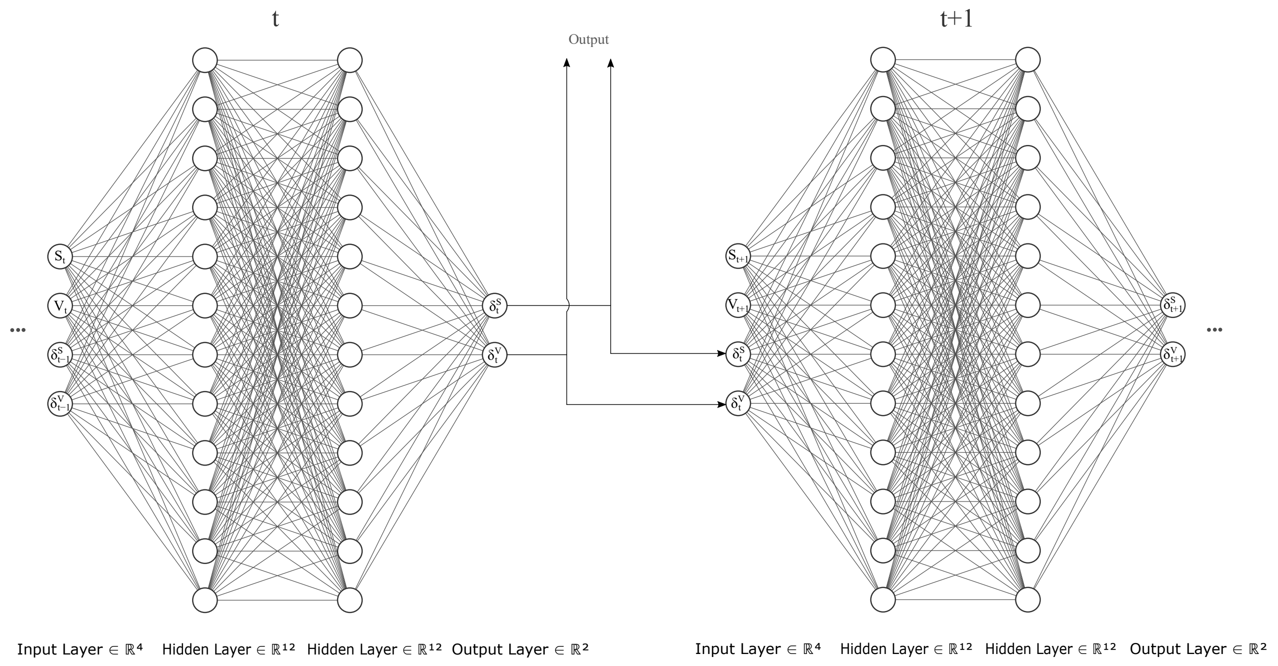


Figure 1. Original Network Archiecture.

Proposition 1. *Define $\mathcal{H}_M$ as in (3) and π^M as in (4). Then for any $Z \in \mathcal{X}$*

$$\lim_{M \to \infty} \pi^M(Z) = \pi(Z),$$

where $\pi(Z)$ denotes the optimal solution of the original optimisaton problem (2).

Corollary 1. *As a direct consequence, the optimal neural network trading strategy δ^{θ^*} that minimises* (4) *converges to the minimiser of* (2) *in the sense that*

$$\lim_{M\to\infty} \delta^{\theta^*_M} = \delta^*,$$

where $\theta^*_M \in \Theta_M$ are the optimal neural network parameters for $M \in \mathbb{N}$.

In (Buehler et al. 2019), this approximation property is demonstrated for Black–Scholes and Heston models both in their original form and in variants including market frictions such as transaction costs. These results demonstrate how deep hedging, which allows us to take a leap beyond the classical type, results in scenarios where the Markovian structure is preserved.

A natural question to ask is how the approximation property of the neural network is affected if the assumption of Markovian structure of the underlying process is no longer satisfied. Rough volatility models (Alòs et al. 2007; Bayer et al. 2015; Fukasawa 2010; Gatheral et al. 2018) represent such a class of non-Markovian models. It is also well-established in a series of recent articles, including the aforementioned works, that rough volatility dynamics are superior to standard Markovian models (such as Black–Scholes and Heston) in terms of reflecting market reality and also that rough volatility models are superior to a number of in terms of allowing close fits to market data.

By taking hedging behaviour under rough volatility models under a closer examination, we gain insight into the non-Markovian aspects of markets in a controlled numerical setting: varying the Hurst parameter $H \in (0,1)$ of the process (see Gatheral et al. (2018)), which governs the deviation from the Markovian setting in a general fractional (or rough) volatility framework, enables us to control for the influence of the Markovianity assumption on the hedging performance of the deep neural network. Therefore, in this work, we investigate the effect of the loss of Markovianity property of the underlying stochastic process by considering market dynamics that are governed in a rough volatility setting. With this in mind, by applying the original feedforward network architecture to a more realistic model class (represented by rough volatility models), we in particular demonstrate how the choice of the network architecture may affect the performance of the deep hedging framework and how it could potentially break down on real-life data. We also note in passing that the approach we take can be applied as a simple routine sanity check for model governance of deep learning models on real data:

- Take a well-understood model class that generalises the modelling to more realistic market scenarios, but where the generalisation no longer satisfies assumptions made in the original architecture.
- Test the robustness of the method if the assumption is violated by controlling for the error as the deviation from the assumption increases.
- Modify the network architecture accordingly if necessary.

3. Hedging and Network Architectures for Rough Volatility

3.1. Hedging under Rough Volatility

Let us now consider the problem of hedging under rough volatility models in general, with an aim to present a theoretical solution, which we use as a benchmark of the deep hedging approach. In this section, we introduce a new filtered probability space $(\Omega, \mathcal{F}, \mathbb{Q}, \mathbb{F})$, where $\mathbb{F} := \{\mathcal{F}_t\}_{0\le t\le T}$ is now a continuous filtration.[1] We know that for a Markovian process of the form

$$\tilde{X}_t = x + \int_0^t b(r, X_r)dr + \int_0^t \sigma(r, X_r)dW_r,$$

where b and σ satisfy suitable conditions. Then the price of a contingent claim $\tilde{Z}_t := \mathbb{E}[g(\tilde{X}_T)|\mathcal{F}_t]$ can be written as

$$\tilde{Z}_t = u(t, \tilde{X}_t),$$

where u solves a parabolic PDE by the Feynman–Kac formula (Kac 1949). However, it was shown in (Bayer et al. 2015) that rough volatility models are not finite-dimensional Markovian, and we therefore have to consider a more general process X and assume it to be a solution to the d-dimensional Volterra SDE:

$$X_t = x + \int_0^t b(t; r, X_.)dr + \int_0^t \sigma(t; r, X_.)dW_r, \qquad t \in [0, T], \tag{5}$$

where W is a m-dimensional standard Brownian motion, $b \in \mathbb{R}^d$ and $\sigma \in \mathbb{R}^{m\times d}$. Both are adapted in a sense that for $\varphi = b, \sigma$ it holds $\varphi(t; r, X_.) = \varphi(t; r, X_{r\wedge.})$.

In this general non-Markovian framework, the contingent claim in the form $Z_t := \mathbb{E}[g(X_T)|\mathcal{F}_t]$ will depend on the entire history of the process $X := (X_t)_{t\geq 0}$ up to time t and not just on the value of the process at that time, i.e.,

$$Z_t = u(t, X_{[0,t]}) \quad \text{with notation} \quad X_{[0,t]} := \{X_r\}_{r\in[0,t]},$$

where u this time solves a Path-dependent PDE (PPDE). The setting where X is a *semi-martingale* has already been explored in, e.g., (Cont and Fournié 2013; Dupire 2019). Be that as it may, we know that fBm is *not* a semi-martingale in general, and as a consequence, the volatility process is not a semi-martingale. Viens and Zhang (2019) are able to cast the problem back in to the semi-martingale framework by rewriting X_t as a orthogonal decomposition to an auxiliary process Θ_t and a process I_t, which is independent of the filtration

$$\begin{aligned} X_s = x &+ \int_0^t b(s; r, X_.)dr + \int_0^t \sigma(s; r, X_.)dW_r \\ &+ \int_t^s b(s; r, X_.)dr + \int_t^s \sigma(s; r, X_.)dW_r \end{aligned} \tag{6}$$

$$=: x + \Theta_s^t + I_s^t \tag{7}$$

for $0 \leq t \leq s$. By exploiting the semi-martingale property of Θ, they go on to show that the contingent claim can be expressed as a solution of a PPDE

$$Z_t = u(t, X_{[0,t)} \otimes_t \Theta^t_{[t,T]}), \tag{8}$$

where $\otimes_t$ denotes concatenation of a path at time t. Moreover, they develop an Itô-type formula for a general non-Markovian process X_t from (5), which we present in the Appendix B. We have to consider different additional hedging instruments to complete such markets. Most convenient from a theoretical as well as from a practical perspective are variance swaps with different maturities, which are, according to the Breeden–Litzenberger formula, just a linear superposition of plain vanilla European calls.

3.2. *The Rough Bergomi Model (rBergomi)*

As an example we consider the rBergomi with a constant initial forward variance curve $\xi_0(t) = V_0$:

$$S_t = S_0 + \int_0^t S_r\sqrt{V_r}\left[\sqrt{1-\rho^2}dB_r + \rho dW_r\right] \tag{9a}$$

$$V_t = V_0\mathcal{E}\left(\sqrt{2H}\nu\int_0^t (t-r)^{H-\frac{1}{2}}dW_r\right), \qquad V_0{=}v_0{>}0, \tag{9b}$$

The model fits into the affine structure of our Volterra SDE in (5) after a simple log-transformation of the volatility process. In this case, we take our auxiliary process to be

$$\Theta_s^t = \sqrt{2H}\nu \int_0^t (s-r)^{H-\frac{1}{2}} dW_r, \qquad t < s. \tag{10}$$

It is easy to check that Θ_s^t is a true martingale for a fixed s. The option price dynamics are obtained by using the Functional Itô formula in (A3). From this, the perfect hedge in terms of a forward variance $\hat{\Theta}_T^t$ with maturity T and a stock S_t follows:

$$dZ_t = \partial_x u(t, S_t, \Theta^t_{[t,T]}) dS_t + \frac{(T-t)^{\frac{1}{2}-H}}{\hat{\Theta}_T^t} \left\langle \partial_\omega u(t, S_t, \Theta^t_{[t,T]}), a^t \right\rangle d\hat{\Theta}_T^t \tag{11}$$

with $a_s^t = (s-t)^{H-\frac{1}{2}}$. The path-wise derivative in (11) is the Gateaux derivative along the direction a^t. For more details and discretisation of the Gateaux derivative, see Appendix A.

3.3. Performance of the Deep Hedging Scheme (with the Original Feedforward Architecture) Compared to the Model Hedge under rBergomi

We choose to hedge a plain vanilla called option $Z_T := \max(S_T - K, 0)$ with $K = 100$ and a monthly maturity $T = 30/365$. The hedging portfolio consists of a stock S with $S_0 = 100$ and a forward variance with maturity $T_{\text{Fwd}} = 45/365$ and is rebalanced daily. For the rBergomi model, forward variance is equal to

$$\hat{\Theta}^t_{T_{\text{Fwd}}} := \mathbb{E}_{\mathbb{Q}}\left[\int_0^{T_{\text{Fwd}}} V_s ds \middle| \mathcal{F}_t\right] = V_0 \exp\left[\Theta^t_{T_{\text{Fwd}}} + \frac{1}{2}\nu^2\left[(T_{\text{Fwd}} - t)^{2H} - T_{\text{Fwd}}^{2H}\right]\right], \tag{12}$$

with $\Theta^t_{T_{\text{Fwd}}}$ defined as in (10). Applying classical Itô's Lemma to $\hat{\Theta}^t_{T_{\text{Fwd}}} = \hat{\Theta}^t_{T_{\text{Fwd}}}(t, \Theta^t_{T_{\text{Fwd}}})$ yields the dynamics of the forward variance under the rough Bergomi

$$d\hat{\Theta}^t_{T_{\text{Fwd}}} = \hat{\Theta}^t_{T_{\text{Fwd}}} \sqrt{2H}\nu (T_{\text{Fwd}} - t)^{H-\frac{1}{2}} dW_t, \tag{13}$$

which is well defined for $t \in [0, T_{\text{Fwd}})$. Therefore, choosing the maturity of the forward variance to be longer than the option maturity allows us to avoid the singularity as $t \to T$. In practice, this would correspond to hedging with a forward variance with a slightly longer maturity than that of the option.

For the simulation of the forward variance, we used the Euler–Mayurama method, whereas paths of the volatility process were simulated with the "turbo-charged" version of the hybrid scheme proposed in (Bennedsen et al. 2017; McCrickerd and Pakkanen 2018). The parameters were chosen such that they describe a typical market scenario with a flat forward variance: $\xi_0 = 0.235 \times 0.235$, $\nu = 1.9$ and $\rho = -0.7$. We were particularly interested in the dependence of the hedging loss on the Hurst parameter. Finally, a quadratic loss function was chosen, and the minimising objective was therefore

$$\pi(Z) = \inf_{\delta^\theta \in \mathcal{H}_M} \mathbb{E}\left[(-Z + p_0 + (\delta^\theta \cdot S)_T)^2\right]$$

where price p_0 was obtained with a Monte-Carlo simulation (e.g., for $H = 0.10$, $p_0 = 2.39$).

Next we implement the perfect hedge from (11) (the details of the discretisation of the Gateaux derivative are presented in Appendix C). For evaluation of the option price, we once again use Monte-Carlo, this time with the generating parameters. In practice, we would calibrate the parameters to the market data. Perfect hedge was implemented on the sample of 10^3 different paths for the same parameters as in the deep hedging case. The results of both hedges under quadratic loss for different Hurst parameters are shown in Table 1. We also take a closer look at the P&L distributions of the deep hedge as well as the model hedge for $H = 0.10$ in Figure 2. Curiously enough, the distributions are very similar to each other. The deep hedge seems to have slightly thinner tails, which is interesting, considering the semi-recurrent architecture makes a strong assumption of Markovianity of the underlying process.

Table 1. Comparison of the quadratic loss between model and deep hedges trained 75 epochs for different H.

	Quadratic Hedging Loss	
H	**Model Hedge**	**Deep Hedge**
0.10	1.45	1.16 (*1.12)
0.20	0.52	0.67
0.30	0.34	0.46
0.40	0.24	0.36

*—on 200 epochs.

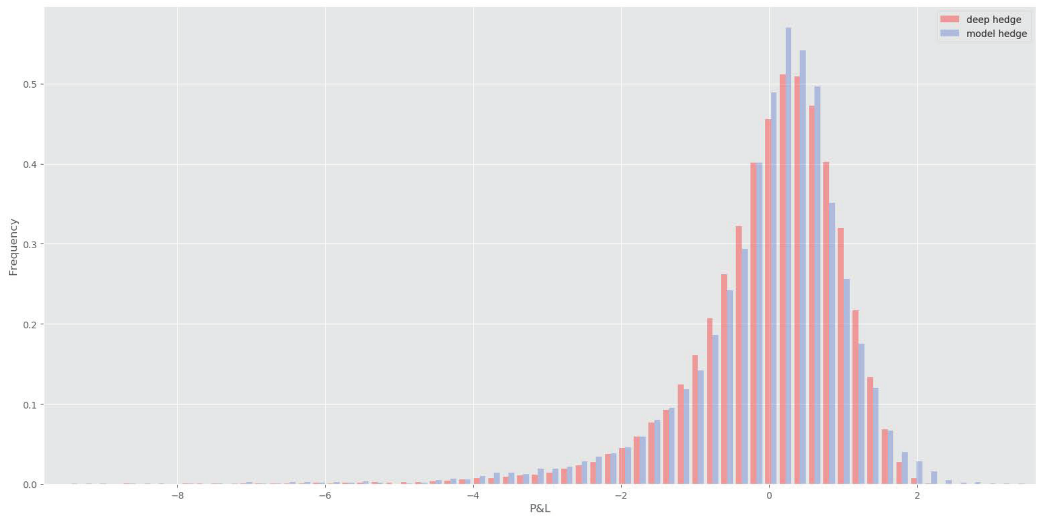


Figure 2. rBergomi model hedge (blue) compared to the deep hedge (red) trained on 75 epochs on rBergomi paths with $H = 0.10$. Note that the option price is only $p_0 = 2.39$ and that such a hedge can result in a *substantial* loss.

Indicators that the assumption of finite dimensional Markovianity is violated might be the heavy left tail of the P&L distribution as well as relatively high hedging losses. This prompted us to question the semi-recurrent architecture and devise a way to relax the Markov assumption on the underlying. Note that the heavy tails of these distributions may also imply a link to jump diffusion models. We expand on this in Section 4.3.

3.4. Implications on the Network Architecture

As discussed before, in (Buehler et al. 2019) authors heavily rely on Remark 1, where they use the Markov property of the underlying process in order to write the trading strategy at time t_k as a function of the information process at t_k and trading strategy in the previous time step $k-1$. Of course, in the case of rough volatility models, one would have to include the entire history of the information process up to t_k in order to get the hedge at that time. However, this would result in numerically infeasible scheme. To illustrate this, take for example a single vanilla call option with maturity $T = 30/365$, where we hedge daily under say the rough Bergomi model. In the 30th time step, the number of input nodes of the NN cell F^{θ}_{30} would be $30 \times 2 + 2 = 62$ or if we hedged twice a day $30 \times 2 \times 2 + 2 = 122$. Obviously, this scheme quickly becomes very computationally expensive even for a single option with a short maturity.

The fBm in (9b) can be written as a linear functional of a Markov process, albeit an infinite-dimensional one. Therefore, if the original Markovian-based architecture can be applied to this setting, we would expect to recover the Hurst parameter also from a Markovian-based sampling procedure, justifying the continued use of the original feed forward architecture. This, however, is not the case: it is known that fBm in (9b) can be rewritten as an infinite-dimensional Markov process in the following way. Take the Riemann–Liouville representation of fBm:

$$B_t^H := \frac{1}{\Gamma(H+\frac{1}{2})}\left(\int_0^t (t-s)^{H-\frac{1}{2}} dW_s\right),$$

where W is a standard Brownian motion. Using the fact that for $\alpha \in (0,1)$ and fixed $x \in [0,\infty)$,

$$\frac{(t-s)^{\alpha-1}}{\Gamma(\alpha)} = \int_0^\infty e^{-(t-s)x}\mu(dx), \quad \text{with} \quad \mu(dx) = \frac{dx}{x^\alpha \Gamma(\alpha)\Gamma(1-\alpha)} \tag{14}$$

we obtain by the Fubini Theorem

$$\begin{aligned} B_t^H &= \int_0^t \int_0^\infty e^{-(t-s)x}\mu(dx)dW_s \\ &= \int_0^\infty \int_0^t e^{-(t-s)x} dW_s \mu(dx) \\ &= \int_0^\infty Y_t^x \mu(dx) \end{aligned}$$

with $Y_t^x = \int_0^t e^{-(t-s)x} dW_s$. Observe that for a fixed $x \in [0,\infty)$, $(Y_t^x)_{t\geq 0}$ is an *Ornstein-Uhlenbeck process* with mean reversion zero and mean reversion speed x, i.e., Gaussian semi-martingale Markov process solution with the dynamics of

$$dY_t^x = -xY_t^x dt + dW_t.$$

Therefore, we have shown that B^H is a linear functional of the infinite-dimensional Markov process. Being able to simulate from Y_t^x would mean that we can still use the architecture in Figure 1, even for a rough process. A numerical simulation scheme for such a process is presented in (Carmona et al. 1998). Regrettably, the estimated Hurst parameter[2] from the generated time series stayed around $H \approx 0.5$ for any chosen input Hurst parameter to the simulation scheme. For a fixed time-step Δt, the scheme does not produce the desired roughness, even if we used a number of OU terms well beyond what authors propose. We believe that this is because the scheme is only valid in the limit, i.e., when the number of terms goes to infnity and $\Delta t \to 0$. Failure to recover the Hurst parameter, together with the fact that the architecture does not allow for any path dependent contingent claims, encouraged us to change the Neural Network architecture itself.

3.5. Proposed Fully Recurrent Architecture

Based on the above insights, we hence modify the original architecture. In this section, we suggest an alternative architecture and show that it is well-suited to the problem. When constructing a new architecture, we would like to change the semi-recurrent structure as little as possible for our purpose, since it seems to perform very well in the Markovian cases. However, in order to account for non-Markovianity, we propose a completely recurrent structure.[3] To that end, we now introduce a hidden state $\tilde{\delta}_k = (\tilde{\delta}_{k-1}^S, \tilde{\delta}_{k-1}^V)$ with $\tilde{\delta}_0 = 0$, which is passed to the cell at time t_k along the information process I_k. Therefore, instead of adding layers to each of the state transitions separately as in (Pascanu et al. 2014), we simply concatenate the input vector I_k with the hidden state vector and feed it into a the neural network cell $F_k^\theta(\cdot)$:

$$F_k^\theta(I_k \oplus \tilde{\delta}_k) = \delta_k \oplus \tilde{\delta}_k$$

For the visual representation, see Figure 3. The *output* is still a trading strategy $\delta_k = (\delta_k^S, \delta_k^V)$, and it is evaluated on the same objective function as before:

$$\mathcal{L}(\theta) := \mathbb{E}\left[\left(-Z + p_0 + (\delta^\theta \cdot S)_T\right)^2\right],$$

whereas the hidden state $\tilde{\delta}_k$ is passed forward to the next cell F_{k+1}^θ. These states can take any value and are not restricted to having any meaningful financial representation as trading strategies do. We illustrate the fact that the fRNN architecture is truly recurrent by showing how hidden states are able to encode the relevant history of the information process. Let us say for example that the information process $I_k = (S_k^1, S_k^2)$ is simply the price of both hedging instruments. The strategies at time t_k now do not depend on the asset holdings δ_{k-1}^x, but on $\tilde{\delta}_{k-1}^x$ for $x \in \{S, V\}$:

$$\delta_k^x := \delta_k^x(S_k^1, S_k^2, \tilde{\delta}_{k-1}^S, \tilde{\delta}_{k-1}^V).$$

For some $\mathcal{F}_{k-1}$-measurable function g_{k-1}, it holds for the hidden states themselves that

$$\tilde{\delta}_{k-1}^x = g_{k-1}^x(S_{k-1}^1, S_{k-1}^2, \tilde{\delta}_{k-2}^S, \tilde{\delta}_{k-2}^V).$$

Recursively, the hidden states are implicitly dependent on the entire history

$$\tilde{\delta}_{k-1}^x = g_{\mathcal{NN}}^x(S_{k-1}^1, S_{k-1}^2, S_{k-2}^1, S_{k-2}^2, \ldots, S_0^1, S_0^2, \tilde{\delta}_0),$$

where $g_{\mathcal{NN}}^x$ is again $\mathcal{F}_{k-1}$-measurable. Structuring the network this way, we hope that the hidden states at time t_k will be able to encode the history of the information process $I_0, \ldots, I_k$. More precisely, what we expect is that the network will itself learn the function $g_{\mathcal{NN}}^x : \mathbb{R}^{2k} \to \mathbb{R}$ for $x \in \{S, V\}$ and with that the path dependency inherent to the liability we are trying to hedge.

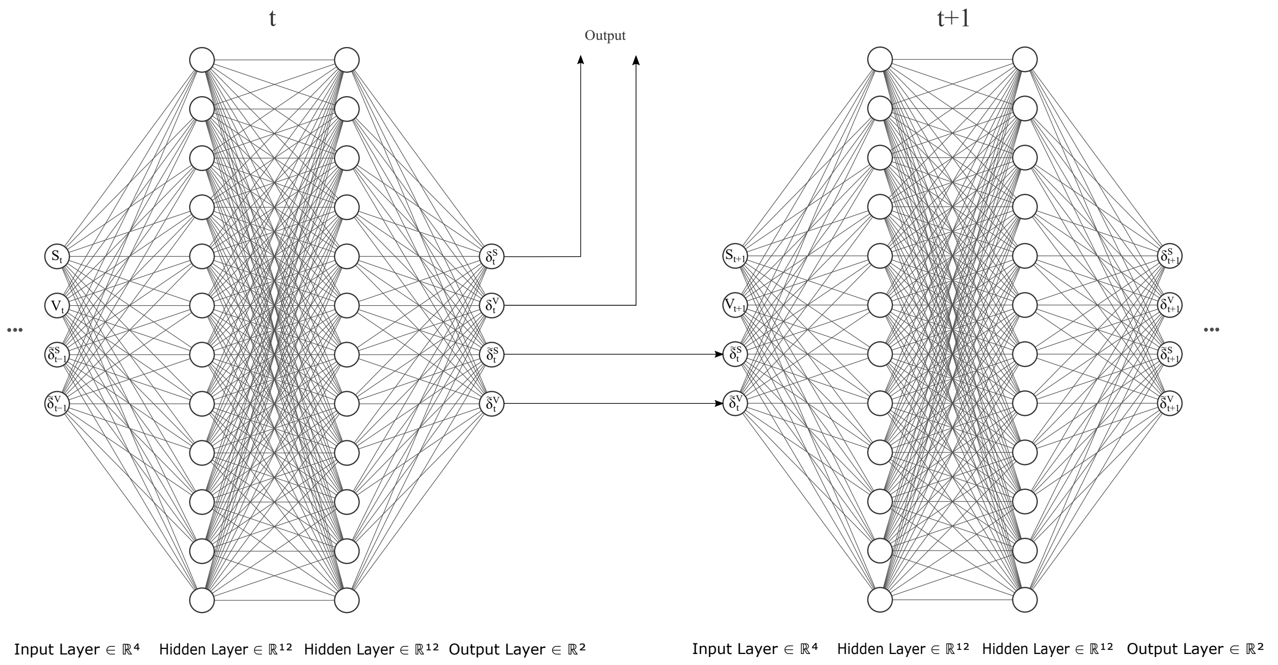


Figure 3. Fully Recurrent Neural Network (fRNN) Architecture. The recurrent structure of this architecture is clearly visible as hidden states are passed on to the next cell at each time step.

Remark 2. *We remark that in order to account for the history of the information process one could also write the trading strategy as*

$$\delta_k := \delta_k(I_k, \tilde{I}_{k-1}),$$

where $\tilde{I}^n_{k-1} = \{I_i\}^{k-1}_{i=(k-1)-n}$ *is the history of the information process with a window length of* $n \in \{1, \dots, k-1\}$*. However, in this case, we would have to optimise the window length and would inevitably face an accuracy and computational efficiency trade-off. We would rather outsource this task to the neural network.*

Remark 3. *While we do think the LSTM architecture (Hochreiter and Schmidhuber 1997) would be more appropriate to capture the non-Markovian aspect of our process, we find that our architecture is adequate in that regard as well. Our architecture therefore has the advantage of being tractable (we can still appeal to the Proposition 1), all while being much simpler and easier to train.*

4. Hedging Performance and Hedging P&L under the Rough Bergomi Model

4.1. Deep Hedge under Rough Bergomi

Since the fRNN should perform just as well in Markovian case as the original one does, we first convinced ourselves that our architecture produces comparable results in the classical case. Quadratic losses as well as the training time for the Heston model were very similar for both.[4,5] We were now ready to test it on the rough Bergomi model. We hedge the ATM call from Section 3.3; the parameters were again $\xi_0 = 0.235 \times 0.235$, $\nu = 1.9$ and $\rho = -0.7$, and we investigate the dependence of the hedging loss on the Hurst parameter. The results are shown in Table 2. Again, the loss seems to exponentially decrease with increasing Hurst parameter and reaches quadratic losses comparable to classic stochastic volatility models at $H \gtrsim 0.5$.

Table 2. Quadratic loss for different Hurst parameters. Run time on 75 epochs was approximately 2 h for each parameter.

	Hurst Parameter							
H	0.10	0.20	0.30	0.40	0.60	0.70	0.80	0.90
Quad. loss	0.834 (*0.628)	0.376	0.263	0.244	0.204	0.206	0.197	0.191

*—loss on 200 epochs.

Comparing these results with both the model hedge and the deep hedge from Section 3.3 (see Table 3), we notice the fRNN does indeed perform *notably* better. With the number of epochs in the training phase increased from 75 to 200, the loss in the case of the deep hedge with original architecture does not improve, while the improvement with the proposed architecture is clearly visible. This indicates that while the semi-recurrent NN saturates at a given error, the new architecture keeps converging and improving. Since the training at 200 epochs was computationally costly (in terms of both memory and time), and since we have reached the model hedge's numbers at the higher end of H range, we did not keep increasing the number of epochs. However, we expect that to keep improving as the number of epochs increases, which definitely indicates the second approaches suitability.

In Figure 4, it is particularly interesting that the P&L distribution becomes increasingly left tailed with lower Hurst parameters. Even under the new architecture, the distribution for $H = 0.10$ is left-skewed with an extremely heavy left tail, where relative losses reached cca. -1000% in one of 10^5 sample paths. What is even more compelling is that the sizeable losses occurred, when the discretised stock process *jumped* by several thousand basis points during the hedging period. An example of such a path is shown in Figure 5. Although jumps are not featured in the rough Bergomi model (the price process is a continuous martingale (Gassiat 2019)), the model clearly exhibits jump-like behaviour when *discretised*.

Table 3. Comparison of the quadratic loss between model and deep hedges with *fRNN architecture* trained on 75 epochs for different H.

	Quadratic Hedging Loss		
H	**Model Hedge**	**Deep Hedge**	**Deep Hedge—fRNN**
0.10	1.45	1.16 (*1.12)	0.83 (*0.63)
0.20	0.52	0.67	0.38
0.30	0.34	0.46	0.26
0.40	0.24	0.36	0.22

*—on 200 epochs.

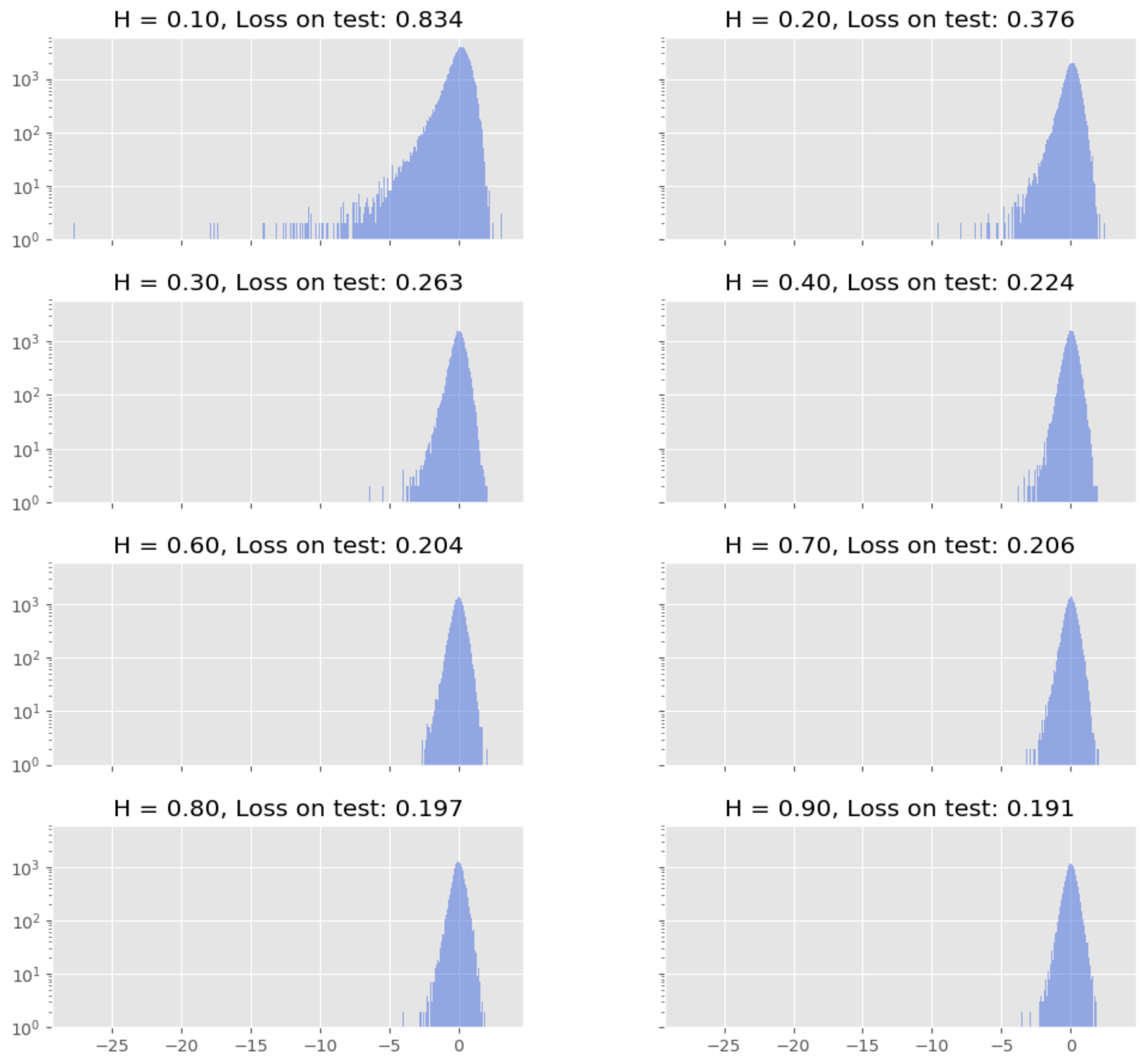


Figure 4. Empirical P&L distributions in log-scale for different Hurst parameters under the fRNN hedge. *Loss on test* denotes the realised quadratic loss on the test set for a network trained on 75 epochs.

Naturally, for $H = 0.10$, where this effect was the most noticeable, we tried increasing the training, test and validation set sizes, as well as number of epochs to 200. By doing this, we managed to decrease the realised loss to 0.628. The performance was notably better compared to 0.834 on smaller set sizes, but still far from the loss of 0.162 we obtained under the Heston model.

As can be seen in Figure 6, model hedge loss distribution exhibits very similar behaviour as the deep hedge distribution. Higher losses of the model hedge can be explained by the slightly fatter tail in comparison to the fully recurrent hedge. We remark that this behaviour is somewhat understandable, since re-hedging is done daily and the hedging frequency is far from being a valid approximation for a continuous hedge. In the next section, we thus implement hedges at different frequencies to see whether the Hölder regularity of the underlying process is problematic only for the deep hedging procedure or is the heavy left-tailed P&L distribution, a general phenomenon, when hedging under a discretised rough model.

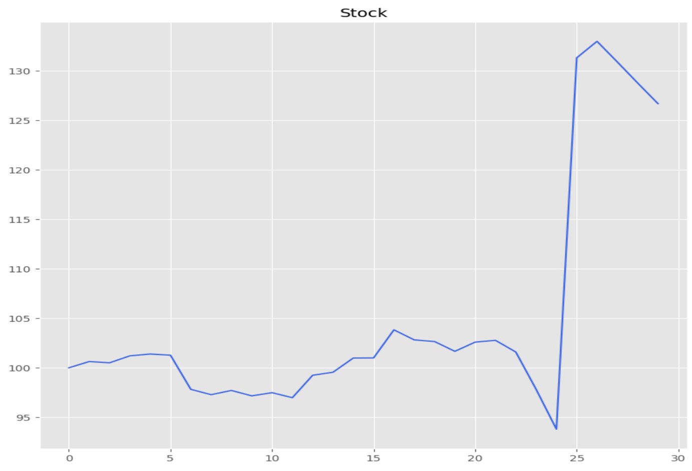
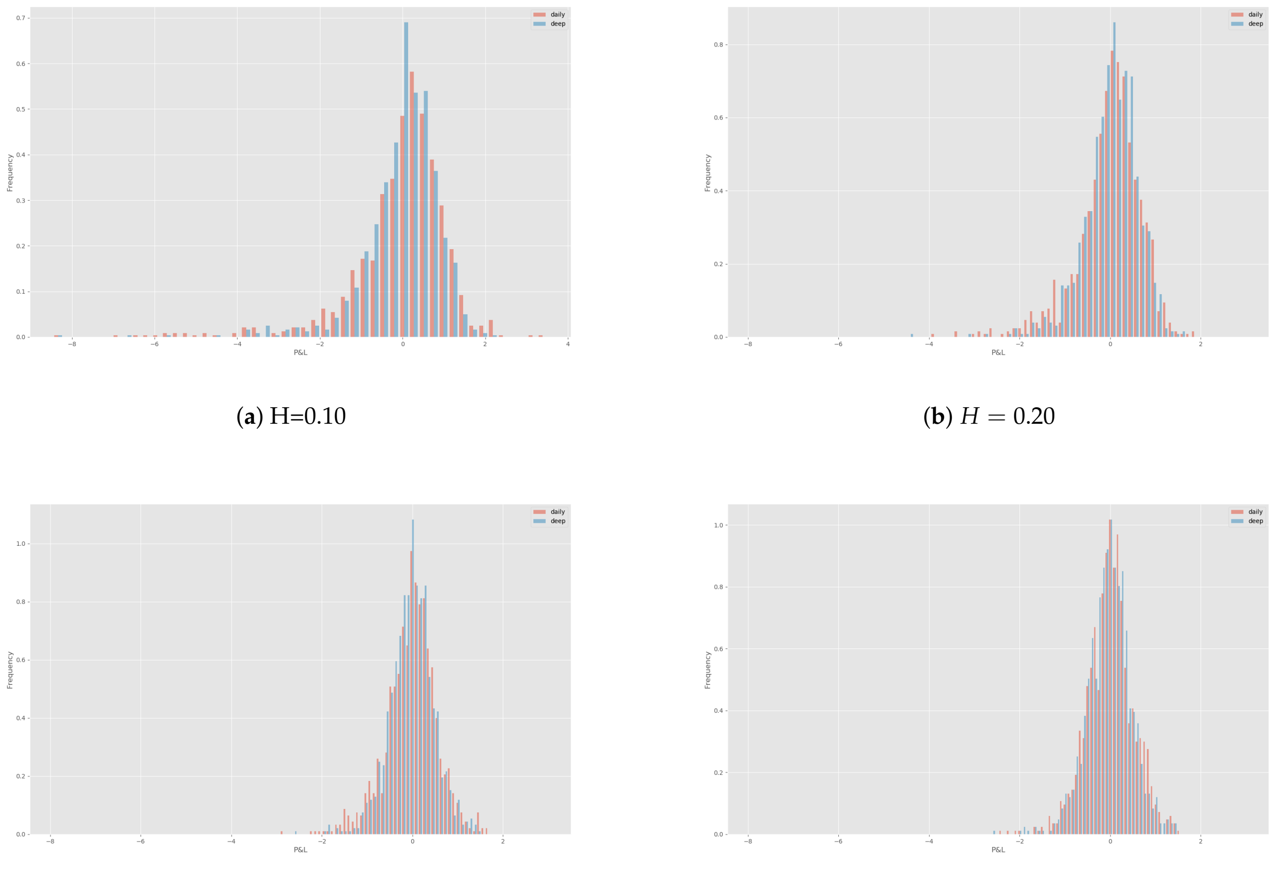


Figure 5. Under the discretised rough Bergomi model, the stock can jump by more than ±30% in a single time step. This stock path caused extreme loss of −27.73 seen in Figure 4.



(**a**) H=0.10 (**b**) $H = 0.20$

(**c**) $H = 0.30$ (**d**) $H = 0.40$

Figure 6. P&L distributions of rBergomi model hedge (red) vs. deep hedge with proposed architecture (blue) for different Hurst parameters realised on 10^3 sample paths.

4.2. Rehedges

We implement deep hedges on rBergomi paths with the Hurst parameter $H = 0.10$, where we re-hedged from every two days, up to four times a day. Again, one can see in Figure 7 that the distribution became slightly less leptocurtic, with more frequent rebalancing. The quadratic losses also decreased with higher frequency (see Table 4). However, this seems to happen at a slower rate than expected. This would essentially mean that as soon as transaction costs are present, small gains from more frequent rebalancing would be completely outweighed by higher transaction fees. As a matter of fact, for the four-time daily re-hedge the loss slightly increased, which indicates the model once again saturates, this time with respect to the hedging frequency. This is quite surprising considering higher hedging frequency usually translates to better performance in a continuous models. This is because the approximation is getting closer and closer to the continuous setting.

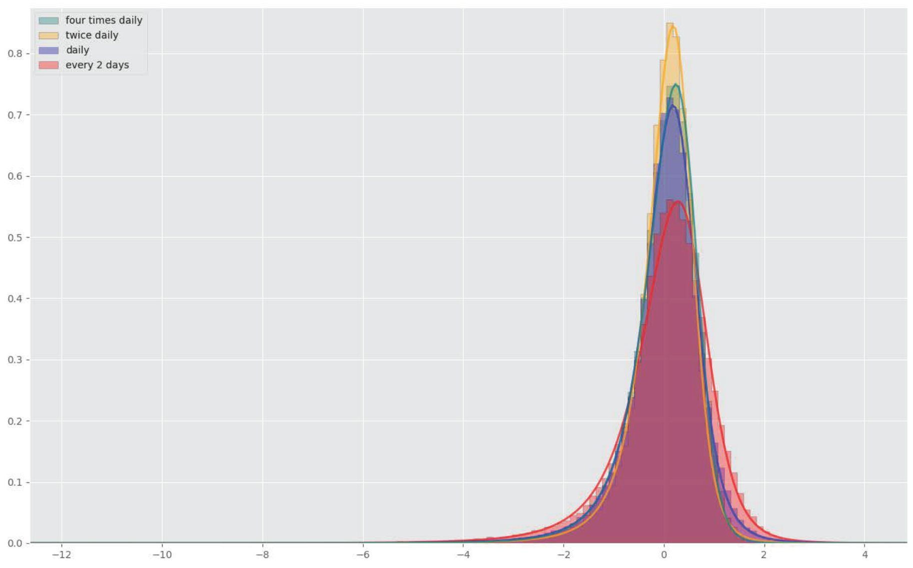


Figure 7. fRNN deep hedge for different hedging frequencies (with $H = 0.10$). The plot was rescaled, because of the massive outliers in the two day rehedge case. Non-central t distribution was fit for better visiblity.

Table 4. Comparison of the fRNN deep hedge quadratic losses for different hedging frequencies (with $H = 0.10$).

	Rehedging Frequency			
$\boldsymbol{H = 0.10}$	**Every Two Days**	**Daily**	**Twice Daily**	**Four Times Daily**
Quadratic loss	1.11	0.65	0.46	0.52
Training time (h)	3.1	7.5	19.6	45.3

Behaviour of distributions as well as hedging losses is in fact quite reminiscent of the behaviour of jump diffusion models analysed by Sepp (2012), which we recall in the following section.

4.3. Relation to the Literature

It is rather interesting that Sepp (2012) observes a similar behaviour when delta hedging under jump diffusion models. Similarly to our observations above, he finds (in the presence of jumps) that after a certain point the volatility of the P&L cannot be reduced by increasing the hedging frequency. More precisely, he shows that for jump diffusion models, there is a lower bound on the volatility of the P&L in relation to the hedging frequency. Not only that, the P&L distributions in Figure 8 for delta hedges under jump diffusion models are generally fairly similar to ours.

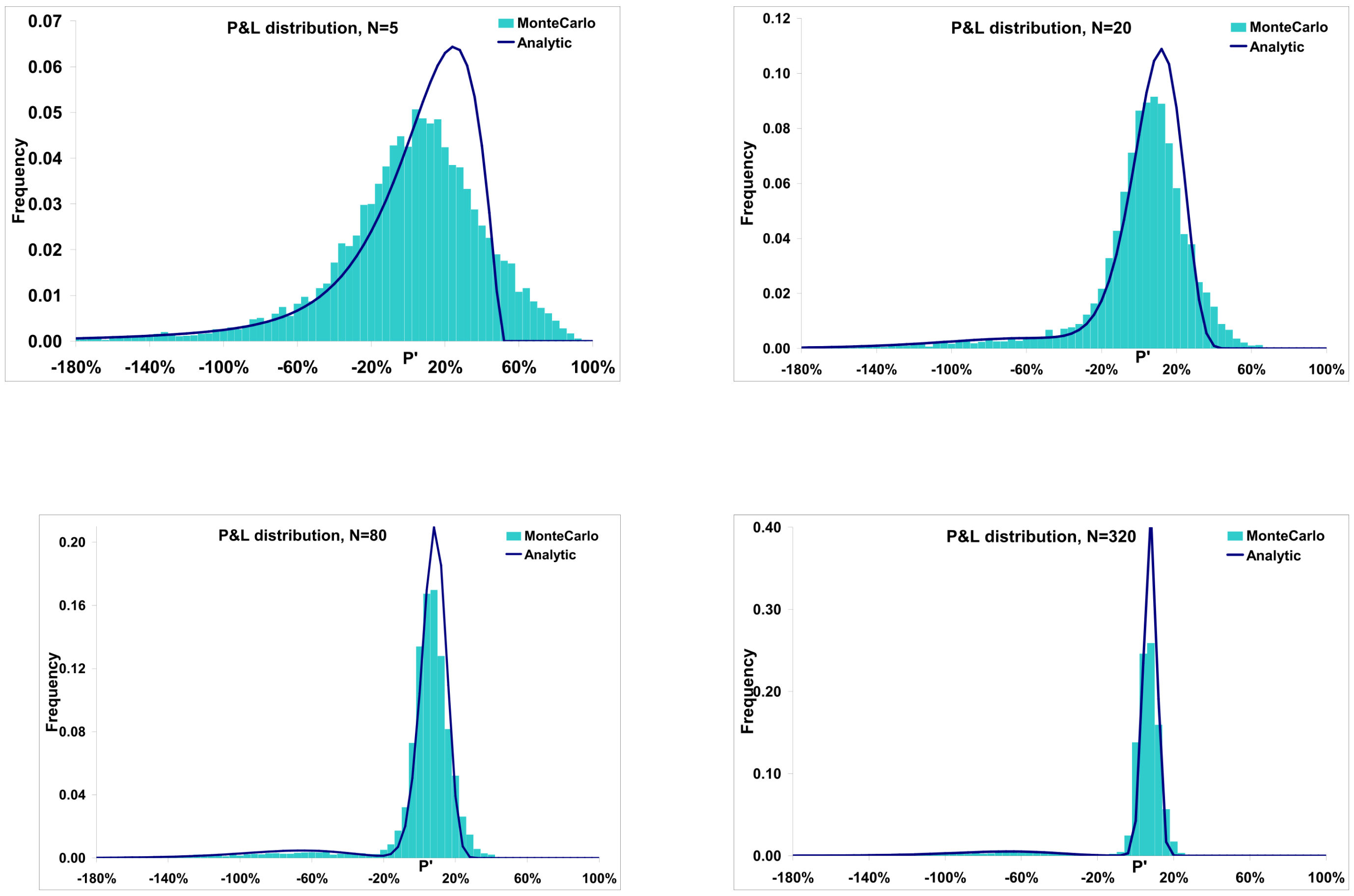


Figure 8. Profit and loss distributions for delta hedging under jump diffusion models (JDM) from (Sepp 2012).

This gives us the idea to treat the discretised rough models as jump models. In this case, the market is incomplete, and it is not possible to perfectly hedge a contingent claim with a portfolio containing a finite number of instruments (He et al. 2007). In practice, traders try to come as close as possible to the perfect hedge by trading a number of different options.

Unfortunately, when trying to implement the hedge approximation, we are quickly faced with the absence of analytical pricing formulas and limitations of the slow Monte-Carlo scheme. In order for us to train the deep hedge, we would have to calculate option prices on every time step of each sample path. In a typical application, we would need around 10 options with different strikes and at least 10^5 sample paths.

5. Conclusions

In this work, we presented and compared different methods for hedging under rough volatility models. More specifically, we analysed and implemented the perfect hedge for the rBergomi model from (Viens and Zhang 2019) and used the deep hedging scheme from (Buehler et al. 2019), which had to be adapted to a non-Markovian framework.

We were particularly interested in the dependence of the P&L on the Hurst parameter. We conclude that the deep hedge with the proposed architecture performs better than the discretised perfect hedge for all H. We also find that the hedging P&L distributions for low H are highly left-skewed and have a lot of mass in the left tail under the model hedge as well as the deep hedge.

To mitigate the heavy losses in cases when H is close to zero, we explored increasing the hedging frequency up to four times a day. The loss did improve and the P&L distribution became less leptocurtic, however only slightly. Intriguingly, slow response to increased hedging frequency and left-skewed P&L distribution are characteristic of delta hedges under jump diffusion models (Sepp 2012). We therefore observe that in terms of hedging, there is a relation between jump diffusion models and rough models. In accordance with the literature, we find that the price process, despite being a continuous martingale, exhibits jump-like behaviour (McCrickerd and Pakkanen 2018). We believe this is an excellent illustration of rough volatility models dynamics. Explosive, almost jump-like patterns in the stock price might be the reason why they can fit the short end of implied volatility so well.

In our view, it is crucial to take into account the jump aspect, when looking for an optimal hedge in discretised rough volatility models. Our suggestion for future research is adapting the objective function in the deep hedge scheme for jump risk optimisation. The first step would be optimisation of the expected shortfall risk measure. Next, more appropriate jump risk measures for discretised rough models can be developed. These risk measures cannot be completely analogous to the risk measures in (Sepp 2012), since rough models themselves do not feature jumps.

Author Contributions: All authors contributed to the conception of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: We gratefully acknowledge support by ETH Foundation and SNF project 179114.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Path Derivatives

Denote by $\mathcal{D}_0$ a càdlàg space and by $\mathcal{D}_t$ and $\mathcal{C}_t$ the space of càdlàg functions on $[t,T]$ and the space of continuous functions on $[t,T]$, respectively. Additionally, we denote by ω the sample paths on $[0,T]$, with ω_t as its value at time t and define

$$\Lambda := [0,T]\times\mathcal{C}([0,T],\mathbb{R}^d), \qquad \bar{\Lambda} := \left\{(t,\omega)\in[0,T]\times\mathcal{D}_0 : \omega|_{[t,T]}\in\mathcal{C}\right\};$$
$$\|\omega\|_T := \sup_{t\in[0,T]}|\omega_t|, \qquad \mathbf{d}\big((t,\omega),(t',\omega')\big) := |t-t'| + \|\omega-\omega'\|_T.$$

Furthermore, we denote the set of all $\mathbf{d}$-continuous functions $u:\bar{\Lambda}\to\mathbb{R}$ by $\mathcal{C}(\bar{\Lambda})$. Define the usual horizontal time derivative for $u\in\mathcal{C}(\bar{\Lambda})$ as in (Dupire 2019):

$$\partial_t u(t,\omega) := \lim_{\delta\downarrow 0}\frac{u(t+\delta,\omega)-u(t,\omega)}{\delta} \qquad \text{for all } (t,\omega)\in\bar{\Lambda}, \tag{A1}$$

requiring of course that the limit exists. For the spatial derivative with respect to ω, however, we use the definition of the Gateaux derivative for any $(t,\omega)\in\bar{\Lambda}$:

$$\langle\partial_\omega u(t,\omega),\eta\rangle = \lim_{\varepsilon\to 0}\frac{u(t,\omega+\varepsilon\eta\mathbb{1}_{[t,T]})-u(t,\omega)}{\varepsilon} \qquad \text{for any } \eta\in\mathcal{C}_t. \tag{A2}$$

Note that the function $u(t,\cdot)$ in the definition of the derivative is "lifted" only on $[t,T]$ and not on $[0,t)$. Hence, the convention we follow is actually

$$\langle\partial_\omega u(t,\omega),\eta\rangle := \left\langle\partial_\omega u(t,\omega),\eta\mathbb{1}_{[t,T]}\right\rangle \qquad \text{for any } s<t \text{ and } \eta\in\mathcal{C}_s.$$

The definition of Gateaux derivative is clearly also equal to

$$\langle\partial_\omega u(t,\omega),\eta\rangle = \frac{d}{d\varepsilon}u(t,\omega+\varepsilon\eta\mathbb{1}_{[t,T]})\Big|_{\epsilon=0}.$$

Remark A1. *We remark that our definition of the spatial derivative is different from the one in (Cont and Fournié 2013; Dupire 2019), where functional derivative quantifies the sensitivity of the functional to the variation solely in the end point of the path, i.e., ω_t. While in our definition, the perturbation takes place throughout the whole interval $[t, T]$.*

We define two more spaces necessary for our analysis:

$$\begin{aligned}\mathcal{C}^{1,2}(\bar{\Lambda}) :=& \Big\{u \in \mathcal{C}(\bar{\Lambda}) : \varphi \in \mathcal{C}(\bar{\Lambda}) \text{ for } \varphi \in \{\partial_t u, \partial_\omega u, \partial^2_{\omega\omega} u\}\Big\},\\ \mathcal{C}^{1,2}_{+}(\bar{\Lambda}) :=& \Big\{u \in \mathcal{C}(\bar{\Lambda}) : \varphi \text{ has polynomial growth for } \varphi \in \{\partial_t u, \partial_\omega u, \partial^2_{\omega\omega} u\} \text{ and}\\ & \Big\langle \partial^2_{\omega\omega} u, (\eta, \eta)\Big\rangle \text{ is locally uniformly continuous in } \omega \text{ with polynomial growth}\Big\}.\end{aligned}$$

Appendix B. Functional Itô Formula

We have to differentiate two cases. The regular case where $H \in (\frac{1}{2}, 1)$ and the singular case where the coefficients b, σ explode, because of the power-kernel in Riemann–Liouville fractional Brownian motion whenever the Hurst exponent H lies in $(0, \frac{1}{2})$. In the singular case, the coefficients $b, \sigma \notin \mathcal{C}_t$, and thus they cannot serve as the test function in the right side of (A2), since Gateaux derivative would not make sense any more. In order to develop an Itô formula for the singular case, definitions need to be slightly amended. Nonetheless, Viens et al. show that both cases yield a similar functional Itô formula.

Assumption A1.

i *The SDE (5) admits a weak solution (X, W).*

ii $\mathbb{E}\Big[\sup_{t\in[0,T]} |X_t|^p\Big] < \infty$ *for all* $p \geq 1$.

Assumption A2.

i *(Regular case) For any $r \in [0, T]$, $\partial_t b(t; r, \cdot), \partial_t \sigma(t; r, \cdot)$ exist for $t \in [r, T]$ and for $\varphi = b, \sigma, \partial_t b, \partial_t \sigma$,*

$$|\varphi(t; r, \omega)| \leq C_0(1 + \|\omega\|^{\kappa_0}_T) \qquad C_0, \kappa_0 > 0.$$

ii *(Singular case) For any $r \in [0, T]$, $\partial_t(t; r, \cdot)$ exists for $t \in (r, T]$ with $\varphi = b, \sigma$. There exists $H \in (0, \frac{1}{2})$ s.t., for some $C_0, \kappa_0 > 0$*

$$|\varphi(t; r, \omega)| \leq C_0(1 + \|\omega\|^{\kappa_0}_T)(t - r)^{H-\frac{1}{2}} \text{ and } |\varphi_t(t; r, \omega)| \leq C_0(1 + \|\omega\|^{\kappa_0}_T)(t - r)^{H-\frac{3}{2}}$$

Theorem A1 (Functional Itô formula)**.** *Let X be a weak solution to the SDE (5) for which $\mathbb{E}\Big[\sup_{t\in[0,T]} |X_t|^p\Big] < \infty$ for all $p \geq 1$ and Assumption A2 hold. Then*

$$\begin{aligned} du(t, X \otimes_t \Theta^t) = \partial_t u(t, X \otimes_t \Theta^t)dt + \frac{1}{2}\Big\langle \partial_{\omega\omega} u(t, X \otimes_t \Theta^t), (\sigma^{t,X}, \sigma^{t,X})\Big\rangle dt +&\\ \Big\langle \partial_\omega u(t, X \otimes_t \Theta^t), b^{t,X}\Big\rangle dt + \Big\langle \partial_\omega u(t, X \otimes_t \Theta^t), \sigma^{t,X}\Big\rangle dW_t, \quad \mathbb{P}\text{-a.s.}& \end{aligned} \tag{A3}$$

for $u \in C^{1,2}_{+}(\Lambda)$ in the regular case and $u \in C^{1,2}_{+,\beta}(\Lambda)$ with regularised Gateaux derivative for the singular case. For $\varphi = b, \sigma$ the notation $\varphi^{t,\omega}_s := \varphi(s; t, \omega)$ only emphasises the dependence on $s \in [t, T]$. For the definition of $C^{1,2}_{+,\beta}(\Lambda)$ and precise statement of the theorem in the singular case, see Theorem 3.17 and Theorem 3.10 in (Viens and Zhang 2019)

Appendix C. Discretisation of the Gateaux Derivative

It can be easily shown that $\hat{\Theta}^t_s = f(\Theta^t_s)$ for some $f : \mathbb{R} \to \mathbb{R}$. Therefore, we have a direct relation between the auxiliary process Θ and the forward variance $\hat{\Theta}$, which allows us to write the option price as the function of the entire *forward variance curve* $\hat{\Theta}^t_{[t,T]}$ at

time $t \in [0,T]$, namely $u(t, S_t, \Theta^t_{[t,T]}) = \tilde{u}(t, S_t, \hat{\Theta}^t_{[t,T]})$. This is important, when performing Monte-Carlo, since in the rough Bergomi model, the forward variance curve is directly modelled in the variance process with $\xi_t(\cdot) = \hat{\Theta}^t_{\cdot}$.
Let us suppose that we are able to trade at times $0 = t_0 < t_1 < \cdots < t_n = T$. In order to get the hedging weights at trading times t_i, we have to discretise the derivatives. The Gateaux derivative with respect to the stock simplifies to the usual derivative, and the discretisation is straightforward:

$$\partial_x \tilde{u}(t, S_t, \hat{\Theta}^t_{[t,T]}) \approx \frac{\tilde{u}(t, S_t + \varepsilon, \hat{\Theta}^t_{[t,T]}) - \tilde{u}(t, S_t, \hat{\Theta}^t_{[t,T]})}{\varepsilon} \qquad \text{for small } \varepsilon > 0. \tag{A4}$$

For the path-wise derivative, the discretisation is not immediately obvious, especially because of the dependence of the option price $\tilde{u}$ at time t on functional over the whole interval $[t,T]$, more precisely since $u : [0,T] \times [0,\infty) \times \mathcal{C}([0,T] \to \mathbb{R})$. First, we remind ourselves of the definition of the Gateaux derivative on a path ω:

$$\langle \partial_\omega u(t,\omega), \eta \rangle = \lim_{\varepsilon \to 0} \frac{u(t, \omega + \varepsilon \eta \mathbb{1}_{[t,T]}) - u(t,\omega)}{\varepsilon} \qquad \text{for any } \eta \in \Omega_t.$$

We proceed as in (Jacquier and Oumgari 2019) by approximating $\hat{\Theta}^t_{[t,T]}$ as a piecewise constant function

$$\hat{\Theta}^t_s \approx \sum_{i \in \mathcal{I}} \hat{\Theta}^t_i \mathbb{1}_{[t_i, t_{i+1})}(s) \qquad\qquad a^t_i = a^t_s \mathbb{1}_{[t_i, t_{i+1})}(s), \tag{A5}$$

where $\mathcal{I} := \{i \in \mathbb{N} : t \le t_i \le T\}$. We introduce the following approximations of the path derivatives along the direction a^t:

$$\begin{aligned}
\left\langle \partial_\omega \tilde{u}(t, S_t, \hat{\Theta}^t_{[t,T]}), a^t \right\rangle &\approx \partial_\varepsilon \tilde{u}\left(t, S_t, \sum_{i \in \mathcal{I}} (\hat{\Theta}^t_i + \varepsilon a^t) \mathbb{1}_{[t_i, t_{i+1})}(s) \Big|_{s \in [t,T]} \right) \Bigg|_{\varepsilon = 0} \\
&= \partial_\varepsilon \hat{u}\left(t, S_t, (\hat{\Theta}^t_i + \varepsilon a^t_i)_{i \in \mathcal{I}} \right) \Big|_{\varepsilon = 0} \\
&= \sum_{i \in \mathcal{I}} \partial_{\hat{\Theta}^t_i} \hat{u}(t, S_t, \boldsymbol{\theta}^t) a^t_i,
\end{aligned}$$

with $\boldsymbol{\theta}^t := (\hat{\Theta}^t_i)_{i \in \mathcal{I}}$ and $\hat{u}$ acts on $[0,T] \times [0,\infty) \times \mathbb{R}^{\#\mathcal{I}}$. Further discretising the derivative, we have for the flat forward variance $\xi(t) = \xi_0$:

$$\left\langle \partial_\omega \tilde{u}(t, S_t, \xi_0), a^t \right\rangle \approx \frac{\tilde{u}(t, S_t, \xi_0 + \varepsilon) - \tilde{u}(t, S_t, \xi_0)}{\varepsilon} a^t \qquad \text{for small } \varepsilon > 0.$$

The option prices $\tilde{u}$ can now be evaluated using Monte-Carlo at each time step to get the hedging weights. Note that the discretisation of the Gateaux derivative is purely heuristic and that a rigorous proof of the convergence to the true derivative is out of scope of this work. For more details we refer to (Jacquier and Oumgari 2019).

Notes

1 For the numerical implementation of the resulting strategies that we consider in the following sections, we naturally consider again the discrete filtration introduced above in Section 2, representing a discretisation of the continuous market.

2 Several estimation procedures of the Hurst parameter were used; see, e.g., (Di Matteo 2007; Di Matteo et al. 2005). Estimations of the paths simulated with the hybrid scheme (Bennedsen et al. 2017; McCrickerd and Pakkanen 2018) were on the other hand in alignment with the input parameter.

3 Note that by completely recurrent we do not mean the same network is used at each time step, but that the hidden state is passed on to the cell in the next time step along with current portfolio positions.

4 For Heston parameters $\alpha = 1, b = 0.04, \sigma = 0.8, V_0 = 0.04, S_0 = 100$ and $\rho = -0.7$ the quadratic losses were 0.20 under original architecture and 0.162 under the fully recurrent one. Both training times were fairly similar as well.

[5] All the experiments were performed on a `Dell-HQIQ2UV` laptop with `Intel i7-8550U CPU` using TENSORFLOW V1.3.

References

Alòs, Elisa, Jorge A. León, and Josep Vives. 2007. On the short-time behavior of the implied volatility for jump-diffusion models with stochastic volatility. *Finance and Stochastics* 11: 571–89. [CrossRef]

Bayer, Christian, Blanka Horvath, Aitor Muguruza, Benjamin Stemper, and Mehdi Tomas. 2019. On deep calibration of (rough) stochastic volatility models. *arXiv* arXiv:1908.08806.

Bayer, Christian, Peter Friz, and Jim Gatheral. 2015. Pricing under rough volatility. *Quantitative Finance* 16: 887–904. [CrossRef]

Bennedsen, Mikkel, Asger Lunde, and Mikko S. Pakkanen. 2017. Hybrid scheme for brownian semistationary processes. *Finance and Stochastics* 21: 931–65. [CrossRef]

Benth, Fred Espen, Nils Detering, and Silvia Lavagnini. 2020. Accuracy of deep learning in calibrating HJM forward curves. *arXiv* arXiv:2006.01911.

Bolko, Anine E., Kim Christensen, Mikko S. Pakkanen, and Bezirgen Veliyev. 2020. Roughness in spot variance? A GMM approach for estimation of fractional log-normal stochastic volatility models using realized measures. *arXiv* arXiv:2010.04610.

Buehler, Hans, Blanka Horvath, Terry Lyons, Imanol Perez Arribas, and Ben Wood. 2020a. A data-driven market simulator for small data environments. *SSRN Electronic Journal*. [CrossRef]

Buehler, Hans, Blanka Horvath, Terry Lyons, Imanol Perez Arribas, and Ben Wood. 2020b. Generating financial markets with signatures. *SSRN Electronic Journal*. [CrossRef]

Buehler, Hans, Lukas Gonon, Josef Teichmann, and Ben Wood. 2019. Deep hedging. *Quantitative Finance* 19: 1271–91. [CrossRef]

Carmona, Philippe, Gérard Montseny, and Laure Coutin. 1998. *Application of a Representation of Long Memory Gaussian Processes*. *Publications du Laboratoire de statistique et probabilités*. Toulouse: Laboratoire de Statistique et Probabilités.

Cont, Rama, and David-Antoine Fournié. 2013. Functional itô calculus and stochastic integral representation of martingales. *The Annals of Probability* 41: 109–33. [CrossRef]

Cuchiero, Christa, Martin Larsson, and Josef Teichmann. 2020. Deep neural networks, generic universal interpolation, and controlled ODEs. *SIAM Journal on Mathematics of Data Science* 2: 901–19. [CrossRef]

Cuchiero, Christa, Wahid Khosrawi, and Josef Teichmann. 2020. A generative adversarial network approach to calibration of local stochastic volatility models. *Risks* 8: 101. [CrossRef]

Di Matteo, Tiziana. 2007. Multi-scaling in finance. *Quantitative Finance* 7: 21–36. [CrossRef]

Di Matteo, Tiziana, Tomaso Aste, and Michel M. Dacorogna. 2005. Long-term memories of developed and emerging markets: Using the scaling analysis to characterize their stage of development. *Journal of Banking & Finance* 29: 827–51.

Dupire, Bruno. 2019. Functional itô calculus. *Quantitative Finance* 19: 721–29. [CrossRef]

Fukasawa, Masaaki. 2010. Asymptotic analysis for stochastic volatility: Martingale expansion. *Finance and Stochastics* 15: 635–54. [CrossRef]

Föllmer, Hans, and Alexander Schied. 2016. *Stochastic Finance: An Introduction in Discrete Time*. New York: De Gruyter.

Gassiat, Paul. 2019. On the martingale property in the rough bergomi model. *Electronic Communications in Probability* 24. [CrossRef]

Gatheral, Jim, Thibault Jaisson, and Mathieu Rosenbaum. 2018. Volatility is rough. *Quantitative Finance* 18: 933–49. [CrossRef]

Gierjatowicz, Patryk, Marc Sabate-Vidales, David Šiška, Lukasz Szpruch, and Žan Žurič. 2020. Robust pricing and hedging via neural SDEs. *SSRN Electronic Journal*. [CrossRef]

He, Changhong, J. Shannon Kennedy, Thomas F. Coleman, Peter A. Forsyth, Yuying Li, and Kenneth R. Vetzal. 2007. Calibration and hedging under jump diffusion. *Review of Derivatives Research* 9: 1–35. [CrossRef]

Henry-Labordere, Pierre. 2019. Generative models for financial data. *SSRN Electronic Journal*. [CrossRef]

Hernandez, Andres. 2016. Model calibration with neural networks. *Risk*. [CrossRef]

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9: 1735–80. [CrossRef] [PubMed]

Horvath, Blanka, Aitor Muguruza, and Mehdi Tomas. 2021. Deep learning volatility: A deep neural network perspective on pricing and calibration in (rough) volatility models. *Quantitative Finance* 21:1, 11–27. [CrossRef]

Ilhan, Aytaç, Mattias Jonsson, and Ronnie Sircar. 2009. Optimal static-dynamic hedges for exotic options under convex risk measures. *Stochastic Processes and their Applications* 119: 3608–32. [CrossRef]

Jacquier, Antoine, and Mugad Oumgari. 2019. Deep curve-dependent pdes for affine rough volatility. *arXiv* arXiv:1906.02551.

Kac, Mark. 1949. On distributions of certain wiener functionals. *Transactions of the American Mathematical Society* 65: 1–13. [CrossRef]

Kondratyev, Alexei, and Christian Schwarz. 2019. The market generator. *Econometrics: Econometric & Statistical Methods - Special Topics eJournal*. [CrossRef]

Liu, Shuaiqiang, Anastasia Borovykh, Lech A. Grzelak, and Cornelis W. Oosterlee. 2019. A neural network-based framework for financial model calibration. *Journal of Mathematics in Industry* 9: 9. [CrossRef]

Livieri, Giulia, Saad Mouti, Andrea Pallavicini, and Mathieu Rosenbaum. 2018. Rough volatility: Evidence from option prices. *IISE Transactions* 50: 767–76. [CrossRef]

McCrickerd, Ryan, and Mikko S. Pakkanen. 2018. Turbocharging monte carlo pricing for the rough bergomi model. *Quantitative Finance* 18: 1877–86. [CrossRef]

Pascanu, Razvan, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. How to construct deep recurrent neural networks. Paper presented at Second International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14–16.

Ruf, Johannes, and Weiguan Wang. 2020. Neural networks for option pricing and hedging: A literature review. *The Journal of Computational Finance* 24. [CrossRef]

Schweizer, Martin. 1995. Variance-optimal hedging in discrete time. *Mathematics of Operations Research* 20: 1–32. [CrossRef]

Sepp, Artur. 2012. An approximate distribution of delta-hedging errors in a jump-diffusion model with discrete trading and transaction costs. *Quantitative Finance* 12: 1119–41. [CrossRef]

Viens, Frederi, and Jianfeng Zhang. 2019. A martingale approach for fractional brownian motions and related path dependent PDEs. *The Annals of Applied Probability* 29: 3489–540. [CrossRef]

Wiese, Magnus, Lianjun Bai, Ben Wood, and Hans Buehler. 2019. Deep hedging: Learning to simulate equity option markets. *arXiv* arXiv:1911.01700.

Wiese, Magnus, Robert Knobloch, Ralf Korn, and Peter Kretschmer. 2020. Quant GANs: Deep generation of financial time series. *Quantitative Finance* 20: 1419–40. [CrossRef]

Xu, Mingxin. 2005. Risk measure pricing and hedging in incomplete markets. *Annals of Finance* 2: 51–71. [CrossRef]

Xu, Tianlin, Li K. Wenliang, Michael Munn, and Beatrice Acciaio. 2020. Cot-gan: Generating sequential data via causal optimal transport. *arXiv* arXiv:2006.08571.