DEEP KERNEL HEDGING

Jean-Loup Dupret, Donatien Hainaut,
Edouard Motte

LIDAM Discussion Paper ISBA
2026 / 37



UCLouvain

ISBA | LIDAM
Louvain Institute of Data Analysis and Modeling in economics and statistics

ISBA

Voie du Roman Pays 20 - L1.04.01

B-1348 Louvain-la-Neuve

Email : lidam-library@uclouvain.be

https://uclouvain.be/en/research-institutes/lidam/isba/publication

Deep kernel hedging

Jean-Loup Dupret[*1,2], Donatien Hainaut[†3], and Edouard Motte[‡3]

[1]University of Amsterdam, Amsterdam School of Economics
[2]ETH Zurich, Department of Mathematics, RiskLab
[3]Université Catholique de Louvain, LIDAM–ISBA

October 14, 2026

Abstract

We introduce a deep kernel hedging framework that combines the flexibility of deep learning with the structural inductive bias of kernel methods. The hedging functional is restricted to a reproducing kernel Hilbert space whose kernel is parameterized through a neural network embedding of the input features. The framework minimizes a regularized empirical risk under convex loss functions and can accommodate path-dependent information through truncated time-augmented signature features. We derive a generalized representer theorem for the joint hedging problem, reducing the empirical optimization to a finite-dimensional problem. To further reduce the computational cost associated with large kernel matrices, we develop a scalable random Fourier feature approximation and establish convergence guarantees. The random Fourier parameters are sampled once and remain fixed throughout training, while the deep kernel adapts to market data through the learned neural representation. We evaluate the performance of the proposed deep kernel approach on both synthetic and real data and compare it with standard kernel methods and classical deep hedging architectures. Numerical results indicate competitive and robust hedging performance, particularly in low-data regimes, which highlights the benefits of combining expressive neural representations with the inductive bias of kernel methods.

KEYWORDS: Hedging, Kernel methods, Deep kernel learning, Random Fourier features, Path-signatures.

1 Introduction

The problem of hedging financial derivatives is a crucial topic in mathematical finance. Given a financial claim exposed to uncertain market fluctuations, the objective is to construct trading strategies that optimally reduce the associated financial risk. In idealized complete market settings, claims can be perfectly replicated by an appropriate trading strategy. However, realistic financial markets are typically incomplete due to the presence of stochastic volatility, liquidity effects or other sources of uncertainty. In such situations, perfect replication is generally impossible and hedging problems are naturally formulated as stochastic control problems aiming at minimizing a suitable loss functional associated with the profit and loss (P&L) of the strategy.

The resulting optimization problems strongly depend on the modeling assumptions for the underlying risk factors and market dynamics. In particular, the optimal hedging strategy may vary significantly depending on whether the dynamics are Markovian or path-dependent, low-dimensional or high-dimensional. Except in a few specific settings with restrictive assumptions (Pham, 2009), explicit characterizations of optimal hedging strategies through stochastic control are rarely available. One well-known example where such a characterization is generally unavailable concerns the hedging under the real-world measure in stochastic volatility models (Jonsson and Sircar, 2002; Motte and Hainaut, 2024). This has motivated the development of efficient numerical and data-driven approaches for solving such hedging problems.

[*] *j.l.dupret@uva.nl*
[†] *donatien.hainaut@uclouvain.be*
[‡] *Corresponding author, edouard.motte@uclouvain.be.*

Over the last few years, the rapid development of machine learning techniques has deeply impacted quantitative finance, particularly in the context of hedging problems. In this direction, the deep hedging framework introduced in Buehler et al. (2019) proposes to parametrize hedging strategies using deep neural networks and to learn optimal trading policies directly from simulated or historical data without any modeling assumption on the chosen market dynamics. Since then, several extensions have been proposed, including robust and empirical formulations of deep hedging problems (Carbonneau and Godin, 2021; Lütkebohmert et al., 2022; Mikkilä and Kanniainen, 2023; Abi Jaber and Gérard, 2025; Gnoatto et al., 2025; Dupret and Hainaut, 2026). Deep learning approaches offer remarkable flexibility and expressive power, especially in complex market environments where classical parametric models become inadequate. Nevertheless, neural network-based methods may suffer from instability, overfitting and substantial data requirements. This has motivated the exploration of alternative function approximation frameworks that combine flexibility with stronger structural properties in the presence of scarce data, in particular kernel-based hedging techniques.

Kernel methods have emerged as a powerful alternative in statistical learning theory. Based on the theory of reproducing kernel Hilbert spaces (RKHS), kernel methods provide flexible nonparametric models while preserving a rich functional analytic structure (Cristianini and Shawe-Taylor, 2000; Herbrich, 2001; Schölkopf and Smola, 2002; Shawe-Taylor and Cristianini, 2004; Williams and Rasmussen, 2006; Kurková and Sanguineti, 2008). Their mathematical tractability, convex optimization structure, and strong generalization properties have led to numerous successful applications across machine learning and statistics. In finance, kernel-based approaches have recently attracted increasing attention for pricing, hedging, portfolio optimization and risk-management problems (Chalup and Mitschele, 2008; Hu and Zastawniak, 2020; Hainaut and Dupret, 2026; Hainaut, 2026). In particular, several recent works have investigated kernel-based hedging strategies (Nian et al., 2018; Ludkovski and Saporito, 2021; Cirone and Salvi, 2025; Futter et al., 2025), highlighting the ability of RKHS methods to efficiently capture nonlinear structures while having robustness and regularization properties.

Despite their appealing theoretical properties, classical kernel methods typically rely on fixed kernels whose representational capacity may be insufficient in highly complex financial environments. Deep kernel learning addresses this limitation by composing a standard kernel with a parametrized neural representation learned from data (Wilson et al., 2016a,b; Jiu and Sahbi, 2017; Ober et al., 2021). Such approaches aim at combining the expressive power of deep architectures with the structured inductive bias and regularization properties of kernel methods. However, the deep kernel learning approaches considered in Wilson et al. (2016a,b); Jiu and Sahbi (2017); Ober et al. (2021) are typically formulated within a Gaussian-process (GP) framework, where the representation parameters and kernel hyperparameters are jointly learned by maximizing the GP marginal likelihood. Although statistically principled, this procedure can be computationally demanding because it involves repeatedly manipulating large kernel matrices, while the resulting nonconvex optimization may be sensitive to initialization and hyperparameter choices.

To retain the flexibility of deep kernel learning while improving its scalability, we introduce a novel deep kernel framework for empirical hedging together with an explicit random-feature approximation that enables efficient stochastic gradient optimization. We consider more precisely hedging functionals belonging to an RKHS associated with a parameterized deep kernel of the form

$$K_\theta(x, x') = K\big(\psi_\theta(x), \psi_\theta(x')\big), \qquad x, x' \in \mathcal{X} \subseteq \mathbb{R}^d,$$

where $\psi_\theta : \mathcal{X} \to \mathbb{R}^p$ is a learned neural representation and $K : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ is a standard base kernel. The inputs $x \in \mathcal{X}$ may consist of standard market-state variables or incorporate path-dependent information through truncated time-augmented signatures, thereby capturing richer temporal features of the underlying asset trajectories. We then formulate the corresponding empirical kernel hedging problem as an infinite-dimensional learning problem. Using a generalized representer theorem, we derive an equivalent finite-dimensional formulation that provides an explicit characterization of the optimal hedging strategy and establishes the well-posedness of the joint learning procedure. While this characterization is theoretically appealing, it still requires constructing and repeatedly updating large kernel matrices, which becomes computationally prohibitive as the number of training paths increases. To address this limitation, we introduce a scalable approximation scheme based on random Fourier features (RFF), as introduced in Rahimi and Recht (2007) and further analyzed in Rahimi and Recht (2008); Sutherland and Schneider (2015). The result-

ing finite-dimensional RFF formulation substantially reduces the computational and memory requirements by enabling joint mini-batch training of the neural representation ψ_θ and the RFF output (linear) coefficients.

Our construction is therefore naturally related to random-feature neural networks (RFNNs, also called *extreme learning machine*, see Huang et al. (2006); Sun (2019); Gonon et al. (2023)), in which the hidden neural weights and biases are randomly sampled and kept fixed, while only the final output linear layer is trained; see also the online RFNN extension of Koppel et al. (2019) and Le Moal (2025). These RFNN methods allow for general activation functions and sampling distributions but do not learn an input representation ψ_θ before the random-feature layer. In our framework, the random Fourier parameters are also sampled once before training and subsequently held fixed, but the resulting random features and kernel evolve throughout training through our learned latent representation ψ_θ, which is optimized jointly with the output RFF coefficients. Other related deep kernel constructions based on RFFs have also been proposed in Mehrkanoon and Suykens (2018); Xie et al. (2019); Mehrkanoon (2019); Fang et al. (2023). The approach of Xie et al. (2019) stacks several RFF layers whose initially random Fourier parameters are subsequently trained by backpropagation, while Mehrkanoon and Suykens (2018) and Mehrkanoon (2019) develop similar hybrid neural-kernel architectures with stacked RFF layers, deeper kernel blocks and pooling constructions for Least Squares Support Vector Machines (LS-SVM). Other methods learn instead kernels directly from data through their random-feature representation (Sinha and Duchi, 2016; Zhen et al., 2020); in particular, Fang et al. (2023) uses a neural generator to learn the spectral distribution from which the Fourier frequencies are sampled. In contrast, we sample the base Fourier frequencies and phases once and keep them fixed during training, thereby learning the deep kernel only through the representation map ψ_θ. The RFF layer is thus used solely as a scalable finite-dimensional approximation of the deep kernel K_θ arising from our empirical hedging formulation.

Our main contribution is therefore the development and analysis of a deep kernel framework specifically built for empirical hedging. More precisely, we:

- introduce a deep kernel hedging architecture that combines a learned nonlinear representation of market information together with RKHS regularization, making it particularly well suited to complex financial environments with limited training data;
- formulate the hedging problem over the RKHS induced by the learned deep kernel, derive an equivalent finite-dimensional representation through a generalized representer theorem, and establish the existence of an optimal deep kernel hedge;
- develop a scalable random Fourier feature formulation that enables joint mini-batch training of the neural representation and output coefficients, along with convergence guarantees;
- provide an extensive numerical evaluation across different derivative products, synthetic and real datasets, and different hedging objectives, including quadratic loss and CVaR minimization. The experiments demonstrate competitive and robust hedging performance, particularly in low-data regimes where purely neural approaches suffer from instability or overfitting.

The remainder of the paper is organized as follows. In Section 2, we recall the main elements of RKHS theory and deep kernel learning. Section 3 presents the proposed deep kernel hedging formulation, derives the finite-dimensional representation of the optimization problem via a generalized representer theorem, and introduces a random Fourier feature approximation for efficient learning with convergence guarantees. Finally, Section 4 contains the numerical experiments on both synthetic and real datasets, including comparisons with benchmark methods (standard kernel methods, deep hedging and RFNN architectures) and the use of signature-based features.

2 Deep kernel learning

In this section, we introduce the kernel learning framework used in the paper. Let $\mathcal{X} \subseteq \mathbb{R}^d$ be a nonempty set. A function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a *kernel* on $\mathcal{X}$ if there exists a Hilbert space $\mathcal{H}_K$ and a map $\Phi : \mathcal{X} \to \mathcal{H}_K$

such that for all $x, x' \in \mathcal{X}$, we have $K(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}_K}$. Equivalently, a function K is a kernel if and only if it is symmetric and positive-semidefinite, that is, for every $m \in \mathbb{N}$, $x_1, \ldots, x_m \in \mathcal{X}$, $c_1, \ldots, c_m \in \mathbb{R}$,

$$\sum_{i=1}^{m} \sum_{j=1}^{m} c_i c_j K(x_i, x_j) \geq 0 \,.$$

Definition 2.1 (Reproducing Kernel Hilbert Space)**.** A Hilbert space $\mathcal{H}_K$ of real-valued functions on $\mathcal{X}$ equipped with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_K}$ is a *reproducing kernel Hilbert space* (RKHS) associated with the kernel K if for every $x \in \mathcal{X}$ and $f \in \mathcal{H}_K$, the point evaluation can be represented as

$$f(x) = \langle f, K(\cdot, x) \rangle_{\mathcal{H}_K}.$$

This property ensures that evaluating a function at a point is a continuous linear functional on $\mathcal{H}_K$. The Moore–Aronszajn theorem then states that every kernel K on $\mathcal{X}$ determines a unique RKHS having K as its reproducing kernel, and this space satisfies $\mathcal{H}_K = \overline{\text{span}\{K(\cdot, x) : x \in \mathcal{X}\}}$. The reproducing property provides in the next sections the key link between the (potentially) infinite-dimensional space $\mathcal{H}_K$ and the finite-dimensional representations arising in kernel-based learning problems.

2.1 Empirical kernel regression

We first recall the classical empirical kernel regression framework. Let $X \in \mathcal{X}$ denote a vector of covariates and $Y \in \mathbb{R}$ a target variable that we aim to approximate by a function $f : \mathcal{X} \to \mathbb{R}$. Kernel methods restrict f to the RKHS $\mathcal{H}_K$ associated with a kernel K on $\mathcal{X}$. In this configuration, given a sample $\{(X_i, Y_i)\}_{i=1}^N$, a continuous convex bounded from below loss function $\mathcal{L} : \mathbb{R} \to \mathbb{R}$ and a regularization parameter $\lambda > 0$, the goal of empirical kernel regression is to solve

$$\inf_{f \in \mathcal{H}_K} \left\{ \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left(Y_i - f(X_i)\right) + \lambda \|f\|_{\mathcal{H}_K}^2 \right\}, \tag{2.1}$$

where $\|.\|_{\mathcal{H}_K}$ denotes the norm induced by the inner product of the Hilbert space $\mathcal{H}_K$, i.e. $\|.\|_{\mathcal{H}_K} = \sqrt{\langle ., . \rangle_{\mathcal{H}_K}}$. The term $\lambda \|f\|_{\mathcal{H}_K}^2$ controls the complexity of the estimator and makes the objective coercive and strictly convex. In particular, for every $\lambda > 0$, problem (2.1) has a unique solution in $\mathcal{H}_K$, see also Lemma 3.1. By the representer theorem (Schölkopf et al. (2001) and Theorem 3.1 below), the solution has the form

$$f_N^\star(\cdot) = \sum_{i=1}^{N} \alpha_i^\star K(\cdot, X_i), \tag{2.2}$$

for some $\boldsymbol{\alpha}^\star = (\alpha_1^\star, \ldots, \alpha_N^\star)^\top \in \mathbb{R}^N$. Define the response vector $\mathbf{Y} \in \mathbb{R}^N$ and Gram matrix $\mathbf{K} \in \mathbb{R}^{N \times N}$ by

$$\mathbf{Y} := (Y_1, \ldots, Y_N)^\top, \qquad \mathbf{K}_{ij} := K(X_i, X_j).$$

The empirical kernel regression can then be reformulated as

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \left\{ \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\Big(Y_i - \sum_{j=1}^{N} \alpha_j K(X_j, X_i)\Big) + \lambda \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} \right\}. \tag{2.3}$$

For a general loss function, the optimal coefficient vector $\boldsymbol{\alpha}^\star$ solving (2.3) does not generally admit a closed-form expression. However, for the squared loss $\mathcal{L}(u) = u^2$, the problem reduces to the quadratic optimization

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \left\{ \frac{1}{N} \|\mathbf{Y} - \mathbf{K}\boldsymbol{\alpha}\|_2^2 + \lambda \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} \right\}.$$

Since $\mathbf{K}$ is positive semidefinite and $\lambda > 0$, the matrix $\mathbf{K} + \lambda N I_N$ is positive definite. A convenient coefficient vector representing the unique minimizer in $\mathcal{H}_K$ for the square loss is then

$$\boldsymbol{\alpha}^\star = (\mathbf{K} + \lambda N I_N)^{-1} \mathbf{Y}, \quad \text{or equivalently} \quad (\mathbf{K} + \lambda N I_N) \boldsymbol{\alpha}^\star = \mathbf{Y}.$$

2.2 Deep kernel regression

While classical kernel ridge regression relies on a fixed kernel K, its performance is strongly influenced by the choice of kernel and its hyperparameters. In some cases, a fixed kernel may fail to capture the relevant structure of the data. To address this limitation, we consider a deep kernel learning approach, initially introduced by Wilson et al. (2016a). The main idea is to combine kernel methods with deep learning by introducing a parametric feature map

$$\psi_\theta : \mathcal{X} \to \mathbb{R}^p,$$

where ψ_θ is typically a neural network with parameters $\theta \in \Theta$. Instead of applying the kernel directly to input features, we define a parametrized function of the form

$$K_\theta(x, x') := K(\psi_\theta(x), \psi_\theta(x')), \quad x, x' \in \mathcal{X}, \tag{2.4}$$

where $K : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ is a standard kernel with feature map Φ. The function K_θ is thus itself a symmetric positive-semidefinite kernel with feature map $x \mapsto \Phi(\psi_\theta(x))$. Therefore, the neural network ψ_θ learns a representation of the input space, while the kernel measures similarity in the resulting latent space. In particular, when ψ_θ reduces to (or learns) the identity function, we retrieve the classical kernel framework. Thus, given the dataset $\{(X_i, Y_i)\}_{i=1}^N$, deep kernel empirical regression jointly selects the representation parameter θ and a regression function in $\mathcal{H}_{K_\theta}$ solving

$$\inf_{\theta\in\Theta} \inf_{f\in\mathcal{H}_{K_\theta}} \left\{ \frac{1}{N}\sum_{i=1}^N \mathcal{L}\big(Y_i - f(X_i)\big) + \lambda \|f\|^2_{\mathcal{H}_{K_\theta}} \right\}, \qquad \lambda > 0. \tag{2.5}$$

For every fixed θ, the representer theorem and equation (2.2) imply that the unique optimal regression function can be written as

$$f^\star_{N,\theta}(\,\cdot\,) = \sum_{i=1}^N \alpha^\star_{\theta,i}\, K_\theta(\,\cdot\,, X_i).$$

for some coefficients $\boldsymbol{\alpha}^\star_\theta = (\alpha^\star_{\theta,1}, \ldots, \alpha^\star_{\theta,N})^\top$. The infinite-dimensional optimization problem hence reduces to

$$\inf_{\theta\in\Theta} \min_{\boldsymbol{\alpha}\in\mathbb{R}^N} \left\{ \frac{1}{N}\sum_{i=1}^N \mathcal{L}\Big(Y_i - \sum_{j=1}^N \alpha_j\, K_\theta(X_i, X_j)\Big) + \lambda \boldsymbol{\alpha}^\top \mathbf{K}_\theta \boldsymbol{\alpha} \right\}, \tag{2.6}$$

where $(\mathbf{K}_\theta)_{ij} = K_\theta(X_i, X_j)$. For fixed $\theta \in \Theta$ and squared loss $\mathcal{L}(u) = u^2$, the optimal coefficients $\boldsymbol{\alpha}^\star_\theta$ are again given by

$$\boldsymbol{\alpha}^\star_\theta = (\mathbf{K}_\theta + \lambda N I_N)^{-1} \mathbf{Y}\,. \tag{2.7}$$

The inverse is well defined because $\mathbf{K}_\theta$ is positive semidefinite and $\lambda > 0$. We now extend this general deep kernel learning framework to the specific construction of hedging strategies.

3 Deep kernel hedging

In this section, we formulate the deep kernel hedging problem under the empirical measure. Let $S = (S_t)_{t\in[0,T]}$ denote the price process of a traded risky asset, let $B = (B_t)_{t\in[0,T]}$ be a strictly positive cash account with $B_0 = 1$, and let Y be a terminal liability payable at time T. Trading takes place at the discrete dates

$$0 = t_0 < t_1 < \cdots < t_n = T.$$

For $k = 0, \ldots, n-1$, let ξ_{t_k} and η_{t_k} denote, respectively, the numbers of units held in the risky asset and in the cash account over the interval $[t_k, t_{k+1})$. These holdings are assumed to be chosen using only the information available at time t_k. In particular, if $(\mathcal{F}_t)_{t\in[0,T]}$ denotes the market filtration, then ξ_{t_k} and η_{t_k} are $\mathcal{F}_{t_k}$-measurable. Immediately after rebalancing at time t_k, the portfolio value is

$$V_{t_k} = \xi_{t_k} S_{t_k} + \eta_{t_k} B_{t_k}, \qquad k = 0, \ldots, n-1,$$

with $V_0 = v_0 \in \mathbb{R}$ denoting the initial value of the hedging portfolio.

Definition 3.1 (Self-financing property)**.** The hedging strategy $(\xi_{t_k}, \eta_{t_k})_{k=0}^{n-1}$ is self-financing if

$$V_{t_{k+1}} - V_{t_k} = \xi_{t_k}\left(S_{t_{k+1}} - S_{t_k}\right) + \eta_{t_k}\left(B_{t_{k+1}} - B_{t_k}\right), \qquad k = 0, ..., n-1. \tag{3.1}$$

This condition means that all changes in the portfolio value are solely due to the gains and losses of the assets already held in the portfolio. In particular, no external capital is injected into or withdrawn from the strategy.

Throughout this paper, we consider hedging strategies satisfying the self-financing condition of Definition 3.1. In this case, the position in the cash account can be expressed in terms of the position in the risky asset as

$$\eta_{t_k} = \frac{V_{t_k} - \xi_{t_k} S_{t_k}}{B_{t_k}}, \qquad k = 0, \ldots, n-1.$$

Hence, the self-financing condition (3.1) implies

$$V_{t_k} = B_{t_k}\left[v_0 + \sum_{j=0}^{k-1} \xi_{t_j}\left(\frac{S_{t_{j+1}}}{B_{t_{j+1}}} - \frac{S_{t_j}}{B_{t_j}}\right)\right], \qquad k = 1, \ldots, n.$$

In particular, the terminal value of the hedging portfolio is

$$V_T = B_T\left[v_0 + \sum_{k=0}^{n-1} \xi_{t_k}\left(\frac{S_{t_{k+1}}}{B_{t_{k+1}}} - \frac{S_{t_k}}{B_{t_k}}\right)\right]. \tag{3.2}$$

Let $X_{t_k} \in \mathcal{X}$ be a vector of features observable at time t_k. We consider feedback strategies of the form

$$\xi_{t_k} = \phi(X_{t_k}), \qquad k = 0, \ldots, n-1,$$

where $\phi : \mathcal{X} \to \mathbb{R}$ is measurable. The input feature vector X_{t_k} can include time-augmented market variables such as the underlying price S_{t_k}, the cash account B_{t_k}, asset volatility, etc. Moreover, in more general non-Markovian settings where the hedging strategy may depend on the path of the market features, the learning inputs may include truncated time-augmented signatures of the market features. Path-signatures, originally introduced by Chen (1957) as sequences of iterated integrals, provide a compact and tractable way to encode path-dependent information. See Abi Jaber and Gérard (2025); Cirone and Salvi (2025) for applications of signatures to the learning of optimal hedging strategies via machine learning approaches. For further details on path-signatures, we refer to Appendix 6.2 and the references therein.

Our objective is now to learn the optimal hedging functional ϕ from sample data using a deep kernel-based approach. We assume access to a dataset of N sample paths observed at the trading dates, defined as

$$\mathcal{D}_N = \left\{\left(\left(S^i_{t_k}, B^i_{t_k}, X^i_{t_k}\right)_{k=0}^{n}, Y^i\right)\right\}_{i=1}^{N}.$$

Each sample path i therefore contains the underlying trajectory, the cash account and input features. For each path i, we define the gain associated with one unit of the risky asset held over $[t_k, t_{k+1})$ by

$$G^i_{t_{k+1}} := B^i_T\left(\frac{S^i_{t_{k+1}}}{B^i_{t_{k+1}}} - \frac{S^i_{t_k}}{B^i_{t_k}}\right).$$

It follows from (3.2) that the terminal hedging error (or negative P&L) $\mathcal{E}_i$ generated by ϕ along path i is the random variable

$$\mathcal{E}_i(\phi; v_0) := Y^i - V^i_T = Y^i - B^i_T v_0 - \sum_{k=0}^{n-1} \phi(X^i_{t_k}) G^i_{t_{k+1}}, \tag{3.3}$$

which we simply write $\mathcal{E}_i(\phi)$ when the initial capital v_0 is clear from the context. Let $\mathcal{L} : \mathbb{R} \to \mathbb{R}$ be a continuous convex bounded from below loss function, and let $\mathcal{M}$ be a prescribed class of admissible measurable hedging functions. The empirical hedging problem can then be written

$$\inf_{\phi \in \mathcal{M}} \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}\left(\mathcal{E}_i(\phi)\right) = \inf_{\phi \in \mathcal{M}} \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}\left(Y^i - B^i_T v_0 - \sum_{k=0}^{n-1} \phi(X^i_{t_k})\, G^i_{t_{k+1}}\right).$$

The choice $\mathcal{L}(u) = u^2$ gives the classical quadratic hedging problem, see Section 3.4.1. Other convex losses can be used to emphasize different properties of the terminal hedging error $\mathcal{E}_i(\phi)$, such as the CVaR loss (3.30) in Section 3.4.2.

We now restrict the space of measurable functionals $\mathcal{M}$ to RKHS. In this sense, for $\theta \in \Theta$, let $\mathcal{H}_{K_\theta}$ be the RKHS associated with the kernel K_θ on $\mathcal{X}$. The regularized deep kernel hedging problem is

$$\inf_{\theta\in\Theta,\, \phi\in\mathcal{H}_{K_\theta}} \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}\left(Y^i - B_T^i v_0 - \sum_{k=0}^{n-1} \phi(X_{t_k}^i)\, G_{t_{k+1}}^i\right) + \lambda\|\phi\|_{\mathcal{H}_{K_\theta}}^2, \tag{3.4}$$

with $\lambda > 0$. For fixed $\theta \in \Theta$, define the functional

$$\mathcal{J}_\theta(\phi) := \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}\left(Y^i - B_T^i v_0 - \sum_{k=0}^{n-1} \phi(X_{t_k}^i)\, G_{t_{k+1}}^i\right) + \lambda\|\phi\|_{\mathcal{H}_{K_\theta}}^2, \quad \phi \in \mathcal{H}_{K_\theta},$$

such that the hedging problem can be rewritten as

$$\inf_{\theta\in\Theta}\left\{\inf_{\phi\in\mathcal{H}_{K_\theta}} \mathcal{J}_\theta(\phi)\right\}. \tag{3.5}$$

We first address in Section 3.1 the well-posedness of the inner problem in (3.5) for fixed $\theta \in \Theta$ and show that, using a generalized version of the representer theorem, this infinite-dimensional optimization problem admits an equivalent finite-dimensional reformulation, leading to an explicit characterization of the unique optimizer $\phi^\star$. Then, in Section 3.2, we reformulate the deep kernel hedging problem and establish the existence of a minimizer $\theta^\star \in \Theta$. Finally, in Section 3.3, we develop a Random Fourier Feature approximation for deep kernel hedging. The following Table 1 summarizes the notations used in following sections.

Table 1: Notation for hedging functions. Here, N is the number of training paths and D the number of RFF.

Notation	Description
ϕ	Generic hedging function, defining the risky-asset position through $\xi_{t_k} = \phi(X_{t_k})$.
$\phi^\star_{N,\theta}$	Optimal exact RKHS hedging function for a fixed kernel parameter θ.
$\phi^\star_N$	A globally optimal exact deep kernel hedge $\phi^\star_N = \phi^\star_{N,\theta^\star}$, where $\theta^\star$ is also optimized.
$\phi^*_{D,\theta}$	Optimal RFF hedging function for a fixed kernel parameter θ.
$\phi^\star_D$	A globally optimal hedge for the RFF approximation, given by $\phi^\star_D(x) = (\boldsymbol{\beta}^\star_D)^\top y_{\theta^\star_D}(x)$, where both $\theta^\star_D$ and $\boldsymbol{\beta}^\star_D$ are optimized.
$\widehat{\phi}_D$	RFF hedging function returned by numerical training (without optimality guarantee).

3.1 Representer theorem and finite-dimensional formulation

We first focus on the inner problem in (3.5) for fixed $\theta \in \Theta$.

Lemma 3.1 (Existence and uniqueness for the inner problem)**.** *Suppose that $\mathcal{L} : \mathbb{R} \to \mathbb{R}$ is continuous, convex and bounded from below, and let $\lambda > 0$. Then, for every fixed $\theta \in \Theta$, the problem*

$$\inf_{\phi\in\mathcal{H}_{K_\theta}} \mathcal{J}_\theta(\phi)$$

admits a unique minimizer $\phi^\star_{N,\theta} \in \mathcal{H}_{K_\theta}$.

Proof. The proof relies on standard arguments for minimization problems in Hilbert spaces that can be found in Bauschke and Combettes (2017). Since the zero function belongs to $\mathcal{H}_{K_\theta}$, we have

$$0 \le \inf_{\phi\in\mathcal{H}_{K_\theta}} \mathcal{J}_\theta(\phi) \le \mathcal{J}_\theta(0) < \infty.$$

Choose a minimizing sequence $(\phi_m)_{m\geq 1} \subset \mathcal{H}_{K_\theta}$ such that

$$\mathcal{J}_\theta(\phi_m) \leq \inf_{\phi\in\mathcal{H}_{K_\theta}} \mathcal{J}_\theta(\phi) + \frac{1}{m}.$$

Since $\mathcal{L}$ is bounded from below, there exists a constant $C_{\mathcal{L}} \in \mathbb{R}$ such that $\mathcal{L} \geq C_{\mathcal{L}}$. Hence

$$C_{\mathcal{L}} + \lambda\|\phi_m\|^2_{\mathcal{H}_{K_\theta}} \leq \mathcal{J}_\theta(\phi_m) \leq \inf_{\phi\in\mathcal{H}_{K_\theta}} \mathcal{J}_\theta(\phi) + \frac{1}{m} \leq \mathcal{J}_\theta(0) + 1.$$

Therefore,

$$\lambda\|\phi_m\|^2_{\mathcal{H}_{K_\theta}} \leq \mathcal{J}_\theta(0) + 1 - C_{\mathcal{L}},$$

and consequently

$$\|\phi_m\|_{\mathcal{H}_{K_\theta}} \leq \sqrt{\frac{\mathcal{J}_\theta(0) + 1 - C_{\mathcal{L}}}{\lambda}}.$$

Thus, $(\phi_m)_{m\geq 1}$ is bounded in $\mathcal{H}_{K_\theta}$. As $\mathcal{H}_{K_\theta}$ is a Hilbert space, there exists a subsequence, still denoted by (ϕ_m), and some $\overline{\phi} \in \mathcal{H}_{K_\theta}$ such that

$$\phi_m \rightharpoonup \overline{\phi} \qquad \text{weakly in } \mathcal{H}_{K_\theta}.$$

For each $i = 1, \ldots, N$, define

$$L_i(\phi) := \sum_{k=0}^{n-1} G^i_{t_{k+1}}\, \phi(X^i_{t_k}).$$

Because point evaluations are continuous linear functionals on an RKHS, L_i is a continuous linear functional on $\mathcal{H}_{K_\theta}$. Hence,

$$L_i(\phi_m) \to L_i(\overline{\phi}), \qquad i = 1, \ldots, N.$$

The continuity of $\mathcal{L}$ then gives

$$\frac{1}{N}\sum_{i=1}^{N} \mathcal{L}\left(Y^i - B^i_T v_0 - L_i(\phi_m)\right) \to \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}\left(Y^i - B^i_T v_0 - L_i(\overline{\phi})\right).$$

Moreover, the squared Hilbert-space norm is weakly lower semicontinuous. Consequently,

$$\mathcal{J}_\theta(\overline{\phi}) \leq \liminf_{m\to\infty} \mathcal{J}_\theta(\phi_m),$$

so $\overline{\phi}$ is a minimizer. Finally, the empirical loss is convex in ϕ, while $\lambda\|\phi\|^2_{\mathcal{H}_{K_\theta}}$ is strictly convex because $\lambda > 0$. Thus, $\mathcal{J}_\theta$ is strictly convex and its minimizer is unique. □

We next derive a finite-dimensional representation of this minimizer $\phi^\star_{N,\theta}$.

Theorem 3.1 (Representer theorem for the inner problem)**.** *Fix $\theta \in \Theta$. For each $i = 1, \ldots, N$, define*

$$g_{\theta,i}(\cdot) := \sum_{k=0}^{n-1} G^i_{t_{k+1}} K_\theta\big(X^i_{t_k}, \cdot\big) \in \mathcal{H}_{K_\theta},$$

and let the associated hedging Gram matrix $\mathbf{Q}_\theta \in \mathbb{R}^{N\times N}$ be given by

$$(\mathbf{Q}_\theta)_{ij} := \langle g_{\theta,i}, g_{\theta,j}\rangle_{\mathcal{H}_{K_\theta}} = \sum_{k,\ell=0}^{n-1} G^i_{t_{k+1}} G^j_{t_{\ell+1}} K_\theta\big(X^i_{t_k}, X^j_{t_\ell}\big). \tag{3.6}$$

Moreover, define

$$\mathbf{Y}^{v_0} := \big(Y^1 - B^1_T v_0, \ldots, Y^N - B^N_T v_0\big)^\top \in \mathbb{R}^N. \tag{3.7}$$

Then,

$$\min_{\phi\in\mathcal{H}_{K_\theta}} \mathcal{J}_\theta(\phi) = \min_{\boldsymbol{\alpha}\in\mathbb{R}^N} \left\{ \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}\left((\mathbf{Y}^{v_0} - \mathbf{Q}_\theta\boldsymbol{\alpha})_i\right) + \lambda\boldsymbol{\alpha}^\top\mathbf{Q}_\theta\boldsymbol{\alpha} \right\}. \tag{3.8}$$

Moreover, the unique optimal hedging function is of the form

$$\phi^\star_{N,\theta}(\cdot) = \sum_{i=1}^{N} \alpha^\star_{\theta,i}\, g_{\theta,i} = \sum_{i=1}^{N} \alpha^\star_{\theta,i} \sum_{k=0}^{n-1} G^i_{t_{k+1}} K_\theta(X^i_{t_k}, \cdot), \tag{3.9}$$

where $\boldsymbol{\alpha}^\star_\theta$ *is any minimizer of* (3.8).

Proof. For $i = 1, \dots, N$, define

$$L_i(\phi) := \sum_{k=0}^{n-1} G^i_{t_{k+1}}\, \phi(X^i_{t_k}).$$

Since $\mathcal{H}_{K_\theta}$ is a RKHS, the reproducing property implies that

$$L_i(\phi) = \langle \phi, g_{\theta,i} \rangle_{\mathcal{H}_{K_\theta}}.$$

Let

$$\mathcal{H}_{\theta,0} := \operatorname{span}\{g_{\theta,1}, \dots, g_{\theta,N}\}.$$

Since $\mathcal{H}_{\theta,0}$ is finite-dimensional, it is closed. Every $\phi \in \mathcal{H}_{K_\theta}$ therefore admits the orthogonal decomposition

$$\phi = \phi_\parallel + \phi_\perp, \qquad \phi_\parallel \in \mathcal{H}_{\theta,0}, \quad \phi_\perp \in \mathcal{H}^\perp_{\theta,0}.$$

Because $g_{\theta,i} \in \mathcal{H}_{\theta,0}$,

$$L_i(\phi) = \langle \phi, g_{\theta,i} \rangle_{\mathcal{H}_{K_\theta}} = \langle \phi_\parallel, g_{\theta,i} \rangle_{\mathcal{H}_{K_\theta}}.$$

Furthermore,

$$\|\phi\|^2_{\mathcal{H}_{K_\theta}} = \|\phi_\parallel\|^2_{\mathcal{H}_{K_\theta}} + \|\phi_\perp\|^2_{\mathcal{H}_{K_\theta}}.$$

It follows that

$$\mathcal{J}_\theta(\phi) = \mathcal{J}_\theta(\phi_\parallel) + \lambda \|\phi_\perp\|^2_{\mathcal{H}_{K_\theta}}.$$

Since $\lambda > 0$, every minimizer must satisfy $\phi_\perp = 0$. Thus,

$$\phi^\star_{N,\theta} \in \mathcal{H}_{\theta,0}.$$

Consequently, there exists $\boldsymbol{\alpha}^\star_\theta \in \mathbb{R}^N$ such that

$$\phi^\star_{N,\theta}(\cdot) = \sum_{j=1}^{N} \alpha^\star_{\theta,j} g_{\theta,j}(\cdot).$$

For such a function,

$$L_i(\phi) = \sum_{j=1}^{N} \alpha^\star_{\theta,j} \langle g_{\theta,j}, g_{\theta,i} \rangle_{\mathcal{H}_{K_\theta}} = (\mathbf{Q}_\theta \boldsymbol{\alpha}^\star_\theta)_i,$$

and

$$\|\phi\|^2_{\mathcal{H}_{K_\theta}} = \sum_{i,j=1}^{N} \alpha^\star_{\theta,i} \alpha^\star_{\theta,j} \langle g_{\theta,i}, g_{\theta,j} \rangle_{\mathcal{H}_{K_\theta}} = (\boldsymbol{\alpha}^\star_\theta)^\top \mathbf{Q}_\theta \boldsymbol{\alpha}^\star_\theta.$$

Substitution into $\mathcal{J}_\theta$ yields (3.8) and (3.9). □

Remark 3.1. For the square loss $\mathcal{L}(u) = u^2$, one choice of optimal coefficients $\boldsymbol{\alpha}^\star_\theta$ is

$$\boldsymbol{\alpha}^\star_\theta = (\mathbf{Q}_\theta + N\lambda I_N)^{-1} \mathbf{Y}^{v_0}. \tag{3.10}$$

3.2 Existence of an optimal deep kernel hedge

We now establish existence of a solution to the joint optimization (3.5) over the kernel parameter $\theta \in \Theta$ and the hedging function $\phi \in \mathcal{H}_{K_\theta}$.

Assumption 1. *The following conditions hold:*

$$\Theta \subset \mathbb{R}^m \text{ is compact,} \tag{H1}$$

$$\theta \mapsto K_\theta(x, x') \text{ is continuous on } \Theta, \qquad \forall x, x' \in \mathcal{X}. \tag{H2}$$

Lemma 3.2. *Under Assumption 1, there exist* $(\theta^\star, \boldsymbol{\alpha}^\star_{\theta^\star}) \in \Theta \times \mathbb{R}^N$ *solution to*

$$\inf_{\theta \in \Theta,\, \boldsymbol{\alpha} \in \mathbb{R}^N} \left\{ \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left((\mathbf{Y}^{v_0} - \mathbf{Q}_\theta \boldsymbol{\alpha})_i \right) + \lambda \boldsymbol{\alpha}^\top \mathbf{Q}_\theta \boldsymbol{\alpha} \right\}. \tag{3.11}$$

Proof. The proof is deferred to Appendix 6.1.1. □

Theorem 3.2 (Existence of an optimal deep kernel hedge)**.** *Under Assumption 1, the deep kernel hedging problem (3.4) admits a global minimizer. More precisely, there exist*

$$\theta^\star \in \Theta \qquad \text{and} \qquad \phi^\star \in \mathcal{H}_{K_{\theta^\star}}$$

such that

$$\mathcal{J}_{\theta^\star}(\phi^\star) = \inf_{\theta \in \Theta} \inf_{\phi \in \mathcal{H}_{K_\theta}} \mathcal{J}_\theta(\phi). \tag{3.12}$$

In particular, $\phi^\star_N(\cdot)$ *defined by*

$$\phi^\star_N(\cdot) := \phi^\star_{N,\theta^\star}(\cdot) = \sum_{i=1}^{N} \alpha^\star_{\theta^\star,i} \sum_{k=0}^{n-1} G^i_{t_{k+1}} K_{\theta^\star}\left(X^i_{t_k}, \cdot\right), \tag{3.13}$$

with $(\theta^\star, \boldsymbol{\alpha}^\star_{\theta^\star})$ *solution of (3.11), is an optimal deep kernel hedging strategy.*

Proof. The proof is a direct consequence of Theorem 3.1 and Lemma 3.2. □

Remark 3.2. The optimal kernel parameter $\theta^\star$ need not be unique. Moreover, if $\mathbf{Q}_{\theta^\star}$ is singular, the coefficient vector $\boldsymbol{\alpha}^\star$ need not be unique either. Nevertheless, Lemma 3.1 ensures that, for each fixed minimizer $\theta^\star$, the corresponding optimal RKHS hedging function $\phi^\star_{N,\theta^\star}$ is unique. ■

The finite-dimensional formulation (3.11) provides a natural starting point for computing the deep kernel hedge characterized by Theorem 3.2. For the squared loss, one may substitute for each θ the closed-form solution (3.10) for the optimal coefficients $\boldsymbol{\alpha}^\star$ into the objective and optimize the resulting reduced objective over θ using gradient-based methods. However, a direct implementation is computationally demanding: constructing the hedging Gram matrix $\mathbf{Q}_\theta \in \mathbb{R}^{N \times N}$ requires $\mathcal{O}(N^2 n^2)$ kernel evaluations, storing it requires $\mathcal{O}(N^2)$ memory, and solving the associated linear system using a standard solver requires $\mathcal{O}(N^3)$ operations. These computations must then be repeated whenever θ changes.

Alternatively, for general loss functions or to avoid repeatedly solving the inner linear system, one may instead optimize θ and $\boldsymbol{\alpha}$ jointly using stochastic gradient methods applied to (3.11). However, this approach again requires storing and manipulating the Gram matrix $\mathbf{Q}_\theta$, resulting in a memory complexity of order $\mathcal{O}(N^2)$ and a computational cost of order $\mathcal{O}(N^2 n^2)$ at every training iteration, or $\mathcal{O}(NBn^2)$ when using mini-batching with batch size B. This cost can be prohibitive in the deep kernel setting because each update of the network parameters θ changes the kernel K_θ and therefore requires the corresponding Gram matrix entries to be recomputed. Moreover, although the number N of paths is typically moderate in financial applications, the total number Nn of training observations can be large, as illustrated in Section 4, particularly for payoffs with relatively long maturities. The next section introduces a random Fourier feature approximation that avoids explicitly constructing this Gram matrix and leads to a substantially more scalable formulation.

3.3 Random Fourier features for deep kernel hedging

To obtain a more scalable formulation of our deep kernel hedging problem, we use a random Fourier feature approach. Random Fourier features (RFF), introduced by Rahimi and Recht (2007) and further analyzed by Rahimi and Recht (2008); Sutherland and Schneider (2015); Avron et al. (2017), provide an explicit finite-dimensional approximation of continuous shift-invariant kernels. Importantly, the deep kernel K_θ needs not itself be shift-invariant as a function of the original inputs. We only require the base kernel K in (2.4) acting on the latent space $\mathcal{Z}_\theta = \psi_\theta(\mathcal{X}) \subseteq \mathbb{R}^p$ to be shift-invariant. More precisely, a base kernel K is called shift-invariant, if it depends only on the difference between its arguments, i.e.,

$$K(z, z') = \kappa(z - z'), \quad \forall z, z' \in \mathbb{R}^p,$$

for some continuous function $\kappa : \mathbb{R}^p \to \mathbb{R}$. Examples of shift-invariant kernels include the radial basis function (RBF) kernel. The starting point for the approximation is Bochner's theorem, which characterizes continuous, positive-definite, shift-invariant kernels.

Theorem 3.3 (Bochner). *A continuous function $\kappa : \mathbb{R}^p \to \mathbb{R}$ is positive definite if and only if there exists a unique finite non-negative Borel measure p_K on $\mathbb{R}^p$ such that*

$$\kappa(z) = \int_{\mathbb{R}^p} e^{\mathrm{i}w^\top z}\, p_K(dw), \qquad z \in \mathbb{R}^p.$$

We further assume that the base kernel is normalized, so that $\kappa(0) = 1$, and that $\kappa \in L^1(\mathbb{R}^p)$. Its spectral measure p_K is then an absolutely continuous probability measure. Bochner's representation then becomes

$$K(z, z') = \int_{\mathbb{R}^p} e^{\mathrm{i}w^\top (z - z')}\, p_K(w) dw. \tag{3.14}$$

This representation (3.14) thus shows that positive definite shift-invariant normalized kernels can be expressed as expectations over complex exponentials. Following Rahimi and Recht (2007) and since K is real-valued, we can rewrite

$$K(z, z') = 2\, \mathbb{E}_{(w,b)\sim p_K \otimes \mathcal{U}(0,2\pi)} \left[\cos\left(w^\top z + b\right) \cos\left(w^\top z' + b\right)\right]. \tag{3.15}$$

Applying this representation to $z = \psi_\theta(x)$ and $z' = \psi_\theta(x')$ gives for the deep kernel (2.4) the formulation

$$K_\theta(x, x') = K(\psi_\theta(x), \psi_\theta(x')) = 2\, \mathbb{E}_{(w,b)\sim p_K \otimes \mathcal{U}(0,2\pi)} \left[\cos\left(w^\top \psi_\theta(x) + b\right) \cos\left(w^\top \psi_\theta(x') + b\right)\right].$$

The key idea behind random Fourier features is to approximate this expectation by Monte Carlo sampling. For a given number of random features $D \ll N$, let

$$(w_j, b_j) \overset{\text{i.i.d.}}{\sim} p_K \otimes \mathcal{U}(0, 2\pi), \qquad j = 1, \ldots, D,$$

and define

$$y_\theta(x) := \sqrt{\frac{2}{D}} \begin{bmatrix} \cos\left(w_1^\top \psi_\theta(x) + b_1\right) \\ \vdots \\ \cos\left(w_D^\top \psi_\theta(x) + b_D\right) \end{bmatrix} \in \mathbb{R}^D. \tag{3.16}$$

The deep kernel K_θ in (2.4) then admits the finite approximation

$$\widetilde{K}_{\theta,D}(x, x') := y_\theta(x)^\top y_\theta(x'), \quad x, x' \in \mathcal{X}. \tag{3.17}$$

The random frequencies and phases $\{(w_j, b_j)\}_{j=1}^D$ defining y_θ are sampled once and kept fixed throughout training. Replacing K_θ by $\widetilde{K}_{\theta,D}$ in (3.4) yields the RFF hedging problem

$$\inf_{\theta\in\Theta} \inf_{\phi \in \mathcal{H}_{\widetilde{K}_{\theta,D}}} \left\{ \frac{1}{N} \sum_{i=1}^N \mathcal{L}\left((\mathbf{Y}^{v_0})_i - \sum_{k=0}^{n-1} \phi(X_{t_k}^i) G_{t_{k+1}}^i \right) + \lambda \|\phi\|^2_{\mathcal{H}_{\widetilde{K}_{\theta,D}}} \right\}. \tag{3.18}$$

Moreover, for each sample path i, we define the aggregated feature vector

$$Z_\theta^i := \sum_{k=0}^{n-1} y_\theta\left(X_{t_k}^i\right) G_{t_{k+1}}^i \in \mathbb{R}^D, \tag{3.19}$$

as well as the matrix of aggregated features

$$\mathbf{Z}_\theta := \begin{bmatrix} (Z_\theta^1)^\top \\ \vdots \\ (Z_\theta^N)^\top \end{bmatrix} \in \mathbb{R}^{N\times D}.$$

Substituting (3.17) into the definition (3.6) of $\mathbf{Q}_\theta$ yields the approximation $\widetilde{\mathbf{Q}}_{\theta,D}$ given by

$$\widetilde{\mathbf{Q}}_{\theta,D} := \mathbf{Z}_\theta \mathbf{Z}_\theta^\top.$$

Applying Theorem 3.1 to $\widetilde{K}_{\theta,D}$ hence yields the following equivalent finite-dimensional formulation of (3.18)[1]

$$\min_{\theta\in\Theta,\, \boldsymbol{\alpha}\in\mathbb{R}^N} \left\{ \frac{1}{N}\sum_{i=1}^N \mathcal{L}\left((\mathbf{Y}^{v_0} - \widetilde{\mathbf{Q}}_{\theta,D}\boldsymbol{\alpha})_i \right) + \lambda \boldsymbol{\alpha}^\top \widetilde{\mathbf{Q}}_{\theta,D}\boldsymbol{\alpha} \right\}. \tag{3.20}$$

The following lemma reformulates the optimization (3.20) over $\boldsymbol{\alpha} \in \mathbb{R}^N$, for fixed θ, as an equivalent problem over $\boldsymbol{\beta} \in \mathbb{R}^D$. This formulation avoids constructing the Gram matrix and forms the basis of the efficient optimization algorithm described below.

Lemma 3.3 (Primal formulation)**.** *Let $\lambda > 0$ and fix $\theta \in \Theta$. Then,*

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^N} \left\{ \frac{1}{N}\sum_{i=1}^N \mathcal{L}\left((\mathbf{Y}^{v_0} - \mathbf{Z}_\theta\mathbf{Z}_\theta^\top\boldsymbol{\alpha})_i \right) + \lambda \|\mathbf{Z}_\theta^\top\boldsymbol{\alpha}\|_2^2 \right\} = \min_{\boldsymbol{\beta}\in\mathbb{R}^D} \left\{ \frac{1}{N}\sum_{i=1}^N \mathcal{L}\left((\mathbf{Y}^{v_0} - \mathbf{Z}_\theta\boldsymbol{\beta})_i \right) + \lambda \|\boldsymbol{\beta}\|_2^2 \right\}. \tag{3.21}$$

Moreover, the unique minimizer $\phi^\star_{D,\theta}$ of the inner RFF problem in (3.18) satisfies

$$\phi^\star_{D,\theta}(x) = (\boldsymbol{\beta}^\star_\theta)^\top y_\theta(x), \quad x \in \mathcal{X} \tag{3.22}$$

where $\boldsymbol{\beta}^\star_\theta \in \mathbb{R}^D$ is the unique minimizer of the right-hand side in (3.21).

Proof. The proof is deferred in Appendix 6.1.2. □

By Lemma 3.3, the RFF approximation of the deep kernel hedging (3.20) is equivalently given by

$$\min_{\theta\in\Theta,\, \boldsymbol{\beta}\in\mathbb{R}^D} \left\{ \frac{1}{N}\sum_{i=1}^N \mathcal{L}\left((\mathbf{Y}^{v_0} - \mathbf{Z}_\theta\boldsymbol{\beta})_i \right) + \lambda \|\boldsymbol{\beta}\|_2^2 \right\}. \tag{3.23}$$

Let $(\theta^\star_D, \boldsymbol{\beta}^\star_D := \boldsymbol{\beta}^\star_{\theta^\star_D})$ solve (3.23). The corresponding optimal RFF hedging strategy minimizing (3.18) is then given by

$$\phi^\star_D(\cdot) := (\boldsymbol{\beta}^\star_D)^\top y_{\theta^\star_D}(\cdot).$$

We now show with Theorem 3.4 that both the optimal value (3.23) of the RFF problem and the minimizer ϕ^*_D converge to those of the exact deep kernel hedging problem (3.4) when the number D of Fourier features increases. First, we need the following extension of (Rahimi and Recht, 2007, Claim 1).

Proposition 3.1 (Uniform approximation)**.** *Let $\mathcal{X} \subset \mathbb{R}^d$ and $\Theta \subset \mathbb{R}^m$ be nonempty compact sets. Suppose that the map $(\theta, x) \mapsto \psi_\theta(x)$ is jointly continuous on $\Theta \times \mathcal{X}$ and that the spectral distribution p_K of the base kernel K satisfies*

$$\mathbb{E}_{w\sim p_K}\left[\|w\|_2^2\right] < \infty.$$

For each D, construct $\widetilde{K}_{\theta,D}$ using the same draw of random Fourier features $\{(w_j, b_j)\}_{j=1}^D \overset{\text{i.i.d.}}{\sim} p_K \otimes \mathcal{U}(0, 2\pi)$ for every $\theta \in \Theta$. Then, there exist constants $C, c > 0$, independent of D and ε, such that, for every $D \geq 1$ and $0 < \varepsilon \leq 1$,

$$\mathbb{P}\left(\sup_{\theta\in\Theta} \sup_{x,x'\in\mathcal{X}} \left| \widetilde{K}_{\theta,D}(x,x') - K_\theta(x,x') \right| > \varepsilon \right) \leq C\varepsilon^{-2}\exp(-cD\varepsilon^2) \tag{3.24}$$

[1] Under Assumption 1, the existence of a solution to (3.18) and (3.20) follows by the same arguments as those used in the proof of Lemma 3.2 and Theorem 3.2 with K_θ replaced by $\widetilde{K}_{\theta,D}$. In particular, the required continuity holds if $\theta \mapsto y_\theta(x)$ is continuous for every $x \in \mathcal{X}$.

In particular,

$$\sup_{\theta\in\Theta}\sup_{x,x'\in\mathcal{X}}\left|\widetilde{K}_{\theta,D}(x,x')-K_\theta(x,x')\right|\to 0 \qquad \text{almost surely as } D\to\infty. \tag{3.25}$$

The corresponding hedging Gram matrices satisfy

$$\sup_{\theta\in\Theta}\left\|\widetilde{\mathbf{Q}}_{\theta,D}-\mathbf{Q}_\theta\right\|_{\mathrm{op}}\to 0 \qquad \text{almost surely as } D\to\infty. \tag{3.26}$$

Proof. The proof is deferred in Appendix 6.1.3. □

We now establish the main result of this section on the convergence of the RFF approximation to the original deep kernel hedging problem.

Theorem 3.4 (Convergence of the RFF hedging problem)**.** *Suppose that Assumption 1 and the assumptions of Proposition 3.1 hold. Let $\mathcal{V}:=\mathcal{J}_{\theta^\star}(\phi^\star)$ denote the optimal value (3.12) of the exact deep-kernel hedging problem, and let $\mathcal{V}_D$ denote the optimal value of its RFF approximation, i.e.,*

$$\mathcal{V}_D:=\min_{\theta\in\Theta,\,\boldsymbol{\beta}\in\mathbb{R}^D}\left\{\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}\left((\mathbf{Y}^{v_0}-\mathbf{Z}_\theta\boldsymbol{\beta})_i\right)+\lambda\|\boldsymbol{\beta}\|^2\right\}.$$

Then

$$\mathcal{V}_D\to\mathcal{V} \qquad \text{almost surely as } D\to\infty.$$

For each D, let

$$(\theta_D^\star,\boldsymbol{\beta}_D^\star)\in\operatorname*{argmin}_{\theta\in\Theta,\,\boldsymbol{\beta}\in\mathbb{R}^D}\left\{\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}\left((\mathbf{Y}^{v_0}-\mathbf{Z}_\theta\boldsymbol{\beta})_i\right)+\lambda\|\boldsymbol{\beta}\|_2^2\right\}, \tag{3.27}$$

and define the associated RFF hedging strategy by

$$\phi_D^\star(x):=\boldsymbol{\beta}_{D,\cdot}^{\star\top}y_{\theta_D^\star}(x), \qquad x\in\mathcal{X}. \tag{3.28}$$

Then every accumulation point of the sequence $\{(\theta_D^\star,\phi_D^\star)\}_{D>1}$ in $\Theta\times C(\mathcal{X})$, where $C(\mathcal{X})$ is equipped with the uniform norm, is a global minimizer of the exact problem. In particular, if the exact problem (3.12) has a unique minimizing pair $(\theta^\star,\phi^\star)$, then

$$\theta_D^*\to\theta^\star \qquad \text{and} \qquad \sup_{x\in\mathcal{X}}|\phi_D^*(x)-\phi^\star(x)|\to 0 \qquad \text{almost surely as } D\to\infty.$$

Proof. The proof is deferred in Appendix 6.1.4. □

Based on the formulation of the hedging problem with the RFF approximation given by (3.23), we can implement a deep approach to jointly optimize $(\theta,\boldsymbol{\beta})$ using a stochastic gradient-based method. The RFF approximation replaces the $N\times N$ Gram matrix $\mathbf{Q}_\theta$ by an explicit $N\times D$ Fourier feature matrix $\mathbf{Z}_\theta$, together with a coefficient vector $\boldsymbol{\beta}\in\mathbb{R}^D$. Thus, the kernel representation scales linearly in the number of random features D, rather than quadratically in the number of training samples. At every training iteration (with θ fixed), the RFF-based approximate kernel approach has a memory complexity of order $\mathcal{O}(ND)$ and a computational cost of order $\mathcal{O}(NnD)$, or $\mathcal{O}(BnD)$ when using mini-batches of size B. Whenever $D\ll nN$, this is significantly lower than the $\mathcal{O}(N^2n^2)$ computational cost of the corresponding full kernel-based hedging method (or $\mathcal{O}(BNn^2)$ when using mini-batches). Hence, the RFF formulation avoids storing or manipulating the full Gram matrix, making the proposed RFF-based kernel hedging approach substantially more scalable than its exact deep-kernel counterpart, especially when combined with stochastic gradient-based optimization as in the following algorithms.

3.4 The deep kernel hedging algorithm

To illustrate the flexibility of our deep kernel hedging framework, we consider two popular and complementary loss functions and design their corresponding algorithm.

3.4.1 Quadratic hedging

Given a dataset of N sample paths with the same given initial conditions, it is common in quadratic hedging problems to also learn the initial price $v_0 \in \mathbb{R}$, such that the problem is given by

$$\inf_{v_0\in\mathbb{R},\ \theta\in\Theta,\ \phi\in\mathcal{H}_{K_\theta}} \frac{1}{N}\sum_{i=1}^{N}\left(Y^i - B_T^i v_0 - \sum_{k=0}^{n-1}\phi(X_{t_k}^i)\,G_{t_{k+1}}^i\right)^2 + \lambda\|\phi\|_{\mathcal{H}_{K_\theta}}^2.$$

This hedging problem fits into our convex hedging setting and is therefore well-posed.

Remark 3.3. The inclusion of v_0 does not affect existence for any continuous, convex and coercive loss function $\mathcal{L}$. Indeed, since $B_T^i > 0$ for every i, the loss controls unbounded sequences in v_0, while the regularization term controls unbounded sequences in ϕ. The same compactness and lower-semicontinuity argument as in Lemma 3.1 and Lemma 3.2 therefore yields a global minimizer $(v_0^\star, \theta^\star, \phi^\star)$. In particular, this applies to the squared loss above. ■

To avoid the computational burden of storing the full $\mathbb{R}^{N\times N}$ hedging Gram matrix $\mathbf{Q}_\theta$, we consider the RFF approximation from the previous section and solve the problem

$$\min_{v_0\in\mathbb{R},\,\theta\in\Theta,\ \boldsymbol{\beta}\in\mathbb{R}^D} \frac{1}{N}\|\mathbf{Y}^{v_0} - \mathbf{Z}_\theta\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|^2, \tag{3.29}$$

where $\mathbf{Z}_\theta$ is defined by (3.19) and the RFF hedging strategy given by (3.28). This formulation leads to Algorithm 1, which implements a fully stochastic gradient descent procedure to this finite-dimensional problem and returns an approximate numerical candidate solution $(\widehat{v}_0, \widehat{\theta}_D, \widehat{\boldsymbol{\beta}}_D)$.

3.4.2 CVaR hedging

The quadratic hedging objective introduced in Section 3.4.1 penalizes positive and negative terminal hedging errors $\{\mathcal{E}_i(\phi)\}_{i=1}^N$ symmetrically and controls their overall dispersion. It does not, however, specifically target adverse tail outcomes. To emphasize large hedging losses, we can replace the quadratic objective by a conditional value-at-risk (CVaR) objective. Let $\tilde{\alpha} \in (0,1)$ denote the lower-tail probability. The CVaR hedging problem consists in minimizing the upper-tail $\mathrm{CVaR}_{1-\tilde{\alpha}}^{\mathrm{upper}}$ of the terminal hedging shortfall $\mathcal{E}_i(\phi)$ at confidence level $1-\tilde{\alpha}$. Equivalently, this amounts to maximizing the lower-tail $\mathrm{CVaR}_{\tilde{\alpha}}^{\mathrm{lower}}$ of the terminal P&L $-\mathcal{E}_i(\phi)$, that is, its average value over the worst $\tilde{\alpha}$ fraction of outcomes. Using the Rockafellar–Uryasev representation (Buehler et al., 2019), the corresponding regularized empirical problem can be written

$$\begin{aligned}\inf_{\theta\in\Theta,\phi\in\mathcal{H}_{K_\theta}} &\left\{\mathrm{CVaR}_{1-\tilde{\alpha}}^{\mathrm{upper}}\left(\{\mathcal{E}_i(\phi;v_0)\}_{i=1}^N\right) + \lambda\|\phi\|_{\mathcal{H}_{K_\theta}}^2\right\} \\ &= \inf_{\theta\in\Theta,\,\phi\in\mathcal{H}_{K_\theta},\,\zeta\in\mathbb{R}}\left\{\zeta + \frac{1}{\tilde{\alpha}N}\sum_{i=1}^{N}\left(Y^i - B_T^i v_0 - \sum_{k=0}^{n-1}\phi(X_{t_k}^i)G_{t_{k+1}}^i - \zeta\right)_+ + \lambda\|\phi\|_{\mathcal{H}_{K_\theta}}^2\right\},\end{aligned} \tag{3.30}$$

where $\zeta \in \mathbb{R}$ is an auxiliary optimization variable. Note that at any optimum $(\theta^*, \phi^*, \zeta^*)$, $\zeta^\star$ is an empirical $(1-\tilde{\alpha})$-quantile of the terminal hedging shortfall, $\zeta^\star \in q_{1-\tilde{\alpha}}^{\mathrm{upper}}\left(\left\{\mathcal{E}_i(\phi_{N,\theta^*}^*; v_0)\right\}_{i=1}^N\right)$.

Remark 3.4. The Rockafellar–Uryasev integrand

$$\mathcal{L}_{\tilde{\alpha}}^{\mathrm{CVaR}}(u,\zeta) := \zeta + \frac{1}{\tilde{\alpha}}(u-\zeta)_+$$

is continuous and jointly convex, but it is neither bounded from below nor coercive jointly in (u,ζ). The initial portfolio value v_0 is therefore treated as a fixed initial constraint. In particular, when $B_T^i = B_T > 0$, the cash-invariance of CVaR gives $\mathrm{CVaR}_{1-\tilde{\alpha}}^{\mathrm{upper}}\left(\{\mathcal{E}_i(\phi;v_0)\}_{i=1}^N\right) = \mathrm{CVaR}_{1-\tilde{\alpha}}^{\mathrm{upper}}\left(\{\mathcal{E}_i(\phi;0)\}_{i=1}^N\right) - B_T v_0$. Consequently, unconstrained minimization over v_0 would make the objective unbounded below. A hedging price may instead be defined through an indifference-pricing criterion, as in Buehler et al. (2019). ■

Although the CVaR integrand is not bounded below jointly in (u,ζ), the regularized empirical problem (3.30) remains well posed.

Algorithm 1 Quadratic deep kernel hedging with RFF approximation

Input: Sample paths $\{S^i_{t_k}, B^i_{t_k}, X^i_{t_k}\}_{i=1,\dots,N;\ k=0,\dots,n}$, payoffs $\{Y^i\}_{i=1}^N$, shift-invariant base kernel K with spectral measure p_K, regularization parameter λ, learning rate η_e, number of epochs E, batch size N_{b}, and number of RFF features D.

1. For every $i = 1, \dots, N$ and $k = 0, \dots, n-1$, set

$$x_{i,k} = X^i_{t_k}, \qquad G^i_{t_{k+1}} = B^i_T \left(\frac{S^i_{t_{k+1}}}{B^i_{t_{k+1}}} - \frac{S^i_{t_k}}{B^i_{t_k}} \right).$$

2. Initialize:
   - Price parameter $v_0 \in \mathbb{R}$.
   - Neural network ψ_θ with parameters $\theta \in \Theta$.
   - Coefficients $\boldsymbol{\beta} \in \mathbb{R}^D$.
   - Random features $(w_j, b_j) \overset{\text{i.i.d.}}{\sim} p_K \otimes \mathcal{U}(0, 2\pi)$ for $j = 1, \dots, D$, and define

$$y_\theta(x) = \sqrt{\frac{2}{D}} \begin{bmatrix} \cos(w_1^T \psi_\theta(x) + b_1) \\ \vdots \\ \cos(w_D^T \psi_\theta(x) + b_D) \end{bmatrix}.$$

3. **For** epoch $e = 1, \dots, E$ **do**
   (a) Sample a mini-batch $\mathcal{B} \subset \{1, \dots, N\}$ of size N_{b}.
   (b) Compute the empirical loss on the batch:

$$\mathcal{L}(v_0, \theta, \boldsymbol{\beta}) = \frac{1}{N_{\mathrm{b}}} \sum_{i \in \mathcal{B}} \left(Y^i - B^i_T v_0 - \boldsymbol{\beta}^\top \sum_{k=0}^{n-1} y_\theta(x_{i,k})\, G^i_{t_{k+1}} \right)^2 + \lambda\, \boldsymbol{\beta}^\top \boldsymbol{\beta}.$$

   (c) Update parameters using a stochastic gradient descent (SGD) method:

$$v_0 \leftarrow \mathtt{SGD}(v_0, \nabla_{v_0} \mathcal{L}, \eta_e), \qquad \theta \leftarrow \mathtt{SGD}(\theta, \nabla_\theta \mathcal{L}, \eta_e), \qquad \boldsymbol{\beta} \leftarrow \mathtt{SGD}(\boldsymbol{\beta}, \nabla_\beta \mathcal{L}, \eta_e).$$

   End for

4. **Output:** Learned initial price v_0 and approximated hedging strategy

$$\widehat{\phi}_D(\cdot) = \sum_{j=1}^D \beta_j\, y^j_\theta(\cdot).$$

Proposition 3.2 (Existence for CVaR hedging)**.** *Let $\tilde{\alpha} \in (0,1)$ and $\lambda > 0$, and suppose that $v_0 \in \mathbb{R}$ is fixed and Assumption 1 holds. Then, (3.30) admits a global minimizer.*

Proof. The proof is deferred in Appendix 6.1.5. □

Using the definitions (3.7)–(3.19) of $\mathbf{Y}^{v_0}$ and $\mathbf{Z}_\theta$, we once again consider the RFF approximate formulation introduced in Section 3.3 and solve

$$\min_{\theta \in \Theta,\ \boldsymbol{\beta} \in \mathbb{R}^D,\, \zeta \in \mathbb{R}} \left\{ \zeta + \frac{1}{\tilde{\alpha} N} \sum_{i=1}^N \left(\left(\mathbf{Y}^{v_0} - \mathbf{Z}_\theta \boldsymbol{\beta} \right)_i - \zeta \right)_+ + \lambda \|\boldsymbol{\beta}\|^2 \right\}.$$

This formulation leads to Algorithm 2, which applies stochastic gradient-based optimization to this finite-dimensional problem and returns a numerical candidate solution $(\widehat{\theta}_D, \widehat{\boldsymbol{\beta}}_D, \widehat{\zeta}_D)$.

Algorithm 2 CVaR deep kernel hedging with RFF approximation

Input: Sample paths $\{S^i_{t_k}, B^i_{t_k}, X^i_{t_k}\}_{i=1,\dots,N;\ k=0,\dots,n}$, payoffs $\{Y^i\}_{i=1}^N$, fixed initial capital v_0, shift-invariant base kernel K with spectral measure p_K, regularization parameter λ, learning rate η_e, number of epochs E, batch size N_{b}, and number of RFF features D.

1. For every $i = 1, \dots, N$ and $k = 0, \dots, n-1$, set

$$x_{i,k} = X^i_{t_k}, \qquad G^i_{t_{k+1}} = B^i_T \left(\frac{S^i_{t_{k+1}}}{B^i_{t_{k+1}}} - \frac{S^i_{t_k}}{B^i_{t_k}} \right).$$

2. Initialize:

   - Neural network ψ_θ with parameters $\theta \in \Theta$.
   - Coefficients $\boldsymbol{\beta} \in \mathbb{R}^D$ and $\zeta \in \mathbb{R}$.
   - Random features $(w_j, b_j) \overset{\text{i.i.d.}}{\sim} p_K \otimes \mathcal{U}(0, 2\pi)$ for $j = 1, \dots, D$, and define

$$y_\theta(x) = \sqrt{\frac{2}{D}} \begin{bmatrix} \cos(w_1^T \psi_\theta(x) + b_1) \\ \vdots \\ \cos(w_D^T \psi_\theta(x) + b_D) \end{bmatrix}.$$

3. **For** epoch $e = 1, \dots, E$ **do**

   (a) Sample a mini-batch $\mathcal{B} \subset \{1, \dots, N\}$ of size N_{b}.

   (b) Compute the empirical loss on the batch:

$$\mathcal{L}(\theta, \boldsymbol{\beta}, \zeta) = \zeta + \frac{1}{\tilde{\alpha} N_{\mathrm{b}}} \sum_{i \in \mathcal{B}} \left(Y^i - B^i_T v_0 - \boldsymbol{\beta}^\top \sum_{k=0}^{n-1} y_\theta(x_{i,k})\, G^i_{t_{k+1}} - \zeta \right)_+ + \lambda\, \boldsymbol{\beta}^\top \boldsymbol{\beta}.$$

   (c) Update parameters using a stochastic gradient descent (SGD) method:

$$\theta \leftarrow \mathtt{SGD}(\theta, \nabla_\theta \mathcal{L}, \eta_e), \qquad \boldsymbol{\beta} \leftarrow \mathtt{SGD}(\boldsymbol{\beta}, \nabla_\beta \mathcal{L}, \eta_e), \qquad \zeta \leftarrow \mathtt{SGD}(\zeta, \nabla_\zeta \mathcal{L}, \eta_e).$$

   End for

4. **Output:** Learned hedging strategy

$$\widehat{\phi}_D(\cdot) = \sum_{j=1}^{D} \beta_j\, y^j_\theta(\cdot).$$

4 Numerical experiments

This section presents the numerical experiments used to evaluate the proposed deep kernel hedging approach. To this end, we consider both synthetic data generated in a Heston stochastic volatility model and real data from S&P500 time series. We compare our deep kernel approach with standard benchmarks, including deep hedging and classical kernel methods. We first detail the architectures of the different learning approaches, before discussing the numerical results.

Deep kernel architecture. We consider a deep kernel construction where the input features are first mapped through a shallow neural network $\psi_\omega : \mathbb{R}^d \to \mathbb{R}^p$, followed by a standard Gaussian (RBF) kernel applied in the learned representation space. The choice of the RBF kernel is motivated by its strong universality and smoothness properties[2], which make it particularly well-suited for capturing nonlinear relationships in a

[2]In particular, if $\mathcal{X} \subset \mathbb{R}^d$ is compact and $\psi_\theta : \mathcal{X} \to \mathbb{R}^p$ is continuous, then $\psi_\theta(\mathcal{X})$ is compact, and the Gaussian kernel is universal on the latent set $\psi_\theta(\mathcal{X})$ (Micchelli et al., 2006). Consequently, the deep kernel K_θ is universal on $\mathcal{X}$ if and only if ψ_θ

wide range of function classes. More specifically, the deep kernel is defined as

$$K_\theta(x,y) := \exp\left(-\frac{\|\tilde{\psi}_\omega(x)-\tilde{\psi}_\omega(y)\|^2}{2\gamma^2}\right),$$

where $\theta := (\omega, \gamma)$, with $\gamma > 0$ controlling the scale of the representation in the RBF kernel, and

$$\tilde{\psi}_\omega(\cdot) := \frac{\psi_\omega(\cdot)}{\max\{\|\psi_\omega(\cdot)\|, \epsilon_{\text{norm}}\}} \in \mathbb{S}^{p-1},$$

denoting the ℓ_2-normalized[3] output of the neural network with $\epsilon_{\text{norm}} = 10^{-6}$. By defining

$$\psi_\theta(\cdot) := \frac{1}{\gamma}\tilde{\psi}_\omega(\cdot),$$

γ can be absorbed into the learned representation, yielding the equivalent formulation

$$K_\theta(x,y) = K_{\text{RBF}}(\psi_\theta(x), \psi_\theta(y)), \tag{4.1}$$

where

$$K_{\text{RBF}}(x,y) := \exp\left(-\frac{\|x-y\|^2}{2}\right).$$

The ℓ_2 normalization removes the scale degree of freedom from the neural representation, allowing the neural network ψ_ω to focus on learning the geometry of the representation, while γ controls the scale of the RBF kernel. Moreover, the representation (4.1) shows that the RBF scaling parameter can be learned jointly with the neural network parameters rather than tuned separately. This is particularly advantageous in our setting, since selecting an appropriate scale parameter for the RBF kernel can be challenging in practice; see for instance Wainer and Fonseca (2021). Thus, by incorporating γ into the learned representation, our approach avoids the need for a separate fine-tuning step for the scale parameter (or any other hyperparameter) of the base kernel.

As detailed in Sections 3.3–3.4, our deep kernel-based hedging algorithm relies on a RFF approximation. Since we consider the standard RBF kernel as the base kernel, its spectral density is given by

$$p_{K_{\text{RBF}}}(w) = (2\pi)^{-p/2}\exp\left(-\frac{\|w\|^2}{2}\right), \qquad w \in \mathbb{R}^p,$$

and hence the random frequencies are sampled as

$$w_i \sim \mathcal{N}(0, I_p), \qquad i = 1, \dots, D.$$

Thanks to the reparameterization in (4.1), the random frequencies w_i are sampled once at the beginning of the algorithm and kept fixed throughout training. Without such a reparameterization, the random Fourier features would have to be sampled from a distribution that explicitly depends on the kernel hyperparameter γ. As a result, any update of γ during training would require resampling the random features, leading to a less stable optimization procedure due to the additional source of randomness at each iteration.

The neural network ψ_ω is composed of one hidden layer of 16 neurons with hyperbolic tangent (`Tanh`) activation function and one output linear layer of dimension 16. For the random Fourier feature approximation, we fix the number of features to $D = 100$. This provides a good trade-off between approximation accuracy and computational cost in our low-data regime, as confirmed by Figure 3 in Appendix 6.3. Note that we also standardize the input variable before it is fed into the neural network. Figure 1 highlights the chosen learning architecture for the deep kernel hedging approach with RFF approximation. The training is done in batches of size $N_{\text{b}} = 32$ for $E = 100$ epochs, with a `AdamW` optimizer, see Loshchilov and Hutter (2019), with weight decay 5e−3, a learning rate η_e ranging from 1e−3 to 1e−4 with a `cosine` scheduler over the training steps, and a ridge regularization parameter $\lambda = 1\text{e}{-6}$.

is injective on $\mathcal{X}$; that is, its associated RKHS $\mathcal{H}_{K_\theta}$ is uniformly dense in $C(\mathcal{X})$.

[3]Note that ℓ_2-normalization improves numerical stability in our experiments, even though it may destroy injectivity of ψ_θ. Similarly, we do not constrain in practice θ to lie in a compact set Θ as doing so limits the expressiveness of the neural network.

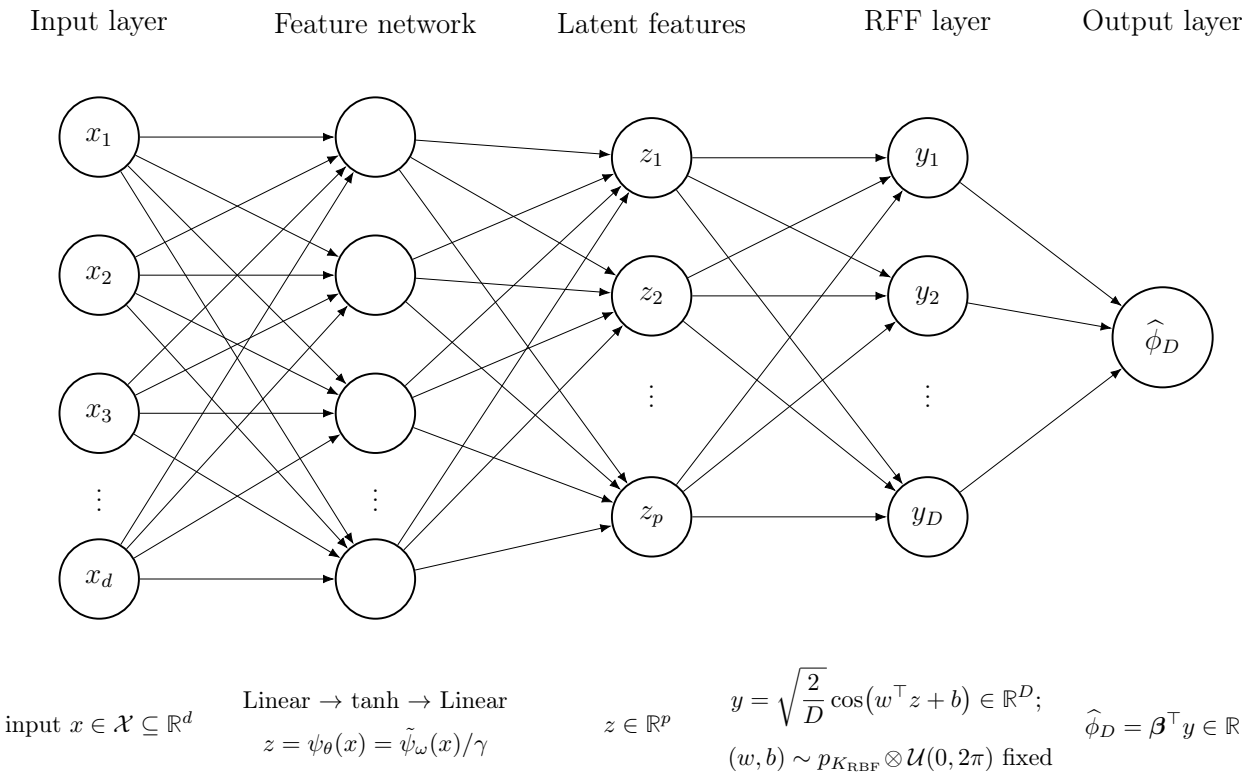


Figure 1: Deep Kernel Hedging architecture

Benchmark methods. We compare the proposed deep kernel hedging approach against several benchmark methods. To this end, we consider both a classical kernel-based hedging strategy and a standard deep hedging approach with feedforward neural network (FNN). All the following benchmarks are trained using the `AdamW` optimizer with the same hyperparameter configuration as the deep kernel hedging method. This ensures that differences in performance are due to modeling choices rather than optimization settings.

For the classical kernel method, we adopt the same overall setting as in the deep kernel approach, but remove the learned representation. In particular, we fix the feature map to the identity, $\psi_\theta(x) = x$, and use an RBF kernel with bandwidth parameter γ. As in the deep kernel model, we rely on a RFF approximation of the kernel, with $D = 100$ features. In this case, the kernel bandwidth γ is the only learned kernel parameter. This design isolates the effect of learning a nonlinear representation, while keeping the kernel structure and RFF approximation scheme identical to the deep kernel framework.

For the classical deep hedging (Buehler et al., 2019), we implement a fully connected FNN as a benchmark model, where the network directly outputs the hedging strategy ϕ. In general non-Markovian settings, one may naturally consider more sophisticated architectures such as recurrent neural networks (RNNs) or convolutional neural networks (CNNs) in order to capture temporal dependencies. However, recent works (Abi Jaber and Gérard, 2025) have shown that FNNs remain highly competitive in such settings when combined with appropriate feature engineering, in particular through time-augmented signature representations of the input features. This further supports the use of a FNN as a strong and versatile baseline across different modeling regimes. The chosen benchmark FNN consists of two hidden layers with 16 neurons each, using `Tanh` activation functions, so as to match the deep kernel hedging architecture and to have a fair numerical comparison[4].

Remark 4.1. To assess whether the observed performance gains are specific to the use of `cosine` random Fourier features (specific to kernel approximation) or instead result from the learned latent representation

[4]Under the considered architectures, the numbers of trainable parameters are $16d + 305$ for deep hedging and $16d + 389$ for deep kernel hedging, where d is the input dimension.

ψ_θ, we also investigate additional RFNN benchmarks with `ReLU` and `Tanh` activations (Huang et al., 2006), although these are not included in the main text. The corresponding results are reported by Table 7 in Appendix 6.3. These experiments show that the choice of the random-feature activation has only a minor impact on performance, with `cosine`, `ReLU`, and `Tanh` features yielding very similar results when combined with the learned representation. In contrast, removing the learned representation and using the random features directly leads to substantially weaker performance. This suggests that the main benefit of our approach comes from learning a nonlinear latent representation ψ_θ prior to the random-feature layer, rather than from the specific choice of the random-feature family. This further supports our use of `cosine` RFFs as the canonical choice throughout the paper. ■

4.1 Results on synthetic data: Heston model

First, we consider numerical experiments on synthetic data. To this end, as in Buehler et al. (2019), we consider the Heston (1993) model. Let $(\Omega, \mathcal{F}, \mathcal{F}_t, \mathbb{P})$ be a filtered probability space, where $\mathbb{P}$ stands for the real measure. The Heston (1993) model is specified by the following stochastic differential equations:

$$\begin{aligned} dS_t &= \mu S_t dt + \sqrt{\nu_t} S_t dW_t^S, \quad S_0 > 0, \\ d\nu_t &= \kappa\left(\bar{\nu} - \nu_t\right) dt + \sigma_\nu \sqrt{\nu_t} dW_t^\nu, \quad \nu_0 > 0, \end{aligned}$$

where (W^S, W^ν) are correlated Brownian motions such that $d\langle W^S, W^\nu\rangle_t = \rho dt$, $\mu, \kappa, \bar{\nu}, \sigma_\nu > 0$ and $\rho \in (-1, 1)$. In addition, the cash account process $(B_t)_{t\geq 0}$ satisfies $B_t = e^{rt}$, where $r \in \mathbb{R}$ denotes the constant risk-free interest rate. For our numerical results, we have chosen the set of parameters from (Mikkilä and Kanniainen, 2023, Table 3) given by

$$S_0 = 1, r = 0.02, \mu = 0.1, \nu_0 = 0.03, \bar{\nu} = 0.04, \kappa = 3.41, \sigma_\nu = 0.62, \rho = -0.79.$$

In this setting, we consider the quadratic and CVaR hedging of at-the-money (ATM) European and Asian call options. In the main text, we focus on the ATM European call with payoff

$$Y_T = (S_T - S_0)_+,$$

while the corresponding results and analysis for the Asian call are provided in Appendix 6.3. To this end, we simulate the process (S, ν) using a discrete Euler scheme with $\delta_t = 1/360$ that generates our training synthetic dataset consisting of N sample paths observed at different trading dates $0 = t_0 < t_1 < \cdots < t_n = T$,

$$\mathcal{D}_N = \left\{ \left(\left(S_{t_k}^i, \nu_{t_k}^i\right)_{k=0}^n, Y_T^i \right)_{i=1}^N \right\}.$$

In the Markovian setting considered here, we follow a similar approach to Buehler et al. (2019); Lütkebohmert et al. (2022); Abi Jaber and Gérard (2025) and use only the time-augmented risk factors as input features, namely[5]

$$X_{t_k}^i = (t_k, S_{t_k}^i, \nu_{t_k}^i), \qquad k = 0, \cdots, n.$$

We remark that in more general non-Markovian settings, path-signature features could be used to enrich the state representation and capture path-dependent effects (Abi Jaber and Gérard (2025); Cirone and Salvi (2025) and Section 4.2 below).

We now compare the different learning approaches for different maturities

$$T \in \{1/12, 1/4, 1/2\},$$

and different training sample sizes

$$N \in \{250, 500, 1000, 2500\}.$$

We restrict ourselves to relatively small training datasets in order to remain in a realistic financial setting. As a benchmark, we also include the classical Black and Scholes (1973) practitioner delta hedging strategy,

[5] We also experimented with path-signature features, but found no additional benefit from their use in the Markovian setting considered here.

where the hedge ratio at time t is given by the Black–Scholes delta evaluated using the current stochastic volatility $\sqrt{\nu_t}$, i.e.,

$$\Delta^{\mathrm{BS}}(S_t, K, T-t, r, \sqrt{\nu_t}).$$

For quadratic hedging, the initial price v_0 is also learned, while for CVaR hedging, we fix the initial price to the corresponding Black–Scholes price with $\sigma = \sqrt{\nu_0}$, see Section 3.4. We evaluate all methods out-of-sample on an independent test set $\mathcal{D}_{N^{\mathrm{OOS}}}$ consisting of $N^{\mathrm{OOS}} = 25{,}000$ simulated paths. To assess hedging performance, we consider several OOS metrics based on the terminal hedging error (3.3), including the mean squared error (MSE), mean absolute error (MAE), as well as tail-risk measures with the 95% quantile $q_{0.95}^{\mathrm{upper}}$ and upper-tail $\mathrm{CVaR}_{0.95}^{\mathrm{upper}}$ of the terminal hedging error with $\tilde{\alpha} = 5\%$,

$$\mathrm{MSE} = \frac{1}{N^{\mathrm{OOS}}} \sum_{i=1}^{N^{\mathrm{OOS}}} \mathcal{E}_i^2(\widehat{\phi}_D), \qquad \mathrm{MAE} = \frac{1}{N^{\mathrm{OOS}}} \sum_{i=1}^{N^{\mathrm{OOS}}} |\mathcal{E}_i(\widehat{\phi}_D)|,$$

$$q_{0.95}^{\mathrm{upper}} = q_{0.95}^{\mathrm{upper}}\big(\{\mathcal{E}_i(\widehat{\phi}_D)\}_{i=1}^{N^{\mathrm{OOS}}}\big), \quad \mathrm{CVaR}_{0.95}^{\mathrm{upper}} = \mathrm{CVaR}_{0.95}^{\mathrm{upper}}\big(\{\mathcal{E}_i(\widehat{\phi}_D)\}_{i=1}^{N^{\mathrm{OOS}}}\big).$$

Moreover, since the learning procedures involve random initialization of the neural network parameters and since the kernel-based methods additionally rely on randomly sampled random Fourier features, we repeat the experiments using different random seeds. We report the mean of each performance metric across the different seeds, together with its corresponding standard deviation.

Table 2: Out-of-sample MSE for the ATM European call for quadratic hedging. Reported values are averages over ten seeds, with standard deviations in parentheses.

$T = 1/12$

N	250	500	1000	2500
BS	2.56e−5	2.56e−5	2.56e−5	2.56e−5
Kernel hedging	4.64e−5 (2.2e−6)	4.21e−5 (1.1e−6)	3.94e−5 (7.1e−7)	3.82e−5 (3.7e−7)
Deep hedging	3.08e−5 (2.4e−6)	2.56e−5 (7.9e−7)	2.33e−5 (3.4e−7)	2.21e−5 (2.4e−7)
Deep kernel hedging	**2.44e−5 (1.2e−6)**	**2.22e−5 (4.8e−7)**	**2.14e−5 (1.5e−7)**	**2.09e−5 (5.9e−8)**

$T = 1/4$

N	250	500	1000	2500
BS	1.07e−4	1.07e−4	1.07e−4	1.07e−4
Kernel hedging	1.56e−4 (8.3e−6)	1.38e−4 (3.0e−6)	1.33e−4 (1.6e−6)	1.28e−4 (7.5e−7)
Deep hedging	1.06e−4 (5.7e−6)	8.88e−5 (3.8e−6)	7.71e−5 (2.4e−6)	7.24e−5 (1.4e−6)
Deep kernel hedging	**7.91e−5 (2.2e−6)**	**7.11e−5 (1.2e−6)**	**6.88e−5 (8.0e−7)**	**6.75e−5 (4.9e−7)**

$T = 1/2$

N	250	500	1000	2500
BS	2.40e−4	2.40e−4	2.40e−4	2.40e−4
Kernel hedging	3.21e−4 (1.3e−5)	2.97e−4 (5.9e−6)	2.87e−4 (3.7e−6)	2.65e−4 (1.7e−6)
Deep hedging	2.04e−4 (9.9e−6)	1.79e−4 (9.1e−6)	1.59e−4 (6.8e−6)	1.43e−4 (3.9e−6)
Deep kernel hedging	**1.62e−4 (3.8e−6)**	**1.44e−4 (2.0e−6)**	**1.37e−4 (2.3e−6)**	**1.31e−4 (5.8e−7)**

We first focus on quadratic hedging. Table 2 reports the out-of-sample MSE across maturities and training sample sizes. Table 3 reports additional out-of-sample metrics across maturities, with a fixed training sample size of $N = 1000$. Several patterns can be observed from the numerical experiments. First, increasing the number of training paths consistently improves performance across all learning-based methods, leading to lower out-of-sample hedging MSE. The improvement is particularly significant in the low-data regime, where additional training paths lead to substantial gains in hedging performance. Overall, deep kernel hedging achieves the lowest MSE across all considered maturities and training sample sizes, and this improvement

Table 3: Out-of-sample metrics for the ATM European call for quadratic hedging with $N = 1000$. Reported values are averages over ten seeds, with standard deviations in parentheses.

$T = 1/12$

	MSE	MAE	$q_{0.95}^{\text{upper}}$	$\text{CVaR}_{0.95}^{\text{upper}}$
BS	2.56e−5	4.01e−3	8.96e−3	1.21e−2
Kernel hedging	3.94e−5 (7.2e−7)	4.79e−3 (4.5e−5)	1.12e−2 (1.2e−4)	1.53e−2 (1.6e−4)
Deep hedging	2.33e−5 (3.4e−7)	3.66e−3 (3.6e−5)	8.28e−3 (7.0e−5)	1.16e−2 (9.2e−5)
Deep kernel hedging	**2.14e−5** (**1.5e−7**)	**3.51e−3** (**7.3e−6**)	**8.05e−3** (**9.0e−5**)	**1.13e−2** (**1.1e−4**)

$T = 1/4$

	MSE	MAE	$q_{0.95}^{\text{upper}}$	$\text{CVaR}_{0.95}^{\text{upper}}$
BS	1.07e−4	8.28e−3	1.95e−2	2.49e−2
Kernel hedging	1.33e−4 (1.6e−6)	8.58e−3 (7.7e−5)	1.99e−2 (1.3e−4)	2.78e−2 (1.7e−4)
Deep hedging	7.71e−5 (2.4e−6)	6.56e−3 (9.5e−5)	1.54e−2 (3.5e−4)	2.16e−2 (4.0e−4)
Deep kernel hedging	**6.87e−5** (**8.0e−7**)	**6.29e−3** (**2.4e−5**)	**1.46e−2** (**1.3e−4**)	**2.06e−2**(**2.5e−4**)

$T = 1/2$

	MSE	MAE	$q_{0.95}^{\text{upper}}$	$\text{CVaR}_{0.95}^{\text{upper}}$
BS	2.40e−4	1.22e−2	3.08e−2	3.98e−2
Kernel hedging	2.87e−4 (3.7e−6)	1.24e−2 (1.3e−4)	2.92e−2 (2.3e−4)	4.31e−2 (2.9e−4)
Deep hedging	1.59e−4 (6.8e−6)	9.34e−3 (1.6e−4)	2.24e−2 (4.9e−4)	3.15e−2 (7.6e−4)
Deep kernel hedging	**1.37e−4** (**2.3e−6**)	**8.68e−3** (**8.1e−5**)	**2.11e−2** (**2.3e−4**)	**3.02e−2** (**4.6e−4**)

can also be seen in the other out-of-sample metrics (MAE, $q_{0.95}^{\text{upper}}$, $\text{CVaR}_{0.95}^{\text{upper}}$) reported in Table 3. These results already highlight the importance of the learning method in the low-data regime, where the choice of method can have a substantial impact on out-of-sample performance.

Second, comparing deep kernel hedging with standard deep hedging shows that the proposed approach systematically outperforms the classical deep hedging method across all considered maturities and training sample sizes. The improvement is particularly pronounced for smaller training sets, while the performance gap progressively narrows as more training paths become available. Nevertheless, deep kernel hedging remains superior in all considered settings. This suggests that the kernel component provides a useful inductive bias that is particularly beneficial when training data are limited, while the neural network part learns a flexible representation of the market state.

Third, the comparison between classical kernel hedging and deep kernel hedging highlights the importance of learning the representation of the state variables. The classical kernel approach performs poorly and even underperforms the Black–Scholes benchmark in several settings, indicating that applying the RBF kernel directly to the original input space does not provide a suitable geometry for the hedging problem. By contrast, deep kernel hedging jointly learns a nonlinear representation of the input space and the kernel-based hedging strategy, resulting in a substantial and consistent improvement in performance. This comparison suggests that the main benefit of the proposed approach does not simply come from the use of a kernel, but rather from combining the kernel structure with a learned latent representation, as confirmed in Appendix 6.3.

Fourth, we examine the variability of the different methods across random seeds. Deep kernel hedging tends to have a lower standard deviation than the other learning-based approaches. This is particularly significant since, in addition to the random initialization of the neural network parameters, the deep kernel method involves an additional source of randomness through the sampling of the random Fourier features. Different realizations can lead to different feature representations and consequently to greater variability in the resulting hedging strategies. In contrast, the learned representation is adapted during training to the fixed

random-feature map, which may mitigate part of the variability induced by different RFF realizations. The lower dispersion observed across seeds therefore suggests that the learned representation not only improves hedging performance, but also contributes to making the method more robust across different training runs.

Table 4: Out-of-sample metrics for the ATM European call for CVaR hedging with $N = 1000$ and $v_0 = \mathrm{BS}(S_0, K, T, r, \sigma = \sqrt{\nu_0})$. Reported values are averages over ten seeds, with standard deviations in parentheses.

$T = 1/12$

	MSE	MAE	$q^{\text{upper}}_{0.95}$	$\text{CVaR}^{\text{upper}}_{0.95}$
BS	**2.56e−5**	**4.01e−3**	8.96e−3	1.21e−2
Kernel hedging	5.02e−5 (1.2e−6)	5.56e−3 (8.1e−5)	1.19e−2 (1.1e−4)	1.56e−2 (1.1e−4)
Deep hedging	3.12e−5 (1.2e−6)	4.47e−3 (1.1e−4)	8.81e−3 (1.8e−4)	1.16e−2 (1.6e−4)
Deep kernel hedging	2.93e−5 (1.2e−6)	4.28e−3 (1.1e−4)	**8.47e−3** (**7.2e−5**)	**1.13e−2** (**1.2e−4**)

$T = 1/4$

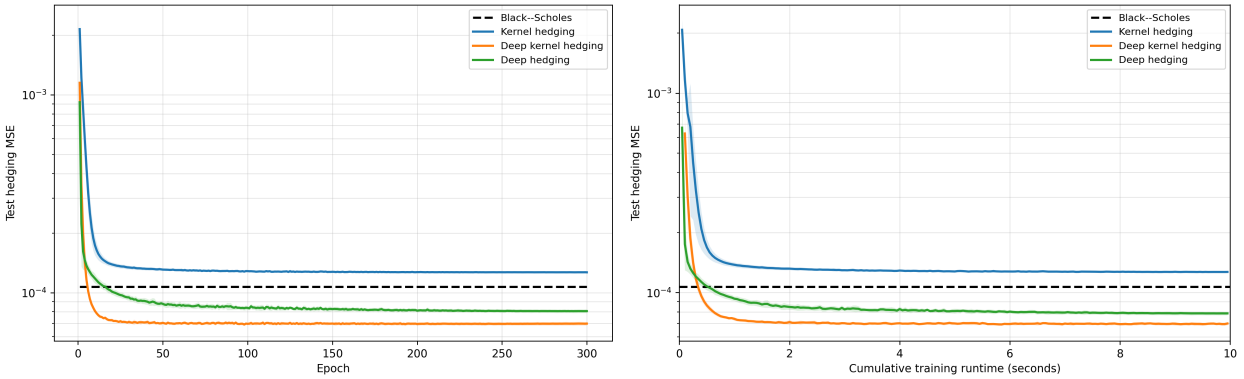
	MSE	MAE	$q^{\text{upper}}_{0.95}$	$\text{CVaR}^{\text{upper}}_{0.95}$
BS	1.07e−4	8.28e−3	1.95e−2	2.49e−2
Kernel hedging	1.63e−4 (6.9e−6)	9.93e−3 (3.3e−4)	2.28e−2 (1.6e−4)	3.06e−2 (1.7e−4)
Deep hedging	1.06e−4 (2.6e−6)	8.25e−3 (1.0e−4)	1.73e−2 (3.6e−4)	2.30e−2 (4.0e−4)
Deep kernel hedging	**9.92e−5** (**4.0e−6**)	**8.02e−3** (**2.2e−4**)	**1.68e−2** (**1.3e−4**)	**2.21e−2** (**2.5e−4**)

$T = 1/2$

	MSE	MAE	$q^{\text{upper}}_{0.95}$	$\text{CVaR}^{\text{upper}}_{0.95}$
BS	2.40e−4	1.22e−2	3.08e−2	3.98e−2
Kernel hedging	3.62e−4 (1.7e−5)	1.49e−2 (5.5e−4)	3.56e−2 (2.5e−4)	4.65e−2 (2.4e−4)
Deep hedging	2.41e−4 (7.0e−6)	1.26e−2 (2.4e−4)	2.80e−2 (4.2e−4)	3.60e−2 (6.2e−4)
Deep kernel hedging	**2.10e−4** (**7.4e−6**)	**1.15e−2** (**3.4e−4**)	**2.67e−2** (**1.6e−4**)	**3.47e−2** (**3.4e−4**)

All the above results are confirmed by Figure 2, which compares the convergence of the three learned hedging methods in terms of the out-of-sample hedging MSE. The left plot reports the MSE as a function of the training epoch, whereas the right plot reports it as a function of cumulative training time. Each curve shows the mean over 10 random seeds, with shaded regions representing ± one standard deviation. The Black–Scholes MSE is shown as a horizontal benchmark as it does not require training.

Finally, to assess hedging performance under a tail-risk objective, we also consider the CVaR hedging problem (3.30) with $\tilde{\alpha} = 5\%$ for ATM European call options. Table 4 reports the corresponding out-of-sample results for $N = 1000$. Overall, the conclusions are consistent with those obtained under quadratic hedging: deep kernel hedging systematically outperforms the other learning-based approaches in terms of CVaR, while also achieving the best (or near-best) performance according to the other reported metrics. This again demonstrates the benefit of the learned representation, also when the hedging strategy is directly optimized for tail risk. Moreover, the deep kernel approach generally exhibits a lower standard deviation across random seeds, indicating a stable learning procedure despite the additional randomness introduced by the RFF approximation. Note that a direct comparison of the absolute CVaR values between quadratic and CVaR hedging does not make sense. In the CVaR setting, the initial portfolio value v_0 is fixed to the Black–Scholes price, whereas it is learned under quadratic hedging. Therefore, the absolute CVaR values obtained under the two hedging objectives are not directly comparable as they correspond to different initial portfolio values. Overall, these results illustrate the flexibility of our framework: the loss function can be selected according to the performance criterion of interest, allowing the learned strategy to prioritize either average hedging accuracy or protection against extreme P&L downside risk.



(a) MSE on test set versus training epoch.

(b) MSE on test set versus cumulative runtime.

Figure 2: Hedging MSE $\sum_{i=1}^{N^{\mathrm{OOS}}} \mathcal{E}_i^2(\widehat{\phi}_D)/N^{\mathrm{OOS}}$ during training on a test set of $N^{\mathrm{OOS}} = 25,000$ synthetic paths of an ATM European call with $T = 1/4$ and $N = 1000$. The curves and shaded regions show the mean and one standard deviation, respectively, over 10 seeds.

Overall, within the considered Heston framework and for the neural architectures selected in this study, the numerical results indicate that the deep kernel approach offers a competitive and robust alternative to standard deep hedging by effectively combining the advantages of both deep and kernel methodologies. Naturally, these conclusions remain conditional on the specific experimental setting under consideration, and different market dynamics or architectures may lead to different relative performances.

4.2 Results on real data: S&P500 time series

Finally, we conclude this paper by focusing on real data of S&P500 time series. We construct our dataset using S&P500 historical data over the period from January 1st, 2010 to May 31st, 2026. The data from January 1st, 2010 to December 31st, 2022 is used for training, while the period from January 1st, 2023 to May 31st, 2026 is reserved for out-of-sample evaluation. For a fixed maturity T_{days} (in days in this section), we extract overlapping trajectories using a sliding-window procedure (rolling windows with overlap), which yields a collection of market paths observed at discrete trading dates $0 = t_0 < t_1 < \cdots < t_n = T_{\mathrm{days}}$. In addition to the spot price process S_t, we compute a realized volatility proxy $\sqrt{\nu_t}$ using a one-month (20 trading days) rolling window, and fix the interest rate r to zero. The resulting dataset is therefore given by

$$\mathcal{D}_{N_T} = \left\{ \left((S^i_{t_k}/S^i_{t_0}, \nu^i_{t_k})_{k=0}^{n}, \, Y^i_T \right) \right\}_{i=1}^{N_T},$$

where N_T denotes the number of overlapping trajectories associated with maturity T_{days}. For the training set, we have $N_T = 3262 - T_{\mathrm{days}}$ sample paths and for the out-of-sample evaluation $N_T^{\mathrm{OOS}} = 834 - T_{\mathrm{days}}$.

Remark 4.2. An alternative approach to increase the amount of training data is to first fit a stochastic or generative model (e.g., Heston, GARCH, or GAN) to the historical data and then use the fitted model to generate additional sample paths. This would effectively bring the problem back to the synthetic-data setting considered above. However, both our experiments and the empirical deep-hedging results reported in (Mikkilä and Kanniainen, 2023, Table 6) suggest that this approach performs worse than training directly on the observed data. ■

As model inputs, we consider two different feature specifications to assess the impact of path-dependent information on hedging performance. In the first, baseline specification, we use the classical time-augmented state variables

$$X_{t_k} = (t_k, S_{t_k}/S_{t_0}, \nu_{t_k}), \quad k = 0, \ldots, n, \qquad \text{(Vanilla features)}$$

which provide a standard Markovian representation of the market state. However, we find that the resulting hedging performance is relatively poor across the considered methods. We therefore report these baseline results in Appendix 6.4 and focus in the main text on a richer feature representation based on time-augmented

path-signatures. More precisely, we consider the time-augmented signature of order $M = 3$ of the path $Z := (S/S_{t_0}, \nu)$ and define

$$X_{t_k} = \hat{\mathbb{Z}}^{\leq 3}_{t_k}, \quad k = 0, \dots, n, \qquad \text{(Signature features)}$$

where $\hat{\mathbb{Z}}$ denotes the time-augmented signature of Z, see Definition 6.1. As discussed in Appendix 6.2, this provides a richer representation of the observed market dynamics and allows the learning methods to exploit path-dependent information beyond the instantaneous state variables without imposing a specific parametric structure on the underlying market dynamics. Numerically, these signature features are computed using the Python library `iisignature`. Since path-signatures depend only on the increments of a path, they are invariant under translations and therefore do not capture its absolute level. This can be problematic in our empirical application, where the data streams have different initial conditions. To address this issue, we use the base-point augmentation introduced in (Chevyrev and Kormilitzin, 2025, Section 3.1.4), which consists in appending the origin 0 to the beginning of each data stream and thereby removes translation invariance. Throughout the paper, signatures are hence understood to be computed after this base-point augmentation.

In this real (scarce) dataset setting, we consider the quadratic and CVaR hedging of ATM European and Asian call options for different maturities,

$$T_{\text{days}} \in \{10 \text{ days}, \ 20 \text{ days}, \ 40 \text{ days}\}.$$

As in the experiments on synthetic data, we focus on ATM European calls in the main text, while the corresponding results for the Asian payoff are reported in Appendix 6.4.

Moreover, following Buehler et al. (2019), we do not learn the initial price v_0, even for the quadratic hedging problem. Instead, for each trajectory, we fix

$$v_0^i = \mathrm{BS}\big(S_0 = S_0^i, K = S_0^i, T, \sigma = \sqrt{\nu_0^i}\big),$$

where ν_0^i denotes the initial realized variance of the i-th path and $\mathrm{BS}(S, K, T, \sigma)$ refers to the Black-Scholes price of a European call option. This choice is motivated by the fact that real market trajectories do not start from a common initial market state. As a consequence, it is natural that the option price at time zero depends on the prevailing market conditions, and in particular on the initial volatility level. In contrast to the synthetic data setting where v_0 can be treated as a single parameter to learn, we therefore fix here v_0^i pathwise and do not learn it from data. More generally, one could consider learning the pricing functional as a function of the initial market state (in particular ν_0^i), but our focus in this paper is mainly on learning the hedging strategy rather than the optimal pricing rule. Finally, we benchmark our learned strategies against the classical Black-Scholes practitioner delta, where the hedge is computed using the Black-Scholes delta evaluated with the realized volatility.

We keep the same architectures as in the synthetic data experiments for all learning-based methods, but use a smaller and constant learning rate of 1e−4, to mitigate overfitting in the noisier real-market setting. Unlike the synthetic experiments, where all simulated paths start from the same initial spot price S_0, the real-data paths are associated with different initial spot prices S_0^i. We therefore normalize the terminal hedging error by the corresponding initial spot price both during training and evaluation, i.e. we use $\mathcal{E}_i^{\text{norm}}(\phi) := \mathcal{E}_i(\phi; v_0^i)/S_0^i$. Specifically, the hedging strategies are trained by minimizing losses defined on the normalized hedging error $\mathcal{E}_i^{\text{norm}}(\phi)$. Similarly, all out-of-sample terminal hedging errors are normalized by their respective initial spot prices before computing the reported performance metrics, making comparisons across periods with different market levels meaningful.

Table 5 reports the out-of-sample performance of the different learning-based methods for the quadratic hedging problem using the time-augmented signature features across different maturities. For the three maturities considered, deep kernel hedging achieves the best overall hedging performance, yielding the lowest MSE and MAE, but also tail risk measures $q_{0.95}^{\text{upper}}$ and $\mathrm{CVaR}_{0.95}^{\text{upper}}$ (which are not directly minimized during training). The aforementioned complementary roles of the kernel and neural network components are therefore particularly beneficial in the real-data setting considered here, where the available training samples are limited and the rolling-window trajectories are highly correlated. Moreover, as in the synthetic-data experiments, deep

Table 5: Out-of-sample metrics for the quadratic hedging of an ATM European call on real dataset with time-augmented signature features. P&Ls are scaled by the underlying price at the beginning of the periods. Reported values are averages over ten seeds, with standard deviations in parentheses.

$T_{\text{days}} = 10$ days.

	MSE	MAE	$q_{0.95}^{\text{upper}}$	$\text{CVaR}_{0.95}^{\text{upper}}$
BS	3.12e−5	3.48e−3	9.13e−3	1.41e−2
Kernel hedging	2.64e−5 (1.5e−6)	3.83e−3 (1.3e−4)	8.61e−3 (4.2e−4)	1.19e−2 (3.8e−4)
Deep hedging	2.24e−5 (2.4e−6)	3.51e−3 (1.2e−4)	7.17e−3 (1.7e−4)	1.02e−2 (6.1e−4)
Deep kernel hedging	**2.06e−5 (1.2e−6)**	**3.33e−3 (4.9e−5)**	**6.76e−3 (2.6e−4)**	**1.00e−2 (4.5e−4)**

$T_{\text{days}} = 20$ days.

	MSE	MAE	$q_{0.95}^{\text{upper}}$	$\text{CVaR}_{0.95}^{\text{upper}}$
BS	5.64e−5	4.83e−3	1.17e−2	1.73e−2
Kernel hedging	4.50e−5 (4.4e−6)	4.81e−3 (2.2e−4)	1.12e−2 (4.8e−4)	1.71e−2 (9.7e−4)
Deep hedging	3.46e−5 (4.3e−6)	4.18e−3 (1.4e−4)	8.63e−3 (3.7e−4)	1.20e−2 (8.8e−4)
Deep kernel hedging	**3.13e−5 (2.8e−6)**	**3.90e−3 (1.2e−4)**	**8.34e−3 (3.7e−4)**	**1.13e−2 (6.8e−4)**

$T_{\text{days}} = 40$ days.

	MSE	MAE	$q_{0.95}^{\text{upper}}$	$\text{CVaR}_{0.95}^{\text{upper}}$
BS	9.75e−5	6.23e−3	1.36e−2	1.76e−2
Kernel hedging	7.29e−5 (1.2e−5)	6.20e−3 (3.6e−4)	1.62e−2 (2.0e−3)	2.30e−2 (3.7e−3)
Deep hedging	4.35e−5 (5.3e−6)	4.75e−3 (1.9e−4)	1.08e−2 (9.3e−4)	1.64e−2 (2.3e−3)
Deep kernel hedging	**3.55e−5 (4.2e−6)**	**4.08e−3 (1.9e−4)**	**8.74e−3 (1.0e−3)**	**1.36e−2 (2.2e−3)**

kernel hedging exhibits greater stability across random seeds, as reflected by the lower standard deviations observed across most of the reported metrics.

Finally, we consider CVaR hedging, with the out-of-sample results reported in Table 6. Compared with Table 5, the CVaR-trained strategies exhibit higher MSE and MAE values, as they are no longer trained to minimize the quadratic hedging error. However, the upper-tail metrics $\text{CVaR}_{0.95}^{\text{upper}}$ are substantially improved[6]. This is consistent with the CVaR training objective, which explicitly penalizes the severity of adverse hedging outcomes in the worst 5% of the P&L distribution. We note, however, that deep kernel hedging does not systematically outperform deep hedging in terms of CVaR in this empirical setting. These results should however be interpreted with some caution since the out-of-sample sample size is only $N_T^{\text{OOS}} = 834 - T_{\text{days}}$. Consequently, the empirical out-of-sample CVaR is estimated from only approximately 40 tail observations (since $\alpha = 5\%$), making this tail-risk metric subject to substantial sampling variability.

Overall, these empirical results on real data further support deep kernel hedging as a robust and competitive alternative to standard deep hedging. In particular, the proposed approach consistently achieves strong hedging performance across different maturities. Taken together, these results suggest that the combination of a learned representation with the structured kernel-based hedging framework provides an effective and flexible approach for learning hedging strategies from real data.

5 Conclusion

In this paper, we introduce a deep kernel hedging framework that combines the flexibility of deep learning with the structural inductive bias of kernel methods. The hedging functional is restricted in a reproducing kernel Hilbert space whose kernel is parameterized through a neural network embedding of the input

[6]Since the methods are trained with the same initial prices, the comparison of the metrics between the hedging strategies optimizing quadratic or CVaR criteria is fair here, unlike in the synthetic-data setting.

Table 6: Out-of-sample metrics for the CVaR hedging of an ATM European call on real dataset with time-augmented signature features. P&Ls are scaled by the underlying price at the beginning of the periods. Reported values are averages over ten seeds, with standard deviations in parentheses.

$T_{\text{days}} = 10$ days.

	MSE	MAE	$q_{0.95}^{\text{upper}}$	$\text{CVaR}_{0.95}^{\text{upper}}$
BS	3.12e−5	**3.48e−3**	9.13e−3	1.42e−2
Kernel hedging	3.04e−5 (2.3e−6)	4.14e−3 (1.7e−4)	8.90e−3 (3.6e−4)	1.20e−2 (5.2e−4)
Deep hedging	2.84e−5 (2.4e−6)	4.02e−3 (1.0e−4)	7.84e−3 (2.9e−4)	1.00e−2 (7.9e−4)
Deep kernel hedging	**2.81e−5 (1.3e−6)**	4.03e−3 (6.8e−5)	**7.01e−3 (2.4e−4)**	**9.19e−3 (4.8e−4)**

$T_{\text{days}} = 20$ days.

	MSE	MAE	$q_{0.95}^{\text{upper}}$	$\text{CVaR}_{0.95}^{\text{upper}}$
BS	5.64e−5	4.83e−3	1.17e−2	1.73e−2
Kernel hedging	5.22e−5 (6.4e−6)	5.38e−3 (3.8e−4)	1.13e−2 (6.6e−4)	1.63e−2 (9.7e−4)
Deep hedging	4.98e−5 (6.0e−6)	4.91e−3 (1.4e−4)	**8.30e−3 (2.3e−4)**	**1.04e−2 (1.0e−3)**
Deep kernel hedging	**4.09e−5 (4.2e−6)**	**4.68e−3 (2.1e−4)**	8.42e−3 (2.3e−4)	1.08e−2 (4.7e−4)

$T_{\text{days}} = 40$ days.

	MSE	MAE	$q_{0.95}^{\text{upper}}$	$\text{CVaR}_{0.95}^{\text{upper}}$
BS	9.75e−5	6.23e−3	1.36e−2	1.76e−2
Kernel hedging	1.08e−4 (1.7e−5)	8.04e−3 (7.0e−4)	1.74e−2 (1.2e−3)	2.42e−2 (2.8e−3)
Deep hedging	1.53e−4 (3.4e−5)	7.36e−3 (3.3e−4)	9.10e−3 (7.4e−4)	**1.10e−2 (1.1e−3)**
Deep kernel hedging	**7.16e−5 (9.1e−6)**	**5.92e−3 (2.1e−4)**	**8.65e−3 (1.4e−3)**	1.15e−2 (1.8e−3)

features. By means of a generalized representer theorem, we derive a finite-dimensional formulation of the original infinite-dimensional hedging problem and propose a random Fourier feature approximation with convergence guarantees that overcomes the computational limitations associated with large kernel matrices.

The numerical results on both synthetic and real-world data support the benefits of this approach. In particular, the proposed deep kernel method consistently achieves competitive (and in most settings superior) hedging performance compared with both classical kernel methods and standard deep hedging architectures. The advantage over standard deep hedging is especially relevant in low-data regimes, where the inductive bias inherited from the kernel appears to be particularly beneficial. Moreover, the proposed approach exhibits greater numerical stability with a reduced sensitivity to initialization and feature randomization across different seeds.

The comparison with classical kernel methods and RFNNs suggests that the performance gains are not specific to the Fourier representation. Our numerical experiments indeed show that replacing the random Fourier features used for approximating the underlying base kernel by alternative random feature constructions leads to similar hedging performance. This confirms that the main benefit of the deep kernel approach lies in learning a data-dependent neural representation of the input space that is adapted to the hedging objective, while the specific choice of base kernel and its associated random feature map appears to play a minor role.

Future work could extend the framework to multiple tradable assets and examine transaction costs and market impact, including their implications for convexity and computational tractability. Finally, when the learning dataset contains paths with different initial conditions, one could extend the framework to learn not only the hedging functional but also the initial pricing functional as a function of the initial market state. This would allow the learned model to jointly capture the dependence of the optimal hedging strategy and the corresponding optimal initial price on the observed market conditions.

6 Appendix

6.1 Proofs

6.1.1 Proof of Lemma 3.2

Proof. For each $\theta \in \Theta$, let $\mathbf{A}_\theta := \mathbf{Q}_\theta^{1/2}$ denote the unique positive-semidefinite square root of $\mathbf{Q}_\theta$. For fixed θ, introduce the variable

$$\mathbf{c} = \mathbf{A}_\theta \boldsymbol{\alpha}.$$

Since $\mathbf{Q}_\theta = \mathbf{A}_\theta^2$, we have

$$\mathbf{Q}_\theta \boldsymbol{\alpha} = \mathbf{A}_\theta \mathbf{c} \qquad \text{and} \qquad \boldsymbol{\alpha}^\top \mathbf{Q}_\theta \boldsymbol{\alpha} = \|\mathbf{A}_\theta \boldsymbol{\alpha}\|_2^2 = \|\mathbf{c}\|_2^2.$$

As $\boldsymbol{\alpha}$ ranges over $\mathbb{R}^N$, the vector $\mathbf{c}$ ranges over $\mathrm{range}(\mathbf{A}_\theta)$. Hence, for every fixed θ, from Theorem 3.1, the inner problem in (3.11) is equivalent to

$$\min_{\mathbf{c} \in \mathrm{range}(\mathbf{A}_\theta)} \left\{ \frac{1}{N} \sum_{i=1}^N \mathcal{L}\left((\mathbf{Y}^{v_0} - \mathbf{A}_\theta \mathbf{c})_i \right) + \lambda \|\mathbf{c}\|_2^2 \right\}.$$

Since $\mathbf{A}_\theta$ is symmetric, $\mathbb{R}^N = \mathrm{range}(\mathbf{A}_\theta) \oplus \ker(\mathbf{A}_\theta)$. Thus, every $\mathbf{c} \in \mathbb{R}^N$ admits a unique decomposition

$$\mathbf{c} = \mathbf{c}_\parallel + \mathbf{c}_\perp,$$

where $\mathbf{c}_\parallel \in \mathrm{range}(\mathbf{A}_\theta)$ and $\mathbf{c}_\perp \in \ker(\mathbf{A}_\theta)$. This decomposition satisfies

$$\mathbf{A}_\theta \mathbf{c} = \mathbf{A}_\theta \mathbf{c}_\parallel \qquad \text{and} \qquad \|\mathbf{c}\|_2^2 = \|\mathbf{c}_\parallel\|_2^2 + \|\mathbf{c}_\perp\|_2^2.$$

Therefore, replacing $\mathbf{c}$ by $\mathbf{c}_\parallel$ leaves the empirical loss unchanged and weakly decreases the regularization term, with a strict decrease whenever $\mathbf{c}_\perp \neq 0$. Consequently, every minimizer of the unconstrained problem belongs to $\mathrm{range}(\mathbf{A}_\theta)$. It follows that the joint problem (3.11) is equivalent to

$$\inf_{\theta \in \Theta,\, \mathbf{c} \in \mathbb{R}^N} \left\{ \frac{1}{N} \sum_{i=1}^N \mathcal{L}\left((\mathbf{Y}^{v_0} - \mathbf{A}_\theta \mathbf{c})_i \right) + \lambda \|\mathbf{c}\|_2^2 \right\}. \tag{6.1}$$

We now establish the existence of a minimizer of (6.1). By (H2) in Assumption 1, and the definition of $\mathbf{Q}_\theta$, the mapping $\theta \mapsto \mathbf{Q}_\theta$ is continuous. Since the positive-semidefinite matrix square-root map is continuous, the mapping $\theta \mapsto \mathbf{A}_\theta$ is also continuous. Together with the continuity of $\mathcal{L}$, this implies that the objective in (6.1) is continuous in $(\theta, \mathbf{c})$. Moreover, since $\mathcal{L}$ is bounded from below, there exists $C_{\mathcal{L}} \in \mathbb{R}$ such that $\mathcal{L} \geq C_{\mathcal{L}}$. Hence,

$$\frac{1}{N} \sum_{i=1}^N \mathcal{L}\left((\mathbf{Y}^{v_0} - \mathbf{A}_\theta \mathbf{c})_i \right) + \lambda \|\mathbf{c}\|_2^2 \geq C_{\mathcal{L}} + \lambda \|\mathbf{c}\|_2^2.$$

Define

$$C_0 := \frac{1}{N} \sum_{i=1}^N \mathcal{L}\left(\mathbf{Y}_i^{v_0} \right) \geq C_{\mathcal{L}}.$$

The value of the objective at $\mathbf{c} = 0$ is $C_0 < \infty$, independently of θ. Therefore, it is sufficient to minimize over the sublevel set on which the objective is at most C_0. Every point in this sublevel set satisfies

$$\|\mathbf{c}\|_2 \leq \sqrt{\frac{C_0 - C_{\mathcal{L}}}{\lambda}}.$$

Thus, as (H1) in Assumption 1 holds, the optimization can be restricted to the compact set

$$\Theta \times \overline{B}_{\mathbb{R}^N}\left(0, \sqrt{\frac{C_0 - C_{\mathcal{L}}}{\lambda}} \right).$$

The Weierstrass theorem therefore yields a minimizer $(\theta^\star, \mathbf{c}_{\theta^\star}^\star)$ of (6.1). By the preceding orthogonal-decomposition argument, $\mathbf{c}_{\theta^\star}^\star \in \mathrm{range}(\mathbf{A}_{\theta^\star})$, and there exists $\boldsymbol{\alpha}_{\theta^\star}^\star \in \mathbb{R}^N$ such that

$$\mathbf{c}_{\theta^\star}^\star = \mathbf{A}_{\theta^\star} \boldsymbol{\alpha}_{\theta^\star}^\star.$$

Hence, $(\theta^\star, \boldsymbol{\alpha}_{\theta^\star}^\star)$ is a minimizer of (3.11). □

6.1.2 Proof of Lemma 3.3

Proof. For fixed $\theta \in \Theta$, define

$$F_\theta(\boldsymbol{\beta}) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}\left(Y_i - (\mathbf{Z}_\theta \boldsymbol{\beta})_i\right) + \lambda \|\boldsymbol{\beta}\|_2^2, \qquad \boldsymbol{\beta} \in \mathbb{R}^D.$$

Because $\mathcal{L}$ is continuous, F_θ is continuous. Moreover, since $\mathcal{L}$ is bounded below by a constant $C_{\mathcal{L}} \in \mathbb{R}$, and $\lambda > 0$,

$$F_\theta(\boldsymbol{\beta}) \geq C_{\mathcal{L}} + \lambda \|\boldsymbol{\beta}\|_2^2,$$

so F_θ is coercive and hence admits a minimizer. Furthermore, the loss term is convex in $\boldsymbol{\beta}$, while $\lambda\|\boldsymbol{\beta}\|_2^2$ is strictly convex. Thus, F_θ is strictly convex and its minimizer $\boldsymbol{\beta}_\theta^\star \in \mathbb{R}^D$ is unique. Moreover, the left-hand side of (3.21) is equal to

$$\min_{\boldsymbol{\beta} \in \mathrm{range}(\mathbf{Z}_\theta^\top)} F_\theta(\boldsymbol{\beta}).$$

We now use the orthogonal decomposition $\mathbb{R}^D = \mathrm{range}(\mathbf{Z}_\theta^\top) \oplus \ker(\mathbf{Z}_\theta)$. Thus, every $\boldsymbol{\beta} \in \mathbb{R}^D$ can be written uniquely as

$$\boldsymbol{\beta} = \boldsymbol{\beta}_\parallel + \boldsymbol{\beta}_\perp, \qquad \boldsymbol{\beta}_\parallel \in \mathrm{range}(\mathbf{Z}_\theta^\top), \quad \boldsymbol{\beta}_\perp \in \ker(\mathbf{Z}_\theta).$$

Since $\mathbf{Z}_\theta \beta = \mathbf{Z}_\theta \beta_\parallel$ and the decomposition is orthogonal, we obtain

$$F_\theta(\boldsymbol{\beta}) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}\left(Y_i - (\mathbf{Z}_\theta \boldsymbol{\beta}_\parallel)_i\right) + \lambda\|\boldsymbol{\beta}_\parallel\|_2^2 + \lambda\|\boldsymbol{\beta}_\perp\|_2^2 = F_\theta(\boldsymbol{\beta}_\parallel) + \lambda\|\boldsymbol{\beta}_\perp\|_2^2, \qquad \boldsymbol{\beta} \in \mathbb{R}^D.$$

Consequently, $F_\theta(\boldsymbol{\beta}) \geq F_\theta(\boldsymbol{\beta}_\parallel)$, with strict inequality whenever $\boldsymbol{\beta}_\perp \neq 0$. It follows that

$$\boldsymbol{\beta}_\theta^\star \in \mathrm{range}(\mathbf{Z}_\theta^\top).$$

In particular, there exists $\boldsymbol{\alpha}^\star \in \mathbb{R}^N$ such that $\mathbf{Z}_\theta^\top \boldsymbol{\alpha}^\star = \boldsymbol{\beta}_\theta^\star$, and

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^N} F_\theta(\mathbf{Z}_\theta^\top \boldsymbol{\alpha}) = \min_{\boldsymbol{\beta} \in \mathrm{range}(\mathbf{Z}_\theta^\top)} F_\theta(\boldsymbol{\beta}) = \min_{\boldsymbol{\beta} \in \mathbb{R}^D} F_\theta(\boldsymbol{\beta}).$$

This proves (3.21). Moreover, applying Lemma 3.1 and Theorem 3.1 to the kernel $\widetilde{K}_{\theta,D}$ show that $\phi_{D,\theta}^\star$ defined by (3.22) is the unique minimizer of the inner RFF problem. □

6.1.3 Proof of Proposition 3.1

Proof. Since $\Theta \times \mathcal{X}$ is compact and $(\theta, x) \mapsto \psi_\theta(x)$ is jointly continuous, the joint latent set

$$\mathcal{Z} := \{\psi_\theta(x) : \theta \in \Theta,\ x \in \mathcal{X}\} \subset \mathbb{R}^p$$

is compact. Define the latent-space RFF approximation by

$$\widetilde{K}_D(z, z') := \frac{2}{D}\sum_{j=1}^{D} \cos(w_j^\top z + b_j)\cos(w_j^\top z' + b_j), \quad z, z' \in \mathcal{Z}.$$

By construction,

$$\widetilde{K}_{\theta,D}(x, x') = \widetilde{K}_D\big(\psi_\theta(x), \psi_\theta(x')\big), \quad x, x' \in \mathcal{X},$$

and hence

$$\sup_{\theta \in \Theta}\sup_{x,x' \in \mathcal{X}} \left|\widetilde{K}_{\theta,D}(x, x') - K_\theta(x, x')\right| \leq \sup_{z,z' \in \mathcal{Z}} \left|\widetilde{K}_D(z, z') - K(z, z')\right|,$$

Standard uniform RFF approximation results (Claim 1 of Rahimi and Recht (2007) and Proposition 2 of Sutherland and Schneider (2015)) then give

$$\mathbb{P}\left(\sup_{z,z' \in \mathcal{Z}} \left|\widetilde{K}_D(z, z') - K(z, z')\right| > \varepsilon\right) \leq C\varepsilon^{-2}\exp(-cD\varepsilon^2),$$

which proves (3.24) for $\varepsilon \in (0,1]$. For every fixed $\varepsilon \in (0,1]$, applying the Borel–Cantelli lemma for $\varepsilon = 1/q$, $q \in \mathbb{N}$, directly proves (3.25). By the definitions of $\mathbf{Q}_\theta$ and $\widetilde{\mathbf{Q}}_{\theta,D}$, we then have

$$\begin{aligned}\left|(\widetilde{\mathbf{Q}}_{\theta,D} - \mathbf{Q}_\theta)_{ij}\right| &\le \sum_{k=0}^{n-1} |G^i_{t_{k+1}}| \sum_{k=0}^{n-1} |G^j_{t_{k+1}}| \sup_{\theta\in\Theta} \sup_{x,x'\in\mathcal{X}} \left|\widetilde{K}_{\theta,D}(x,x') - K_\theta(x,x')\right| \\ &= a_i a_j \sup_{\theta\in\Theta} \sup_{x,x'\in\mathcal{X}} \left|\widetilde{K}_{\theta,D}(x,x') - K_\theta(x,x')\right|,\end{aligned}$$

where $a_i := \sum_{k=0}^{n-1} |G^i_{t_{k+1}}|$. Consequently, for any unit vectors $u, v \in \mathbb{R}^N$,

$$\begin{aligned}\left|u^\top(\widetilde{\mathbf{Q}}_{\theta,D} - \mathbf{Q}_\theta)v\right| &\le \sup_{\theta\in\Theta} \sup_{x,x'\in\mathcal{X}} \left|\widetilde{K}_{\theta,D}(x,x') - K_\theta(x,x')\right| \left(\sum_{i=1}^N a_i|u_i|\right)\left(\sum_{j=1}^N a_j|v_j|\right) \\ &\le \left(\sum_{i=1}^N a_i^2\right) \sup_{\theta\in\Theta} \sup_{x,x'\in\mathcal{X}} \left|\widetilde{K}_{\theta,D}(x,x') - K_\theta(x,x')\right|,\end{aligned}$$

where the last inequality follows from Cauchy–Schwarz. Taking the supremum over u, v and θ proves (3.26). □

6.1.4 Proof of Theorem 3.4

Proof. We work on the probability-one event on which the uniform RFF convergence in Proposition 3.1 holds.
Step 1: convergence of $\mathcal{V}_D$. First, let us show the convergence of $(\mathcal{V}_D)_{D>1}$. Let

$$\mathbf{A}_\theta := \mathbf{Q}_\theta^{1/2}, \qquad \widetilde{\mathbf{A}}_{\theta,D} := \widetilde{\mathbf{Q}}_{\theta,D}^{1/2}.$$

By Proposition 3.1 and continuity of the positive-semidefinite square-root map,

$$\eta_D := \sup_{\theta\in\Theta} \left\|\widetilde{\mathbf{A}}_{\theta,D} - \mathbf{A}_\theta\right\|_{\mathrm{op}} \to 0, \quad \text{as } D \to \infty. \tag{6.2}$$

Then, using the square-root reformulation from the proof of Lemma 3.2, the exact and approximate optimal values can be written as

$$\mathcal{V} = \min_{\theta\in\Theta,\, \mathbf{c}\in\mathbb{R}^N} F(\theta,\mathbf{c}), \qquad \mathcal{V}_D = \min_{\theta\in\Theta,\, \mathbf{c}\in\mathbb{R}^N} F_D(\theta,\mathbf{c}),$$

where

$$F(\theta,\mathbf{c}) = \frac{1}{N}\sum_{i=1}^N \mathcal{L}\left((\mathbf{Y}^{v_0} - \mathbf{A}_\theta\mathbf{c})_i\right) + \lambda\|\mathbf{c}\|_2^2$$

and

$$F_D(\theta,\mathbf{c}) = \frac{1}{N}\sum_{i=1}^N \mathcal{L}\left((\mathbf{Y}^{v_0} - \widetilde{\mathbf{A}}_{\theta,D}\mathbf{c})_i\right) + \lambda\|\mathbf{c}\|_2^2.$$

Set

$$C_0 := \frac{1}{N}\sum_{i=1}^N \mathcal{L}(\mathbf{Y}_i^{v_0}), \qquad R := \sqrt{\frac{C_0 - C_{\mathcal{L}}}{\lambda}},$$

where $C_{\mathcal{L}} \in \mathbb{R}$ is the lower bound of $\mathcal{L}$. We observe that every minimizer $\mathbf{c}^\star$ of either problem belongs to the ball $\{\mathbf{c} : \|\mathbf{c}\|_2 \le R\}$. We next show that $F_D \to F$ uniformly on

$$\Theta \times \overline{B}_R, \qquad \overline{B}_R := \{\mathbf{c} \in \mathbb{R}^N : \|\mathbf{c}\|_2 \le R\},$$

as $D \to \infty$. By compactness of Θ and continuity of $\theta \mapsto \mathbf{A}_\theta$,

$$\sup_{\theta\in\Theta} \|\mathbf{A}_\theta\|_{\mathrm{op}} < \infty.$$

Together with (6.2), this implies that, for all sufficiently large D,

$$\sup_{\theta\in\Theta}\|\widetilde{\mathbf{A}}_{\theta,D}\|_{\mathrm{op}} \le B$$

for some $B<\infty$. Hence, for $\|\mathbf{c}\|_2 \le R$, all arguments of $\mathcal{L}$ appearing in F and F_D belong to the compact interval $[-M,M]$, where

$$M := \max_{1\le i\le N}|\mathbf{Y}_i^{v_0}| + BR.$$

The continuity of $\mathcal{L}$ therefore implies its uniform continuity on $[-M,M]$. Let

$$\omega_{\mathcal{L},M}(\delta) := \sup_{\substack{u,v\in[-M,M]\\ |u-v|\le\delta}}|\mathcal{L}(u)-\mathcal{L}(v)|$$

denote its modulus of continuity on this interval. For every $\theta\in\Theta$ and $\|\mathbf{c}\|_2\le R$,

$$\left|\left((\widetilde{\mathbf{A}}_{\theta,D}-\mathbf{A}_\theta)\mathbf{c}\right)_i\right| \le R\eta_D.$$

It follows that

$$\sup_{\theta\in\Theta,\,\|\mathbf{c}\|_2\le R}|F_D(\theta,\mathbf{c})-F(\theta,\mathbf{c})| \le \omega_{\mathcal{L},M}(R\eta_D)\to 0.$$

almost surely as $D\to\infty$. It follows directly that

$$|\mathcal{V}_D-\mathcal{V}|\to 0\,, \tag{6.3}$$

almost surely as $D\to\infty$.

Step 2: convergence of $(\theta_D^\star,\phi_D^\star)_{D>1}$. It remains to establish convergence of the corresponding hedging strategies. The definition of the RKHS norm and the regularization bound give

$$\|\phi_D^\star\|_{\mathcal{H}_{\widetilde{K}_{\theta_D^\star,D}}} \le \|\boldsymbol{\beta}_D^\star\|_2 \le R. \tag{6.4}$$

Consider an arbitrary subsequence. By compactness of Θ, it has a further subsequence, still indexed by D, such that

$$\theta_D^\star\to\bar\theta\,.$$

The uniform RFF approximation of Proposition 3.1 and the continuity of $\theta\mapsto K_\theta$ in $C(\mathcal{X}\times\mathcal{X})$ imply

$$\begin{aligned}\varepsilon_D &:= \sup_{x,x'\in\mathcal{X}}\left|\widetilde{K}_{\theta_D^\star,D}(x,x')-K_{\bar\theta}(x,x')\right|\\ &\le \sup_{\theta\in\Theta}\sup_{x,x'\in\mathcal{X}}\left|\widetilde{K}_{\theta,D}(x,x')-K_\theta(x,x')\right| + \sup_{x,x'\in\mathcal{X}}\left|K_{\theta_D^\star}(x,x')-K_{\bar\theta}(x,x')\right|\to 0.\end{aligned} \tag{6.5}$$

We next verify uniform boundedness. By the reproducing property and (6.4),

$$|\phi_D^\star(x)| \le R\sqrt{\widetilde{K}_{\theta_D^\star,D}(x,x)}.$$

By (6.5), there exists D_0 such that $\varepsilon_D\le 1$ for all $D\ge D_0$. Moreover, since $K_{\bar\theta}$ is continuous on the compact set $\mathcal{X}\times\mathcal{X}$, we may define

$$M := \sup_{x\in\mathcal{X}}K_{\bar\theta}(x,x)<\infty.$$

Hence, for all $D\ge D_0$,

$$\sup_{x\in\mathcal{X}}\widetilde{K}_{\theta_D^\star,D}(x,x)\le M+1,$$

and therefore

$$\sup_{x\in\mathcal{X}}|\phi_D^\star(x)|\le R\sqrt{M+1}.$$

For the finitely many indices $D < D_0$, since each $\phi_D^\star$ is continuous on the compact set $\mathcal{X}$, we have

$$C_0 := \max_{1 \le D < D_0} \sup_{x \in \mathcal{X}} |\phi_D^\star(x)| < \infty.$$

Thus, defining

$$C := \max\left\{C_0, R\sqrt{M+1}\right\} < \infty,$$

we obtain the uniform bound

$$\sup_D \sup_{x \in \mathcal{X}} |\phi_D^\star(x)| \le C < \infty.$$

We now establish equicontinuity. To this end, let us define

$$\rho_D(x,x')^2 := \widetilde{K}_{\theta_D^\star, D}(x,x) + \widetilde{K}_{\theta_D^\star, D}(x',x') - 2\widetilde{K}_{\theta_D^\star, D}(x,x')$$

and

$$\rho_{\bar\theta}(x,x')^2 := K_{\bar\theta}(x,x) + K_{\bar\theta}(x',x') - 2K_{\bar\theta}(x,x').$$

The RKHS inequality gives

$$|\phi_D^\star(x) - \phi_D^\star(x')| \le R\rho_D(x,x'). \tag{6.6}$$

Furthermore, (6.5) implies

$$\rho_D(x,x')^2 \le \rho_{\bar\theta}(x,x')^2 + 4\varepsilon_D.$$

Since $K_{\bar\theta}$ is continuous and $\mathcal{X}$ is compact,

$$\sup_{\|x-x'\|_2 \le \delta} \rho_{\bar\theta}(x,x') \to 0 \qquad \text{as } \delta \downarrow 0.$$

Combining this property with $\varepsilon_D \to 0$ and (6.6) shows that $(\phi_D^\star)_D$ is equicontinuous. Since the family $(\phi_D^\star)_D$ is uniformly bounded and equicontinuous on the compact set $\mathcal{X}$, the Arzelà–Ascoli theorem therefore yields a further subsequence and a function $\bar\phi \in C(\mathcal{X})$ such that

$$\sup_{x \in \mathcal{X}} |\phi_D^*(x) - \bar\phi(x)| \to 0, \tag{6.7}$$

as $D \to \infty$. Moreover, the uniform convergence of the kernels (6.5) and of the strategies (6.7), together with the bound (6.4), allows us to pass to the limit in the positive-semidefiniteness condition of (Paulsen and Raghupathi, 2016, Theorem 3.11), yielding

$$\bar\phi \in \mathcal{H}_{K_{\bar\theta}}, \qquad \|\bar\phi\|_{\mathcal{H}_{K_{\bar\theta}}} \le \liminf_{D\to\infty} \|\phi_D^\star\|_{\mathcal{H}_{\widetilde{K}_{\theta_D^\star, D}}}. \tag{6.8}$$

We next pass to the limit in the empirical hedging loss. For $f \in C(\mathcal{X})$, define

$$\mathcal{R}(f) := \frac{1}{N}\sum_{i=1}^N \mathcal{L}\left(Y^i - B_T^i v_0 - \sum_{k=0}^{n-1} G_{t_{k+1}}^i f(X_{t_k}^i)\right).$$

For each training path i,

$$\left|\sum_{k=0}^{n-1} G_{t_{k+1}}^i \left(\phi_D^\star(X_{t_k}^i) - \bar\phi(X_{t_k}^i)\right)\right| \le \left(\sum_{k=0}^{n-1} |G_{t_{k+1}}^i|\right) \sup_{x\in\mathcal{X}} |\phi_D^\star(x) - \bar\phi(x)| \to 0.$$

Since $\mathcal{L}$ is continuous and the dataset is finite, it follows that

$$\mathcal{R}(\phi_D^\star) \to \mathcal{R}(\bar\phi).$$

Combining this convergence with (6.3), (6.4) and (6.8), we obtain

$$\begin{aligned}
\mathcal{J}_{\bar\theta}(\bar\phi) = \mathcal{R}(\bar\phi) + \lambda\|\bar\phi\|_{\mathcal{H}_{K_{\bar\theta}}}^2 &\le \mathcal{R}(\bar\phi) + \lambda \liminf_{D\to\infty} \|\phi_D^\star\|_{\mathcal{H}_{\widetilde{K}_{\theta_D^\star, D}}}^2 \\
&\le \liminf_{D\to\infty} \left\{\mathcal{R}(\phi_D^\star) + \lambda\|\boldsymbol{\beta}_D^\star\|_2^2\right\} \\
&= \liminf_{D\to\infty} \mathcal{V}_D = \mathcal{V}.
\end{aligned}$$

On the other hand, $\bar{\theta} \in \Theta$ and $\bar{\phi} \in \mathcal{H}_{K_{\bar{\theta}}}$, so the definition of $\mathcal{V}$ gives

$$\mathcal{V} \leq \mathcal{J}_{\bar{\theta}}(\bar{\phi}).$$

Hence,

$$\mathcal{J}_{\bar{\theta}}(\bar{\phi}) = \mathcal{V},$$

and $(\bar{\theta}, \bar{\phi})$ is a global minimizer of the exact deep kernel hedging problem. The argument applies to every subsequence. Consequently, every accumulation point of $\{(\theta_D^\star, \phi_D^\star)\}_{D\geq 1}$ is a global minimizer. If the exact minimizing pair $(\theta^\star, \phi^\star)$ is unique, every subsequence has a further subsequence converging to that same pair. Therefore,

$$\theta_D^\star \to \theta^\star, \qquad \sup_{x\in\mathcal{X}} |\phi_D^*(x) - \phi^\star(x)| \to 0,$$

almost surely as $D \to \infty$. □

6.1.5 Proof of Proposition 3.2

Proof. Since infima over product sets commute, we may exchange the order of minimization and write (3.30) as

$$\inf_{\zeta\in\mathbb{R}} \inf_{\theta\in\Theta,\, \phi\in\mathcal{H}_{K_\theta}} \left\{ \zeta + \frac{1}{\tilde{\alpha}N} \sum_{i=1}^{N} \left(Y^i - B_T^i v_0 - \sum_{k=0}^{n-1} \phi(X_{t_k}^i) G_{t_{k+1}}^i - \zeta \right)_+ + \lambda \|\phi\|_{\mathcal{H}_{K_\theta}}^2 \right\}, \tag{6.9}$$

For every fixed $\zeta \in \mathbb{R}$, the loss

$$u \mapsto \mathcal{L}_{\tilde{\alpha}}^{\mathrm{CVaR}}(u, \zeta) = \zeta + \frac{1}{\tilde{\alpha}}(u - \zeta)_+$$

is continuous, convex and bounded from below. Therefore, Theorem 3.2 implies that the inner minimum is attained by some $(\theta_\zeta^\star, \phi_\zeta^\star)$. It remains to minimize the resulting value function in ζ. Using the finite-dimensional representation from Theorem 3.1 and the proof of Lemma 3.2 with $\mathbf{c} = \mathbf{Q}_\theta^{1/2}\boldsymbol{\alpha} \in \mathbb{R}^N$, we write

$$F(\theta, \mathbf{c}, \zeta) := \zeta + \frac{1}{\tilde{\alpha}N} \sum_{i=1}^{N} \left(Y^i - B_T^i v_0 - (\mathbf{Q}_\theta^{1/2}\mathbf{c})_i - \zeta \right)_+ + \lambda \|\mathbf{c}\|_2^2, \quad \text{and} \quad \mathcal{V}(\zeta) := \min_{\theta\in\Theta,\, \mathbf{c}\in\mathbb{R}^N} F(\theta, \mathbf{c}, \zeta),$$

such that (6.9) is equivalent to

$$\inf_{\zeta\in\mathbb{R}} \mathcal{V}(\zeta).$$

Moreover, we observe that

$$F(\theta, \mathbf{c}, \zeta) \geq \zeta,$$

and completing the square in $\mathbf{c}$, we also have that there exists a constant $C_0 < \infty$, independent of $(\theta, \mathbf{c}, \zeta)$, such that

$$F(\theta, \mathbf{c}, \zeta) \geq -C_0 - \frac{1 - \tilde{\alpha}}{\tilde{\alpha}}\zeta.$$

The value function $\mathcal{V}(\zeta)$ therefore tends to $+\infty$ as $\zeta \to +\infty$ and as $\zeta \to -\infty$, and is therefore coercive. Moreover, under Assumption 1, the continuity of $\theta \mapsto \mathbf{Q}_\theta^{1/2}$ and the uniform coercivity in $\mathbf{c}$ imply that $\mathcal{V}$ is continuous. Hence, $\mathcal{V}$ attains its minimum at some $\zeta^\star \in \mathbb{R}$. Since the corresponding inner minimum is also attained, there exists a global minimizer $(\theta^\star, \phi^\star, \zeta^\star)$ of (3.30). □

6.2 Path-signatures

In this appendix, we provide a brief reminder on path-signatures. For $T > 0$, let $X := (X_t)_{t\leq T}$ be a (continuous) path with finite variation in $\mathbb{R}^d$. Let $\{e_1, \ldots, e_d\} \subset \mathbb{R}^d$ be the canonical basis of $\mathbb{R}^d$ and $A_d = \{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{d}\}$ be the corresponding alphabet. For $i \in \{1, \ldots, d\}$, we write e_i as the blue letter $\mathbf{i}$ and for $n \geq 1, i_1, \ldots, i_n \in \{1, \ldots, d\}$, we write $e_{i_1} \otimes \cdots \otimes e_{i_n}$ as the concatenation of letters $\mathbf{i_1} \cdots \mathbf{i_n}$, that we call a word of length n. We note that $(e_{i_1} \otimes \cdots \otimes e_{i_n})_{(i_1,\ldots,i_n)\in\{1,\ldots,d\}^n}$ is a basis of $(\mathbb{R}^d)^{\otimes n}$ that can be identified with the set of words of length n defined by

$$V_n := \{\mathbf{i_1} \cdots \mathbf{i_n} : \mathbf{i_k} \in A_d \text{ for } k = 1, 2, \ldots, n\}.$$

Moreover, $\boldsymbol{\emptyset}$ denotes the empty word and $V_0 = \{\boldsymbol{\emptyset}\}$ serves as a basis for $(\mathbb{R}^d)^{\otimes 0} = \mathbb{R}$.

Definition 6.1. The signature of X is defined by $\mathbb{X}$ such that

$$\mathbb{X}_t := \left(\mathbb{X}_t^{\mathbf{i_1}\cdots\mathbf{i_n}}\right)_{(\mathbf{i_1}\cdots\mathbf{i_n})\in V_n,\ n\in\mathbb{N}}, \quad t \le T,$$

where $\mathbb{X}^{\emptyset} := 1$, and

$$\mathbb{X}_t^{\mathbf{i_1}\ldots\mathbf{i_n}} := \int_{0<u_1<\cdots<u_n<t} dX_{u_1}^{\mathbf{i_1}} \cdots dX_{u_n}^{\mathbf{i_n}}, \quad (\mathbf{i_1}\ldots\mathbf{i_n}) \in V_n, \quad n \in \mathbb{N}_0.$$

Similarly, the truncated signature of order $M \in \mathbb{N}$ is defined by

$$\mathbb{X}_t^{\le M} := \left(\mathbb{X}_t^{\mathbf{i_1}\cdots\mathbf{i_n}}\right)_{(\mathbf{i_1}\cdots\mathbf{i_n})\in V_n,\ n\le M}, \quad t \le T.$$

such that $\mathbb{X}_t^{\le M} \in \mathbb{R}^{\frac{d^{M+1}-1}{d-1}}$.

In dimension $d = 1$, the signature of X reduces to the sequence of monomials $\left(\frac{1}{n!}(X_t - X_0)^n\right)_{n\in\mathbb{N}}$. In higher dimensions, however, the signature contains iterated integrals involving different components of the path. This makes path-signatures particularly useful for describing path-dependent quantities, especially when augmenting the path with time. The collection of all time-augmented signature terms therefore provides a rich summary of the path's history. As shown in Cuchiero et al. (2025), signatures provide a universal and algebraic representation of paths. In fact, under suitable conditions, the signatures uniquely characterize paths, up to the natural equivalence induced by tree-like pieces. This provides the theoretical foundation for using time-augmented signatures as input features in learning problems involving path-dependent quantities. Rather than using the entire past trajectory, one can represent the path through its time-augmented signature coordinates. In particular, when the signature is truncated at a finite order, this yields a finite-dimensional representation of a path, with higher truncation orders providing increasingly rich information about its history.

For further details on path-signatures and their applications in hedging context, we refer among others to Lyons et al. (2020); Cuchiero et al. (2025); Abi Jaber and Gérard (2025); Abi Jaber et al. (2025); Chevyrev and Kormilitzin (2025).

6.3 Additional experiments on synthetic data

In this appendix, we provide additional results for the numerical experiments on synthetic data.

Comparison to Random Feature Neural Networks (RFNNs). To assess whether the benefits of our approach are specific to the random Fourier approximation (3.16) with `cosine` features and Gaussian kernel, or extend to more general random-feature families, we additionally consider RFNNs based on `ReLU` and `Tanh` activations (Huang et al., 2006; Rahimi and Recht, 2008; Gonon et al., 2023). We first note that our current implementation of (standard) kernel hedging with RFF is a particular RFNN with `cosine` activation. More precisely, for the Gaussian kernel $K(x, x') = \exp(-\|x - x'\|_2^2/2\gamma^2)$, the normalized RFF map is

$$y_D^{\mathrm{RFF}}(x) = \sqrt{\frac{2}{D}} \begin{bmatrix} \cos\left(w_1^\top x/\gamma + b_1\right) \\ \vdots \\ \cos\left(w_D^\top x/\gamma + b_D\right) \end{bmatrix},$$

where the Fourier frequencies and phases are sampled according to

$$w_j \sim p_K = \mathcal{N}(0, I_d), \qquad b_j \sim \mathcal{U}(0, 2\pi),$$

and remain fixed throughout training. For the `ReLU` and `Tanh` alternatives, we similarly sample the hidden weights and biases once according to

$$w_j \sim \mathcal{N}(0, I_d), \qquad b_j \sim \mathcal{N}(0, 1),$$

and keep them fixed during optimization. The corresponding normalized random-feature maps are given by

$$y_D^{\mathrm{ReLU}}(x) = \frac{1}{\sqrt{D}} \begin{bmatrix} \mathrm{ReLU}\left(w_1^\top x + b_1\right) \\ \vdots \\ \mathrm{ReLU}\left(w_D^\top x + b_D\right) \end{bmatrix}, \qquad y_D^{\tanh}(x) = \frac{1}{\sqrt{D}} \begin{bmatrix} \tanh\left(w_1^\top x + b_1\right) \\ \vdots \\ \tanh\left(w_D^\top x + b_D\right) \end{bmatrix}. \tag{6.10}$$

For the classical RFNN hedging strategies, only the output coefficients $\boldsymbol{\beta}$ and the initial portfolio value v_0 are optimized. Their deep counterparts instead apply the random-feature maps in (6.10) to the learned representation $\psi_\theta(x)$, see (3.16). Consequently, the representation parameters θ, the output coefficients $\boldsymbol{\beta}$, and the initial portfolio value v_0 are optimized jointly, while the random weights and biases (w, b) remain fixed. To isolate the effect of the activation function, the `ReLU` and `Tanh` models use identical realizations of w and b within each seed, and all methods share the same number of random features, training seeds, loss function, and optimization settings. These comparisons therefore allow us to disentangle the contribution of the learned representation from that of the underlying random-feature family. The main observation from Table 7 is that the learned representation ψ_θ is far more important than the random-feature activation. Without it, kernel hedging with `cosine` RFF and both shallow RFNNs perform even worse than Black-Scholes. Once the representation is learned, deep kernel hedging with `cosine` and deep RFNN hedging with `ReLU` or `Tanh` features yield very similar results.

Table 7: Out-of-sample metrics for the ATM European call under quadratic hedging with $T = 1/4$, $N = 1000$. Reported values are averages over ten seeds, with standard deviations in parentheses.

	MSE	MAE	$q_{0.95}^{\mathrm{upper}}$	$\mathrm{CVaR}_{0.95}^{\mathrm{upper}}$
BS	1.07e−4	8.28e−3	1.95e−2	2.49e−2
Kernel hedging (`cosine`)	1.33e−4 (1.6e−6)	8.58e−3 (7.7e−5)	1.99e−2 (1.3e−4)	2.78e−2 (1.7e−4)
RFNN hedging (`ReLU`)	1.38e−4 (1.9e−6)	8.91e−3 (7.1e−5)	2.06e−2 (2.8e−4)	2.87e−2 (4.2e−4)
RFNN hedging (`Tanh`)	1.45e−4 (2.9e−6)	8.96e−3 (9.9e−5)	2.15e−2 (3.8e−4)	2.99e−2 (4.8e−4)
Deep kernel hedging (`cosine`)	**6.87e−5** (**8.0e−7**)	**6.29e−3** (**2.4e−5**)	1.46e−2 (1.3e−4)	2.06e−2 (2.5e−4)
Deep RFNN hedging (`ReLU`)	6.90e−5 (4.4e−7)	6.32e−3 (2.4e−5)	1.46e−2 (7.3e−5)	2.06e−2 (1.2e−4)
Deep RFNN hedging (`Tanh`)	6.88e−5 (5.3e−7)	6.33e−3 (3.8e−5)	**1.45e−2** (**1.0e−4**)	**2.04e−2** (**1.4e−4**)

We do not include the heavier deep kernel architectures of Xie et al. (2019) and Mehrkanoon (2019) in our numerical comparison, as they lead to substantial overfitting in our limited-data financial setting.

Robustness of the RFF approximation. To assess the accuracy-efficiency trade-off of the RFF approximation, we first learn the deep representation and kernel bandwidth $\hat{\theta} = (\hat{\omega}, \hat{\gamma})$ and then keep them fixed (such as to fix the learned kernel and thus only assess the RFF accuracy). Using $N = 1000$ training paths and conditionally on the learned kernel $K_{\hat{\theta}}$, we compute the exact RKHS optimizer $\phi^\star_{N,\hat{\theta}}$ through the representer theorem (3.13) and compare it with the RFF solution $\widehat{\phi}_{D,\hat{\theta}}$ in (3.28) for increasing numbers D of random features. Approximation accuracy is measured on an independent test set of $N^{\mathrm{OOS}} = 25,000$ simulated sample paths by the mean squared error between the two strategies

$$\mathrm{MSE}_D = \frac{1}{N^{\mathrm{OOS}}\, n} \sum_{i=1}^{N^{\mathrm{OOS}}} \sum_{k=0}^{n-1} \left(\phi^\star_{N,\hat{\theta}}(X^i_{t_k}) - \widehat{\phi}_{D,\hat{\theta}}(X^i_{t_k}) \right)^2, \tag{6.11}$$

while the reported runtime includes the construction of the RFF features, the solution of the resulting ridge-regression problem, and test-set evaluation, excluding the shared learning of the deep representation $\hat{\theta}$. Figure 3 reports means and standard deviations over 10 RFF draws and shows that the strategy MSE decreases sharply as D increases, approaching 10^{-5}, whereas the computational cost rises roughly linearly. Most of the approximation gain is achieved for moderate values of D around 100, with diminishing improvements beyond approximately $D = 250$, illustrating the trade-off between fidelity to the exact RKHS solution and computational efficiency.

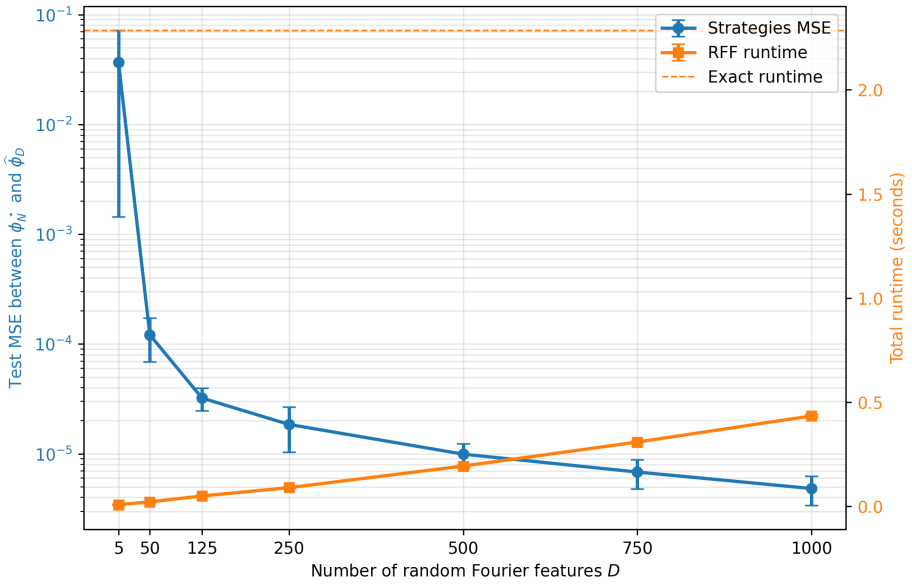


Figure 3: Comparison of the exact RKHS hedging strategy $\phi^\star_{N,\hat{\theta}}$ and its RFF approximation $\widehat{\phi}_{D,\hat{\theta}}$ for $N = 1000$ training paths of an ATM European call under quadratic hedging. The blue curve shows the test-set MSE_D (6.11), while the orange curve reports the runtime required to construct the RFF features, solve the RFF hedging problem (3.29), and evaluate $\widehat{\phi}_{D,\hat{\theta}}$ on the test set. The dashed orange line indicates the corresponding runtime for computing and evaluating the exact RKHS solution $\phi^\star_{N,\hat{\theta}}$. Markers and error bars represent the mean and one standard deviation over 10 RFF draws. The learned deep representation and kernel bandwidth $\hat{\theta} = (\hat{\omega}, \hat{\gamma})$ are held fixed throughout the experiment.

Asian option hedging. We consider the quadratic hedging of ATM arithmetic Asian call options of the form

$$Y_T = \left(\frac{1}{T} \int_0^T S_t dt - S_0 \right)_+ . \tag{6.12}$$

Since the Asian payoff depends on the average of the path, we augment the state space with the running average process $A = (A_t)_{0<t\leq T}$, defined by

$$A_t := \frac{1}{t} \int_0^t S_s \, ds, \qquad 0 < t \leq T,$$

with $A_0 := S_0$. As the underlying state (S_t, ν_t) is Markovian, the augmented state (S_t, ν_t, A_t) is also Markovian. Table 8 reports the out-of-sample results based on synthetic data. Note that, unlike for the European call payoff, we do not include the Black-Scholes benchmark, as there is no explicit hedging strategy for an arithmetic Asian call of the form (6.12) in the Black-Scholes setting. Across all three maturities, the deep kernel hedging strategy substantially outperforms the benchmarks in terms of all the out-of-sample metrics, and its advantage becomes more pronounced as the maturity increases.

6.4 Additional experiments on S&P500 data

In this appendix, we provide additional results for the numerical experiments on real data.

Vanilla features. As first baseline, we use vanilla features as inputs for learning the quadratic hedging strategy of an ATM European call based on real data. Among the learning methods, we observe in Table 9 that the deep kernel hedging remains the best-performing method under quadratic hedging when using the vanilla features as first baseline on S&P500 data, achieving the lowest MSE and MAE, as well as the most favorable $q^{\mathrm{upper}}_{0.95}$ and $\mathrm{CVaR}^{\mathrm{upper}}_{0.95}$ in most settings. However, comparison with the results obtained using

Table 8: Out-of-sample metrics for the ATM Asian call for quadratic hedging with $N = 1000$. Reported values are averages over ten seeds, with standard deviations in parentheses.

$T = 1/12$

	MSE	MAE	$q_{0.95}^{\text{upper}}$	$\text{CVaR}_{0.95}^{\text{upper}}$
Kernel hedging	1.42e−5 (2.9e−7)	2.97e−3 (3.5e−5)	5.64e−3 (1.3e−4)	7.86e−3 (1.1e−4)
Deep hedging	8.75e−6 (4.2e−7)	2.33e−3 (5.8e−5)	4.30e−3 (1.1e−4)	6.08e−3 (1.3e−4)
Deep kernel hedging	**7.91e−6 (1.8e−7)**	**2.23e−3 (3.1e−5)**	**4.10e−3 (1.2e−4)**	**5.80e−3 (1.5e−4)**

$T = 1/4$

	MSE	MAE	$q_{0.95}^{\text{upper}}$	$\text{CVaR}_{0.95}^{\text{upper}}$
Kernel hedging	4.48e−5 (1.1e−6)	5.28e−3 (6.5e−5)	9.31e−3 (1.1e−4)	1.35e−2 (1.9e−4)
Deep hedging	2.52e−5 (1.2e−6)	3.97e−3 (1.0e−4)	6.42e−3 (1.3e−4)	9.48e−3 (2.2e−4)
Deep kernel hedging	**2.12e−5 (5.1e−7)**	**3.67e−3 (4.3e−5)**	**5.93e−3 (9.9e−5)**	**8.77e−3 (2.0e−4)**

$T = 1/2$

	MSE	MAE	$q_{0.95}^{\text{upper}}$	$\text{CVaR}_{0.95}^{\text{upper}}$
Kernel hedging	1.04e−4 (2.4e−6)	8.09e−3 (9.4e−5)	1.41e−2 (2.2e−4)	2.04e−2 (3.0e−4)
Deep hedging	5.86e−5 (3.5e−6)	6.19e−3 (1.8e−4)	9.32e−3 (3.3e−4)	1.42e−2 (5.1e−4)
Deep kernel hedging	**5.00e−5 (1.2e−6)**	**5.78e−3 (7.8e−5)**	**8.86e−3 (2.3e−4)**	**1.34e−2 (3.6e−4)**

truncated time-augmented signatures in Table 5 shows that the vanilla features do not attain the same level of hedging performance since they fail to include any path-dependent information. Moreover, the benefits of signature-based features become increasingly pronounced as the option maturity increases.

Asian option hedging. We consider the quadratic hedging of ATM arithmetic Asian call options of the form (6.12). Based on real data, for each trajectory, we set the initial price of the Asian call used for hedging as

$$v_0^i = \text{AMM}(S_0 = S_0^i, K = S_0^i, T, \sigma = \sqrt{\nu_0^i}),$$

where ν_0^i denotes the initial realized variance of the i-th path and $\text{AMM}(S, K, T, \sigma)$ is the closed-form moment-matching approximation of the arithmetic Asian call price given in (Lo et al., 2014, Proposition 6). Table 10 reports the results. Across all three maturities, the deep kernel hedging achieves strong performances, with performance gaps widening as maturity increases from 10 to 40 days.

Disclosure of interest

No potential conflict of interest was reported by the authors.

Funding

E. Motte is grateful for the financial support from the Fonds de la Recherche Scientifique (F.R.S. - FNRS) through a FRIA grant.

Data Availability Statement

The S&P500 historical data (January 1st, 2010 to May 31st, 2026) used in the numerical experiments are publicly available and can be accessed directly through the `yfinance` Python package. No proprietary or newly collected data were used.

Table 9: Out-of-sample metrics for the quadratic hedging of an ATM European call on real dataset with vanilla features. P&Ls are scaled by the underlying price at the beginning of the periods. Reported values are averages over ten seeds, with standard deviations in parentheses.

$T_{\text{days}} = 10$ days.

	MSE	MAE	$q_{0.95}^{\text{upper}}$	$\text{CVaR}_{0.95}^{\text{upper}}$
BS	3.12e−5	3.48e−3	9.13e−3	1.41e−2
Kernel hedging	2.62e−5 (3.9e−7)	3.72e−3 (2.6e−5)	7.99e−3 (1.9e−4)	1.16e−2 (1.7e−4)
Deep hedging	2.49e−5 (9.3e−7)	3.70e−3 (6.8e−5)	7.67e−3 (2.5e−4)	1.06e−2 (4.0e−4)
Deep kernel hedging	**2.20e−5(5.6e−7)**	**3.35e−3(3.7e−5)**	**6.82e−3 (1.2e−4)**	**9.76e−3 (3.0e−4)**

$T_{\text{days}} = 20$ days.

	MSE	MAE	$q_{0.95}^{\text{upper}}$	$\text{CVaR}_{0.95}^{\text{upper}}$
BS	5.64e−5	4.83e−3	1.17e−2	1.73e−2
Kernel hedging	4.75e−5 (7.7e−7)	4.73e−3 (4.3e−5)	1.05e−2 (3.3e−4)	1.63e−2 (2.2e−4)
Deep hedging	4.45e−5 (2.7e−6)	4.69e−3 (1.4e−4)	1.01e−2 (4.7e−4)	1.45e−2 (1.0e−3)
Deep kernel hedging	**3.82e−5 (7.4e−7)**	**4.29e−3 (3.1e−5)**	**9.08e−3 (2.1e−4)**	**1.25e−2 (4.4e−4)**

$T_{\text{days}} = 40$ days.

	MSE	MAE	$q_{0.95}^{\text{upper}}$	$\text{CVaR}_{0.95}^{\text{upper}}$
BS	9.75e−5	**6.23e−3**	**1.36e−2**	**1.76e−2**
Kernel hedging	1.13e−4 (2.8e−6)	7.71e−3 (1.1e−4)	1.97e−2 (8.6e−4)	2.68e−2 (7.1e−4)
Deep hedging	1.01e−4 (5.5e−6)	7.21e−3 (2.2e−4)	1.82e−2 (7.5e−4)	2.49e−2 (1.3e−3)
Deep kernel hedging	**8.93e−5 (3.5e−6)**	6.58e−3 (1.1e−4)	1.68e−2 (6.1e−4)	2.35e−2 (1.2e−3)

Table 10: Out-of-sample metrics for the quadratic hedging of an ATM Asian call on the real dataset with time-augmented signature features. Reported values are averages over ten seeds, with standard deviations in parentheses.

$T_{\text{days}} = 10$ days.

	MSE	MAE	$q_{0.95}^{\text{upper}}$	$\text{CVaR}_{0.95}^{\text{upper}}$
Kernel hedging	1.81e−5 (5.1e−7)	2.84e−3 (3.2e−5)	5.94e−3 (1.9e−4)	1.02e−2 (2.1e−4)
Deep hedging	1.59e−5 (9.0e−7)	2.77e−3 (6.2e−5)	5.30e−3 (2.7e−4)	8.61e−3 (4.3e−4)
Deep kernel hedging	**1.30e−5 (5.4e−7)**	**2.51e−3 (2.5e−5)**	**4.71e−3 (6.2e−5)**	**7.90e−3 (2.1e−4)**

$T_{\text{days}} = 20$ days.

	MSE	MAE	$q_{0.95}^{\text{upper}}$	$\text{CVaR}_{0.95}^{\text{upper}}$
Kernel hedging	2.31e−5 (6.5e−7)	3.34e−3 (5.1e−5)	6.82e−3 (2.3e−4)	1.06e−2 (2.7e−4)
Deep hedging	2.15e−5 (1.5e−6)	3.33e−3 (9.8e−5)	6.69e−3 (4.3e−4)	9.80e−3 (9.4e−4)
Deep kernel hedging	**1.76e−5 (5.5e−7)**	**2.95e−3 (3.1e−5)**	**5.92e−3 (1.2e−4)**	**8.53e−3 (3.8e−4)**

$T_{\text{days}} = 40$ days.

	MSE	MAE	$q_{0.95}^{\text{upper}}$	$\text{CVaR}_{0.95}^{\text{upper}}$
Kernel hedging	4.11e−5 (1.1e−6)	4.62e−3 (6.5e−5)	9.56e−3 (2.5e−4)	1.41e−2 (3.2e−4)
Deep hedging	3.38e−5 (2.8e−6)	4.07e−3 (7.6e−5)	**7.58e−3(2.4e−4)**	1.27e−2 (1.6e−3)
Deep kernel hedging	**2.96e−5 (8.3e−7)**	**3.95e−3 (4.6e−5)**	7.72e−3 (1.6e − 4)	**1.13e−2 (3.7e−4)**

Declaration of generative AI use

Generative AI was used to improve the spelling, grammar and flow of the article. It was also used to generate the flow chart in Figure 1.

Author contributions

CRediT: **J. Dupret:** Methodology, Software, Validation, Formal analysis, Writing; **D. Hainaut:** Conceptualization, Validation, Supervision; **E. Motte:** Conceptualization, Methodology, Software, Validation, Formal analysis, Writing.

References

Eduardo Abi Jaber and Louis-Amand Gérard. Hedging with memory: shallow and deep learning with signatures. *arXiv preprint arXiv:2508.02759*, 2025.

Eduardo Abi Jaber, Donatien Hainaut, and Edouard Motte. Signature approach for pricing and hedging path-dependent options with frictions. *arXiv preprint arXiv:2511.23295*, 2025.

Haim Avron, Michael Kapralov, Cameron Musco, Christopher Musco, Ameya Velingker, and Amir Zandieh. Random Fourier features for kernel ridge regression: Approximation bounds and statistical guarantees. In *International Conference on Machine Learning*, pages 253–262. PMLR, 2017.

Heinz H. Bauschke and Patrick L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2017.

Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.

Hans Buehler, Lukas Gonon, Josef Teichmann, and Ben Wood. Deep hedging. *Quantitative Finance*, 19(8): 1271–1291, 2019.

Alexandre Carbonneau and Frédéric Godin. Equal risk pricing of derivatives with deep hedging. *Quantitative Finance*, 21(4):593–608, 2021.

Stephan K. Chalup and Andreas Mitschele. Kernel methods in finance. In *Handbook on Information Technology in Finance*, pages 655–687. Springer Berlin, Heidelberg, 2008.

Kuo-Tsai Chen. Integration of paths, geometric invariants and a generalized Baker-Hausdorff formula. *Annals of Mathematics*, 65(1):163–178, 1957.

Ilya Chevyrev and Andrey Kormilitzin. A primer on the signature method in machine learning. In *Signature Methods in Finance: An Introduction with Computational Applications*, pages 3–64. Springer Nature Switzerland Cham, 2025.

Nicola Muca Cirone and Cristopher Salvi. Rough kernel hedging. *arXiv preprint arXiv:2501.09683*, 2025.

Nello Cristianini and John Shawe-Taylor. *An Introduction to Support vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

Christa Cuchiero, Francesca Primavera, and Sara Svaluto-Ferro. Universal approximation theorems for continuous functions of càdlàg paths and Lévy-type signature models. *Finance and Stochastics*, pages 1–54, 2025.

Jean-Loup Dupret and Donatien Hainaut. Deep learning for high-dimensional continuous-time stochastic optimal control without explicit solution. *Operations Research*, 74(3):1241–1262, 2026.

Kun Fang, Fanghui Liu, Xiaolin Huang, and Jie Yang. End-to-end kernel learning via generative random Fourier features. *Pattern Recognition*, 134:109057, 2023.

Owen Futter, Nicola Muça Cirone, and Blanka Horvath. Kernel learning for mean-variance trading strategies. *arXiv preprint arXiv:2507.10701*, 2025.

Alessandro Gnoatto, Silvia Lavagnini, and Athena Picarelli. Deep quadratic hedging. *Mathematics of Operations Research*, 50(4):2972–3009, 2025.

Lukas Gonon, Lyudmila Grigoryeva, and Juan-Pablo Ortega. Approximation bounds for random neural networks and reservoir systems. *The Annals of Applied Probability*, 33(1):28–69, 2023.

Donatien Hainaut. American option pricing with model constrained Gaussian process regressions. *Applied Mathematics and Computation*, 512:129747, 2026.

Donatien Hainaut and Jean-Loup Dupret. Optimal control by policy iterations and constrained Gaussian process regressions. *European Journal of Operational Research*, 330(2):525–539, 2026.

Ralf Herbrich. *Learning Kernel Classifiers: Theory and Algorithms*. The MIT Press, 2001.

Steven L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The Review of Financial Studies*, 6(2):327–343, 1993.

Wenbin Hu and Tomasz Zastawniak. Pricing high-dimensional American options by kernel ridge regression. *Quantitative Finance*, 20(5):851–865, 2020.

Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006.

Mingyuan Jiu and Hichem Sahbi. Nonlinear deep kernel learning for image annotation. *IEEE Transactions on Image Processing*, 26(4):1820–1832, 2017.

Mattias Jonsson and K. Ronnie Sircar. Partial hedging in a stochastic volatility environment. *Mathematical Finance*, 12(4):375–409, 2002.

Alec Koppel, Garrett Warnell, Ethan Stump, and Alejandro Ribeiro. Parsimonious online learning with kernels via sparse projections in function space. *Journal of Machine Learning Research*, 20(3):1–44, 2019.

Věra Kurková and Marcello Sanguineti. Approximate minimization of the regularized expected error over kernel models. *Mathematics of Operations Research*, 33(3):747–756, 2008.

Quentin Le Moal. *Parsimonious Online Learning with Kernels and Random Features with Applications to Stochastic Optimal Control*. PhD thesis, EPFL, 2025.

Chien-Ling Lo, Ken Palmer, and Min-Teh Yu. Moment-matching approximations for Asian options. *Journal of Derivatives*, 21(4):103–122, 2014.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Machine Learning*. PMLR, 2019.

Mike Ludkovski and Yuri Saporito. Krighedge: Gaussian process surrogates for delta hedging. *Applied Mathematical Finance*, 28(4):330–360, 2021.

Eva Lütkebohmert, Thorsten Schmidt, and Julian Sester. Robust deep hedging. *Quantitative Finance*, 22 (8):1465–1480, 2022.

Terry Lyons, Sina Nejad, and Imanol Perez Arribas. Non-parametric pricing and hedging of exotic derivatives. *Applied Mathematical Finance*, 27(6):457–494, 2020.

Siamak Mehrkanoon. Deep neural-kernel blocks. *Neural Networks*, 116:46–55, 2019.

Siamak Mehrkanoon and Johan A.K. Suykens. Deep hybrid neural-kernel networks using random Fourier features. *Neurocomputing*, 298:46–54, 2018.

Charles A. Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research*, 2006.

Oskari Mikkilä and Juho Kanniainen. Empirical deep hedging. *Quantitative Finance*, 23(1):111–122, 2023.

Edouard Motte and Donatien Hainaut. Partial hedging in rough volatility models. *SIAM Journal on Financial Mathematics*, 15(3):601–652, 2024.

Ke Nian, Thomas F. Coleman, and Yuying Li. Learning minimum variance discrete hedging directly from the market. *Quantitative Finance*, 18(7):1115–1128, 2018.

Sebastian W. Ober, Carl E Rasmussen, and Mark van der Wilk. The promises and pitfalls of deep kernel learning. In *Uncertainty in Artificial Intelligence*, pages 1206–1216. PMLR, 2021.

Vern I. Paulsen and Mrinal Raghupathi. *An introduction to the theory of reproducing kernel Hilbert spaces*, volume 152. Cambridge university press, 2016.

Huyên Pham. *Continuous-time Stochastic Control and Optimization with Financial Applications*. Springer Berlin, Heidelberg, 2009.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2007.

Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in Neural Information Processing Systems*, volume 21, 2008.

Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels*. MIT Press, 2002.

Bernhard Schölkopf, Ralf Herbrich, and Alexander J. Smola. A generalized representer theorem. In *International Conference on Computational Learning Theory*, pages 416–426. Springer, 2001.

John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge University Press, 2004.

Aman Sinha and John C. Duchi. Learning kernels with random features. *Advances in Neural Information Processing Systems*, 29, 2016.

Yitong Sun. *Random Features Methods in Supervised Learning*. PhD thesis, University of Michigan, 2019.

Danica J Sutherland and Jeff Schneider. On the error of random Fourier features. In *Uncertainty in Artificial Intelligence*, pages 862–871, 2015.

Jacques Wainer and Pablo Fonseca. How to tune the RBF SVM hyperparameters? An empirical evaluation of 18 search algorithms. *Artificial Intelligence Review*, 54(6):4771–4797, 2021.

Christopher K.I. Williams and Carl Edward Rasmussen. *Gaussian Processes for Machine Learning*. MIT press Cambridge, MA, 2006.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial Intelligence and Statistics*, pages 370–378. PMLR, 2016a.

Andrew Gordon Wilson, Zhiting Hu, Russ R Salakhutdinov, and Eric P Xing. Stochastic variational deep kernel learning. *Advances in Neural Information Processing Systems*, 29, 2016b.

Jiaxuan Xie, Fanghui Liu, Kaijie Wang, and Xiaolin Huang. Deep kernel learning via random Fourier features. *arXiv preprint arXiv:1910.02660*, 2019.

Xiantong Zhen, Haoliang Sun, Yingjun Du, Jun Xu, Yilong Yin, Ling Shao, and Cees Snoek. Learning to learn kernels with variational random features. In *International Conference on Machine Learning*, pages 11409–11419. PMLR, 2020.