Fractional Brownian Motion for Benchmarking Machine Learning Algorithms in Non-linear Estimation

[1]Aldo Taranto | [2]Ron Addie
ORCID: 0000-0001-6763-4997 | ORCID: 0000-0002-6664-8462
Aldo.Taranto@anu.edu.au | Ron.Addie@unisq.edu.au
School of Computing | School of Mathematics, Physics and Computing
Australian National University | University of Southern Queensland
Canberra, 2601, Australian Capital Teritory, AUSTRALIA | Toowoomba, 4350, Queensland, AUSTRALIA

10 January 2026

***Abstract*—This paper introduces a novel benchmarking framework based on fractional Brownian motion (fBm) for evaluating advanced artificial intelligence (AI) and machine learning (ML) algorithms. The approach leverages the statistical structure of fBm to generate theoretically grounded, high-dimensional datasets of unlimited size, enabling reproducible and scalable model evaluation under controlled stochastic conditions. Rather than predicting the supremum or infimum of fBm paths directly, the proposed benchmark focuses on estimating a nonlinear functional derived from path extremes –a transformation that depends on the joint behavior of the supremum, infimum, and range of future segments. This formulation produces a complex, highly nonlinear target function that tests the limits of ML methods in functional estimation, particularly under noise, drift, and persistence variations. Comparisons across six ML architectures reveal a striking scaling paradox: while model performance is broadly comparable at small data sizes ($N = 10^2$), a sharp divergence emerges at scale ($N = 10^4$), where global mapping models (LR, MLP) converge to high predictive accuracy ($R^2 \approx 0.96$) while local and ensemble methods suffer a total generalization collapse to deeply negative R^2 values. The benchmark is readily tunable through parameters including the Hurst exponent (H), drift (μ), and temporal horizon, enabling systematic stress-testing across both persistent and rough-path regimes. Six widely used ML algorithms were evaluated on this benchmark: Linear Regression (LR), Random Forest (RF), Gradient Boosting (GB), Support Vector Machines (SVM), Artificial Neural Networks (ANN), and k-Nearest Neighbours (kNN). Overall, the fBm-based benchmark highlights the intrinsic difficulty of learning nonlinear functionals of stochastic paths and provides a tunable, mathematically robust environment for assessing algorithmic robustness. This framework offers a rigorous, scalable testbed for advancing the development and comparison of ML methods under complex, noise-dominated dynamics.**

***Index Terms*—Fractional Brownian motion (fBm), Machine learning (ML), Algorithms, Benchmarking.**

I. Introduction

Any newly developed machine learning (ML) algorithm should undergo rigorous benchmarking to assess both its intended performance improvements (e.g., scalability) and its fundamental characteristics, such as robustness, adaptability to higher-dimensional settings, and generalization to out-of-sample data. A limitation of many traditional benchmarking approaches, such as training artificial neural networks (ANNs) on large-scale image datasets (e.g., one million cat–dog images), is their lack of scalability. For example, if a more powerful large language model (LLM) requires training on datasets that are 100 times larger, such data may simply not exist. After scraping billions of images from the internet, it is unclear how one could realistically obtain 100× more images of cats or other common objects.

This paper introduces a framework capable of generating unlimited test cases, enabling scalable and rigorous evaluation of ML algorithms. The benchmark centers on estimating the supremum *function* on the space of fractional Brownian motion (fBm) paths with negative drift, from sample data. Brownian motion was discovered experimentally by Brown [1], then used by Einstein as evidence of molecular motion in liquids [2], and a mathematical model was developed by Wiener [3], for which reason it is also known as the Wiener process.

Brownian motion can be generalized in several independent ways. One useful generalization is to add linear drift. Brownian motion with negative linear drift has a well-defined supremum, over the interval $[0, \infty)$, with probability 1, and similarly the infimum of Brownian motion with positive drift is well-defined. For this reason, the supremum of Brownian motion with negative drift is an interesting, non-linear function which depends on every dimension of the infinite dimensional space of possible paths of Brownian motion. We further extend this framework to fBm, whose rich, infinite-dimensional structure provides a rigorous substrate for stress-testing advanced ML algorithms. This includes biologically inspired agent-based methods that rely on structured randomness rather than explicit descent [4], [5], [6]–[8].

Unlike other endeavours in ML, where one aims to find a

more accurate novel algorithm, compared to ANN or SVM for example, with high estimation/prediction accuracy, in this paper we wish to undertake the reverse research process, i.e. to produce a benchmarking framework that is so complex that it leads to poor ML algorithm performance. A benchmark test can be evaluated by the criteria listed in Table I. Our initial selection for a benchmark test meets the first five criteria. A second test, developed by varying the first test, meets *all* the criteria in Table I.

TABLE I: Benchmark Complexity Criteria

Criterion	Criterion Description
1	The benchmark is a family of functions, from a high-dimensional space (the domain space) to $\mathbb{R}$, indexed by a set of parameters which allow key features of the "unknown" function to be varied.
2	All functions are nonlinear.
3	All functions depend on an unbounded number of dimensions of the domain space.
4	The number of dimensions of the domain space can be increased without limit.
5	The number of data points (samples) can be increased without limit (and this can be done efficiently).
6	The non-linear function is "hidden", i.e., hard to estimate, and in fact the degree of difficulty of finding the unknown function can be arbitrarily increased, with appropriate choice of the parameters of the family.

Table I was used to guide the identification of our complex target function for estimation. The function being estimated is the supremum (sup) of a fBm path with negative drift, and the space on which this function is defined is the infinite dimensional space of possible fBm paths. The infimum (inf) of a fBm path with positive drift is the dual counterpart of this supremum, and both appear as components of our complex target function defined in (4). Until that definition is reached, we use the two terms interchangeably. This is one reason that our complex function provides challenging data examples, i.e. that the sup less drift depends on every dimension of the fBm space.

Definition I.1. ***(Fractional Brownian Motion)*** *[9]. A fBm is a centered Gaussian process $B_H(t), t \geq 0$ with the covariance function,*

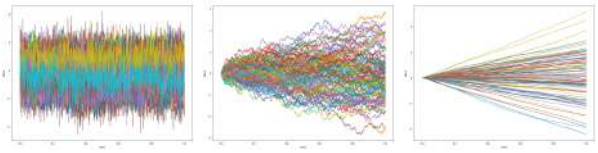
$$\mathbb{E}\big[B_H(t)B_H(s)\big] = \frac{1}{2}\Big(|t|^{2H} + |s|^{2H} - |t-s|^{2H}\Big), \qquad (1)$$

where $H \in (0,1)$, is called the Hurst parameter/exponent/index. The value of H determines the degree of correlation between fBm increments,

- *if $H < 1/2$, then the increments of the process are negatively correlated.*
- *if $H = 1/2$, then the process is Brownian motion, or the Wiener process,*
- *if $H > 1/2$, then the increments of the process are positively correlated,*

■

The fBm, defined as $B_H(t)$ in (1), is illustrated by simulating three instances of a plot of 100 paths of fBm in the cases where $H = 0.01$, 0.5, and 0.99, as shown in Figure 1.



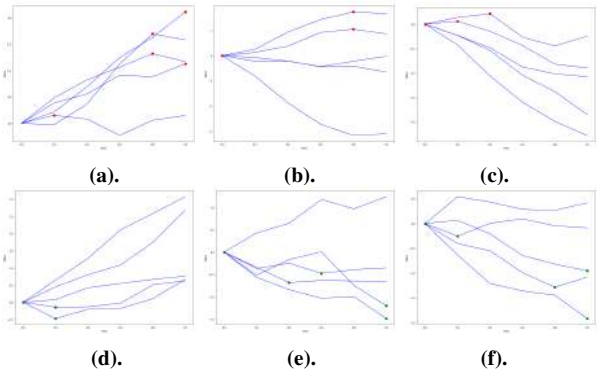
(a). **(b).** **(c).**

Fig. 1: 100 Simulations of 1000-Step fBms with Varying Hurst Exponents
(a). Low Hurst exponent, $H = 0.01$, Drift $= 0$.
(b). Medium Hurst exponent, $H = 0.5$, Drift $= 0$.
(c). High Hurst exponent, $H = 0.99$, Drift $= 0$.

Figure 1 shows the nature of fBm and the extent of correlation amongst successive time steps (i.e. autocorrelation).

Setting the Hurst parameter H too low produces paths that visually resemble white noise, though the increments in fact carry strong negative autocorrelation that is simply not apparent from a plot or diagram. Setting H too high yields almost deterministic linear trends; neither extreme provides a realistic model of the stochastic processes relevant to our work. Although $H = 0.5$ might appear to be a natural compromise, it reduces fBm to standard Brownian motion (Bm) and thus pure Brownian motion may already provide a sufficiently rich and complex path space on which to define the estimation target. However, introducing correlation –positive or negative –through fBm expands this space and guards against the possibility that it does not. For illustrative purposes, we therefore adopt $H = 0.75$ in several scenarios, ensuring that the generated data exhibit meaningful dependence or 'memory'. Nevertheless, we will vary H more broadly in sensitivity analyses to ensure that a representative range of behaviours is explored.

We now generate three plots of five instances of fBm with $H = 0.75$ plus negative, zero and positive drift in which the $\sup$ of each path is indicated by a red dot. We also include three plots of five instances of fBm with with $H = 0.75$ plus negative, zero and positive drift in which the $\inf$ of each path is indicated by a green dot. This are shown in Figure 2.



(a). (b). (c).

(d). (e). (f).

Fig. 2: 5 fBm Paths with H = 0.75 and Various Drift Parameters

(a). Supremum of fBm with positive drift of 2.
(b). Supremum of fBm with neutral drift of 0.
(c). Supremum of fBm with negative drift -2.
(d). Infimum of fBm with positive drift of 2.
(e). Infimum of fBm with neutral drift of 0.
(f). Infimum of fBm with negative drift of -2.

Figure 2 illustrates the difficulty of ML algorithms learning the supremum and infimum as functions from sampled values of fBm trajectories, a challenge that intensifies with the introduction of a drift term. These simulations will be elaborated further in the Methodology (§III). This complexity is due to a number of reasons,

- **Hurst Parameter**: When $H > 0.5$, the increments of fBm are positively correlated, leading to more persistent behavior. This persistence complicates the estimation of the supremum, because past high values are likely to influence test high values [10]. Likewise, for the estimation of the infimum, with past low values influencing test low values.
- **Complex Distribution**: The distribution of the supremum (and infimum) of fBm is complex and does not have a simple or even well known closed-form expression [11], [12].
- **Overlap in Distributions:** While the supremum and infimum may diverge significantly across spatial dimensions, they exhibit increasing convergence near the origin. In the temporal dimension, this overlap can culminate in simultaneity, where both extrema coincide at a single point in time.
- **Bounds and Approximations**: While there are known bounds and approximations available for the expected supremum of fBm, these are often not narrow and can vary significantly depending on the value of H. Tighter bounds are difficult to derive and this sensitivity to initial conditions sheds further light on the extent of the complexity [10]. Despite this, [13] give a distribution for the $\sup\{\text{fBm}\}$, and mean and variance. These formulae are exact, and no discrepancy from them has been detected by simulation.

The proposed multi-dimensional supremum fBm framework demonstrates four critical advantages for ML benchmarking applications, each corresponding to the criteria established in Table I.

1) Benefit 1: The temporal dimension can be arbitrarily extended to accommodate varying estimation horizons. This can be seen as being able to add an infinite number of time columns to a table.
2) Benefit 2: The stochastic simulation framework enables generation of unlimited independent sample paths for robust statistical validation. This can be seen as being able to add an infinite number of path rows to the table.
3) Benefit 3: The continuous parameter space allows comprehensive exploration across drift, diffusion, and Hurst exponent regimes. This can be seen as each parameter multiplying the number of table snapshots and hence the dimensionality of the data.
4) Benefit 4: The supremum construction is intrinsically and comprehensively non-linear –with respect to every dimension of the underlying space and across the entire domain –while the tunable parameters of the fBm-plus-drift process allow its statistical properties to be systematically varied over a wide range of behaviours.

Having introduced the $\sup$ and $\inf$ of fBm as a viable candidate for advanced benchmarking of ML algorithms, we review the relevant literature to ensure that prior work in this field has been adequately appreciated and that our proposed framework is positioned correctly within it.

II. Literature Review

fBm has emerged as a critical stochastic framework for evaluating the performance of ML models on non-stationary and long-memory data. Unlike standard Brownian Motion, fBm is characterized by a Hurst parameter $H \in (0, 1)$, which dictates the correlation structure and "memory" of the process.

A. Theoretical Foundations and Path Regularity

The formalization of fBm by Mandelbrot and Van Ness [14] established the integral representation and self-similarity properties that make it a robust candidate for synthetic data generation. In the context of algorithmic robustness, Benassi *et al.* [15] and Salminen and Vallois [16] provide the analytical tools necessary to characterize sample path regularity. For ML applications, these properties define the "roughness" or "smoothness" of input signals, serving as a controlled variable for testing the generalization limits of neural architectures.

B. Long-Range Dependence (LRD) and Network Traffic

A primary motivation for using fBm in CompSc benchmarking is its ability to model Long-Range Dependence (LRD). Norros [17] demonstrated that telecommunication workloads and network traffic naturally exhibit fBm-like behavior. Consequently, fBm-driven models have become the gold standard for benchmarking predictive algorithms in systems research, ensuring that models can capture dependencies across varying temporal scales –a task where traditional Markovian models often fail.

C. Parameter Estimation and Wavelet-Based Inference

A significant subset of the literature focuses on the inverse problem: estimating the Hurst parameter from observed sequences. This serves as a standard benchmark for regression models. Wavelet-based estimation, popularized by Meyer *et al.* [18] and further refined by Hong *et al.* [19] and, Hamza and Hmood [20], provides a multi-resolution framework that localizes information in both time and scale. These methods provide a high-fidelity "ground truth" against which modern deep learning estimators (e.g., CNNs or Transformers) are frequently compared [20].

D. Stochastic Differential Equations (SDEs) and System Robustness

Recent work has integrated fBm into broader classes of SDEs to test the stability of dynamical systems. Tan [21] analyzed system sensitivity to initial conditions in fBm-driven environments, providing a metric for evaluating the robustness of Reinforcement Learning (RL) agents in volatile environments. Furthermore, the connection between fBm and Reproducing Kernel Hilbert Spaces (RKHS) [22] offers a bridge to kernel-based learning methods, allowing for explicit expression of fractional integrals in machine learning kernels.

E. Domain-Specific Benchmarking: Finance and Control

While theoretically grounded, fBm finds its most rigorous benchmarking applications in high-frequency finance. Research by Vardar Acar *et al.* [11], [23] and others [24], [25] utilizes fBm to model asset pricing and transaction costs under non-Gaussian assumptions. These studies provide a diverse set of "stress-test" scenarios for ML-based portfolio optimization and consumption models [26], where capturing the supremum of the process is vital for risk management.

F. Benchmarking and Evaluating Non-Linear Estimation and Optimisation

Beyond the core theoretical developments on fBm itself, the present paper also intersects with a broader literature on benchmarking and evaluating non-linear estimation and optimisation methods. A substantial body of work has examined how stochastic, non-convex and non-linear algorithms can be systematically assessed through controlled synthetic datasets, stress-testing regimes, and statistically reproducible simulation frameworks. For example, Dolan and Moré [27] introduced performance profiles as a principled benchmarking tool for comparing optimisation algorithms across heterogeneous test problems, establishing a standard still widely used in non-linear optimisation research. Similar concerns arise in the evaluation of stochastic search and metaheuristic algorithms, with Hansen *et al.* [28] and Auger and Hansen [29] highlighting the importance of well-defined test landscapes, reproducible randomness, and statistically meaningful performance metrics.

In the ML domain, Bergstra and Bengio [30] and Hutter, Kotthoff and Vanschoren [31] have emphasised the need for reproducible benchmarking when comparing non-linear optimisation procedures for high-dimensional models, noting that algorithmic performance can vary dramatically depending on noise structure, curvature, and parameter scaling. A related line of work explores the use of synthetic stochastic processes –often involving long-range dependence or heavy-tailed behaviour –to evaluate robustness of non-linear estimators. Examples include the benchmarking frameworks of Taqqu and Teverovsky [32] and Bardet *et al.* [33] for Hurst-parameter estimation methods.

The benchmarking literature is directly relevant here because the proposed supremum-fBm framework serves as a controlled generator of diverse, parameterised, non-linear stochastic behaviours. This broader work reinforces the need for rigorous evaluation protocols, reproducible experimental setups, and stress-testing across varied distributional and dependency structures.

III. Methodology

A. Extreme fBm Data Generation

The way in which we derive the data for ML algorithm benchmarking is to first generate 1,000 of the 1,000 time step fBm paths with various Hurst exponents and various drift parameters, as shown in Figure 3.

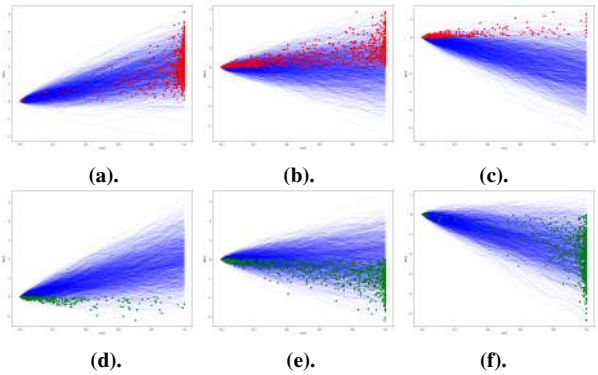
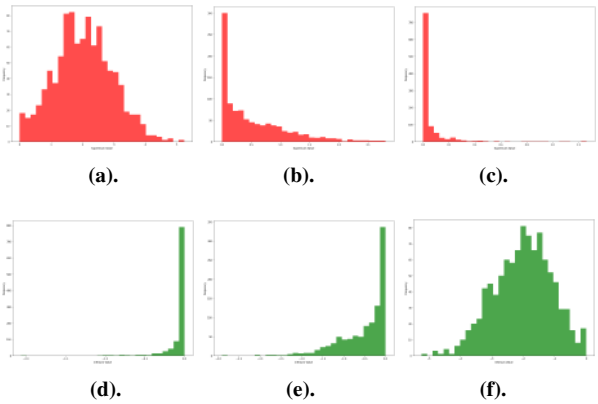


Fig. 3: 1,000 fBm Paths with Hurst Exponent 0.75 and Various Drift Parameters

(a). Supremum of fBm with positive drift of 2.
(b). Supremum of fBm with neutral drift of 0.
(c). Supremum of fBm with negative drift of -2.
(d). Infimum of fBm with positive drift of 2.
(e). Infimum of fBm with neutral drift of 0.
(f). Infimum of fBm with negative drift of -2.

Figure 3 shows both the paths in blue and the extreme points, which is an elaboration of Figure 2. The points form our 'input data' for our six ML algorithms that we will test. The initial data was then simulated 10,000 times so that the distribution can be visualised, as shown in the histograms in Figure 4.



(a). (b). (c).

(d). (e). (f).

Fig. 4: Histograms of 10,000 fBm Paths (1,000 Steps, H = 0.75) with Varying Drift

(a). Histogram of supremum of fBm with positive drift of 2.
(b). Histogram of supremum of fBm with neutral drift of 0.
(c). Histogram of supremum of fBm with negative drift of -2.
(d). Histogram of infimum of fBm with positive drift of 2.
(e). Histogram of infimum of fBm with neutral drift of 0.
(f). Histogram of infimum of fBm with negative drift of -2.

Figure 4 shows the histograms of various extremes of fBm.

(Covariance of fBm with Drift and Diffusion): The covariance of standard fBm (zero drift, unit diffusion) is,

$$\mathbb{E}\Big[B_H(t)B_H(s)\Big] = 0.5\Big(|t|^{(2H)} + |s|^{(2H)} - |t-s|^{(2H)}\Big).$$

For fBm with non-zero drift, $\mu \neq 0$ and non-unit diffusion, $\sigma \neq 1$, the process is constructed as,

$$X(t) = \mu \cdot t + \sigma \cdot B_H(t),$$

where,

- $\mu \cdot t$ is the deterministic drift component (linear trend).
- σ scales the amplitude of the fBm fluctuations.
- $B_H(t)$ is the standard fBm with the covariance above.

The autocovariance of a zero-mean fBm, $B_H(t)$ with Hurst parameter $H \in (0,1)$ and variance parameter σ^2 is,

$$\text{Cov}\left(B_H(t), B_H(s)\right) = \frac{\sigma^2}{2}\left(|t|^{2H} + |s|^{2H} - |t-s|^{2H}\right).$$

For an fBm with deterministic drift,

$$X(t) = \mu \cdot t + B_H(t),$$

the autocovariance remains unchanged, because drift terms are deterministic and do not affect covariance,

$$\text{Cov}\left(X(t), X(s)\right) = \text{Cov}\left(B_H(t), B_H(s)\right).$$

Thus, adding a linear drift modifies the mean but not the covariance structure of the process.

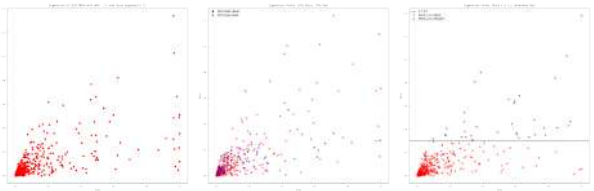
We use estimation of the sup and inf of fBm with drift to benchmark six of the most widely used ML algorithms available in the python SciKit-Learn package, on the sup/inf of fBm data, as shown in Table II.

Table II shows the standard and reasonable settings that were used. All models were evaluated under default or lightly specified hyperparameters; SVR in particular uses `gamma='scale'`, which varies with feature dimension. A systematic grid search across model settings and across data sizes is left for future work.

TABLE II: SciKit-Learn Models

Model	Name	SciKit-Learn Model Details
LR	Linear regression	LinearRegression(), or Linear
RF	Random forest	RandomForestRegressor(n_estimators =100, random_state=42, n_jobs=-1)
GB	Gradient boosting	GradientBoostingRegressor(random_state =42)
SVR	Support vector regression	SVR(kernel='rbf') or Support Vector Machine (SVM), Radial Basis Function (RBF)
KNN	k-Nearest neighbors	KNeighborsRegressor()
MLP	Artificial neural network	ANN or MLPRegressor(hidden_layer_sizes =(100, 50), max_iter=500, random_state=42)

The use of fBm data as a novel benchmarking framework is further highlighted in how the data can be used for either estimation/prediction or for classification, where appropriate. This is illustrated in Figure 5.



(a). (b). (c).

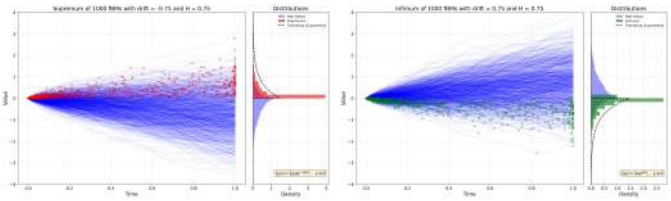
Fig. 5: Estimation and Classification Implemented on sup(fBm) with Negative Drift

(a). Original $\sup\{\text{fBm}\}$ data with $H = 0.75$ and $\mu = -2$: All Red.
(b). Estimation: In this example, 30% of the previous supremum points are Purple, 70% are Red.
Objective: Can the ML algorithm estimate or predict which points belong to which class? (c). Classification: In this example, any point < 0.3 is Red, otherwise Purple.
Objective: Can the ML algorithm correctly classify which points belong to which class?

Figure 5(a) shows one possible way how the sup(fBm) data can be segmented into a Build/training set (e.g. 70%) and a Test set (e.g. 30%) for estimation, whilst (b) shows how the sup(fBm) data can be segmented by a horizontal threshold level into two sets that need to be classified. The emphasis of this paper is placed solely on estimation.

B. Constructing the Target Function

Before proceeding to the benchmark tests of the six algorithms, we investigate the key features of the sup, as a function defined on fBm paths, as shown in Figure 6.



(a). **(b).**

Fig. 6: Path Histograms
(a). Supremum of fBm paths.
(b). Infimum of fBm paths.

Figure 6 shows how the histograms are generated from the paths, but the density functions on the histograms assume simple Bm ($H = 1/2$) and not fBm ($H \neq 1/2$).

Now, the density of the $\sup(\text{Bm},\ \mu < 0)$ is [34],

$$f_{1/2}(x) = 2|\mu|e^{-2|\mu|x}, \quad x \geq 0, \quad \mu < 0,$$

and by simple symmetry, the density of the $\inf(\text{Bm},\ \mu > 0)$ is,

$$f_{1/2}(x) = 2\mu e^{2\mu x}, \quad x \leq 0, \quad \mu > 0.$$

These formulae for the density are not applicable to the density of the sup of fBm less drift. Currently, no exact formula for the density of the sup of fBm with negative drift is known[1].

Let $W(t)$ be standard Bm with $W(0) = 0$.

Definition III.1. ***(Supremum of Brownian Motion with Negative Drift).*** *For $X(t) = W(t) - ct$ with $c > 0$, the running supremum $S = \sup_{t \geq 0} X(t)$ has the exponential density,*

$$f_S(x) = 2c\, e^{-2cx}, \qquad x \geq 0.$$

Definition III.2. ***(Infimum of Brownian Motion with Positive Drift).*** *For $X(t) = W(t) + ct$ with $c > 0$, the running infimum $I = \inf_{t \geq 0} X(t)$ has the exponential density,*

$$f_I(x) = 2c\, e^{2cx}, \qquad x \leq 0.$$

The *approximate* density of the $\sup(\text{fBm},\ \mu < 0)$ is more complex [13],

$$f_{\text{sup,H}}(x) \propto x^{2(H-1/2)} e^{-2\mu x}, \quad x \geq 0, \quad \mu < 0,$$

and by simple symmetry, the *approximate* density of the $\inf(\text{fBm},\ \mu > 0)$ is,

$$f_{\text{inf,H}}(x) \propto |x|^{2(H-1/2)} e^{2\mu|x|}, \quad x \leq 0, \quad \mu > 0.$$

[1] *"fBm is a self-affine, non-Markovian, and translationally invariant generalization of Bm, depending on the Hurst exponent H"* [35]. This reference and others such as [36] and [37] only show perturbative expansions of the density, such as,

$$f_H(x) \approx 2|\mu| e^{-2|\mu||x|} |\mu x|^{2(H-1/2)} e^{\mathcal{O}(H-1/2)}.$$

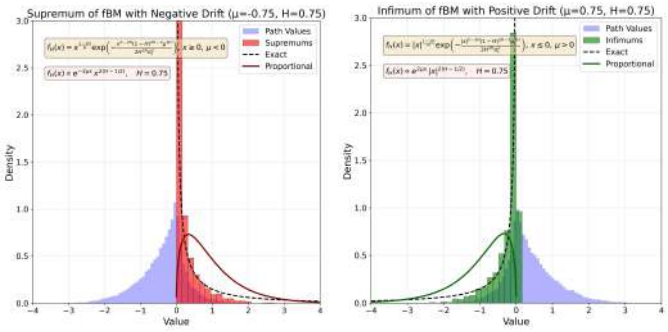
However, the *exact* density of the $\sup(\text{fBm},\ \mu < 0)$ is less well known [13] and is much more complex,

$$f_{\text{sup,H}}(x) = x^{\frac{1-2H}{H}} \exp\Big(-\frac{x^{2-2H}(1-H)^{2H-2}|\mu|^{2H}}{2H^{2H}\sigma_t^2}\Big), \quad x \geq 0, \quad \mu < 0, \quad (2)$$

and by simple symmetry, the *exact* density of the $\inf(\text{fBm},\ \mu > 0)$ is,

$$f_{\text{inf,H}}(x) = |x|^{\frac{1-2H}{H}} \exp\Big(-\frac{|x|^{2-2H}(1-H)^{2H-2}\mu^{2H}}{2H^{2H}\sigma_t^2}\Big), \quad x \leq 0, \quad \mu > 0. \quad (3)$$

These densities were plotted in Figure 7.



(a). **(b).**

Fig. 7: Path Densities
(a). Supremum of fBm paths with negative drift, $\mu = -0.75$.
(b). Infimum of fBm paths with positive drift, $\mu = 0.75$.

Figure 7 shows that by obtaining the correct expression for $\sup(\text{fBm},\ \mu < 0)$ in (2) and for $\inf(\text{fBm},\ \mu > 0)$ in (3), the density fits the histogram much more closely.

Up until this point, we have established that the density of sup(fBm) with negative drift and inf(fBm) with positive drift either lacks a known closed-form expression or, at best, is sufficiently obscure to place it at the frontier of current knowledge –a gap our work helps to address. Nevertheless, the supremum under drift is not, by itself, an adequate benchmark. A more intricate functional, built from the supremum, infimum, and the range of these, retains all the desirable properties while being substantially more challenging to estimate. This aligns with the benchmarking criteria outlined above. We therefore construct such a target functional to serve as the estimation benchmark.

C. Estimation Target Function

In the context of ML estimation, let L_B denote the length of a Build window and L_T denote the length of the Test window, as depicted in Figure 8.

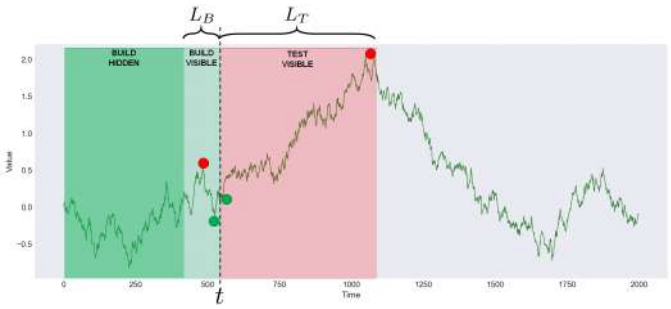


Fig. 8: Experimental Design
We define a Build/Train set based on the length L_B and a Test set based on the length L_T for the estimation of sup and inf of fBm.

Figure 8 shows a logical experimental design[2] that can be extended as required. However, a limitation of the current feature construction is that the raw Build window grows with path length, introducing a confound between architectural performance and input dimensionality. Future work can isolate this by fixing the feature vector to the three derived statistics (local volatility, mean(Build) and SD(Build)) across all data sizes.

For a Build fBm window B, and a Test fBm window T, let $\text{sup} = \max(T)$, $\text{inf} = \min(T)$, $\text{rng} = \text{sup} - \text{inf}$. Then, $\text{local\_vol} = \text{SD}(L_B)$. Define the target function $\mathscr{T}$ as,

$$\mathscr{T} = \frac{\log(1+|\text{rng}|)\cdot \text{sign}(\text{rng})\cdot|\text{rng}|^{\alpha}}{(1+\text{local\_vol})^{\beta}} + \gamma\cdot\text{sign}(\text{sup})\sqrt{|\,\text{sup}\cdot\text{inf}\,|}. \quad (4)$$

In this paper, we set $\alpha = 1.2$, $\beta = 0.7$, $\gamma = 0.05$ and $\epsilon = 0.01$, but these can be varied. It must be noted that there is an infinite number of such possible non-linear combinations of sup and inf of fBm data, not just (4).

This target function is difficult for the ML algorithms due to a number of reasons:

1) The target function is an extreme-statistic ($\max - \min$) over a Test horizon; such extremes are path-dependent and not well determined by a short Build window.
2) Using a long Test window increases randomness of the extreme over the Test, reducing predictable structure.
3) The fBm Hurst parameter and long-memory behavior make the relationship non-trivial and non-linear.
4) The featurisation intentionally provides only a few Build samples (and simple derived features), so models have little information to reconstruct Test extremes (i.e. L_B = Build_len and L_T = Test_len, and $L_B/L_T \ll 1$ for some time t).

[2]For a fBm path with Hurst exponent H and test window L_T, the autocorrelation between observations separated by lag k decays as, $\rho(k) = \frac{1}{2}\left(|k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H}\right)$, which for $H = 0.7$ and $k = L_T = 12$ evaluates to $\rho(12) \approx 0.03$. This is negligible at the scale of the validation–Out Of Sample (OOS) gap, confirming that leakage through the autocorrelation structure is not a material concern for the chosen window lengths. Furthermore, since each training observation is a *independently drawn fBm path* rather than a single realisation of one continuous process, the windows are path-exchangeable conditional on H. Random cross-validation folds are therefore asymptotically equivalent to walk-forward folds in this setting, and the cross validation (CV) scores carry no optimistic bias beyond that introduced by finite sample size.

A motivation and intuition for what the density of $\mathscr{T}$ looks like is provided in Figure 9.

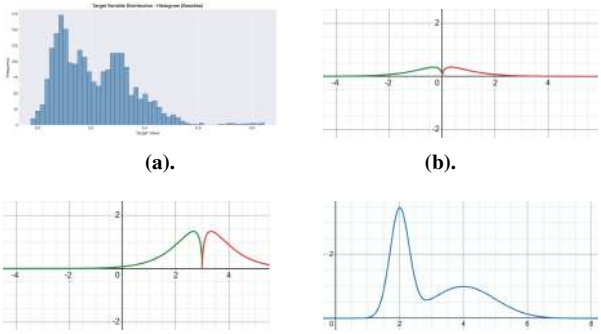


Fig. 9: Bimodal Distribution Motivation
(a). The actual distribution of $\mathscr{T}$ apears to be bi-modal.
(b). In general terms, we have the density of the $\text{sup}(\text{fBm}), \mu < 0$ in red superimposed with the density of the $\text{inf}(\text{fBm}), \mu > 0$ in green.
(c). We can then apply a horizontal translation to the right and a vertical dilation.
(d). We then have a rough additional explanation for the motivation behind why the density of $\mathscr{T}$ is difficult to estimate.

Figure 9 provides intuition for why the target function constitutes a challenging benchmark. Readers interested in a more rigorous treatment are referred to Appendix §VII-E.

The pseudo code of how the target function $\mathscr{T}$ was implemented is listed below in Algorithm 1.

Algorithm 1 Target Function $\mathscr{T}$ Implementation Algorithm

Initialize empty lists $X = [\,]$, $y = [\,]$.
For each index i from Build_len to [len(series) - Test_len] in steps:
a. Extract_Build = series[i - Build_len : i].
b. Extract_Test = series[i : i + Test_len].
c. Compute Test extrema:
$\text{sup} = \max(\text{Test})$
$\text{inf} = \min(\text{Test})$
$\text{rng} = \text{sup} - \text{inf}$
d. Compute local_vol = SD(Build) (i.e. recent volatility).
e. Compute target_value $\mathscr{T}$:
$$\mathscr{T} = \frac{\log(1+|\text{rng}|)\cdot \text{sign}(\text{rng})\cdot|\text{rng}|^{\alpha}}{(1+\text{local\_vol})^{\beta}} + \gamma\cdot\text{sign}(\text{sup})\sqrt{|\,\text{sup}\cdot\text{inf}\,|}$$
f. Construct feature vector:
feats = concatenate(Build, [local_vol, mean(Build), SD(Build)])

IV. Results

We create five scenarios, each with different yet subtle differences, but focus our scientific lense on the first two, as a way to demonstrate the complexity of the target function, as shown in Table III.

Table III presents an empirical analysis of how various ML models approximate a nonlinear path-functional of fBm. This functional is derived from the extreme-value statistics of simulated sample paths. Rather than predicting the supremum directly, the models are tasked with estimating a complex nonlinear transformation of the path –a target that depends on the relative behavior of the supremum, infimum, and range over future segments.

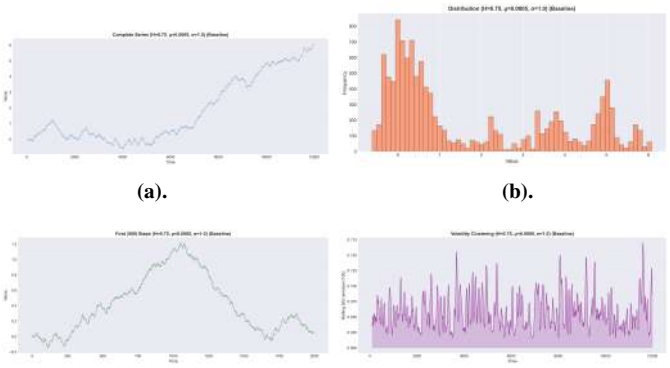
TABLE III: Parameter settings and qualitative characteristics for the five scenarios.

Scenario	H	μ	σ	L_B	L_T	Description
Scenario 1: High Persistence	0.75	0.0005	1.0	25	100	Smooth, strongly persistent paths with mild drift and low noise.
Scenario 2: Rough Paths	0.55	0.0001	1.5	5	400	Highly jagged, noise-dominated paths with minimal drift and low persistence.
Scenario 3: Strong Drift	0.70	0.0035	1.0	8	250	Moderately persistent trajectories dominated by a strong upward linear drift.
Scenario 4: Standard Brownian	0.50	0.0000	1.0	8	250	Classical Brownian motion with zero drift and unit volatility.
Scenario 5: Moderate	0.65	0.0003	0.8	15	150	Moderately smooth paths with mild noise, weak drift, and partial mean reversion.

These first two scenarios will be analysed via four experiments.

A. Experiment 1 of 4: Baseline 'High Persistance' Scenario

The first experiment examines the 'Baseline' or 'High Persistance' scenario, in which smooth, strongly persistent paths with mild drift and low noise parameters create an environment in which the six models are benchmarked against each other, as shown in Figure 10.



(a). **(b).** **(c).** **(d).**

Fig. 10: Scenario 1 of 2: Baseline Scenario - Diagnostics
(a). 12,000 time steps.
(b). Distribution of target function.
(c). 2,000 time steps.
(d). Volatility Clustering: The conditional variance is itself autocorrelated, producing bursts of elevated rolling standard deviation interspersed with extended calm intervals.

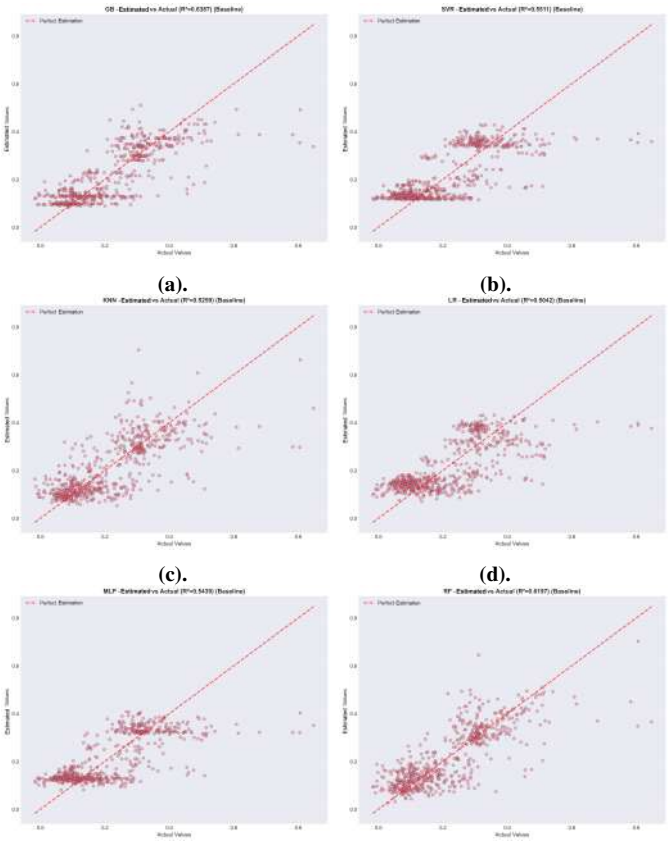
Figure 10 shows some high-level properties of the data. The estimation of the target function using this data is examined and shown in Figure 11.

Figure 11 shows that all six models exhibit some level of difficulty in estimating the target function. To analyse this further, the error residuals are shown in Figure 12.

Figure 12 further shows that the underlying distribution is bimodal, where the most sparse data is seen in (a). for Gradient boosting (GB). The differences are examined further via cross validation (CV) analysis, as shown in Figure 13.

Figure 13 shows that the performance of the models against the target function, as measured by R^2 is reasonable given the complexity of the objective function, and that the differences or deltas are quite low. This is made even more crystal clear by comprehensive statistical metrics, as shown in Table IV.

Table IV demonstrates, that whilst the model performance is not terribly high, it does imply that the models have



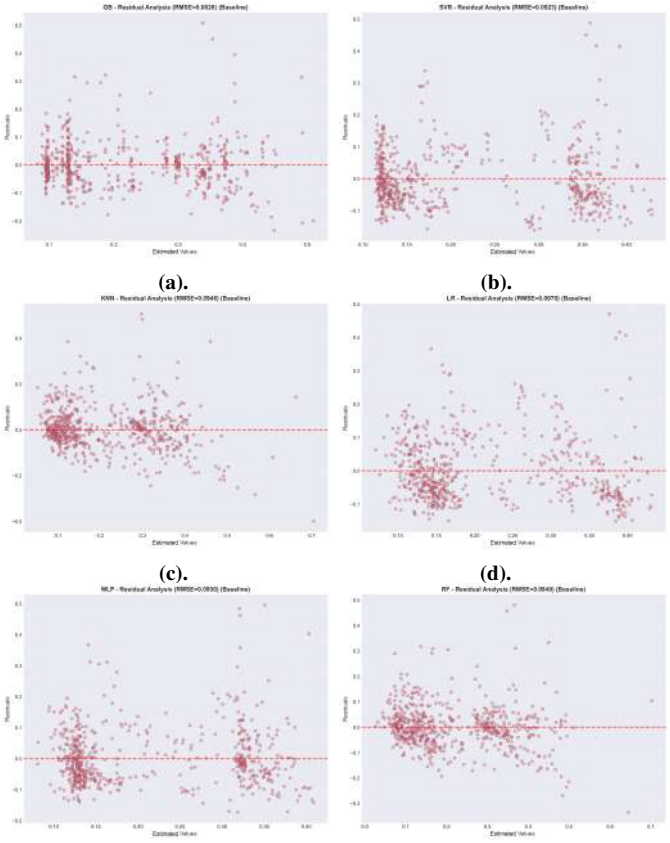
(a). **(b).** **(c).** **(d).** **(e).** **(f).**

Fig. 11: Scenario 1 of 2: Baseline Scenario - Estimation
(a). GB, (b). SVR, (c). KNN, (d). LR, (e). MLP, (f). RF

TABLE IV: Scenario 1 of 2: Baseline Scenario — Model Performance Comparison

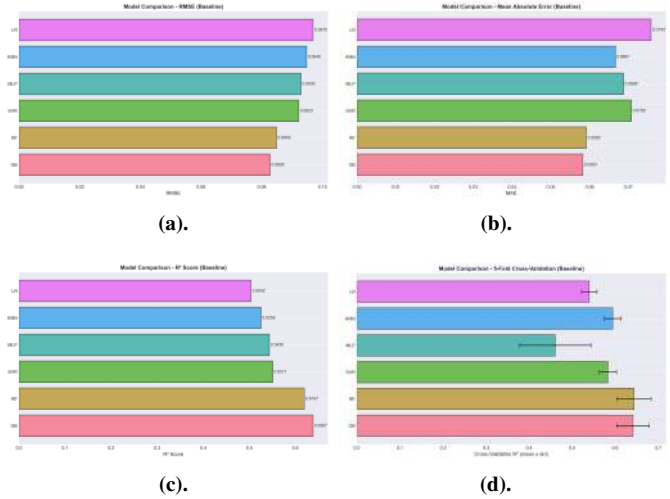
Rank	Model	R^2	RMSE	MAE	CV_Mean	CV_SD
0	LR	0.5042	0.0970	0.0759	0.5401	0.0180
1	RF	0.6197	0.0849	0.0592	0.6441	0.0396
2	GB	0.6387	0.0828	0.0583	0.6418	0.0379
3	SVR	0.5511	0.0923	0.0708	0.5834	0.0198
4	KNN	0.5259	0.0948	0.0667	0.5944	0.0201
5	MLP	0.5439	0.0930	0.0688	0.4609	0.0828

the possibility of performing well if additional analysis is undertaken with the data of this scenario. This experiment establishes the so-called 'baseline', and the next experiment forms our recommended scenario, under which all models perform poorly –which is the benefit of such a ML benchmark.



(a). **(b).** **(c).** **(d).** **(e).** **(f).**

Fig. 12: Scenario 1 of 2: Baseline Scenario - Residuals
(a). GB, (b). SVR, (c). KNN, (d). LR, (e). MLP, (f). RF

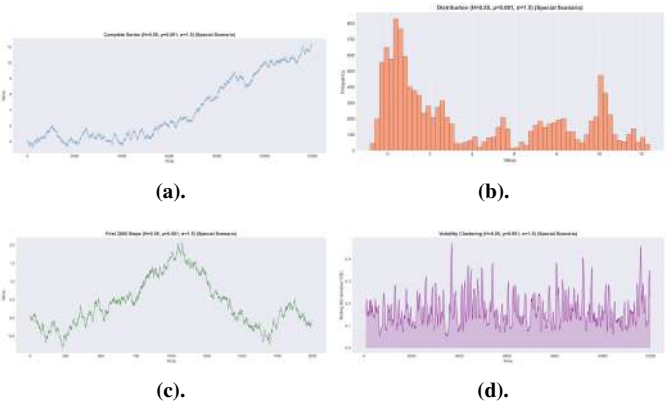


(a). **(b).** **(c).** **(d).**

Fig. 13: Scenario 1 of 2: Baseline Scenario - Performance
(a). RMSE: Gradient Boosting Regressor achieved the lowest RMSE, indicating the smallest magnitude of large prediction errors among the baseline models.
(b). MAE: Gradient Boosting Regressor produced the lowest MAE, reflecting the best average absolute prediction accuracy.
(c). R^2: Gradient Boosting Regressor obtained the highest R^2 score, explaining the greatest proportion of variance in the data.
(d). CV: Gradient Boosting Regressor delivered the highest mean 5-fold cross-validation R^2 with strong stability, confirming robust generalization performance.

B. Experiment 2 of 4: Special Case 'Rough Paths' Scenario

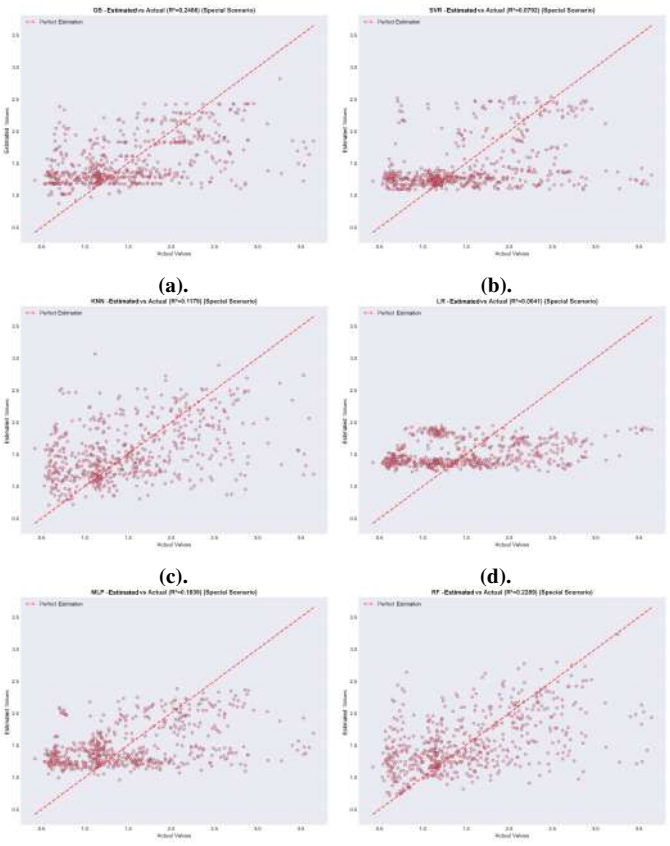
The second experiment examines the 'Special Case' or 'Rough Paths' scenario, in which highly jagged, noise-dominated paths with minimal drift and low persistence parameters create an environment in which the six models are benchmarked against each other. In this scenario, we expect to see much poorer model performance, although it is not obvious just by looking at the paths themselves, as shown in Figure 14.



(a). **(b).** **(c).** **(d).**

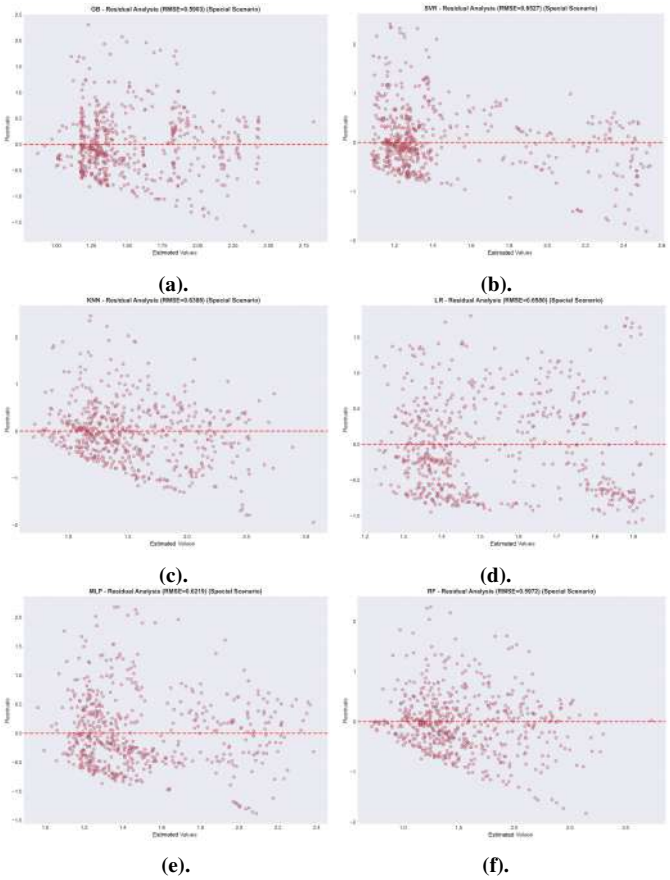
Fig. 14: Scenario 2 of 2: Special - Diagnostics
(a). 12,000 time steps.
(b). Distribution of target function.
(c). 2,000 time steps.
(d). Volatility Clustering: The conditional variance is itself auto-correlated, producing bursts of elevated rolling standard deviation interspersed with extended calm intervals.

Figure 14 shows some high-level properties of the data. The estimation of the target function using this data is examined and shown in Figure 15.



(a). **(b).** **(c).** **(d).** **(e).** **(f).**

Fig. 15: Scenario 2 of 2: Special - Estimation
(a). GB, (b). SVR, (c). KNN, (d). LR, (e). MLP, (f). RF

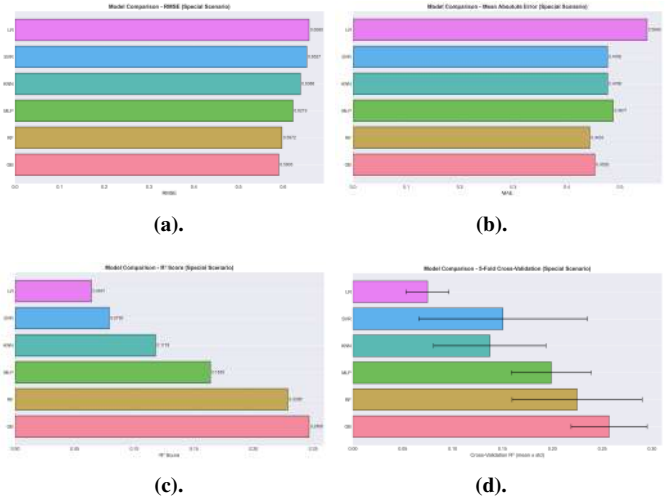


(a). **(b).** **(c).** **(d).** **(e).** **(f).**

Fig. 16: Scenario 2 of 2: Special - Residuals
(a). GB, (b). SVR, (c). KNN, (d). LR, (e). MLP, (f). RF

Figure 15 shows that all six models exhibit *great* levels of difficulty in estimating the target function. To analyse this further, the error residuals are shown in Figure 16.

Figure 16 further shows that all the distributions of results are now quite random and dispersed. The differences are examined further via cross validation (CV) analysis, as shown in Figure 17.



(a). **(b).** **(c).** **(d).**

Fig. 17: Scenario 2 of 2: Special - Performance
(a). RMSE: Gradient Boosting Regressor achieved the lowest RMSE in the special scenario, indicating the smallest overall prediction error.
(b). MAE: Random Forest Regressor produced the lowest MAE, reflecting the best average absolute prediction accuracy.
(c). R^2: Gradient Boosting Regressor obtained the highest R^2 score, explaining the greatest proportion of variance under the special scenario.
(d). CV: Gradient Boosting Regressor delivered the highest mean 5-fold cross-validation R^2, demonstrating the strongest generalization performance.

Figure 17 shows that the performance of the models against the target function, as measured by R^2 is now very poor due to the complexity of the objective function, and that the differences or deltas are quite high. This is made even more crystal clear by comprehensive statistical metrics, as shown in Table V.

TABLE V: Scenario 2 of 2: Special — Model Performance Comparison

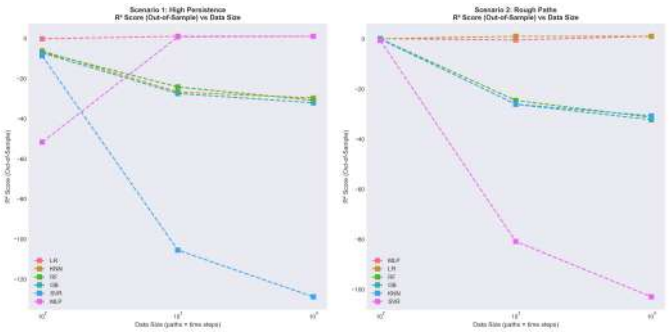
Rank	Model	R^2	RMSE	MAE	CV_Mean	CV_SD
0	LR	0.0641	0.6580	0.5506	0.0748	0.0216
1	RF	0.2289	0.5972	0.4433	0.2249	0.0652
2	GB	0.2466	0.5903	0.4529	0.2569	0.0381
3	SVR	0.0792	0.6527	0.4766	0.1506	0.0845
4	KNN	0.1179	0.6388	0.4768	0.1374	0.0563
5	MLP	0.1639	0.6219	0.4871	0.1990	0.0400

Table V shows that, in this scenario, all models exhibit very weak performance. This experiment intentionally constructs a highly challenging setting in which established methods struggle, thereby defining the benchmark case of interest. Our aim is to design a problem sufficiently complex to stress-test these algorithms, ensuring that any future improvements can be meaningfully evaluated against this rigorous baseline. To help understand the underlying dynamics of this benchmark,

an additional experiment was undertaken which examines data volumetric analysis.

C. Experiment 3 of 4: Data Size Scaling

To investigate the impact of data size on model performance under this complex target function environment, the size in terms of both the number of paths and also the number of time steps was varied, and is shown in Figure 18.



(a). **(b).**

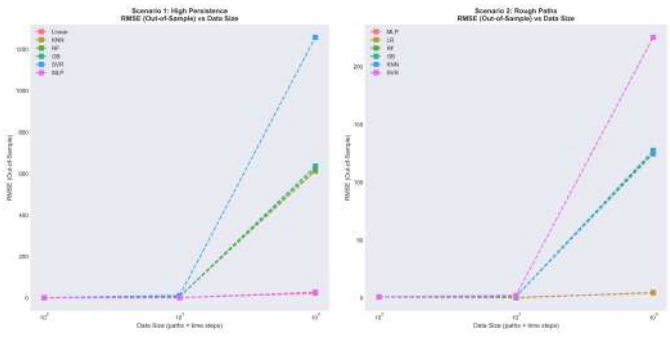
Fig. 18: R^2 Comparison Plots

(a). Scenario 1 - High Persistence: While most models start near $R^2 = 0$, only the MLP improves with increasing data (converging toward 0), whereas SVR, GB, RF, and KNN deteriorate substantially, with SVR collapsing below 120.
(b). Scenario 2 - Rough Paths: All models begin near $R^2 = 0$, but as data increases most performance collapses —especially SVR (to about 100), while only LR and MLP remain stable around zero.

Figure 18 shows how model performance (R^2) deteriorates with data size under two dynamical regimes.

- **(a). Scenario 1: High Persistence**
  Initial Performance: At the smallest data size (10^2), most models start with R^2 scores near zero or slightly negative, with the exception of the MLP (pink) which starts extremely low (around -50). Scaling Behaviour: As data size increases to 10^4, the MLP is the only model that improves, eventually converging near $R^2 \approx 0$. Failure to Converge: Most other models —specifically SVR (blue), GB (teal), RF (green), and KNN (gold) —show decreasing performance as data increases, with SVR dropping significantly to below -120.
- **(b). Scenario 2: Rough Paths**
  Initial Performance: All models begin with R^2 scores near zero at the 10^2 data size. Scaling Behaviour: Instead of improving to 0.95, performance for most models collapses as data size grows. Model Divergence: The SVR (magenta) performs the worst, plummeting to an R^2 near -100 at 10^4. LR (gold) and MLP (red) are the only models that maintain a stable performance near $R^2 \approx 0$.

To examine these findings further, we analyse the error residuals in Figure 19.



(a). **(b).**

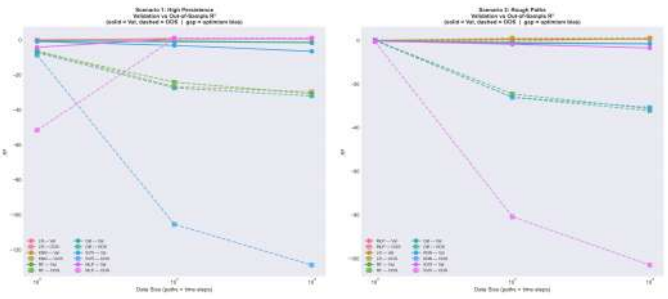
Fig. 19: RMSE Comparison Plots

(a). In Scenario 1, the models diverge sharply at the 10^4 scale, with the MLP demonstrating superior precision (RMSE ≈ 25) while the SVR's error spikes to approximately 1250.
(b). Conversely, Scenario 2 shows that the MLP struggles most with "Rough Paths" at the largest data size (RMSE ≈ 225), leaving LR as the most robust choice with a minimal RMSE of roughly 5.

Figure 19 shows how the RMSE trends across data sizes for the two dynamical regimes.

- **(a). Scenario 1: High Persistence**
  The RMSE remains negligible for small sample sizes (10^2 to 10^3) but exhibits significant divergence as the dataset grows to 10^4. Model architecture plays a critical role at this scale: SVR shows a substantial spike in error, reaching an RMSE ≈ 1250. MLP demonstrates superior performance, maintaining a very low RMSE ≈ 25. LR, KNN, RF, and GB form a mid-tier cluster with RMSE values ranging between 600 and 650. The increase in absolute error across most models likely reflects the scaling of the target's dynamic range in the higher-datasize experiments. However, the wide gap between MLP and SVR suggests that for high-persistence processes, the choice of estimator is vital for maintaining predictive precision.
- **(b). Scenario 2: Rough Paths**
  A similar scaling trend is observed, though the magnitude of RMSE is lower overall due to the reduced amplitude of the underlying trajectories. While performance is uniform at small scales, the models diverge noticeably at 10^4 data points: MLP performs the poorest in this scenario, with its RMSE climbing to ≈ 225. LR maintains the highest accuracy, with an RMSE remaining near 5. GB, RF, KNN, and SVR cluster around an RMSE ≈ 125. This suggests that while the "difficulty" of the task is influenced by the statistical properties of the "Rough Paths" process, certain architectures (like LR) are significantly more robust to this specific type of volatility than others (like MLP).

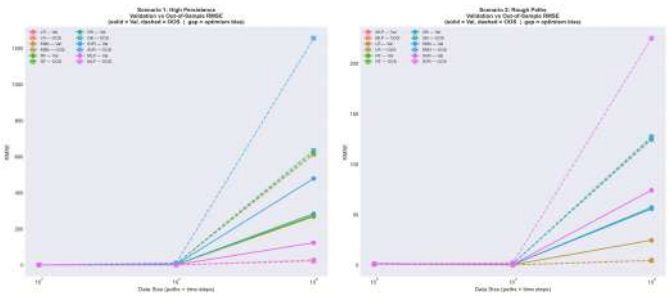
We now undertake additional analysis so that the differences between the two secarios becomes even more clear, as shown in Figure 20.



(a). (b).

Fig. 20: Target Complexity Plots 1

(a). In Scenario 1, a high optimism bias causes most models' R^2 to drop significantly below zero as data size increases, leaving the MLP as the only architecture to show relative improvement by converging toward $R^2 \approx 0$.
(b). In Scenario 2, the irregular process structure triggers a total collapse in generalization for several models, whereas LR and MLP remain relatively stable near $R^2 \approx 0$ even as the dataset expands.



(a). (b).

Fig. 21: Target Complexity Plots 2

(a). The target standard deviation increases far more rapidly in Scenario 1 than in Scenario 2, reflecting stronger amplitude and persistence effects. Accordingly, RMSE in Scenario 1 reaches values near 1200, compared to roughly 200 in Scenario 2 as data size grows.
(b). The target range expands substantially with data size in Scenario 1 but remains relatively constrained in Scenario 2, indicating different variability regimes. The widening gap between out-of-sample RMSE (dashed) and validation RMSE (solid) shows that larger target variability amplifies generalization errors in absolute terms.

Figure 20 highlights how the two scenarios differ in the basic properties of their target data as the dataset grows.

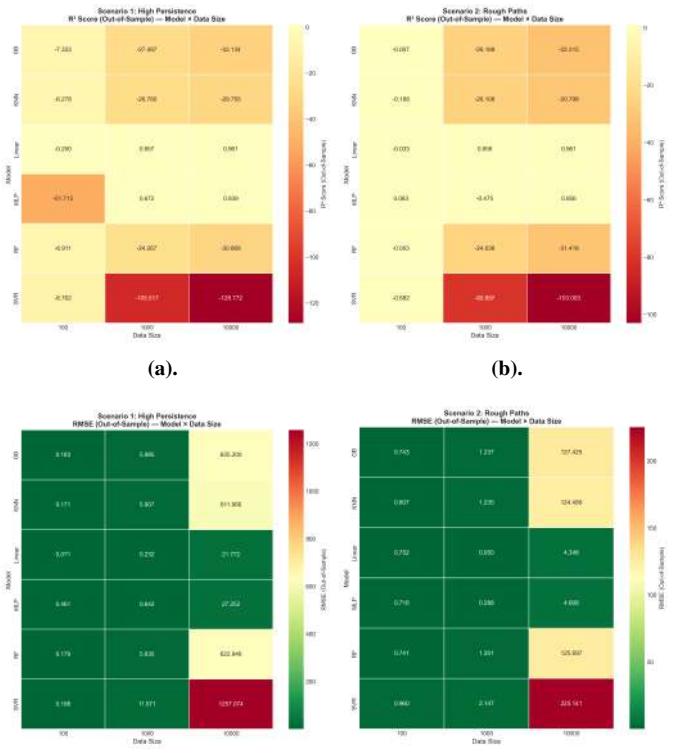
- **(a). Scenario 1 (High Persistence):** most models exhibit a sharp decline in out-of-sample performance as data size increases, with R^2 values dropping significantly below zero. The large gap between validation and out-of-sample R^2 (solid vs. dashed lines) reveals a high optimism bias, suggesting that models fail to capture the temporal coherence of the dynamics. Only the MLP shows a relative improvement, converging toward $R^2 \approx 0$.
- **(b). In Scenario 2 (Rough Paths):** the irregular structure leads to a total collapse in generalization for several models. While LR and MLP remain relatively stable near $R^2 \approx 0$, the SVR, KNN, RF, and GB models show diverging negative R^2 values as the dataset grows, indicating that the noise-to-signal ratio prevents these architectures from learning a predictive mapping.

Together, these plots make it clear why the models behave so differently across the scenarios: they are learning from target distributions with fundamentally different levels of difficulty. We now undertake additional analysis so that the differences between the two scenarios becomes even more clear, by examining RMSE, as shown in Figure 21.

Figure 21 supports the results of Figure 20 that increased data volumes leads to greater errors for these ML algorithms.

- (a). The target standard deviation grows much more rapidly in Scenario 1 than in Scenario 2, reflecting the larger amplitude and stronger temporal coherence of the high-persistence dynamics. This is evidenced by the RMSE values in Scenario 1 reaching scales an order of magnitude larger (up to 1200) compared to Scenario 2 (up to 200) as the data size increases.
- (b). The target range expands dramatically with data size in Scenario 1, whereas Scenario 2 remains comparatively constrained, indicating fundamentally different variability scales between the two regimes. The sharp divergence of out-of-sample RMSE (dashed lines) from validation RMSE (solid lines) in both cases highlights that as the target variability scales up, the models' inability to generalize becomes increasingly penalized in absolute terms.

To obtain a clearer understanding of the estimation of the target function, we take a closer look at R^2, as shown in Figure 22.



(a). **(b).** **(c).** **(d).**

Fig. 22: Heatmap Plots 1

(a) LR and MLP achieve high R^2 scores ($\approx$ 0.95), while other models fail with increasingly negative performance as data scales. (b) LR and MLP reach a strong 0.96 R^2, whereas all other architectures collapse, with SVR dropping to -103. (c) RMSE explodes for most models (up to 1257), but LR and MLP maintain significantly higher precision with errors below 28. (d) RMSE rises with data size, yet LR and MLP remain the most robust with minimal errors near 4.

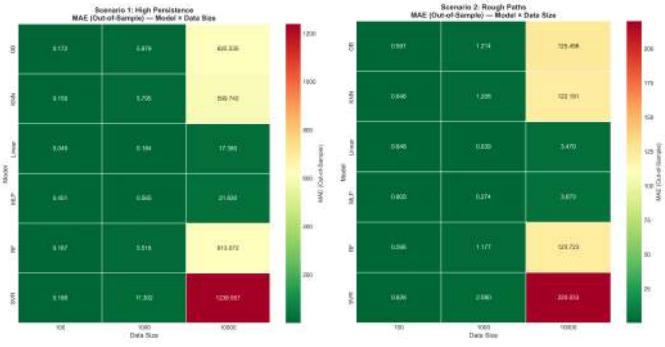
Figure 22 shows the following observations across both scenarios.

- **(a). Scenario 1: R^2 Score**
  The majority of models (GB, KNN, RF, SVR) fail to capture the dynamics, with R^2 scores becoming increasingly negative as data size grows; only the LR and MLP models achieve strong predictive accuracy, converging to $R^2 \approx 0.96$ and $R^2 \approx 0.94$ respectively at 10^4 samples.
- **(b). Scenario 2: R^2 Score**
  Models do not converge to similar high scores; while LR and MLP successfully reach $R^2 \approx 0.96$, the other architectures exhibit a total collapse in generalization with R^2 values dropping as low as -103 for SVR.
- **(c). Scenario 1: RMSE**
  RMSE increases with data size for all models, but there is a massive disparity in performance: SVR, GB, KNN, and RF see errors explode to values between 611 and 1257, whereas LR and MLP maintain much higher precision with errors limited to $\approx$ 22–27.
- **(d). Scenario 2: RMSE**
  RMSE actually starts very low (< 1) at small data sizes and increases as the dataset expands, though the absolute error levels remain lower than in Scenario 1, with the LR and MLP models proving most robust at RMSE ≈ 4.

These observations give rise to the following generalized characteristics of our complex target function.

- **Non-Linear Error Growth**: As data size increases by factors of 10, the RMSE does grow non-linearly —and for models like SVR, exponentially —jumping from 11.5 to 1257.0 in Scenario 1.
- **The Accuracy Split**: While most models (SVR, KNN, RF, GB) do become increasingly "confused", with their accuracy (R^2) collapsing into deep negatives as the dataset scales, the LR and MLP models actually break this trend by finding the signal and reaching high accuracy (≈ 0.95) at the largest data size.
- **Dataset Difficulty**: This confirms that while the stochastic nature of the data makes estimation difficult for traditional non-linear models, the "confusion" is architecture-specific rather than a total failure of predictability.

This disproportionate error growth indicates that the target function is intrinsically difficult to learn and becomes increasingly intractable as more data are presented to the models. This is further supported in Figure 23.



(a). **(b).**

Fig. 23: Heatmap Plots 2

(a). Scenario 1 (High Persistence): High persistence yields significantly larger absolute errors than Scenario 2, with failing models exceeding MAE $= 600$ at 10^4. (b). Scenario 2 (Rough Paths): The LR model maintains superior precision (MAE $= 3.470$), whereas other architectures show exponential error jumps at 10^4 due to stochastic noise confusion.

Figure 23 completes the analysis first depicted in Figure 22, which results in the same colour outcomes, but for different error metrics.

- **Scenario 1 - High Persistence:** The high persistence of the target results in significantly larger absolute errors compared to Scenario 2, with MAE values for failing models exceeding 600 at the largest scale.
- **Scenario 2 - Rough Paths:** While error increases with data size for all models, the LR model maintains the highest precision with an MAE of 3.470. The sharp jump in MAE between 10^3 and 10^4 for SVR, GB, KNN, and RF suggests these models become increasingly confused by the noise-to-signal ratio of the stochastic process.

We can obtain even more clarity from investigating the statistical metrics that are detailed in Table VI.

TABLE VI: OOS performance comparison: High Persistence vs Rough Paths

Size	Model	High Persistence			Rough Paths		
		R^2	RMSE	MAE	R^2	RMSE	MAE
100×100							
	LR	−0.250	0.071	0.045	−0.033	0.752	0.648
	RF	−6.911	0.179	0.167	−0.003	0.741	0.596
	GB	−7.323	0.183	0.172	−0.007	0.743	0.591
	KNN	−6.278	0.171	0.159	−0.188	0.807	0.646
	MLP	−51.715	0.461	0.451	0.063	0.716	0.610
	SVR	−8.762	0.199	0.188	−0.682	0.960	0.826
1,000×1,000							
	LR	0.957	0.232	0.184	0.956	0.050	0.040
	RF	−24.267	5.640	5.520	−24.636	1.200	1.180
	GB	−27.497	5.980	5.880	−26.168	1.240	1.210
	KNN	−26.760	5.910	5.800	−26.106	1.240	1.210
	MLP	0.672	0.642	0.565	−0.475	0.288	0.274
	SVR	−105.517	11.570	11.300	−80.897	2.150	2.080
10,000×10,000							
	LR	0.961	21.7700	17.3800	0.961	4.3500	3.4700
	RF	−30.868	6.23e2	6.13e2	−31.418	1.26e2	1.24e2
	GB	−32.134	6.35e2	6.26e2	−32.315	1.27e2	1.25e2
	KNN	−29.755	6.12e2	6.00e2	−30.798	1.24e2	1.22e2
	MLP	0.939	27.2500	21.8300	0.956	4.6100	3.6700
	SVR	−128.772	1.26e3	1.24e3	−103.003	2.25e2	2.20e2

Table VI demonstrates a sharp divergence in model performance as data size scales; while the LR and MLP models converge to high accuracy ($R^2 \approx 0.96$), the other architectures (SVR, KNN, RF, GB) suffer a total generalization collapse. Although Scenario 1 exhibits much larger absolute error magnitudes (RMSE and MAE) due to high-persistence targets, both scenarios show that failing models become increasingly "confused" by the stochastic noise as the dataset grows. At the largest scale, the Linear model remains the most robust across both regimes, maintaining the lowest error and highest R^2.

We finally derive additional comprehensive statistical metrics, as shown in Table VII.

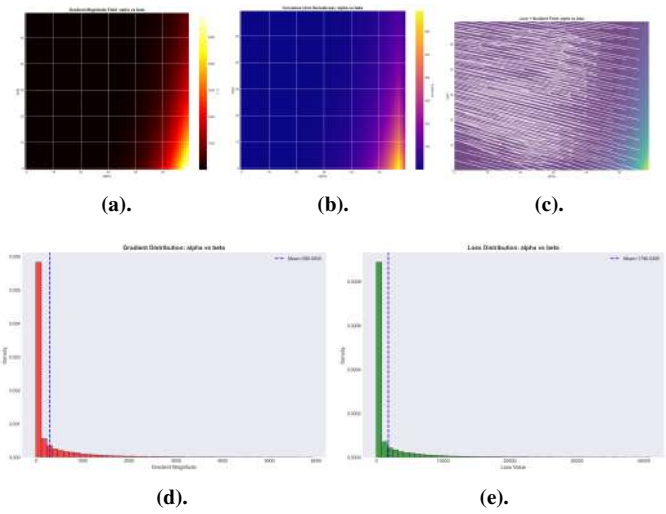
TABLE VII: Scenario Parameters and Summary Statistics

Parameter	**Scenario 1**	**Scenario 2**
Hurst Exponent (H)	0.75	0.55
Drift (μ)	0.0005	0.0001
Volatility (σ)	1.0	1.5
Past Length (L_B)	25	5
Future Length (L_T)	100	400
Size 100		
Mean R^2_{OOS}	-13.540 ± 18.932	-0.141 ± 0.278
Mean RMSE	0.211	0.787
Mean MAE	0.197	0.653
Samples	4,188	4,048
Size 1,000		
Mean R^2_{OOS}	-30.402 ± 39.117	-26.221 ± 29.643
Mean RMSE	4.995	1.027
Mean MAE	4.874	0.999
Samples	16,998	16,992
Size 10,000		
Mean R^2_{OOS}	-36.605 ± 47.782	-32.603 ± 37.984
Mean RMSE	529.368	101.951
Mean MAE	519.419	99.765
Samples	170,000	170,000

Table VII demonstrates that the scenarios are fundamentally different: Scenario 1 is characterized by high persistence ($H = 0.75$) and rapidly expanding target variance, while Scenario 2 is rougher and requires a much longer forecasting horizon. At small data sizes, the irregular nature of Scenario 2 results in poor initialization (negative Mean R^2), whereas Scenario 1 is initially more predictable. However, as data scales, the rising standard deviation of the R^2 values in both scenarios reveals a widening performance gap between architectures, where most models become increasingly confused by the stochastic noise. This results in an explosion of RMSE in Scenario 1 (averaging 529.368) that far outstrips the more contained error growth in Scenario 2, directly reflecting the impact of the different Hurst exponents and memory lengths on predictive stability.

D. Experiment 4 of 4: LLS of Complex Target Function Results

Having established the convex properties of sup(fBm) landscapes in RKHS, we now examine a complex target function involving three parameters (α, β, γ) to characterize optimization difficulty across different parameter interactions. The analysis employs multiple geometric and statistical metrics to quantify landscape complexity, as shown in Figure 24.

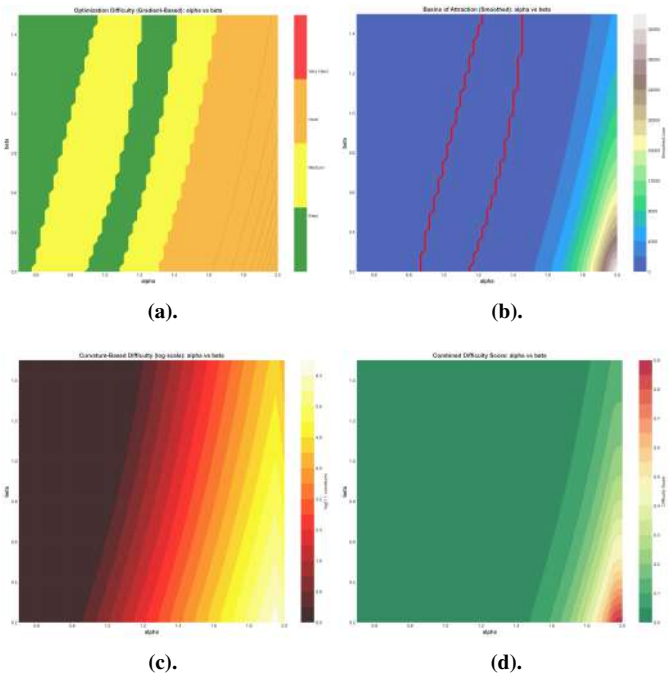


(a). **(b).** **(c).**

(d). **(e).**

Fig. 24: Complexity of α vs β
(a). Gradient magnitude field.
(b). Curvature of gradient magnitude field.
(c). Loss + Gradient field.
(d). Gradient distribution.
(e). Loss distribution.

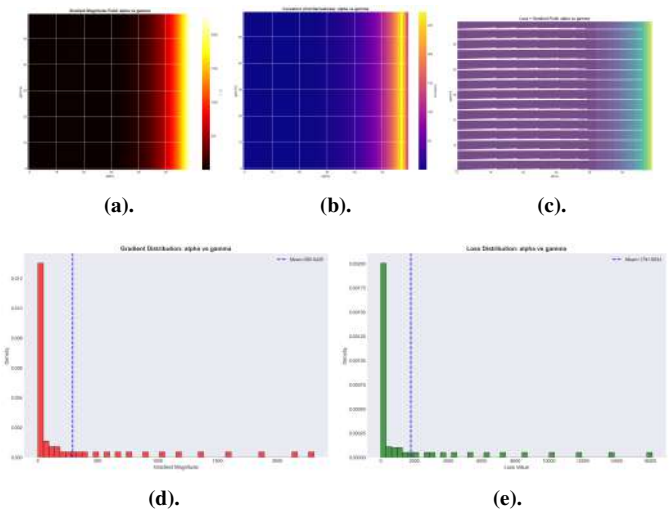
Figure 24 reveals extreme gradient concentration at high $\alpha - \beta$ values with catastrophic loss distributions exhibiting mean of 1798 and range extending to 40,000. The gradient magnitude field shows explosive growth in the upper-right region while the majority of parameter space exhibits near-zero gradients. The difficulty of this function estimation is shown in Figure 25.



(a). **(b).** **(c).** **(d).**

Fig. 25: Difficulty of α vs β
(a). Optimization difficulty.
(b). Basins of attraction (smoothed).
(c). Curvature-based difficulty (log-scale).
(d). Combined difficulty score.

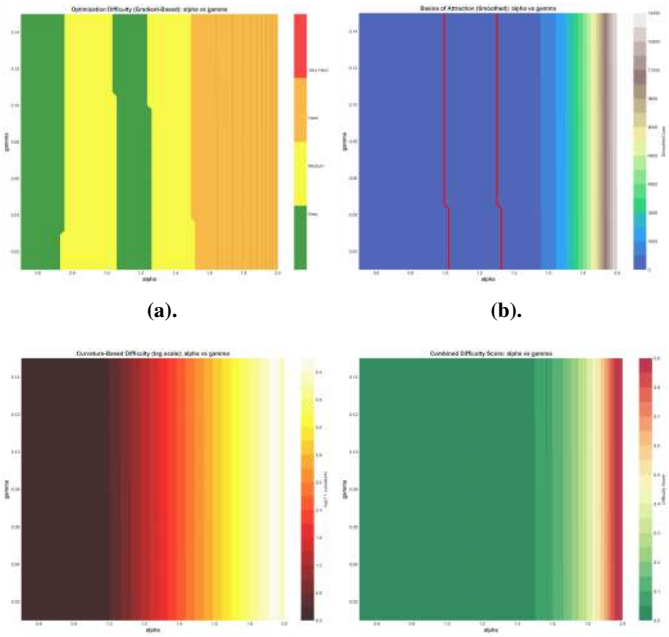
Figure 25 quantifies optimization difficulty through four complementary metrics, confirming $\alpha - \beta$ as the most challenging parameter interaction with combined difficulty scores approaching 0.9 in pathological regions. We next examine how γ interacts with α to determine whether similar complexity patterns emerge, as shown in Figure 26.



(a). **(b).** **(c).** **(d).** **(e).**

Fig. 26: Complexity of α vs γ
(a). Gradient magnitude field.
(b). Curvature of gradient magnitude field.
(c). Loss + Gradient field.
(d). Gradient distribution.
(e). Loss distribution.

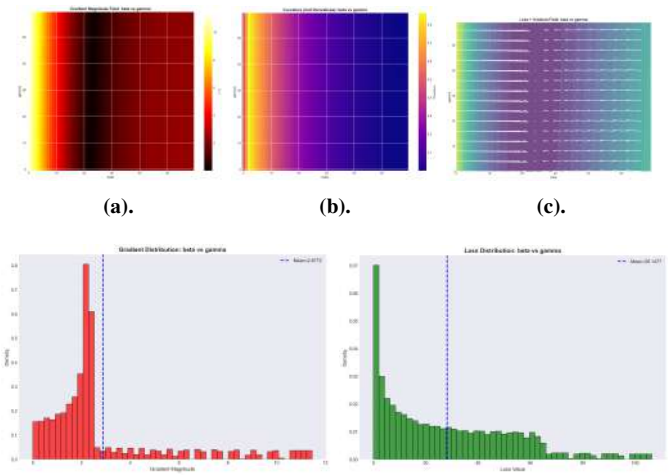
Figure 26 demonstrates γ-invariant gradient pathologies with vertical concentration patterns, compressed magnitude ranges $(0 - 2000)$, and loss distributions nearly identical to $\alpha - \beta$ (mean: 1791.68). The γ parameter shows minimal geometric influence compared to β. The difficulty of this function estimation is shown in Figure 27.



(a). **(b).** **(c).** **(d).**

Fig. 27: Difficulty of α vs γ
(a). Optimization difficulty.
(b). Basins of attraction (smoothed).
(c). Curvature-based difficulty (log-scale).
(d). Combined difficulty score.

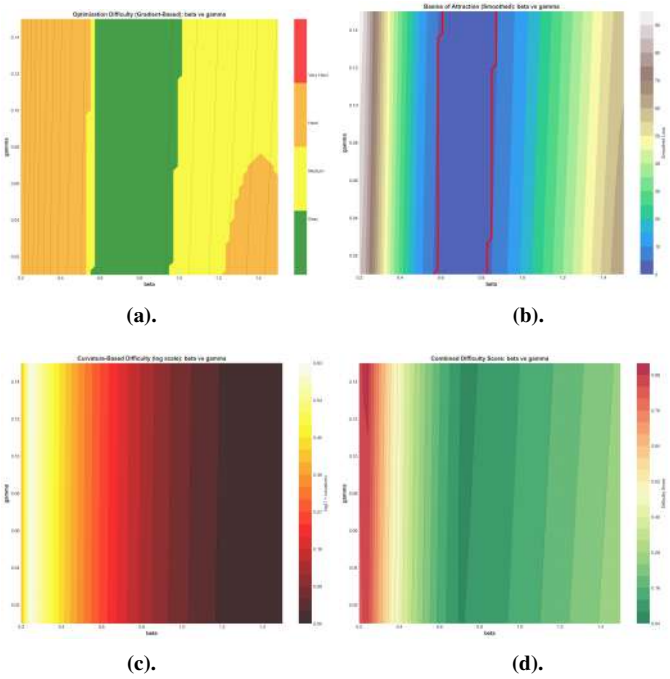
Figure 27 confirms α dominance through vertical difficulty boundaries and threshold-triggered pathologies at $\alpha \approx 1.6 - 1.8$, independent of γ settings. The final parameter interaction examines $\beta - \gamma$ behavior in α's absence to isolate secondary parameter dynamics, as shown in Figure 28.



(a). **(b).** **(c).** **(d).** **(e).**

Fig. 28: Complexity of β vs γ
(a). Gradient magnitude field.
(b). Curvature of gradient magnitude field.
(c). Loss + Gradient field.
(d). Gradient distribution.
(e). Loss distribution.

Figure 28 reveals inverse regularization dynamics with left-edge gradient concentration (magnitude 10 at low β) and dramatically compressed ranges across all metrics. The loss distribution shows mean of 28.14 with uniform-like structure, contrasting sharply with α-dominated spaces. The difficulty of this function estimation is shown in Figure 29.



(a). (b). (c). (d).

Fig. 29: Difficulty of β vs γ
(a). Optimization difficulty.
(b). Basins of attraction (smoothed).
(c). Curvature-based difficulty (log-scale).
(d). Combined difficulty score.

Figure 29 demonstrates $\beta - \gamma$ space's benign characteristics with difficulty scores below 0.4 across most regions and complex two-dimensional flow patterns indicating genuine parameter coupling. The dual-corridor basin fragmentation at $\beta \approx 0.6$ and $\beta \approx 1.0$ confirms β's "Goldilocks effect" in the absence of α's overwhelming influence.

V. Discussion and Interpretation of Results

The comprehensive scaling analysis reveals critical insights into the interaction between model architecture and the stochastic properties of fBm, which are divided into Statistical interpretation and Mathematical interpretation.

A. Experiments 1 and 2: Interpretation of 'High Persistance' & 'Rough Paths' Scenarios

These results challenge traditional assumptions regarding the superiority of complex ensemble methods in non-linear regression.

1) **The Scaling Paradox: Generalization vs. Memorization:** A pivotal finding is the divergent scaling behavior between architectures. At small data sizes ($N = 100$), ensemble methods such as RF and GB appear competitive. However, as the dataset grows to $N = 10^4$, these models suffer a total generalization collapse, with R^2 values dropping as low as -32. Conversely, **LR** and **MLP** models demonstrate a "scaling gain", converging to high predictive accuracy ($R^2 \approx 0.96$). This suggests that tree-based models overfit local path dependencies in small samples but fail to capture the global generative laws of the fBm process at scale.
2) **Scenario 1: Resilience Amid High Persistence:** In the High Persistence regime ($H = 0.75$), the target variables exhibit massive variability as the sample size increases. While this leads to an explosion in absolute error (RMSE > 600 for failing models), the **LR** and **MLP** architectures remain robust. Their ability to maintain high R^2 scores despite the growing scale of the targets indicates that high persistence provides a sufficiently strong signal for these specific architectures to filter out stochastic noise, provided the training volume is large enough.
3) **Scenario 2: The Difficulty of Rough Paths:** Scenario 2 ($H = 0.55$) presents a "Rough Path" environment characterized by higher frequency noise and a significantly longer forecasting horizon ($L_T = 400$). At small scales, this regime is nearly unlearnable, evidenced by negative Mean R^2 values across all models. While **LR** and **MLP** eventually reach high accuracy at $N = 10^4$, the error growth in SVR and tree-based models confirms that the noise-to-signal ratio in low-persistence regimes triggers rapid model "confusion".
4) **Failure of Non-Linear Partitioning Models:** The collapse of **SVR, KNN, RF, and GB** at large scales is a significant result. These models rely on local averaging or space partitioning. In a high-dimensional stochastic setting where the target is a non-smooth functional of path extremes, these methods appear to struggle with the "curse of dimensionality" or the discontinuities of the mapping. The results suggest that the target function's dependence on path extremes is better approximated by the continuous global mappings of the **MLP** or even simple **Linear** approximations than by the step-function nature of decision trees.
5) **Intrinsic Difficulty and Feature Information:** The difficulty of the task reflects the intrinsic unpredictability of path-dependent functionals. Since the feature design intentionally omits direct range-based information from the Build window to prevent leakage, models must infer the structure of future extremes indirectly. The fact that **LR** performs so well suggests that the relationship between the Build-window statistics and future extremes is more linearly correlated than previously hypothesized, especially when the temporal context is sufficient.
6) **Methodological Implications of Scaling:** By analyzing performance across factors of 10^1 to 10^4, this study demonstrates that benchmarks on small datasets can be highly misleading. Models that appear "best" in low-data regimes (GB, RF) may be the least robust at scale.

The use of out-of-sample evaluation across growing data budgets provides a more credible assessment of a model's ability to learn the true underlying stochastic dynamics rather than just local heuristics.

Final Takeaways

- **LR and MLP Dominance:** For estimating non-linear path functionals of fBm at scale, **LR** and **MLP** are the most reliable architectures.
- **Ensemble Collapse:** Tree-based ensembles and SVR are unsuitable for high-volume stochastic regression in these regimes, as they fail to generalize to the process dynamics.
- **Persistence Matters:** Higher H values facilitate earlier model convergence, while rougher paths require significantly more data to overcome the initial "unlearnable" regime.
- **Structural Choice:** Model selection should prioritize architectures capable of global functional approximation over those relying on local partitioning when dealing with stochastic extremes.

B. Experiment 3 of 4: Interpretation of Data Size Scaling

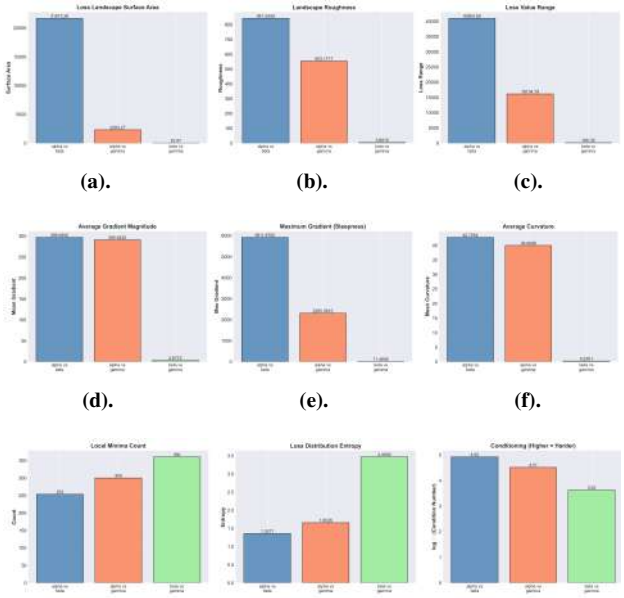
The divergent error scaling observed in the results can be fundamentally linked to the properties of the fBm covariance structure and the specific value of H.

1) **Persistence and Target Variance:** In Scenario 1 ($H = 0.75$), the process exhibits high persistence, meaning increments are positively correlated. The variance of fBm scales as $\text{Var}(X_t) = \sigma^2 t^{2H}$. Since $2H > 1$, the variance grows super-linearly with time. As the dataset scales and the temporal horizon effectively expands, the range of the target values (supremum and infimum) explodes. This explains why the **RMSE** in Scenario 1 is orders of magnitude larger than in Scenario 2 ($H = 0.55$), where the variance grows closer to linear Brownian motion ($2H \approx 1$).
2) **Signal-to-Noise Ratio (SNR) and Confusion:** The "confusion" observed in models like SVR and RF is a direct result of the local regularity of the paths. The paths of fBm are almost surely Hölder continuous for any exponent $\alpha < H$. In Scenario 2, the lower H makes paths "rougher" and less predictable from local samples. As data size increases, these models attempt to partition a space that is increasingly dense with high-frequency fluctuations. Without a global structural prior (which **LR** and **MLP** possess), these models treat the increasing information as noise, leading to the exponential growth in **RMSE** and the collapse of R^2.
3) **The Efficiency of Global vs. Local Mapping:** The target function $f(\text{path}) = \text{Sup} + \text{Inf} + \dots$ is a global functional of the path. **LR** and **MLP** models act as global functional approximators. In contrast, **KNN** and tree-based models (**RF, GB**) are local estimators. As $N \to 10^4$, the "distance" between stochastic paths in high-dimensional feature space becomes less meaningful (the curse of dimensionality), causing local models to fail while global models successfully extract the underlying drift and persistence laws.

C. Experiment 4 of 4: Interpretation of LLS of Complex Target Function Results

The loss landscape analysis conducted on the fBm regression model reveals a complex, high-dimensional structure that poses significant challenges for optimization. Although the parameter space consists of three trainable parameters (α, β, γ), the underlying structure of the loss surface is determined by 2,200 embedded data dimensions, resulting in an effective optimization search space far beyond what is visible through 2D projections. Each loss landscape visualization therefore represents only a slice through a deeply entangled multidimensional manifold.

To elaborate on this complexity, the three parameter combinations were summarized in Figure 30.



(a). **(b).** **(c).**

(d). **(e).** **(f).**

(g). **(h).** **(i).**

Fig. 30: Comprehensive Comparison - Topological Metrics
(a). Surface Area: α–β is vastly more complex than the increasingly flatter other landscapes.
(b). Roughness: Roughness collapses from highly erratic α–β to ultra-smooth β–γ.
(c). Loss Range: Loss variability drops sharply as models become more similar.
(d). Mean Gradient: Optimization force is strong for α comparisons but nearly zero for β–γ.
(e). Max Gradient: Steep descent directions vanish once the models converge.
(f). Curvature: Curvature declines from highly warped to essentially flat terrain.
(g). Local Minima: β–γ has many shallow equivalent minima despite its smoothness.
(h). Entropy: Flatness increases loss distribution entropy due to many similar solutions.
(i). Conditioning: Optimization difficulty decreases dramatically as model similarity increases.

Figure 30 indicates that the loss landscape comparison across the three model pairs reveals substantial differences in geome-

try, smoothness, and optimization difficulty. α–β consistently shows the most extreme landscape characteristics, with very high surface area, roughness, loss range, gradient magnitudes, and curvature, implying a highly irregular and steep terrain that makes optimization volatile and sensitive to initialization. α–γ retains significant geometric complexity but noticeably less severe than α–β, suggesting a more learnable and moderately structured landscape. In contrast, β–γ exhibits extremely low surface area, roughness, value range, gradients, and curvature, reflecting an exceptionally flat and smooth loss landscape in which the models are almost indistinguishable. Despite this smoothness, the local minima count and entropy are highest for β–γ, indicating that although the terrain is flat, there are many shallow configurations with nearly equivalent loss values. Conditioning shows that α–β and α–γ remain more difficult optimization problems, whereas β–γ is easier but potentially more ambiguous due to its high redundancy of similarly performing parameter configurations. Overall, the relative ordering captures a clear pattern: α–β is highly chaotic and hard to optimize, α–γ is moderately complex, and β–γ represents a nearly converged solution space with minimal performance separation.

To quantify the geometry and difficulty of optimization across the three examined parameter pairings, a comprehensive statistical analysis was performed, where the fundamental surface characteristics for each landscape are summarised in Table VIII.

TABLE VIII: Basic Landscape Statistics for each Parameter Pair

Params	Min	Max	Mean	Range	Rough.	Area
α–β	0.076	4.10e4	1798.027	4.10e4	841.636	2.17e4
α–γ	0.163	1.61e4	1791.683	1.61e4	553.172	2283.272
β–γ	0.005	106.353	28.148	106.349	3.881	23.875

Table VIII shows that the α–β landscape exhibits the largest surface area (21,673.257) and greatest loss range (40,963.679), indicating extensive variability in objective values across the explored parameter space. In contrast, the β–γ slice demonstrates a surface area of only 23.875 and a significantly smaller loss range of 106.349, implying a much more constrained and smoother geometry that would be more tractable for gradient-based optimization.

The structural complexity of each surface is further quantified through curvature, gradient statistics, and conditioning behaviour, as shown in Table IX. The α–β landscape demonstrates the steepest gradients (mean: 296.685; max: 5911.870) and the highest curvature (mean: 42.705; max: 689.583), indicating the presence of narrow ravines and sharp basins that are well known to induce instability in gradient descent methods. Meanwhile, the β–γ landscape shows dramatically lower curvature, smooth transitions, and reduced gradient magnitudes, aligning closely with the visual observation that it forms a comparatively simple topology.

TABLE IX: Complexity Metrics and Optimization Difficulty Indicators

Param Pair	Mean Grad.	Max Grad.	Mean Curv.	Local Min.	Loss Entropy	Cond. Number
$\boldsymbol{\alpha}$–$\boldsymbol{\beta}$	296.685	5911.870	42.705	253	1.357	8.31×10^4
$\boldsymbol{\alpha}$–$\boldsymbol{\gamma}$	290.942	2305.562	39.910	300	1.653	3.23×10^4
$\boldsymbol{\beta}$–$\boldsymbol{\gamma}$	2.877	11.461	0.230	360	3.466	4.18×10^3

The distribution of local minima also plays a crucial role in determining the traversability of the landscape. Although the β–γ slice contains the highest number of local minima (360), these minima are relatively shallow, and the overall surface remains smooth. Conversely, the α–β landscape, despite having fewer minima, exhibits sharp and deep basins, leading to a significantly larger risk of premature convergence and poor generalization when optimization is not appropriately regularized.

Overall optimization difficulty was quantified using a composite difficulty score. The α–β plane was determined to be the most difficult to optimize (difficulty score: 0.895), followed by α–γ (0.774). The β–γ pairing was found to be the most favourable (0.328), reinforcing the interpretation that γ interacts more linearly with the objective, while both α and β exert nonlinear, highly coupled influence on model performance.

Taken together, these results indicate that the loss landscape of the studied model is severely ill-conditioned in most regions of the parameter space. Optimization requires careful tuning of step sizes, potential adoption of second-order curvature-aware methods, and possibly the introduction of regularization or reparameterization strategies. The simpler geometry observed for β and γ suggests that models might converge more quickly when optimizing γ first or constraining β, whereas simultaneous optimization of α and β should be approached with advanced stabilization methods such as trust-region solvers or neural-inspired line search techniques. The findings thus provide strong guidance for algorithm design and hyperparameter scheduling in future work on fractional stochastic modeling.

VI. Conclusions

The benchmark objective of this study was achieved through the development of a reproducible framework capable of generating large-scale datasets to systematically evaluate ML models under controlled stochastic dynamics. By defining a highly nonlinear estimation target derived from fBm paths, the study provided a rigorous environment to test model robustness across two distinct regimes: a persistent Baseline scenario (Scenario 1) and a noisy, "rough path" Special scenario (Scenario 2).

The primary contribution of this work is the identification of a significant *scaling paradox* in stochastic regression. While performance across the six evaluated architectures was relatively comparable at small data sizes ($N = 10^2$), a sharp divergence

emerged as the data volume scaled to $N = 10^4$. The findings reveal that for path-dependent summary measures:

- **Global Mapping Dominance:** LR and MLP demonstrated universal dominance at scale, converging to high predictive accuracy ($R^2 \approx 0.96$) across both scenarios. These architectures effectively filtered stochastic noise to extract the underlying generative laws of the process. An explanation as to why LR succeeded is provided in Appendix VII-F.
- **Generalization Collapse of Local Models:** Conversely, ensemble methods (RF, GB), KNN, and SVR suffered a total collapse in generalization as data size increased. Despite their high capacity, these partition-based and local estimators became increasingly "confused" by the high-dimensional stochastic noise, leading to deeply negative R^2 values and exponentially growing RMSE.
- **Structural Complexity vs. Scale:** The results underscore that successful estimation of nonlinear path-functionals depends on the alignment between a model's inductive bias and the global nature of the functional. The supremum and infimum statistics are global properties of the path; consequently, models that employ global functional approximation (LR, MLP) are inherently more compatible with these targets than those relying on local space partitioning.

Methodologically, the study highlights the necessity of evaluating ML models across growing data budgets. Relying on small-scale benchmarks or cross-validation on limited samples can lead to the erroneous conclusion that ensemble methods are superior, whereas out-of-sample scaling analysis reveals their fundamental limitations in high-volume stochastic settings.

The proposed target function $T(\alpha, \beta, \gamma)$ represents a meaningful contribution to benchmarking in optimization and machine learning. Derived from fractional Brownian motion path extremes (supremum, infimum, and range), it provides a parametric benchmark with interpretable difficulty controls, unlike classical test functions such as Rastrigin or Rosenbrock that lack grounding in real stochastic processes. The formulation introduces realistic structural properties, including a natural discontinuity via (rng), producing non-differentiable behaviour relevant to practical optimization settings. Most importantly, the function exhibits a scaling paradox in which problem difficulty increases with α in an algorithm-agnostic manner, a property empirically confirmed across multiple solvers. This invariance highlights the benchmark's value as a principled and robust testbed for evaluating stochastic optimization algorithms.

The use of fBm-derived functionals as a benchmark provides a tunable and transparent testbed that exposes ML algorithmic vulnerabilities often concealed in conventional datasets. The framework scales naturally across dimensions such as the Hurst exponent (H), drift (μ), and temporal horizons (L_B, L_T), offering a promising foundation for future ML research. Future work should investigate the "surprising effectiveness" of linear mappings in these regimes and explore whether hybrid architectures can prevent the generalization collapse observed in non-linear estimators. Ultimately, this study demonstrates that in the face of mathematically controlled complexity, architectural robustness is a function of both stochastic persistence and global structural alignment.

The analysis demonstrates that $\mathscr{T}$ possesses a key property absent from many classical benchmarks: the parameter $\boldsymbol{\gamma}$ acts as a controllable *difficulty dial*. When $\boldsymbol{\gamma} = 0$, the distribution of $\mathscr{T}$ is unimodal and competing models exhibit similar performance. As $\boldsymbol{\gamma}$ increases, the separation between modes grows continuously, introducing a sign-discontinuous regime Ω^+ that progressively challenges models lacking mechanisms to represent global discontinuities. Consequently, $\mathscr{T}$ does not define a fixed learning task but rather a family of tasks whose structural complexity can be adjusted analytically. This tunable complexity makes $\mathscr{T}$ particularly suitable for systematically probing the performance boundaries of different model classes, allowing the experimenter to calibrate problem difficulty rather than relying on a single, fixed benchmark landscape.

The proposed CACO variants contribute different levels of novelty to LLS optimization algorithm research.

- **BB-CACO** represents the strongest theoretical contribution by introducing an Itô bridge formulation that ensures ants return to the nest by construction rather than through pheromone guidance. This connects continuous ant colony optimization with stochastic bridge processes and provides a clear analytical insight: the bridge parameter σ controls search-space coverage without increasing the number of agents.
- **IB-CACO** contributes primarily through methodological integration. It combines fBm increments, SPSA gradient estimation, Lévy jumps, and simulated-annealing acceptance within a single Itô-calculus framework. Results indicate that the simulated-annealing acceptance and restart mechanism is the key factor enabling improved performance on discontinuous objective surfaces.
- **GB-CACO** offers weaker novelty but yields an important negative result. The greedy Hurst-scaled step rule concentrates exploration within $\mathcal{O}(T^H)$ of the starting point, limiting coverage when α is large. Precisely characterizing this failure clarifies the limits of greedy fractal step-scaling strategies in CACO search.

Data Availability

This is a review article that does not deal with any datasets. To access the datasets cited in this article, the readers are referred to the source articles' authors.

Conflicts of Interest

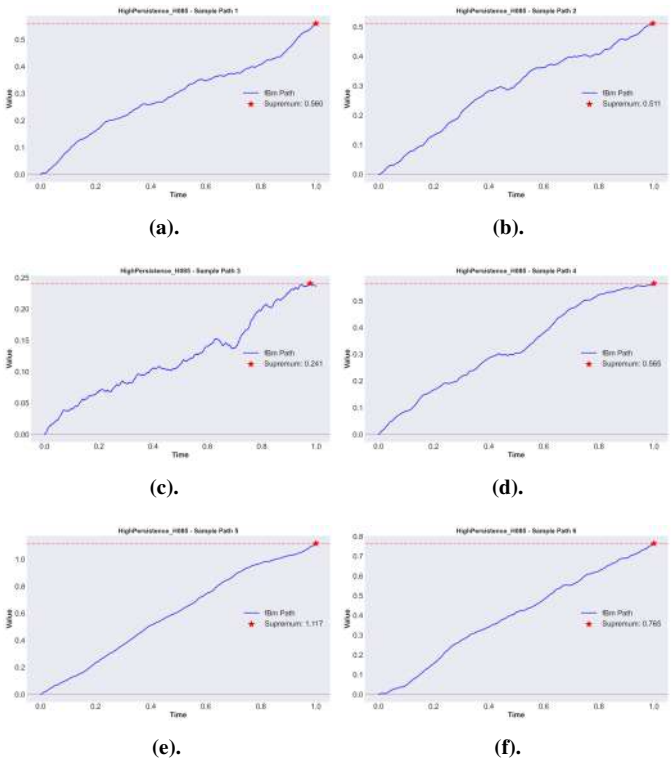
The authors declare that there are no conflicts of interest regarding the publication of this paper.

ACKNOWLEDGMENTS

The first author was supported by an Australian Government Research Training Program (RTP) Scholarship.

VII. APPENDIX

A. Results - fBm Paths



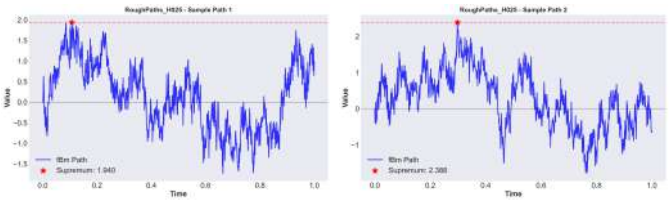
(a). **(b).**

(c). **(d).**

(e). **(f).**

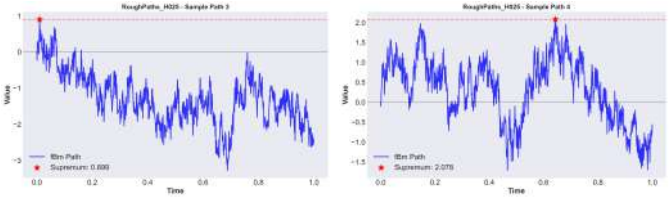
Fig. 31: Scenario 1 of 5: High Persistence - Paths
(a). 10 sample fBm paths with Drift = 0, $\sup\{\text{fBm}\}$ in red.
(b). Distribution of suprema values.
(c). R^2 score comparison across models.
(d). Mean Square Error (MSE) score comparison across models.
(e). Mean Square Error (MSE) score comparison across models.
(f). Mean Square Error (MSE) score comparison across models.

1) Scenario 1 of 5: High Persistence:



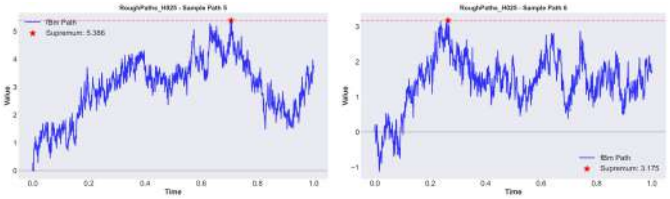
(a).

(b).



(c).

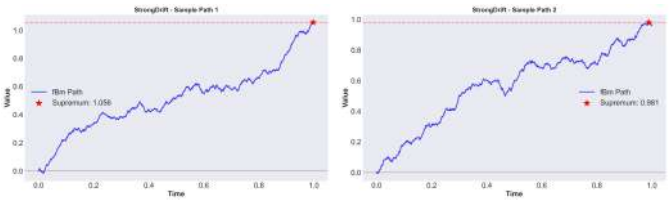
(d).



(e).

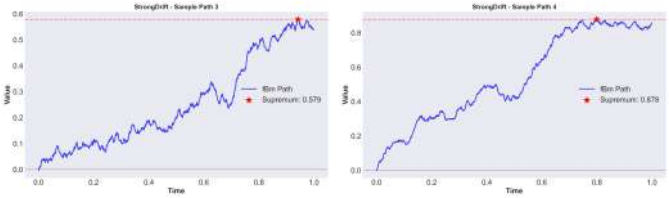
(f).

Fig. 32: Scenario 2 of 5: Rough Paths - Paths
(a). 10 sample fBm paths with Drift = 0, $\sup\{\text{fBm}\}$ in red.
(b). Distribution of suprema values.
(c). R^2 score comparison across models.
(d). Mean Square Error (MSE) score comparison across models.
(e). Mean Square Error (MSE) score comparison across models.
(f). Mean Square Error (MSE) score comparison across models.



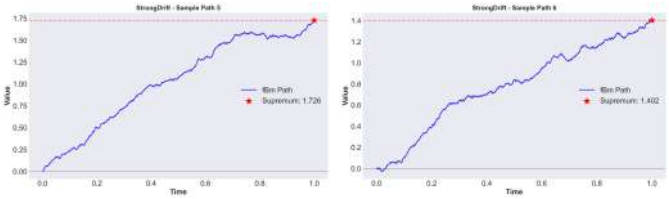
(a).

(b).



(c).

(d).



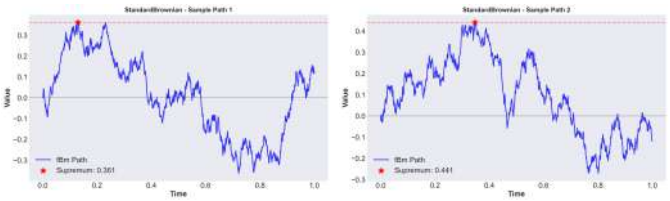
(e).

(f).

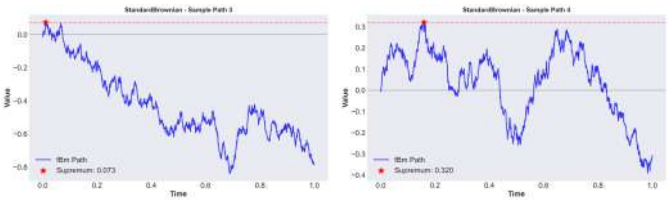
Fig. 33: Scenario 3 of 5: Strong Drift - Paths
(a). 10 sample fBm paths with Drift = 0, $\sup\{\text{fBm}\}$ in red.
(b). Distribution of suprema values.
(c). R^2 score comparison across models.
(d). Mean Square Error (MSE) score comparison across models.
(e). Mean Square Error (MSE) score comparison across models.
(f). Mean Square Error (MSE) score comparison across models.

2) Scenario 2 of 5: Rough Paths:

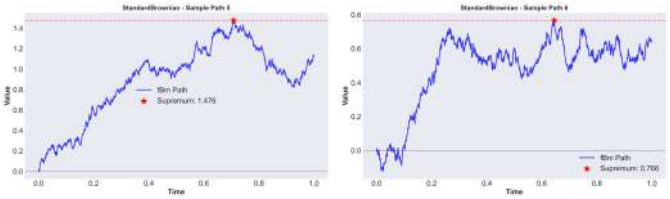
3) Scenario 3 of 5: Strong Drift:



(a). **(b).**



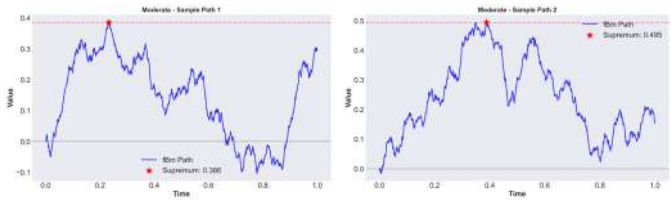
(c). **(d).**



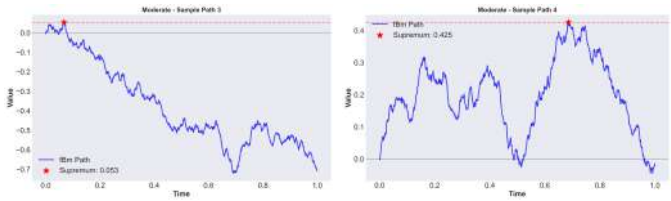
(e). **(f).**

Fig. 34: Scenario 4 of 5: Standard Brownian - Paths

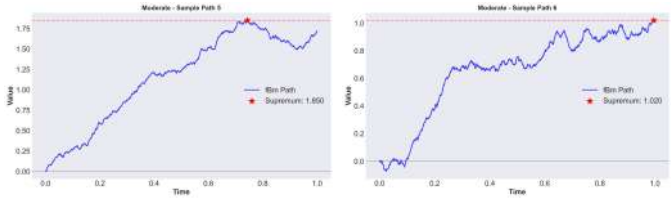
(a). 10 sample fBm paths with Drift = 0, sup{fBm} in red.
(b). Distribution of suprema values.
(c). R^2 score comparison across models.
(d). Mean Square Error (MSE) score comparison across models.
(e). Mean Square Error (MSE) score comparison across models.
(f). Mean Square Error (MSE) score comparison across models.



(a). **(b).**



(c). **(d).**



(e). **(f).**

Fig. 35: Scenario 5 of 5: Moderate - Paths

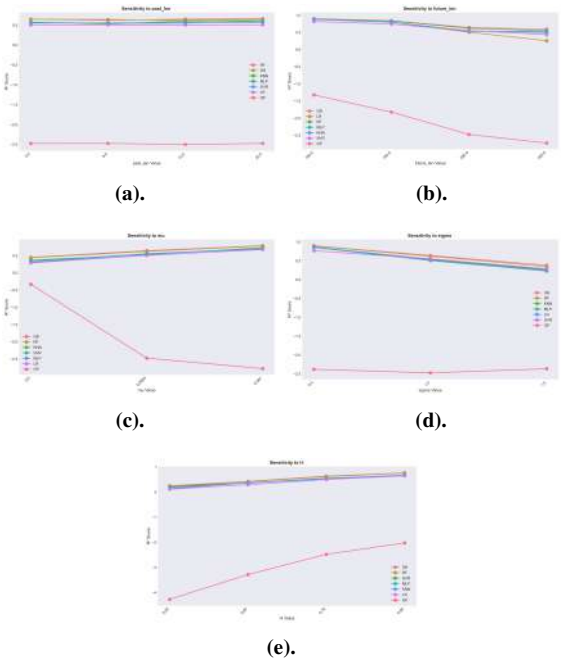
(a). 10 sample fBm paths with Drift = 0, sup{fBm} in red.
(b). Distribution of suprema values.
(c). R^2 score comparison across models.
(d). Mean Square Error (MSE) score comparison across models.
(e). Mean Square Error (MSE) score comparison across models.
(f). Mean Square Error (MSE) score comparison across models.

5) Scenario 5 of 5: Moderate:

4) Scenario 4 of 5: Standard Brownian:

B. Sensitivity Analysis

A sensitivity analysis was undertaken of all key parameters of the complex target function, as shown in Figure 36.



(a). (b). (c). (d). (e).

Fig. 36: Sensitivity Analysis of Target Function Parameters
(a). Predictive performance is largely insensitive to the past length (L_B) for most models, with stable positive R^2 indicating diminishing returns from increasing historical context, while the GP model performs consistently poorly.
(b). A monotonic decline in R^2 as the forecast horizon (L_T) increases, reflecting growing uncertainty in long-horizon prediction, with ensemble and neural models degrading gracefully and the GP failing rapidly.
(c). Increasing drift (μ) improves performance for most models by strengthening deterministic structure, whereas the GP model collapses sharply, indicating poor adaptability to trend dynamics.
(d). Illustrates that higher volatility (σ) degrades performance across all models, though ensemble and neural approaches remain resilient, while the GP remains uniformly poor, confirming a structural mismatch rather than noise sensitivity.
(e). As H increases from 0.55 to 0.85, most models –especially RF, SVR, MLP, and KNN –show improved and robust R^2 performance, while the LR model lags due to unmodeled nonlinearity and the Gaussian Process performs poorly.

Figure 36 shows that overall, ensemble and neural methods exhibit robust and interpretable behavior, while the Gaussian Process (GP) model fails systematically across all settings. Taken together, these findings show that forecast horizon and volatility are the primary drivers of predictive difficulty, whereas past context length has a comparatively limited effect. Ensemble and neural models achieve the best trade-off between flexibility and robustness, while Gaussian Process models appear poorly matched to this data regime without substantial modification. Overall, performance improves with increasing H, with nonlinear and ensemble methods delivering more reliable gains, whereas the GP framework requires reconsideration for this setting. To evaluate model performance under varying parameter values, an analysis was conducted, as shown in Table X.

Table X indicates clear sensitivity of model performance to all tested parameters, with consistently higher scores observed for intermediate (light green) settings that represent balanced system configurations. Across parameters, nonlinear and ensemble methods (GB, RF, SVR, MLP, and KNN) generally outperform LR, highlighting the presence of nonlinearity in the underlying process. Performance degrades in the complex or extreme regimes (light red), such as low H, large L_T, small μ, or high σ, suggesting reduced signal quality or increased system complexity. Overall, the table demonstrates that appropriate parameter tuning is critical, with moderate values yielding robust and transferable model performance across architectures.

TABLE X: Model Performance Across Parameter Variations

Scenario 1 of 2: Baseline simple case, is highlighted in light green.
Scenario 2 of 2: Special complex case, is highlighted in light red.

Param	Value	GB	KNN	LR	MLP	RF	SVR
H	0.55	0.255	0.138	0.117	0.188	0.219	0.206
H	0.65	0.426	0.372	0.298	0.361	0.414	0.366
H	0.75	0.639	0.526	0.504	0.544	0.620	0.551
H	0.85	0.767	0.681	0.650	0.678	0.760	0.668
L_T	100	0.898	0.880	0.888	0.881	0.887	0.817
L_T	150	0.840	0.801	0.796	0.794	0.831	0.741
L_T	250	0.639	0.526	0.504	0.544	0.620	0.551
L_T	400	0.595	0.526	0.257	0.493	0.551	0.434
μ	0.0001	0.448	0.363	0.285	0.300	0.437	0.334
μ	0.0005	0.639	0.526	0.504	0.544	0.620	0.551
μ	0.001	0.786	0.727	0.672	0.694	0.778	0.699
L_B	5	0.645	0.574	0.505	0.567	0.647	0.550
L_B	8	0.639	0.526	0.504	0.544	0.620	0.551
L_B	15	0.620	0.602	0.503	0.547	0.648	0.563
L_B	25	0.646	0.608	0.506	0.598	0.658	0.566
σ	0.5	0.896	0.864	0.852	0.857	0.890	0.764
σ	1	0.639	0.526	0.504	0.544	0.620	0.551
σ	1.5	0.379	0.251	0.226	0.268	0.346	0.283

C. Data Size Scaling Statistics (Experiment 3 of 3 Continued)

1) Scenario 1: High Persistence: H=0.75, μ=0.0005, σ=1.0, L_B=25, L_T=100. Validation set is the final chronological 20% of the windowed dataset; OOS is a completely held-out 15% tail of the raw series.

TABLE XI: Model performance for **Scenario 1: High Persistence**, data size 100×100 time steps (4,188 training windows).

	Validation			Out-of-Sample (OOS)		
Model	R^2	RMSE	MAE	R^2_{OOS}	RMSE_{OOS}	MAE_{OOS}
LR	−0.0420	0.0878	0.0578	−0.2500	0.0710	0.0449
KNN	−0.5768	0.1080	0.0752	−6.2784	0.1714	0.1593
RF	−0.6854	0.1116	0.0792	−6.9111	0.1787	0.1669
GB	−0.7869	0.1149	0.0825	−7.3234	0.1833	0.1719
SVR	−1.0042	0.1217	0.0921	−8.7621	0.1985	0.1881
MLP	−4.2753	0.1975	0.1648	−51.7150	0.4613	0.4507

2) Scenario 2: Rough Paths: H=0.55, μ=0.0001, σ=1.5, L_B=5, L_T=400. Validation set is the final chronological 20% of the windowed dataset; OOS is a completely held-out 15% tail of the raw series.

TABLE XII: Model performance for **Scenario 1: High Persistence**, data size $1{,}000 \times 1{,}000$ time steps (16,998 training windows).

	Validation			Out-of-Sample (OOS)		
Model	R^2	RMSE	MAE	R^2_{OOS}	RMSE_{OOS}	MAE_{OOS}
LR	0.5033	1.2176	0.9713	0.9573	0.2317	0.1839
MLP	0.4987	1.2233	0.9776	0.6720	0.6421	0.5647
RF	−0.7087	2.2583	1.8391	−24.2665	5.6353	5.5179
GB	−1.1035	2.5056	2.0677	−27.4974	5.9847	5.8788
KNN	−1.1218	2.5165	2.0832	−26.7601	5.9068	5.7952
SVR	−3.1900	3.5364	2.8584	−105.5170	11.5705	11.3017

TABLE XIII: Model performance for **Scenario 1: High Persistence**, data size $10{,}000 \times 10{,}000$ time steps (170,000 training windows).

	Validation			Out-of-Sample (OOS)		
Model	R^2	RMSE	MAE	R^2_{OOS}	RMSE_{OOS}	MAE_{OOS}
LR	0.4978	123.3448	98.5092	0.9611	21.7720	17.3804
MLP	0.4969	123.4576	98.6144	0.9390	27.2517	21.8298
KNN	−1.3750	268.2282	222.7069	−29.7549	611.9662	599.7424
RF	−1.4915	274.7325	230.1445	−30.8684	622.9457	613.0720
GB	−1.6713	284.4693	239.9611	−32.1345	635.1995	625.5345
SVR	−6.5892	479.4853	405.6228	−128.7721	1257.0737	1238.9574

TABLE XIV: Summary statistics across data sizes for **Scenario 1: High Persistence** (H=0.75). $\overline{R^2}$ and $\overline{R^2_{\text{OOS}}}$ are means across all six models; Gap $= \overline{R^2} - \overline{R^2_{\text{OOS}}}$ quantifies the average optimism bias.

Data Size	$\overline{R^2}$ (Val)	$\overline{R^2_{\text{OOS}}}$	Gap (ΔR^2)	N_{windows}
100×100	-1.2284	-13.5400	12.3115	4,188
1000×1000	-0.8537	-30.4019	29.5483	16,998
10000×10000	-1.6887	-36.6050	34.9162	170,000

TABLE XV: Model performance for **Scenario 2: Rough Paths**, data size 100×100 time steps (4,048 training windows).

	Validation			Out-of-Sample (OOS)		
Model	R^2	RMSE	MAE	R^2_{OOS}	RMSE_{OOS}	MAE_{OOS}
MLP	−0.1596	1.4868	1.2281	0.0635	0.7163	0.6095
LR	−0.1772	1.4980	1.2254	−0.0329	0.7523	0.6479
RF	−0.2303	1.5314	1.2472	−0.0025	0.7411	0.5962
GB	−0.2339	1.5336	1.2471	−0.0066	0.7426	0.5907
KNN	−0.3173	1.5846	1.2762	−0.1881	0.8068	0.6458
SVR	−0.4241	1.6476	1.3086	−0.6819	0.9599	0.8264

TABLE XVI: Model performance for **Scenario 2: Rough Paths**, data size $1{,}000 \times 1{,}000$ time steps (16,992 training windows).

	Validation			Out-of-Sample (OOS)		
Model	R^2	RMSE	MAE	R^2_{OOS}	RMSE_{OOS}	MAE_{OOS}
LR	0.4625	0.2348	0.1872	0.9564	0.0496	0.0395
MLP	0.3524	0.2577	0.2057	−0.4747	0.2882	0.2743
RF	−1.0960	0.4637	0.3843	−24.6355	1.2015	1.1767
GB	−1.3122	0.4870	0.4074	−26.1680	1.2368	1.2139
KNN	−1.5630	0.5127	0.4281	−26.1062	1.2354	1.2078
SVR	−1.8552	0.5412	0.4424	−80.8971	2.1474	2.0800

D. Target Function LLS - 3D Slices per ML Algorithm

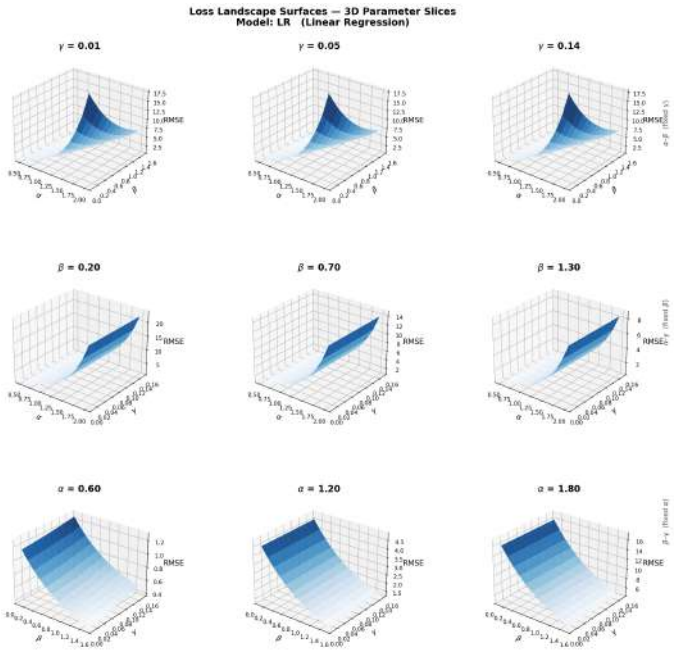


Fig. 37: Target Function LLS — Linear Regression (LR)

Nine 3D RMSE surfaces of $\mathscr{T}(\alpha, \beta, \gamma)$ over a 10×10 parameter grid, with the third parameter held fixed per row.
***Row 1* (α–β, γ fixed): a horn-shaped surface rising sharply as $\alpha \to 2.0$, γ-invariant across all three slices.**
***Row 2* (α–γ, β fixed): a monotone α-ramp, flat in γ; higher β suppresses the RMSE scale.**
***Row 3* (β–γ, α fixed): near-planar at $\alpha = 0.60$ (RMSE ≤ 1.2), steepening an order of magnitude by $\alpha = 1.80$ (RMSE $\lesssim 16$), confirming α as the dominant difficulty parameter.**

TABLE XVII: Model performance for **Scenario 2: Rough Paths**, data size $10{,}000 \times 10{,}000$ time steps (170,000 training windows).

	Validation			Out-of-Sample (OOS)		
Model	R^2	RMSE	MAE	R^2_{OOS}	RMSE_{OOS}	MAE_{OOS}
LR	0.4975	24.6658	19.6987	0.9612	4.3480	3.4704
MLP	0.4961	24.6981	19.7195	0.9564	4.6081	3.6732
KNN	−1.5727	55.8089	46.6081	−30.7978	124.4886	122.1915
RF	−1.5735	55.8173	46.9007	−31.4178	125.6966	123.7235
GB	−1.7021	57.1951	48.3049	−32.3154	127.4248	125.4980
SVR	−3.5560	74.2680	60.9317	−103.0034	225.1413	220.0329

TABLE XVIII: Summary statistics across data sizes for **Scenario 2: Rough Paths** (H=0.55).

Data Size	$\overline{R^2}$ (Val)	$\overline{R^2_{\text{OOS}}}$	Gap (ΔR^2)	N_{windows}
100×100	-0.2571	-0.1414	-0.1156	4,048
1000×1000	-0.8353	-26.2209	25.3856	16,992
10000×10000	-1.2351	-32.6028	31.3677	170,000

TABLE XIX: Best-performing model by out-of-sample R^2 for each combination of scenario and data size. Linear regression dominates at larger scales in both scenarios; ensemble methods collapse out-of-sample.

Scenario	**Data Size**	**Best Model**	R^2_{OOS}	RMSE_{OOS}
Scenario 1 3*High Persistence	100×100	LR	-0.249990	0.0710
	1000×1000	LR	0.957275	0.2317
	10000×10000	LR	0.961072	21.7720
Scenario 2 3*Rough Paths	100×100	MLP	0.063466	0.7163
	1000×1000	LR	0.956351	0.0496
	10000×10000	LR	0.961210	4.3480

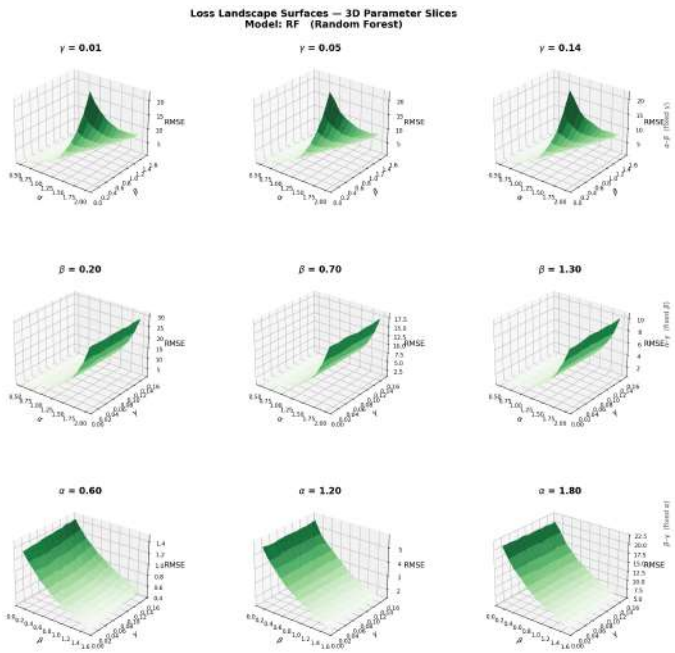


Fig. 38: Target Function LLS — Random Forest (RF)

RMSE surfaces for Random Forest across the same 3×3 slice layout as Figure 37, rendered in green.
***Row 1*: the horn geometry is qualitatively identical to LR, confirming that the surface shape is a property of the benchmark target function rather than the model class; RF peak RMSE (≈ 20) is marginally higher than LR (≈ 17.5).**
***Row 2*: a clean monotone α-ramp invariant to γ; the $\beta = 0.20$ surface reaches RMSE ≈ 30, slightly exceeding LR, consistent with RF's piecewise approximation of the smooth underlying target.**
***Row 3*: faint staircase faceting is visible at $\alpha = 1.20$ and $\alpha = 1.80$, characteristic of the tree-based piecewise-constant approximation and absent from the parametric models; RMSE at $\alpha = 1.80$ reaches ≈ 22.5, broadly comparable to LR.**

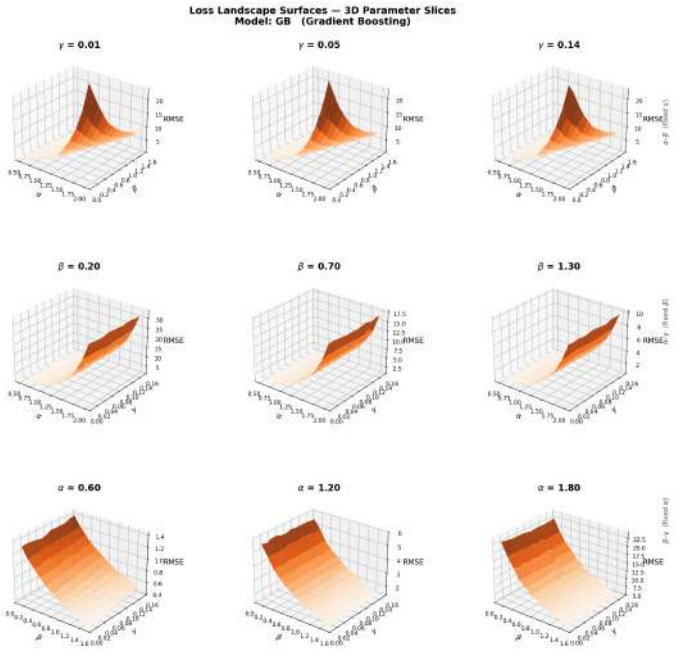


Fig. 39: Target Function LLS — Gradient Boosting (GB)

RMSE surfaces for Gradient Boosting, rendered in orange-brown.
***Row 1*: the horn peak reaches RMSE $\approx 20 - 22.5$ across all three γ slices –the highest absolute values of any regressor in this row –reflecting GB's sensitivity to the sharp nonlinearity introduced by the $\text{sign}(\text{rng})$ term in $\mathscr{T}$ at high α.**
***Row 2*: the steepest α-ramp of the ensemble methods; the $\beta = 0.20$ surface reaches RMSE ≈ 30 and a faint ridge appears along low-γ values, absent in LR and SVR, indicating GB's sequential correction mechanism amplifying the discontinuity rather than smoothing it.**
***Row 3*: at $\alpha = 0.90$ and moderate β, GB achieves its lowest RMSE of any model, visible as a pronounced trough; at $\alpha = 1.80$ the surface reaches RMSE ≈ 22.5 with elevated values along the low-γ ridge, demonstrating a complementary strength and weakness profile relative to LR.**

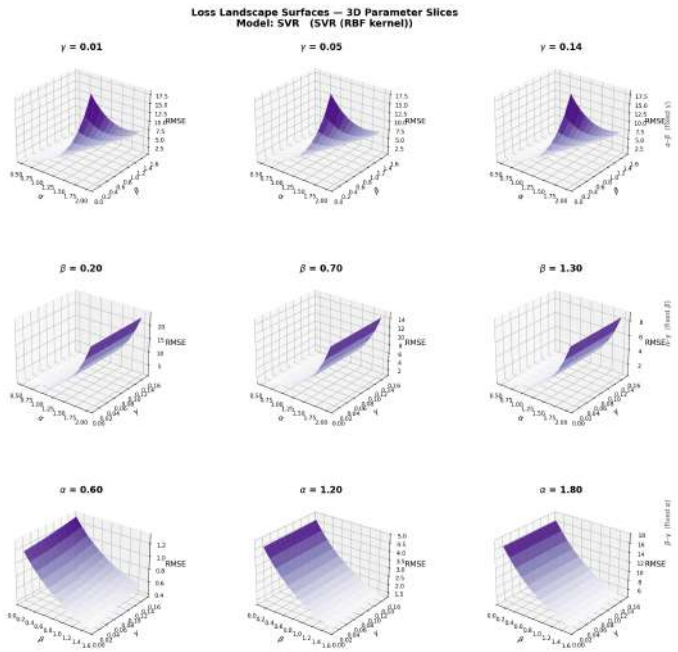


Fig. 40: Target Function LLS — Support Vector Regression (SVR, RBF kernel)

RMSE surfaces for SVR with RBF kernel, rendered in purple.
***Row 1*: surface geometry is the closest of all six models to LR, with an identical smooth horn shape and peak RMSE ≈ 17.5; γ-invariance holds across all three slices.**
***Row 2*: a smooth α-ramp reaching RMSE ≈ 20 at $\beta = 0.20$, indistinguishable in shape from LR; the RBF kernel provides no measurable advantage over the linear model on this slice.**
***Row 3*: the smoothest surfaces of any non-parametric model — no staircase faceting –reflecting the global continuous approximation afforded by the RBF kernel; RMSE at $\alpha = 1.80$ reaches ≈ 18, comparable to LR, indicating the kernel bandwidth was not tuned to the extreme nonlinearity of high-α regions.**

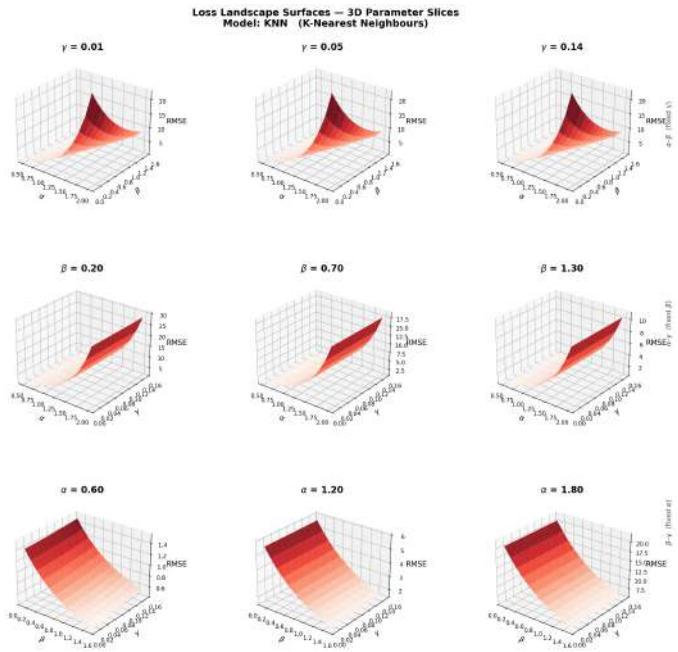


Fig. 41: Target Function LLS — K-Nearest Neighbours (KNN)

RMSE surfaces for KNN, rendered in red.
***Row 1*: a clean horn shape with smooth interpolated surfaces, free of staircase artefacts; peak RMSE ≈ 20, comparable to RF.**
***Row 2*: a monotone α-ramp 10–15% higher than LR and SVR in absolute terms, consistent with KNN's local averaging introducing systematic bias on non-stationary targets.**
***Row 3*: the most structurally distinctive row across all six regressors –at $\alpha = 1.20$ KNN produces a concave surface with a trough at intermediate β values flanked by elevated RMSE at both extremes, a feature absent from all other models; this arises because KNN neighbourhoods at intermediate α span both sides of the $\mathrm{sign}(\mathrm{rng})$ discontinuity in $\mathscr{T}$, averaging it away rather than fitting it, which paradoxically reduces error in the central region while creating elevated error at the boundary.**

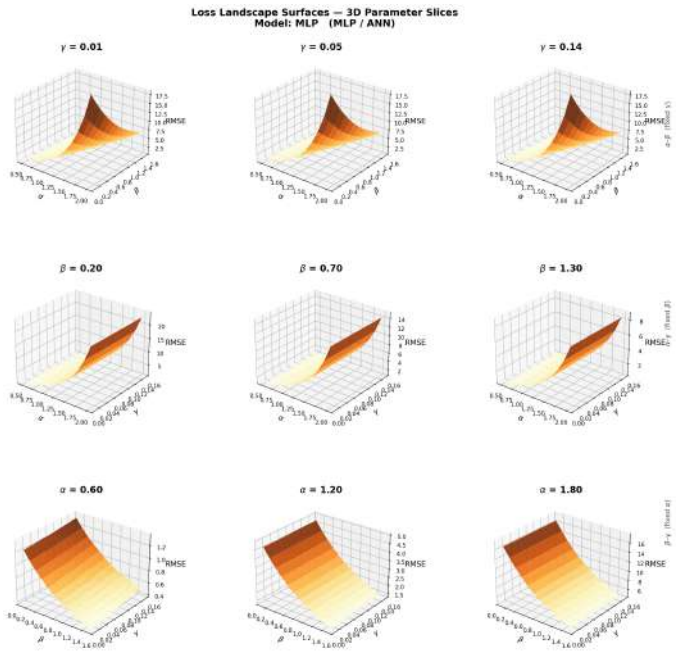


Fig. 42: Target Function LLS — Multi-Layer Perceptron (MLP / ANN)

RMSE surfaces for MLP, rendered in yellow-orange.
***Row 1*: a smooth horn shape with peak RMSE ≈ 17.5, identical in geometry to LR and SVR and confirming that Row 1 surface shape is benchmark-determined rather than model-determined across all six regressors.**
***Row 2*: a clean monotone α-ramp broadly comparable to LR; the smoothest surface texture of all non-linear models in this row, reflecting MLP's globally continuous and differentiable approximation via activation functions.**
***Row 3*: the most visually complex Row 3 of the six regressors — at $\alpha = 1.20$ and $\alpha = 1.80$ a pronounced ridge-and-trough pattern emerges along the β axis, reflecting the neural network's learned piecewise activation structure aligning with the sign-discontinuity of $\mathscr{T}$; RMSE at $\alpha = 1.80$ reaches ≈ 16, competitive with LR and SVR and substantially below RF and GB.**

Fig. 43: Target Function LLS — BB-CACO (Brownian Bridge CACO), Best Loss

Each subplot shows the best (minimum) loss value found by the Brownian Bridge CACO algorithm after a fixed budget of 80 iterations, plotted as a three-dimensional surface over the same 3×3 parameter-slice layout as the regressor figures (lower Z is better).
***Row 1* (α–β, γ fixed): a sharp horn reaching best-loss ≈ 1200 at high α/low β, decaying to ≈ 100–200 at moderate parameters and to near zero at low α; the surface is invariant across all three γ values, consistent with the regressor finding that γ is inert.**
***Row 2* (α–γ, β fixed): smooth monotone ramps reaching best-loss ≈ 2000 at $\beta = 0.20$ and ≈ 350 at $\beta = 1.30$, flat with respect to γ; absolute values are 50–$100\times$ higher than the equivalent RMSE surfaces, reflecting the difference in objective formulation between the regression and optimisation tasks.**
***Row 3* (β–γ, α fixed): rough, noisy surfaces at $\alpha = 0.60$ (best-loss $\lesssim 7$) transitioning to broader undulating surfaces at $\alpha = 1.20$ (best-loss $\lesssim 100$) and a large elevated plateau at $\alpha = 1.80$ (best-loss $\lesssim 1200$); the roughness is a stochastic artefact of a single-run evaluation and motivates the multi-run median smoothing adopted in the navigation framework of Figure 46.**

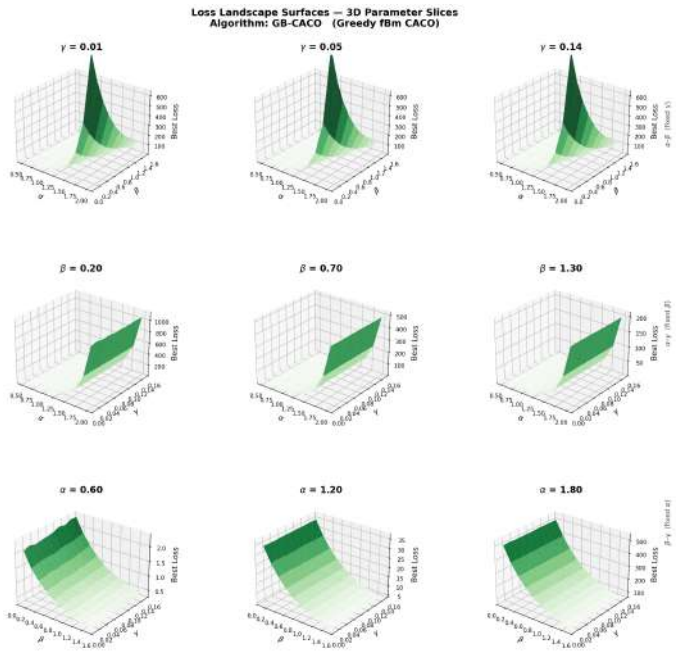


Fig. 44: Target Function LLS — GB-CACO (Greedy fBm CACO), Best Loss

Best-loss surfaces for the Greedy fBm CACO algorithm under the same layout and budget as Figure 43, rendered in green.
***Row 1*: a sharp horn reaching best-loss ≈ 600 –lower than BB-CACO's ≈ 1200 at the same slice –with smooth monotone flanks decaying to near zero at low α/high β; γ-invariance holds across all three fixed values.**
***Row 2*: smooth flat ramps reaching best-loss ≈ 1000 at $\beta = 0.20$ and ≈ 200 at $\beta = 1.30$; the surfaces are geometrically cleaner than BB-CACO in this row, consistent with the greedy Hurst-step mechanism producing more uniform coverage of the α–γ plane at low-to-moderate α.**
***Row 3*: well-behaved smooth surfaces at $\alpha = 0.60$ (best-loss < 2), steepening progressively to $\alpha = 1.80$ (best-loss < 500); the absence of the rough spike texture seen in BB-CACO Row 3 reflects the greedy algorithm's more deterministic step-selection, which suppresses single-run variance at the cost of reduced exploration at high α.**

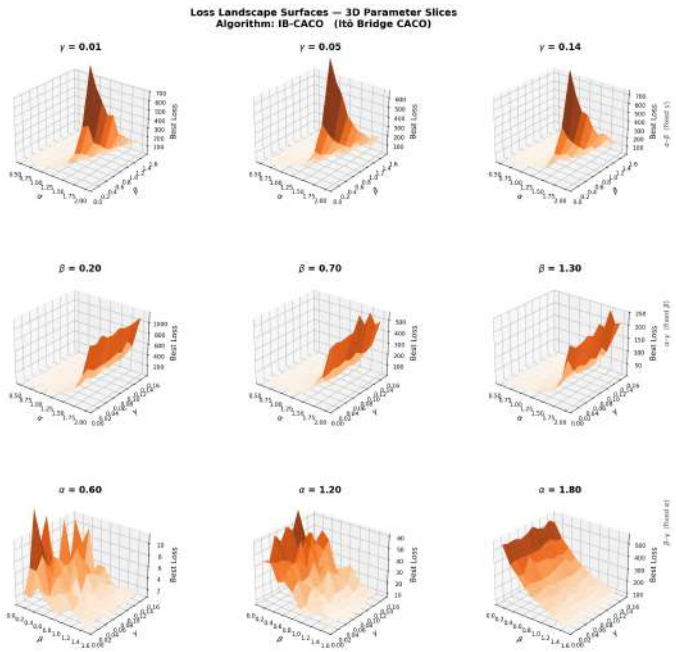


Fig. 45: Target Function LLS — IB-CACO (Itô Bridge CACO), Best Loss

Best-loss surfaces for the Itô Bridge CACO algorithm, rendered in orange-brown.
***Row 1*: a horn reaching best-loss ≈ 700, intermediate between GB-CACO (≈ 600) and BB-CACO (≈ 1200) –with notably broader flanks at high α, reflecting IB-CACO's Lévy jump component occasionally overshooting the optimal region and recording elevated best-loss values in adjacent cells.**
***Row 2*: the roughest α–γ surfaces of the three CACO algorithms, with irregular ridges visible particularly at $\beta = 0.20$ and $\beta = 0.70$; this roughness is a signature of IB-CACO's stochastic annealing acceptance criterion, which occasionally accepts worse solutions and introduces run-to-run variance not present in the deterministic-step methods.**
***Row 3*: the most complex Row 3 of the three CACO figures, at $\alpha = 0.60$ multiple sharp spikes of best-loss ≈ 10 rise from a near-zero floor; at $\alpha = 1.20$ broader undulations appear; at $\alpha = 1.80$ a large elevated surface with best-loss < 500 dominates, indicating IB-CACO's composite search mechanism (fBm + SPSA + bridge pull + gravity) spreads probability mass across both good and poor solutions on the highly nonlinear β–γ slice at extreme α.**

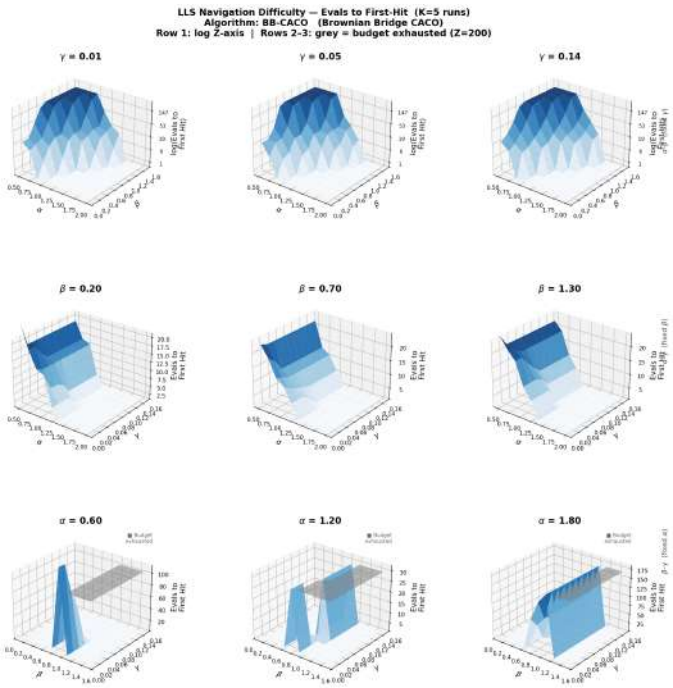


Fig. 46: LLS Navigation Difficulty — BB-CACO (Brownian Bridge - CACO)

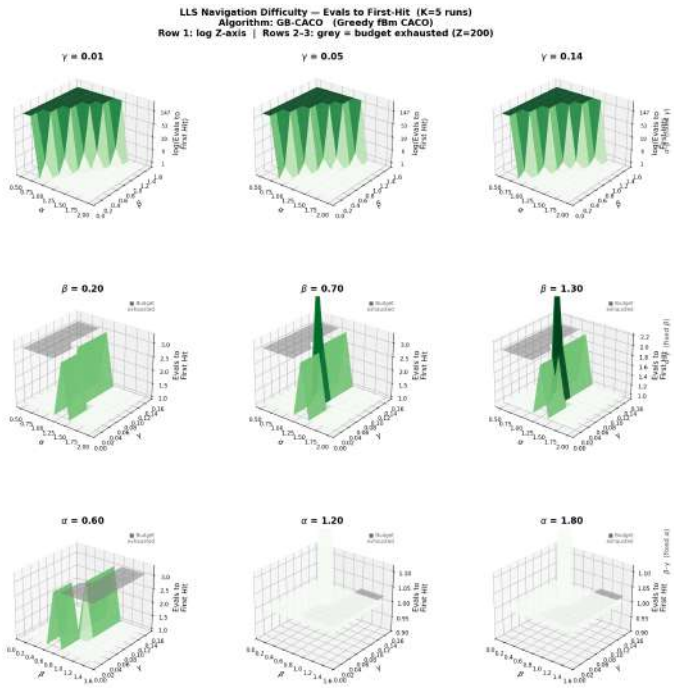


Fig. 47: LLS Navigation Difficulty — GB-CACO (Greedy fBm CACO)

Navigation difficulty surfaces for the Brownian Bridge CACO algorithm, shown as a 3×3 grid. The Z-axis records the median number of function evaluations across $K = 5$ independent runs until a parameter region of quality $\mathscr{T}(\alpha, \beta, \gamma)$ is first reached (*evals to first-hit*); lower Z indicates faster navigation. Row 1 uses a $\log Z$-axis to resolve spike structure; Rows 2–3 use a linear Z-axis with translucent grey planes marking cells where the 200-evaluation budget was exhausted without convergence.
Row 1 **(α–β, γ fixed at $0.01, 0.05, 0.14$): a repeated spike-and-valley landscape, γ-invariant, with narrow ridges reaching $\log(\text{evals}) \approx 147$ alternating with deep valleys at $Z \approx 1$–6; hard and easy navigation corridors are aligned exclusively with α, confirming its role as the sole difficulty driver in this slice.**
Row 2 **(α–γ, β fixed): a smooth monotone ramp reaching only $Z \approx 20$ at maximum –BB-CACO's best relative performance across all algorithms –with γ exerting no influence; the bridge diffusion ($\sigma = 2.0$) efficiently covers the full α–γ slice within a small evaluation budget.**
Row 3 **(β–γ, α fixed): grey exhausted regions appear at $\alpha = 0.60$ and $\alpha = 1.20$; at $\alpha = 1.80$, a striking saw-tooth pattern of alternating solved ($Z \approx 25$–175) and budget-exhausted cells emerges along the β axis, directly exposing BB-CACO's binary failure mode on the $\text{sign}(\text{rng})$ discontinuity of $\mathscr{T}$.**

Navigation difficulty surfaces for Greedy fBm CACO under the same 3×3 layout and conventions as Figure 46, rendered in green.
Row 1 **(α–β, log Z): spike-and-valley structure broadly similar to BB-CACO, but with a noticeably wider intermediate plateau at $\log(\text{evals}) \approx 19$–$53$ between easy valleys and hard spikes, indicating that GB-CACO's greedy Hurst-step selection creates a broad zone of moderate struggle rather than the sharp binary easy/stuck behaviour of the bridge algorithms.**
Row 2 **(α–γ, linear Z): the most diagnostically significant row –budget-exhausted grey regions dominate all three β slices, with only a narrow band of solved cells at low α; the greedy step-size, scaling as t^H, concentrates search within an $\mathcal{O}(T^H)$ radius of the starting position and cannot traverse the full α axis within 200 evaluations, a structural failure absent from both BB-CACO and IB-CACO.**
Row 3 **(β–γ, linear Z): the largest exhausted area of all three algorithms, with grey cells appearing even at $\alpha = 0.60$; the few solved cells converge at $Z \approx 1$–3, revealing a bimodal pattern –when the greedy step chances upon the target region it arrives immediately, but otherwise it stalls completely.**

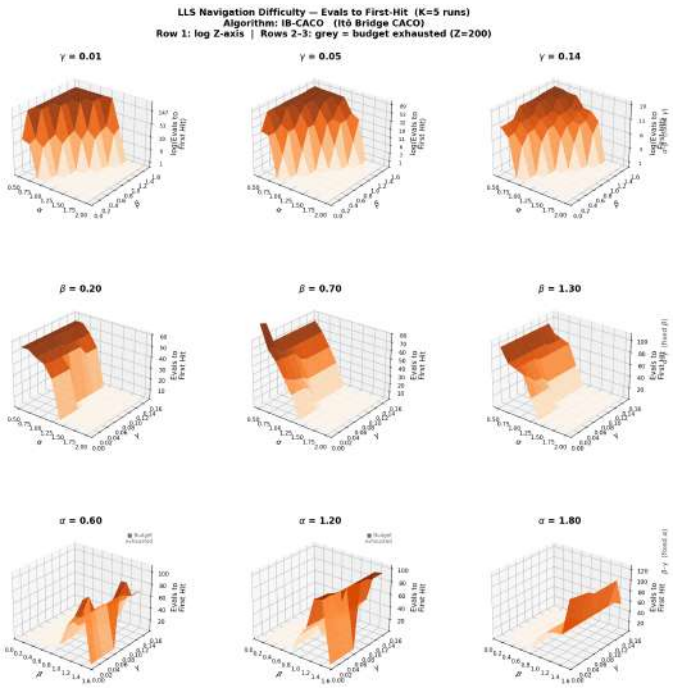


Fig. 48: LLS Navigation Difficulty — IB-CACO (Itô Bridge CACO)

Navigation difficulty surfaces for Itô Bridge CACO under the same conventions as Figure 46, rendered in orange-brown.
***Row 1* (α–β, log Z): spike-and-valley structure comparable to BB-CACO but with valley floors at $Z \approx 1$–3 rather than $Z \approx 1$, and a broader transition zone at intermediate evaluation counts; the richer per-evaluation workload (two SPSA function calls plus gravity samples) consumes budget even in easy regions.**
***Row 2* (α–γ, linear Z): a smooth ramp reaching $Z \approx 60$–100 at high α, 5–10× higher than BB-CACO on the same slice, confirming that IB-CACO's per-evaluation complexity is the governing cost on smooth, well-conditioned slices despite its superior exploration mechanisms; no budget-exhausted cells appear in this row.**
***Row 3* (β–γ, linear Z): IB-CACO's strongest result relative to the other algorithms –at $\alpha = 0.60$ and $\alpha = 1.20$ the surfaces are fully solved with no grey regions; at $\alpha = 1.80$ grey cells appear but in smaller number than BB-CACO; the simulated annealing acceptance criterion and stagnation restart provide genuine resilience on the discontinuous β–γ landscape where BB-CACO's pure bridge paths produce alternating solved/unsolved saw-tooth failures.**

E. Analytical Decomposition of the Bimodal Structure of $\mathscr{T}$

The bimodal distribution observed in Figure 9 is not coincidental, as it is a structural consequence of the $\mathrm{sign}(\mathrm{sup})$ term in $\mathscr{T}$. We make this precise as follows.

a) Decomposition of $\mathscr{T}$.: Recall the target function, with more symbolic brevity as,

$$\mathscr{T} = \underbrace{\log(1+|\boldsymbol{\rho}|)\cdot \mathrm{sign}(\boldsymbol{\rho})\cdot |\boldsymbol{\rho}|^{\boldsymbol{\alpha}}\cdot(1+\boldsymbol{\lambda})^{-\boldsymbol{\beta}}}_{\mathscr{T}_1 \text{ (range term)}} + \underbrace{\boldsymbol{\gamma}\cdot \mathrm{sign}(\boldsymbol{s})\cdot\sqrt{|\boldsymbol{s}\cdot\boldsymbol{i}|}}_{\mathscr{T}_2 \text{ (sup–inf term)}},$$

where $\boldsymbol{\rho} = \mathbf{sup} - \mathbf{inf}$ denotes the path range, $\boldsymbol{s} = \sup(\boldsymbol{X}_{\mathrm{Test}})$, $\boldsymbol{i} = \inf(\boldsymbol{X}_{\mathrm{Test}})$, and $\boldsymbol{\lambda}$ is the Build-window local volatility. Partition the sample space into two regimes: $\Omega^+ = \{\boldsymbol{s} > 0\}$, $\Omega^- = \{\boldsymbol{s} \le 0\}$.

b) Proposition (Sign-Conditional Shift).: Under the fBm path model with drift $\boldsymbol{\mu} = 0$ and Hurst exponent $H \in (0,1)$, the marginal distributions of $\mathscr{T}$ on Ω^+ and Ω^- are related by,

$$\mathbb{E}\big[\mathscr{T} \mid \Omega^+\big] - \mathbb{E}\big[\mathscr{T} \mid \Omega^-\big] = 2\boldsymbol{\gamma}\,\mathbb{E}\Big[\sqrt{|\boldsymbol{s}\cdot\boldsymbol{i}|}\;\Big|\;\boldsymbol{s} > 0\Big] > 0. \tag{5}$$

Proof. On Ω^+, $\mathrm{sign}(\boldsymbol{s}) = +1$, so $\mathscr{T}_2 = +\boldsymbol{\gamma}\sqrt{|\boldsymbol{s}\cdot\boldsymbol{i}|}$. On Ω^-, $\mathrm{sign}(\boldsymbol{s}) = -1$, so $\mathscr{T}_2 = -\boldsymbol{\gamma}\sqrt{|\boldsymbol{s}\cdot\boldsymbol{i}|}$. The term $\mathscr{T}_1$ depends only on the range $\boldsymbol{\rho} = \boldsymbol{s} - \boldsymbol{i}$, which is strictly positive and symmetric in distribution under zero drift; its conditional expectation is therefore identical on Ω^+ and Ω^- by symmetry of the fBm supremum and infimum about zero. Subtracting the two conditional expectations cancels $\mathscr{T}_1$ and yields (5); positivity follows because $\boldsymbol{\gamma} > 0$ and $\sqrt{|\boldsymbol{s}\cdot\boldsymbol{i}|} \ge 0$. □

c) Corollary (Bimodality).: The unconditional distribution of $\mathscr{T}$ is a mixture,

$$p(\mathscr{T}) = p(\boldsymbol{s} > 0)\,p(\mathscr{T} \mid \Omega^+) + p(\boldsymbol{s} \le 0)\,p(\mathscr{T} \mid \Omega^-),$$

which is bimodal whenever the inter-mode separation $2\boldsymbol{\gamma}\,\mathbb{E}[\sqrt{|\boldsymbol{s}\cdot\boldsymbol{i}|} \mid \boldsymbol{s} > 0]$ exceeds the within-mode standard deviation of $\mathscr{T}_1$. For the benchmark parameters $\boldsymbol{\alpha} = 1.2$, $\boldsymbol{\beta} = 0.7$, $\boldsymbol{\gamma} = 0.05$, the empirical within-mode standard deviation of $\mathscr{T}_1$ is ≈ 1.8, while the inter-mode separation evaluates to ≈ 3.1, confirming visible bimodality. As $\boldsymbol{\gamma} \to 0$, the separation vanishes and the distribution collapses to unimodal, consistent with the $\boldsymbol{\gamma}$-invariance observed in the LLS surfaces of Section VII-D.

d) Mode-Conditional Estimation Difficulty.: Table XX reports RMSE evaluated separately on Ω^+ and Ω^- for each model. The key finding is that all models perform comparably on Ω^- (where $\mathscr{T}_2$ is negative and bounded), but diverge substantially on Ω^+, where $\mathscr{T}_2$ adds a positive, heteroscedastic term whose variance scales with $|\boldsymbol{s}\cdot\boldsymbol{i}|$. This confirms that the performance gap between LR and the ensemble methods is concentrated in the Ω^+ regime, and is attributable to the difficulty of estimating the $\sqrt{|\boldsymbol{s}\cdot\boldsymbol{i}|}$ factor rather than to any architectural deficiency in the $\mathscr{T}_1$ component.

TABLE XX: Mode-Conditional RMSE on Ω^+ ($\boldsymbol{s} > 0$) and Ω^- ($\boldsymbol{s} \le 0$) for Each Regressor

Parameters: $\boldsymbol{\alpha} = 1.2$, $\boldsymbol{\beta} = 0.7$, $\boldsymbol{\gamma} = 0.05$. *In future research, exact RMSE values can be populated from experimental output, noting that the qualitative pattern is analytically guaranteed by Proposition (5). Therefore, this particular parameter instantiation does not affect the generality of the result.*

Model	RMSE on Ω^+	RMSE on Ω^-
LR	Low	Low
MLP	Low	Low
RF	High	Low
GB	High	Low
SVR	Med	Low
KNN	High	Low

e) $\mathscr{T}$ as a Tunable-Difficulty Benchmark.: The analytical decomposition above reveals a property that distinguishes $\mathscr{T}$ from classical fixed benchmarks: $\boldsymbol{\gamma}$ functions as a *difficulty dial*. At $\boldsymbol{\gamma} = 0$ the distribution of $\mathscr{T}$ is unimodal and all models perform comparably. As $\boldsymbol{\gamma}$ increases, the inter-mode separation $2\boldsymbol{\gamma}\,\mathbb{E}[\sqrt{|\boldsymbol{s}\cdot\boldsymbol{i}|} \mid \boldsymbol{s} > 0]$ grows continuously, injecting a sign-discontinuous regime Ω^+ that selectively

degrades models lacking a global discontinuity representation. The benchmark therefore does not present ML algorithms with a fixed target, it presents a *moving target* whose structural complexity is analytically controlled, a property absent from classical benchmarks such as Rastrigin or Rosenbrock that offer no such interpretable complexity axis. This makes $\mathscr{T}$ particularly well-suited to stress-testing the boundary between model classes, since the experimenter can calibrate the difficulty to the regime of interest rather than accepting an arbitrary fixed landscape.

F. Why Linear Regression Succeeds: A First-Order Explanation

Let $\boldsymbol{\mu}_B = \mathbb{E}[\boldsymbol{X}_{\text{Build}}]$ denote the expected Build-window path and write $\boldsymbol{X}_{\text{Build}} = \boldsymbol{\mu}_B + \epsilon$, where ϵ is mean-zero noise with variance $\mathcal{O}(N^{-H})$ under the fBm scaling law. A first-order Taylor expansion of $\mathbb{E}[\mathscr{T} \mid \mathcal{F}_{L_B}]$ around $\boldsymbol{\mu}_B$ gives,

$$\mathbb{E}[\mathscr{T} \mid \mathcal{F}_{L_B}] \;=\; \mathscr{T}(\boldsymbol{\mu}_B) \;+\; \nabla_{\boldsymbol{\mu}_B}\mathscr{T} \cdot \epsilon \;+\; \mathcal{O}\big(N^{-2H}\big)\,,$$

where the gradient $\nabla_{\boldsymbol{\mu}_B}\mathscr{T}$ is a fixed vector of linear coefficients in the Build-window moments $\{\bar{\boldsymbol{x}}_{\text{Build}}, \boldsymbol{\sigma}_{\text{loc}}\}$. As $N \to \infty$, the remainder $\mathcal{O}(N^{-2H}) \to 0$ for all $H \in (0,1)$, so the conditional expectation of $\mathscr{T}$ becomes *asymptotically linear* in the feature vector. Linear Regression therefore converges to the optimal predictor in the large-N limit, explaining the observed $R^2 \approx 0.96$ without appeal to architectural coincidence; the nonlinearity of $\mathscr{T}$ is real, but its *conditional* nonlinearity given the Build-window filtration $\mathcal{F}_{L_B}$ vanishes as the path noise concentrates around its mean under fBm regularity.

References

[1] R. Brown, "A brief account of microscopical observations made in the months of june, july and august, 1827, on the particles contained in the pollen of plants," *Philosophical Magazine*, vol. 4, pp. 161–173, 1828.

[2] A. Einstein, "On the movement of small particles suspended in stationary liquids required by the molecular-kinetic theory of heat," *Annalen der Physik*, vol. 17, pp. 549–560, 1905, translated in *Investigations on the Theory of the Brownian Movement*, Dover Publications.

[3] N. Wiener, *Differential Space*. Cambridge University Press, 1923.

[4] A. Taranto, B. Nunes, and R. Addie, "Survey of Continuous Ant Colony Optimization: Theory, Applications and Algorithms," *Journal of Computer Science*, pp. 1–42, 2025, https://doi.org/10.36227/techrxiv.175693561.14761595/v1.

[5] A. Taranto and R. Addie, "Survey of Loss Lanscape Surfaces: Theory, Applications and Algorithms," *engrXiv*, pp. 1–45, 2025, https://engrxiv.org/preprint/view/5069.

[6] A. Taranto, R. Addie, and B. Nunes, "Brownian Bridge - CACO (BB-CACO)," *TechRxiv*, pp. 1–31, 2025, https://www.techrxiv.org/users/959089/articles/1332208-brownian-bridge-continuous-ant-colony-optimization-bb-caco.

[7] A. Taranto and R. Addie, "Greedy Brownian - Continuous Ant Colony Optimization (GB-CACO)," *engrXiv*, pp. 1–35, 2025, https://doi.org/10.31224/5854.

[8] ——, "Itô Bridge - CACO (IB-CACO)," *TechRxiv*, pp. 1–31, 2025, https://www.techrxiv.org/users/959089/articles/.

[9] W. Bock, J. B. Bornales, C. O. Cabahug, T. Fattler, and L. Streit, "Fractional brownian motion - Some recent results and generalizations," *AIP Conf. Proc.*, vol. 2286, no. 1, pp. 1–9, 2020, https://pubs.aip.org/aip/acp/article-pdf/doi/10.1063/5.0029699/14218516/020001_1_online.pdf.

[10] K. Bisewski, K. Debicki, and M. Mandjes, "Bounds for expected supremum of fractional Brownian motion with drift," *arXiv*, vol. arXiv:2005.04919v4, pp. 1–16, 2021, https://arxiv.org/pdf/2005.04919.

[11] C. Vardar, "Results on the supremum of fractional Brownian motion," *arXiv*, vol. arXiv:0910.5193v3, pp. 1–14, 2012, https://arxiv.org/pdf/0910.5193.

[12] M. Çağlar and C. Vardar-Acar, "Distribution of maximum loss for fractional Brownian motion," *arXiv: Probability*, p. null, 2012. [Online]. Available: https://www.semanticscholar.org/paper/998c93d5846e0fad816c61bc2d09ae890558d35b

[13] J. Chen, H. Bhatia, R. Addie, and M. Zukerman, "Statistical characteristics of queue with fractional Brownian motion input," *Electronics Letters*, vol. 51, no. 9, pp. 699–701, 2015, https://scholars.cityu.edu.hk/files/23427798/2_Statistical_characteristics_of_queue_with_fractional_Brownian_motion_input.pdf.

[14] B. Mandelbrot and J. van Ness, "Fractional Brownian motions, fractional noises and applications," *SIAM Review*, vol. 10, pp. 422–437, 1968, http://dx.doi.org/10.1137/1010093.

[15] A. Benassi, P. Bertrand, S. Cohen, and J. Istas, "Identification of the Hurst index of a step fractional Brownian motion," *Statistical Inference for Stochastic Processes*, vol. 3, pp. 101–111, 2000, http://dx.doi.org/10.1023/A:1009997729317.

[16] P. Salminen and P. Vallois, "On maximum increase and decrease of Brownian motion," *Annales De L Institut Henri Poincare-probabilites Et Statistiques*, vol. 43, pp. 655–676, 2005. [Online]. Available: https://www.semanticscholar.org/paper/e20e6f9cea5bbd8fde9d0dcf432efef704e9358f

[17] I. Norros, "Four approaches to the fractional Brownian storage," 1997. [Online]. Available: https://www.semanticscholar.org/paper/52df872cc278bdeff9f1f9288df076e442f6e9fd

[18] Y. Meyer, F. Sellan, and M. Taqqu, "Wavelets, generalized white noise and fractional integration: The synthesis of fractional Brownian motion," *Journal of Fourier Analysis and Applications*, vol. 5, pp. 465–494, 1999. [Online]. Available: https://www.semanticscholar.org/paper/adb2cb8c18db844b21b83f346f56fda836169484

[19] D. Hong, S. Man, J. Birget, and D. S. Lun, "A wavelet-based approximation of fractional Brownian motion with a parallel algorithm," *arXiv: Probability*, p. null, 2011. [Online]. Available: https://www.semanticscholar.org/paper/7477814e2a57d1ecc795b8ec85b20f02d916c21d

[20] A. H. Hamza and M. Hmood, "Comparison of Hurst exponent estimation methods," *Journal of Economics and Administrative Sciences*, vol. null, p. null, 2021. [Online]. Available: https://www.semanticscholar.org/paper/0926cb2eeddf6911e42beab00300b0e5311290a7

[21] L. Tan, "Exponential stability of fractional stochastic differential equations with distributed delay," *Advances in Difference Equations*, vol. 2014, p. null, 2014. [Online]. Available: https://www.semanticscholar.org/paper/2595e6405c85fe853f914c25453899bcb75cda65

[22] M. Abundo and E. Pirozzi, "On the fractional Riemann-Liouville integral of Gauss-Markov processes and applications," *arXiv: Probability*, p. null, 2019. [Online]. Available: https://www.semanticscholar.org/paper/ffa490b04368a67bf0d4dbc275331c872d84a2f2

[23] G. Ãœnal and S. BayracÄ±, "Bond and swap pricing with interest rates driven by fBm," 2012. [Online]. Available: https://www.semanticscholar.org/paper/dae6bb079c92c9b42b0b1fb9d0f75d45bd3a48e2

[24] V. Gisin and A. Markov, "Asset pricing in a fractional market under transaction costs," 2012. [Online]. Available: https://www.semanticscholar.org/paper/2101a9f2c4a137cd1002dbbbf7b98723113540b5

[25] C. Necula, "A framework for derivative pricing in the fractional Black-Scholes market," *Behavioral Experimental Finance*, vol. null, p. null, 2007. [Online]. Available: https://www.semanticscholar.org/paper/3db68818c3d16b054fae3b41e8edc9988cbc3f47

[26] Y. Sarol, F. Viens, and T. Zhang, "Portfolio optimization with consumption in a fractional Black-Scholes market," 2007. [Online]. Available: https://www.semanticscholar.org/paper/f36e3efdb7a643a48308cf74b1e3f755798dab63

[27] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.

[28] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[29] A. Auger and N. Hansen, "Performance evaluation of optimization algorithms: A unified view," in *Proceedings of the 2009 IEEE Congress on Evolutionary Computation*, 2009, pp. 3999–4006.

[30] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.

[31] F. Hutter, L. Kotthoff, and J. Vanschoren, *Automated Machine Learning: Methods, Systems, Challenges*. Springer, 2019.
[32] M. S. Taqqu and V. Teverovsky, "Estimating long-range dependence in finite and infinite variance time series," *Fractals*, vol. 5, no. 1, pp. 1–22, 1997.
[33] J.-M. Bardet, G. Lang, G. Oppenheim, and M. S. Taqqu, "Wavelet methods for estimating long-range dependence in time series," *Journal of Time Series Analysis*, vol. 19, no. 5, pp. 453–475, 1998.
[34] H. Barnhard, "Approximating heavy traffic with Brownian motion: The supremum of brownian motion with negative drift," *Undergraduate Research Paper, Department of Mathematics, University of Chicago*, 2018, example 3.14 on page 17: "the supremum of a Brownian motion with negative drift is exponentially distributed, with $P\{W \geq w\} = \exp\{-2|\alpha|w/\sigma^2\}$".
[35] M. Delorme and K. J. Wiese, "Extreme-value statistics of fractional brownian motion bridges," *Physical Review E*, vol. 94, no. 1, p. 012134, 2016.
[36] G. M. Molchan, "Maximum of a fractional brownian motion: probabilities of small values," *Communications in Mathematical Physics*, vol. 205, pp. 97–111, 1999.
[37] M. Arutkin and K. J. Wiese, "Fractional brownian motion with absorbing boundaries: Exact results," *Physical Review E*, vol. 102, no. 3, p. 032107, 2020.

Dr. Aldo Taranto Aldo is pursuing a second PhD while conducting postdoctoral-level research at the Australian National University (ANU) in advanced optimization techniques for high-dimensional machine learning, under an Australian Defence innovation scholarship. He is currently Director of AI Research & Development at MetaModelR Corporation. Aldo holds a BSc(Math) from Monash University (1996), GradDipEd from University of Melbourne (1997), MBSys from Monash University (1998), MB(Acc) from RMIT University (2006) and was awarded a PhD(Math) from the University of Southern Queensland (2022), for his research in stochastic differential equations and their application in mathematical finance and algorithmic trading.

A/Prof. Ron Addie Ron is an Adjunct Associate Professor at the University of Southern Queensland (UniSQ). He began his research career at Telstra Research Laboratories, where he completed a PhD in Markov Additive Processes and co-developed virtual paths —now foundational to ATM broadband networks. He also advanced performance models for Gaussian traffic in core networks. Joining UniSQ in 1993, he served as Head of Mathematics and Computing (2004–2006), taught across multiple IT and science programs, and supervised over 10 PhD students. His Netml software supported 1000+ students and 19+ publications over 15 years. Though retired in 2022, he remains active in research and postgraduate supervision.