



Classical versus Data-Driven Approximation of a Damped Oscillatory Function: Convergence Rates, Uniform Error, and Extrapolation Behaviour

Marwa Abdulfattah Almabrouk¹

¹ Assistant Lecturer in Mathematics, Department of Administrative Sciences, Higher Institute of Science and Technology, Sabratha, Libya

Corresponding author: Marwa Abdulfattah Almabrouk, e-mail: [mma647591@gmail.com](mailto:mmma647591@gmail.com)

Received: 10/8/2026 Revised: 13/8/2026 Accepted: 05/9/2026 Publication: 05/10/2026

Abstract

We study the approximation of the entire function $f(x) = e^{-x}\sin(2\pi x) + 0.3\cos(5x)$ on $[0,5]$ from 5,000 exact, equally spaced samples. A random forest and two multilayer perceptrons (MLPs), regressors now widely used as general-purpose approximators, are compared with classical approximants: least-squares polynomials in the monomial and Chebyshev bases, piecewise-linear interpolation and cubic-spline interpolation. Accuracy is assessed by MAE, RMSE and R^2 and by the uniform error $E_\infty = \|f - \hat{g}\|_\infty$, and the observed behaviour is related to a-priori error estimates. Under an 80/20 random partition, the random forest achieved the lowest test error among the original data-driven configurations ($R^2 = 0.999988$, $E_\infty = 6.19 \times 10^{-3}$), ahead of a degree-8 polynomial ($R^2 = 0.377$) and an MLP trained with early stopping ($R^2 = 0.901$). On the same data, however, piecewise-linear interpolation, cubic-spline interpolation and a degree-40 Chebyshev polynomial reached $E_\infty = 7.73 \times 10^{-5}$, 2.62×10^{-9} and 5.03×10^{-14} , respectively. Under mesh refinement, the observed convergence rates were 0.87 (random forest), 2.00 (piecewise-linear) and 4.06 (cubic spline), consistent with the $O(h)$, $O(h^2)$ and $O(h^4)$ estimates, whereas the Chebyshev error decayed super-geometrically with the degree. The degree-8 result was sensitive to the ill-conditioning of the monomial basis, MLP accuracy depended strongly on the optimiser, and no method extrapolated reliably beyond the sampled interval ($R^2 \leq 0.41$ on $(4,5]$ for all methods). A coefficient of determination close to one therefore indicates accurate in-domain interpolation, not approximation-theoretic superiority.

Keywords: function approximation; uniform error; convergence rate; Chebyshev polynomials; cubic-spline interpolation; random forest; multilayer perceptron; extrapolation

1. Introduction

Approximating a function by simpler objects is one of the central problems of mathematical analysis. The Weierstrass theorem guarantees that every continuous function on a compact interval is a uniform limit of polynomials [1], and a mature theory quantifies how fast polynomial, spline and other classical approximants converge in terms of the smoothness of the target [2–4]. In parallel, machine learning has popularised flexible regressors — tree ensembles and neural networks in particular — that are increasingly used as general-purpose function approximators and as surrogates for expensive numerical models [5–7].

Neural networks are frequently motivated by universal approximation theorems [8–10], whose modern refinements address vector-valued outputs, constructive proofs and approximation rates for smooth functions [11–14]. These are existence results: they assert that a sufficiently large network *can* approximate a continuous function, not that a particular finite network trained by a particular optimiser *will* do so, nor how its accuracy compares with that of classical methods for the same amount of data [15]. Tree ensembles [16, 17] approximate in yet another way, by adaptive piecewise-constant partitions.

Comparisons between such regressors are commonly reported through test-set statistics such as R^2 . From the standpoint of approximation theory this practice raises three questions. First, how do data-driven approximants compare with classical ones *on exactly the same samples*? Second, do their errors decrease at predictable rates as the data are refined, and do those rates agree with what theory predicts? Third, how much of the reported accuracy survives when the evaluation is moved from interpolation to extrapolation, or from an averaged metric to the uniform norm?

We address these questions on a deliberately transparent benchmark: a closed-form entire function, sampled without noise on a compact interval. Because the target is known exactly, every error can be measured rather than estimated. The study extends an earlier comparison of three regressors (a degree-8 polynomial, a random forest and a multilayer perceptron), which is retained here and placed in its mathematical context.

Contributions. (i) A reproducible benchmark in which data-driven and classical approximants are fitted to identical samples and assessed by both statistical metrics and the uniform norm. (ii) Elementary a-priori error estimates for each scheme, including a bound for random forests fitted to noise-free data (Proposition 2), and their numerical verification under mesh refinement. (iii) A diagnosis of two pitfalls that affect conclusions drawn from such studies: loss of accuracy caused by an ill-conditioned polynomial basis, and dependence of neural-network rankings on the optimiser. (iv) An extrapolation test showing that high interpolation accuracy does not transfer outside the sampled interval.

The paper is organised as follows. Section 2 fixes notation and establishes the analytic properties of the target. Section 3 reviews the approximation schemes and their error estimates. Section 4 describes the experimental protocols, Section 5 reports the results and Section 6 discusses them. Section 7 lists limitations and Section 8 concludes. Proofs are collected in Appendix A.

2. Problem Setting

2.1 The target function

Throughout, $I = [a, b] = [0, 5]$ and

$$f(x) = g(x) + h(x), \quad g(x) = e^{-x} \sin(2\pi x), \quad h(x) = 0.3 \cos(5x), \quad x \in I. \quad (1)$$

The component g is an exponentially damped oscillation with five periods on I and envelope $\pm e^{-x}$; the component h is an undamped oscillation of smaller amplitude with about four periods on I . Near the origin both components contribute, whereas for larger x the cosine dominates (Figure 1). The superposition is therefore smooth but non-stationary: its amplitude and local curvature vary across the interval.

Writing $z = -1 + 2\pi i$, we have $g(x) = \text{Im} e^{zx}$, which yields the following elementary facts (the proof is given in Appendix A).

Proposition 1. *The function f extends to an entire function of $x \in \mathbb{C}$. For every integer $k \geq 0$ and every $x \in I$,*

$$|f^{(k)}(x)| \leq (1 + 4\pi^2)^{k/2} e^{-x} + 0.3 \cdot 5^k. \quad (2)$$

In particular, $\|f'\|_\infty \leq 7.87$, $\|f''\|_\infty \leq 47.98$ and $\|f^{(4)}\|_\infty \leq 1826.1$. Evaluating the derivatives on a grid of 2×10^5 points gives the sharper values $\|f\|_\infty = 0.9435$, $\|f'\|_\infty = 2\pi$ (attained at $x = 0$), $\|f''\|_\infty = 38.66$ and $\|f^{(4)}\|_\infty = 1583.6$, which are used in the error estimates of Section 3.

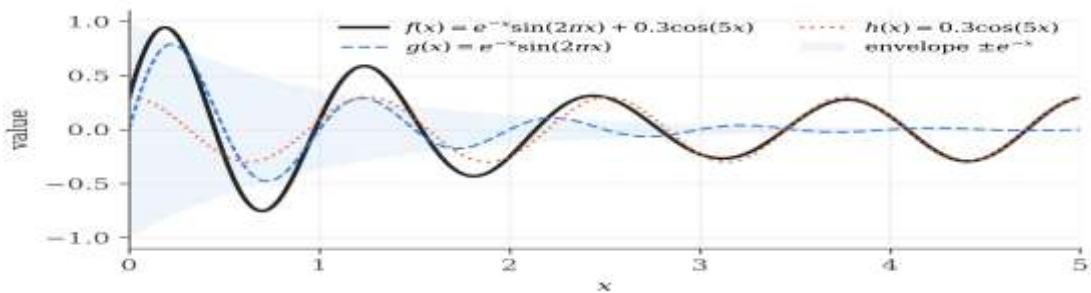


Figure 1. The target function f and its components g and h on $[0, 5]$; the shaded band is the envelope $\pm e^{-x}$ of g .

2.2 Data and error measures

The data consist of $N = 5000$ equally spaced nodes and exact values,

$$x_i = a + (i - 1) \frac{b - a}{N - 1}, \quad y_i = f(x_i), \quad i = 1, \dots, N, \quad (3)$$

so that the grid spacing is $\Delta x = 5/4999 \approx 1.0 \times 10^{-3}$. Given a training set $\{(x_i, y_i)\}_{i \in \mathcal{T}}$, each method produces an approximant $\hat{g}$. On a test set $\{(x_i, y_i)\}_{i \in \mathcal{V}}$ with $n = |\mathcal{V}|$ we report the customary statistical metrics

$$\text{MAE} = \frac{1}{n} \sum_{i \in \mathcal{V}} |y_i - \hat{g}(x_i)|, \quad \text{RMSE} = \left(\frac{1}{n} \sum_{i \in \mathcal{V}} (y_i - \hat{g}(x_i))^2 \right)^{1/2}, \quad (4)$$

$$R^2 = 1 - \frac{\sum_{i \in \mathcal{V}} (y_i - \hat{g}(x_i))^2}{\sum_{i \in \mathcal{V}} (y_i - \bar{y})^2}, \quad (5)$$

where $\bar{y}$ is the mean of the test responses. Since $1 - R^2$ is simply the squared RMSE divided by the test variance, it is reported alongside R^2 when R^2 is indistinguishable from one. The quantity of primary interest in approximation theory, however, is the uniform error

$$E_\infty(\hat{g}) = \|f - \hat{g}\|_{\infty, J} = \max_{x \in J} |f(x) - \hat{g}(x)|, \quad (6)$$

where J is the convex hull of the training nodes. We approximate E_∞ by the maximum over 20,001 equally spaced points of J ; this grid is five times finer than the data, so the estimate also probes the behaviour of $\hat{g}$ *between* the samples.

3. Approximation Schemes and Error Estimates

3.1 Global polynomials

Let $\mathcal{P}_n$ denote the polynomials of degree at most n , and map I to $[-1, 1]$ by $t = (2x - a - b)/(b - a)$. A least-squares polynomial $p_n \in \mathcal{P}_n$ minimises $\sum_{i \in \mathcal{T}} (y_i - p_n(x_i))^2$. Mathematically p_n does not depend on the basis in which it is expressed, but numerically it does. In the monomial basis the design matrix is the Vandermonde matrix $V_{ij} = x_i^j$, whose condition number grows exponentially with n [18]. For $n = 8$ on our nodes, $\kappa_2(V) \approx 4.79 \times 10^7$ (and $\kappa_2 \approx 4.25 \times 10^7$ after the column centring used by common regression software), whereas the Chebyshev–Vandermonde matrix $T_{ij} = T_j(t_i)$ has $\kappa_2(T) \approx 3.34$.

The rate at which p_n can converge is governed by the analyticity of f . If f is analytic and bounded by M inside the Bernstein ellipse E_ρ with foci ± 1 and semi-axis sum $\rho > 1$, then the Chebyshev truncation f_n satisfies [2, Thm. 8.2]

$$\|f - f_n\|_{\infty, [-1, 1]} \leq \frac{2M\rho^{-n}}{\rho - 1}. \quad (7)$$

Since f is entire (Proposition 1), Eq. (7) holds for every $\rho > 1$ with $M = M(\rho)$, and the best approximation error therefore decays faster than any geometric sequence. The bound is asymptotic in character: before a polynomial can resolve the oscillations of f , its degree must exceed a threshold proportional to the number of wavelengths in the interval (roughly π degrees of freedom per wavelength [2, 4]). With about five wavelengths on I , rapid convergence cannot be expected below $n \approx 16$; a degree-8 polynomial is necessarily in the pre-asymptotic regime.

3.2 Piecewise polynomial interpolation

Let the sorted training nodes have maximal spacing h . The piecewise-linear interpolant $I_h f$ and the cubic-spline interpolant s_h satisfy the classical estimates [3, 19]

$$\|f - I_h f\|_\infty \leq \frac{h^2}{8} \|f''\|_\infty, \quad \|f - s_h\|_\infty \leq C h^4 \|f^{(4)}\|_\infty. \quad (8)$$

For the complete (clamped) spline the optimal constant is $C = 5/384$ [19]; for the not-a-knot spline used here the order h^4 is the same but the constant is larger, chiefly because of the end conditions [3]. Both schemes are local: the error at x depends only on f in a neighbourhood of size $O(h)$.

3.3 Random forests as adaptive piecewise-constant approximants

A regression tree partitions I into intervals (leaves) and predicts, on each leaf, the mean of the training responses that fall into it. A random forest averages B such trees grown on bootstrap samples [16],

$$\hat{g}_{\text{RF}}(x) = \frac{1}{B} \sum_{b=1}^B T_b(x). \quad (9)$$

For noise-free data, the following bound holds (the proof is given in Appendix A).

Proposition 2. *Let f be Lipschitz continuous on I with constant $L = \|f'\|_\infty$, and let the observations be noise-free, $y_i = f(x_i)$. Let $\hat{g}_{\text{RF}}$ be the average (9) of B trees, each of which predicts, on every leaf, the mean of the (bootstrap) training responses in that leaf. If every leaf containing x , in every tree, has diameter at most $\delta(x)$, then*

$$|f(x) - \hat{g}_{\text{RF}}(x)| \leq L \delta(x). \quad (10)$$

When the leaves shrink in proportion to the data spacing, Eq. (10) predicts first-order convergence, $E_\infty = O(h)$. This rate is intrinsic to piecewise-constant approximation and cannot be improved by adding trees: averaging reduces the variance of the partition but each tree remains a step function. The forest is, in this sense, a randomised relative of nearest-neighbour interpolation.

3.4 Multilayer perceptrons

A perceptron with two hidden layers of width m and activation σ computes

$$\hat{g}_{\text{MLP}}(x) = W_3 \sigma(W_2 \sigma(W_1 x + b_1) + b_2) + b_3, \quad (11)$$

with $W_1 \in \mathbb{R}^{m \times 1}$, $W_2 \in \mathbb{R}^{m \times m}$ and $W_3 \in \mathbb{R}^{1 \times m}$. Universal approximation theorems [8–10] guarantee that functions of this form are dense in $C(I)$, but the error actually obtained decomposes into an approximation part, which depends on the architecture, and an optimisation part, which depends on the training algorithm and its stopping rule [15]. Gradient-trained networks also exhibit a spectral bias towards low frequencies [20], which is relevant for an oscillatory target. Unlike the methods of Sections 3.1–3.3, no a-priori rate in h is available for a network trained by a stochastic optimiser.

4. Numerical Experiments

4.1 Methods and configurations

Table 1 lists the eight approximants. The first three reproduce the original study; the remaining five were added to place it in context. MLP-B uses the same architecture as MLP-A but a quasi-Newton optimiser (L-BFGS [21]) without early stopping, isolating the effect of training from that of capacity.

Table 1. Approximants and their configurations.

Label	Method	Configuration
Mono-8	Least-squares polynomial, monomial basis	degree 8; solver default relative singular-value cut-off 10^{-6}
Cheb-8	Least-squares polynomial, Chebyshev basis	degree 8 (same polynomial space as Mono-8)
Cheb-40	Least-squares polynomial, Chebyshev basis	degree 40
PL	Piecewise-linear interpolation	nodes = training points
CS	Cubic-spline interpolation	not-a-knot end conditions
RF	Random forest [16]	$B = 200$ trees, maximum depth 18
MLP-A	Multilayer perceptron, Adam optimiser [22]	ReLU, widths (64, 64), standardised input, early stopping
MLP-B	Multilayer perceptron, L-BFGS optimiser [21]	tanh, widths (64, 64), standardised input, no early stopping

4.2 Protocols

Five protocols were used.

- **P1 (original partition).** The 5,000 samples are split at random into 4,000 training and 1,000 test points with seed 42, exactly as in the original study. The maximal gap between consecutive training nodes is $h = 9.0 \times 10^{-3}$.
- **P2 (repeated partitions).** P1 is repeated with seeds 0,1, ...,19; means and sample standard deviations are reported.
- **P3 (extrapolation).** All methods are trained on the 4,000 nodes in $[0,4]$ and tested on the 1,000 nodes in $(4,5]$.
- **P4 (mesh refinement).** The interpolatory methods, RF and MLP-B are trained on $N \in \{50,100, \dots, 6400\}$ equally spaced nodes (doubling each time), and E_∞ is measured on 20,001 points of $[0,5]$. The empirical order q is the least-squares slope of $\log E_\infty$ against $\log h$, with $h = 5/(N - 1)$.
- **P5 (degree sweep).** Chebyshev least-squares polynomials of degree $n = 2,4, \dots, 60$ are fitted to the P1 training set.

4.3 Implementation

All computations were carried out in double precision with NumPy, SciPy and scikit-learn [23]. Every random component has a fixed seed, and the dataset is regenerated from Eq. (3), so the results can be reproduced without external data (see *Data and Code Availability*).

5. Results

5.1 The original comparison and its reproducibility

Under P1, the random forest, MLP-A and the exact degree-8 least-squares polynomial (Cheb-8) give $\text{MAE} = 9.21 \times 10^{-4}$, 5.36×10^{-2} and 0.233; $\text{RMSE} = 1.25 \times 10^{-3}$, 0.115 and 0.288; and $R^2 = 0.999988$, 0.901394 and 0.377458, respectively. These values agree to six digits with those reported in the original study.

The degree-8 monomial pipeline (Mono-8), however, does *not* reproduce the original polynomial result under scikit-learn 1.9.1 (the original models were serialised with version 1.8.0): it gives $\text{RMSE} = 0.352$ and $R^2 = 0.0696$. The cause is numerical. The regression routine now passes a relative singular-value cut-off of 10^{-6} to its least-squares solver; with $\kappa_2 \approx 4.25 \times 10^7$, three of the eight singular values of the centred monomial design matrix fall below the cut-off and are discarded, so the computed solution has effective rank 5 instead of 8. Expressed in the Chebyshev basis the same least-squares problem is perfectly conditioned and returns the original value $R^2 = 0.377458$. The

reported performance of a “degree-8 polynomial” therefore depended on a solver tolerance, not only on the polynomial space.

5.2 Classical approximants on the same data

Table 2 adds the classical approximants, fitted to the same 4,000 training points. Figure 2 shows the three original approximants over the whole interval, and Figure 3 the local maximum of the pointwise error.

Table 2. Accuracy under protocol P1 (4,000 training and 1,000 test points). E_∞ is measured on 20,001 points of the training hull.

Method	MAE	RMSE	$1 - R^2$	E_∞
Mono-8	2.75×10^{-1}	3.52×10^{-1}	9.30×10^{-1}	8.99×10^{-1}
Cheb-8	2.33×10^{-1}	2.88×10^{-1}	6.23×10^{-1}	6.67×10^{-1}
MLP-A	5.36×10^{-2}	1.15×10^{-1}	9.86×10^{-2}	8.10×10^{-1}
MLP-B	2.75×10^{-3}	3.40×10^{-3}	8.63×10^{-5}	1.78×10^{-2}
RF	9.21×10^{-4}	1.25×10^{-3}	1.18×10^{-5}	6.19×10^{-3}
PL	7.46×10^{-6}	1.23×10^{-5}	1.12×10^{-9}	7.73×10^{-5}
CS	7.60×10^{-11}	2.28×10^{-10}	3.88×10^{-19}	2.62×10^{-9}
Cheb-40	5.10×10^{-15}	5.99×10^{-15}	2.69×10^{-28}	5.03×10^{-14}

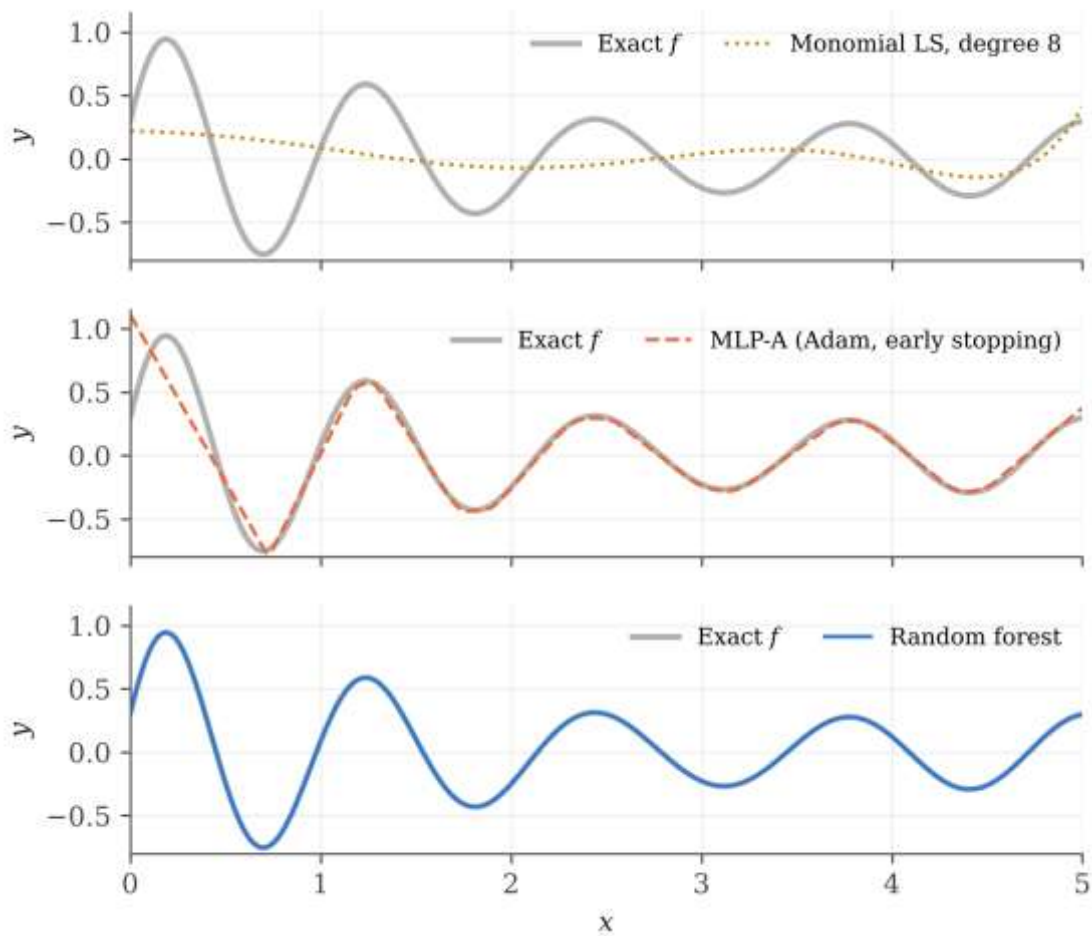


Figure 2. Approximants of the original study on $[0,5]$ (protocol P1): degree-8 monomial least squares (top), MLP-A (middle) and random forest (bottom). The exact function is shown in grey.

Three observations follow. First, the ranking of the original study (RF ahead of MLP-A, which is ahead of the polynomial) is confirmed, but all three are far less accurate than elementary interpolation: the uniform error of the random forest is about 80 times that of piecewise-linear interpolation, 2.4×10^6 times that of the cubic spline and 1.2×10^{11} times that of the degree-40 Chebyshev polynomial. Second, the observed errors respect the estimates of Section 3: with $h = 9.0 \times 10^{-3}$, Eq. (8) bounds the piecewise-linear error by $h^2 \|f''\|_\infty / 8 = 3.92 \times 10^{-4}$, and the measured value is 7.73×10^{-5} . Third, E_∞ separates methods that the statistical metrics do not. MLP-A has $R^2 = 0.901$, yet its uniform error (0.810) is comparable to that of the degree-8 polynomials; Figures 2 and 3 show that this error is concentrated near $x = 0$, where $|f'|$ is largest and the network has not resolved the first oscillation.

5.3 Robustness over repeated partitions

Table 3 summarises protocol P2. The random forest and the classical methods are insensitive to the partition. MLP-A is not: over 20 partitions its mean R^2 is 0.987 ± 0.025 , so the value 0.901 obtained with seed 42 is an unfavourable outcome rather than a typical one. Under seed 42 the early-stopping rule terminated training after 69 iterations. With the same architecture, MLP-B attains $R^2 = 0.99971 \pm 0.00023$; the gap between the two networks is thus caused by the optimiser and the stopping rule, not by the network's capacity.

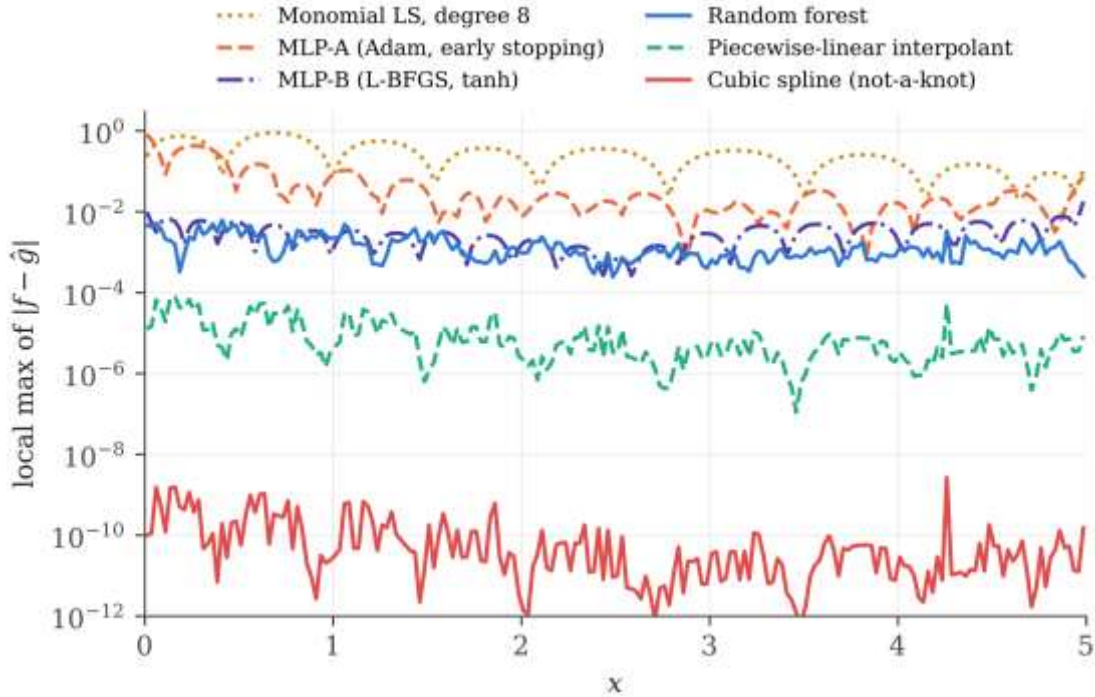


Figure 3. Local maximum of the pointwise error $|f - \hat{g}|$ over 200 consecutive windows of $[0, 5]$ (protocol P1), logarithmic scale.

Table 3. Mean $\pm$ standard deviation over 20 random 80/20 partitions (protocol P2). MaxAE is the largest absolute error over the test points.

Method	RMSE	MaxAE	R^2
Mono-8	0.343 ± 0.006	0.903 ± 0.008	0.077 ± 0.025
Cheb-8	0.292 ± 0.007	0.653 ± 0.033	0.329 ± 0.023
MLP-A	$(3.16 \pm 2.68) \times 10^{-2}$	0.287 ± 0.192	0.987 ± 0.025
MLP-B	$(5.61 \pm 2.39) \times 10^{-3}$	$(3.69 \pm 2.21) \times 10^{-2}$	0.99971 ± 0.00023
RF	$(1.30 \pm 0.09) \times 10^{-3}$	$(8.73 \pm 1.98) \times 10^{-3}$	0.999987 ± 0.000002
PL	$(4.04 \pm 6.84) \times 10^{-5}$	$(1.04 \pm 2.26) \times 10^{-3}$	$1 - 4.6 \times 10^{-8}$
CS	$(2.74 \pm 1.51) \times 10^{-10}$	$(4.04 \pm 2.93) \times 10^{-9}$	$1 - O(10^{-18})$
Cheb-40	$(6.09 \pm 0.19) \times 10^{-15}$	$(3.33 \pm 1.60) \times 10^{-14}$	$1 - O(10^{-28})$

The comparatively large spread of the piecewise-linear errors has a simple explanation. In 3 of the 20 partitions the endpoint $x = 0$ falls into the test set; the interpolant is then extended as a constant over one grid step, and the error at $x = 0$ becomes $6.27 \times 10^{-3} \approx |f'(0)| \Delta x$. In the remaining partitions, the maximal test error lies between 6.39×10^{-5} and 2.45×10^{-4} . Even in the worst case, this one-step extrapolation error is smaller than the typical error of the random forest.

5.4 Convergence under mesh refinement

Figure 4 shows E_∞ under protocol P4 together with reference slopes, and Table 4 gives the fitted orders. The interpolants follow theory closely: $q = 2.00$ for piecewise-linear interpolation and $q = 4.06$ for cubic-spline interpolation. The linear-interpolation error lies at or just below the bound $h^2 \|f''\|_\infty/8$ at every resolution (ratios between 0.989 and 1.000), so the classical constant is attained, as expected when the error is dominated by the point where $|f''|$ is largest. For the not-a-knot spline the ratio $E_\infty/(h^4 \|f^{(4)}\|_\infty)$ decreases from 0.028 at $N = 50$ to 0.0205 at $N = 6400$, larger than the clamped constant $5/384 \approx 0.013$, in line with the remark after Eq. (8).

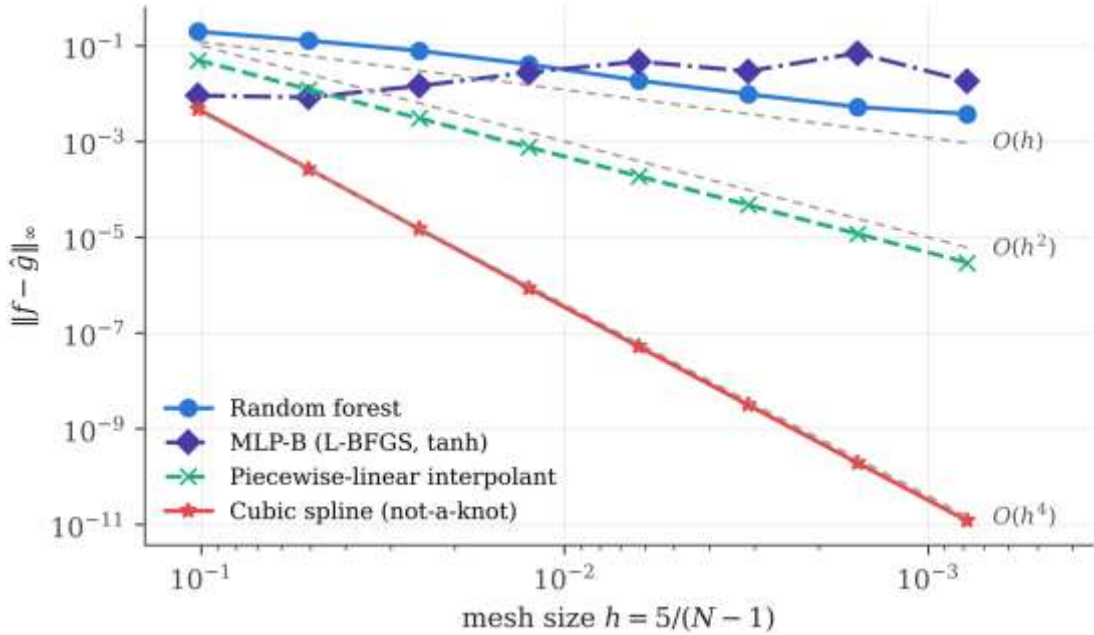


Figure 4. Uniform error against mesh size under protocol P4, with reference slopes $O(h)$, $O(h^2)$ and $O(h^4)$. Mesh size decreases to the right.

The random forest converges at the rate $q = 0.87$, close to the first order predicted by Proposition 2. Its error is the largest of all methods for $N \geq 200$, and halving it requires roughly doubling the data. MLP-B shows no convergence at all: its error stays between 8.3×10^{-3} and 7.0×10^{-2} and does not decrease monotonically with N . Beyond $N \approx 100$ the network error is limited by optimisation rather than by data, which is precisely the regime in which universal approximation results give no guidance.

Table 4. Empirical orders of convergence under protocol P4 (least-squares slope of $\log E_\infty$ against $\log h$, $N = 50, \dots, 6400$).

Method	Predicted order	Observed order q	E_∞ at $N = 50$	E_∞ at $N = 6400$
RF	1 (Prop. 2)	0.87	1.97×10^{-1}	3.76×10^{-3}
PL	2 (Eq. 8)	2.00	4.99×10^{-2}	2.95×10^{-6}
CS	4 (Eq. 8)	4.06	4.80×10^{-3}	1.21×10^{-11}
MLP-B	—	no convergence	9.07×10^{-3}	1.81×10^{-2}

5.5 Spectral convergence of polynomial approximation

Figure 5 shows the uniform error of the Chebyshev least-squares polynomials under protocol P5. Up to degree about 12 the error stays between 0.35 and 1: the polynomials cannot yet resolve the five wavelengths of f , and degree 8 — the choice of the original study — lies in this pre-asymptotic plateau. From $n \approx 14$ onwards, the error falls super-geometrically, from 1.86×10^{-1} at $n = 14$ to 1.36×10^{-4} at $n = 24$, 2.40×10^{-9} at $n = 32$ and 5.03×10^{-14} at $n = 40$, after which it reaches the rounding level of a few multiples of 10^{-15} . This is the behaviour predicted by Eq. (7) for an entire function. A polynomial with 41 coefficients thus approximates f to machine precision, whereas the random forest with 200 trees of depth up to 18 reaches 6.19×10^{-3} .

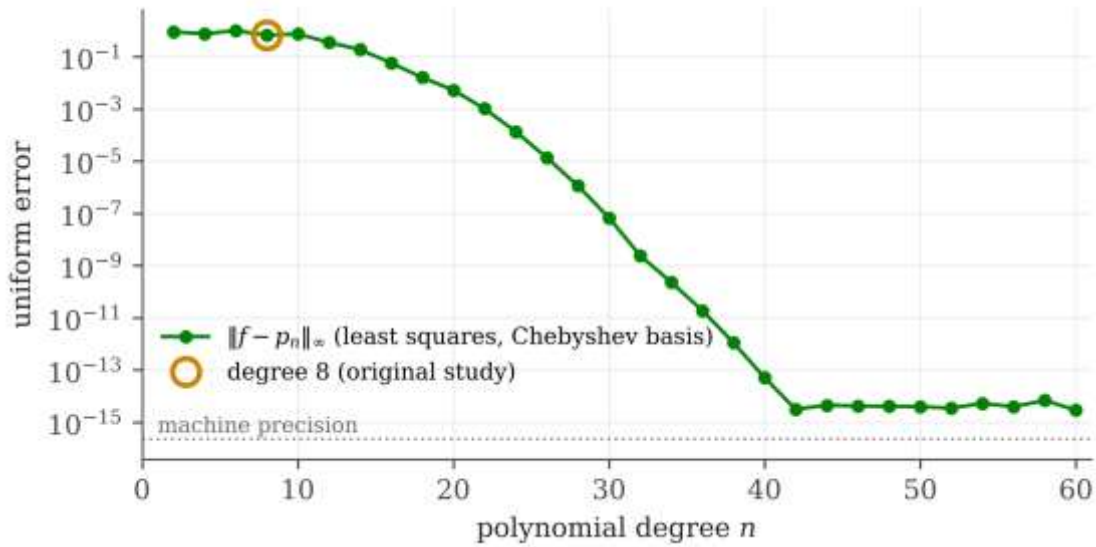


Figure 5. Uniform error of Chebyshev least-squares polynomials against degree (protocol P5). The circle marks degree 8, used in the original study.

5.6 Extrapolation

Table 5 and Figure 6 report protocol P3. Every method has $R^2 \leq 0.41$ on $(4,5]$, and all but one have negative R^2 . A negative value is a valid outcome of Eq. (5), not a computational error: it means that the squared test error of the model exceeds that of the constant prediction $\bar{y}$, the mean of the test responses. The random forest and the piecewise-linear interpolant both extend the last training value as a constant, which gives almost identical errors ($R^2 = -0.96$ and -0.97). The cubic spline and MLP-B

follow f for a short distance and then diverge. The degree-8 polynomials diverge rapidly, with maximum absolute errors of 20.1 (Mono-8) and 447.3 (Cheb-8); the difference between the two reflects the different, rank-deficient Mono-8 solution of Section 5.1. The degree-40 Chebyshev polynomial tracks f for about 0.6 units beyond the boundary and attains the best, though still poor, value $R^2 = 0.41$. Extrapolation of an oscillatory function from samples alone is ill-posed: none of the methods encodes the structure that would make it possible.

Table 5. Extrapolation accuracy under protocol P3 (training on $[0,4]$, testing on $(4,5]$).

Method	MAE	RMSE	R^2	MaxAE
Cheb-40	0.064	0.144	0.41	0.46
PL	0.226	0.262	-0.96	0.42
RF	0.226	0.263	-0.97	0.42
MLP-A	0.243	0.296	-1.51	0.66
CS	0.271	0.540	-7.31	1.91
MLP-B	0.414	0.587	-8.82	1.20
Mono-8	5.44	7.58	-1.6×10^3	20.1
Cheb-8	107.7	162.0	-7.5×10^5	447.3

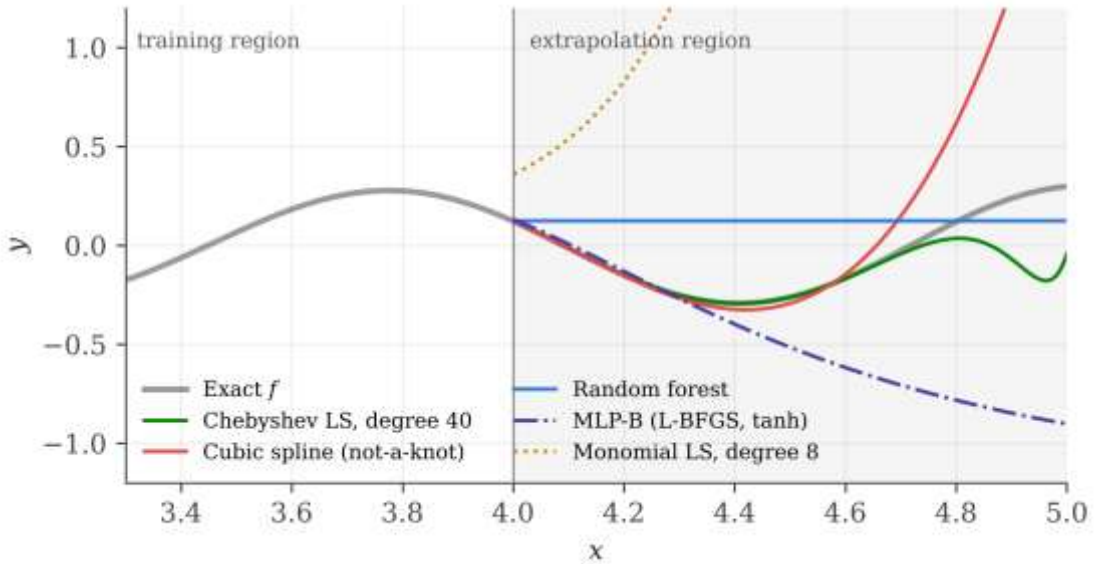


Figure 6. Extrapolation beyond the training interval (protocol P3). The shaded region $(4,5]$ contains no training data. The piecewise-linear interpolant coincides with the random forest and is omitted, as are MLP-A and Cheb-8 (see Table 5); the Mono-8 curve leaves the plotted range shortly after $x = 4$.

6. Discussion

6.1 What the random forest's accuracy means

The value of R^2 close to one obtained by the random forest is explained by the geometry of the experiment rather than by any special affinity between tree ensembles and the target. The problem is one-dimensional and noise-free, and under P1 every test point lies within 9×10^{-3} of a training point. Proposition 2 then bounds the error by $\|f'\|_\infty$ times the leaf width, and the measured first-order convergence confirms that this is the operative mechanism. The forest is an accurate in-domain interpolant because the data are dense; it converges at the slowest rate of all the methods studied, it stores 444,737 leaf values (200 trees) where the cubic spline uses about 16,000 piecewise coefficients and the Chebyshev polynomial 41, and it does not extrapolate. It also does not recover the analytic

form of f . These remarks do not diminish the practical value of random forests for noisy, high-dimensional or heterogeneous data, where classical interpolation is unavailable; they show that a one-dimensional smooth benchmark is not a setting in which they can be expected to compete with numerical analysis.

6.2 The basis matters numerically

Mono-8 and Cheb-8 represent the same polynomial space, yet they returned different approximants because the monomial basis is ill-conditioned on $[0,5]$. That a solver tolerance could change R^2 from 0.377 to 0.0696 — and change the extrapolated values by an order of magnitude — illustrates a general principle: polynomial models should be expressed in an orthogonal or well-conditioned basis on the mapped interval, as is standard in numerical analysis [2, 18]. Studies that report a polynomial baseline in raw powers of x risk attributing to the model class a failure that belongs to the linear algebra.

6.3 Capacity versus optimisation in neural networks

MLP-A and MLP-B share the same architecture, so they also share the same approximation capacity. Their order-of-magnitude difference in accuracy, and the large variance of MLP-A across partitions, are optimisation effects: early stopping on a validation subset, the first-order optimiser and the spectral bias of gradient training [20] together leave the steepest part of f near $x = 0$ unresolved. Even MLP-B does not improve as the data are refined (Section 5.4). This is consistent with the error decomposition emphasised in the numerical analysis of learned approximants [15]: a small approximation error, guaranteed by universality, says nothing about the training error actually achieved. Rankings of neural networks against other methods should therefore be accompanied by a description of the optimiser and by results over several initialisations.

6.4 What R^2 does and does not certify

R^2 is a normalised mean-squared error on a finite test set. It cannot distinguish an error of 10^{-5} from one of 10^{-15} once both are rounded to six digits; it averages away localised failures such as the boundary error of MLP-A; and under a random partition of dense data it measures interpolation, not generalisation. The uniform error over a grid finer than the data, the empirical order of convergence and an extrapolation test are inexpensive complements that place a data-driven approximant on the same scale as the classical ones.

6.5 Implications for surrogate modelling

When a target is smooth, low-dimensional and cheap to sample, classical approximation is both more accurate and more economical than generic regressors, and comes with error guarantees. Data-driven surrogates earn their place when one or more of these conditions fails — high input dimension, noisy or scattered data, or targets with no exploitable smoothness [5, 6]. The present benchmark suggests a simple discipline for such studies: always include a classical baseline that matches the structure of the problem, and report its error in the same norm.

7. Limitations and Further Work

The conclusions are drawn from a single analytic target in one dimension, sampled without noise. Functions with limited smoothness (e.g. kinks or discontinuities), non-uniform or scattered nodes, measurement noise and higher input dimension will change the balance between methods, and in particular the advantage of interpolation over regression. The data-driven methods were not tuned exhaustively; a systematic search, or networks with Fourier-feature inputs, could improve their

accuracy. Computational cost was not measured systematically. Natural extensions include (i) noisy data, where interpolation must be replaced by smoothing splines or regularised least squares and the comparison becomes one of bias and variance; (ii) multivariate targets, where tensor-product and sparse-grid methods compete with tree ensembles and networks; (iii) probabilistic surrogates such as Gaussian processes, which provide error bars [7]; and (iv) learning methods that incorporate known structure, such as periodic or damped-oscillatory bases, which may address the extrapolation failure documented in Section 5.6.

8. Conclusions

We compared data-driven and classical approximants of the damped oscillatory function f in Eq. (1) on $[0,5]$, using identical, noise-free samples. Under the selected dense, noise-free, one-dimensional interpolation protocol, the random forest achieved the lowest test error among the original data-driven configurations ($R^2 = 0.999988$), and this result is reproducible. Classical interpolation was nevertheless more accurate on this smooth one-dimensional target: piecewise-linear interpolation, cubic-spline interpolation and the degree-40 Chebyshev polynomial reduced the uniform error by factors of about 10^2 , 10^6 and 10^{11} , respectively. Their errors followed the predicted rates ($O(h^2)$, $O(h^4)$ and super-geometric convergence), whereas the random forest converged at first order and MLP-B did not converge under refinement. The poor degree-8 result was partly caused by an ill-conditioned basis, the network ranking reflected the optimiser rather than the architecture, and no method extrapolated reliably. A value of R^2 close to one should therefore be read as evidence of accurate in-domain interpolation; claims of approximation quality also require the uniform error, convergence rates and a classical baseline.

Data and Code Availability

All data are generated deterministically from Eq. (3); no external, proprietary or personal data are used. The generated data, Python scripts, numerical results and trained models used to reproduce the experiments are available in the accompanying reproducibility package.

Appendix A. Proofs

Proposition 1. *The function f extends to an entire function of $x \in \mathbb{C}$, and the derivative bound (2) holds for every integer $k \geq 0$ and every $x \in I$.*

Proof. The functions e^{zx} and $\cos(5x)$ are entire, and $g = \text{Im}e^{zx} = (e^{zx} - e^{\bar{z}x})/(2i)$ is a combination of entire functions; hence $f = g + h$ is entire. For real x , $g^{(k)}(x) = \text{Im}(z^k e^{zx})$, so $|g^{(k)}(x)| \leq |z|^k e^{\text{Re}(z)x} = (1 + 4\pi^2)^{k/2} e^{-x}$. Moreover, $h^{(k)}(x) = 0.3 \cdot 5^k \cos(5x + k\pi/2)$, so $|h^{(k)}(x)| \leq 0.3 \cdot 5^k$. The triangle inequality gives Eq. (2), and the numerical bounds follow from $e^{-x} \leq 1$ on I .

Proposition 2. *Under the assumptions stated in Section 3.3 (Lipschitz continuity of f with constant L , noise-free observations, leaves of diameter at most $\delta(x)$ and averaging over B trees), the error bound (10) holds.*

Proof. Fix $x \in I$ and a tree T_b . Let ℓ be the leaf containing x and let $S \subset \ell$ be the (non-empty) multiset of bootstrap training nodes in ℓ . Since the observations are exact, $T_b(x) = |S|^{-1} \sum_{x_j \in S} f(x_j)$ is a convex combination of values of f at points of ℓ . For every such point, $|f(x_j) - f(x)| \leq$

$L|x_j - x| \leq L \operatorname{diam}(\ell) \leq L \delta(x)$; hence $|T_b(x) - f(x)| \leq L \delta(x)$. Averaging over $b = 1, \dots, B$ and applying the triangle inequality gives Eq. (10).

Remark. Proposition 2 uses only the Lipschitz continuity of f and the absence of noise. With noisy responses, an additional stochastic term appears whose size depends on the number of samples per leaf; this is the term that averaging over trees is designed to reduce [17].

References

- [1] K. Weierstrass, “Über die analytische Darstellbarkeit sogenannter willkürlicher Functionen einer reellen Veränderlichen,” *Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften zu Berlin*, pp. 633–639 and 789–805, 1885.
- [2] L. N. Trefethen, *Approximation Theory and Approximation Practice*, extended ed. Philadelphia, PA, USA: SIAM, 2019, doi: 10.1137/1.9781611975949.
- [3] C. de Boor, *A Practical Guide to Splines*, rev. ed. New York, NY, USA: Springer, 2001.
- [4] J. P. Boyd, *Chebyshev and Fourier Spectral Methods*, 2nd ed. Mineola, NY, USA: Dover, 2001.
- [5] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, “Physics-informed machine learning,” *Nature Reviews Physics*, vol. 3, pp. 422–440, 2021, doi: 10.1038/s42254-021-00314-5.
- [6] Z. Azarhoosh and M. Ilchi Ghazaan, “A review of recent advances in surrogate models for uncertainty quantification of high-dimensional engineering applications,” *Computer Methods in Applied Mechanics and Engineering*, vol. 433, Art. no. 117508, 2025, doi: 10.1016/j.cma.2024.117508.
- [7] A. Marrel and B. Iooss, “Probabilistic surrogate modeling by Gaussian process: A review on recent insights in estimation and validation,” *Reliability Engineering & System Safety*, vol. 247, Art. no. 110094, 2024, doi: 10.1016/j.ress.2024.110094.
- [8] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989, doi: 10.1007/BF02551274.
- [9] K. Hornik, “Approximation capabilities of multilayer feedforward networks,” *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991, doi: 10.1016/0893-6080(91)90009-T.
- [10] A. Pinkus, “Approximation theory of the MLP model in neural networks,” *Acta Numerica*, vol. 8, pp. 143–195, 1999.
- [11] M. T. Augustine, “A survey on universal approximation theorems,” arXiv:2407.12895, 2024.
- [12] T. Bui-Thanh, “A unified and constructive framework for the universality of neural networks,” *IMA Journal of Applied Mathematics*, vol. 89, no. 1, pp. 197–230, 2024, doi: 10.1093/imamat/hxad032.
- [13] M. E. Valle, W. L. Vital, and G. Vieira, “Universal approximation theorem for vector- and hypercomplex-valued neural networks,” *Neural Networks*, vol. 180, Art. no. 106632, 2024, doi: 10.1016/j.neunet.2024.106632.
- [14] Z. Li, H. Li, J. Wang, and K. Tang, “Approximation to smooth functions by low-rank Swish networks,” in *Proc. 42nd Int. Conf. Machine Learning (ICML)*, PMLR vol. 267, pp. 35259–35291, 2025.
- [15] T. De Ryck and S. Mishra, “Numerical analysis of physics-informed neural networks and related models in physics-informed machine learning,” *Acta Numerica*, vol. 33, pp. 633–713, 2024, doi: 10.1017/S0962492923000089.
- [16] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
- [17] G. Biau and E. Scornet, “A random forest guided tour,” *TEST*, vol. 25, no. 2, pp. 197–227, 2016, doi: 10.1007/s11749-016-0481-7.
- [18] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed. Baltimore, MD, USA: Johns Hopkins Univ. Press, 2013.
- [19] C. A. Hall and W. W. Meyer, “Optimal error bounds for cubic spline interpolation,” *Journal of Approximation Theory*, vol. 16, no. 2, pp. 105–122, 1976, doi: 10.1016/0021-9045(76)90040-X.
- [20] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, “On the spectral bias of neural networks,” in *Proc. 36th Int. Conf. Machine Learning (ICML)*, PMLR vol. 97, pp. 5301–5310, 2019.

- [21] D. C. Liu and J. Nocedal, “On the limited memory BFGS method for large scale optimization,” *Mathematical Programming*, vol. 45, pp. 503–528, 1989, doi: 10.1007/BF01589116.
- [22] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. 3rd Int. Conf. Learning Representations (ICLR)*, 2015, arXiv:1412.6980.
- [23] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.