Intrinsic and Post-hoc Interpretability of Gaussian-Based Kolmogorov–Arnold Networks

Pasquale De Luca(✉), Emanuel Di Nardo, Emilio Garzia, Livia Marcellino, and Angelo Ciaramella

Department of Science and Technology, Parthenope University of Naples, Centro Direzionale C4, 80143 Naples, Italy
{pasquale.deluca,emanuel.dinardo,livia.marcellino, angelo.ciaramella}@uniparthenope.it, emilio.garzia001@studenti.uniparthenope.it

Abstract. Standard feedforward nets hang their nonlinearity on the nodes and keep the edges linear. Kolmogorov–Arnold Networks swap the two: each edge is now a learnable one-variable map, each node a plain summation. A small architectural move, but with a payoff we want to test here, namely, that the trained network becomes partly readable, edge by edge, without extra tooling. Our test bed is the shared-basis Gaussian variant of the KAN, recently published, and the task is regression on California Housing. We train a three-layer network with five Gaussian kernels per layer and then look at it from four angles. The first angle is the tensor of first-layer coefficients, summarized into a per-feature importance number. The second is KernelSHAP run globally; the third is LIME, pointwise; the fourth, and the one that only makes sense on a KAN, is the direct drawing of the learned edge activations as curves. Intrinsic and SHAP rankings land on the same two features at the top, MedInc and Latitude, and the same feature at the bottom (Population). They shuffle the middle, and that shuffle is diagnostic: coefficient size reports on encoding strength in layer zero, SHAP on end-to-end marginal effect. The plotted edge activations confirm what domain knowledge would suggest: income acts roughly monotonically on price, geography acts locally, crowding has a threshold effect. Taken together, the four views make the model easier to audit than a comparable black box would be.

Keywords: Kolmogorov–Arnold networks · explainable AI · radial basis functions · SHAP · feature importance · tabular regression

1 Introduction

A wrong prediction in loan approval, in property valuation, in medical triage or cancer prognosis [17] is not just a misclassification, it has downstream costs someone has to pay. In those settings, being able to audit *why* the model predicted

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026 Workshops, LNCS 16763, pp. 173–183, 2027.
https://doi.org/10.1007/978-3-032-30530-5_11

what it did weighs about as much as the prediction's accuracy on a held-out fold. The trouble is that modern deep networks, stacking nonlinearity after nonlinearity, are hard to audit in any direct way; so the machine-learning community has taken two routes to address the problem. One route is external: keep the black box, and build tools around it that probe and summarize its behavior, locally or globally [1,2]. The other is internal: rethink the architecture so the model is readable from the outset [3].

Kolmogorov–Arnold Networks [4] belong to the second tradition, and they take their justification from a theorem by Kolmogorov [5]. Any continuous $f\colon [0,1]^d \to \mathbb{R}$, the theorem says, factors through univariate maps:

$$f(x_1,\dots,x_d) = \sum_{q=0}^{2d} \Phi_q\Big(\sum_{p=1}^{d} \phi_{q,p}(x_p)\Big), \tag{1}$$

where every $\phi_{q,p}$ and every Φ_q is a scalar-to-scalar function. A KAN turns (1) into an architecture: edges carry the univariate functions, and those functions are learned as coefficients in some basis. For us, the relevant point is that each learned $\phi_{q,p}$ takes a single real number and returns a single real number. Which means it can be drawn. And drawn things, unlike opaque tensors, admit inspection.

The B-spline parameterization of the original KAN was recently replaced in [6], with Gaussian radial basis functions under a sharing rule: all edges of a layer pull from the same small pool of kernels. The benefit is a drop in parameter count from $\mathcal{O}(3KW)$ down to $\mathcal{O}(KW + 2LK)$, no loss of Sobolev convergence order, and clear gains on physics-informed benchmarks [18], especially hyperbolic PDEs or complex reaction-diffusion systems [15,16], where the C^∞ smoothness of Gaussians matters for high-order derivatives. Interpretability, though it was implicit in the design, was not the focus of that paper.

1.1 Contribution and Outline

This paper takes up interpretability explicitly. We pick California Housing because $d = 8$ is small, the features are interpretable to a human reader (income, latitude, age of the housing stock), and the target is familiar. Our protocol combines the two XAI traditions without privileging either: we read the trained coefficients where they talk, we probe the model with SHAP and LIME where they do, and we plot the edge activations because the architecture lets us. The central question is whether the four viewpoints, put side by side, tell the same story, or a richer one when they disagree.

Section 2 fixes the notation, recalls the shared-basis construction, and isolates the two interpretability handles its structure gives us for free. Section 3 describes the protocol and the four-level pipeline. Section 4 walks through the results, with attention to the places where the four levels do not line up. Section 5 summarizes, and flags a few open ends.

2 Mathematical Background

The purpose of this section is twofold: to state what a shared-basis RBF-KAN is, exactly, and to identify two observable quantities its structure gives us at no extra cost.

2.1 Shared-Basis RBF-KAN Architecture

Every layer $l \in \{0, \dots, L-1\}$ owns a pool of K Gaussian kernels, parametrized by centres $\boldsymbol{\mu}^{(l)} = (\mu_1^{(l)}, \dots, \mu_K^{(l)})$ and widths $\boldsymbol{\sigma}^{(l)} = (\sigma_1^{(l)}, \dots, \sigma_K^{(l)})$. To stop the widths from collapsing to zero, which would blow up the exponent in (2), they are clamped at $\sigma_{\min} = 10^{-6}$ during training [6]. The edge from input neuron j to output neuron i at layer l carries the activation

$$\phi_{ij}^{(l)}(x) = \sum_{k=1}^{K} c_{ijk}^{(l)} \, \psi_k^{(l)}(x), \qquad \psi_k^{(l)}(x) = \exp\left(-\frac{(x - \mu_k^{(l)})^2}{2(\sigma_k^{(l)})^2} \right), \tag{2}$$

with learnable scalar coefficients $c_{ijk}^{(l)} \in \mathbb{R}$ and shared kernels $\psi_k^{(l)}$. Propagation at layer l sends $\mathbf{h}^{(l)} \in \mathbb{R}^{d_l}$ to $\mathbf{h}^{(l+1)} \in \mathbb{R}^{d_{l+1}}$ through

$$h_i^{(l+1)} = \sum_{j=1}^{d_l} \phi_{ij}^{(l)}\big(h_j^{(l)}\big), \qquad i = 1, \dots, d_{l+1}. \tag{3}$$

The sharing is what makes this parsimonious. All $d_l \cdot d_{l+1}$ edges of a layer plug into the same K Gaussians; per-edge variation lives entirely in the coefficient vector $\mathbf{c}_{ij}^{(l)} \in \mathbb{R}^K$. Summing up: $\sum_{l=0}^{L-1}(K \cdot d_l \cdot d_{l+1} + 2K)$ parameters, i.e. $\mathcal{O}(KW + 2LK)$ with $W = \sum_l d_l d_{l+1}$.

2.2 Intrinsic Interpretability Mechanisms

Two quantities drop out of the formulae above with no added machinery.

Coefficient-Based Feature Importance. The first-layer tensor $\mathbf{C}^{(0)} \in \mathbb{R}^{d_0 \times d_1 \times K}$ has, at entry (i, j, k), the weight that input feature j puts on hidden unit i through kernel k. A simple per-feature summary is the mean absolute value,

$$\mathcal{I}_j^{\mathrm{RBF}} = \frac{1}{d_1 K} \sum_{i=1}^{d_1} \sum_{k=1}^{K} |c_{ijk}^{(0)}|, \tag{4}$$

which tells us, roughly, how loud the first-layer writing of feature j into the hidden state is. No sampling, no surrogate, no perturbation: one reads (4) off the trained weights.

Activation Function Visualization. Each $\phi_{ij}^{(l)}$ is a scalar function of a scalar variable, written in a known Gaussian basis with learned coefficients, so plotting $x \mapsto \phi_{ij}^{(l)}(x)$ is not a proxy for an explanation, it is the actual trained object. Drawing all edges of a three-layer network would not be readable, though, so we rank them by their ℓ^2 coefficient norm

$$\nu_{ij}^{(l)} = \Big(\sum_{k=1}^{K} (c_{ijk}^{(l)})^2\Big)^{1/2} \tag{5}$$

and keep the top few per layer. The next bound says the ranking is not arbitrary.

Proposition 1 (Pointwise activation bound). *If* $\|\psi_k^{(l)}\|_\infty = 1$ *for every* k, l, *then for any edge* (i, j) *in layer* l

$$\sup_{x \in \mathbb{R}} |\phi_{ij}^{(l)}(x)| \;\le\; \sqrt{K}\, \nu_{ij}^{(l)}. \tag{6}$$

Proof. Apply Cauchy–Schwarz to the sum in (2): $|\phi_{ij}^{(l)}(x)| \le \nu_{ij}^{(l)} \big(\sum_k |\psi_k^{(l)}(x)|^2\big)^{1/2}$. Since $|\psi_k^{(l)}(x)| \le 1$ everywhere, the inner sum is at most K and its square root is at most $\sqrt{K}$.

So $\nu_{ij}^{(l)}$ sets, up to a $\sqrt{K}$ factor, a ceiling on the loudness of edge (i, j). Ranking by $\nu_{ij}^{(l)}$ and keeping the top edges keeps, by construction, the edges that have room to matter most.

3 Explainability Framework

Section 2 gave us the parts. This section describes the pipeline that puts them to work. The design principle is a simple one: no single explainer is trusted on its own; four structurally different tools are run on the same trained model, and we look at where they converge and where they do not.

3.1 Experimental Setup

California Housing [7] is the data source, $n = 20{,}640$ rows, $d = 8$ numeric features (MedInc, HouseAge, AveRooms, AveBedrms, Population, AveOccup, Latitude, Longitude), and a target representing median house value in units of \$100k.

The model is a three-hidden-layer RBF-KAN of widths $[8, 8, 8, 1]$, $K = 5$ Gaussian kernels per layer, `kan_layer = rbf_linear`, initialization noise $\varepsilon_{\text{noise}} = 0.1$. Fifty L-BFGS epochs, learning rate $\eta = 1.0$, batch size $B = 512$, MSE loss. Inputs and target are standardized. Split 80/20, seed fixed.

3.2 Four-Level Interpretability Pipeline

From most structural to least, the four levels are these.

Level 1: Intrinsic RBF Coefficient Importance. Compute $\mathcal{I}_j^{\mathrm{RBF}}$ from (4) directly on the trained first-layer tensor. Nothing else to run.

Level 2: Global Post-Hoc Attribution via KernelSHAP. KernelSHAP [1] approximates Shapley values [8] by weighted least squares over sampled coalitions. We take $N_{\mathrm{bg}} = 100$ background points, explain $N_{\mathrm{test}} = 300$ test points, and use $n_{\mathrm{samples}} = 128$ kernel evaluations per explained instance. Mean absolute SHAP values $\overline{|\phi_j|}$ give the global ranking.

Level 3: Local Post-hoc Attribution via LIME. LIME [2] fits a sparse linear model locally around the point being explained. We pick three representative test points, the lowest, the median, and the highest predicted values, and explain each with 1000 perturbations.

Level 4: Structural Visualization. Two plots: first, $\phi_{ij}^{(l)}$ for the top-4 edges of each layer (by the norm in (5)), shown together with their individual weighted basis components; second, a per-layer heatmap of $\frac{1}{d_{l+1}} \sum_i |c_{ijk}^{(l)}|$ over input neuron j and knot index k. This is the level that requires the KAN formulation; nothing here has a direct counterpart for an MLP.

4 Results

We cover two things in this section: how the network fits (briefly, to establish that we have a sensible model to explain), and what the four interpretability levels say about it (at length, since that is the contribution).

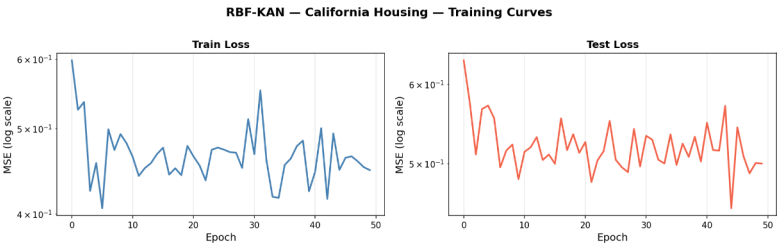


Fig. 1. Training and test MSE (log scale), 50 L-BFGS epochs. Jaggedness is a property of the optimiser, not of the loss surface.

4.1 Predictive Performance and Training Dynamics

After fifty L-BFGS epochs the test MSE lands near 4.7×10^{-1} in standardized units, enough for the model to have learned something non-trivial about the data, not so much that it has memorized it. The loss curves in Fig. 1 are jagged rather than smooth. This is expected: L-BFGS relies on a line search that occasionally overshoots before correcting, which is the cost one pays for using curvature information. Train and test curves move together throughout; the model does not overfit. Total parameters: about 1,070, noticeably fewer than what a dense MLP for this task would need, which is one of the things the sharing rule was designed to deliver [6].

4.2 Feature Importance: Intrinsic vs Post-Hoc Rankings

Table 1 lines up the four rankings. Intrinsic score first: MedInc leads at $\mathcal{I}^{\mathrm{RBF}} = 0.1911$, Latitude follows at 0.1656, AveOccup third at 0.1567. SHAP reorders: Latitude takes first place at $\overline{|\phi|} = 0.7264$, Longitude second at 0.5886, MedInc drops to third at 0.3170.

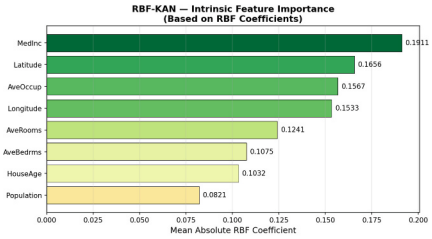
Table 1. Feature importance rankings across the four interpretability levels. Rank 1 is the most influential. The LIME column shows the dominant rank over three representative instances (low/median/high prediction).

Feature	$\mathcal{I}^{\mathrm{RBF}}$ Rank	SHAP Rank	LIME (dominant)
MedInc	1	3	2–3
Latitude	2	1	2
AveOccup	3	4	1
Longitude	4	2	3
AveRooms	5	5	4–5
AveBedrms	6	7	5–6
HouseAge	7	6	6–7
Population	8	8	7–8

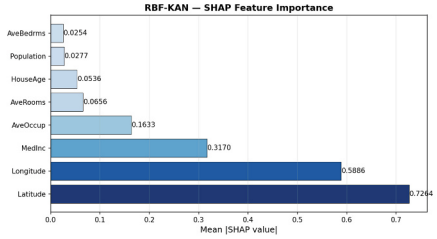
The two rankings are measuring different objects, and the difference is informative. $\mathcal{I}_j^{\mathrm{RBF}}$ weighs coefficients in the first layer: it records how emphatically feature j is injected into the initial representation. SHAP measures something else: once all three layers have run, how much does feature j shift the output at the margin. Latitude and Longitude carry only moderate first-layer weight, but the composition of the subsequent layers amplifies their effect, which is exactly what a high SHAP value records. MedInc, conversely, enters with large first-layer coefficients but part of its weight ends up baked into interactions rather than into marginal contribution, so SHAP sees less of it at the output end.

Despite the reshuffle in the middle of the ranking, the two views agree on the endpoints: MedInc and Latitude are in the top three for both, and Population

sits at the bottom for both. Where they diverge, on Longitude, AveOccup, the internal ordering of the geographic variables, the divergence is not noise, it is the signature of multi-layer interaction one would miss looking at any single view alone (Fig. 2).



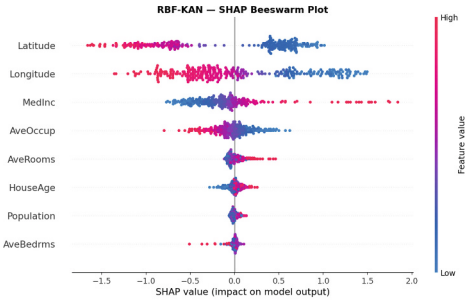
(a) Intrinsic importance (mean $|c_{ijk}^{(0)}|$).



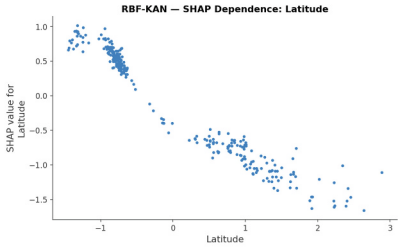
(b) SHAP global importance (mean $|\phi_j|$).

Fig. 2. Intrinsic coefficient importance (a) versus KernelSHAP (b). Both put MedInc and Latitude in the top tier. They disagree on where the two geographic variables belong.

SHAP also has a sign, which the bar plot throws away. The beeswarm in Fig. 3a brings it back. High Latitude values, Northern California, pull the prediction down; high MedInc values push it up. The Latitude dependence plot (Fig. 3b) shows the shape of that pull: smooth, monotonic, nearly linear over the observed range. The network, then, has not learned a threshold or a partition of the state into regions, but a continuous geographic gradient. None of this should surprise anyone who has looked at California house prices recently; that non-surprise is the point. A model whose explanations violate common knowledge about its domain would be a model one should hesitate to deploy.



(a) SHAP beeswarm plot.



(b) Dependence plot for Latitude.

Fig. 3. Directional SHAP views. (a) Beeswarm, showing monotonic trends for Latitude and MedInc. (b) Latitude dependence plot, a smooth near-linear descent, matching the known north–south gradient in California prices.

At the local level, LIME confirms the global picture one instance at a time (Fig. 4). AveOccup and the two geographic coordinates dominate the explanation for the low-value, median-value, and high-value test points alike; they do not drop in and out across the prediction range the way a less robust importance signal would. MedInc flips sign between the low and high instances, negative in the former, positive in the latter, which is what one expects of a feature that drives the outcome monotonically: where the prediction is low, the feature pushed it low; where it is high, the feature pushed it high. The sign flip at the local level is consistent with the monotonicity SHAP picked up globally, which is the sort of internal consistency one wants from a model one is about to trust.

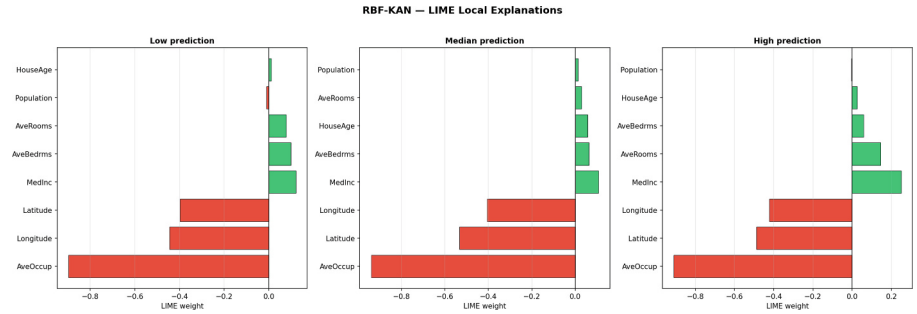


Fig. 4. LIME local explanations, three instances (low, median, high prediction). AveOccup and the geographic variables dominate in all three. MedInc flips sign cleanly between the low and high cases.

Level 4 is where the KAN formulation really earns its keep. Figure 5 plots the top four edges per layer as honest-to-goodness univariate curves, each the superposition of the five weighted Gaussians the edge has learned. First-layer Latitude→out is a single clean Gaussian peak, geographically, the Bay Area bump sitting in the region of the feature axis where the most expensive submarket lives once features have been standardized. MedInc→out is broader and roughly monotonic, which is the shape one wants from a feature that tracks the target linearly. AveOccup→out is a sharp peak near the standardized mean; past the peak the curve descends sharply, which reads cleanly as a threshold effect: once occupancy exceeds a certain level, it starts to pull prices down.

The second and third rows of Fig. 5 are busier. The shapes there are multimodal, with kinks and sharp transitions. That is the composition step doing its work: subsequent layers take the features-wise transforms of the first layer and combine them into higher-order structure. What matters for interpretability is that this structure is still, at every depth, a sum of univariate transforms of scalar arguments, so the decomposition of (1) holds through the entire network, not just in the first layer.

The heatmaps in Fig. 6 add a complementary picture. In layer 0, knot 0 has large coefficients for MedInc and Latitude; knot 2 is where AveOccup and

AveRooms put most of their weight. Different knots serve different features, and not uniformly. This specialization is possible precisely because the kernels are shared: all edges pull from the same pool of five Gaussians, so the coefficient pattern tells us which pool member each feature calls on, and how strongly.

5 Conclusions

Four windows on the same trained network: coefficients, SHAP, LIME, activation plots. None of them is redundant with the others. Coefficients read the encoding

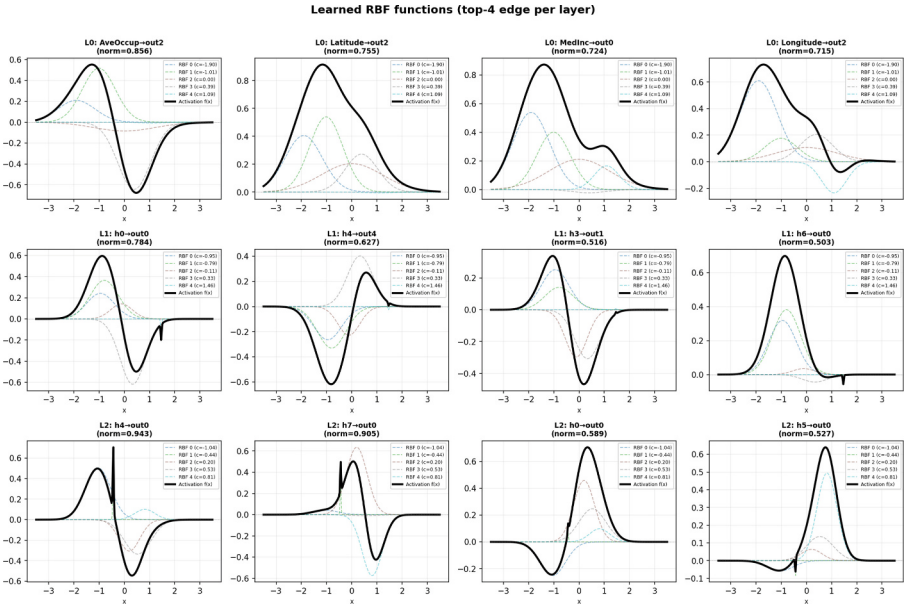


Fig. 5. Learned RBF activation functions, top-4 edges per layer (by coefficient norm). Dashed: the individual weighted Gaussian components. Solid: their sum, which is the activation the network actually applies on that edge.

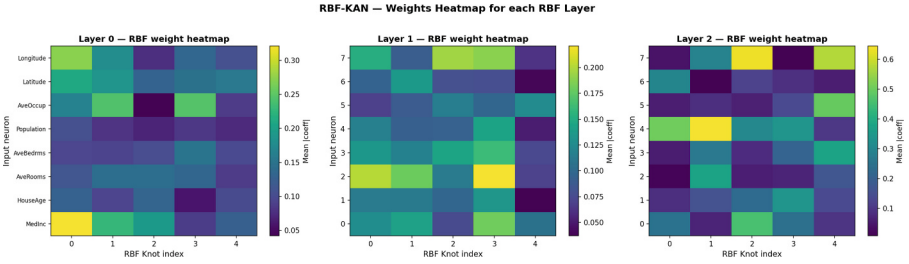


Fig. 6. Coefficient heatmaps, one per layer: mean absolute coefficient over input neurons (rows) and knot indices (columns). Unevenness is knot specialization.

stage; SHAP reads the output margin; LIME zooms in on single points; the activation plots show the nonlinearities as drawings one can inspect. When the four windows agree, the attribution is on solid ground. When they disagree, as they do on the middle of our ranking, or on the relative importance of the two geographic coordinates, the disagreement is informative, pointing to multi-layer structure that no single window would have revealed.

What sets this exercise apart from standard XAI on a black box is the last window. SHAP and LIME summarize influence as numbers, one per feature per instance; the KAN, by construction, preserves the functional form itself, and lets us look at it. That is what Rudin [3] had in mind when she argued for intrinsically interpretable models over post-hoc explainers: not more scalars to read, but the actual learned function to examine.

5.1 Limitations and Outlook

Eight features is friendly. Scaling activation plots to $d \gg 10$ is not straightforward, first-layer edges grow as $d \times d_1$, and at some point the human reader's attention runs out even though the mathematics is unchanged. One natural response is to use the norm (5) as a pruning criterion: apply Prop. 1 to filter out edges whose ceiling is below some threshold, and plot only what remains. A second limitation is that $\mathcal{I}_j^{\mathrm{RBF}}$ reads the first layer alone; a multi-layer variant, along the lines of a product of edge norms on paths from input to output, would plausibly track SHAP more closely, and would be worth developing. Beyond tabular regression, a natural follow-up is to take this analysis back to the physics-informed setting of [6,18], particularly for modeling complex dynamical systems and PDEs [15,16], where the shapes of the learned activations could be compared not merely against domain intuition but against the differential operator the network is approximating, a more stringent interpretability test than one can set up on Housing data.

Acknowledgments. This work was partially supported by INdAM - GNCS project 2026 "Metodi Numerici e Machine Learning per Sistemi Dinamici Complessi: Identificazione, Approssimazione"', ref. no. CUP E53C25002010001. P. De Luca, E. Di Nardo, L. Marcellino and A. Ciaramella are members of INdAM-GNCS.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Lundberg, S.M., Lee, S.-I.: A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
2. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why should I trust you?": explaining the predictions of any classifier. In: 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135–1144 (2016)

3. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Mach. Intell. **1**(5), 206–215 (2019)
4. Liu, Z., et al.: KAN: Kolmogorov–Arnold networks. arXiv preprint arXiv:2404.19756 (2024)
5. Kolmogorov, A.N.: On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. Dokl. Akad. Nauk SSSR **114**, 953–956 (1957)
6. De Luca, P., Di Nardo, E., Marcellino, L., Ciaramella, A.: Stable and efficient Gaussian-based Kolmogorov-Arnold networks. Mathematics **14**(3), 513 (2026)
7. Pace, R.K., Barry, R.: Sparse spatial autoregressions. Stat. Probabil. Lett. **33**(3), 291–297 (1997)
8. Shapley, L.S.: A value for n-person games. In: Kuhn, H.W., Tucker, A.W. (eds.) Contributions to the Theory of Games, vol. 2, pp. 307–317. Princeton University Press (1953)
9. Molnar, C.: Interpretable Machine Learning, 2nd edn. (2022). https://christophm.github.io/interpretable-ml-book/
10. Park, J., Sandberg, I.W.: Universal approximation using radial-basis-function networks. Neural Comput. **3**(2), 246–257 (1991)
11. Schaback, R.: Error estimates and condition numbers for radial basis function interpolation. Adv. Comput. Math. **3**, 251–264 (1995)
12. Wendland, H.: Scattered Data Approximation. Cambridge University Press (2005)
13. Buhmann, M.D.: Radial Basis Functions: Theory and Implementations. Cambridge University Press (2003)
14. Lipton, Z.C.: The mythos of model interpretability. Queue **16**(3), 31–57 (2018)
15. De Luca, P., Marcellino, L.: Conservation law analysis in numerical schema for a tumor angiogenesis PDE system. Mathematics **13**(1), 28 (2025). https://doi.org/10.3390/math13010028
16. Cardone, A., De Luca, P., Galletti, A., Marcellino, L.: Solving Time-Fractional reaction-diffusion systems through a tensor-based parallel algorithm. XXPhys. A **611**, 128472 (2023)
17. Ciaramella, A., Di Nardo, E., Terracciano, D., et al.: A new biomarker panel of ultraconserved long non-coding RNAs for bladder cancer prognosis by a machine learning based methodology. BMC Bioinformatics **23**(Suppl 6), 569 (2022). https://doi.org/10.1186/s12859-023-05167-6
18. Colace, F., Conte, D., Pagano, G., Paternoster, B., Valentino, C.: Physics-informed neural networks for a Lithium-ion batteries model: a case of study. Adv. Comput. Sci. Eng. (ACSE) **2**(4) (2024)