Scaling of Gaussian Kolmogorov–Arnold Networks

Amir Noorizadegan[1,*], Sifan Wang[2]

[1]Department of Mathematics, Hong Kong Baptist University, Hong Kong SAR, China

[2]Institution for Foundations of Data Science, Yale University, New Haven, CT 06520, USA

*Corresponding author: amir_noori@hkbu.edu.hk

April 24, 2026

Abstract

The Gaussian scale parameter ϵ is central to the behavior of Gaussian Kolmogorov–Arnold Networks (KANs), yet its role in deep edge-based architectures has not been studied systematically. In this paper, we investigate how ϵ affects Gaussian KANs through first-layer feature geometry, conditioning, and approximation behavior. Our central observation is that scale selection is governed primarily by the first layer, since it is the only layer constructed directly on the input domain and any loss of distinguishability introduced there cannot be recovered by later layers.
From this viewpoint, we analyze the first-layer feature matrix and identify a practical operating interval,

$$\epsilon \in \left[\frac{1}{G-1}, \frac{2}{G-1}\right],$$

where G denotes the number of Gaussian centers. For the standard shared-center Gaussian KAN used in current practice, we interpret this interval not as a universal optimality result, but as a stable and effective design rule, and validate it through brute-force sweeps over ϵ across function-approximation problems with different collocation densities, grid resolutions, network architectures, and input dimensions, as well as a physics-informed Helmholtz problem. We further show that this range is useful for fixed-scale selection, variable-scale constructions, constrained training of ϵ, and efficient scale search using early training MSE. Finally, using a matched Chebyshev reference, we show that a properly scaled Gaussian KAN can already be competitive in accuracy relative to another standard KAN basis. In this way, the paper positions scale selection as a practical design principle for Gaussian KANs rather than as an ad hoc hyperparameter choice. The implementation is available at https://github.com/AmirNoori68/Gaussian-KAN.

Keywords: Gaussian KAN; Kolmogorov–Arnold networks; Gaussian shape parameter; conditioning; kernel methods; radial basis functions; physics-informed neural networks.

1 Introduction

Kolmogorov–Arnold Networks have recently emerged as a structured alternative to classical multilayer perceptrons (MLPs), replacing fixed node activations with learnable univariate functions placed on edges [1, 2]. This edge-based construction provides a different inductive bias for multivariate approximation and has attracted growing interest in scientific machine learning, regression, and physics-informed modeling [3, 4]. The original KAN formulation is built from B-spline bases [1], and a rapidly expanding literature has explored alternative basis families to improve smoothness, computational efficiency, and approximation quality.

Recent KAN variants have employed many different basis functions, including Chebyshev-type constructions [5–7], Jacobi and rational Jacobi bases [8,9], Fourier-based activations [10,11], ReLU-based and adaptive piecewise bases [12–14], as well as wavelet, finite-basis, and polynomial variants [15–17]. In parallel, recent work has also focused on improving adaptability and efficiency through geometric or grid refinement strategies [18], grid-adaptive physics-informed KANs [14], sparse-identification frameworks [20], JAX/GPU implementations [6, 14], and improved initialization strategies [19].

Among these basis families, Gaussian radial basis functions (RBFs) are particularly attractive because of their smoothness, locality, and simple analytical form. More broadly, Gaussian KANs are part of the long development of RBF methods. Starting from Hardy's multiquadric formulation [21] and Kansa's extension to PDEs [22], RBF methods have developed into a large literature on approximation, interpolation, and scientific computing.

In the KAN setting, replacing B-spline activations with Gaussian bases was first proposed in FastKAN [23], where Gaussian functions were introduced as computationally efficient surrogates for spline-based activations. In that work, the

scale parameter was fixed, mainly to show that Gaussian bases can approximate spline constructions while reducing implementation complexity and computational cost. Subsequent works extended Gaussian-based KANs in several directions, including residual formulations [24], operator-learning architectures with learnable Gaussian bases [25], improved training efficiency and accuracy [26, 27], and complex-valued Gaussian KANs [35, 36]. Beyond regression and PDE-related applications, Gaussian-based KANs have also shown promising performance in classification, with reduced parameter counts, faster training, and improved stability in large-scale or high-dimensional settings [37]. Taken together, these works show that Gaussian bases are not merely substitutes for splines, but rather constitute a flexible and expressive class of KAN representations.

What remains largely unresolved, however, is how to choose the Gaussian scale parameter ϵ in a neural-network setting, or more generally the scale in RBFs that admit such a parameter, since not all RBFs do. In most existing Gaussian-KAN works, ϵ is either fixed heuristically [23, 38], selected by trial and error, or made trainable without a clear understanding of its role [26, 37, 39]. A recent comparative physics-informed study also included Gaussian RBF KANs among several learnable basis families, but treated the Gaussian scale as a fixed preset hyperparameter rather than the subject of a dedicated analysis [38]. In that study, the authors further noted that RBFs may suffer from low accuracy and vanishing gradients when the scaling factor is not chosen appropriately, again highlighting that basis scaling should be handled carefully for stable training.

In classical Gaussian RBF interpolation, the scale parameter governs the balance between approximation power and numerical stability, and poor choices can lead either to weak approximation or to severe ill-conditioning [28–32, 34]. This is closely related to the stability–accuracy *trade-off* and the *uncertainty-principle* viewpoint in the RBF literature [33, 34, 40], and the general optimal-scaling problem remains open even in classical meshfree methods.

This difficulty in choosing the scale parameter, together with the associated conditioning issues, is one reason why strong-form meshfree RBF methods, despite their potentially much higher accuracy [41], have not replaced the more stable weak-form finite element framework. However, as Larsson and Schaback emphasized,

> *the strong dependence of radial basis function techniques on scaling is a feature, not a bug* [28].

Rather than treating scale sensitivity as a defect, it can be viewed as a mechanism that, when understood properly, allows one to control approximation behavior.

Noorizadegan et al. [42–44], inspired by Schaback's conditioning–accuracy trade-off, showed that working within a numerically safe regime, while pushing the conditioning as high as safely allowed by the implementation and the available reliable digits [45], can lead to good accuracy. One step further in this direction is the flat-limit regime [46, 47]. We do not consider that regime here, and instead keep the analysis within the simple Gaussian-KAN setting used in current practice.

The present work does not aim to introduce a new approximation theorem for Gaussian radial basis functions. Rather, it studies a question that is specific to Gaussian KANs: how the Gaussian scale parameter behaves inside a deep edge-based architecture, how this behavior is reflected in first-layer conditioning, and how it can be used to guide practical model design. In this sense, the paper translates classical RBF scaling ideas into a neural setting where the scale parameter affects not only approximation quality, but also representation collapse, trainability, and architectural robustness across layers.

Our main contributions are as follows. First, we identify the first layer as the decisive layer for scale selection in Gaussian KANs, both theoretically and experimentally, and show that its conditioning tracks the error behavior of the full network more closely than the deeper layers. Second, we formulate the first Gaussian KAN layer in kernel language and use this viewpoint to define practical conditioning diagnostics directly at the feature-matrix level. Third, for the standard shared-center Gaussian KAN used in current practice, we obtain a simple operating interval for the scale parameter and show that it remains effective across different target functions, collocation densities, grid sizes, architectures, dimensions, and a representative physics-informed PDE problem. Finally, we show that this interval is useful not only for stable fixed-scale selection, but also for variable-scale constructions, constrained scale training, and low-cost scale search using early training MSE. We also use a matched Chebyshev reference to show that a properly scaled Gaussian KAN is already competitive in accuracy relative to another standard KAN basis.

The remainder of this paper is organized as follows. Section 2 introduces the motivation from Kolmogorov superposition and presents the Gaussian KAN formulation, together with the experimental setup and benchmark target functions. Section 3 develops the first-layer kernel viewpoint, establishes the first layer as the main representational bottleneck, and supports this role with layer-wise numerical evidence. Section 4 studies first-layer conditioning and derives a practical interval for the Gaussian scale parameter, then examines how this interval behaves under changes in collocation density, number of Gaussian centers, network architecture, and input dimension, and finally discusses shared variable scales, training-MSE-based scale search, and a representative physics-informed Helmholtz problem. Section 5 concludes the paper, and Section 6 discusses limitations and directions for future work.

2 From Kolmogorov Superposition to Gaussian KAN Formulation

2.1 Kolmogorov Superposition as Motivation for KANs

Hilbert's question. At the 1900 International Congress of Mathematicians, Hilbert posed what later became known as Hilbert's 13th problem [48]. In modern terms, the question asks whether multivariate continuous functions can be represented by finite superpositions of lower-dimensional functions. Hilbert conjectured that such a reduction should fail in general.

Kolmogorov superposition theorem. This conjecture was overturned by the work of Kolmogorov and Arnol'd [49–51].

Theorem 1 (Kolmogorov superposition theorem [51])**.** *Let $d \geq 2$ and let $C([0,1]^d)$ denote the space of continuous real-valued functions on $[0,1]^d$. Then there exist continuous functions $\theta_{q,p} : [0,1] \to \mathbb{R}$, independent of the target function f, such that every $f \in C([0,1]^d)$ admits a representation of the form*

$$f(x_1, \ldots, x_d) = \sum_{q=0}^{2d} \Psi_q \left(\sum_{p=1}^{d} \theta_{q,p}(x_p) \right), \tag{1}$$

where the outer functions $\Psi_q : \mathbb{R} \to \mathbb{R}$ are continuous and depend on f.

Equation (1) shows that multivariate dependence can be expressed through coordinate-wise univariate transformations, followed by summation and nonlinear recombination. This structural principle is the main connection to KANs.

However, KANs should be viewed as architectures inspired by KST rather than exact realizations of the theorem. A more detailed discussion of the relation between KST and KAN can be found in [52].

2.2 Gaussian KAN Layer Formulation

Consider a layer ℓ with input width n_ℓ and output width $n_{\ell+1}$. Given an input vector $x^{(\ell)} \in \mathbb{R}^{n_\ell}$, a KAN layer assigns a univariate function

$$\psi_{j,i}^{(\ell)} : \mathbb{R} \to \mathbb{R}$$

to each edge from input coordinate i to output coordinate j. The output $x^{(\ell+1)} \in \mathbb{R}^{n_{\ell+1}}$ is defined componentwise by

$$x_j^{(\ell+1)} = \sum_{i=1}^{n_\ell} \psi_{j,i}^{(\ell)} \left(x_i^{(\ell)} \right), \qquad j = 1, \ldots, n_{\ell+1}. \tag{2}$$

Thus, each input coordinate is processed independently through a univariate map, and the resulting contributions are summed to form each output coordinate. This edge-based structure is the defining feature of KANs and the starting point for the Gaussian construction considered here.

In the Gaussian KAN studied in this work, every edge function in (2) is expanded in the same finite Gaussian basis. We fix a shared set of centers

$$\mathcal{C} = \{c_1, \ldots, c_G\} \subset [0,1] \tag{3}$$

and a shared Gaussian scale

$$\epsilon > 0. \tag{4}$$

These quantities are shared across all layers. The restriction $\mathcal{C} \subset [0,1]$ is motivated by the first layer, where the original inputs are scaled to this interval. Since deeper-layer coordinates are not constrained to remain in $[0,1]$, the Gaussian feature map is defined on all of $\mathbb{R}$:

$$\varphi : \mathbb{R} \to \mathbb{R}^G, \qquad \varphi(t) = \begin{bmatrix} \exp\left(-\dfrac{(t-c_1)^2}{\epsilon^2} \right) \\ \vdots \\ \exp\left(-\dfrac{(t-c_G)^2}{\epsilon^2} \right) \end{bmatrix}. \tag{5}$$

where G is the number of centers. For each layer ℓ, each edge function is represented as

$$\psi_{j,i}^{(\ell)}(t) = \left(w_{j,i}^{(\ell)} \right)^\top \varphi(t), \qquad w_{j,i}^{(\ell)} \in \mathbb{R}^G. \tag{6}$$

Hence, the nonlinear behavior of the layer is carried by the Gaussian feature map, while the dependence on the coefficients remains linear.

To write the layer compactly, define the stacked feature vector

$$\Phi\Big(x^{(\ell)}\Big) = \begin{bmatrix} \varphi\Big(x_1^{(\ell)}\Big) \\ \vdots \\ \varphi\Big(x_{n_\ell}^{(\ell)}\Big) \end{bmatrix} \in \mathbb{R}^{n_\ell G}, \tag{7}$$

and the block coefficient matrix

$$W^{(\ell)} = \begin{bmatrix} \big(w_{1,1}^{(\ell)}\big)^\top & \cdots & \big(w_{1,n_\ell}^{(\ell)}\big)^\top \\ \vdots & \ddots & \vdots \\ \big(w_{n_{\ell+1},1}^{(\ell)}\big)^\top & \cdots & \big(w_{n_{\ell+1},n_\ell}^{(\ell)}\big)^\top \end{bmatrix} \in \mathbb{R}^{n_{\ell+1} \times n_\ell G}. \tag{8}$$

Then the Gaussian KAN layer takes the form

$$x^{(\ell+1)} = W^{(\ell)} \Phi\Big(x^{(\ell)}\Big). \tag{9}$$

For later use, define the layer operator

$$F^{(\ell)} : \mathbb{R}^{n_\ell} \to \mathbb{R}^{n_{\ell+1}}, \qquad F^{(\ell)}(z) = W^{(\ell)} \Phi(z). \tag{10}$$

A deep Gaussian KAN with L layers is then written as

$$f = F^{(L-1)} \circ F^{(L-2)} \circ \cdots \circ F^{(0)}, \qquad f : [0,1]^{n_0} \to \mathbb{R}^{n_L}. \tag{11}$$

Thus, every layer is built from the same mechanism: a Gaussian feature map followed by a linear coefficient map. The key distinction is that the first layer acts directly on the original input coordinates, whereas deeper layers act on learned intermediate representations. For this reason, the first layer is the natural starting point for the kernel analysis in the next section.

3 First-Layer Kernel Structure and Bottleneck Role

3.1 First-Layer Feature Map and Induced Kernel

The first layer is the only layer whose Gaussian basis is evaluated directly on the original input domain. For an input $x = (x_1, \ldots, x_d)^\top \in [0,1]^d$, the first-layer feature representation is obtained by stacking the one-dimensional Gaussian features coordinate-wise:

$$\Phi(x) = \begin{bmatrix} \varphi(x_1) \\ \vdots \\ \varphi(x_d) \end{bmatrix} \in \mathbb{R}^{dG}. \tag{12}$$

The first hidden representation is therefore

$$x^{(1)} = W^{(0)} \Phi(x), \tag{13}$$

where $W^{(0)} \in \mathbb{R}^{n_1 \times dG}$. Thus, the first layer is a finite-dimensional feature map followed by a linear transformation.

The associated one-dimensional kernel induced by the Gaussian feature map (5) is

$$k(s,t) = \varphi(s)^\top \varphi(t), \qquad s,t \in [0,1]. \tag{14}$$

Accordingly, the first-layer kernel between two inputs $x, x' \in [0,1]^d$ is

$$K_0(x,x') = \Phi(x)^\top \Phi(x'). \tag{15}$$

Using the stacked structure in (12), this becomes

$$K_0(x,x') = \sum_{i=1}^{d} \varphi(x_i)^\top \varphi(x_i') = \sum_{i=1}^{d} k(x_i, x_i'). \tag{16}$$

Hence the first Gaussian KAN layer induces an additive kernel over the input coordinates. This representation is exact and directly reflects the edge-wise structure of KANs in (2): each coordinate is processed separately, and the resulting contributions are summed.

For later conditioning analysis, consider a finite sample set

$$\mathcal{X} = \{x^1, \dots, x^N\} \subset [0,1]^d. \tag{17}$$

For each coordinate $i = 1, \dots, d$, define the feature block

$$\Phi^{(i)} = \begin{bmatrix} \varphi(x_i^1)^\top \\ \vdots \\ \varphi(x_i^N)^\top \end{bmatrix} \in \mathbb{R}^{N \times G}. \tag{18}$$

Stacking these blocks horizontally gives the full first-layer feature matrix

$$\Phi = \begin{bmatrix} \Phi^{(1)} & \cdots & \Phi^{(d)} \end{bmatrix} \in \mathbb{R}^{N \times dG}. \tag{19}$$

The empirical first-layer kernel matrix is then

$$K_0 = \Phi\Phi^\top \in \mathbb{R}^{N \times N}, \tag{20}$$

and, by the block structure of Φ,

$$K_0 = \sum_{i=1}^{d} \Phi^{(i)} \left(\Phi^{(i)}\right)^\top. \tag{21}$$

Thus, the empirical first-layer kernel matrix is the sum of the coordinate-wise Gram matrices. This matrix is the starting point for the later analysis of rank, conditioning, and admissible Gaussian scales.

3.2 The First Layer as a Representational Bottleneck

Because the first layer acts directly on the original variables, any loss of distinguishability introduced at this stage is inherited by all later layers. Writing the full network as

$$f = T \circ G, \qquad G(x) = W^{(0)}\Phi(x), \tag{22}$$

where $T = F^{(L-1)} \circ \cdots \circ F^{(1)}$ denotes the composition of the remaining layers, makes this dependence explicit.

Proposition 2 (First-layer bottleneck)**.** *Let f be the Gaussian KAN in (22). Then:*

(i) If two inputs $x, x' \in [0,1]^d$ satisfy

$$G(x) = G(x'), \tag{23}$$

then

$$f(x) = f(x'). \tag{24}$$

(ii) If two inputs $x, x' \in [0,1]^d$ satisfy

$$\Phi(x) = \Phi(x'), \tag{25}$$

then they induce the same first-layer kernel section,

$$K_0(x, z) = K_0(x', z), \qquad z \in [0,1]^d, \tag{26}$$

and therefore

$$f(x) = f(x') \tag{27}$$

for every choice of $W^{(0)}$ and every downstream map T.

(iii) On a finite sample set $\mathcal{X} \subset [0,1]^d$, as $\epsilon \to \infty$,

$$\Phi(x) \to \mathbf{1}_{dG} \qquad \text{for every } x \in [0,1]^d, \tag{28}$$

and hence

$$\Phi \to \mathbf{1}_N \mathbf{1}_{dG}^\top. \tag{29}$$

Consequently,

$$K_0 = \Phi\Phi^\top \to dG\,\mathbf{1}_N\mathbf{1}_N^\top, \tag{30}$$

so the empirical first-layer kernel collapses to a rank-one limit. Moreover,

$$x^{(1)} = W^{(0)}\Phi(x) \to W^{(0)}\mathbf{1}_{dG}, \tag{31}$$

which means that the first hidden representation becomes constant across the sample set.

Proposition 2 formalizes the special role of the first layer. Once two inputs become indistinguishable after the first-layer feature map, no later layer can separate them. In particular, large values of the shared Gaussian scale ϵ force the first-layer features toward a rank-one limit, causing collapse of both the empirical kernel matrix and the representation itself. For this reason, the numerical behavior of the Gaussian KAN is governed first and foremost by the geometry of the first-layer feature matrix.

3.3 Experimental Setup and Benchmark Target Functions

Unless stated otherwise, we use a Gaussian KAN with architecture

$$[2, 12, 12, 1],$$

that is, a two-dimensional input, two hidden layers of width 12, and a one-dimensional output. The Gaussian scale parameter ϵ is swept over 100 logarithmically spaced values in the interval

$$\epsilon \in [5 \times 10^{-3},\, 5].$$

The learning rate is fixed at

$$10^{-3}.$$

This dense sweep is used as a brute-force reference, so that the proposed conditioning-based interval can be evaluated against the best empirical choices observed within the tested search range.

Training points are sampled on $[0, 1]^2$ using Halton sequences; see Fasshauer [40] and the discussion in Fasshauer and Iske [47]. Halton points are low-discrepancy sequences that provide a more uniform coverage of the domain than purely random samples. Such point sets are widely used in radial basis function meshfree methods.

For each experiment, 4 or 5 random seeds are used, accounting for both the random initialization of the coefficients and the point distribution. In the plots, the solid curve shows the geometric mean of the error over the seeds in log scale, while the shaded region indicates the minimum and maximum values across seeds.

As the main error measure, we use the validation root mean square error (RMSE), defined by

$$\mathrm{RMSE} = \left(\frac{1}{M} \sum_{i=1}^{M} \big(\hat{u}(x_i, y_i) - u(x_i, y_i)\big)^2 \right)^{1/2}, \tag{32}$$

where $\{(x_i, y_i)\}_{i=1}^M$ are the validation points, u is the target function, and $\hat{u}$ is the network prediction.

We consider four benchmark target functions, denoted by F1, F2, F3, and F4. They were chosen to represent smooth localized structure, periodic oscillation, sharp variation, and discontinuity. The functions F1, F2, and F4 are defined on $(x, y) \in [0, 1]^2$, whereas F3 is defined on $(x, y) \in [-1, 1]^2$.

- **F1: smooth multi-bump function**

$$\begin{aligned} F1(x,y) = {} & 0.75\exp\!\left(-\frac{(9x-2)^2+(9y-2)^2}{4}\right) + 0.75\exp\!\left(-\frac{(9x+1)^2}{49} - \frac{9y+1}{10}\right) \\ & + 0.5\exp\!\left(-\frac{(9x-7)^2+(9y-3)^2}{4}\right) - 0.2\exp\!\big(-\big((9x-4)^2+(9y-7)^2\big)\big)\,. \end{aligned} \tag{33}$$

- **F2: periodic oscillatory function**

$$F2(x, y) = \sin(4\pi x)\sin(4\pi y). \tag{34}$$

- **F3: smooth function with a sharp localized peak**

$$F3(x, y) = \frac{1}{1 + 10^3(x^2 - 0.25)^2(y^2 - 0.25)^2}, \qquad (x, y) \in [-1, 1]^2. \tag{35}$$

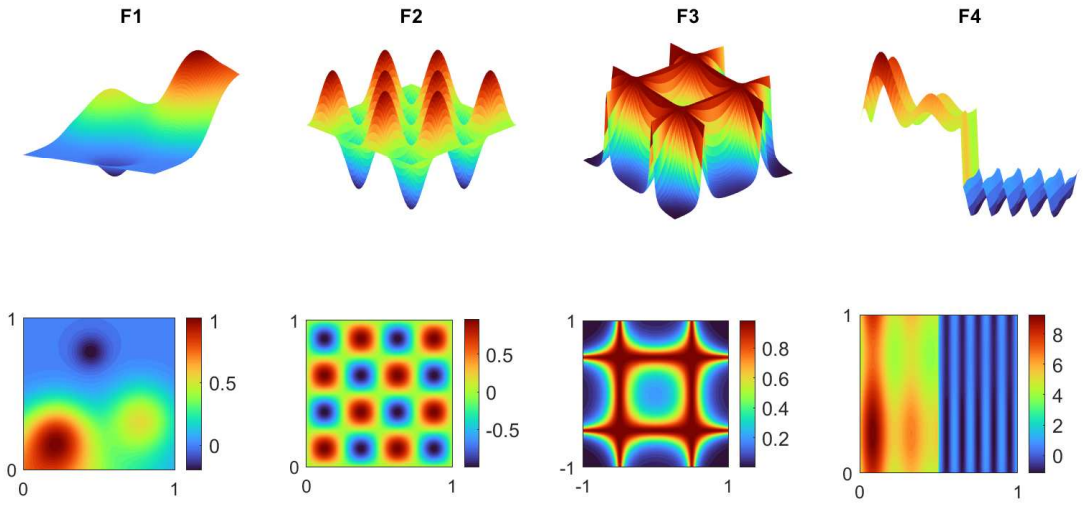


Figure 1: Three-dimensional surfaces of the four target functions used in the regression experiments: F1 (smooth multi-bump), F2 (periodic oscillatory), F3 (sharp localized peak), and F4 (discontinuous piecewise oscillatory). These examples span different approximation challenges, including smooth variation, repeated oscillation, strong localization, and discontinuity.

- **F4: discontinuous piecewise oscillatory function**

$$F4(x,y) = g(x)\left(1 + 0.15\sin(2\pi y)\right), \tag{36}$$

where

$$g(x) = \begin{cases} 5 + \sum_{k=1}^{4} \sin(2\pi k x), & x < 0.5, \\ \cos(20\pi x), & x \geq 0.5. \end{cases} \tag{37}$$

Figure 1 shows the surfaces of these four target functions.

3.4 Numerical Verification of First-Layer Dominance

We now provide numerical evidence for Proposition 2.

(i) Layer-wise sensitivity of the Gaussian scale parameter. We first study how the validation error changes when the Gaussian scale is varied in only one layer at a time. For the Gaussian KAN used in Figure 2, the architecture is $[2, 12, 12, 1]$, with a two-dimensional input, two hidden layers of width 12, and a one-dimensional output. This gives rise to three successive Gaussian mappings,

$$\epsilon_1 \text{ for } 2 \to 12, \qquad \epsilon_2 \text{ for } 12 \to 12, \qquad \epsilon_3 \text{ for } 12 \to 1,$$

so three scale parameters are involved, one for each applied layer. Let

$$\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \epsilon_3). \tag{38}$$

We fix the non-varied layers at the reference value 0.1 and compare the following schedules:

$$\text{Case 1: } (\epsilon, 0.1, 0.1), \quad \text{Case 2: } (0.1, \epsilon, 0.1), \quad \text{Case 3: } (0.1, 0.1, \epsilon), \quad \text{Shared-}\epsilon \text{ case: } (\epsilon, \epsilon, \epsilon). \tag{39}$$

That is, we vary the scale in only one layer while keeping the others fixed at a good reference value, and compare the resulting response with the fully shared-ϵ case. This allows us to identify the dominant layer, namely the layer whose scale variation has the strongest effect on the overall accuracy.

Figure 2 shows the validation RMSE as a function of ϵ for the smooth target $F1$ and the discontinuous target $F4$. In both cases, the first-layer sweep $(\epsilon, 0.1, 0.1)$ tracks the shared-parameter sweep $(\epsilon, \epsilon, \epsilon)$ much more closely than the sweeps

obtained by varying only the deeper layers. This shows that changing the first-layer scale has a much stronger effect on accuracy than changing the scales in the later layers. We also observe that a single good scale can work well across all layers, so there is no clear need to select different scales for different layers.

Thus, for this architecture, the dominant sensitivity to the Gaussian scale is carried by the first layer.

(ii) Conditioning and controlled layer collapse. We next examine the same phenomenon through conditioning and explicit layer collapse. Here the architecture is

$$[2, 12, 12, 12, 1],$$

so the network contains four successive Gaussian layers.

To isolate the effect of representation loss, we force one layer at a time into the large-ϵ collapse regime described in Proposition 2, while keeping the other layers at the fixed value $\epsilon = 0.1$. The left panel of Figure 3 shows the resulting validation RMSE for the smooth target $F1$. The no-collapse RMSE corresponds to the case in which no layer is collapsed and all layers use $\epsilon = 0.1$. Collapsing an early layer causes the largest degradation, with the most severe effect occurring when the first layer is collapsed. By contrast, collapse in later layers is less harmful, especially in the deepest layers.

The right panel of Figure 3 plots the spectral condition number $\kappa(\Phi_\ell)$ of each layer as a function of ϵ for the uncollapsed network, that is, without any forced collapse. The first layer is the only one that exhibits a clear low-conditioning regime, whereas the deeper layers remain highly ill-conditioned throughout the sweep. The apparent drop in the condition numbers of Layers 2 to 4 after about $\epsilon \approx 0.3$ is a finite-precision artifact in `float32`, not a genuine improvement in conditioning. It occurs because the smallest singular values are no longer reliably resolved at that precision. In higher precision, such as `float64`, these layers remain highly ill-conditioned and the artificial drop is delayed or disappears.

Together, Figures 2 and 3 support Proposition 2 and confirm that the first layer is the primary bottleneck in Gaussian KANs: it is the layer most sensitive to the Gaussian scale, and any loss introduced there has the strongest effect on the final accuracy.

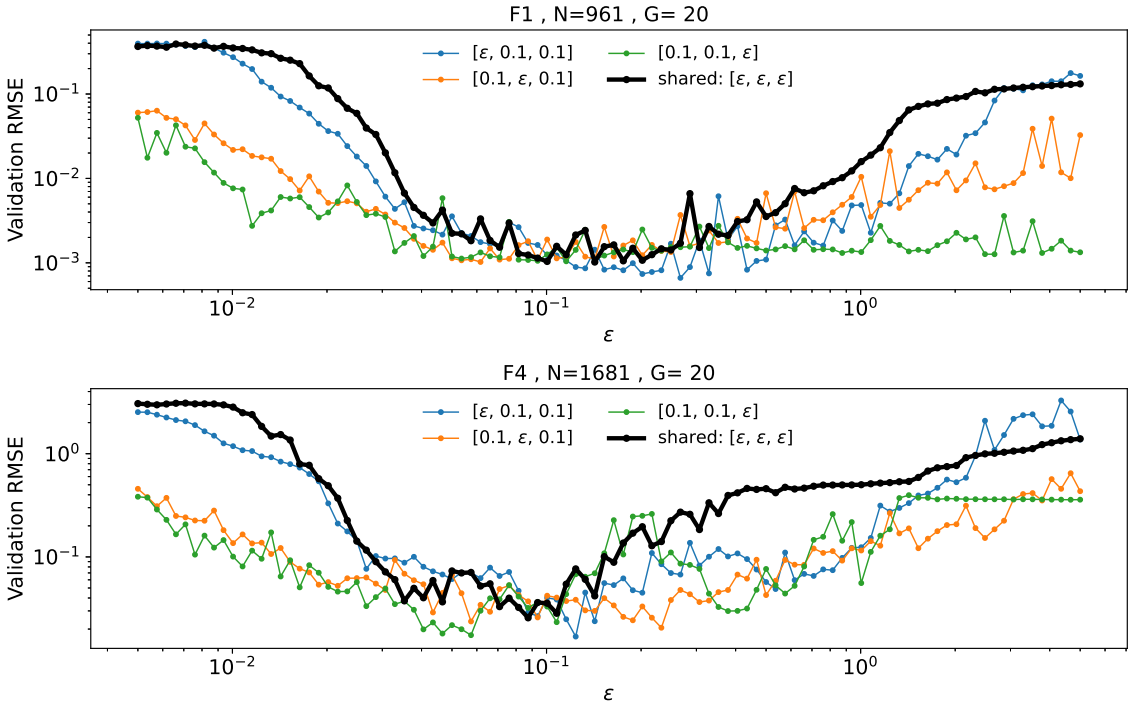


Figure 2: Layer-wise sensitivity of the validation RMSE to the Gaussian scale for the architecture $2 \to 12 \to 12 \to 1$. The reference value is $\epsilon_0 = 0.1$, and the four schedules are $(\epsilon, \epsilon_0, \epsilon_0)$, $(\epsilon_0, \epsilon, \epsilon_0)$, $(\epsilon_0, \epsilon_0, \epsilon)$, and $(\epsilon, \epsilon, \epsilon)$. For both $F1$ and $F4$, varying the first-layer scale $(\epsilon, \epsilon_0, \epsilon_0)$ tracks the all-layer response $(\epsilon, \epsilon, \epsilon)$ much more closely than varying only the deeper layers. This confirms that the first layer carries the dominant sensitivity to the Gaussian scale.

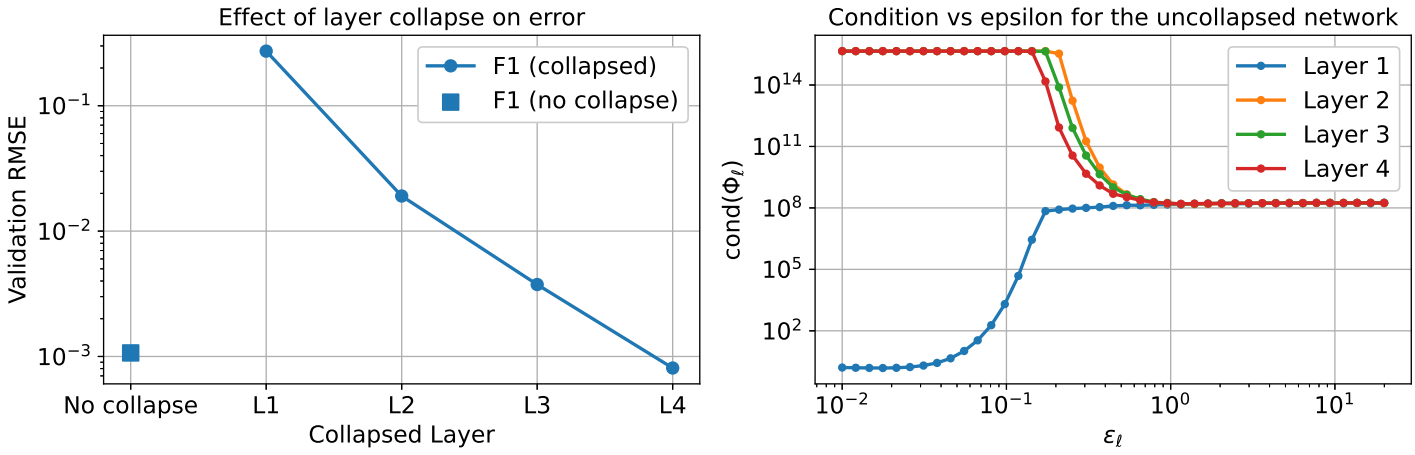


Figure 3: Numerical evidence for first-layer dominance for the architecture $[2, 12, 12, 12, 1]$. **Left:** validation RMSE after forcing one layer at a time into the large-ϵ collapse regime, while keeping the remaining layers fixed at $\epsilon = 0.1$. The no-collapse case corresponds to using $\epsilon = 0.1$ in all layers. Collapsing earlier layers, especially the first one, causes the largest increase in error. **Right:** spectral condition number $\kappa(\Phi_\ell)$ of the feature matrix in each layer versus ϵ for the uncollapsed network. The first layer is the only one with a clear low-conditioning regime.

4 First-Layer Conditioning and a Practical Gaussian Scale Interval

The goal of this section is to identify a numerically stable operating regime for the first-layer Gaussian feature construction and to translate that regime into a practical rule for choosing the Gaussian scale parameter. Since the empirical first-layer kernel matrix in (20) is generated by the pre-weighting feature matrix

$$\Phi \in \mathbb{R}^{N \times dG},$$

defined in (19), the relevant object for stability analysis is the feature matrix Φ itself, or equivalently its coordinate-wise blocks $\Phi^{(i)}$, rather than the learned coefficient matrix applied afterward.

4.1 Conditioning Diagnostics for the First-Layer Feature Matrix

To see how the Gram matrix arises, consider one output component and let $y \in \mathbb{R}^N$ denote the target values on the sample set $\mathcal{X}$. For a fixed first-layer feature matrix Φ, fitting coefficients by least squares gives

$$L(w) = \frac{1}{2}\|\Phi w - y\|_2^2, \qquad w \in \mathbb{R}^{dG}. \tag{40}$$

The corresponding normal equations are

$$\Phi^\top \Phi\, w = \Phi^\top y, \tag{41}$$

so the Gram matrix $\Phi^\top\Phi$ appears naturally in coefficient recovery. However, column dependence, numerical rank loss, and basis degeneracy originate in Φ before they are inherited and amplified by $\Phi^\top\Phi$.

Let

$$\sigma_1(\Phi) \geq \sigma_2(\Phi) \geq \cdots \geq \sigma_r(\Phi) > 0 \tag{42}$$

denote the positive singular values of Φ, where $r = \operatorname{rank}(\Phi)$. Define the spectral condition number

$$\kappa(\Phi) = \frac{\sigma_{\max}(\Phi)}{\sigma_{\min}(\Phi)} = \frac{\sigma_1(\Phi)}{\sigma_r(\Phi)}, \tag{43}$$

when Φ has full column rank, and $\kappa(\Phi) = \infty$ otherwise. Similarly,

$$\kappa(\Phi^\top\Phi) = \frac{\lambda_{\max}(\Phi^\top\Phi)}{\lambda_{\min}(\Phi^\top\Phi)} \tag{44}$$

when $\Phi^\top\Phi$ is nonsingular, and $\kappa(\Phi^\top\Phi) = \infty$ otherwise.

Proposition 3 (Relation between feature-matrix and Gram-matrix conditioning)**.** *Let $\Phi \in \mathbb{R}^{N\times M}$ with $M = dG$. Then:*

(i) If Φ has full column rank, then

$$\kappa(\Phi^\top \Phi) = \kappa(\Phi)^2. \tag{45}$$

(ii) If Φ is rank deficient, then $\Phi^\top\Phi$ is singular and

$$\kappa(\Phi) = \kappa(\Phi^\top \Phi) = \infty. \tag{46}$$

Proof. The eigenvalues of $\Phi^\top\Phi$ are the squared singular values of Φ, namely

$$\lambda_j(\Phi^\top \Phi) = \sigma_j(\Phi)^2.$$

Hence, if Φ has full column rank,

$$\kappa(\Phi^\top \Phi) = \frac{\sigma_1(\Phi)^2}{\sigma_r(\Phi)^2} = \kappa(\Phi)^2.$$

If Φ is rank deficient, then some singular value of Φ is zero, so $\Phi^\top\Phi$ has a zero eigenvalue and is singular as well. □

Proposition 3 shows that the Gram matrix does not introduce a new instability mechanism; it squares the instability already present in the feature matrix. For this reason, the primary diagnostic for the first-layer Gaussian feature construction is $\kappa(\Phi)$, not $\kappa(\Phi^\top\Phi)$.

The numerical quality of the first-layer Gaussian basis is determined by the extreme singular values

$$\sigma_{\max}(\Phi) = \sigma_1(\Phi), \qquad \sigma_{\min}(\Phi) = \sigma_r(\Phi). \tag{47}$$

To assess numerical rank, we use the standard floating-point tolerance

$$\mathrm{tol} = \max(N, M)\, \sigma_{\max}(\Phi)\, \varepsilon_{\mathrm{mach}}, \qquad M = dG. \tag{48}$$

Accordingly, the first-layer feature matrix is numerically full rank precisely when

$$\sigma_{\min}(\Phi) > \mathrm{tol}. \tag{49}$$

In float32 arithmetic, a practical additional requirement is that the feature matrix should remain moderately conditioned. Motivated by Proposition 3, we impose

$$\kappa(\Phi) < 3 \times 10^3. \tag{50}$$

This corresponds to

$$\kappa(\Phi^\top \Phi) < 9 \times 10^6,$$

which remains within a conservative numerical range for float32-based training.

Combining (49) and (50), we define the first-layer diagnostic criteria as

$$\sigma_{\min}(\Phi) > \mathrm{tol} \qquad \text{and} \qquad \kappa(\Phi) < 3 \times 10^3. \tag{51}$$

The first condition detects numerical rank loss, while the second controls near-dependence before rank is lost. Together, they identify the stable operating regime of the first-layer Gaussian feature construction. These diagnostics may be applied either to the full matrix Φ or to the coordinate-wise blocks $\Phi^{(i)}$, but in all cases the assessment is performed before weighting.

4.2 Derivation of a Practical Gaussian Scale Interval

Our goal is to identify a practically useful interval for the Gaussian scale parameter ϵ, of the form

$$\epsilon \in \left[\frac{1}{G-1},\, \epsilon_{\mathrm{cond}}(N, G)\right], \tag{52}$$

where G is the number of Gaussian centers per coordinate. The lower bound $\frac{1}{G-1}$ is the center spacing, while the upper bound $\epsilon_{\mathrm{cond}}(N, G)$ is chosen from first-layer conditioning. The reason is that, as ϵ becomes too large, the Gaussian columns become increasingly similar, the first-layer feature matrix becomes ill-conditioned, and, by Proposition 2, the

representation moves toward collapse. We then simplify $\epsilon_{\text{cond}}(N, G)$ into a rule depending only on G, leading to an explicit interval that is easy to use in practice.

The lower endpoint

$$\epsilon_{\min} = \frac{1}{G-1} \tag{53}$$

is the spacing between adjacent uniformly distributed centers. This is the natural geometric scale below which the Gaussian basis becomes too localized. The main task is therefore to determine a practical upper boundary.

To do this, we use the first-layer feature matrix $\Phi(\epsilon)$ and define two diagnostic markers. First, let

$$\epsilon_\kappa = \arg\min_{\epsilon>0} \left|\log \kappa\big(\Phi(\epsilon)\big) - \log\big(3 \times 10^3\big)\right| \tag{54}$$

denote the Gaussian scale at which the first-layer condition number is closest to the practical threshold. Second, let

$$\epsilon_{\text{rank}} = \inf\Big\{\epsilon > 0 : \sigma_{\min}\big(\Phi(\epsilon)\big) \le \text{tol}(\epsilon)\Big\} \tag{55}$$

denote the onset of numerical rank loss.

Figure 4 compares these two markers with the validation RMSE and the first-layer condition number. Across all four test cases, ϵ_κ and ϵ_{rank} occur very close to one another. Thus, for the first-layer Gaussian feature matrix, the practical onset of ill-conditioning and the onset of numerical rank loss are nearly simultaneous.

As an additional accuracy reference, Figure 4 also includes the validation RMSE obtained by a Chebyshev KAN of the same architecture. In that comparison, the Chebyshev degree is chosen as

$$m = G - 1,$$

so that each Chebyshev edge expansion has $m + 1 = G$ basis coefficients, matching the G Gaussian coefficients used per edge in the Gaussian KAN. The Chebyshev result is shown only as a reference accuracy level, not as part of the first-layer conditioning analysis.

The same figure also shows that the minimum validation RMSE occurs near this transition region. For smaller values of ϵ, the basis is too localized and the error remains relatively large. As ϵ increases toward ϵ_κ and ϵ_{rank}, neighboring Gaussian features overlap sufficiently to improve approximation while the feature matrix remains numerically stable. Beyond this regime, $\kappa(\Phi)$ grows rapidly, the numerical full-rank condition fails, and the validation error deteriorates. In most cases, the best-performing region lies between the geometric scale $1/(G-1)$ and the conditioning boundary.

This motivates defining the upper endpoint in (52) by

$$\epsilon_{\text{cond}}(N, G) = \arg\min_{\epsilon>0} \left|\log \kappa\big(\Phi(\epsilon)\big) - \log\big(3 \times 10^3\big)\right|. \tag{56}$$

That is, $\epsilon_{\text{cond}}(N, G)$ is the value of ϵ at which the first-layer feature matrix reaches the practical conditioning threshold. Hence the initial conditioning-based interval is

$$\epsilon \in \left[\frac{1}{G-1},\, \epsilon_{\text{cond}}(N, G)\right]. \tag{57}$$

The next question is whether $\epsilon_{\text{cond}}(N, G)$ can be simplified. For each pair (N, G), we compute $\epsilon_{\text{cond}}(N, G)$ by sweeping ϵ and selecting the value whose condition number is closest to the threshold. In the regression study, we use $N \in [400, 3600]$ and $G \in [6, 30]$, with multiple low-discrepancy sample realizations for each configuration. The resulting values are aggregated across seeds by a geometric mean, yielding one representative conditioning scale for each pair (N, G). The full set of pairs is then divided into training and validation subsets using a 70%/30% split.

To identify the dominant dependence of $\epsilon_{\text{cond}}(N, G)$, we fit the log-linear model

$$\log \epsilon_{\text{cond}} = \log C - q \log(G-1), \tag{58}$$

or equivalently,

$$\epsilon_{\text{cond}} \approx C\,(G-1)^{-q}. \tag{59}$$

The purpose of this fit is to test whether the conditioning boundary is governed mainly by the center spacing and only weakly by the number of samples.

The regression results show that the dominant dependence of ϵ_{cond} is on G, while the influence of N is weak once the samples cover $[0, 1]$ sufficiently densely. This is consistent with the fact that the overlap of neighboring Gaussian columns is controlled primarily by the center spacing $1/(G-1)$.

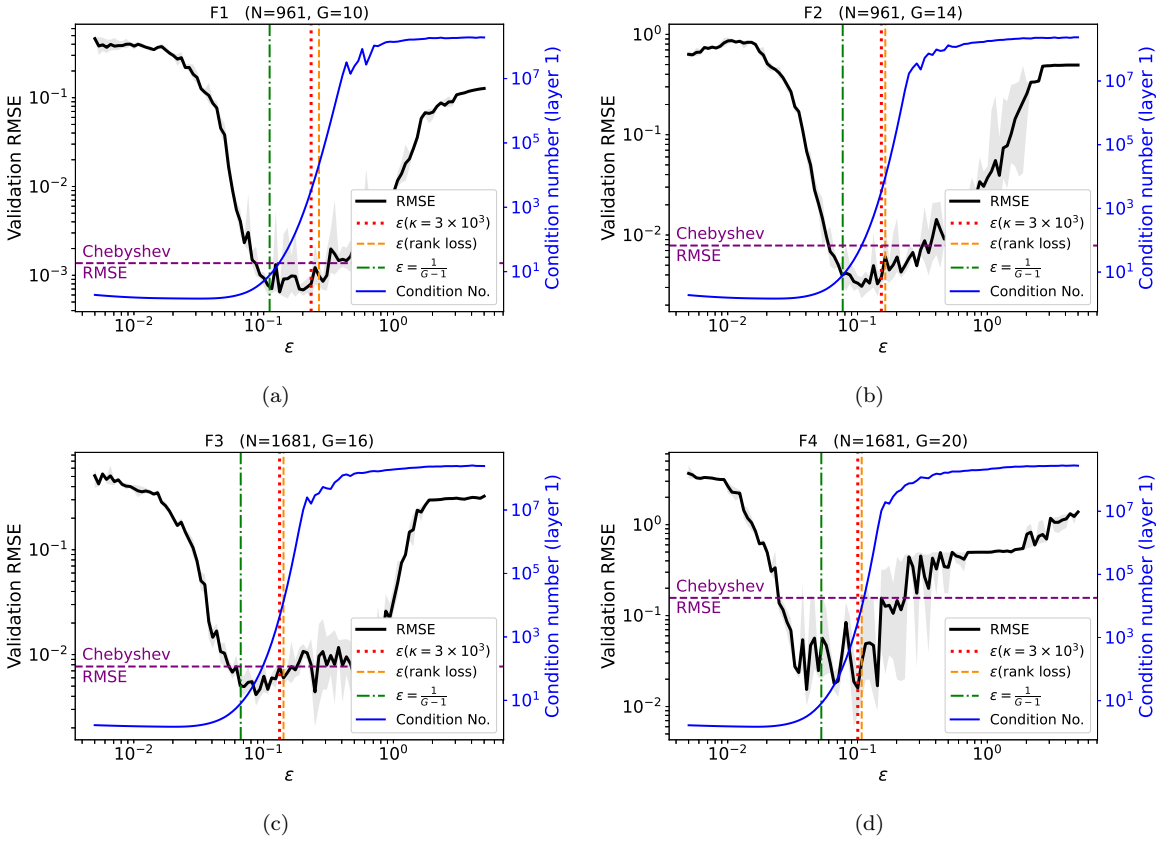


Figure 4: Empirical localization of the conditioning-based interval. Each panel shows the validation RMSE, the first-layer condition number $\kappa(\Phi)$, the lower geometric scale $\epsilon = 1/(G-1)$, the conditioning-based marker ϵ_κ, and the rank-loss marker ϵ_{rank} defined by $\sigma_{\min}(\Phi) \le \text{tol}$. Across all cases, ϵ_κ and ϵ_{rank} are nearly coincident, and the minimum validation RMSE occurs near this transition. This identifies the upper boundary of the useful scale interval. The horizontal purple line in each panel marks the validation RMSE of the corresponding Chebyshev KAN reference model, with degree $m = G-1$, so that the number of basis coefficients per edge matches the Gaussian case (see paragraph 4.2 for the Chebyshev discussion.)

Fitting (59) on the training set yields

$$\epsilon_{\text{cond}} \approx 2.8127\,(G-1)^{-1.1297}, \tag{60}$$

with validation coefficient of determination $R^2 \approx 0.997$. Since the fitted exponent is close to 1, this rule can be simplified to

$$\epsilon_{\text{cond}} \approx \frac{2}{G-1}, \tag{61}$$

which remains highly accurate on the validation set, with $R^2 \approx 0.985$.

To isolate the dependence on G, the validation values of ϵ_{cond} are also aggregated over all samples sharing the same number of centers, using a geometric mean across N. The resulting curve follows (61) closely across the full tested range of G, again showing that the principal dependence is on the center spacing rather than on the sample count.

We therefore replace the conditioning-based interval (57) by the explicit practical rule

$$\epsilon \in \left[\frac{1}{G-1}, \frac{2}{G-1}\right]. \tag{62}$$

The lower endpoint keeps the Gaussian scale comparable to the spacing of adjacent centers, while the upper endpoint places the first-layer feature matrix near the empirically observed conditioning boundary.

Figure 5 validates this simplification. The fitted power law gives an excellent approximation to the measured values of ϵ_{cond}, and the simplified rule $2/(G-1)$ remains highly accurate on held-out data. Thus, little is lost by replacing the fitted law with the simpler expression (61).

The conclusion of this section is therefore direct: the useful operating range for ϵ is

$$\epsilon \in \left[\frac{1}{G-1}, \frac{2}{G-1}\right],$$

and this range is determined primarily by the number of centers G, not by the sample count N.

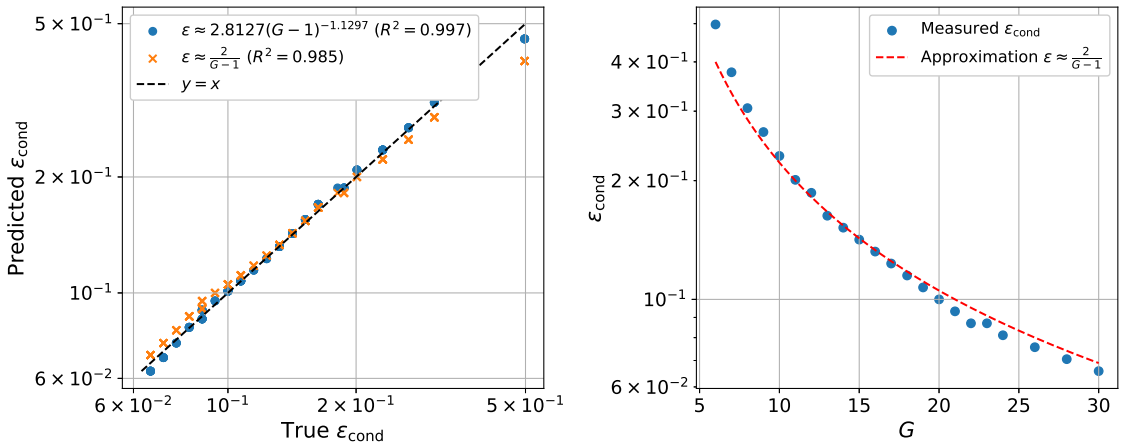


Figure 5: Simplification of the conditioning-based upper bound $\epsilon_{\text{cond}}(N, G)$. **Left:** predicted versus measured values of ϵ_{cond} on held-out (N, G) pairs. The fitted power law (60) achieves $R^2 \approx 0.997$, while the simplified rule (61) remains highly accurate with $R^2 \approx 0.985$. The dashed line denotes exact agreement. **Right:** geometric-mean aggregation of the measured values over validation samples sharing the same G. The empirical curve is well approximated by (61), showing that the conditioning-based upper bound is governed primarily by the number of centers.

As a final remark, one may also define the conditioning-based upper bound for ϵ using classical bounds from meshfree RBF theory, for example through estimates involving both the smallest and largest singular values of the feature matrix [40]. Such bounds are theoretically meaningful, but in the present setting they are too impractical for routine use. Since our aim here is a rule that is both simple and usable in practice, we do not pursue that direction.

Chebyshev reference. With the practical interval now identified, we return briefly to Figure 4 to comment on the Chebyshev reference line. The Chebyshev implementation is included only as an additional accuracy benchmark. For a fair comparison, the Chebyshev degree is chosen as $m = G - 1$, so that each Chebyshev edge expansion has $m + 1 = G$ trainable coefficients, matching the G Gaussian basis coefficients per edge. In this reference model, tanh is used between layers to keep the inputs within the admissible range of the Chebyshev basis, and the final layer also uses a Chebyshev expansion rather than a linear head. Under this matched per-edge coefficient budget, Figure 4 shows that a suitably scaled Gaussian KAN can attain validation errors comparable to, and in some cases lower than, the corresponding Chebyshev KAN.

This comparison also highlights a practical difference between the two representations. In the Gaussian KAN, conditioning and accuracy can be adjusted directly through the centers and the scale parameter ϵ. In contrast, the Chebyshev KAN is tied more rigidly to the polynomial degree: increasing the degree may improve approximation power, but often at the cost of reduced stability (see Figure 4(d)), whereas lower degrees are usually more stable but may sacrifice accuracy. Various heuristic remedies have therefore been introduced in the Chebyshev literature, including localization of the basis [7], interlayer normalization [53], and architectural modifications such as additional tanh nonlinearities or alternative readout layers [6]. By comparison, the Gaussian KAN offers a cleaner mechanism in which the same scale parameter directly influences both conditioning and accuracy.

The role of the present comparison is to show that the Gaussian KAN, when equipped with a properly chosen scale, can already be as accurate as a standard Chebyshev KAN. This supports the main purpose of the present work, namely, to understand how the Gaussian scale parameter ϵ governs both conditioning and accuracy, and how it can be selected in a principled and practically useful way. A definitive comparison between Chebyshev and Gaussian KANs would require a separate study based on carefully tuned models under matched architectures, training strategies, and stabilization techniques.

4.3 Effect of Collocation Density N

We now study how the number of collocation points affects approximation in the Gaussian KAN. The goal is to separate two effects: N controls how well the domain is resolved, while ϵ controls the quality and stability of the Gaussian feature representation. We show that increasing N improves accuracy, but only when ϵ is chosen in a suitable range.

All experiments in this subsection are performed on the unit square

$$\Omega = [0,1]^2, \tag{63}$$

using Halton points

$$X_N = \{x^1, \dots, x^N\} \subset \Omega \tag{64}$$

as the training set. Since Halton points are quasi-uniform, their geometric resolution is described by the fill distance

$$h_{X_N,\Omega} = \sup_{x\in\Omega} \min_{x^n \in X_N} \|x - x^n\|_2, \tag{65}$$

which in two dimensions satisfies

$$h_{X_N,\Omega} \sim N^{-1/2}. \tag{66}$$

Thus, increasing N improves the geometric resolution of the training set.

To generate a systematic refinement sequence, we choose

$$N \in \{k^2 : \ k = 5, 7, 9, \dots, 87\}. \tag{67}$$

This gives a monotone progression of collocation densities and makes the scaling in (66) easy to interpret.

Throughout this subsection, the grid size is fixed at $G = 20$. Therefore, the practical interval from (62) becomes

$$\epsilon \in \left[\frac{1}{19}, \frac{2}{19}\right] \approx [0.0526,\ 0.1053]. \tag{68}$$

We now examine how this interval behaves as N changes.

Error landscape in ϵ for different N. For fixed architecture and grid size, let

$$E(N, \epsilon) \tag{69}$$

denote the validation RMSE obtained with N collocation points and Gaussian scale ϵ. Figure 6 plots $E(N, \epsilon)$ for two representative training sizes, $N = 961$ and $N = 3249$, with the vertical lines marking the interval in (68).

Across all four targets, the same basic structure appears: the error is large for very small ϵ, decreases into a low-error region, and rises again for large ϵ. Thus the dependence on ϵ remains essentially U-shaped as N changes. Increasing N lowers the overall error level, but shifts the useful ϵ-window only mildly. In particular, the error-minimizing region remains inside or close to (68). This shows that the conditioning-based interval remains informative even as the collocation density changes.

Best attainable error as a function of N. To isolate the effect of collocation density, define

$$E_{\mathrm{opt}}(N) = \min_{\epsilon} E(N, \epsilon), \tag{70}$$

that is, the best validation RMSE obtained over the ϵ-sweep for a given N. Figure 7 plots $E_{\mathrm{opt}}(N)$ against N. The best error decreases monotonically for all four targets, confirming that denser collocation improves approximation quality once the Gaussian scale is chosen well.

The observed decay also reflects target regularity. For the smoother targets F1 and F2, the visible pre-asymptotic behavior is approximately aligned with

$$E_{\mathrm{opt}}(N) \sim h^3 \sim N^{-3/2}, \tag{71}$$

whereas the less regular targets F3 and F4 are closer to

$$E_{\mathrm{opt}}(N) \sim h^2 \sim N^{-1}. \tag{72}$$

This is consistent with the standard expectation that smoother targets admit faster convergence, while reduced regularity lowers the observed rate; see Chapter 17 of Fasshauer [40].

Interaction between N and ϵ. Figure 8 shows the validation RMSE versus N for the fixed Gaussian scales $\epsilon \in \{0.06, 0.08, 0.1\}$. In all cases, the error decreases as N increases, but the decay still depends on the choice of ϵ. Thus, N and ϵ play different roles: N increases geometric resolution, while ϵ controls how effectively the Gaussian feature representation uses that resolution.

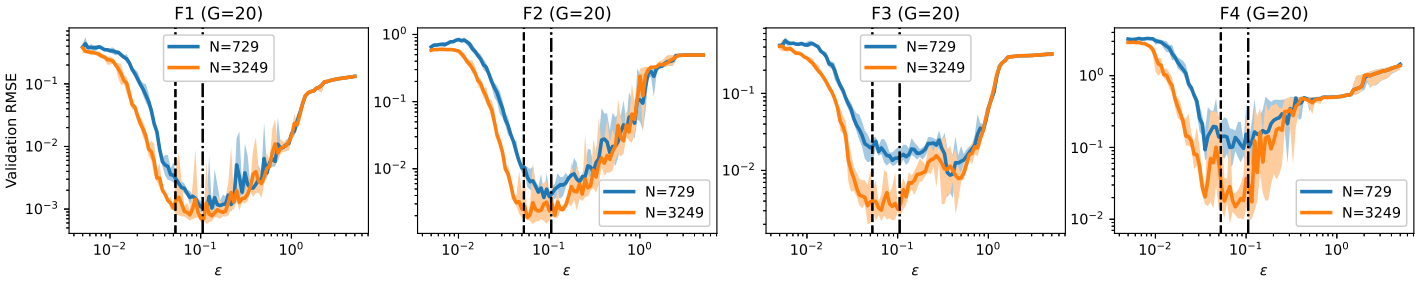


Figure 6: Validation RMSE as a function of the Gaussian scale ϵ for two representative collocation densities, $N = 729$ and $N = 3249$, with $G = 20$. The vertical markers indicate $\epsilon = 1/(G-1)$ and $\epsilon = 2/(G-1)$. Across all four targets, increasing N lowers the error, while the low-error region remains concentrated near the conditioning-based interval.

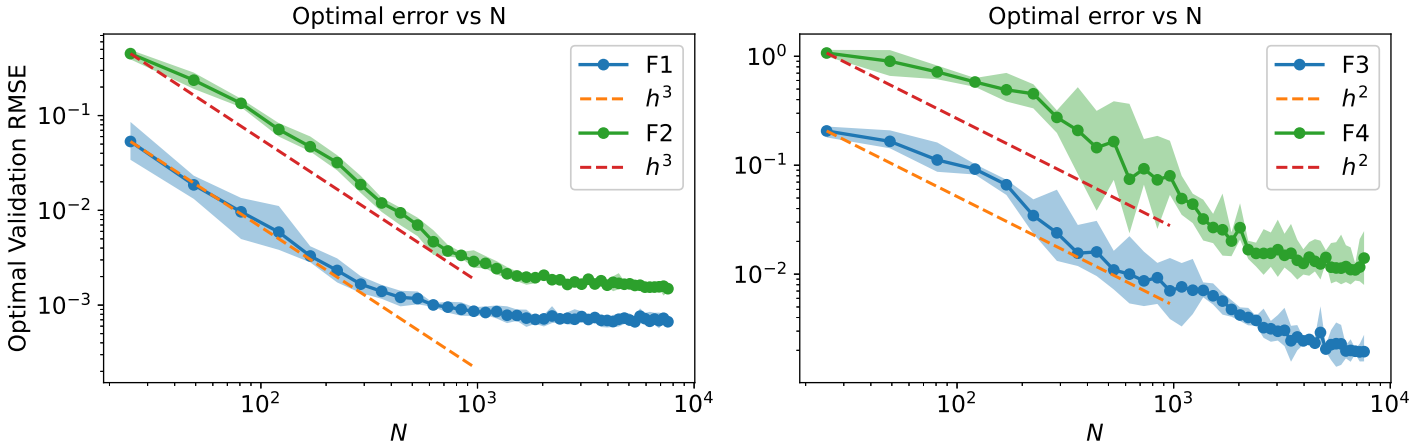


Figure 7: Best validation RMSE $E_{\mathrm{opt}}(N)$ from (70) as a function of the number of collocation points. For the smoother targets F1 and F2, the visible decay is approximately aligned with the h^3 reference, while F3 and F4 are closer to the h^2 reference. In all cases, denser Halton sampling improves the best attainable accuracy.

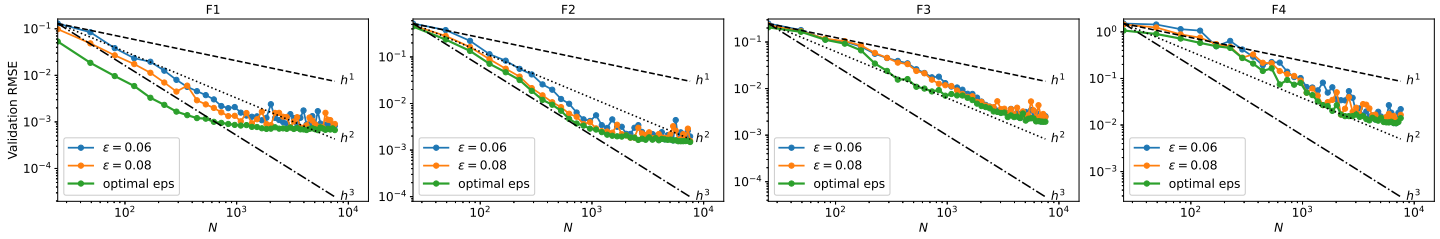


Figure 8: Validation RMSE versus N for the fixed Gaussian scales $\epsilon = 0.06, 0.08$, together with the near-optimal choice at each N with $G = 20$.

4.4 Effect of the Number of Gaussian Centers G

We next study the effect of the grid resolution, that is, the number of shared Gaussian centers in each one-dimensional feature map. For fixed architecture and training size N, let

$$\mathcal{C}_G = \{c_1, \ldots, c_G\} \subset [0, 1] \tag{73}$$

denote the uniformly distributed centers. The conditioning-based rule in (61) and (62) already shows that G and ϵ cannot be chosen independently. As G increases, the center spacing Δ_c decreases, so the useful scale range must also move to smaller ϵ.

Fixed ϵ versus adaptive ϵ. Figure 9 shows $E(G, \epsilon)$ for several fixed values of ϵ, together with a near-optimal choice selected separately for each G. The fixed-ϵ curves depend strongly on the chosen scale: a value that works well for one

grid size can be clearly suboptimal for another. By contrast, the near-optimal choice gives a much more stable error curve across the full range of G. Thus, refining the grid alone is not enough; ϵ must be adjusted with the center spacing.

Shift of the low-error ϵ-window. Figure 10 compares the full ϵ-sweeps for two representative grid sizes, $G = 10$ and $G = 32$. Both cases show the same U-shaped dependence on ϵ, but the effect of increasing G is different from the effect of increasing N. Increasing N mainly pushes the curve downward, reducing the error level while leaving the useful ϵ-window roughly in the same place. Increasing G, instead, shifts the low-error region to the left, that is, toward smaller ϵ, and also makes the good-ϵ range wider. This is exactly what the proposed range predicts, since that range is tied to the center spacing.

At the same time, the minimum error itself changes much less than the location of the minimizer. This is consistent with the matrix viewpoint developed earlier: increasing G mainly changes the geometry of the basis and the position of the stable operating range, rather than acting as a direct accuracy parameter by itself.

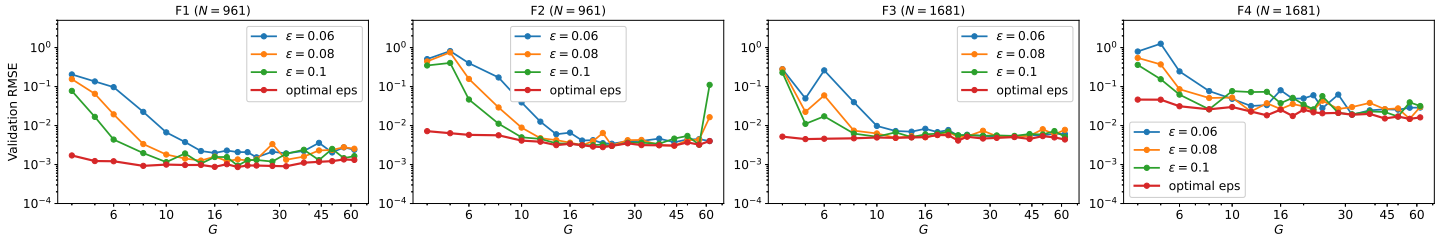


Figure 9: Validation RMSE versus grid resolution G for several fixed Gaussian scales and for a near-optimal scale chosen separately at each G. The fixed-ϵ curves vary strongly with G, whereas the near-optimal choice yields a much more stable error profile.

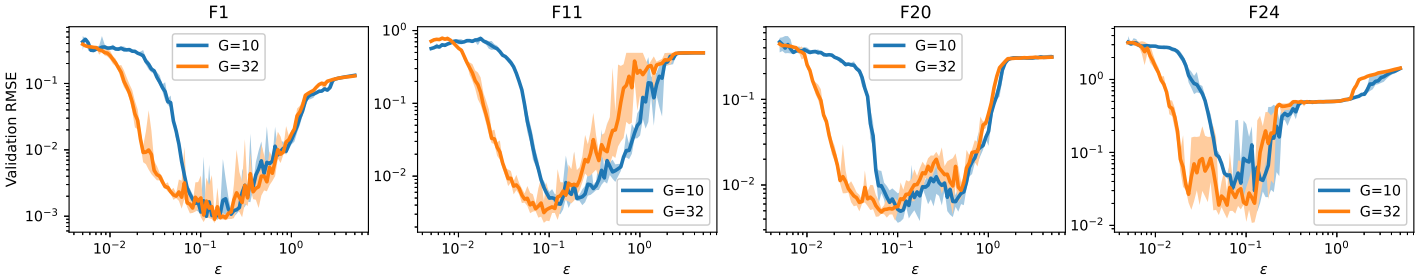


Figure 10: Validation RMSE as a function of the Gaussian scale ϵ for two representative grid resolutions, $G = 10$ and $G = 32$. As G increases, the low-error region shifts to smaller ϵ and becomes wider.

4.5 Effect of Network Architecture

We next study how the network architecture affects the relation between approximation error and the Gaussian scale. Throughout this subsection, the grid resolution is fixed at $G = 20$, so the conditioning-based reference interval is $\epsilon \in [1/19, 2/19] \approx [0.0526, 0.1053]$. Figure 11 shows that all tested architectures retain the same U-shaped dependence on ϵ. What changes is the minimum error and the width of the low-error region. Since the first layer is the only layer defined directly on the input domain, it still determines the relevant scale regime. Increasing width or depth does not move the left side of the U-shaped curve, which is the part controlled by the first-layer geometry. Instead, larger architectures mainly enlarge the range of good ϵ from the right. This is different from increasing G, which shifts the useful range to the left. The effect is more visible for the less smooth targets F3 and F4. In the top row, the wider architecture $(2, 20, 20, 1)$ generally attains lower error than $(2, 4, 4, 1)$ and makes the low-error region broader. In the bottom row, the deeper architecture $(2, 8, 8, 8, 1)$ also lowers the error and broadens the admissible range compared with $(2, 8, 1)$, again mainly on the larger-ϵ side. Thus, a scale range that is suitable for a shallower Gaussian representation remains admissible for a deeper Gaussian KAN; deeper layers may enlarge the useful range, but they do not shrink the range already set by the first layer.

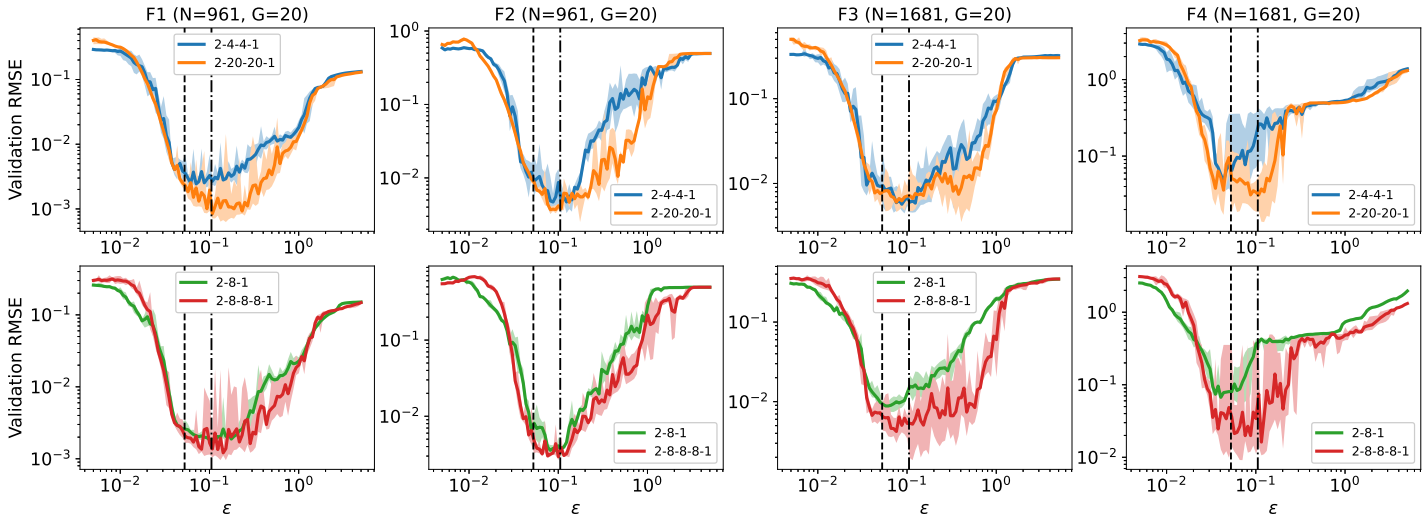


Figure 11: Validation RMSE versus the Gaussian scale ϵ for different Gaussian KAN architectures at fixed grid size $G = 20$. The two vertical markers indicate $\epsilon = 1/(G-1)$ and $\epsilon = 2/(G-1)$.

4.6 Effect of Input Dimension

We finally investigate how the Gaussian scale behaves as the input dimension increases. For each dimension d, we consider the smooth target

$$f_d(x) = \exp\left(\frac{1}{d}\sum_{i=1}^{d}\left[\sin(\pi x_i) + \frac{1}{2}x_i^2\right]\right), \qquad x \in [0,1]^d, \tag{74}$$

which remains bounded and of comparable magnitude across dimensions. The experiments in Figure 12 use the settings listed in Table 1.

Table 1: Experimental settings used in the input-dimension study. The last column lists the hidden-layer widths, so for example [12] denotes one hidden layer with 12 neurons, while $[12, 12, 12]$ denotes three hidden layers with 12 neurons each.

Dimension d	Collocation points N	Hidden-layer widths
1	30	[12]
2	1000	$[12, 12]$
3	30000	$[12, 12, 12]$
4	50000	$[12, 12, 12, 12]$

Figure 12 shows that the validation RMSE keeps the same U-shaped dependence on ϵ from 1D to 4D. Very small ϵ leads to overly localized features, while very large ϵ leads to the flat and numerically unfavorable regime. Most importantly, the low-error region remains close to the same two reference lines so increasing the ambient dimension does not introduce a new preferred scale.

This is consistent with the first-layer matrix construction in (19). The feature matrix is built by concatenating coordinate-wise Gaussian blocks,

$$\Phi = \begin{bmatrix}\Phi^{(1)} & \cdots & \Phi^{(d)}\end{bmatrix}, \tag{75}$$

so increasing d adds more one-dimensional blocks but does not change the spacing mechanism inside each block. The same shared scale ϵ still controls the overlap between neighboring centers, and therefore the same conditioning-based interval remains relevant.

The minimum error does increase with dimension, as expected, because the approximation problem becomes harder. However, the shape of the error curve and the location of the useful ϵ-window remain essentially unchanged. Thus, in the present separable Gaussian KAN, the selection of ϵ is still controlled primarily by the first-layer Gaussian geometry rather than by the ambient dimension itself.

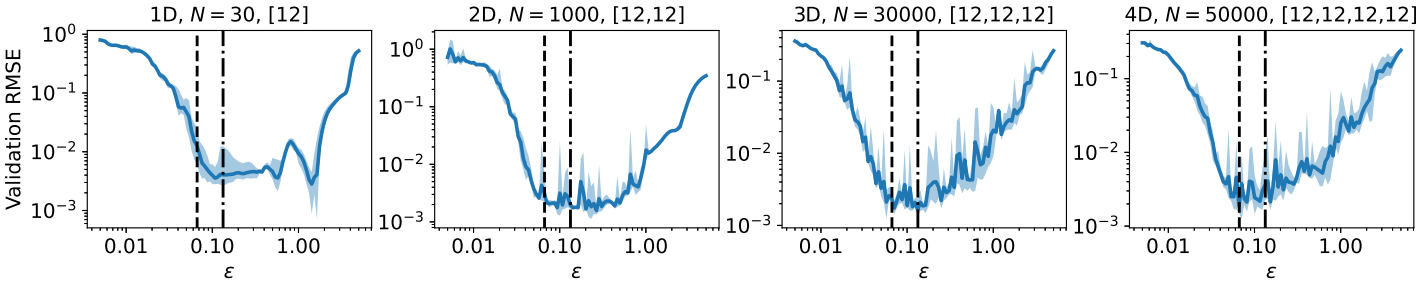


Figure 12: Validation RMSE versus the Gaussian scale ϵ for input dimensions $d = 1, 2, 3, 4$. The settings are those listed in Table 1. The two vertical markers indicate the reference scales $\epsilon = 1/(G-1)$ and $\epsilon = 2/(G-1)$. Across dimensions, the same U-shaped dependence on ϵ is observed, and the low-error region remains close to the same conditioning-based interval.

4.7 Gaussian KAN with Shared Centerwise Variable Scales

The fixed-scale Gaussian feature map in (5) can be generalized by assigning a different scale to each center while still sharing the same scale vector across all neurons and all layers. Let

$$\mathcal{E} = \{\epsilon_1, \ldots, \epsilon_G\}, \qquad \epsilon_g > 0, \quad g = 1, \ldots, G, \tag{76}$$

be the vector of centerwise Gaussian scales associated with the shared centers $\mathcal{C} = \{c_1, \ldots, c_G\} \subset [0, 1]$. Then the Gaussian feature map becomes

$$\varphi_{\mathcal{E}}(t) = \begin{bmatrix} \exp\left(-\dfrac{(t-c_1)^2}{\epsilon_1^2}\right) \\ \vdots \\ \exp\left(-\dfrac{(t-c_G)^2}{\epsilon_G^2}\right) \end{bmatrix} \in \mathbb{R}^G, \qquad t \in \mathbb{R}. \tag{77}$$

Thus, each center c_g has its own width ϵ_g, while the same vector $\mathcal{E}$ is reused throughout the whole network. In the experiments below, the variable scales are sampled from a normal distribution on

$$\epsilon_g \in [\epsilon_{\min}, \epsilon_{\max}], \qquad \epsilon_{\max} = \frac{2}{G-1}, \tag{78}$$

and then assigned to the shared centers $\{c_g\}_{g=1}^G$, randomly. Hence, this model keeps the same Gaussian KAN architecture, but introduces controlled nonuniformity in the basis widths across centers. Although the centers themselves remain uniformly distributed, allowing the widths to vary makes the *effective* coverage of the basis nonuniform across the domain. In this sense, the model gains some of the flexibility usually associated with nonuniform or data-adapted center placement, while still retaining the simplicity and interpretability of a fixed uniform center grid. Figure 13 compares this variable-scale model with the fixed-scale choice $\epsilon = 1/(G-1)$.

As Figure 13 shows, the shared centerwise variable-scale model can be more accurate than the single-ϵ version. In the present experiment, the scales are assigned randomly within the admissible range, so this result should be interpreted only as an initial demonstration rather than an optimized construction. The use of variable shape parameters has a long history in classical RBF methods, where nonuniform scales often improve accuracy over constant-scale formulations [54–56], and the present results indicate that this advantage can also carry over to the neural-network setting. More generally, one may vary not only the interval but also the sampling law for the scales, for example by using normal, log-normal, log-uniform, gamma, or chi-square distributions [54]. Thus, even with uniformly distributed centers, variable-scale Gaussian KANs can induce a more adaptive and spatially nonuniform representation, making them a flexible and promising extension of the fixed-scale model.

4.8 Training-MSE-Based Scale Search Within the Admissible Interval

The first-layer diagnostics do not select a single value of ϵ; they reduce the search to a numerically admissible interval. This is the main practical role of the range: instead of sweeping over all candidate scales, one only needs to search inside the conditioning-controlled region. Figure 14 shows that, within this reduced interval, the training MSE provides a useful criterion for selecting a working scale.

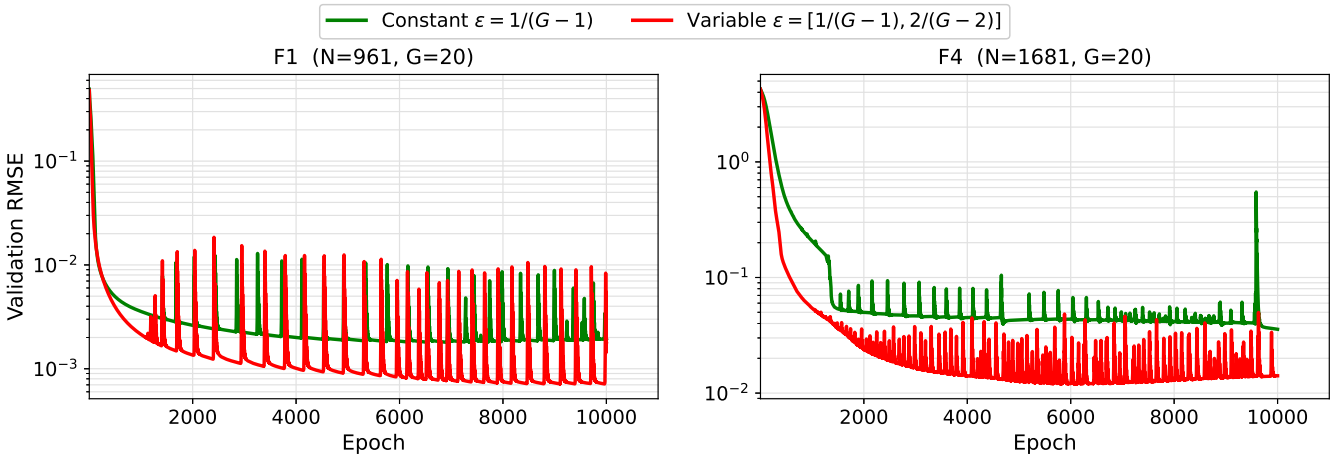


Figure 13: Validation RMSE versus epoch for the fixed-scale Gaussian KAN and the Gaussian KAN with shared variable scales. Left: $F1$ with $N = 961$, where the variable model uses range $R5$. Right: $F4$ with $N = 1681$. The fixed model uses the common scale $\epsilon = 1/(G-1)$, while the variable model assigns one scale ϵ_g to each center c_g, with the same scale vector shared across all neurons and all layers.

For the four representative problems $F1$, $F2$, $F3$, and $F4$, the low-training-error region at 500 epochs is located in essentially the same part of the ϵ-axis as the low-validation-RMSE region at 10000 epochs. The agreement is clearest for the smooth and oscillatory cases $F1$ and $F2$, and remains informative for the more difficult cases $F3$ and $F4$, where the curves are rougher but still identify the same favorable range.

The practical implications are straightforward:

- The admissible interval sharply reduces the ϵ-search space.
- Inside that interval, training MSE is sufficient to localize a good scale.
- This localization already appears after a small number of iterations.
- Hence one can perform a short sweep over ϵ, choose a candidate from the training-MSE curve, and run full training only for that value.
- This is especially useful when the exact solution is unavailable, so validation against the true target cannot be used for scale selection.

Therefore, the proposed interval is useful not only for stability control, but also for reducing the cost of scale selection. It turns the search for ϵ into a smaller optimization problem over a numerically justified range, and the early training MSE is usually sufficient to identify the relevant part of that range.

4.9 Helmholtz Problem: Physics-Informed Validation

To test whether the conditioning-based scale interval also remains useful in a physics-informed setting, we consider the two-dimensional Helmholtz problem

$$-\Delta u - \lambda u = f \qquad \text{in } \Omega = [0,1]^2, \tag{79}$$

with homogeneous Dirichlet boundary condition

$$u = 0 \qquad \text{on } \partial\Omega. \tag{80}$$

We choose the exact solution

$$u(x,y) = \sin(a_1 \pi x)\sin(a_2 \pi y), \qquad (x,y) \in [0,1]^2, \tag{81}$$

which vanishes on the boundary and therefore satisfies (80). Substituting (81) into (79) gives the forcing term

$$f(x,y) = \Big((a_1^2 + a_2^2)\pi^2 - \lambda\Big)\sin(a_1 \pi x)\sin(a_2 \pi y). \tag{82}$$

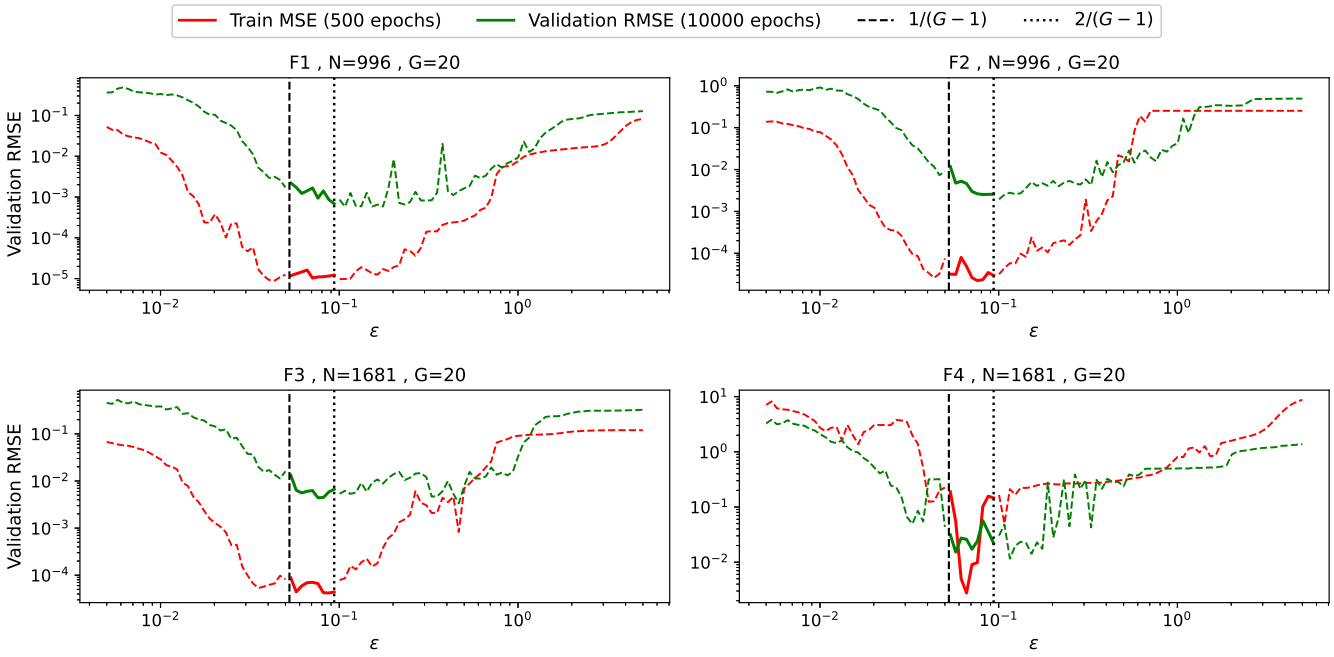


Figure 14: Training-MSE-based scale selection inside the admissible interval. Each panel shows the training MSE at 500 epochs and the validation RMSE at 10000 epochs as functions of ϵ. The vertical markers indicate the lower reference scale and the conditioning-based upper boundary of the search interval. Across all four examples, the early training-MSE curve already localizes essentially the same favorable ϵ-region as the final validation RMSE, so the admissible interval can be searched efficiently using only a short preliminary sweep.

In the experiments below, the PDE residual is weighted by

$$w_{\text{pde}} = 1, \tag{83}$$

and the boundary loss is weighted by

$$w_{\text{bc}} = 100. \tag{84}$$

The collocation size and Gaussian grid are fixed at

$$(N_{BC}, N_{PDE}) = (800, 2000), \qquad G = 20, \tag{85}$$

so the conditioning-based reference interval from (62) becomes

$$\epsilon \in \left[\frac{1}{19}, \frac{2}{19}\right] \approx [0.0526, 0.1053]. \tag{86}$$

We study three values of the Helmholtz parameter,

$$\lambda \in \{0, 10, 100\}, \tag{87}$$

for two exact solutions: the lower-frequency case

$$(a_1, a_2) = (1, 2), \tag{88}$$

and the anisotropic higher-frequency case

$$(a_1, a_2) = (1, 4). \tag{89}$$

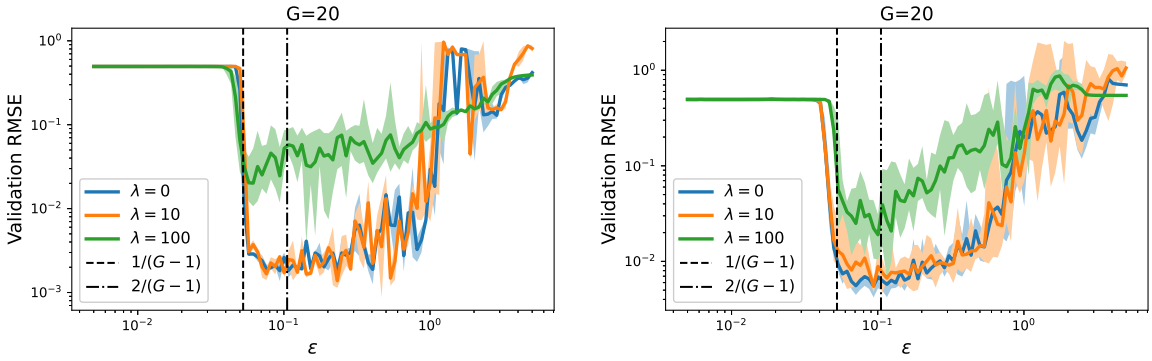
Figure 15 shows the validation RMSE as a function of the Gaussian scale ϵ, with the two vertical markers corresponding to the endpoints of (86). In both test cases, the same qualitative behavior seen in the regression experiments is recovered: the error is large for very small ϵ, decreases sharply into a low-error regime, and then increases again as ϵ becomes too large. Thus, the Helmholtz problem exhibits the same basic trade-off between insufficient overlap at small scales and degradation of the Gaussian representation at large scales.

Most importantly, the best-performing region remains within or close to the interval (86) for all three values of λ. For $\lambda = 0$ and $\lambda = 10$, the minimum error is attained clearly inside this interval in both problem settings. For the more

challenging case $\lambda = 100$, the error level is higher and the low-error region becomes broader and less stable, but it is still centered near the same conditioning-based range. This indicates that the first-layer scale rule derived earlier remains informative even after the loss is replaced by a PDE residual and boundary penalty.

The two exact solutions also illustrate a useful distinction. The case $(a_1, a_2) = (1, 2)$ is smoother and easier, so the minimum RMSE is lower and the low-error region is broader. The case $(a_1, a_2) = (1, 4)$ is more oscillatory in the y-direction, and therefore more demanding, which raises the minimum error and makes the curves more sensitive, especially for larger λ. Nevertheless, across both exact solutions and all tested values of λ, the preferred ϵ-window remains within or close to the same conditioning-based interval and stays tied to the same geometric scale set by the shared Gaussian centers. This shows that larger λ and more oscillatory solutions may increase the error and make the curves less stable, but they do not substantially change the preferred ϵ-window.

This example is intentionally limited to a single PDE family with a tunable parameter λ. Its role is not to provide an exhaustive PDE study, but to show that the interval suggested by the first-layer conditioning analysis continues to work in a representative physics-informed problem. A broader investigation across other PDEs, boundary conditions, and training regimes can be carried out in future work.



(a) (b)

Figure 15: Validation RMSE versus the Gaussian scale ϵ for the Helmholtz problem (79)–(80) with $(N_{BC}, N_{PDE}) = (800, 2000)$, $G = 20$, $w_{\text{pde}} = 1$, and $w_{\text{bc}} = 100$. The two vertical markers indicate $\epsilon = 1/(G-1)$ and $\epsilon = 2/(G-1)$, i.e. the endpoints of the conditioning-based interval (86). **(a)** Exact solution (81) with $(a_1, a_2) = (1, 2)$. **(b)** Exact solution (81) with $(a_1, a_2) = (1, 4)$. In both cases, and for all tested values $\lambda \in \{0, 10, 100\}$, the best-performing region lies within or close to the same conditioning-based range.

5 Conclusion

To the best of our knowledge, this work is the first to study Gaussian KANs from a kernel and conditioning viewpoint, with emphasis on how the Gaussian scale parameter ϵ governs both numerical stability and approximation accuracy. A matched Chebyshev reference was included as an additional benchmark, and the results show that, when ϵ is chosen properly, a Gaussian KAN can achieve accuracy comparable to, and in some cases better than, the corresponding Chebyshev KAN with the same number of trainable coefficients per edge. This comparison is not intended as a definitive optimized benchmark between two basis families; rather, it shows that Gaussian KANs are already highly competitive once the scale parameter is selected appropriately.

The results of this study consistently indicate that scale selection in Gaussian KANs is governed primarily by the first layer, since it is the only layer constructed directly on the input domain and any loss of distinguishability introduced there cannot be recovered by later layers. This has been supported both theoretically and experimentally, with the conditioning of the first layer closely reflecting the error behavior of the full network. From this viewpoint, the relevant stability object is the first-layer feature matrix Φ, rather than primarily the Gram matrix $\Phi^\top \Phi$. Using numerical full rank together with a practical conditioning threshold, we identified the interval

$$\epsilon \in \left[\frac{1}{G-1}, \frac{2}{G-1} \right],$$

which provides a simple and effective admissible range for Gaussian KANs. The experiments further show that this interval remains informative across different target functions, collocation densities N, grid sizes G, network architectures, input

dimensions, and also in a physics-informed Helmholtz problem. In all cases, the validation error exhibits a clear U-shaped dependence on ϵ, with the best-performing region lying within or close to the proposed interval.

An additional practical observation concerns whether the Gaussian scale should be shared across layers. The layer-wise sweeps show that varying the first-layer scale produces an error curve much closer to that of the fully shared-ϵ network than varying the scale only in deeper layers. This further supports the first layer as the dominant layer for scale selection. At the same time, allowing different scales across layers does not lead to a clear improvement in the best attainable accuracy. Although deeper layers often admit a broader range of acceptable scales, using the same well-chosen ϵ in all layers yields nearly the same optimum. Therefore, a shared Gaussian scale across layers is a sensible choice: it keeps the model simpler while maintaining essentially the same accuracy.

The numerical studies also clarify the roles of N and G. Increasing the number of collocation points N improves the best attainable accuracy and yields clear convergence with respect to N, provided that ϵ is chosen in a suitable range. Increasing the number of centers G mainly shifts the low-error region toward smaller ϵ and broadens the admissible range, rather than guaranteeing a substantially smaller optimum by itself. In practice, a relatively small G can sometimes attain nearly the same optimal error as a larger one while being computationally much cheaper, which makes the proposed scale rule useful not only for accuracy but also for efficient model design.

Beyond fixed-scale selection, the proposed interval also proved useful in more practical settings. A shared centerwise variable-scale construction defined within this range can improve accuracy over a single fixed ϵ. In addition, the training MSE was shown to be an effective proxy for locating a good scale within the admissible interval, often after only a small number of iterations. This greatly reduces the cost of scale search and is especially important in applications where no exact solution is available, so that validation against the true target cannot be used.

Overall, the results show that the Gaussian scale parameter in Gaussian KANs should not be viewed as an arbitrary hyperparameter. Rather, it can be selected in a principled way that simultaneously supports stability, accuracy, and efficient computation. In this sense, the conclusion of Larsson and Schaback that *the strong dependence of radial basis function techniques on scaling is a feature, not a bug* [28] naturally extends to neural architectures based on Gaussian radial basis functions.

6 Limitations and Future Work

This work focuses on first-layer conditioning. Deeper layers may still be ill-conditioned, but our results suggest that this does not necessarily harm performance when ϵ is chosen in the proposed range. In later layers, ill-conditioning may occur because the fixed centers are no longer aligned with the transformed coordinates, or because those coordinates become clustered or unevenly distributed. Improving this would likely require adaptive center relocation and a more complex multilevel training strategy, possibly together with updating collocation points during training. Such extensions would substantially complicate the model, while better conditioning would still not necessarily guarantee better accuracy. For this reason, we retain the standard Gaussian KAN architecture used in current practice.

Several directions remain open for future work. One natural extension is to study Gaussian KANs beyond regression and the representative PDE setting considered here, for example in classification and operator-learning problems. It would also be interesting to investigate other RBF-based KANs [57–59], to examine how the present conditioning-based perspective extends beyond the Gaussian case. Another promising direction is to use the proposed interval as a constrained training region for a learnable ϵ, instead of training the scale blindly over an unrestricted domain. This may improve stability, avoid collapse, and reduce tuning cost. Variable-scale Gaussian KANs also deserve further study. Although a shared centerwise variable-scale model has already been introduced here, the preliminary results suggest that it can outperform the fixed-scale version. This opens the possibility of studying not only the admissible range of variable scales, but also different distribution families and sampling laws, such as chi-square distributions, random perturbations with different variances, or other nonuniform choices. Since variable shape parameters have a long history in classical RBF methods and often improve accuracy over constant-scale formulations [54–56], this direction appears especially promising. Finally, more PDEs should be tested in order to assess how far the present scale-selection rule extends in physics-informed settings.

References

[1] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, "KAN: Kolmogorov-Arnold networks," *arXiv preprint* arXiv:2404.19756, 2024.

[2] Z. Liu, P. Ma, Y. Wang, W. Matusik, and M. Tegmark, "KAN 2.0: Kolmogorov-Arnold networks meet science," *arXiv preprint* arXiv:2408.10205, 2024. [Online]. Available: https://github.com/KindXiaoming/pykan

[3] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686–707, 2019.

[4] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nat. Rev. Phys.*, vol. 3, no. 6, pp. 422–440, 2021.

[5] S. Sidharth, K. A. R., and A. K. P., "Chebyshev polynomial-based Kolmogorov-Arnold networks: An efficient architecture for nonlinear function approximation," *arXiv preprint* arXiv:2405.07200, 2024.

[6] N. A. Daryakenari, K. Shukla, and G. E. Karniadakis, "Representation meets optimization: Training PINNs and PIKANs for gray-box discovery in systems pharmacology," *arXiv preprint* arXiv:2504.07379, 2025.

[7] J. D. Toscano, L.-L. Wang, and G. E. Karniadakis, "KKANs: Kurkova-Kolmogorov-Arnold networks and their learning dynamics," *Neural Netw.*, vol. 191, p. 107831, 2025.

[8] A. A. Aghaei, "fKAN: Fractional Kolmogorov-Arnold networks with trainable Jacobi basis functions," *arXiv preprint* arXiv:2406.07456, 2024. [Online]. Available: https://github.com/alirezaafzalaghaei/fKAN

[9] A. Kashefi and T. Mukerji, "Kolmogorov–Arnold PointNet: Deep learning for prediction of fluid fields on irregular geometries," *arXiv preprint* arXiv:2504.06327, 2025. [Online]. Available: https://github.com/Ali-Stanford/Physics_Informed_KAN_PointNet

[10] J. Xu, Z. Chen, J. Li, S. Yang, W. Wang, X. Hu, and E. C. H. Ngai, "FourierKAN-GCF: Fourier Kolmogorov-Arnold network—An effective and efficient feature transformation for graph collaborative filtering," *arXiv preprint* arXiv:2406.01034, 2024. [Online]. Available: https://github.com/Jinfeng-Xu/FKAN-GCF

[11] J. Zhang, Y. Fan, K. Cai, and K. Wang, "Kolmogorov-Arnold Fourier networks," *arXiv preprint* arXiv:2502.06018, 2025. [Online]. Available: https://github.com/kolmogorovArnoldFourierNetwork/KAF

[12] C. C. So and S. P. Yung, "Higher-order ReLU-KANs (HRKANs) for solving physics-informed neural networks (PINNs) more accurately, robustly, and faster," *arXiv preprint* arXiv:2409.14248, 2024.

[13] Q. Qiu, T. Zhu, H. Gong, L. Chen, and H. Ning, "ReLU-KAN: New Kolmogorov-Arnold networks that only need matrix addition, dot multiplication, and ReLU," *arXiv preprint* arXiv:2406.02075, 2024. [Online]. Available: https://github.com/quiqi/relu_kan

[14] S. Rigas, M. Papachristou, T. Papadopoulos, F. Anagnostopoulos, and G. Alexandridis, "Adaptive training of grid-dependent physics-informed Kolmogorov-Arnold networks," *IEEE Access*, vol. 12, pp. 176982–176998, 2024. [Online]. Available: https://github.com/srigas/jaxKAN

[15] Z. Bozorgasl and H. Chen, "Wav-KAN: Wavelet Kolmogorov-Arnold networks," *arXiv preprint* arXiv:2405.12832, 2024. [Online]. Available: https://github.com/zavareh1/Wav-KAN

[16] A. A. Howard, B. Jacob, S. H. Murphy, A. Heinlein, and P. Stinis, "Finite basis Kolmogorov-Arnold networks: Domain decomposition for data-driven and physics-informed problems," *arXiv preprint* arXiv:2406.19662, 2024. [Online]. Available: https://github.com/pnnl/neuromancer/tree/feature/fbkans/examples/KANs

[17] S. T. Seydi, "Exploring the potential of polynomial basis functions in Kolmogorov-Arnold networks: A comparative study of different groups of polynomials," *arXiv preprint* arXiv:2406.02583, 2024.

[18] J. A. Actor, G. Harper, B. Southworth, and E. C. Cyr, "Leveraging KANs for expedient training of multichannel MLPs via preconditioning and geometric refinement," *arXiv preprint* arXiv:2505.18131, 2025.

[19] S. Rigas, D. Verma, G. Alexandridis, and Y. Wang, "Initialization schemes for Kolmogorov-Arnold networks: An empirical study," *arXiv preprint* arXiv:2509.03417, 2025. https://github.com/srigas/KAN_Initialization_Schemes

[20] A. A. Howard, N. Zolman, B. Jacob, S. L. Brunton, and P. Stinis, "SINDy-KANs: Sparse identification of nonlinear dynamics through Kolmogorov-Arnold networks," 2026.

[21] R. L. Hardy, Multiquadric equations of topography and other irregular surfaces, Journal of Geophysical Research (1896-1977), 76(8), 1905–1915, 1971.

[22] E. Kansa, Multiquadrics scattered data approximation scheme with applications to computational fluid-dynamics solutions to parabolic, hyperbolic and elliptic partial differential equations, Computers and Mathematics with Applications, 19(8), 147–161, 1990.

[23] Z. Li, "Kolmogorov-Arnold networks are radial basis function networks," *arXiv preprint* arXiv:2405.06721, 2024. [Online]. Available: https://github.com/ZiyaoLi/fast-kan

[24] B. C. Koenig, S. Kim, and S. Deng, "LeanKAN: A parameter-lean Kolmogorov-Arnold network layer with improved memory efficiency and convergence behavior," *arXiv preprint* arXiv:2502.17844, 2025. [Online]. Available: https://github.com/DENG-MIT/LeanKAN

[25] D. W. Abueidda, P. Pantidis, and M. E. Mobasher, "DeepOKAN: Deep operator network based on Kolmogorov-Arnold networks for mechanics problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 436, p. 117699, 2025. GitHub: https://github.com/DiabAbu/Dee

[26] S. T. Chiu, S. W. Cheung, U. Braga-Neto, C. S. Lee, and R. P. Li, "Free-RBF-KAN: Kolmogorov-Arnold Networks with Adaptive Radial Basis Functions for Efficient Function Learning," *arXiv preprint* arXiv:2601.07760, 2026.

[27] A. Delis, "FasterKAN," GitHub repository, 2024. [Online]. Available: https://github.com/AthanasiosDelis/faster-kan

[28] E. Larsson and R. Schaback, "Scaling of radial basis functions," *IMA Journal of Numerical Analysis*, vol. 44, no. 2, pp. 1130–1152, 2024.

[29] R. Cavoretto and A. De Rossi, "An adaptive refinement scheme for radial basis function collocation," in *Proc. Int. Conf. Numer. Comput.: Theory Algorithms*, pp. 19–26, 2019.

[30] R. Cavoretto, S. Lancellotti, and F. Romaniello, "A Bayesian approach for simultaneously radial kernel parameter tuning in the partition of unity method," in *Proc. Int. Conf. Numer. Comput.: Theory Algorithms*, pp. 215–222, 2023.

[31] T. Wenzel and G. Santin, "On the optimal shape parameter for kernel methods: Sharp direct and inverse statements," *arXiv preprint* arXiv:2601.14070, 2026.

[32] T. Wenzel and A. Iske, "Spectral alignment of kernel matrices and applications," *SIAM Journal on Matrix Analysis and Applications*, vol. 47, no. 1, pp. 265–281, 2026.

[33] R. Schaback, Error estimates and condition numbers for radial basis function interpolation. Advances in Computational Mathematics, 3(3): 251–264, 1995.

[34] R. Schaback, "Small errors imply large evaluation instabilities," *Advances in Computational Mathematics*, vol. 49, no. 2, p. 25, 2023.

[35] M. Wolff, F. Eilers, and X. Jiang, "CVKAN: Complex-valued Kolmogorov-Arnold networks," *arXiv preprint* arXiv:2502.02417, 2025. [Online]. Available: https://github.com/M-Wolff/CVKAN

[36] R. Che, L. af Klinteberg, and M. Aryapoor, "Improved Complex-Valued Kolmogorov–Arnold Networks with Theoretical Support," in *Proc. 24th EPIA Conf. on Artificial Intelligence (EPIA)*, Faro, Portugal, Oct. 2025, Part I, pp. 439–451. Springer, Heidelberg.

[37] Z. Chao, X. Liu, Z. Wu, and X. Li, "RBF-KAN: Radial Basis Function-Kolmogorov-Arnold Networks," *IEEE Internet Things J.*, 2026.

[38] A. Farea, and M.S. Celebi, Learnable activation functions in physics-informed neural networks for solving partial differential equations. Computer Physics Communications, 2025. 315: p. 109753.

[39] Z. Wen, Q. Zhang, J. Chen, *et al.*, "Computing-in-memory architecture for Kolmogorov-Arnold networks based on tunable Gaussian-like memory cells," *Nat. Commun.*, 2026.

[40] G.E. Fasshauer, "Meshfree approximation methods with Matlab" (Vol. 6). World Scientific Publishing Company, 2007.

[41] J. Li, A. H. D. Cheng, and C. S. Chen, "A comparison of efficiency and error convergence of multiquadric collocation method and finite element method," *Eng. Anal. Bound. Elem.*, vol. 27, no. 3, pp. 251–257, 2003.

[42] A. Noorizadegan, C.-S. Chen, D. L. Young, and C. S. Chen, "Effective condition number for the selection of the RBF shape parameter with the fictitious point method," *Applied Numerical Mathematics*, vol. 178, pp. 280–295, 2022.

[43] C.-S. Chen, A. Noorizadegan, D. L. Young, and C. S. Chen, "On the selection of a better radial basis function and its shape parameter in interpolation problems," *Applied Mathematics and Computation*, vol. 442, p. 127713, 2023.

[44] A. Noorizadegan, A. Naji, T. L. Lee, R. Cavoretto, and D. L. Young, "Bending analysis of quasicrystal plates using adaptive radial basis function method," *J. Comput. Appl. Math.*, vol. 450, p. 115990, 2024.

[45] A. Noorizadegan, and R. Schaback, "Introducing the evaluation condition number: A novel assessment of conditioning in radial basis function methods. Engineering Analysis with Boundary Elements," 166, p.105827, 2024.

[46] E. Larsson and B. Fornberg, "Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions," *Comput. Math. Appl.*, vol. 49, no. 1, pp. 103–130, 2005.

[47] G.E. Fasshauer, and M.J. McCourt, "Stable evaluation of Gaussian radial basis function interpolants," SIAM Journal on Scientific Computing, 34(2), pp.A737-A762, 2012.

[48] D. Hilbert, "Mathematical problems," *Bull. Amer. Math. Soc.*, vol. 8, pp. 437–479, 1902.

[49] A. N. Kolmogorov, "On the representation of continuous functions of several variables as superpositions of continuous functions of a smaller number of variables," *Dokl. Akad. Nauk SSSR*, vol. 108, no. 2, pp. 179–182, 1956. (In Russian.)

[50] V. I. Arnol'd, "On functions of three variables," *Dokl. Akad. Nauk SSSR*, vol. 114, pp. 679–681, 1957. (In Russian.)

[51] A. N. Kolmogorov, "On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition," *Doklady Akademii Nauk*, vol. 114, pp. 953–956, 1957.

[52] A. Noorizadegan, S. Wang, , L. Ling, and J.P. Dominguez-Morales, A Practitioner's Guide to Kolmogorov-Arnold Networks. arXiv preprint arXiv:2510.25781, 2025.

[53] T. Yu, J. Qiu, J. Yang, and I. Oseledets, "Sinc Kolmogorov-Arnold network and its applications on physics-informed neural networks," *arXiv preprint* arXiv:2410.04096, 2024. [Online]. Available: https://github.com/DUCH714/SincKAN

[54] J. Wertz, E. J. Kansa, and L. Ling, "The role of the multiquadric shape parameters in solving elliptic partial differential equations," *Computers & Mathematics with Applications*, vol. 51, no. 8, pp. 1335–1348, 2006.

[55] S. N. Chiu, L. Ling, and M. McCourt, "On variable and random shape Gaussian interpolations," *Applied Mathematics and Computation*, vol. 377, p. 125159, 2020.

[56] G. E. Fasshauer and J. G. Zhang, "On choosing "optimal" shape parameters for RBF approximation," *Numer. Algorithms*, vol. 45, no. 1, pp. 345–368, 2007.

[57] H. Wendland, *Scattered Data Approximation*, Cambridge, U.K.: Cambridge Univ. Press, 2005.

[58] M. S. Floater, and A. Iske, " Multistep scattered data interpolation using compactly supported radial basis functions," Journal of Computational and Applied Mathematics, 73(1-2), 65-78, 1996.

[59] S. Müller and R. Schaback, "A Newton basis for kernel spaces," *J. Approx. Theory*, vol. 161, no. 2, pp. 645–655, 2009.