Kolmogorov Arnold Neural Networks

Justus Prass

July 7, 2025

Abstract

Today's best-performing neural networks typically rely on global function representations. Building upon the recent findings of Liu et al. [LWV+24], we propose a new, simple architecture called Kolmogorov Arnold neural networks (KANNs), which leverage local function representations. Experiments on synthetic regression tasks demonstrate that KANNs outperform deep neural networks (DNNs) and the Kolmogorov Arnold networks (KANs) proposed in [LWV+24], especially in scenarios with noise and irrelevant features. For example, on the Friedman regression dataset [Fri91] with additive Gaussian noise ($\sigma^2 = 1$) and 95 irrelevant features, our approach achieved a test RMSE of $7.6 \cdot 10^{-2}$, compared to $6.0 \cdot 10^{-1}$ for DNNs and 2.7 for KANs.

1 Introduction

In recent years, the field of machine learning has undergone rapid advancement, propelled by the increasing availability of large datasets, improved computational resources, and significant algorithmic innovations. Among the most transformative developments in this domain is the emergence of DNNs, which have proven to be highly effective in modeling complex, nonlinear relationships across a wide range of tasks. These networks have become foundational in numerous applications due to their flexibility and expressive power. However, despite their widespread adoption, DNNs exhibit several significant limitations. Among the most notable are their tendency to overfit, particularly when trained on limited or noisy data [SL19], and the opacity of their internal decision-making processes, which hinders interpretability [ZTLT21]. Furthermore, these models frequently necessitate substantial amounts of labeled data to function optimally, a prerequisite that is not invariably feasible in practical real-world scenarios [Mar18]. Their training and deployment are computationally intensive, demanding substantial hardware, energy, and time resources [TGLM22]. Finally, DNNs frequently struggle to generalize to out-of-distribution data, limiting their adaptability in dynamic environments [HSW+21]. These limitations have prompted the proposal of an alternative architecture, called KANs, by Liu et al. [LWV+24]. In their study, Liu et al. hypothesized that this alternative architecture could mitigate some of the existing limitations. Inspired by the aforementioned KANs, this paper offers a novel perspective by investigating the learning and representation of a specific set of functions that can be decomposed into compositions of "simpler" functions.

The primary objective of this paper is twofold. First, it seeks to build upon the theoretical foundations established in previous research, such as [LWV+24], to gain a deeper understanding of the existing methodologies. Second, it aims to leverage these insights to propose a novel approach in the field of DNNs.

The discussion unfolds in several steps:

In the preliminary Section 2 central tenets of the paper are unveiled. This segment encompasses the delineation of the set of functions to be approximated. Section 3 deals with the substantiation of pivotal theoretical outcomes, concerning the previously defined set of functions.

The ensuing Section 4 builds upon these foundational results by introducing a novel methodology in the field of DNNs. This methodology is then subjected to a thorough analysis and evaluation. The proposed approach is benchmarked against existing methods, such as KANs and DNNs, through extensive empirical evaluations. These evaluations extend beyond the scope of those conducted in [LWV+24], as they are situated within the context of engineering applications, incorporating noise and irrelevant features. This expanded scope offers valuable insights, particularly concerning robustness and generalization.

Finally, Section 5 offers a concise summary of the primary findings and discusses avenue for future research.

2 Definition of the Set of Functions $\mathcal{F}_{X,I,n}$

This section delineates the set of functions, called multivariate functions represented as compositions of superpositions of univariate functions, which we aim to approximate. In order to define a multivariate function $f : I_1^{n_1} \to I_N^{n_N}$ represented as a composition of superpositions of univariate functions, it is first necessary to define the univariate functions that serve as the building blocks. Subsequent to this, superpositions are constructed, aggregating their contributions to yield the desired composition. For the univariate functions consider:

$$\begin{gathered}\Psi_{l,k,j} : I_l \to \mathbb{R}, \\ I_l \subseteq I, \quad l \in \{1, \dots, N-1\}, \quad k \in \{1, \dots, n_{l+1}\}, \quad j \in \{1, \dots, n_l\}, \\ N = |n|,\end{gathered} \tag{2.1}$$

where $n = (n_1, \dots, n_N)$ is a tuple with $N \geq 2$ and $n_i \in \mathbb{N}, i = 1, \dots, N$. I is a compact set. As previously mentioned, the objective is to apply compositions to these functions, which necessitates a constraint on the co-domain of the functions $\Psi_{l,k,j}$. To elaborate, the constraint stipulates the existence of sets $I_1, I_2, \dots I_{N-1} \subseteq I$ such that

$$\Psi_{l,k,1}(I_l) + \Psi_{l,k,2}(I_l) + \cdots + \Psi_{l,k,n_l}(I_l) \subseteq I_{l+1} \tag{2.2}$$

$$l = 1 \dots N-1, \;\; j = 1, \dots n_l$$

holds. The application of the superposition principle yields the following expression for the function Φ_l^k:

$$\Phi_l^k : I_l^{n_l} \to I_{l+1} : x \mapsto \sum_{j=1}^{n_l} \Psi_{l,k,j}(x_j), \tag{2.3}$$

which, from the previous context, makes the constraint in Eq. (2.2) evident. Subsequently, the functions Φ_l^k are arranged in a layered configuration to establish the mapping Φ_l. The formal expression is as follows:

$$\Phi_l : I_l^{n_l} \to I_{l+1}^{n_{l+1}} : x \mapsto (\Phi_l^1(x), \dots, \Phi_l^{n_{l+1}}(x)). \tag{2.4}$$

Sometimes we refer to the functions Φ_i as "inner" functions. It is finally posited that a function $f : I^{n_1} \to I^{n_N}$ is a multivariate function represented as a composition of superpositions of univariate functions, if it can be written as

$$f = \bigcirc_{l=1}^{N-1} \Phi_l, \tag{2.5}$$

for some tuple n and some compact interval I. It is important to acknowledge that I_N is not necessarily a subset of I (see Eq. (2.2)). Rather, I_N is an arbitrary set because we can refrain from enforcing any conditions on I_N, as Φ_{N-1}^k is not applied to a subsequent function. It is important to note that the compactness of I_N is contingent upon the continuity of each function $\Psi_{l,k,j}$, as evidenced by the compactness of I and the observation that the image of a continuous function, when considered over a compact set, is itself compact. As well as the fact that the sum of compact sets in a finite-dimensional space results in another compact set. The set of all such functions is denoted by

$$\mathcal{F}_{I,(n_1,\dots,n_N)} := \{f : I_1^{n_1} \to I_N^{n_N} | f(x) = (\bigcirc_{l=1}^{N-1} \Phi_l)(x), \text{ as per Eq. ((2.1) - (2.5))}\}. \tag{2.6}$$

If the tuple n is understood, we write $\mathcal{F}_{I,n} := \mathcal{F}_{I,(n_1,\dots,n_N)}$. In certain circumstances, it can be advantageous to delineate the characteristics of functions in terms of the number of "inner" functions, denoted by $N-1$ (cf. Eq. (2.5)). We write $f_{N-1} \in \mathcal{F}_{I,n}$ if $|n| = N$. More specifically for a function $f \in \mathcal{F}_{I,n}$, f_j denotes the composition of the first j inner functions. That is to say:

$$f = \Phi_{N-1} \circ \Phi_{N-2} \circ \cdots \circ \underbrace{\Phi_j \circ \cdots \circ \Phi_1}_{f_j}. \tag{2.7}$$

Figure 1 provides a graphical representation of a function $f_3 \in \mathcal{F}_{I,(2,3,3,1)}$.

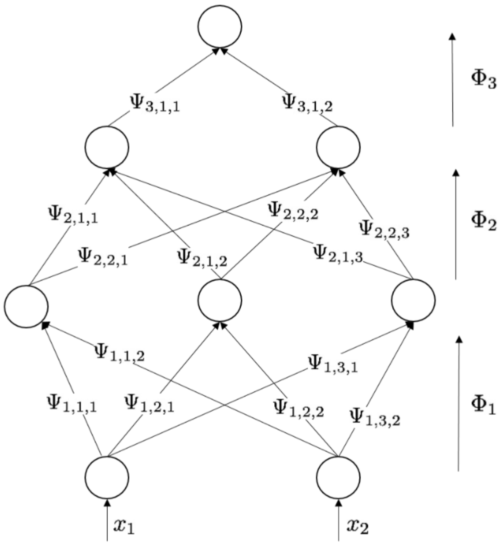


Figure 1: Graphical representation of a function $f \in \mathcal{F}_{I,(2,3,2,1)}$. Each arrow represents a function $\Psi_{l,k,j}$ applied to an input, which is depicted by a circle. The graph is read from bottom to top, where multiple lines joining at a circle in the upward direction indicate the application of a summation operation.

The following are some examples of functions $f \in \mathcal{F}_{I,n}$.

1. Obviously, all univariate functions over a compact domain I are in $\mathcal{F}_{I,(1,1)}$
2. By [Buc79, Theorem 6] we have that any function defined over a compact n-cell I^n is in $\mathcal{F}_{I,(n,1,1)}$, that is
$$\mathcal{F}[I^n] = \mathcal{F}_{I,(n,1,1)}[I^n]. \tag{2.8}$$
The set $\mathcal{F}_{I,(n,1,1)}$ is referred to as the set of nomographic functions [Buc79] . The set $\mathcal{F}[I^n]$ denotes the set of all functions over I^n.

It is imperative to recognize that the objective of this study is to approximate the function $f \in \mathcal{F}_{I,n}$ by approximating each univariate function $\Psi_{l,k,j}$. The quality of this approximation is significantly influenced by the following factors:

1. The properties of the individual functions $\Psi_{l,k,j}$, as for example the functions $\Psi_{l,k,j}$ in Eq. (2.8) may even be discontinuous and therefore unfeasible for approximation. To streamline the ensuing discussion, we introduce the following notation that highlights the key properties of the individual univariate functions $\Psi_{l,k,j}$:

$$\mathcal{F}_{X,I,n} = \{f : I_1^{n_1} \to I_N^{n_N} | f(x) = (\bigcirc_{l=1}^{N-1} \Phi_l)(x), \text{ as per Eq. } (2.1 - 2.5), \Psi_{l,k,j} \in X\}, \tag{2.9}$$

   where X denotes a predefined set of functions, such as continuous functions, Lipschitz continuous functions, or smooth functions. It is posited that a set of functions, denoted by K, admits a representation of type X if, for some tuple n and a compact interval I, we can write $K \subseteq \mathcal{F}_{X,I,n}$.

2. The method employed for approximating univariate functions, as some approximation techniques can yield substantially different results depending on the properties of the function $\Psi_{l,k,j}$ being approximated. To streamline the ensuing discourse, let $A : X \to Y$ be an approximation operator for univariate functions, as defined in Section A.1. We subsequently proceed to define an approximation operator for functions $f \in \mathcal{F}_{X,I,n}$, as

$$\mathbf{A} : \mathcal{F}_{X,I,n} \to \mathcal{F}_{Y,I,n}, \tag{2.10}$$

   where $\mathbf{A}$ is the composition- and component-wise application of the operator A. For functions $f \in \mathcal{F}_{X,I,n}$, this is tantamount to applying the operator A to each $\Psi_{l,k,j}$ individually. We say the operator A induces $\mathbf{A}$.

It is finally possible to quantify the quality of approximation of a function $f \in \mathcal{F}_{C[I],I,n}$ by $\|f - \mathbf{A}f\|_{\mathcal{F}_{C[I],I,n},\infty}$, where we have defined the normed linear space:

$$\begin{gathered}(C[I_1^{n_1}, I_N^{n_N}], \|\cdot\|_{\mathcal{F}_{C[I],I,n},\infty}),\\ \text{where } \|f\|_{\mathcal{F}_{C[I],I,n},\infty} = \sup_{x \in I_1^{n_1}} \|f(x)\|_{I_N^{n_N},\infty}.\end{gathered} \tag{2.11}$$

The fact that $\|f\|_{\mathcal{F}_{C[I],I,n},\infty}$ indeed defines a norm, as per the defining axioms, is discussed in Appendix A.2.1. The subsequent discussion will focus on several examples of functions in the set $\mathcal{F}_{X,I,n}$.

1. Every function $f \in C[I^n]$ belongs to the set $\mathcal{F}_{C[I],I,(n,2n+1,1)}$, which follows directly from the Kolmogorov-Arnold theorem [Kol57]. This theorem posits that any continuous function can be expressed as a superposition of univariate continuous functions. Specifically, for any continuous function $f : [0,1]^n \to \mathbb{R}$, there exist univariate continuous functions $\phi_{q,p} : [0,1] \to \mathbb{R}$ and $\Phi_q : \mathbb{R} \to \mathbb{R}$ such that the relation

$$f(\mathbf{x}) = f(x_1, \ldots, x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right) \tag{2.12}$$

   holds. In fact each of inner functions $(\phi_{q,p})$ are independent of f. Since $\phi_{q,p}$ is a continuous function defined on a compact domain for each q and for all p, its image remains compact. Consequently, Φ_q, which is applied to the sum of such functions, also has a compact domain. This is because, in finite-dimensional spaces, the sum of compact sets is compact. However, the Kolmogorov–Arnold theorem does not guarantee any desirable properties—such as smoothness—for the resulting univariate functions. In fact, these univariate functions often exhibit highly irregular behavior and may even have fractal-like properties.

In the following examples, we provide both an analytical as well as a polynomial representation of multivariate polynomials, utilizing the developed framework.

2. Any multivariate polynomial $p \in P_d[a,b]^m$ where $a > 0$ is an element of $\mathcal{F}_{C^\infty[I],I,(m,\binom{d+m}{m}-1,1)}$. This can be easily verified by directly defining the mapping Φ_1 (cf. Eq. (2.4)), as

$$\Phi_1 : x \mapsto \{\sum_{j=1}^{m_1} a_{1,j}\log(x_j), \sum_{j=1}^{m_2} a_{2,j}\log(x_j), \dots, \sum_{j=1}^{m_n} a_{\binom{d+m}{m},j}\log(x_j)\}$$

$$\text{such that for all } i = 1, \dots, \binom{d+m}{m} - 1 \ \sum_{j=1}^{m_i} a_{i,j} \leq d, a_{i,j} \in \mathbb{N},$$

and the mapping Φ_2, as

$$\Phi_2 : x \mapsto \sum_j \gamma_j \exp x_j,$$

where each sum in the mapping Φ_1 represents a monomial term. This follows directly from the identity

$$\prod_{i=1}^{n} x_i^{\alpha_i} = \exp\left(\sum_{i=1}^{n} \alpha_i \log(x_i)\right).$$

The mapping Φ_2, is then employed to apply the corresponding coefficients γ_j. We note that the constant monomial has been neglected; however, by adding its coefficient to any of the univariate functions in Φ_2, it can be easily incorporated. To enumerate the monomials in a polynomial of degree d, we examine the following equation:

$$a_1 + a_2 + \dots + a_n = k,$$

which has exactly $\binom{k+n-1}{n-1}$ integer solutions for the a_i's. By summing over all possible values of k, we obtain the following identity:

$$\sum_{k=0}^{d} \binom{k+n-1}{n-1} = \binom{d+n}{n},$$

which is a well-known identity in combinatorics. Thus providing the total number of monomials in a polynomial of degree d in n variables.

3. Before presenting the subsequent example, it is necessary to state a technical lemma.

Lemma 2.1. *There exist exactly $\binom{d-1}{w}$ tuples $\{x_1, x_2, \dots, x_w\} \subset \mathbb{N}$ of cardinality w, such that $\sum_{i=1}^{w} x_i < d$.*

Proof. We refer the reader to Appendix A.2.4. □

Any multivariate polynomial $p \in P_d[I^n]$ belongs to the space $\mathcal{F}_{P_d[I],I,(n,\sum_{w=1}^{d} 2^w \binom{n}{w}\binom{d}{w},1)}$. This can be verified as follows. Initially, the subsequent crucial identity is presented:

$$C\prod_{i=1}^{n} x_i = \sum_{S \subseteq \{x_1,\dots,x_n\}} (-1)^{n-|S|} \left(\sum_{x_i \in S} x_i\right)^n, \tag{2.13}$$

for a proof, the reader is referred to Appendix A.2.2. We proceed to define the set S as the collection of all univariate monomials with coefficient 1, formed from the variables $\{x_1, \dots, x_n\}$, up to degree d, excluding the constant term; that is,

$$S = \{x_1^1, x_1^2, \dots, x_1^d, \dots, x_n^1, x_n^2, \dots, x_n^d\}.$$

In the subsequent step, we identify suitable subsets of S according to Eq. (2.13). These subsets are selected such that the total sum of their respective elements' exponents is less than or equal to d. This set is denoted by $\mathcal{T}_d$. Formally, we obtain

$$\mathcal{T}_d(S) = \left\{ \mathcal{P}(S') \mid S' \subseteq S, \sum_{y_i \in S'} \text{exponent}(y_i) \leq d \right\},$$

with *exponent* : $\mathbb{R} \to \mathbb{N} : x_i^k \mapsto k$. For the sake of clarity, it should be noted that each y_i in the aforementioned equation represents an element $x_i^k \in S$. We note that $|\mathcal{T}_d(S)| \leq \sum_{w=1}^{d} \binom{n}{w}\binom{d}{w}$, where w can be conceptualized as the quantity of elements contained within the appropriate subsets $S' \subset S$. $\binom{d}{w}$ represents the number of possible exponent combinations for the w-variables that satisfy the degree condition, which directly follows from Lemma 2.1. By $\mathcal{T}_d^i(S)$ we denote the i-th element in $\mathcal{T}_d(S)$ and $\mathcal{P}$ denotes the power set. Now by defining the tuple $T_{d,flat}(S) := \bigcup_i \mathcal{T}_d^i(S)$ we can define, using Eq. (2.13), the mapping Φ_1 as in Eq. (2.4), as

$$\Phi_1 : x \mapsto \left\{\Phi_1^1(x), \ldots \Phi_1^N(x)\right\},$$

$$\text{with } \Phi_1^i : \{x_1, \ldots, x_n\} \mapsto \left\{ \sum_{y \in T_{d,flat}^i(S)} y \right\},$$

where $N = |\mathcal{T}_{d,flat}(S)| \leq \sum_{w=1}^{d} 2^w \binom{n}{w}\binom{d}{w}$. Finally, we define the second mapping Φ_2 as

$$\Phi_2 : \{x_1, \ldots, x_n\} \mapsto \sum_k C_k' x_k^{d'}, \quad d' \leq d.$$

The reconstructed polynomial has coefficients $\alpha_i = C_k' C$. Note that C is the same for all terms reconstructing a monomial, as indicated in Eq. (2.13). Here, d' denotes the number of terms in the corresponding multivariate monomial (n in Eq. (2.13)).

Remark 2.1. *The latter example is particularly noteworthy because it generalizes the former by representing polynomials over arbitrary compact intervals. Moreover, it utilizes univariate polynomials, which occupy a distinctive position in approximation theory. These polynomials are known for their ease of approximation and represent one of the simplest classes of functions used for approximating more complex functions. Finally, the famous Stone-Weierstrass theorem (see for example [Wer18, Theorem VIII.4.7, pg. 455]) ensures the validity of the following statement:* $\bigcup_{d=1}^{\infty} P_d[I^n]$ *is dense in* $C[I^n]$*, and consequently,*

$$\bigcup_{d=1}^{\infty} \mathcal{F}_{P_d, I, (n, \sum_{w=1}^{d} 2^w \binom{n}{w}\binom{d}{w}, 1)},$$

is dense in $C[I^n]$*, offering a universal approximation property, analogous to that of neural networks. An extension to deeper representation is straightforward. This extension is achieved by copying the inputs to subsequent layers via the identity function. The identity function* $I : x \mapsto x$ *is a polynomial. This representation is particularly advantageous compared to the one given in Example 1 in that it expresses functions through "simple" univariate functions rather than the highly intricate form presented in Example 1.*

Hence, for any function $f \in C[I^n]$, there exists a representation in $\mathcal{F}_{C[I],I,(n,m,1)}$ with $m \geq 2n + 1$ (cf. Example 1). Increasing m has a smoothing effect on the univariate functions, as they converge to polynomials as m approaches infinity (cf. Example 3).

3 Approximation of Functions in $\mathcal{F}_{Lip_L,I,n}$

Pursuant to the definitions established in the preceding Section 2, we are now prepared to proceed with the articulation of the ensuing theorem.

Theorem 3.1 (Approximation Theorem)**.** *For any $f \in \mathcal{F}_{Lip_L,I,n}$ and any approximation operator A on $Lip_L[I]$, we have the inequality,*

$$\|f - \mathbf{A}f\|_{\mathcal{F}_{C[I],I,n},\infty} \leq C_{L,n}\,\mathrm{dist}(Lip_L, A),$$

where $C_{L,n}$ is a constant depending on the Lipschitz-constant L of function $\Psi_{l,k,j} \in Lip_L[I] \subset C[I]$ and on the tuple n. The distance measure dist *is defined in Eq.* (A.1.23) *and is calculated with respect to the metric induced by $\|\cdot\|_\infty$. The operator $\mathbf{A}$ is induced by A, cf. Eq.* (2.10)*. For a Proof please see Appendix A.3.*

Theorem 3.1 essentially states that if each of the functions $\Psi_{l,k,j}$ is Lipschitz continuous, then the approximation error, resulting from approximating each $\Psi_{l,k,j}$ individually, increases linearly with $\mathrm{dist}(Lip_L[I], A)$. A slightly converse statement is made in the following remark.

Remark 3.1. *For any approximation operator $\mathbf{A}$ (cf. Eq.* (2.10)*) induced by a operator of the form $A_L : C[I] \to \mathrm{Lip}_L[I]$, the statements of Theorem 3.1 remain valid for functions $f \in \mathcal{F}_{C[I],I,n}$, except that the constant $C_{L,n}$ now depends on the Lipschitz constant L of the approximation operator A_L. For a proof of the remark we refer the reader to Appendix A.2.3*

The statement of Remark 3.1 may appear promising; however, it should be noted that, in general, the Lipschitz constant of an approximating function increases with the complexity of the target function. We now proceed to provide an example that aims to place the work of Liu et al. [LWV$^+$24] within the methodology and definitions developed in this paper.

Example 3.1. *In this example, the focus will be on the approximation of a function.*

$$f \in \mathcal{F}_{C_b^{d+1}[I],I,n}{}^{1} \tag{3.14}$$

by the operator

$$\mathbf{A_{t,d}} : \mathcal{F}_{C_b^d[I],I,n} \to \mathcal{F}_{S_{d,t},I,n}, \tag{3.15}$$

which is induced by the operator

$$A_{t,d} : C^d[I] \to S_{d,t}. \tag{3.16}$$

The latter operator utilizes splines of degree at most d and over a grid t (see for example [dB01]) to approximate smooth functions. The distance measure $\mathrm{dist}(C^d, A_{t,d})$ (cf. Eq. (A.1.23)*), with respect to the uniform norm, is upper bounded in Theorem A.3 for $r = 0$. In accordance with the established result in Theorem 3.1, the subsequent inequality is derived:*

$$\|f - \mathbf{A_{t,d}}f\|_{\mathcal{F}_{C[I],I,n}} \leq K_{L,n,L_d,C_0}(\Delta_{max}t)^{d+1}.$$

Here, K_{L,n,L_d,C_0} is a constant that depends on L, defined as $L = \sup_{l,j,k}\|D\Psi_{l,j,k}\|_{C[I],\infty}$, on the tuple n, on the constant C_0 given in Theorem A.3, and on L_d, defined as $L_d = \sup_{l,j,k}\|D^{d+1}\Psi_{l,j,k}\|_{C[I],\infty}$. The constant K_{L,n,L_d,C_0} can be bounded, following the steps of the proof of Theorem 3.1, by the inequality

$$K_{L,n,L_d,C_0} \leq \sum_{l=1}^{|n|-1}\prod_{k=l}^{|n|-1} n_k L L_d C_0 \leq \left(\prod_{i=1}^{|n|-1} n_i\right)(LL_dC_0)^{|n|-1}(|n|-1).$$

[1] $\mathcal{C}_b^d[U] = \{f \in \mathcal{C}^d[U] \mid \sup_{x\in U}|D^i f(x)| \leq M$ for some $M \geq 0$ and $i = 1, 2, \ldots d\} \subset Lip_L[I]$

The statements in Example 3.1 delineate the process of approximating a function, $f \in \mathcal{F}_{C_b^d[I],I,n}$ with a KAN, as defined in [LWV+24]. It should be noted that the minor exception exists of considering functions in $\mathcal{F}_{C_b^d[I],I,n}$, whereas the original paper focused on functions $f \in \mathcal{F}_{C^d[I],I,n}$, as stated in [LWV+24, Thm 2.1].

4 Proposed Approach and Evaluation

In this section, the derived results will be applied to develop a new machine learning approach, which will be referred to as KANNs. We will subsequently assess its performance on common regression problems to evaluate its effectiveness and compare it with existing methods. To streamline the ensuing discussion of model evaluation, we first provide a concise overview of the training process. Training refers to the process of learning a representation from a finite dataset. Formally, a dataset D is defined as a finite collection of input-output pairs:

$$D = \{(x_i, y_i) \mid x_i \in X \subseteq \mathbb{R}^n, y_i \in Y \subseteq \mathbb{R}^m, i = 1, \ldots, N\}.$$

Here, each x_i represents an input vector in $\mathbb{R}^n$, and the corresponding y_i is the associated output in $\mathbb{R}^m$. The objective of training is to ascertain a set of parameters w that minimizes a specified loss function, denoted by $L : Y \times Y \to \mathbb{R}^+$, which quantifies the discrepancy between the model's predictions and the actual outputs. The following is a formal definition of the aforementioned process:

$$\min_w \sum_i L(f_w(x_i), y_i),$$

where $f_w : X \to Y$ represents the model parameterized by the parameters w. In this work, we utilize the root mean squared error (RMSE) as the performance criterion, analogous to the loss function, which is defined as:

$$L_{\text{RMSE}}(f_w, D) = \sqrt{\frac{1}{N} \sum_{(x_i,y_i) \in D} (y_i - f_w(x_i))^2}.$$

We note that L_{RMSE} can be interpreted as a discrete approximation of the metric induced by the L_2-norm. Within the framework of machine learning, a distinction is often drawn between training loss and test loss. The training loss quantifies the model's error on the dataset used for optimization, denoted by D. Conversely, the test loss serves to evaluate the model's performance on previously unseen data points $(x, y) \notin D$, thereby providing an estimate of its generalization ability.

4.1 Kolmogorov Arnold Neural Networks

This section is initiated with a concise review of the previously introduced notation. The operator A is employed to denote an approximation operator that maps functions of a designated type to a typically less complex set. In this section, the operator $A : X \to Y$, is employed for continuous univariate functions, that is $X, Y \subset C[I]$. Subsequently, the operator $\mathbf{A}$ represents the composition- and component-wise application of the operator A to functions in $\mathcal{F}_{X,I,n}$, that is

$$\mathbf{A} : \mathcal{F}_{X,I,n} \to \mathcal{F}_{Y,I,n}.$$

We say that A induces $\mathbf{A}$. A comprehensive investigation of this matter was conducted within Section 2. In the ensuing discourse, an examination will be conducted of the operator $A : C[I] \to \mathcal{D}^{\sigma,t,l}[I]$, where $\mathcal{D}^{\sigma,t,l}$ denotes the space of DNNs of width l and depth

t within the framework of the methodology that has been developed in Section 2. The operator A induces, as per Eq. (2.10), the operator

$$\mathbf{A} : \mathcal{F}_{C[I],I,n} \to \mathcal{F}_{\mathcal{D}^{\sigma,t,l},I,n}. \tag{4.17}$$

The subsequent lemma furnishes an especially compelling property of the set $\mathcal{F}_{\mathcal{D}^{\sigma,t,l},I,n}$.

Lemma 4.1. *The relation:*

$$\mathcal{F}_{\mathcal{D}^{\sigma,t,l},I,n} \subset \mathcal{D}^{\sigma}[\mathbb{R}^{n_1}, \mathbb{R}^{n_N}],$$

where $\mathcal{D}^{\sigma} := \bigcup_{t,l}^{\infty} \mathcal{D}^{\sigma,t,l}$, is valid. For a proof we refer the reader to Appendix A.4.1

The set $\mathcal{F}_{\mathcal{D}^{\sigma},I,n}$ is referred to as the set of KANNs. Figure 2 illustrates the relationship between a KANN and a DNN. Of particular note is the emphasis on the $\subset$ relation between the respective approaches. It is imperative to recall that a DNN is a function of the following form:

$$f : x \mapsto A\sigma\left(W_{t-1} \cdots \sigma\left(W_2\sigma\left(W_1x + b_1\right) + b_2\right) \cdots + b_{t-1}\right).$$

For the specific case illustrated in the figure ($t = 2$), the sparse matrices W_1 and A assume the following form:

$$W_1 = \begin{bmatrix} w_{11} & 0 \\ w_{21} & 0 \\ 0 & w_{32} \\ 0 & w_{42} \\ w_{51} & 0 \\ w_{61} & 0 \\ 0 & w_{72} \\ 0 & w_{82} \end{bmatrix}, \; w_{ij} \in \mathbb{R}.$$

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{25} & a_{26} & a_{27} & a_{28} \end{bmatrix}, \; a_{ij} \in \mathbb{R}.$$

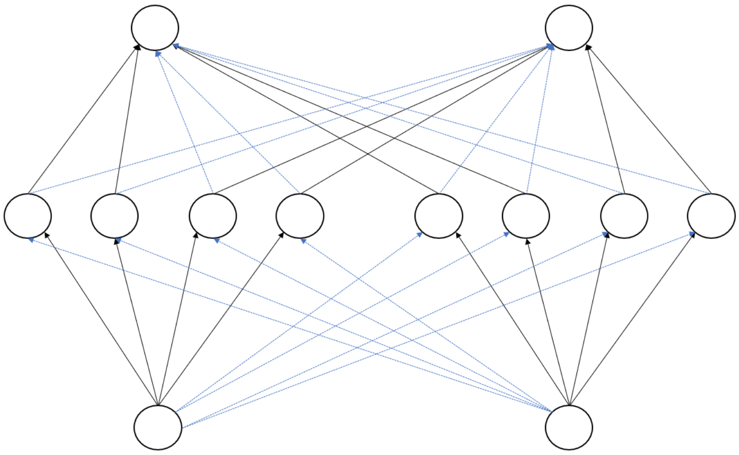


Figure 2: Visualization illustrating the inclusion $\mathcal{F}_{\mathcal{D}^{\sigma,1,2},I,\{2,2\}} \subset \mathcal{D}^{\sigma,8,1}[\mathbb{R}^2, \mathbb{R}^2]$. In this diagram, the circles represent the neurons in the DNN. The black lines represent the weights of the KANN $\mathcal{F}_{\mathcal{D}^{\sigma,1,2},I,\{2,2\}}$, while the combination of black and blue lines illustrates the fully connected DNN $\mathcal{D}^{\sigma,8,1}[\mathbb{R}^2, \mathbb{R}^2]$.

We note that KANNs can be interpreted as compositions of generalized additive models [Pot99, KTWZ24], stacked in a way that balances interpretability with expressive power.

Rather than using additive models in isolation, KANNs layer them sequentially, allowing each stage to transform and build upon the previous one. This compositional design retains the transparency and modularity of classical additive models, while enabling the network to capture rich, nonlinear patterns. Understanding KANNs through this lens helps clarify their structure and sheds light on how they generalize traditional modeling approaches in a principled yet flexible way.

4.2 Evaluation

The code utilized in this section is available at [Pra25].

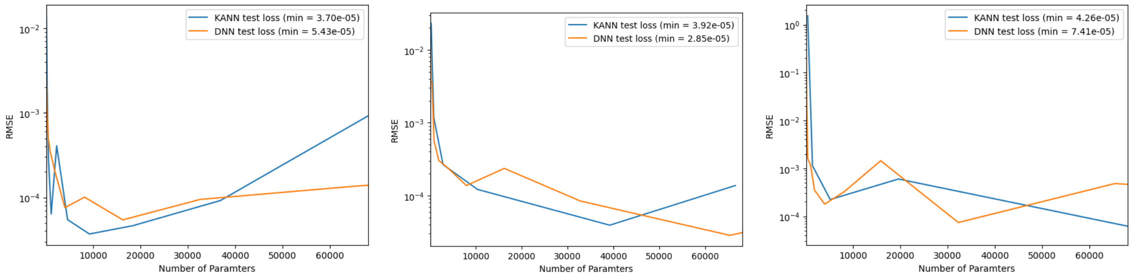
In this section, the objective is to assess the efficacy of operators **A** as delineated in Eq. (2.10). A particular focus of this study is the analysis of the effects of employing the structure[2] of functions $f \in \mathcal{F}_{Lip_L,I,n}$, as defined in Section 2. In order to facilitate our analysis, it is necessary to reduce the effects of the univariate function operator A on the approximation error. The effects in question are encapsulated by the expression $dist(Lip_L, A)$, as elucidated in Theorem 3.1. Alternatively, these effects can be characterized in terms of the Lipschitz constant of functions belonging to the co-domain of A. This notion was further elaborated upon in Remark 3.1. To explore this, we compare the operator **A** induced by $A : C[I] \to \mathcal{D}^{\sigma,l,t}$ with DNN based operators $\mathbf{D} : C[\mathbb{R}^{n_1}, \mathbb{R}^{n_N}] \to \mathcal{D}^{\sigma,t}[\mathbb{R}^{n_1}, \mathbb{R}^{n_N}]$. This comparison is meaningful in terms of the structural properties due to the subset relationship between these operators (see Lemma 4.1). Moreover, the operator **A** furnishes a lucid framework for appraising the strengths and limitations of our approach in relation to standard DNNs, as it sequesters performance from the influence of *different* optimization techniques or training strategies, a distinction that likewise emanates from the subset relationship.

In the subsequent stage of our investigation, we will undertake an empirical evaluation of performance across a range of scenarios. As a preliminary measure, a comparison is made of the parameter efficiency of the two approaches. Initially, we examine the task of learning functions $f \in P_2[I^2]$, a set of functions that is particularly well-suited for this analysis, as it admits the following form:

$$P_2[I^2] \subset \mathcal{F}_{P_2,I,(2,m,1)}, \tag{4.18}$$

as illustrated in Example 3. The coefficients $\{a_i\}$ of the monomials of f, where randomly chosen according to $a_i \sim \mathcal{N}(0,1)$, yet are maintained constant across all trials. Figure 3 shows the performance of a KANN $f_{KANN} \in \mathcal{F}_{\mathcal{D}^{\sigma,t,l},I,(2,12,1)}$ in comparison with a DNN $f_{DNN} \in \mathcal{D}^{\sigma,t,l}$ with respect to their parameter count. Each plot presents the respective test losses for a fixed t while steadily increasing the parameter l. For clarity, after increasing the parameter count, each model was trained starting with randomly distributed parameters. The results demonstrate that KANNs perform optimally when the univariate functions are approximated using shallow neural networks. Specifically, the test loss reaches its minimum at $\mathcal{F}_{\mathcal{D}^{\sigma,2,32},I,n}$, which supports the notion of approximating complex multivariate functions through simpler univariate ones. However, in this scenario, the DNNs outperform the KANNs, achieving the lowest overall test loss.

[2] the structure is described by the tuple n



(a) $\mathcal{F}_{\mathcal{D}^{\sigma,2,l},I,(2,12,1)}$ vs. $\mathcal{D}^{\sigma,3,l}$ (b) $\mathcal{F}_{\mathcal{D}^{\sigma,3,l},I,(2,12,1)}$ vs. $\mathcal{D}^{\sigma,5,l}$ (c) $\mathcal{F}_{\mathcal{D}^{\sigma,4,l},I,(2,12,1)}$ vs. $\mathcal{D}^{\sigma,7,l}$

Figure 3: At each parameter count (biases neglected) the models were trained over 1000 epochs, each epoch consists of 10 optimizing steps, where at each step gradients were computed with respect to 512 randomly drawn samples. The KANNs use tuple $n = (2, 12, 1)$ (cf. Example 3). The neural networks used had shape $[2, \underbrace{r, \dots, r}_{t-times}, 1]$, where r was increased, in order to increase the total number of parameters. The loss was computed on 10000 unseen samples. As optimizer AdamW was employed (cf. [PyT25])

Nonetheless, it can be posited that the functions $f \in P_2[I^2]$, may not possess the requisite complexity to ensure the reliability and robustness of the results, a consequence of their relatively uncomplicated structural nature. This simplicity is reflected in the small value of m and $n_1 = 2$ in Eq. (4.18). Consequently, in the subsequent stage, we will examine learning functions $f \in P_2[I^{12}]$, which exhibit a similar complexity of inner functions to those in Eq. (4.18), but with a considerably larger value of m. This increase in m introduces a substantial degree of complexity, rendering it impractical to identify all univariate polynomials in a straightforward manner. Figure 4 illustrates the test loss of the respective approaches with respect to the parameter count. Due to hardware limitations, a detailed analysis was only feasible for $f \in \mathcal{F}_{\mathcal{S}^\sigma,I,n}$[3]. This limitation will be discussed in more detail in Section 5. As demonstrated, KANNs not only exhibit a test loss that is roughly 3 times lower than that of standard DNNs, but they also attain this enhanced performance with a substantially reduced parameter count. Notably, the optimum was attained with $f \in \mathcal{F}_{\mathcal{S}^{\sigma,14},I,n}$.

[3] Recall, that $\mathcal{S}^\sigma = \mathcal{D}^{\sigma,2}$

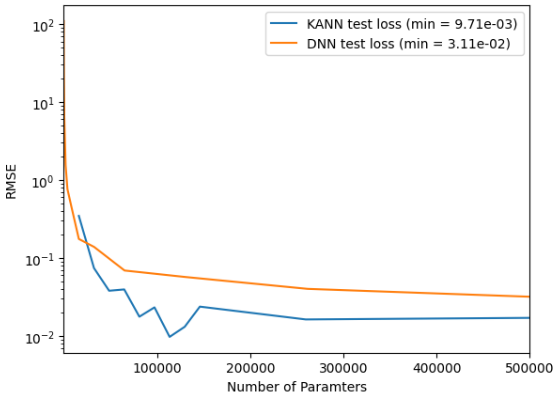
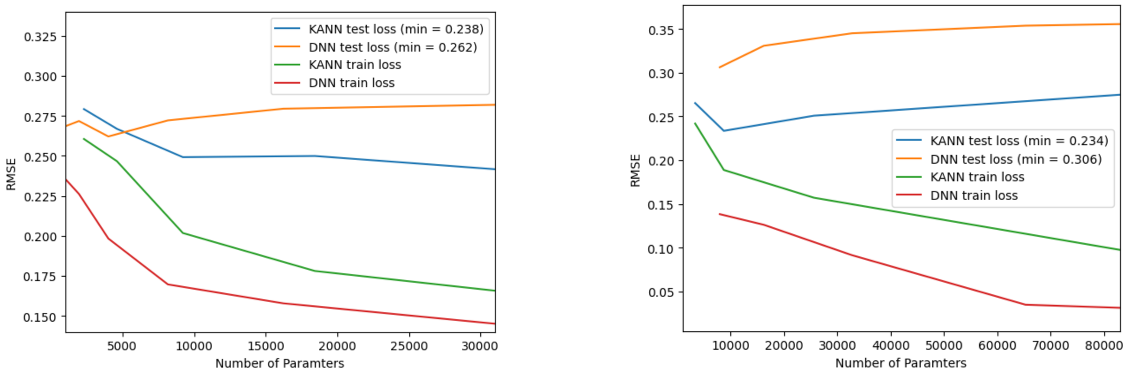


Figure 4: At each parameter count (biases neglected) the models were trained over 1000 epochs, each epoch consists of 10 optimizing steps, where at each step gradients were computed with respect to 512 randomly drawn samples. The KANNs use tuple $n = (12, 312, 1)$ (cf. Example 3). The neural networks used had shape $[12, \underbrace{r, \dots, r}_{t-times}, 1]$, where r was increased, in order to increase the toal number of parameters. The loss was computed on 10000 unseen samples. As optimizer AdamW was employed (cf. [PyT25])

The observed increase in test loss around $150,000$ parameters can be attributed to the introduction of redundant and unnecessary complexity. At this juncture, the KANN has already captured the essential patterns, and the addition of further parameters no longer contributes meaningfully to the learning process. Consequently, this results in inefficient weight updates. This assertion aligns with the concept of approximating complex multivariate functions through simpler univariate ones. It is imperative to acknowledge that prior experiments primarily concentrated on the general expressiveness, with respect to the number of parameters, of the respective approaches. This is because the training data was sampled directly from the objective function, thereby effectively inducing a dataset D (cf. Section 4) with an infinite number of samples. Within the scope of the conducted experiments KANNs attain a level of expressiveness that is comparable to, if not superior to, that of DNNs.

Subsequently, the practical applicability of the proposed method is evaluated through a comparative analysis of the various approaches on the California housing regression dataset [lD24]. The primary distinction is that the optimal tuple n for the function $f \in \mathcal{F}_{\mathcal{D}^{\sigma,t,l},I,n}$ is unknown, prompting an investigation into the effects of using specific structures. Figure 5 compares $\mathcal{F}_{\mathcal{S}^{\sigma,32},I,(8,\underbrace{m,\dots,m}_{\nu\text{-times}},1)}$ with $\mathcal{D}^{\sigma,\nu+2,l}$, where the parameter count was increased by increasing the parameters m and l respectively. It is evident from the conducted experiments that KANNs exhibit superior performance in estimating functions with unknown representations, compared to the baseline DNN approaches.



(a) $f \in \mathcal{F}_{\mathcal{S}^{\sigma,32},I,(8,m,1)}$ vs $\mathcal{D}^{\sigma,3,l}$

(b) $f \in \mathcal{F}_{\mathcal{S}^{\sigma,32},I,(8,m,m,1)}$ vs $\mathcal{D}^{\sigma,5,l}$

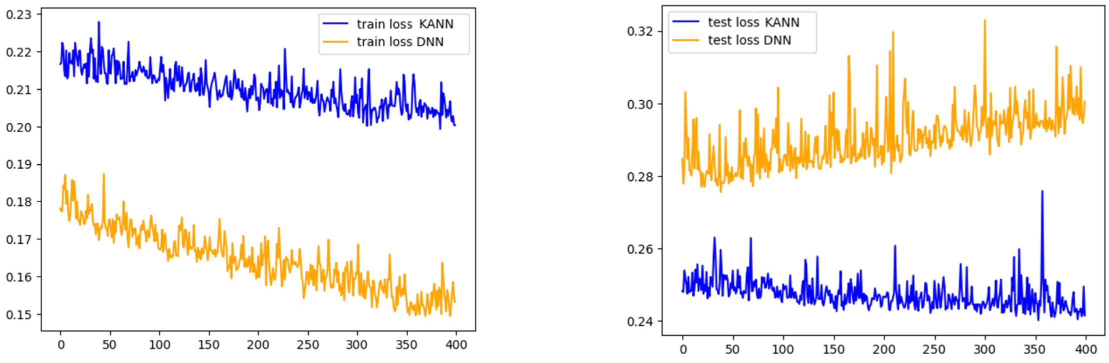
Figure 5: Comparison of parameter efficiency between KANNs and DNNs. The number of parameters was increased by varying m and l, respectively. The networks were trained on the California Housing dataset [lD24] for 1000 epochs using the AdamW optimizer [PyT25].

It can be posited that, in the case of this particular experiment, KANNs mitigate the repercussions of overfitting. Overfitting manifests when a model acquires a memory of the training data instead of discerning general patterns, resulting in the capture of noise rather than the inherent structure of the data. This phenomenon is most commonly observed when a model contains a large number of parameters, as the model may become overly complex and fit even the noise in the data, thereby reducing its ability to generalize. As illustrated in Figure 5, both approaches exhibit distinct patterns in their respective train and test losses. The DNNs consistently outperform the KANNs in terms of train loss, yet the reverse is observed in test loss. Furthermore, as the number of parameters increases, the test loss of the DNNs increases, which does not occur for the test loss of the KANNs.

The findings from both observations lend support to the hypothesis that, in the case of this particular experiment, KANNs are capable of mitigating the effects of overfitting. This conjecture is also observable during the training process. Figure 6 illustrates the final 400 training epochs of the models, with a parameter count of around $1 \cdot 10^4$ (cf. Figure 5a). It is evident that, the train loss of the DNN demonstrates a consistent decline at a rate that is notably more rapid compared to the train loss of the KANNs (cf. Figure 6a). However, the test loss for the DNNs does increase. This phenomenon is indicative of overfitting, wherein the model exhibits difficulty in generalizing to novel, unseen data. Rather than discerning underlying patterns, the model tends to memorize the training data, including its noise and particular details. Consequently, the model's performance on the test set experiences a decline, leading to an increase in the test loss. In contrast, the test loss for the KANNs remains consistent or exhibits a decline, as evidenced in Figure 6b. The following remark proffers an explanation as to why this may be the case. It is important to note, however, that the general validity of these statements is not claimed.

Remark 4.1. *The phenomenon of mitigating the effects of overfitting using KANNs aligns with the observations detailed in Remark 2.1, wherein it was demonstrated that any function can be represented by $\mathcal{F}_{P[I],I,n}$ a composition of "simple"[4] univariate functions. The KANNs efficiently capture these localized univariate functions with a limited number of parameters, thereby reducing the likelihood of overfitting. This is due to the fact that estimating a function with fewer parameters reduces the estimation's complexity, thereby preventing it from fitting noise. Conversely, DNNs learn more complex global structures with a large number of parameters.*

[4] Here, "simple" refers to functions that can be effectively approximated with a small number of parameters, such as univariate polynomials.



(a) Train Loss

(b) Test Loss

Figure 6: Comparison of train and test loss between KANNs $f_{\text{KAN NN}} \in \mathcal{F}_{\mathcal{S}^{\sigma,32},I,(8,32,1)}$ (9,216 parameters) and DNNs $f_{DNN} \in \mathcal{D}^{\sigma,3,95}$ (9,880 parameters) over the final 400 epochs of a total 1,000 training epochs. The models were trained on the California housing dataset.

Finally, we will assess the robustness of each approach to noise. Specifically, the approaches will be evaluated using the so-called Friedmann regression dataset [Fri91], which considers, among others, functions of the form:

$$f_{friedmann} : \mathbb{R}^{n_{in}} \to \mathbb{R} : x \mapsto g(x_1, \ldots, x_m) + \mathcal{N}(0, \sigma^2), \tag{4.19}$$

where $g : \mathbb{R}^m \to \mathbb{R}$ and in general $n_{in} \gg m$. These datasets are particularly useful for testing robustness because they introduce noise in high-dimensional settings, simulating real-world scenarios where irrelevant features can obscure the true signal, making it challenging for models to generalize effectively. We employ $f_{friedmann} \in \mathcal{F}_{Lip_L,I,n}$, where $n = (n_{in}, 6, 2, 1)$. The following approaches are considered for the regression task:

- fixed-shape (fs) KANNs, where the Friedmann function is estimated as $\hat{f}_{\text{Friedmann}} \in \mathcal{F}_{\mathcal{S}^{\sigma},I,n}$.
- arbitrary-width (aw) KANNs, where the Friedmann function is estimated as $\hat{f}_{\text{Friedmann}} \in \mathcal{F}_{\mathcal{S}^{\sigma},I,(n_{\text{in}},m,1)}$ for some $m \in \mathbb{N}$. This is in line with the statements in Remark 2.1.
- Standard DNNs.

In addition, to guarantee the comparability with contemporary methodologies, the following methods will be given due consideration:

- fs KANs, where the Friedmann function is estimated as $\hat{f}_{\text{Friedmann}} \in \mathcal{F}_{S_{d,t},I,n}$, as in Example 3.1.
- aw KANs, where the Friedmann function is estimated as $\hat{f}_{\text{Friedmann}} \in \mathcal{F}_{S_{d,t},I,(n_{\text{in}},m,1)}$ for some $m \in \mathbb{N}$.

Table 1 illustrates the test losses for the various approaches. It is evident that when training is conducted on the regression task devoid of noise and a limited number of irrelevant features, the KAN-based approaches demonstrate superior performance in comparison to alternative methods. However, as the presence of noise or irrelevant features is augmented, the performance undergoes a steady decline, a phenomenon that can be anticipated when employing an interpolation technique such as splines. Additionally, in scenarios with minimal noise, the DNNs consistently exhibit superior performance in comparison to the KANNs. This superiority can be attributed to the utilization of a larger number of parameters by the DNNs. Conversely, as the input dimension and additive noise increase (cf. Eq. (4.19)), the performance of the DNNs declines progressively. In contrast, KANN-based methods exhibit stable performance, as evidenced by their significantly lower variance in test loss. In light of these observations, it can be concluded that, within the scope of the conducted experiments, KANN-based methods exhibit remarkable robustness against noise and irrelevant features and exhibit a notable performance advantage over other approaches in this regard. Moreover, the findings suggest that aw KANNs surpasses other approaches in terms of performance on the given regression problem, particularly demonstrating superior performance compared to fs KANNs. The hypothesis is proposed that increasing the number of parameters in fs KANNs could reduce the mean loss, potentially enabling them to outperform the aw versions. However, a more detailed analysis was not feasible due to practical limitations, which will be discussed later (cf. Section 5). Conversely, the impact of augmenting the number of parameters of the DNNs on its performance was not found to be substantial.

		fs KANN		DNN	aw KANN	fs KAN	aw KAN
σ^2	n_{in}	$\mathcal{F}_{\mathcal{S}^{\sigma,16},I,n}$	$\mathcal{F}_{\mathcal{S}^{\sigma,32},I,n}$	$\mathcal{D}^{\sigma,3,128}$	$\mathcal{F}_{\mathcal{S}^{\sigma,32},I(n_{in},16,1)}$	$\mathcal{F}_{S_{3,t},I,n}$	$\mathcal{F}_{S_{3,t},I,(n_{in},16,1)}$
0.0	5	$2.3 \cdot 10^{-2}$	$2.3 \cdot 10^{-2}$	$6 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$6 \cdot 10^{-3}$	$\mathbf{1.8 \cdot 10^{-5}}$
0.0	10	$3.7 \cdot 10^{-2}$	$7.5 \cdot 10^{-3}$	$2.3 \cdot 10^{-3}$	$1 \cdot 10^{-2}$	$1.2 \cdot 10^{-4}$	$\mathbf{3.1 \cdot 10^{-5}}$
0.0	15	$2.5 \cdot 10^{-2}$	$1.7 \cdot 10^{-2}$	$4.5 \cdot 10^{-3}$	$2.5 \cdot 10^{-2}$	$5.3 \cdot 10^{-2}$	$\mathbf{2.5 \cdot 10^{-4}}$
0.0	100	$5.7 \cdot 10^{-2}$	$1.1 \cdot 10^{-2}$	$2.4 \cdot 10^{-2}$	$\mathbf{2.3 \cdot 10^{-3}}$	1.8	$7.1 \cdot 10^{-1}$
0.2	5	$2.3 \cdot 10^{-2}$	$\mathbf{5 \cdot 10^{-3}}$	$8 \cdot 10^{-3}$	$7 \cdot 10^{-3}$	$7.8 \cdot 10^{-3}$	$3.8 \cdot 10^{-2}$
0.2	10	$2.2 \cdot 10^{-2}$	$2 \cdot 10^{-2}$	$\mathbf{1 \cdot 10^{-2}}$	$\mathbf{1 \cdot 10^{-2}}$	$1.9 \cdot 10^{-2}$	$7.7 \cdot 10^{-2}$
0.2	15	$2.6 \cdot 10^{-2}$	$2.5 \cdot 10^{-2}$	$2.2 \cdot 10^{-2}$	$\mathbf{2.1 \cdot 10^{-2}}$	$3.1 \cdot 10^{-1}$	$9.4 \cdot 10^{-2}$
0.2	100	$2.3 \cdot 10^{-2}$	$2.1 \cdot 10^{-2}$	$4 \cdot 10^{-2}$	$\mathbf{1.5 \cdot 10^{-2}}$	2.8	$2.6 \cdot 10^{-1}$
0.5	5	$3 \cdot 10^{-2}$	$\mathbf{1 \cdot 10^{-2}}$	$2.3 \cdot 10^{-2}$	$\mathbf{1 \cdot 10^{-2}}$	$7.3 \cdot 10^{-2}$	$2.6 \cdot 10^{-1}$
0.5	10	$3.6 \cdot 10^{-2}$	$4.5 \cdot 10^{-2}$	$9.3 \cdot 10^{-2}$	$\mathbf{1.5 \cdot 10^{-2}}$	$3.1 \cdot 10^{-2}$	$4.5 \cdot 10^{-1}$
0.5	15	$4.2 \cdot 10^{-2}$	$3.8 \cdot 10^{-2}$	$7.2 \cdot 10^{-2}$	$\mathbf{2.9 \cdot 10^{-2}}$	$1.8 \cdot 10^{-1}$	$5.3 \cdot 10^{-1}$
0.5	100	$4.1 \cdot 10^{-2}$	$3.6 \cdot 10^{-2}$	$1.6 \cdot 10^{-1}$	$\mathbf{2.3 \cdot 10^{-2}}$	$8 \cdot 10^{-1}$	2.2
1.0	5	$3.9 \cdot 10^{-2}$	$3 \cdot 10^{-2}$	$9 \cdot 10^{-2}$	$\mathbf{2.5 \cdot 10^{-2}}$	$1 \cdot 10^{-1}$	1.03
1.0	10	$6 \cdot 10^{-2}$	$6 \cdot 10^{-2}$	$2.3 \cdot 10^{-1}$	$\mathbf{3.1 \cdot 10^{-2}}$	$2 \cdot 10^{-1}$	1.8
1.0	15	$5.7 \cdot 10^{-2}$	$5.5 \cdot 10^{-2}$	$2.5 \cdot 10^{-1}$	$\mathbf{3.7 \cdot 10^{-2}}$	$8 \cdot 10^{-1}$	2.1
1.0	100	$\mathbf{7.6 \cdot 10^{-2}}$	$1.1 \cdot 10^{-1}$	$6 \cdot 10^{-1}$	$9 \cdot 10^{-2}$	3.7	2.7
Mean		$3.8 \cdot 10^{-2}$	$3.2 \cdot 10^{-2}$	$1 \cdot 10^{-1}$	$2.1 \cdot 10^{-2}$	$6.7 \cdot 10^{-1}$	$7.6 \cdot 10^{-1}$
Variance		$2.5 \cdot 10^{-4}$	$6.5 \cdot 10^{-4}$	$2.2 \cdot 10^{-2}$	$4.1 \cdot 10^{-4}$	1.1	$7.8 \cdot 10^{-1}$

Table 1: Performance evaluation of models on the Friedmann dataset [ld25a] under varying noise levels and input dimensions. The reported values correspond to the RMSE on noise-free test data. The fs models shape is given by $n = (n_{in}, 6, 2, 1)$. The KANNs and DNNs were trained for 1000 epochs using the AdamW optimizer [PyT25]. The KANs were trained on ordered tuples t with cardinalities of 3, 10, and 20, with the best-performing result across all cardinalities presented in the table. Additionally, the KANs were optimized using the L-BFGS algorithm [Dev24] for 100 epochs.

It is noteworthy that the KANs were optimized using the L-BFGS algorithm, a choice that may shed light on their observed performance decline in the presence of irrelevant features and additive noise. However, it was observed that employing the Adam optimizer resulted in a substantially diminished performance. The approaches are evaluated for the other Friedmann-regression problems [Fri91] in Appendix A.4.2. These approaches address issues concerning non-Lipschitz-continuous representations and the effective convergence during the training process.

5 Conclusion and Future Directions

This paper investigated KANNs as a principled alternative to standard (DNNs), grounded in classical approximation theory. We established a rigorous framework for representing multivariate functions as compositions of superpositions of univariate functions, provided bounds on the approximation error, and demonstrated encouraging empirical results suggesting that KANNs can offer robustness and parameter efficiency, particularly in settings with redundant features.

Nevertheless, important limitations persist. The current implementation does not scale efficiently for larger models due to the computational cost of handling large, sparse weight matrices. Additionally, techniques critical for deep architectures—such as normalization, regularization, or residual connections—remain to be systematically adapted for KANNs. The design and effectiveness of deeper or multivariate extensions also warrant further investigation.

Future research should therefore address:

- Investigating training strategies that control the Lipschitz constant while increasing depth and expressive capacity (cf. Remark 3.1).
- Extending the KANN framework to multivariate representations (cf. Appendix A.2.5) and to other neural network architectures, such as recurrent neural networks.
- Benchmarking KANNs on large-scale, complex tasks to better quantify their practical benefits and limitations.

In summary, this work provides a solid theoretical and empirical basis for KANNs as robust, interpretable alternatives to conventional deep learning models. Continued research along these lines may lead to more efficient and generalizable architectures for diverse machine learning challenges.

A Appendix

A.1 Definition of Approximation Error

Before defining the concept of approximation error, it is necessary to extend the notion of distance between points (metrics) to encompass the distance between a set and a point. To this end, let (X, d_X) be a metric space, and let $Y \subseteq X$ and $f \in X$ be given, then we define the distance between these objects denoted by dist as follows:

$$\operatorname{dist}(f, Y) = \inf_{\hat{f} \in Y} d_X(f, \hat{f}). \tag{A.1.20}$$

In the context of approximation, this can be interpreted as the extent to which a function, $f \in X$, can be represented using elements from a set, $Y \subset X$. It is noteworthy that the latter set is often characterized as a "simpler" set of functions when compared to the original set, X. The point that minimizes the distance between the point $f \in X$ and the set Y (cf. Eq. (A.1.20)) is called the projection of f onto Y. Formally, we have the following relation:

$$\operatorname{proj}(f, Y) = \arg\min_{\hat{f} \in Y} d_X(f, \hat{f}). \tag{A.1.21}$$

The proj operator is equivalent to finding a point in $\hat{f} \in Y$ that is closest to some point $f \in X$ with respect to the metric d_X. For the projection operator to be well defined, it is necessary that it be a nonempty set; that is, the minimum must exist. This condition is guaranteed by the following theorem.

Theorem A.1 (Theorem on Existence of Best Approximations in a Metric Space [Che01, Chapter 1, pg.4])**.** *Let (X, d_X) be a metric space, then for all compact sets $Y \subset X$ we have that for all $f \in X$ proj(f, Y) is non empty.*

The present focus is on the approximation of points with specific properties; that is, the necessity for distances between sets arises. In this context, let (X, d_X) be a metric space and consider $Y \subseteq X$ and $Z \subseteq X$. The distance between these sets is then defined as follows:

$$\operatorname{dist}(Y, Z) = \sup_{f \in Y} \operatorname{dist}(f, Z) = \sup_{f \in Y} \inf_{\hat{f} \in Z} d_X(f, \hat{f}). \tag{A.1.22}$$

In the context of approximation, this can be interpreted as the extent to which we can represent any function with certain properties (Y) — such as smoothness — using elements from the "simpler" set of functions Z. Eq. (A.1.22) pertains to the scenario where it is possible to determine a point $\hat{f} \in Z$ in the $\arg\inf_{\hat{f} \in Z} d_X(f, \hat{f})$ of the distance measure for any given point $f \in Y$. However, in most practical cases, this is not feasible, which motivates the introduction of an approximation operator. To this end, let (X, d_X) be a metric space and $Y \subset X$. A mapping $A : X \to Y$ is called an approximation operator because it maps points from the potentially complex set X to the presumably simpler set Y, providing an approximate representation. The distance between an operator A and a set $Z \subset X$ is defined as follows:

$$\operatorname{dist}(Z, A) = \sup_{f \in Z} d_X(f, Af), \tag{A.1.23}$$

where $\operatorname{dist}(Z, A)$ can be interpreted as the maximum approximation error when representing elements of Z using the operator A. A more thorough examination of the properties of approximation operators necessitates the introduction of the concept of sets induced by approximation operators. Let (X, d_X) be a metric space and A an arbitrary approximation operator on X. Additionally, consider the set $Y \subset X$, then the set $AY := \{Af \mid f \in Y\} \subset X$ is the set induced by A with respect to Y. In the event that the condition $dist(Y, A) = dist(Y, AY)$ is satisfied, it can be deduced that A projects elements of Y onto AY. If A projects elements of Y onto AY, then for any approximation operator $B : Y \to AY$, we have $dist(Y, A) \leq dist(Y, B)$. It is noteworthy that the projection of A onto AY is possible under the condition that AY is compact, as substantiated by Theorem A.1. The operator A_1 is said to be superior to the operator A_2 if $dist(Y, A_1) < dist(Y, A_2)$. It is noteworthy that A_1Y and A_2Y may represent entirely distinct sets.

A.2 Section 2

A.2.1 Validity of the Norm Defined in Eq. (2.11)

We now proceed to show that the normed linear space given by:

$$\begin{gathered}(C[I_1^{n_1}, I_N^{n_N}], \|\cdot\|_{\mathcal{F}_{C[I],I,n},\infty}),\\ \text{where } \|f\|_{\mathcal{F}_{C[I],I,n},\infty} = \sup_{x \in I_1^{n_1}} \|f(x)\|_{I_N^{n_N},\infty},\end{gathered} \tag{A.2.37}$$

in fact satisfies the defining axioms of such. In particular we show that the norm $\|f\|_{\mathcal{F}_{C[I],I,n},\infty} = \sup_{x \in I_1^{n_1}} \|f(x)\|_{I_N^{n_N},\infty}$, indeed statisfies the defining axioms of a norm.

1. (Triangle Inequality) For $f, g \in \mathcal{F}_{C[I],I,n}$

   $$\begin{aligned}\|f+g\|_{\mathcal{F}_{C[I],I,n},\infty} \leq \sup_{x \in I_1^{n_1}} \|f(x)+g(x)\|_{I_N^{n_N},\infty} &\leq \sup_{x \in I_1^{n_1}} \|f(x)\|_{I_N^{n_N},\infty} + \sup_{x \in I_1^{n_1}} \|g(x)\|_{I_N^{n_N},\infty}\\ &= \|f\|_{\mathcal{F}_{C[I],I,n},\infty} + \|g\|_{\mathcal{F}_{C[I],I,n},\infty}.\end{aligned}$$

2. (Homogeneity) For $f \in \mathcal{F}_{C[I],I,n}, \alpha \in \mathbb{R}$

   $$\|\alpha f\|_{\mathcal{F}_{C[I],I,n},\infty} = |\alpha| \|f\|_{\mathcal{F}_{C[I],I,n},\infty},$$

   we note here that f is bounded by definition, as continuous functions over compact domains are necessarily bounded.

3. (Positivity) For $f \in \mathcal{F}_{C[I],I,n}$,

   $$\|f\|_{\mathcal{F}_{C[I],I,n},\infty} \geq 0, \|f\|_{\mathcal{F}_{C[I],I,n},\infty} = 0 \iff \forall_i f^i = 0,$$

   which is a direct consequence of the definition of $\|\cdot\|_{I_N^{n_N},\infty}$.

A.2.2 Proof of Eq. (2.13)

We now proceed to proof that the identity:

$$C \prod_{i=1}^{n} x_i = \sum_{S \subseteq \{x_1, \ldots, x_n\}} (-1)^{n-|S|} \left(\sum_{x_i \in S} x_i \right)^n \tag{A.2.39}$$

holds. Before stating the proof, we introduce the following notation for convenience. Let $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ be a vector, we then define the following norms:

$$\|\alpha\|_0 = |\{i \mid \alpha_i \neq 0, \, i = 1, 2, \ldots, n\}|,$$

$$\|\alpha\|_1 = \sum_{i=1}^{n} |\alpha_i|.$$

Additionally we state that the multinomial expansion of $(x_1 + x_2 + \cdots + x_k)^n$ is given by:

$$\begin{aligned}(x_1 + x_2 + \cdots + x_k)^n &= \sum_{a_1+a_2+\cdots+a_k=n} \binom{n}{a_1, a_2, \ldots, a_k} x_1^{a_1} x_2^{a_2} \ldots x_k^{a_k} \\ &= \sum_{\|\alpha\|_1=n} \binom{n}{\alpha_1, \alpha_2, \ldots, \alpha_k} x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_k^{\alpha_k},\end{aligned} \tag{A.2.40}$$

where

$$\binom{n}{a_1, a_2, \ldots, a_k} = \frac{n!}{a_1! a_2! \ldots a_k!}.$$

Proof. We begin by slight rewriting the right hand side of Eq. (A.2.39) to obtain

$$\sum_{S\subseteq\{x_1,\ldots,x_n\}} (-1)^{n-|S|} \left(\sum_{x_i\in S} x_i\right)^n = \sum_{k=1}^{n} (-1)^{n-k} \sum_{\substack{S\subseteq\{x_1,\ldots,x_n\}\\ |S|=k}} (\sum_{x_i\in S} x_i)^n.$$

Now, expanding the inner two sums, we get using Eq. (A.2.40)

$$\sum_{\substack{S\subseteq\{x_1,\ldots,x_n\}\\ |S|=k}} (\sum_{x_i\in S} x_i)^n = \sum_{\substack{\|\alpha\|_1=n\\ \|\alpha\|_0\leq k}} \binom{n}{\alpha_1, \ldots, \alpha_n} \prod_{i=1}^{n} x_i^{\alpha_i} \binom{n-\|\alpha\|_0}{k-\|\alpha\|_0}, \tag{A.2.41}$$

where the binomial coefficient $\binom{n-\|\alpha\|_0}{k-\|\alpha\|_0}$ represents the number of ways a sequence $s \subseteq \{x_1, x_2, \ldots, x_n\} := S$ of cardinality $\|\alpha\|_0 = |s|$ can be part of a subset of size k drawn from the set S of cardinality n. In other words, if we consider all subsets of size k from S, the sequence s appears in exactly $\binom{n-\|\alpha\|_0}{k-\|\alpha\|_0}$ of them. This binomial coefficient precisely quantifies the frequency with which subsets of size k contribute the term $\binom{n}{\alpha_1,\ldots,\alpha_n} \prod_{i=1}^{n} x_i^{\alpha_i}$ in Eq. (A.2.41), which is readily verified by Eq. (A.2.40). As a result, the final polynomial expression in Eq. (A.2.39) contains each multivariate monomial term $\prod_{i=1}^{n} x_i^{\alpha_i}$ with corresponding coefficient:

$$\binom{n}{\alpha_1, \ldots, \alpha_n} \sum_{k=1}^{n} (-1)^{n-k} \binom{n-\|\alpha\|_0}{k-\|\alpha\|_0}.$$

Finally, we notice that

$$\sum_{k=1}^{n} (-1)^{n-k} \binom{n-\|\alpha\|_0}{k-\|\alpha\|_0} = \begin{cases} 1 & \text{if } \|\alpha\|_0 = n \\ 0 & \text{else} \end{cases},$$

where we used the convention that $\binom{n}{m} = 0$ if $n < 0 \vee m < 0$. □

A.2.3 Proof of Remark 3.1

Proof. Analogous to the proof of Theorem 3.1 we obtain

$$\|\Psi_{n+1,k,j} \circ f_n^j - \mathbf{A}(\Psi_{n+1,k,j} \circ f_n^j)\|_{\mathcal{F}_{C[I],I,(n_1,\ldots n_{n+1},1),\infty}}$$

(cf. Eq (A.3.35)). By adding 0 we then obtain

$$\|\Psi_{n+1,k,j} \circ f_n^j - \mathbf{A}(\Psi_{n+1,k,j} \circ f_n^j) + \mathbf{A}(\Psi_{n+1,k,j}) \circ f_n^j - \mathbf{A}(\Psi_{n+1,k,j}) \circ f_n^j\|_{\mathcal{F}_{C[I],I,(n_1,\ldots n_{n+1},1),\infty}},$$

from this, it follows that Remark 3.1 holds if $A\Psi_{l,k,j}$ is Lipschitz continuous. To see this, consider the following inequality:

$$\|\mathbf{A}(\Psi_{n+1,k,j}) \circ f_n^j - \mathbf{A}(\Psi_{n+1,k,j} \circ f_n^j)\| \leq \omega(A\Psi_{n+1,k,j}, \|f_n^j - \mathbf{A} f_n^j\|)$$

and the reasoning used in the proof of Theorem 3.1. The same argument yields the assertion. □

A.2.4 Proof of Lemma 2.1

Lemma A.2.1. *There exist exactly* $\binom{d-1}{w}$ *tuples* $\{x_1, x_2, \ldots, x_w\} \subset \mathbb{N}$ *of cardinality* w*, such that* $\sum_{i=1}^{w} x_i < d$*.*

Proof. Let $y_i = x_i - 1$, then

$$\sum_{i=1}^{w}(y_i + 1) = \sum_{i=1}^{w} x_i < d$$

Rearranging gives:

$$\sum_{i=1}^{w} y_i \leq d - w - 1.$$

Counting the number of non-negative integer solutions to $\sum_{i=1}^{w} y_i = k$, we obtain $\binom{k+w-1}{w-1}$. Hence the total number of solutions is:

$$\sum_{k=0}^{d-w-1} \binom{k+w-1}{w-1}.$$

Using the identity for the sum of binomial coefficients:

$$\sum_{k=0}^{m} \binom{k+r}{r} = \binom{m+r+1}{r+1},$$

where $m = d - w - 1$ and $r = w - 1$, we find:

$$\sum_{k=0}^{d-w-1} \binom{k+w-1}{w-1} = \binom{(d-w-1)+(w-1)+1}{w} = \binom{d-1}{w}.$$

□

A.2.5 Extension to Multivariate Functions Represented as a Composition of Superpositions of Functions of Fewer Variables

We shall now extend Theorem 3.1 to multivariate functions ($f : I_1^{n_1} \to I_N^{n_N}$), represented as a composition of superpositions of functions of fewer variables. To achieve this objective, a minor adjustment is made to the definition presented in the opening of Section 2. Specifically, the following modification is made to Eq. (2.1):

$$\Psi_{l,k,j} : I_l^m \to \mathbb{R}, m \leq n_l, \tag{A.2.27}$$

which results in functions $\Phi_l^k : I_l^{n_l} \to I_{l+1} : x \mapsto \sum_{j=1}^{2^{n_l}-2} \Psi_{l,k,j}(X_j), X_j \subset \{x_1, x_2, \ldots x_{n_l}\}$. This concept is visualized in Figure 7. Similar to the definition of functions represented by compositions of superpositions of univariate functions (cf. Eq. (2.9)), we can define the set of functions of multivariate functions represented as the composition of superposition of functions of fewer variables, in the following way:
$\mathcal{F}_{X,I,n}^{mult} = \{f : I_1^{n_1} \to I_N^{n_N} | f(x) = (\bigcirc_{l=1}^{N-1} \Phi_l)(x)$, as per Eq. ((2.2) - (2.5) and (A.2.27)), $\Psi_{l,k,j} \in X\}$. We define the set $\mathcal{L}_{n,L} := \bigcup_{k=1}^{n-1} \mathrm{Lip}_L[I^k]$. Analogously to the proof of Theorem 3.1, it follows that for any $f \in \mathcal{F}_{\mathcal{L}_L,I,n}^{\mathrm{mult}}$, the following holds:

$$\|f - \mathbf{A}f\|_{\mathcal{F}_{C[I],I,n}^{mult},\infty} \leq C_{L,n} dist(\mathcal{L}_{\max n,L}, A),$$

where $A : C[I^m] \to C[I^m]$ is defined for all m and $\mathbf{A}$ is induced by A as per Eq. (2.10). For dist we use the measure of distance:

$$d(f,g) = \begin{cases} \sup_{x \in dom(f)} |f(x) - g(x)|, & \text{if } dom(f) = dom(g), \\ \infty, & \text{else.} \end{cases}$$

Note that this is not a metric, as it takes values outside of $\mathbb{R}^+$. However, by the definition of the operator A, it still provides reliable results for our purposes.

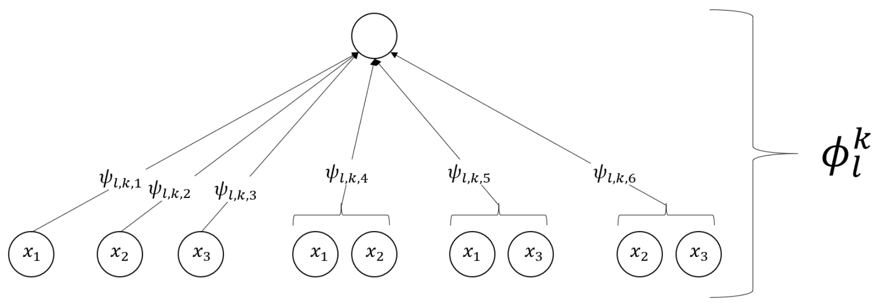


Figure 7: Graphical representation of a function $\Phi_l^k : I_l^3 \to I_{l+1}$. Each arrow represents a function $\Psi_{l,k,j}$ applied to an input, which is depicted by a circle. The graph is read from bottom to top, where multiple lines joining at a circle in the upward direction indicate the application of a summation operation.

A.3 Section 3

Theorem A.2 (Approximation Theorem). *For any $f \in \mathcal{F}_{Lip_L,I,n}$ and any approximation operator A on $Lip_L[I]$, we have the inequality,*

$$\|f - \mathbf{A}f\|_{\mathcal{F}_{C[I],I,n},\infty} \leq C_{L,n}\,\mathrm{dist}(Lip_L, A),$$

where $C_{L,n}$ is a constant depending on the Lipschitz-constant L of function $\Psi_{l,k,j} \in Lip_L[I] \subset C[I]$ and on the tuple n. The distance measure dist *is defined in Eq.* (A.1.23) *and is calculated with respect to the metric induced by $\|\cdot\|_\infty$. The operator $\mathbf{A}$ is induced by A, cf. Eq.* (2.10)*.*

Before stating the proof we need the notion of *modulus of continuity* given by the following definition:

Definition A.1 (Modulus of Continuity). *Let $f : (X, d_X) \to (Y, d_Y)$ be a function. The modulus of continuity $\omega(f, \cdot) : \mathbb{R}^+ \to \mathbb{R}^+$ of f is defined as*

$$\omega(f, h) := \sup\{d_Y(f(x), f(x')) : x, x' \in X, d_X(x, x') \leq h\}.$$

Proof. We prove the statement using induction on $N - 1$, where N denotes the cardinality of the tuple n.

Base Case:

We shall now proceed to derive the distance between f_1[5] and $\mathbf{A}f_1$, measured in the norm $\|\cdot\|_{\mathcal{F}_{C[I],I,(n_1,n_2)},\infty}$. Consequently, the subsequent relation is derived:

$$\begin{aligned}
\|f_1 - \mathbf{A}f_1\|_{\mathcal{F}_{C[I],I,(n_1,n_2)}\infty} &= \|\Phi_1 - \mathbf{A}\Phi_1\|_{\mathcal{F}_{C[I],I,(n_1,n_2),\infty}} && \text{(A.3.28)}\\
&= \sup_{k\in\{1,\ldots,n_2\}} \|\Phi_1^k - \mathbf{A}\Phi_1^k\|_{\mathcal{F}_{C[I],I,(n_1,1),\infty}} && \text{(A.3.29)}\\
&= \sup_{k\in\{1,\ldots,n_2\}} \sup_{x\in I_1^{n_1}} |\sum_{j=1}^{n_1} \Psi_{1,k,j}(x_j) - A\Psi_{1,k,j}(x_j)| && \text{(A.3.30)}\\
&\leq n_1 \sup_{\substack{k\in\{1,\ldots,n_2\}\\ j\in\{1,\ldots,n_1\}}} \|\Psi_{1,k,j} - A\Psi_{1,k,j}\|_{C[I],\infty} && \text{(A.3.31)}\\
&\leq n_1\,\mathrm{dist}(Lip_L, A), && \text{(A.3.32)}
\end{aligned}$$

where the equality in Equation (A.3.28) follows directly from the definition of the set $\mathcal{F}_{\mathrm{Lip}_L,I,n}$ (cf. Section 2). The equivalence between Eq.s (A.3.28) and (A.3.29) follows

[5] This notation is defined in Eq. (2.7))

immediately from the definition of the norm $\|\cdot\|_{\mathcal{F}_{C[I],I,(n_1,n_2),\infty}}$ (see Eq. (2.11)). By said definition, this is also equal to the expression in Eq. (A.3.30). Applying the triangle inequality to Eq. (A.3.30) yields Eq. (A.3.31), and finally, the inequality to Eq. (A.3.32) follows from the definition of the distance measure dist (cf. Eq. (A.1.23)).

Induction hypothesis:

We define the residue R_m as

$$R_m = f_m - \mathbf{A}f_m.$$

The induction hypothesis, is then given by the following inequality:

$$\|R_m\|_{\mathcal{F}_{C[I],I,(n_1,\dots,n_{m+1}),\infty}} \leq C'_{L,n}\,\mathrm{dist}(Lip_L, A),$$

where $C'_{L,n}$ is a constant, as in the statement of the theorem.

Induction Step:

We shall now proceed to derive an upper bound on $\|R_{m+1}\|_{\mathcal{F}_{C[I],I,n,\infty}}$. Consequently, the subsequent relation is derived:

$$\begin{aligned}\|R_{m+1}\|_{\mathcal{F}_{C[I],I,n,\infty}} &= \|f_{m+1} - \mathbf{A}f_{m+1}\|_{\mathcal{F}_{C[I],I,n},\infty} \\ &= \|\Phi_{m+1} \circ f_m - \mathbf{A}(\Phi_{m+1} \circ f_m)\|_{\mathcal{F}_{C[I],I,(n_1,\dots,n_{m+1},n_{m+2}),\infty}}\end{aligned} \tag{A.3.33}$$

$$= \sup_{k\in\{1,\dots,n_{n+2}\}} \sup_{x\in I_1^{n_1}} |\sum_{j=1}^{n_{m+1}} \Psi_{m+1,k,j} \circ f_m^j(x) - \mathbf{A}\left(\Psi_{m+1,k,j} \circ f_m^j(x)\right)| \tag{A.3.34}$$

$$\leq n_{m+1} \sup_{\substack{k\in\{1,\dots,n_{m+2}\}\\ j\in\{1,\dots,n_{m+1}\}}} \|\Psi_{m+1,k,j} \circ f_m^j - \mathbf{A}(\Psi_{m+1,k,j} \circ f_m^j)\|_{\mathcal{F}_{C[I],I,(n_1,\dots n_{m+1},1),\infty}}, \tag{A.3.35}$$

where the relation in Eq. (A.3.33) is derived form the utilization of the fact that any $f_{m+1} \in \mathcal{F}_{C[I],I,(n_1,\dots n_{m+2})}$ can be expressed as $f_{m+1} = \Phi_{m+1} \circ f_m$ (cf. Eq. (2.7)). Subsequently, the equivalence to Eq. (A.3.34) follows in a manner similar to the base case. The notation f_m^j refers to the j-th component of f_m. Finally, the inequality between Eq. (A.3.34) and Eq. (A.3.35) follows by applying the triangle inequality, analogous to the base case. We shall now focus on the term $\|\cdot\|_{\mathcal{F}_{C[I],I,(n_1,\dots n_{m+1},1)}}$ in Eq. (A.3.35). By adding 0 and applying the triangle inequality, we obtain the following inequality:

$$\begin{aligned}&\left\|\Psi_{m+1,k,j} \circ f_m^j - \mathbf{A}(\Psi_{m+1,k,j} \circ f_m^j) + \Psi_{m+1,k,j} \circ \mathbf{A}f_m^j - \Psi_{m+1,k,j} \circ \mathbf{A}f_m^j\right\|_{\mathcal{F}_{C[I],I,(n_1,\dots,n_{m+1},1),\infty}} \\ &\leq \underbrace{\left\|\Psi_{m+1,k,j} \circ f_m^j - \Psi_{m+1,k,j} \circ \mathbf{A}f_m^j\right\|_{\mathcal{F}_{C[I],I,(n_1,\dots,n_{m+1},1),\infty}}}_{\square} \\ &\quad + \underbrace{\left\|\Psi_{m+1,k,j} \circ \mathbf{A}f_m^j - \mathbf{A}(\Psi_{m+1,k,j} \circ f_m^j)\right\|_{\mathcal{F}_{C[I],I,(n_1,\dots,n_{m+1},1),\infty}}}_{\triangle}.\end{aligned}$$

We now proceed to derive an upper bound by analyzing each summand separately.

1. For □ we obtain the upper bound

$$\begin{aligned}&\|\Psi_{m+1,k,j} \circ f_m^j - \Psi_{m+1,k,j} \circ \mathbf{A}f_m^j\|_{\mathcal{F}_{C[I],I,(n_1,\dots n_{m+1},1)}} \\ &\leq \omega\left(\Psi_{m+1,k,j}, \|f_m^j - \mathbf{A}f_m^j\|_{\mathcal{F}_{C[I],I,(n_1,\dots n_m,1),\infty}}\right) \\ &\leq L\|f_m^j - \mathbf{A}f_m^j\|_{\mathcal{F}_{C[I],I,(n_1,\dots n_m,1)}} \leq C''_{L,n_{m+1}}\|R_m\|_{\mathcal{F}_{C[I],I,(n_1,\dots n_m,n_{m+1}),\infty}},\end{aligned} \tag{*}$$

   where ω was already defined before (cf. Definition A.1) and the last inequality follows from the fact that

$$\|f_m^j - \mathbf{A}f_m^j\|_{\mathcal{F}_{C[I],I,(n_1,\dots n_m,1),\infty}} \leq \|f_m - \mathbf{A}f_m\|_{\mathcal{F}_{C[I],I,(n_1,\dots n_{m+1}),\infty}}$$

2. For $\triangle$ we obtain, analogous to the Base Case, the upper bound

$$\|\Psi_{m+1,k,j} \circ \mathbf{A} f_m^j - \mathbf{A}(\Psi_{m+1,k,j} \circ f_m^j)\|_{\mathcal{F}_{C[I],I,(n_1,\dots n_{m+1},1),\infty}} \leq \mathrm{dist}(Lip_L, A). \qquad (**)$$

The Inequalities (*) and (**), together with the induction hypothesis, yield the assertion. □

A.4 Section 4

A.4.1 Proof of Lemma 4.1

We shall now proceed to proof the subsequent lemma.

Lemma A.4.1. *The relation:*

$$\mathcal{F}_{\mathcal{D}^{\sigma,t,l},I,n} \subset \mathcal{D}^{\sigma}[\mathbb{R}^{n_1}, \mathbb{R}^{n_N}],$$

where $\mathcal{D}^\sigma := \bigcup_{t,l}^{\infty} \mathcal{D}^{\sigma,t,l}$*, is valid.*

Before giving the proof we state an auxiliary lemma.

Lemma A.4.2. *Let* $f : \mathbb{R}^l \to \mathbb{R}^m$ *and* $f' : \mathbb{R}^n \to \mathbb{R}^l$ *be functions such that* $f, f' \in \mathcal{D}^\sigma$*, then*

$$f \circ f' \in \mathcal{D}^\sigma$$

Proof. The functions in $\mathcal{D}^\sigma$ take the form:

$$f : x \mapsto A\sigma\left(W_{t-1} \cdots \sigma\left(W_2 \sigma\left(W_1 x + b_1\right) + b_2\right) \cdots + b_{t-1}\right).$$

The composition of two such functions is then given by:

$$f \circ f' : x \mapsto A\sigma\left(W_{t-1} \cdots \sigma\left(W_1 A' \sigma\left(W'_{t-1} \cdots \sigma\left(W'_1 x + b'_1\right) \cdots + b'_{t-1}\right) + b_1\right) \cdots + b_{t-1}\right),$$

which is of the desired form and hence yields the assertion. □

We are now prepared to continue with the proof of Lemma A.4.1.

Proof. We prove the statement using induction on $N-1$, where N denotes the cardinality of the tuple n.

Base Case:

The univariate functions of $f \in \mathcal{F}_{\mathcal{D}^{\sigma,t,l},I,(n_1,n_2)}$ are given in the form:

$$\Psi_{1,k,j} : x \mapsto A_{k,j}\sigma\left(W_{k,j,t-1} \cdots \sigma\left(W_{k,j,2}\sigma\left(W_{k,j,1}x + b_{k,j,1}\right) + b_{k,j,2}\right) \cdots + b_{k,j,t-1}\right),$$

where $k = 1, \dots, n_2$ and $j = 1, \dots, n_1$. We can now generalize the previous representation by incorporating matrices W_m and biases b_m, for $m = 1, \dots, t-1$, which denotes the depth of the univariate DNNs, in the following form:

$$W_m = \begin{bmatrix}
W_{1,1,m} & 0 & \cdots & 0 \\
0 & W_{1,2,m} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & W_{1,n_1,m} \\
W_{2,1,m} & 0 & \cdots & 0 \\
0 & W_{2,2,m} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & W_{2,n_1,m} \\
\vdots & \vdots & \vdots & \vdots \\
W_{n_2,1,m} & 0 & \cdots & 0 \\
0 & W_{n_2,2,m} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & W_{n_2,n_1,m}
\end{bmatrix}$$

and

$$b_m = \begin{bmatrix} b_{1,1,m} \\ b_{1,2,m} \\ \vdots \\ b_{1,n_1,m} \\ b_{2,1,m} \\ \vdots \\ b_{n_2,n_1,m} \end{bmatrix}.$$

Hence we may write for $f \in \mathcal{F}_{\mathcal{D}^{\sigma,t,l},I,(n_1,n_2)}$:

$$f : x \mapsto A\sigma\left(W_{t-1} \cdots \sigma\left(W_2\sigma\left(W_1 x + b_1\right) + b_2\right) \cdots + b_{t-1}\right),$$

where A introduces the summation operator and is defined as follows:

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,n_1} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & A_{2,1} & \cdots & A_{2,n_1} & 0 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \ddots & & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & \cdots & \cdots & A_{n_2,1} & \cdots & A_{n_2,n_1} \end{bmatrix},$$

which yields the base case. An example of the construction of these matrices can be found in Section 4.1.

Induction hypothesis:

We shall now assume that $\mathcal{F}_{\mathcal{D}^{\sigma,t,l},I,(n_1,n_2,\ldots,n_m)} \subset \mathcal{D}^{\sigma}$

Induction Step:

In accordance with the results from Section 2 we now consider the form $f_{m+1} = \Phi_{m+1} \circ f_m$ (cf. Eq. (2.7)), where $\Phi_{m+1} \in \mathcal{F}_{\mathcal{D}^{\sigma,t,l},I,(n_m,n_{m+1})}$, $f_m \in \mathcal{F}_{\mathcal{D}^{\sigma,t,l},I,(n_1,n_2,\ldots,n_m)}$ and $f_{m+1} \in \mathcal{F}_{\mathcal{D}^{\sigma,t,l},I,(n_1,n_2,\ldots,n_{m+1})}$. As per the results shown in the base case $\Phi_{m+1} \in \mathcal{D}^{\sigma}$ and in accordance with the induction hypothesis $f_m \in \mathcal{D}^{\sigma}$. These facts in conjunction with Lemma A.4.2 yield the assertion. □

A.4.2 Friedmann 2 and Friedmann 3

Remark A.1. *The results of Theorem 3.1 can be extended to the set $\mathcal{F}_{Lip_{L,\alpha},I,n}$ for $\alpha < 1$ as follows:*

$$\|f - \mathbf{A}f\|_{\mathcal{F}_C[I],I,n} \leq C_{L,n}\,\mathrm{dist}(\mathrm{Lip_{L,\alpha}[I], A})^{\alpha^{\mathrm{N\text{-}2}}},$$

where $N = |n|$. The proof of this extension follows similarly to the one presented for Theorem 3.1.

Tables 2 and 3 present the performance of the respective approaches introduced in Section 4.2 for Friedmann regression problems. The regression problems are given by

$$\begin{aligned} f_2(x_1,x_2,x_3,x_4) &= \left(x_1^2 + (x_4x_2 - (x_4x_3)^{-1})^2\right)^{0.5} + \mathcal{N}(0,\sigma^2) \qquad \text{Table 2} \\ f_3(x_1,x_2,x_3,x_4) &= \arctan\left(\frac{x_2x_3}{x_1} - \frac{1}{x_1x_2x_4}\right) + \mathcal{N}(0,\sigma^2) \qquad \text{Table 3}, \end{aligned} \tag{A.4.36}$$

they denote the magnitude of the impedance and the phase of a series RLC circuit respectively (cf. [Fri91, Eq. 63]). It is observed that neither of these functions belongs to $\mathcal{F}_{Lip_L,I,n}$. Specifically, the function f_2 contains Hölder-continuous functions, i.e., $f(x) = \sqrt{x}$. We emphasize that for f_2 in Eq. (A.4.36), the $\sqrt{\cdot}$ function is applied to values in the interval $[2.6 \times 10^{-9}, 3.1 \times 10^{6}]$. Although it is Lipschitz continuous on any closed interval bounded away from zero, the extremely small lower bound results in a large Lipschitz constant. As a result, and to better reflect the behavior near the lower end of the interval, we treat f_2 as Hölder continuous in the following analysis. Conversely, the function f_3 does not even admit a uniformly continuous representation as $x_1 \in [0, 100]$ (cf. [ld25c])[6]. Moreover, both functions in Eq. (A.4.36) possess a broadly analogous overall structure, with the exception of the outermost function. The function f_3 smoothed the outputs by applying a Lipschitz-continuous, sigmoidal-like function (arctan), which bound the range and mitigated extreme variations. Conversely, the function f_2 introduces a Hölder-continuous function ($\cdot^{0.5}$) to its outputs, resulting in increased variability and irregularity. This, in turn, leads to heightened sensitivity to input fluctuations. In these cases, we do not include irrelevant features, resulting in smaller input dimensions. This allows us to add more parameters to each approach for improved flexibility in model tuning.

	fs KANN		DNN	aw KANN		aw KAN
σ	$\mathcal{F}_{\mathcal{S}^{\sigma,32},I,n}$	$\mathcal{F}_{\mathcal{S}^{\sigma,128},I,n}$	$\mathcal{D}^{\sigma,3,512}$	$\mathcal{F}_{\mathcal{S}^{\sigma,64},I(4,16,1)}$	$\mathcal{F}_{\mathcal{S}^{\sigma,16},I(4,128,1)}$	$\mathcal{F}_{S_{3,t},I,(4,64,1)}$
0.0	$2.9 \cdot 10^{-1}$	$3.9 \cdot 10^{-1}$	$5.7 \cdot 10^{-1}$	$4.11 \cdot 10^{-1}$	$1.7 \cdot 10^{-1}$	$\mathbf{4.1 \cdot 10^{-2}}$
0.5	1.02	6.2	2	$4.17 \cdot 10^{-1}$	$2.8 \cdot 10^{-1}$	$\mathbf{7.1 \cdot 10^{-2}}$
1.0	3.8	$3.7 \cdot 10^{-1}$	5	$6 \cdot 10^{-1}$	$4.6 \cdot 10^{-1}$	$\mathbf{2.1 \cdot 10^{-1}}$
2.0	1.5	1.1	1.6	$8 \cdot 10^{-1}$	$\mathbf{4.8 \cdot 10^{-1}}$	1
5.0	3.7	10.8	2.8	1.46	**1.3**	9

Table 2: Performance evaluation of models on the Friedmann dataset [ld25b] across varying noise levels. The reported values represent the RMSE on test data without noise. The shape of the fixed-size approaches is given as $n = (4, 6, 3, 2, 1, 1)$. All models were trained for 1000 epochs using the AdamW optimizer [PyT25]. The input data was standardized cf. [ld21]. For the KANs we choose $|t| = 10$.

The results presented in Table 2 are obtained by training on the function f_2 in Eq. (A.4.36). Higher levels of noise are employed, as the range of this function is significantly larger, that is, $f_2(x) \in (5.1 \cdot 10^{-5}, 1762.13)$ compared to other Friedmann functions. The substantial variations in outcomes, particularly for the fs KANNs $\mathcal{F}_{\mathcal{S}^{\sigma,128},I,n}$ can be attributed to a highly unstable training process. It was observed that convergence was achieved even in the absence of noise, provided that adaptive learning rate strategies were employed, and these strategies were applied once a predefined loss threshold was attained. This approach was further extended to the DNNs [7]. However, this was not a prerequisite for the aw KANNs. One plausible explanation for the more robust training process of aw KANNs is that the increased sparsity of the respective weight matrices leads to smaller operator norms of the gradients, which in turn reduces the effective learning rate. A notable observation is that reducing the learning rate from the outset causes both the fs KANNs and DNNs to converge too slowly. Furthermore, fluctuations in the training process are observed even when adjusting the learning rate during training (cf. Table 2). Furthermore, aw KANNs may initially converge slower but still at a significant pace, remaining sufficiently fast. One possible interpretation of these phenomena is that this instability lies in the challenging trade-off between the Lipschitz constant of the univariate approximation function and the accurate estimation of univariate Hölder-continuous functions in the case of employing fs approaches (cf. Remark 3.1). The latter generally necessitates substantial Lipschitz constants for approximation functions, which can result in unstable minima, particularly in the presence of noise. Furthermore, the aw KANNs demonstrated a convergence pattern consistent with the estimation of smooth continuous functions, as discussed in Remark 2.1. It is noteworthy that we have observed satisfactory convergence even when employing aw and arbitrary-depth KANNs. However, we do not assess these in their entirety due to the extended training time required (cf. Section 5). However, for the specific case of $\mathcal{F}_{\mathcal{S}^{\sigma,16},I,(4,64,64,1)}$ and $\sigma^2 = 1$, we observed a test loss of

[6] We identify for $f : x \mapsto \frac{1}{x}$ $f(0) := \lim_{x \to 0^+} f(x)$

[7] For the fs KANs, no substantial convergence was observed during the training process.

$1.9 \cdot 10^{-1}$. The mitigation of these counterproductive effects could be achieved through the implementation of specific training strategies. The results presented in Table 3 are

	fs KANN		DNN	aw KANN		fs KAN	aw KAN
σ^2	$\mathcal{F}_{\mathcal{S}^{\sigma,32},I,n}$	$\mathcal{F}_{\mathcal{S}^{\sigma,128},I,n}$	$\mathcal{D}^{\sigma,3,512}$	$\mathcal{F}_{\mathcal{S}^{\sigma,64},I(4,16,1)}$	$\mathcal{F}_{\mathcal{S}^{\sigma,16},I(4,128,1)}$	$\mathcal{F}_{S_{3,t},I,n}$	$\mathcal{F}_{S_{3,t},I,(4,64,1)}$
0.0	$\mathbf{5.1 \cdot 10^{-5}}$	$1.1 \cdot 10^{-4}$	$1.1 \cdot 10^{-4}$	$1.4 \cdot 10^{-4}$	$2.4 \cdot 10^{-4}$	$6.8 \cdot 10^{-3}$	$5.6 \cdot 10^{-2}$
0.2	$\mathbf{1.4 \cdot 10^{-3}}$	$2 \cdot 10^{-3}$	$3.1 \cdot 10^{-2}$	$2.2 \cdot 10^{-3}$	$2.8 \cdot 10^{-3}$	$1 \cdot 10^{-2}$	$3.5 \cdot 10^{-2}$
0.5	$\mathbf{6.2 \cdot 10^{-3}}$	$1.6 \cdot 10^{-2}$	$2.2 \cdot 10^{-1}$	$1.1 \cdot 10^{-2}$	$1.9 \cdot 10^{-2}$	$7.8 \cdot 10^{-2}$	$2.1 \cdot 10^{-1}$
1.0	$\mathbf{2.9 \cdot 10^{-2}}$	$6.2 \cdot 10^{-2}$	1.02	$4.7 \cdot 10^{-2}$	$7.2 \cdot 10^{-2}$	$2.8 \cdot 10^{-1}$	$8.3 \cdot 10^{-1}$

Table 3: Performance evaluation of models on the Friedmann dataset [ld25c] across varying noise levels. The reported values represent the RMSE on test data without noise. The shape of the fs KANNs is given as $n = (4, 13, 1, 1)$. All models were trained for 1000 epochs using the AdamW optimizer [PyT25]. The input data was standardized cf. [ld21]. For the KANs we choose $|t| = 10$.

obtained by training on the function f_3 in Eq. (A.4.36). In this context, the significant variations (leading to non-uniform continuity) observed near 0 of the functions $\frac{1}{x}$ are effectively mitigated by the sigmoidal nature of the arctan function. The behavior of the function arctan is more smooth for large values, which reduces the sensitivity to large fluctuations. This effect culminates in a more stable approximation of the function. The fs KANNs demonstrates superior performance in this instance, capitalizing on its smoothing property. Interestingly, as opposed to the other experiments, the KAN-based approaches do not exhibit superior performance in the noise free case. This can be explained by the large range of the function to which the arctan is applied, leading to large distances between the grid points of the splines. Assuming the grid points are uniformly distributed, this results in suboptimal outcomes, as the non-uniform complexity of sigmoid functions is not well captured. The results presented in Tables 1, 2, and 3 illustrate that the optimal approach varies depending on the specific task, emphasizing the importance of selecting an appropriate model for each scenario

A.5 Miscellaneous Results

Theorem A.3 ([dBF73, Section 3, Thm 2.1])**.** *Let $f \in C^d[I]$ and let $A_{t,d} : C^d[I] \to S_{d,t}$, then for the approximation error in the r-th derivative the following relation holds:*

$$\|D^r f - D^r A_{t,d} f\|_\infty \le C_r \omega(f^d, \Delta t) \Delta t^{d-r}, r \le d, \tag{A.5.37}$$

where C_r is some constant dependent on r and $\Delta t = \max_i t_i - t_{i-1}$.

References

[Buc79] R. Buck. Approximate complexity and functional representation. *J. Math. Anal. Appl.*, 70(1):280–298, 1979.

[Che01] E. W. Cheney. *Introduction to Approximation Theory*. AMS Chelsea Publishing, Providence, RI, 2001.

[dB01] Carl de Boor. *A Practical Guide to Splines (Revised Edition)*. Springer, 2001.

[dBF73] C. de Boor and G. J. Fix. Spline approximation by quasiinterpolants. *Journal of Approximation Theory*, 8:19–45, 1973. Communicated by Oued Shisha.

[Dev24] PyTorch Developers. torch.optim.lbfgs, 2024. Accessed: March 22, 2025.

[Fri91] J. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67, 1991.

[HSW$^+$21] Rui Hu, Jitao Sang, Jinqiang Wang, Rui Hu, and Chaoquan Jiang. Understanding and testing generalization of deep networks on out-of-distribution data. *arXiv preprint arXiv:2111.09190*, 2021.

[Kol57] A. N. Kolmogorov. On the representation of continuous functions of several variables by superposition of continuous functions of one variable and addition. *Doklady Akademii Nauk SSSR*, 114:953–956, 1957.

[KTWZ24] Mathias Kraus, Daniel Tschernutter, Sven Weinzierl, and Patrick Zschech. Interpretable generalized additive neural networks. *European Journal of Operational Research*, 317(2):303–316, 2024.

[ld21] Scikit learn developers. Standardscaler. Scikit-learn documentation, 2021. Accessed: 2025-03-19.

[lD24] Scikit learn Developers. California housing dataset, 2024.

[ld25a] Scikit learn developers. makefriedman1, 2025. Accessed: 2025-03-13.

[ld25b] Scikit learn developers. makefriedman2, 2025. Accessed: 2025-03-13.

[ld25c] Scikit learn developers. makefriedman3. Scikit-learn Documentation, 2025. Accessed: 2025-03-19.

[LWV$^+$24] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, and Max Tegmark. KAN: Kolmogorov–Arnold Networks. *arXiv preprint arXiv:2404.19756v1*, 2024. License: CC BY 4.0.

[Mar18] Gary Marcus. Deep learning: A critical appraisal, 2018.

[Pot99] William J. E. Potts. Generalized additive neural networks. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, page 194–200, New York, NY, USA, 1999. Association for Computing Machinery.

[Pra25] Justus Prass. https://github.com/Justus-ui/KAN_NN, 2025. Accessed: 2025-04-08.

[PyT25] PyTorch. torch.optim.adamw, 2025. Accessed: 2025-03-04.

[SL19] Shaeke Salman and Xiuwen Liu. Overfitting mechanism and avoidance in deep neural networks, 2019.

[TGLM22] Neil C. Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F. Manso. The computational limits of deep learning, 2022.

[Wer18] D. Werner. *Funktionalanalysis*. Springer Spektrum, Berlin, 8. überarbeitete auflage edition, 2018.

[ZTLT21] Yu Zhang, Peter Tino, Ales Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(5):726–742, October 2021.