VARIATIONAL KOLMOGOROV-ARNOLD NETWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

Kolmogorov-Arnold Networks (KANs) offer a theoretically grounded alternative to multi-layer perceptrons by representing multivariate functions as compositions of univariate basis functions. However, a critical limitation of KANs is the need to manually specify the number of basis functions per layer, a hyperparameter that directly controls model capacity and substantially impacts performance, yet whose optimal value varies unpredictably across tasks. We present INFINITYKAN, a variational inference framework that eliminates this design choice by learning the number of basis functions during training. Our approach models the basis count as a latent variable with a truncated exponential prior, introducing a differentiable weighting function that enables gradient-based optimization. We establish the Lipschitz continuity of the variational objective, ensuring stable training dynamics. Experiments across 18 datasets spanning synthetic, image, tabular, and graph domains demonstrate that INFINITYKAN matches or exceeds the performance of KANs while requiring no manual selection of the number of bases for each layer.

1 INTRODUCTION

Kolmogorov-Arnold Networks (KANs) (Liu et al., 2024) have recently gained attention in the machine learning community as a potential alternative to the widely-used Multi-Layer Perceptrons (MLPs) (Hornik et al., 1989). MLPs have been instrumental in transforming machine learning due to their ability to approximate any continuous function, a capability supported by the universal approximation theorem (Hornik et al., 1989). The Kolmogorov-Arnold Theorem (KAT), originally developed to address Hilbert's 13th problem, is a fundamental mathematical result with numerous implications (Kolmogorov, 1961). While the universal approximation theorem suggests that any continuous function can be approximated using an MLP of bounded width, KAT represents any multivariate function exactly using a finite and known number of univariate functions. KAT's influence extends beyond pure mathematics, finding applications in diverse fields such as fuzzy logic, pattern recognition, and neural networks (Laczkovich, 2021; Kreinovich et al., 1996; Köppen, 2002; Kŭrková, 1992; Liu et al., 2024). This versatility has contributed to its growing importance in the machine-learning community. KAT-based results have been applied in several ways and stand as a potential alternative to MLPs in solving arbitrary tasks (Xu et al., 2024b; Carlo et al., 2024).

KAT leads to various alternative representations (Guilhoto & Perdikaris, 2025; Alesiani et al., 2025), where the internal univariate functions are independent of the specific target function being represented (Ismayilova & Ismailov, 2024). Historically, the construction of these univariate functions has been addressed either through constructive approximation procedures (Kůrková, 1991; Kŭrková, 1992) or via numerical algorithms defined on dense subsets of dyadic rationals (Katsura & Sprecher, 1994; Sprecher, 1996; 1997). However, due to their highly oscillatory and non-smooth nature, these classical universal representations do not generalize well nor are easy to incorporate in a differentiable model. To address this limitation, recent work has shifted toward parameterized, trainable univariate functions (Liu et al., 2024) designed as expansions over smooth base functions.

While KAT argues for the existence of a univariate functions that represent the target function exactly, when representing this function using basis functions, *the choice of the number of basis functions that model each univariate function remains an open problem*. It is of no surprise that KANs' effectiveness in addressing complex, high-dimensional problems heavily relies on the choice, construction, and training of appropriate basis functions. Various proposals have been made, such as orthogonal polynomials, spline, sinusoidal, or wavelets, which may depend on the specific problem at hand (SS et al., 2024; Mostajeran & Faroughi, 2024; Bozorgasl & Chen, 2024; Xu et al., 2024a). Beyond

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

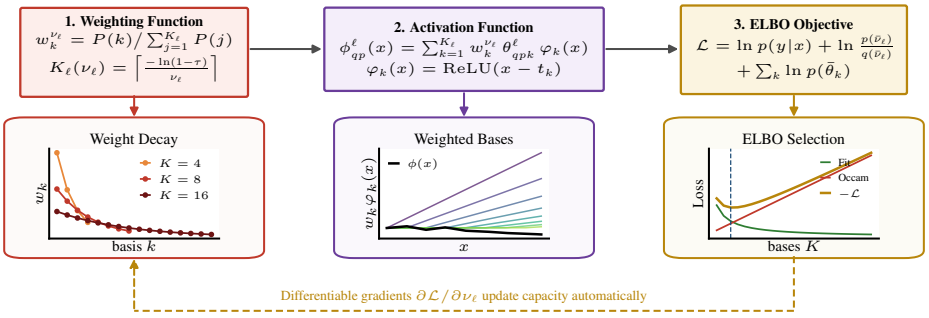


Figure 1: INFINITYKAN dynamically selects layer capacity using: (1) a **weighting function** $w_k^{\nu_\ell}$ defining basis truncation K_ℓ, (2) an **activation function** $\phi_{qp}^{\ell}(x)$ composed of weighted generative bases, and (3) an **ELBO loss** that penalizes model complexity using continuous gradients $\partial \mathcal{L} / \partial \nu_\ell$.

choosing the family of basis functions, the number of basis functions to use is also unknown *a priori*, and an inappropriate choice can substantially degrade a KAN's representational capacity.

We therefore present (i) INFINITYKAN which models the univariate functions using an adaptive and potentially infinite number of bases (Section 3). INFINITYKAN handles the unbounded number of bases in a way that provides gradient information for it to be updated. The model's design stems from a variational treatment of an intractable maximum likelihood learning problem. We provide (ii) theoretical evaluation of the smoothness of the ELBO and expectation approximations (Sections A to E), and (iii) experimental validation on various datasets (Sections 4 and 5).

2 RELATED WORKS

Recent work (Lai & Shen, 2021) has expanded on KAT foundations, exploring the capabilities of KAN-based models in high-dimensional spaces and their potential to mitigate the curse of dimensionality (Poggio, 2022). Various KAN architectures have been proposed: KAN has been combined with Convolutional Neural Networks (CNNs) (Ferdaus et al., 2024), or with transformer models (Yang & Wang, 2024), leading to improved efficiency in sequence modeling tasks. Furthermore, EKAN incorporates matrix group equivariance (Hu et al., 2024) into KANs, while GKSN (Alesiani et al., 2025) explores the extension to invariant and equivariant functions to model physical and geometrical symmetries. KANs have demonstrated their versatility across a wide spectrum of machine learning applications (Somvanshi et al., 2024), particularly in scenarios demanding efficient (i.e., small number of parameters) function approximation with a limited parameter budget. Their effectiveness in high-dimensional regression problems, where traditional neural networks often face scalability issues, was notably demonstrated by Kůrková (1992). Adaptive architectures have been proposed for MLP models. For example, (Fahlman & Lebiere, 1989) extends the network with an additional hidden units as the end of a training phase, while firefly network descent (Wu et al., 2020) grows the width and depth of a neural network during training. In continual learning (Yoon et al., 2018), network models are updated based on new tasks, or neurons are duplicated or removed according to heuristics to create more capacity (Qiang Liu, 2019; Mitchell et al., 2023). Variational approaches have been used for adaptive architectures to select the number of layers or neurons in neural networks (Nazaret & Blei, 2022; Errica et al., 2025b), or for graph neural networks (Errica et al., 2025a). However, these approaches are not trivially applicable for learning the number of bases in edge functions, since the output is not additive in KAN models.

3 INFINITE KOLMOGOROV-ARNOLD NETWORK

We first recap the definition of a KAN layer before introducing our extension to learn an unbounded number of basis functions. Figure 1 provides an overview of INFINITYKAN.

KAT, KAN layer and basis functions According to the KAT theorem, a generic continuous d-dimensional multivariate function $f(x_1, \ldots, x_d) : \mathbb{R}^d \to \mathbb{R}$ defined over a compact space is represented as a composition of continuous univariate functions as

$$f(x_1, \ldots, x_d) = \sum_{q=1}^{2d+1} \phi'_q \left(\sum_{p=1}^{d} \phi_{qp}(x_p) \right) \tag{1}$$

with $\boldsymbol{x} = (x_1, \ldots, x_d) \in [0,1]^d$ and where $\phi_{qp}, \phi'_q : \mathbb{R} \to \mathbb{R}$ are continuous univariate functions. However, the KAN is composed of L KAN layers, where each layer $\ell \in \{1, \ldots, L\}$ implements the mapping from $[0,1]^{d_{\ell-1}} \to [0,1]^{d_\ell}$, where $d_{\ell-1}$ and d_ℓ are the layer input and output dimensions, using the univariate functions $\boldsymbol{\phi}^\ell = \{\phi^\ell_{qp} : \mathbb{R} \to \mathbb{R}\}$. Each layer computes the hidden variables $\boldsymbol{x}^\ell = \{x^\ell_q \mid x^\ell_q = h^\ell_q(x^{\ell-1}_1, \ldots, x^{\ell-1}_{d_{\ell-1}}) = \sum_{p=1}^{d_{\ell-1}} \phi^\ell_{qp}(x^{\ell-1}_p), \forall q \in [d_\ell] = \{1, \ldots, d_\ell\}\}$, from previous layer outputs $\boldsymbol{x}^{\ell-1} = \{x^{\ell-1}_p, \forall p \in [d_{\ell-1}]\}$, i.e., $\boldsymbol{x}^\ell = \boldsymbol{\phi}^\ell(\boldsymbol{x}^{\ell-1})$. The KAT does not tell us how to find the univariate functions $\boldsymbol{\phi}^\ell$, but it is possible to build a convergent series for any uniformly continuous function $\phi(x)$ as a linear combination of other base functions $\varphi^n_k(x)$. Therefore, w.l.o.g. we represent each univariate function in a KAN layer ℓ as the limit of the linear combination of the basis functions $\varphi^n_k(x)$

$$\phi^\ell_{qp}(x) = \lim_{n \to \infty} \sum_{k=1}^{n} \theta^{\ell n}_{qpk} \varphi^n_k(x) \tag{2}$$

Therefore

$$\phi(x) = \lim_{n \to \infty} \phi^n(x), \quad \phi^n(x) = \sum_{k=1}^{n} \phi^n_k(x) = \sum_{k=1}^{n} \theta^n_k \varphi^n_k(x)$$

where $\varphi^n_k(x)$ can either be a Heaviside step function or a rectified linear unit (ReLU) function (Jarrett et al., 2009), as we show in Theorem A.2 and Theorem A.3, while θ^n_k are the parameters of the linear combination. In the following, we refer to $\varphi^n_k(x)$ as the generative functions of the basis $\phi^n_k(x)$.

The intuition behind our contribution is that we would like to *learn* using a **finite** number of basis n for each layer that is powerful enough for the task at hand, and therefore training on the finite set of parameters $\{\theta^{\ell n}_{qpk}\}_{k \in [n]}$, where $[n] = \{1, \ldots, n\}$ for each layer ℓ, using the simple and efficient back-propagation mechanism.

Variational Training Objective Bayesian variational inference approximates an intractable target posterior distribution p by optimizing the parameters of a tractable variational distribution q to minimize their divergence. We consider a regression or a classification problem and the corresponding dataset $\mathcal{D}$ composed of i.i.d. samples $(\boldsymbol{X}, \boldsymbol{Y}) = \{(x_i, y_i)\}_{i=1}^D$, with $x_i \in \mathbb{R}^d$, and $y_i \in \mathbb{R}^{d'}$. If we build a probabilistic model implementing the posterior distribution $p(\boldsymbol{Y}|\boldsymbol{X})$, the objective corresponds to maximize the dataset log-likelihood

$$\mathcal{L}(\mathcal{D}) = \ln p(\boldsymbol{Y}|\boldsymbol{X}) = \sum_{i=1}^{D} \ln p(y_i|x_i). \tag{3}$$

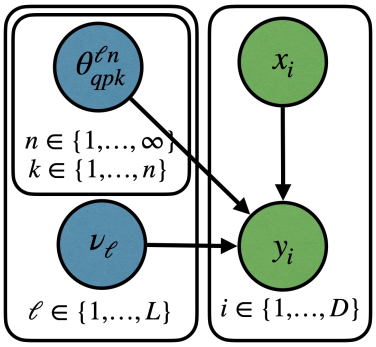


Figure 2: The graphical model of INFINITYKAN, with the observable variables (in green) x_i, y_i and latent variables (in blue) $\theta^{\ell n}_{qpk}, \nu^\ell$.

If we modeled the probability distribution with a multi-layer KAN network, we would need to optimize Equation (3) with respect to the set of continuous univariate functions $\boldsymbol{\phi}^\ell = \{\phi^\ell_{qp}\}$. However, based on Equation (2), we first introduce an infinite-dimensional family of KANs. Because the right value n for each layer is unknown, we introduce two latent variables that parameterize such a family. First, each layer has a set of latent variables $\boldsymbol{\theta}^\ell = \{\boldsymbol{\theta}^{\ell n}\}_{n=1}^\infty = \{\theta^{\ell n}_{qpk}, k \in [n]\}_{n=1}^\infty$ (see Equation (2)), with $\theta^{\ell n}_{qpk}$ is itself a multivariate variable over the learnable weights of the k-th basis function at layer ℓ and for the qp univariate function. We further introduce a latent variable ν^ℓ, the rate parameter of a

truncated exponential distribution, similar to (Errica et al., 2025b), that controls the number of basis functions n used at layer ℓ. For a KAN of L layers, we define $\boldsymbol{\theta} = \left\{\boldsymbol{\theta}^{\ell}\right\}_{\ell\in[L]}$ and $\boldsymbol{\nu} = \{\nu^{\ell}\}_{\ell\in[L]}$ and we assume independence across all layers, which allows us to write $p(\boldsymbol{Y}|\boldsymbol{X}) = \int d\boldsymbol{\theta} d\boldsymbol{\nu} p(\boldsymbol{Y}, \boldsymbol{\theta}, \boldsymbol{\nu}|\boldsymbol{X})$. Similar to (Nazaret & Blei, 2022), we now assume that $\boldsymbol{\theta}, \boldsymbol{\nu}$ are independent, i.e., $p(\boldsymbol{\theta}, \boldsymbol{\nu}) = p(\boldsymbol{\theta})p(\boldsymbol{\nu})$ and, based on the graphical model of Figure 2, we write the following distributions

$$p(\boldsymbol{Y}, \boldsymbol{\theta}, \boldsymbol{\nu}|\boldsymbol{X}) = p(\boldsymbol{Y}|\boldsymbol{\theta}, \boldsymbol{\nu}, \boldsymbol{X})p(\boldsymbol{\theta})p(\boldsymbol{\nu}), \quad p(\boldsymbol{\nu}) = \prod_{\ell=1}^{L} p(\nu_\ell) = \prod_{\ell=1}^{L} \mathrm{Exp}(\nu_\ell; \eta_\ell), \tag{4}$$

$$p(\boldsymbol{\theta}) = \prod_{\substack{\ell\in[L],\\ n=1,\dots,\infty, k\in[n],\\ q\in[d_\ell], p\in[d_{\ell-1}]}} p(\theta_{qpk}^{\ell n}), \quad p(\theta_{qpk}^{\ell n}) = \mathcal{N}(\theta_{qpk}^{\ell n}; \mathbf{0}, \mathrm{diag}\,(\sigma_\ell)), \tag{5}$$

where we assume that prior on the rate parameter ν of the truncated exponential distribution follows an exponential distribution (i.e., $\mathrm{Exp}(\nu; \eta)$), and we further assume that the weights of the basis follow a Gaussian distribution (i.e., $\mathcal{N}(\theta; \mu, \sigma)$). The predictive model $p(\boldsymbol{Y}|\boldsymbol{\theta}, \boldsymbol{\nu}, \boldsymbol{X})$ is based on the KAN architecture and is described later. The distributions depend on the prior's hyper-parameters $\boldsymbol{\eta} = \{\eta_\ell\}$ and $\boldsymbol{\sigma} = \{\sigma_\ell\}$, while the KAN is parametrized by $\boldsymbol{\theta}$ and $\boldsymbol{\nu}$. Maximizing directly Equation (3) would require computing an intractable integral, therefore, we apply the mean-field variational inference approach (Jordan et al., 1999), which entails maximizing the expected lower bound (ELBO). By introducing a learnable variational distribution $q(\boldsymbol{\theta}, \boldsymbol{\nu})$ and using the concavity of the logarithmic function, write the objective as (see Section B for the derivation)

$$\ln p(\boldsymbol{Y}|\boldsymbol{X}) \geq \mathbb{E}_{q(\boldsymbol{\nu},\boldsymbol{\theta})}\left[\ln \frac{p(\boldsymbol{Y}, \boldsymbol{\nu}, \boldsymbol{\theta}|\boldsymbol{X})}{q(\boldsymbol{\nu}, \boldsymbol{\theta})}\right] \tag{6}$$

Using the same intuition from (Nazaret & Blei, 2022), we then assume that the variational distribution can be written by conditioning on the number of basis, as

$$q(\boldsymbol{\theta}, \boldsymbol{\nu}) = q(\boldsymbol{\theta}|\boldsymbol{\nu})q(\boldsymbol{\nu}), \quad q(\boldsymbol{\nu}) = \prod_{\ell=1}^{L} q(\nu_\ell) = \prod_{\ell=1}^{L} \mathrm{TruncExp}(\nu_\ell; \bar{\nu}_\ell) \tag{7}$$

$$q(\boldsymbol{\theta}|\boldsymbol{\nu}) = \prod_{\substack{\ell\in[L],\\ n=K_\ell,\\ k\in[K_\ell],\\ q\in[d_\ell], p\in[d_{\ell-1}]}} q(\theta_{qpk}^{\ell n}) \prod_{\substack{\ell\in[L],\\ n=1,\dots,\infty, n\neq K_\ell\\ k\in[n]\\ q\in[d_\ell], p\in[d_{\ell-1}]}} p(\theta_{qpk}^{\ell n}), \quad q(\theta_{qpk}^{\ell n}) = \mathcal{N}(\theta_{qpk}^{\ell n}; \bar{\theta}_{qpk}^{\ell n}, \boldsymbol{I}), \tag{8}$$

where $\bar{\nu}_\ell$ and $\bar{\theta}_{qpk}^{\ell n}$ are the variational parameters to be learned, while K_ℓ is the truncation number computed from the rate parameter ν_ℓ at a specified quantile threshold τ (see Section 3), representing the current number of basis functions at layer ℓ.

By modeling the distribution of the parameters belonging to a different function in the infinite series with the same *a priori* distribution p, its influence on the maximization problem is removed. While we could model the variance of the basis's coefficients with additional trainable parameters, in the following, we see how the variance is ignored. We can now write the final objective by using the previous assumptions and the first-order approximation of the expectation, i.e., $\mathbb{E}_{q(\boldsymbol{\nu};\bar{\boldsymbol{\nu}})}[f(\boldsymbol{\nu})] = f(\bar{\boldsymbol{\nu}})$, and $\mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\nu};\bar{\boldsymbol{\theta}})}[f(\boldsymbol{\theta})] = f(\bar{\boldsymbol{\theta}})$, (see Section E) in Equation (6),

$$\overbrace{\sum_{i=1}^{D} \ln p(y_i|\boldsymbol{\nu} = \bar{\boldsymbol{\nu}}, \boldsymbol{\theta} = \bar{\boldsymbol{\theta}}, x_i)}^{\text{Divergence over data}} + \overbrace{\sum_{\ell=1}^{L} \ln \frac{p(\bar{\nu}_\ell; \eta_\ell)}{q(\bar{\nu}_\ell; \bar{\nu}_\ell)}}^{\text{Divergence over bases}} + \overbrace{\sum_{\substack{k\in[K_\ell],\\ q\in[d_\ell], p\in[d_{\ell-1}]}} \ln p(\bar{\theta}_{qpk}^{\ell K_\ell}; \mathbf{0}, \mathrm{diag}\,(\sigma^\ell))}^{\text{Divergence over parameters}}, \tag{9}$$

where we remove the constant term arising from the evaluation of q distribution at its mean value, i.e., $q(\bar{\theta}_{qpk}^{\ell K_\ell}) = \mathcal{N}(\bar{\theta}_{qpk}^{\ell K_\ell}; \bar{\theta}_{qpk}^{\ell K_\ell}, \boldsymbol{I}) = \text{const}$ and σ_ℓ, η_ℓ are the priors' hyper-parameters (see derivation in Equation (33)). While using second-order approximation seems attractive and may lead to a tighter ELBO, it would require to compute second order derivaties, therefore increasing the computation cost of the proposed method excessively (see Equation (34)). Equipped with Equation (9), we can now train the basis parameters $\bar{\boldsymbol{\theta}}$ and the rate parameters $\bar{\boldsymbol{\nu}}$ using a standard optimization algorithm based

on stochastic gradient descent. The Equation (6) contains discrete variables, the number of basis functions. We are therefore faced with two problems: 1) how the gradient propagates, and 2) whether the function is continuous to allow the use of stochastic gradient descent algorithms. We resolve the first question in Section 3, while we provide the following statement for the second, proved in Section C,

ELBO Lipschitz continuity

Theorem 3.1. *The ELBO loss of Equation* (9)*, with respect to the change in the number of basis K_ℓ (or ν^ℓ) for the layer ℓ, is Lipschitz continuous.*

Proof Sketch. We isolate the terms of the ELBO in Equation (9) that depend on the number of basis functions $\mathbf{K} = \{K_\ell\}_{\ell=1}^{L}$. The objective splits into a variational likelihood term, $f_2(\mathbf{K}) = \ln p(\boldsymbol{Y}|\bar{\boldsymbol{\nu}}, \bar{\boldsymbol{\theta}}, \boldsymbol{X})$, and the prior-to-proposal KL-divergence terms, which we group as: $f_1(K_\ell) = \sum_{k=1}^{K_\ell} \ln \frac{p(\bar{\boldsymbol{\theta}}_k^\ell)}{q(\bar{\boldsymbol{\theta}}_k^\ell|\nu_\ell)}$. We can then show that the Lipschitz condition holds for both of these terms separately. Combining these results, the total ELBO loss is a sum of Lipschitz continuous functions, preserving Lipschitz continuity overall. See Section C for the full derivation. □

Remark 3.2 (Piecewise Lipschitz in ν)**.** *Because $K_\ell(\nu_\ell) = \lceil -\ln(1-\tau)/\nu_\ell \rceil$ is piecewise constant, the ELBO is* piecewise *Lipschitz continuous in ν_ℓ (Lemma D.7): within each interval where K_ℓ is fixed the map is smooth via $w_k^{\nu_\ell}$, while jumps across intervals are controlled by Theorem D.2.*

The Weighting Function for the Basis We now introduce the KAN-based model that implements the prediction model $p(\boldsymbol{Y}|\boldsymbol{\nu} = \bar{\boldsymbol{\nu}}, \boldsymbol{\theta} = \bar{\boldsymbol{\theta}}, \boldsymbol{X})$, given the data samples $\boldsymbol{X}$ and the variational parameters $\{\boldsymbol{\nu}, \boldsymbol{\theta}\}$. The truncation number K_ℓ determines how many basis functions to use at layer ℓ, and is computed from the rate parameter ν_ℓ at a specified quantile threshold τ. To provide gradient information for learning ν_ℓ, we introduce a *weighting function* $\boldsymbol{w}(\boldsymbol{\nu}) = \{w_k^{\nu_\ell}\}_{\ell=1}^{L}$ parametrized by $\boldsymbol{\nu}$ that multiplies the basis coefficients, and write the edge function as

$$\phi_{qp}^\ell(x) = \sum_{k=1}^{K_\ell(\nu_\ell)} w_k^{\nu_\ell} \theta_{qpk}^{\ell K_\ell(\nu_\ell)} \varphi_k^{K_\ell(\nu_\ell)}(x) \quad (10)$$

The weights $w_k^{K_\ell}$ are computed as a renormalized probability vector from the truncated exponential distribution:

$$w_k^{\nu_\ell} = \frac{P(k)}{\sum_{j=1}^{K_\ell(\nu_\ell)} P(j)}, \quad P(k) = F(k+1) - F(k), \quad F(x) = 1 - e^{-\nu_\ell x} \quad (11)$$

Figure 3: INFINITYKAN Training Procedure

Input:
  $\mathcal{D}$: dataset
  $\mathcal{B}$: basis functions
Output: Trained INFINITYKAN Model $\mathcal{M}$
Initialize the basis $\mathcal{B}$ and rate parameters $\boldsymbol{\nu}$
for each training epoch **do**
  for (x, y) in $\mathcal{D}$ **do**
    for layer ℓ in $\mathcal{M}$.KAN_layers **do**
      $K_\ell \leftarrow \lceil -\ln(1-\tau)/\nu_\ell \rceil$ // compute truncation number for truncated exponential
      $w_k^{\nu_\ell} \leftarrow$ compute_weights(ν_ℓ, K_ℓ) // build weights using Equation (11)
      Update layer parameters if K_ℓ changed
    end for
    $\hat{y} \leftarrow \mathcal{M}(x)$
    loss $\leftarrow$ ELBO$(\mathcal{M}, x, \hat{y})$ // Equation (9)
    $\mathcal{M} \leftarrow$ back-propagation$(\mathcal{M}, \text{loss})$
  end for
end for

where $F(x)$ is the CDF of the exponential distribution with rate ν_ℓ, and $P(k)$ is the discretized probability mass at index k. The truncation number is computed as $K_\ell(\nu_\ell) = \lceil -\ln(1-\tau)/\nu_\ell \rceil$ for a specified quantile threshold τ (e.g., $\tau = 0.9$). While K_ℓ is not differentiable, K_ℓ is a deterministic function of ν_ℓ, therefore we propagete gradient using Equation (11). This formulation, indeed, ensures that the weights $w_k^{\nu_\ell}$ depend differentiably on ν_ℓ, enabling gradient-based optimization of the number of basis functions. We notice that by construction $\sum_{j=1}^{K_\ell(\nu_\ell)} w_k^{\nu_\ell} = 1$ (see Equation (11)). For completeness, we derive the analytic gradient in Section I. Beyond differentiability, we show that soft weighting is a controlled perturbation of hard truncation (Theorem D.1), that the ELBO jump when $K_\ell \mapsto K_\ell + 1$ admits an explicit Lipschitz constant in terms of coefficient and basis norms (Theorem D.2), and that the variational prior terms induce an Occam penalty (Jefferys & Berger, 1992; Rasmussen & Ghahramani, 2000) linear in the basis count (Theorem D.3).

Dataset	KAN	MLP	INFINITYKAN
CIFAR10	43.00 (2.05)	44.03 (1.04)	**46.99** (0.38)
CIFAR100	14.04 (0.76)	15.62 (0.40)	**19.27** (0.46)
DoubleMoon	98.43 (4.44)	97.73 (4.57)	**100.0** (0.00)
Electricity	84.51 (0.40)	82.05 (0.47)	**84.54** (0.49)
EuroSAT	68.89 (0.79)	62.61 (0.59)	**69.51** (0.32)
EyeMovements	**53.21** (1.15)	50.80 (0.85)	50.14 (2.11)
FashionMNIST	85.44 (1.30)	86.55 (0.20)	**86.84** (0.30)
House	**89.18** (0.36)	88.47 (0.43)	89.11 (0.25)
Jannis	**70.00** (0.28)	68.15 (0.52)	68.87 (0.57)
MNIST	96.02 (0.19)	95.33 (0.24)	**96.54** (0.20)
MagicTelescope	**88.77** (0.24)	87.75 (0.39)	88.18 (0.39)
MiniBooNE	94.12 (0.16)	94.19 (0.14)	**94.20** (0.10)
NCI1	79.30 (1.13)	**80.00** (1.40)	76.92 (2.27)
POL	99.20 (0.17)	99.25 (0.17)	**99.33** (0.16)
Phoneme	**87.39** (0.77)	85.99 (1.24)	87.21 (0.72)
REDDIT_BINARY	**91.70** (1.17)	89.90 (1.90)	83.60 (3.46)
Spiral	99.25 (0.45)	**99.99** (0.03)	99.92 (0.07)
SpiralHard	**99.89** (0.06)	98.88 (1.05)	98.49 (2.13)
Wins	7	2	**9**

Table 1: Comparison of KAN, MLP, and INFINITYKAN on 18 datasets. Results are reported as mean accuracy with standard deviation (in parentheses) over 10 runs. Bold indicates the best performance for each dataset.

	KAN			INFINITYKAN		
Dataset	L	d_emb	Total# bases	L	d_emb	Total # bases
CIFAR10	1	16	32	2	16	8.00 (0.00)
CIFAR100	1	16	2	2	16	23.70 (20.51)
DoubleMoon	1	2	2	1	2	6.00 (0.00)
Electricity	1	16	8	2	16	8.00 (0.00)
EuroSAT	2	16	4	2	16	9.10 (2.02)
EyeMovements	1	5	128	2	5	33.90 (51.26)
FashionMNIST	2	16	64	2	16	8.00 (0.00)
House	2	5	64	2	5	9.30 (0.90)
Jannis	2	16	16	2	16	8.00 (0.00)
MNIST	2	16	64	2	16	10.50 (2.20)
MagicTelescope	2	5	256	1	16	6.00 (0.00)
MiniBooNE	2	16	16	1	16	6.00 (0.00)
NCI1	5	16	10	5	16	14.00 (0.00)
POL	2	5	4	1	16	21.00 (32.02)
Phoneme	1	16	128	2	16	23.60 (3.41)
REDDIT_BINARY	2	16	16	2	16	88.90 (51.44)
Spiral	1	2	8	1	5	6.50 (0.67)
SpiralHard	1	5	8	1	5	9.40 (2.06)

Table 2: Comparison of selected hyperparameters and learned total number of bases for KAN and INFINITYKAN across datasets. For KAN, L is the number of layers, d_emb is the embedding dimension, and Total # bases is the fixed number of bases used. For INFINITYKAN, Total # bases is reported as mean (std) over 10 runs.

Updating Weights When the truncation number K_ℓ changes (due to changes in the rate parameter ν_ℓ), we update the parameter tensors accordingly. If the number of bases increases from n to $n' > n$, we extend the parameter tensors by initializing new entries with random values (using normal initialization). If the number of bases decreases from n to $n' < n$, we truncate the parameter tensors to keep only the first n' entries. This heuristic is a first-order approximation of the continuous update that preserves parameters for the overlapping indices.

4 EXPERIMENTAL VALIDATION

INFINITYKAN overcomes the limitation of manually selecting the number of basis functions for each layer of a KAN. Our experimental goals are twofold: (1) verify that the training procedure is stable, and (2) confirm that performance is competitive with KANs using a fixed, optimally-tuned number of bases. We focus mostly on classification tasks. In particular, we compare KAN, the propose variational variant (INFINITYKAN), and a standard Multi-Layer Perceptrons (MLP) across a broad range of datasets, spanning synthetic, tabular, image, time-series, and graph-like domains.

Datasets. We consider a total of 18 datasets. Synthetic classification tasks include DoubleMoon, Spiral, and SpiralHard (Errica et al., 2025b), which exhibit increasing levels of nonlinearity and decision-boundary complexity. Image classification benchmarks comprise MNIST, FashionMNIST, CIFAR10, CIFAR100, and EuroSAT (Helber et al., 2019), covering grayscale and color images with varying numbers of classes and degrees of classification complexity. Tabular and time-series datasets include Electricity, House, Jannis, MagicTelescope, MiniBooNE, Phoneme, POL, and EyeMovements (Beyazit et al., 2023), which vary substantially in dimensionality and sample size. Finally, we include graph-structured datasets NCI1 and REDDIT_BINARY (Morris et al., 2020), where node connectivity and graph topology are essential for classification.

Model selection and evaluation protocol. For all tabular datasets, we follow a 70%/10%/10% stratified hold-out split for risk assessment, where the validation set is used for early stopping. We perform model selection by further splitting the training set into 90% training and 10% validation sets. We reuse data splits when available. EuroSAT follows the same split percentages as MNIST. Risk assessment results, measured with accuracy, are reported as averages and standard deviations over ten independent runs with different random seeds. For graph datasets, we follow the same experimental setup of Errica et al. (2020) and apply each of the three architectures to a classical GIN baseline (Xu et al., 2019) to run the experiments.

We report the hyper-parameters tried for all models in Section K. For the experiments, we used a server with 64 cores, 1.5TB of RAM, and 2 NVIDIA A40 GPUs with 48GB of RAM.

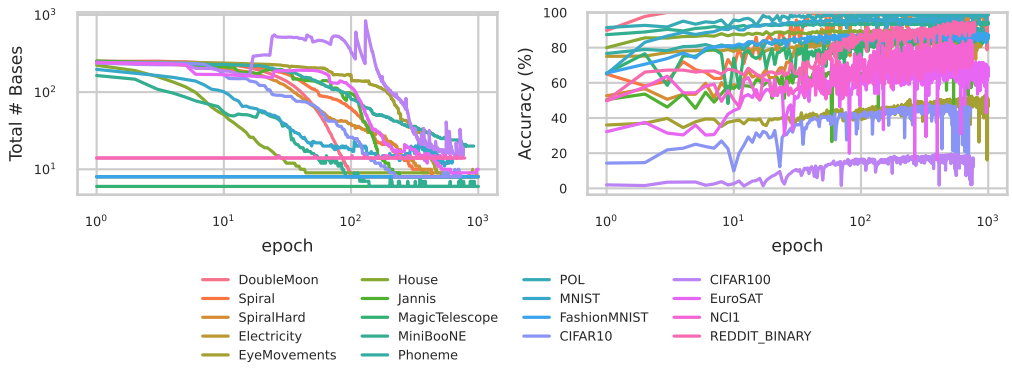


Figure 4: Convergence of INFINITYKAN during training across all 18 datasets. **Left:** Evolution of the total number of basis functions (width) over epochs. Most datasets converge to a stable width within the first 100 epochs, demonstrating that INFINITYKAN can efficiently identify the appropriate model complexity. **Right:** Test accuracy (score) convergence over epochs. The adaptive selection of basis functions does not hinder convergence, with most datasets reaching stable performance early in training.

5 RESULTS

In this section, we present the results to evaluate the ability of INFINITYKAN to automatically learn the number of bases, and whether the performance of these methods relates to the configuration selected using model selection.

Test accuracy. In Table 1 we observe that INFINITYKAN outperforms KAN and MLP on 9 out of 18 datasets, with notable improvements on CIFAR10 and CIFAR100. On 7 datasets (EyeMovements, Jannis, MagicTelescope, Phoneme, REDDIT_BINARY), KAN with a carefully tuned fixed number of bases achieves the best performance, while MLP wins on 2 datasets (NCI1, Spiral, SpiralHard).

Crucially, INFINITYKAN achieves this competitive performance *without* requiring the practitioner to tune the number of basis functions–a hyperparameter that KAN requires and that varies from 2 to 128 across datasets in our experiments (see Table 2). This represents a significant reduction in the hyperparameter search space, as the optimal number of bases is dataset-dependent and not known *a priori*. The results demonstrate that INFINITYKAN provides a principled alternative to manual tuning, achieving comparable or superior performance while eliminating an important degree of freedom in model selection.

Hyperparameters Table 2 presents the model-selected hyperparameters and learned total number of bases for KAN and INFINITYKAN across datasets. For KAN, we report the number of layers (L), embedding dimension (d_emb), and the fixed total number of bases used. For INFINITYKAN, we report the learned total number of bases as mean and standard deviation over 10 runs. We find that INFINITYKAN often learns a number of bases that is different from the fixed number used in KAN, demonstrating its ability to adaptively select the appropriate model complexity for each dataset. The variability in the learned number of bases across runs indicates that INFINITYKAN can explore different configurations during training, potentially leading to improved performance. This adaptive behavior highlights the advantage of INFINITYKAN in automatically determining the model capacity needed for effective learning, without requiring manual tuning of the number of basis functions.

For instance, in the CIFAR100 dataset, KAN uses only 2 bases, while INFINITYKAN learns an average of 23.7 bases with a high standard deviation, indicating significant variability across runs. In contrast, for the DoubleMoon dataset, both KAN and INFINITYKAN use a similar number of bases (2 for KAN and an average of 6 for INFINITYKAN), suggesting that a simpler model suffices for this task. These examples illustrate how INFINITYKAN can flexibly adjust its complexity based on the dataset characteristics, potentially leading to better generalization and performance. In some instances, such as Electricity and FashionMNIST, INFINITYKAN converges to the same number of bases (8) across all runs, indicating a stable selection process for these datasets. This consistency

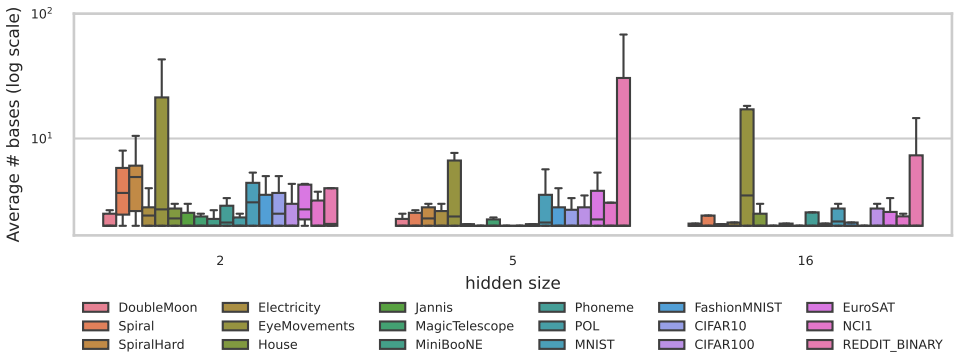


Figure 5: Average number of basis functions learned by INFINITYKAN across different hidden sizes (2, 5, and 16) for all datasets. The y-axis is shown on a log scale. Error bars indicate standard deviation over 10 runs. The learned number of bases varies across datasets and hidden sizes, with some datasets (e.g., EyeMovements, REDDIT_BINARY) exhibiting higher variability, while others (e.g., DoubleMoon, Electricity) converge consistently to lower values.

suggests that for certain tasks, there may be an optimal model complexity that INFINITYKAN can reliably identify. However, in other cases like REDDIT_BINARY, INFINITYKAN exhibits high variability in the learned number of bases (mean of 88.9 with a standard deviation of 51.44), reflecting the complexity and diversity of the dataset. This variability may allow INFINITYKAN to explore a wider range of model configurations, potentially leading to improved performance on challenging tasks.

Training dynamics. Figure 4 illustrates the convergence behavior of INFINITYKAN during training across all 18 datasets. The left panel shows the evolution of the total number of basis functions (width) over epochs, while the right panel displays the test accuracy (score) convergence. We observe that for most datasets, the total number of basis functions stabilizes within the first 100 epochs, indicating that INFINITYKAN can efficiently identify the appropriate model complexity early in training. This rapid convergence suggests that the adaptive mechanism effectively guides the model towards a suitable configuration without excessive exploration. The right panel shows that the test accuracy also converges steadily over epochs, with most datasets reaching stable performance early in training. This indicates that the adaptive selection of basis functions does not hinder convergence, and INFINITYKAN can achieve competitive accuracy while dynamically adjusting its complexity.

Number of bases. Figure 5 presents the average number of basis functions learned by INFINITYKAN across different hidden sizes (2, 5, and 16) for all datasets. The y-axis is shown on a logarithmic scale to better visualize the range of learned bases. Error bars indicate the standard deviation over 10 runs. We observe that the learned number of bases varies significantly across datasets and hidden sizes. For instance, datasets like EyeMovements and REDDIT_BINARY exhibit higher variability in the number of learned bases, suggesting that these tasks may require more complex representations and that INFINITYKAN explores a wider range of configurations. In contrast, datasets such as DoubleMoon and Electricity consistently converge to lower values of learned bases, indicating that simpler models are sufficient for these tasks. The results highlight the flexibility of INFINITYKAN in adapting its complexity based on the dataset characteristics and the chosen hidden size. This adaptability allows INFINITYKAN to effectively balance model capacity and generalization, leading to improved performance across diverse tasks.

Model selection. To show the advantage of INFINITYKAN, Table 3 shows the training time of INFINITYKAN against the model selection of KAN on the number of basis. From the table, we see that INFINITYKAN converges faster and, in most cases, achieves a similar accuracy to the baseline model. From the lower part of Table 3, with longer training time for INFINITYKAN, we have better accuracy, except for REDDIT_BINARY.

Dataset	% Time diff. (↑)	% Accuracy diff. (↑)	% Bases diff. (↑)	% Params diff. (↑)
DoubleMoon	96.92	1.60	0.00	-41.67
POL	94.72	0.13	-250.00	-625.46
Electricity	89.93	0.03	75.00	50.24
MagicTelescope	85.23	-0.66	98.44	91.55
EyeMovements	83.46	-5.76	93.38	91.73
House	83.42	-0.08	92.72	89.51
SpiralHard	82.15	-1.40	60.88	57.37
MiniBooNE	80.85	0.08	75.00	74.94
Jannis	78.33	-1.61	75.00	69.21
EuroSAT	75.15	0.91	-14.00	-0.18
Phoneme	73.91	-0.21	95.39	92.20
Spiral	60.08	0.68	72.88	-22.54
NCI1	15.89	-3.00	94.24	92.97
CIFAR10	-12.79	9.28	93.75	91.79
REDDIT_BINARY	-16.44	-8.83	76.16	31.13
CIFAR100	-50.58	37.25	-196.00	-18.97
FashionMNIST	-98.03	1.64	93.75	91.84
MNIST	-115.50	0.54	91.81	86.57

Table 3: Automatic **model selection** capability is shown by comparing INFINITYKAN against KANwith model selection on the number of bases. In most of the datasets, INFINITYKAN converges faster (↑ higher is better for all four columns, where "diff" is the relative difference $(a-b)/b$, with b the value for KAN) to a better solution. With INFINITYKAN, the number of bases is learned end-to-end during training. The model selection only sweeps across a small predefined number of initial bases for each layer. The table shows the flexibility introduced by INFINITYKAN. Interestingly, excluding one case, even when the training time is larger, the model possesses better overall accuracy. Datasets are ordered by completion time difference; the lower part shows the datasets that took longer to train with INFINITYKAN. INFINITYKAN, consistently with Table 1, wins 10 times over 8 for the KAN in accuracy.

6 CONCLUSIONS AND FUTURE DIRECTIONS

We have presented INFINITYKAN, a variational inference method for training Kolmogorov-Arnold Networks with an adaptive number of basis functions per layer. By formulating basis function selection as a probabilistic inference problem, INFINITYKAN automatically determines appropriate model complexity during training, removing this critical hyperparameter from model selection.

We establish Lipschitz continuity of the ELBO with respect to the basis count, ensuring stable optimization, via a differentiable weighting function derived from a truncated exponential distribution. Complementary results show that soft weighting approximates hard truncation with an explicit error, that ELBO jumps at $K \mapsto K+1$ admit a quantitative bound, and that the Gaussian coefficient prior induces an Occam penalty linear in K.

Across 18 datasets spanning synthetic, image, tabular, and graph domains, InfinityKAN matches or exceeds KAN and MLP baselines ($9/18$ wins), with notable gains on CIFAR10, CIFAR100, and EuroSAT. While in some cases the accuracy margin is small, INFINITYKAN often leads to a reduced number of bases, parameters, or overall computation time. The learned basis count varies with task complexity, e.g., converging to 8 for Electricity/FashionMNIST versus over 80 for REDDIT_BINARY, and typically stabilizes within the first 100 epochs.

INFINITYKAN introduces two hyperparameters (priors on the rate and coefficient distributions), but ablations show these are less sensitive than the basis count itself and become unimportant in the large-data limit. Training incurs roughly 2× overhead from dynamic tensor resizing, offset by eliminating the basis-count search in model selection.

Future work includes tighter ELBO bounds (second-order or importance-weighted approximations), formal convergence guarantees connecting to neural architecture search and Bayesian model selection, and extending the framework to convolutional, recurrent, and transformer-based architectures, and studying activation function adaptation in neural networks.

AI USE STATEMENT

We used generative AI tools to assist with LaTeX formatting and limited copy-editing of the manuscript. We did not use generative AI tools to generate research ideas, main theoretical claims, experimental results, plots, or code that produced the reported numbers. All scientific content was written, checked, and is the responsibility of the authors.

ETHICS STATEMENT

This work studies a variational training method for Kolmogorov–Arnold networks. INFINITYKAN's main practical effect is reducing compute spent on hyperparameter search for the basis count, which can lower the energy footprint of training KAN-based models relative to manual tuning. We are not aware of other specific societal risks beyond those common to machine learning research. The paper does not involve human-subject studies, new dataset releases, or dual-use applications.

REPRODUCIBILITY STATEMENT

The method is specified in the main text, including the generative model, variational objective, and the first-order approximation used in training. Complete proofs and additional derivations are in the appendix (Theorems, ELBO derivation, Lipschitz continuity, soft-vs-hard truncation, jump bounds, Occam penalty, and related sections). Dataset splits, hyper-parameters, and ablation studies are documented in the experimental section and the appendix. Source code will be released with the camera-ready version.

ACKNOWLEDGMENTS

Acknowledgments, including funding, will be included in the camera-ready version.

REFERENCES

Francesco Alesiani, Takashi Maruyama, Henrik Christiansen, and Viktor Zaverkin. Geometric Kolmogorov-Arnold Superposition Theorem. *arXiv:2502.16664*, 2025.

Ege Beyazit, Jonathan Kozaczuk, Bo Li, Vanessa Wallace, and Bilal Fadlallah. An inductive bias for tabular deep learning. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Zavareh Bozorgasl and Hao Chen. Wav-KAN: Wavelet kolmogorov-arnold networks. *arXiv:2405.12832*, 2024.

Gianluca De Carlo, Andrea Mastropietro, and Aris Anagnostopoulos. Kolmogorov-Arnold graph neural networks. *arXiv:2406.18354*, 2024.

Federico Errica, Marco Podda, Davide Bacciu, and Alessio Micheli. A fair comparison of graph neural networks for graph classification. In *8th International Conference on Learning Representations (ICLR)*, 2020.

Federico Errica, Henrik Christiansen, Viktor Zaverkin, Takashi Maruyama, Mathias Niepert, and Francesco Alesiani. Adaptive message passing: A general framework to mitigate oversmoothing, oversquashing, and underreaching. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025a.

Federico Errica, Henrik Christiansen, Viktor Zaverkin, Mathias Niepert, and Francesco Alesiani. Adaptive width neural networks. *arXiv:2501.15889*, 2025b.

Scott Fahlman and Christian Lebiere. The cascade-correlation learning architecture. In *Proceedings of the 3rd Conference on Neural Information Processing Systems (NIPS)*, 1989.

Md Meftahul Ferdaus, Mahdi Abdelguerfi, Elias Ioup, David Dobson, Kendall N. Niles, Ken Pathak, and Steven Sloan. KANICE: Kolmogorov-Arnold Networks with Interactive Convolutional Elements, October 2024.

Leonardo Ferreira Guilhoto and Paris Perdikaris. Deep learning alternatives of the kolmogorov superposition theorem. In *The Thirteenth International Conference on Learning Representations*, 2025.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Lexiang Hu, Yisen Wang, and Zhouchen Lin. Incorporating arbitrary matrix group equivariance into KANs. *arXiv:2410.00435*, 2024.

Aysu Ismayilova and Vugar E Ismailov. On the kolmogorov neural networks. *Neural Networks*, pp. 106333, 2024.

Kevin Jarrett, Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun. What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th international conference on computer vision*, pp. 2146–2153. IEEE, 2009.

William H Jefferys and James O Berger. Ockham's razor and bayesian analysis. *American scientist*, 80(1):64–72, 1992.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37:183–233, 1999.

H. Katsura and D. A. Sprecher. Computational aspects of Kolmogorov's superposition theorem. *Neural Networks*, 7:455–461, 1994.

Andrei Nikolaevich Kolmogorov. *On the representation of continuous functions of several variables by superpositions of continuous functions of a smaller number of variables*. American Mathematical Society, 1961.

Mario Köppen. On the training of a kolmogorov network. In *Artificial Neural Networks—ICANN 2002: International Conference Madrid, Spain, August 28–30, 2002 Proceedings 12*, pp. 474–479. Springer, 2002.

Vladik Kreinovich, Hung T. Nguyen, and David A. Sprecher. Normal Forms For Fuzzy Logic — An Application Of Kolmogorov's Theorem. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 04(04):331–349, 1996.

V. Kůrková. Kolmogorov's theorem is relevant. *Neural Computation*, 3:617–622, 1991.

Věra Kůrková. Kolmogorov's theorem and multilayer neural networks. *Neural networks*, 5(3): 501–506, 1992.

Miklós Laczkovich. A superposition theorem of Kolmogorov type for bounded continuous functions. *Journal of Approximation Theory*, 269:105609, 2021.

Ming-Jun Lai and Zhaiming Shen. The Kolmogorov superposition theorem can break the curse of dimensionality when approximating high dimensional functions. *arXiv:2112.09963*, 2021.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, and Max Tegmark. KAN: Kolmogorov-Arnold Networks. *arXiv:2404.19756*, 2024.

Edward James McShane. Extension of range of functions. *Bulletin of the American Mathematical Society*, 40(12):837–842, 1934.

Rupert Mitchell, Martin Mundt, and Kristian Kersting. Self expanding neural networks. *arXiv:2307.04526*, 2023.

Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020. URL `www.graphlearning.io`.

Farinaz Mostajeran and Salah A Faroughi. EPi-cKANs: Elasto-plasticity informed kolmogorov-arnold networks using chebyshev polynomials. *arXiv:2410.10897*, 2024.

Achille Nazaret and David Blei. Variational inference for infinitely deep neural networks. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.

Tomaso Poggio. How deep sparse networks avoid the curse of dimensionality: Efficiently computable functions are compositionally sparse. *CBMM Memo*, 10:2022, 2022.

Dilin Wang Qiang Liu, Lemeng Wu. Splitting steepest descent for growing neural architectures. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Carl Rasmussen and Zoubin Ghahramani. Occam's razor. *Advances in neural information processing systems*, 13, 2000.

Shriyank Somvanshi, Syed Aaqib Javed, Md Monzurul Islam, Diwas Pandit, and Subasish Das. A Survey on Kolmogorov-Arnold Network. *arXiv:2411.06078*, 2024.

D. A. Sprecher. A numerical implementation of Kolmogorov's superpositions. *Neural Networks*, 9: 765–772, 1996.

D. A. Sprecher. A numerical implementation of Kolmogorov's superpositions II. *Neural Networks*, 10:447–457, 1997.

Sidharth SS, Keerthana AR, Gokul R, and Anas KP. Chebyshev polynomial-based Kolmogorov-Arnold networks: An efficient architecture for nonlinear function approximation. *arXiv:2405.07200*, 2024.

Lemeng Wu, Bo Liu, Peter Stone, and Qiang Liu. Firefly neural architecture descent: a general approach for growing neural networks. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

Jinfeng Xu, Zheyu Chen, Jinze Li, Shuo Yang, Wei Wang, Xiping Hu, and Edith C. H. Ngai. FourierKAN-GCF: Fourier Kolmogorov-Arnold network – an effective and efficient feature transformation for graph collaborative filtering. *arXiv:2406.01034*, 2024a.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations (ICLR)*, 2019.

Kunpeng Xu, Lifei Chen, and Shengrui Wang. Are KAN effective for identifying and tracking concept drift in time series? *arXiv:2410.10041*, 2024b.

Xingyi Yang and Xinchao Wang. Kolmogorov-Arnold Transformer. *arXiv:2409.10594*, 2024.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *6th International Conference on Learning Representations (ICLR)*, 2018.

VARIATIONAL KOLMOGOROV-ARNOLD NETWORK - SUPPLEMENTARY MATERIAL

A THEOREMS, PROOFS, AND DEFINITIONS

Definition A.1. (Uniformly continuous function) f is uniformly continuous function on X, metric space, if $\forall \epsilon > 0$, $\exists \delta > 0$ such that $\forall x, y \in X$ and $|x - y| < \delta$, we have that $|f(x) - f(y)| < \epsilon$.

While the convergence analysis is based on standard mathematical analysis tools, we provide in the following two special cases.

Convergence of step and piecewise functions

Theorem A.2. *Let's $f \in C([a, b] = [-1, 1], X)$ uniformly continuous on the metric space X, and f_n the sequence of step functions, such that*

$$f_n(t) = f(t_k^n),\ t \in [t_k^n, t_{k+1}^n),\ k = 1, \dots, n$$

or a piece-wise linear function, such that

$$f_n(t) = f(t_k^n)(1 - s) + sf(t_{k+1}^n),$$

with $t \in [t_k^n, t_{k+1}^n),\ k = 1, \dots, n$ and $t_1 = a = -1 \leq t_k \leq t_{k+1} \leq t_n = 1 = b$, with $s = t - t_k^n$. Then f_n converges to f.

Representation with piecewise linear and Relu functions

Theorem A.3. *Any piecewise linear function can be represented as a linear combination of ReLU functions, $g(t) = \max\{0, x\}$, and any uniformly continuous function can be the limit of a sequence of combinations of ReLU functions.*

Proof. (Theorem A.2) Since X is a metric space, f_n converges uniformly to f iff

$$\forall \epsilon > 0,\ \exists N \in \mathbb{N},\ \forall n \geq N : \|f_n - f\|_\infty < \epsilon$$

Let's take an $\epsilon > 0$ and the corresponding δ for uniformly continuity of f, and choose N such that $(b - a)/N = 2/N \leq \delta$, then for $n \geq N$ we have

$$|f(t_k^n) - f(t)| < \epsilon$$

and

$$|f(t) - f(t_{k+1}^n)| < \epsilon$$

for $t \in [t_k^n, t_{k+1}^n)$ and $\|f_n - f\|_\infty \leq \epsilon$. □

Proof. (Theorem A.3) Following Theorem A.2, we consider the segment $[t_k^n, t_{k+1}^n)$, and

$$f_n(t) = f(t_k^n)(1 - s) + sf(t_{k+1}^n),\ t \in [t_k^n, t_{k+1}^n),\ k = 1, \dots, n,$$

with $s = t - t_k^n$, then

$$f_n(t) = f(t_k^n) + g(t - t_k^n)\frac{f(t_{k+1}^n) - f(t_k^n)}{\delta_k^n}\ t \in [t_k^n, t_{k+1}^n),$$

with $\delta_k^n = t_{k+1}^n - t_k^n$, and $g(t) = \max\{0, t\}$ the relu function. When we stick together the linear functions, we need to remove the contribution of the previous relu functions in the form of $-\alpha_k^n g(t - t_k^n)$ with $\alpha_k^n = -\frac{f(t_k^n) - f(t_{k-1}^n)}{\delta_{k-1}^n}$. Writing in as a single equation

$$f_n(t) = f(t_1^n) + \sum_{k=1}^{n} [g(t - t_k^n) - g(t - t_{k+1}^n)]\frac{[f(t_{k+1}^n) - f(t_k^n)]}{\delta_k^n}$$

□

B ELBO DERIVATION

Here, we want to formalize the ELBO derivation.

Variational ELBO

Theorem B.1. *Given the assumptions on log likelihood and the variational distribution $q(\boldsymbol{\nu}, \boldsymbol{\theta})$ from Section 3, we have the following ELBO:*

$$\ln p(\boldsymbol{Y}|\boldsymbol{X}) \geq \mathbb{E}_{q(\boldsymbol{\nu},\boldsymbol{\theta})}\left[\ln \frac{p(\boldsymbol{Y}, \boldsymbol{\nu}, \boldsymbol{\theta}|\boldsymbol{X})}{q(\boldsymbol{\nu}, \boldsymbol{\theta})}\right] \tag{12}$$

Proof. We start from the objective function Equation (3) and marginalize over the $\boldsymbol{\nu}, \boldsymbol{\theta}$ variable

$$\ln p(\boldsymbol{Y}|\boldsymbol{X}) = \ln \int d\boldsymbol{\nu} d\boldsymbol{\theta} p(\boldsymbol{Y}, \boldsymbol{\nu}, \boldsymbol{\theta}|\boldsymbol{X}) \tag{13}$$

We then divide and multiply by the variational distribution and recognize the expected value against this distribution

$$\begin{aligned}
\ln p(\boldsymbol{Y}|\boldsymbol{X}) &= \ln \int d\boldsymbol{\nu} d\boldsymbol{\theta} p(\boldsymbol{Y}, \boldsymbol{\nu}, \boldsymbol{\theta}|\boldsymbol{X}) \\
&= \ln \int d\boldsymbol{\nu} d\boldsymbol{\theta} p(\boldsymbol{Y}, \boldsymbol{\nu}, \boldsymbol{\theta}|\boldsymbol{X}) \frac{q(\boldsymbol{\nu}, \boldsymbol{\theta})}{q(\boldsymbol{\nu}, \boldsymbol{\theta})} \\
&= \ln \int d\boldsymbol{\nu} d\boldsymbol{\theta} q(\boldsymbol{\nu}, \boldsymbol{\theta}) \frac{p(\boldsymbol{Y}, \boldsymbol{\nu}, \boldsymbol{\theta}|\boldsymbol{X})}{q(\boldsymbol{\nu}, \boldsymbol{\theta})} \\
&= \ln \mathbb{E}_{q(\boldsymbol{\nu},\boldsymbol{\theta})}\left[\frac{p(\boldsymbol{Y}, \boldsymbol{\nu}, \boldsymbol{\theta}|\boldsymbol{X})}{q(\boldsymbol{\nu}, \boldsymbol{\theta})}\right]
\end{aligned}$$

We then apply the concavity of the logarithm function

$$\begin{aligned}
&\ln \mathbb{E}_{q(\boldsymbol{\nu},\boldsymbol{\theta})}\left[\frac{p(\boldsymbol{Y}, \boldsymbol{\nu}, \boldsymbol{\theta}|\boldsymbol{X})}{q(\boldsymbol{\nu}, \boldsymbol{\theta})}\right] \\
\geq &\mathbb{E}_{q(\boldsymbol{\nu},\boldsymbol{\theta})}\left[\ln \frac{p(\boldsymbol{Y}, \boldsymbol{\nu}, \boldsymbol{\theta}|\boldsymbol{X})}{q(\boldsymbol{\nu}, \boldsymbol{\theta})}\right]
\end{aligned}$$

and we obtain the ELBO

$$\ln p(\boldsymbol{Y}|\boldsymbol{X}) \geq \mathbb{E}_{q(\boldsymbol{\nu},\boldsymbol{\theta})}\left[\ln \frac{p(\boldsymbol{Y}, \boldsymbol{\nu}, \boldsymbol{\theta}|\boldsymbol{X})}{q(\boldsymbol{\nu}, \boldsymbol{\theta})}\right] \tag{14}$$

□

C LIPSCHITZ CONTINUITY OF THE ELBO

Similar to (Errica et al., 2025a), we provide a derivation of the Lipschitz continuity of the ELBO. Although we provide different approaches on how to smoothly transition in the representation of the univariate function ϕ^{ℓ} using interpolation, there will still be a sudden change in the KAN functions. By providing the following property, we show that this jump is bounded. Therefore, we provide a theoretical result (Theorem 3.1) that shows that the ELBO satisfies the Lipschitz continuity.

Remark C.1. ***ELBO continuity on*** $(\boldsymbol{X}, \boldsymbol{\theta}, \boldsymbol{u})$
Since the ELBO (i.e. Equation (12)*) as a function of $\boldsymbol{X}, \boldsymbol{\theta}, \boldsymbol{u}$ is implemented as a composition of differentiable functions in $\boldsymbol{X}, \boldsymbol{\theta}, \boldsymbol{u}$, the ELBO itself is continuous and differentiable. The only discrete variable is the number of bases (K_{ℓ}), which has a structural impact on the architecture of the network. In the following, we show that the change on the ELBO due to a change in K_{ℓ} is bounded.*

ELBO Lipschitz continuity

Theorem C.2. *The ELBO loss of Equation* (9)*, with respect to the change in the number of basis K_ℓ (or ν^ℓ) for the layer ℓ, is Lipschitz continuous.*

Proof. We focus on the term involving K_ℓ of the ELBO, we write Equation (9) as

$$\underbrace{\ln p(\boldsymbol{Y}|\bar{\boldsymbol{\nu}}, \bar{\boldsymbol{\theta}}, \boldsymbol{X})}_{f_2(\boldsymbol{K})} + \ln \frac{p(\bar{\boldsymbol{\nu}})}{q(\bar{\boldsymbol{\nu}})} + \underbrace{\ln \frac{p(\bar{\boldsymbol{\theta}})}{q(\bar{\boldsymbol{\theta}}|\bar{\boldsymbol{\nu}})}}_{f_1(\boldsymbol{K})}$$

where only the first and last terms depend on $\mathbf{K} = \{K_\ell\}_{\ell=1}^L$. Since K_ℓ is a deterministic function of ν_ℓ, we consider them, in the following, equivalent. Let's first define

$$\ln \frac{p(\bar{\boldsymbol{\theta}})}{q(\bar{\boldsymbol{\theta}}|\bar{\boldsymbol{\nu}})} = \sum_{\ell=1}^{L} \sum_{k=1}^{K_\ell} \ln \frac{p(\bar{\boldsymbol{\theta}}_k^\ell)}{q(\bar{\boldsymbol{\theta}}_k^\ell|\nu_\ell)} = \sum_{\ell=1}^{L} f_1(K_\ell)$$

We have that $f_1(K_\ell)$ is Lipschitz continuous, indeed, when K_ℓ changes to K'_ℓ, we have

$$|f_1(K'_\ell) - f_1(K_\ell)| = |\sum_{k=K_\ell}^{K'_\ell} \ln \frac{p(\bar{\boldsymbol{\theta}}_k^\ell)}{q(\bar{\boldsymbol{\theta}}_k^\ell|\nu_\ell)}| \tag{15}$$

$$\leq \sum_{k=K_\ell}^{K'_\ell} |\ln \frac{p(\bar{\boldsymbol{\theta}}_k^\ell)}{q(\bar{\boldsymbol{\theta}}_k^\ell|\nu_\ell)}| \tag{16}$$

$$\leq \max_n |\log \frac{p(\boldsymbol{\theta}_n)}{q(\boldsymbol{\theta}_n|\boldsymbol{\nu})}||K_{\ell'} - K_\ell| \tag{17}$$

Therefore

$$|f_1(K'_\ell) - f_1(K_\ell)| \leq M|K'_\ell - K_\ell|$$

with $M = \max_k |\ln \frac{p(\bar{\boldsymbol{\theta}}_k^\ell)}{q(\bar{\boldsymbol{\theta}}_k^\ell|\nu_\ell)}|$. We now look at the first term,

$$f_2(\mathbf{K}) = \log p(Y|\boldsymbol{\nu}, \boldsymbol{\theta}, X)$$

If we use bounded derivative continuous univariate functions in the KAT representation (Equation (1)), and since f_2 is the composition of continuous univariate functions (see also Equation (10)), the resulting function is continuous and of bounded derivative and therefore Lipschitz continuous. An explicit constant for unit increments $K \mapsto K+1$ is given in Theorem D.2. □

D Soft Truncation, Jump Bounds, and Occam Penalty

We collect three complementary results that strengthen the analysis of the weighting function and of adaptive basis growth. Throughout, we fix a layer ℓ and drop the layer index when unambiguous. Write $K = K(\nu) = \lceil -\ln(1-\tau)/\nu \rceil$,

$$\phi_K^{\text{hard}}(x) = \sum_{k=1}^{K} \theta_k \, \varphi_k^K(x), \qquad \phi^\nu(x) = \sum_{k=1}^{K} w_k^\nu \, \theta_k \, \varphi_k^K(x),$$

with w_k^ν as in Equation (11), and assume the generative bases are uniformly bounded on the compact domain of interest: $\sup_{k \leq K} \|\varphi_k^K\|_\infty \leq B_\varphi < \infty$. Note that KAT is defined on $[0, 1]$ for example.

Soft truncation approximates hard truncation

Theorem D.1. *Let* $W_K(\nu) = \sum_{j=1}^{K} P(j) = e^{-\nu}(1 - e^{-K\nu})$ *and recall* $w_k^\nu = (1 - e^{-\nu})e^{-\nu(k-1)}/(1 - e^{-K\nu})$. *Then*

$$\left\|\phi^\nu - \phi_K^{\mathrm{hard}}\right\|_\infty \le B_\varphi \sum_{k=1}^{K} |w_k^\nu - 1|\,|\theta_k|$$
$$\le B_\varphi \,\|\theta\|_{1:K} \sum_{k=1}^{K} (1 - w_k^\nu) = B_\varphi \,\|\theta\|_{1:K}\,(K-1), \tag{18}$$

where $\|\theta\|_{1:K} = \max_{k\le K} |\theta_k|$ *(using* $0 < w_k^\nu \le 1$ *and* $\sum_k w_k^\nu = 1$*). Moreover, as* $\nu \to 0^+$ *with* $K(\nu) \to \infty$ *under fixed* τ*, one has* $w_k^\nu \to 1/K$ *uniformly on compact* k*-ranges, while for fixed* K *and* $\nu \to \infty$, $w_1^\nu \to 1$ *and* $w_k^\nu \to 0$ *for* $k \ge 2$. *In particular, for any* $\varepsilon > 0$ *there exists* ν_0 *such that for all* $\nu \ge \nu_0$ *with* $K(\nu) = K$ *fixed,* $\|\phi^\nu - \theta_1 \varphi_1^K\|_\infty \le \varepsilon$.
Furthermore, on every open interval where $K(\nu)$ *is constant,* ϕ^ν *is differentiable in* ν *with*

$$\partial_\nu \phi^\nu(x) = \sum_{k=1}^{K} \theta_k \,\varphi_k^K(x)\,\partial_\nu w_k^\nu, \tag{19}$$

and $\partial_\nu w_k^\nu$ *is given in closed form by Equation* (63)*. The sign pattern of* $\partial_\nu w_k^\nu$ *reallocates mass from large* k *toward small* k *when* ν *increases (basis shrinkage) and conversely when* ν *decreases (basis growth).*

Proof sketch. From Section I, $P(k) = e^{-\nu k}(1 - e^{-\nu})$ and $\sum_{j=1}^{K} P(j) = e^{-\nu}(1 - e^{-K\nu})$, which yields the stated formula for w_k^ν. Then

$$\phi^\nu - \phi_K^{\mathrm{hard}} = \sum_{k=1}^{K} (w_k^\nu - 1)\,\theta_k\,\varphi_k^K,$$

and taking $\|\cdot\|_\infty$ with $\|\varphi_k^K\|_\infty \le B_\varphi$ gives Equation (18). The limiting regimes follow by elementary asymptotics of the geometric weights. Differentiability on fixed-K intervals is immediate from the closed-form w_k^ν; the shrinkage/growth interpretation follows by inspecting

$$\frac{\partial_\nu w_k^\nu}{w_k^\nu} = \frac{e^{-\nu}}{1 - e^{-\nu}} - (k-1) - \frac{K e^{-K\nu}}{1 - e^{-K\nu}},$$

which is strictly decreasing in k. Full algebra is in Section I. We also use from Section J that $sum_{k=1}^{K}(1 - w_k^\nu) = K - 1$. □

Explicit ELBO jump bound

Theorem D.2. *Fix all variational parameters except the basis count of a single layer, and write* $\mathrm{ELBO}(K)$ *for the first-order objective Equation* (9) *evaluated at truncation* K*. Assume: (i) Gaussian coefficient priors* $p(\theta_k) = \mathcal{N}(0, \sigma^2)$ *with variational means bounded by* $|\bar{\theta}_k| \le M_\theta$*; (ii) layer maps use bases with* $\|\varphi_k^K\|_\infty \le B_\varphi$ *and weighting* $\|w\|_\infty \le 1$*; (iii) the predictive log-likelihood* $\ell(f) = \sum_{i=1}^{D} \ln p(y_i \mid f(x_i))$ *is* L_ℓ*-Lipschitz in the network output under* $\|\cdot\|_\infty$ *(e.g. cross-entropy or Gaussian NLL on a compact logit/output range). Then*

$$\big|\mathrm{ELBO}(K{+}1) - \mathrm{ELBO}(K)\big| \le C_{\mathrm{KL}} + C_{\mathrm{lik}}, \tag{20}$$

with complexity jump

$$C_{\mathrm{KL}} = \max\Big\{\big|\ln p(\bar{\theta}_{K+1})\big|,\ \tfrac{1}{2}\big(\tfrac{M_\theta^2}{\sigma^2} + \ln(2\pi\sigma^2)\big)\Big\}$$

and likelihood jump

$$C_{\mathrm{lik}} = L_\ell\, L_{\mathrm{net}}\, M_\theta\, B_\varphi,$$

where L_{net} *is a Lipschitz factor of the remaining KAN composition w.r.t. the modified univariate edge (product of layer Lipschitz constants;* $L_{\mathrm{net}} = 1$ *for a single-layer model). Consequently,* $\mathrm{ELBO}(\cdot)$ *is Lipschitz on* $\mathbb{N}$ *with constant* $C_{\mathrm{KL}} + C_{\mathrm{lik}}$*.*

Proof sketch. Decompose $\mathrm{ELBO}(K) = f_2(K) + f_1(K) + \mathrm{const}(\nu)$ as in Section C. For the prior term,

$$\big|f_1(K{+}1) - f_1(K)\big| = \big|\ln p(\bar{\theta}_{K+1})\big| \le C_{\mathrm{KL}},$$

using the Gaussian log-density and $|\bar{\theta}_{K+1}| \le M_\theta$. For the likelihood, the only architectural change when appending basis $K{+}1$ is an additive edge perturbation of size at most $|w_{K+1}\bar{\theta}_{K+1}\varphi_{K+1}| \le M_\theta B_\varphi$ in $\|\cdot\|_\infty$. Propagating through the remaining Lipschitz network yields an output change of at most $L_{\mathrm{net}} M_\theta B_\varphi$, and the L_ℓ-Lipschitz predictive log-likelihood gives the C_{lik} term. Summing the two contributions proves Equation (20). □

Occam penalty of the variational objective

Theorem D.3. *Under independent Gaussian priors* $p(\theta_k) = \mathcal{N}(0, \sigma^2)$ *and the first-order ELBO of Equation* (9)*, the coefficient prior contribution at layer* ℓ *satisfies*

$$R_\theta(K) := \sum_{k=1}^{K} \ln p(\bar{\theta}_k) = -\frac{K}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{k=1}^{K}\bar{\theta}_k^2. \tag{21}$$

Hence $R_\theta(K)$ *is strictly decreasing in* K *whenever the newly added coefficient is nonzero, and*

$$R_\theta(K{+}1) - R_\theta(K) = -\tfrac{1}{2}\ln(2\pi\sigma^2) - \frac{\bar{\theta}_{K+1}^2}{2\sigma^2} \le -\tfrac{1}{2}\ln(2\pi\sigma^2).$$

Moreover, if the truncated-exponential rate prior contribution $R_\nu(\nu) = \ln p(\nu;\eta) - \ln q(\nu;\bar{\nu})$ *is held fixed, the map* $K \mapsto R_\theta(K) + R_\nu$ *implements an automatic complexity penalty of order* $\Theta(K\log(1/\sigma))$ *that favors smaller basis counts unless compensated by a sufficient increase in the data log-likelihood. In the large-data limit* $D \to \infty$ *with fixed priors, the data term dominates and the penalty becomes relatively negligible, consistent with the empirical observation that prior hyperparameters matter less for large datasets.*

Proof. Equation (21) is the standard Gaussian log-density summed over $k = 1, \dots, K$. The one-step decrement follows by direct subtraction. The $\Theta(K\log(1/\sigma))$ rate is immediate from the $-K\frac{1}{2}\ln(2\pi\sigma^2)$ term when σ is treated as a fixed prior hyper-parameter. Relative negligibility as $D \to \infty$ follows because the likelihood term in Equation (9) scales as $\Theta(D)$ while $R_\theta(K)$ depends on K but not on D. □

Remark D.4 (Implications for training)**.** *Theorem D.1 justifies using w_k^ν as a differentiable surrogate of hard truncation; Theorem D.2 makes the Lipschitz claim of Theorem 3.1 quantitative; Theorem D.3 explains why InfinityKAN does not systematically inflate K in practice and why the Gaussian prior variance σ^2 acts as a capacity knob complementary to ν.*

Remark D.5 (Lipschitz on $\mathbb{N}$)**.** *Theorem Theorem C.2 can also be derived from the general Theorem D.6 and $\mathbb{N} \subset \mathbb{Z}$.*

Theorem D.6. *Let $f : \mathbb{R} \to \mathbb{R}$ be an L-Lipschitz continuous function on $\mathbb{R}$ for some constant $L \geq 0$. Then the restriction $f|_{\mathbb{Z}} : \mathbb{Z} \to \mathbb{R}$ is L-Lipschitz continuous on $\mathbb{Z}$ with a Lipschitz constant $L_{\mathbb{Z}} \leq L$.*

Proof. By assumption, $f : \mathbb{R} \to \mathbb{R}$ is L-Lipschitz continuous, which means that there exists a constant $L \geq 0$ such that for every pair $x, y \in \mathbb{R}$, the inequality

$$|f(x) - f(y)| \leq L|x - y| \tag{22}$$

holds. Since the set of integers $\mathbb{Z}$ is a subset of the real numbers $\mathbb{R}$ ($\mathbb{Z} \subset \mathbb{R}$), any two elements $m, n \in \mathbb{Z}$ are also elements of $\mathbb{R}$. Applying inequality (1) specifically to the integer points $x = m$ and $y = n$ yields

$$|f|_{\mathbb{Z}}(m) - f|_{\mathbb{Z}}(n)| = |f(m) - f(n)| \leq L|m - n| \quad \forall m, n \in \mathbb{Z}. \tag{23}$$

Hence, $f|_{\mathbb{Z}}$ satisfies the Lipschitz condition on $\mathbb{Z}$ with constant L. Furthermore, the optimal Lipschitz constant $L_{\mathbb{Z}}$ over $\mathbb{Z}$, defined as

$$L_{\mathbb{Z}} = \sup_{\substack{m,n \in \mathbb{Z} \\ m \neq n}} \frac{|f(m) - f(n)|}{|m - n|},$$

satisfies $L_{\mathbb{Z}} \leq L$ since the supremum is taken over a subset of $\mathbb{R} \times \mathbb{R}$. □

Lemma D.7. *Let $f : \mathbb{R} \to \mathbb{R}$ be an L-Lipschitz continuous function on $\mathbb{R}$ for some constant $L \geq 0$. Then the restriction $f|_{\mathbb{N}} : \mathbb{N} \to \mathbb{R}$ is L-Lipschitz continuous on $\mathbb{Z}$ with a Lipschitz constant $L_{\mathbb{N}} \leq L$.*

The proof is analogous to Theorem D.6 but restricted to $\mathbb{N}$.

Lipschitz Extensions and the McShane Theorem. The converse relationship between discrete and continuous Lipschitz functions is established by the McShane–Whitney Extension Theorem (McShane, 1934). If a function $g : \mathbb{Z} \to \mathbb{R}$ is L-Lipschitz on the integers (which on $\mathbb{Z}$ simplifies to the local condition $|g(n+1) - g(n)| \leq L$ for all $n \in \mathbb{Z}$), it can always be extended to an L-Lipschitz function $G : \mathbb{R} \to \mathbb{R}$ defined on the entire real line without increasing the Lipschitz constant. One canonical extension formula is given explicitly by $G(x) = \inf_{n \in \mathbb{Z}} \{g(n) + L|x - n|\}$. Alternatively, a linear interpolation between consecutive integers yields a continuous piece-wise linear extension with the exact same Lipschitz constant L.

E First-Order approximation and n-th order approximation

The first-order approximation requires the function $f \in C^0$ to be continuous in a neighbor of $\mu_x = \mathbb{E}_x[x]$, then

$$\mathbb{E}_x[f(x)] = \mathbb{E}_x[f(\mu_x) + O((x - \mu_x))] \approx f(\mathbb{E}_x[x])$$

If $f \in C^1$ we would similarly have

$$\begin{aligned} \mathbb{E}_x[f(x)] &= \mathbb{E}_x[f(\mu_x) + f'(\mu_x)(x - \mu_x) + O((x - \mu_x)^2)] && (24) \\ &\approx f(\mathbb{E}_x[x]) && (25) \end{aligned}$$

while, with $f \in C^2$ we would similarly have

$$\begin{aligned} \mathbb{E}_x[f(x)] &= \mathbb{E}_x[f(\mu_x) + f'(\mu_x)(x - \mu_x) && (26) \\ &\quad + \frac{1}{2} f''(\mu_x)(x - \mu_x)^2 + O((x - \mu_x)^3)] && (27) \\ &\approx f(\mathbb{E}_x[x]) + \frac{1}{2} f''(\mathbb{E}_x[x])(\mathbb{E}_x[x^2] - \mathbb{E}_x^2[x]) && (28) \\ &= f(\mu_x) + \frac{1}{2} f''(\mu_x)\sigma_x^2 && (29) \end{aligned}$$

The n-order approximation

$$\mathbb{E}_x[f(x)] = \mathbb{E}_x[f(\mu_x) + \sum_{k=1}^{n} \frac{f^{(k)}(\mu_x)}{k!}(x-\mu_x)^k + O((x-\mu_x)^{n+1})] \tag{30}$$

$$\approx f(\mu_x) + \sum_{k=1}^{n} \frac{f^{(k)}(\mu_x)}{k!}\mu_x^{(k)} \tag{31}$$

with $\mu_x^{(k)}$ the k-th momentum of the distribution. When we consider the special case of $f(x) = \log p(x)$, and we assume that the mean is optimal, i.e. that $p'(\mu_x) = 0$, then the second order approximation is

$$\mathbb{E}_x[\log p(x)] \approx \log p(\mu_x) + \frac{1}{2}\left[\frac{p''(\mu_x)}{p(\mu_x)} - \left(\frac{p'(\mu_x)}{p(\mu_x)}\right)^2\right]\sigma_x^2$$

$$= \log p(\mu_x) + \frac{1}{2}\frac{p''(\mu_x)}{p(\mu_x)}\sigma_x^2 \tag{32}$$

First Order ELBO approximation If we then apply the first-order approximation to the ELBO, we have

$$\ln p(\boldsymbol{Y}|\boldsymbol{X}) \geq \mathbb{E}_{q(\boldsymbol{\nu},\boldsymbol{\theta})}\left[\ln \frac{p(\boldsymbol{Y},\boldsymbol{\nu},\boldsymbol{\theta}|\boldsymbol{X})}{q(\boldsymbol{\nu},\boldsymbol{\theta})}\right]$$

$$= \mathbb{E}_{q(\boldsymbol{\nu},\boldsymbol{\theta})}\left[\ln p(\boldsymbol{Y} \mid \boldsymbol{\nu},\boldsymbol{\theta},\boldsymbol{X}) + \ln \frac{p(\boldsymbol{\nu},\boldsymbol{\theta})}{q(\boldsymbol{\nu},\boldsymbol{\theta})}\right]$$

$$\approx \ln p(\boldsymbol{Y} \mid \bar{\boldsymbol{\nu}},\bar{\boldsymbol{\theta}},\boldsymbol{X}) + \ln \frac{p(\bar{\boldsymbol{\nu}},\bar{\boldsymbol{\theta}})}{q(\bar{\boldsymbol{\nu}},\bar{\boldsymbol{\theta}})} \tag{33}$$

Second Order ELBO approximation If we then apply the second-order approximation to the ELBO, we have

$$\ln p(\boldsymbol{Y}|\boldsymbol{X}) \geq \mathbb{E}_{q(\boldsymbol{\nu},\boldsymbol{\theta})}\left[\ln \frac{p(\boldsymbol{Y},\boldsymbol{\nu},\boldsymbol{\theta}|\boldsymbol{X})}{q(\boldsymbol{\nu},\boldsymbol{\theta})}\right]$$

$$= \mathbb{E}_{q(\boldsymbol{\nu},\boldsymbol{\theta})}\left[\ln p(\boldsymbol{Y} \mid \boldsymbol{\nu},\boldsymbol{\theta},\boldsymbol{X}) + \ln \frac{p(\boldsymbol{\nu},\boldsymbol{\theta})}{q(\boldsymbol{\nu},\boldsymbol{\theta})}\right]$$

$$\approx \ln p(\boldsymbol{Y} \mid \bar{\boldsymbol{\nu}},\bar{\boldsymbol{\theta}},\boldsymbol{X}) + \ln \frac{p(\bar{\boldsymbol{\nu}},\bar{\boldsymbol{\theta}})}{q(\bar{\boldsymbol{\nu}},\bar{\boldsymbol{\theta}})} \tag{34}$$

$$+ \frac{1}{2}D^2_{\boldsymbol{\nu},\boldsymbol{\theta}} \ln p(\boldsymbol{Y} \mid \bar{\boldsymbol{\nu}},\bar{\boldsymbol{\theta}},\boldsymbol{X})\boldsymbol{\Sigma}^2_{\boldsymbol{\nu},\boldsymbol{\theta}}$$

$$+ \frac{1}{2}D^2_{\boldsymbol{\nu},\boldsymbol{\theta}} \ln \frac{p(\bar{\boldsymbol{\nu}},\bar{\boldsymbol{\theta}})}{q(\bar{\boldsymbol{\nu}},\bar{\boldsymbol{\theta}})}\boldsymbol{\Sigma}^2_{\boldsymbol{\nu},\boldsymbol{\theta}}$$

Proposition E.1 (First-order ELBO gap)**.** *Suppose the map $(\boldsymbol{\nu},\boldsymbol{\theta}) \mapsto \ln p(\boldsymbol{Y} \mid \boldsymbol{\nu},\boldsymbol{\theta},\boldsymbol{X}) + \ln \frac{p(\boldsymbol{\nu},\boldsymbol{\theta})}{q(\boldsymbol{\nu},\boldsymbol{\theta})}$ is twice continuously differentiable with Hessian spectral norm bounded by L_H in a neighborhood of the variational mean $(\bar{\boldsymbol{\nu}},\bar{\boldsymbol{\theta}})$, and let $\boldsymbol{\Sigma}^2$ collect the variational second moments. Then the gap between the true ELBO expectation and the first-order (mean-field MAP) surrogate satisfies*

$$\left|\mathbb{E}_q[\ell(\boldsymbol{\nu},\boldsymbol{\theta})] - \ell(\bar{\boldsymbol{\nu}},\bar{\boldsymbol{\theta}})\right| \leq \tfrac{1}{2}L_H \operatorname{tr}(\boldsymbol{\Sigma}^2),$$

where ℓ denotes the integrand of the ELBO. Thus the first-order training objective is a controlled approximation whose error scales with the variational variance.

Proof. Let $\boldsymbol{z} = (\boldsymbol{\nu},\boldsymbol{\theta})^T \in \mathbb{R}^d$ collect all parameters, with mean vector $\bar{\boldsymbol{z}} = (\bar{\boldsymbol{\nu}},\bar{\boldsymbol{\theta}})^T = \mathbb{E}_q[\boldsymbol{z}]$. The integrand function $\ell(\boldsymbol{z})$ is twice continuously differentiable with Hessian matrix $\mathbf{H}(\boldsymbol{z}) = \nabla^2\ell(\boldsymbol{z})$.

By Taylor's theorem with Lagrange remainder around the mean $\bar{\boldsymbol{z}}$, for any $\boldsymbol{z}$ there exists some point $\boldsymbol{\xi}$ on the line segment joining $\boldsymbol{z}$ and $\bar{\boldsymbol{z}}$ such that

$$\ell(\boldsymbol{z}) = \ell(\bar{\boldsymbol{z}}) + \nabla\ell(\bar{\boldsymbol{z}})^T(\boldsymbol{z} - \bar{\boldsymbol{z}}) + \frac{1}{2}(\boldsymbol{z} - \bar{\boldsymbol{z}})^T\mathbf{H}(\boldsymbol{\xi})(\boldsymbol{z} - \bar{\boldsymbol{z}}).$$

Taking the expectation under $q(\boldsymbol{z})$ on both sides:

1. The linear first-order term vanishes because $\mathbb{E}_q[\boldsymbol{z} - \bar{\boldsymbol{z}}] = \mathbf{0}$:
$$\mathbb{E}_q\big[\nabla\ell(\bar{\boldsymbol{z}})^T(\boldsymbol{z} - \bar{\boldsymbol{z}})\big] = \nabla\ell(\bar{\boldsymbol{z}})^T\mathbb{E}_q[\boldsymbol{z} - \bar{\boldsymbol{z}}] = 0.$$
2. For the quadratic remainder term, applying the bound on the spectral norm of the Hessian $\|\mathbf{H}(\boldsymbol{\xi})\|_2 \leq L_H$, we have:
$$\big|(\boldsymbol{z} - \bar{\boldsymbol{z}})^T\mathbf{H}(\boldsymbol{\xi})(\boldsymbol{z} - \bar{\boldsymbol{z}})\big| \leq \|\mathbf{H}(\boldsymbol{\xi})\|_2\|\boldsymbol{z} - \bar{\boldsymbol{z}}\|^2 \leq L_H\|\boldsymbol{z} - \bar{\boldsymbol{z}}\|^2.$$

Subtracting $\ell(\bar{\boldsymbol{z}})$ and applying the expectation yields:

$$\begin{aligned}
\big|\mathbb{E}_q[\ell(\boldsymbol{z})] - \ell(\bar{\boldsymbol{z}})\big| &= \left|\frac{1}{2}\mathbb{E}_q\left[(\boldsymbol{z} - \bar{\boldsymbol{z}})^T\mathbf{H}(\boldsymbol{\xi})(\boldsymbol{z} - \bar{\boldsymbol{z}})\right]\right| \\
&\leq \frac{1}{2}\mathbb{E}_q\left[L_H\|\boldsymbol{z} - \bar{\boldsymbol{z}}\|^2\right] \\
&= \frac{1}{2}L_H\,\mathbb{E}_q\left[\mathrm{tr}\big((\boldsymbol{z} - \bar{\boldsymbol{z}})(\boldsymbol{z} - \bar{\boldsymbol{z}})^T\big)\right] \\
&= \frac{1}{2}L_H\,\mathrm{tr}\left(\mathbb{E}_q\big[(\boldsymbol{z} - \bar{\boldsymbol{z}})(\boldsymbol{z} - \bar{\boldsymbol{z}})^T\big]\right) \\
&= \frac{1}{2}L_H\,\mathrm{tr}(\boldsymbol{\Sigma}^2),
\end{aligned}$$

where $\boldsymbol{\Sigma}^2 \equiv \mathrm{Cov}_q(\boldsymbol{z})$ represents the matrix of variational second moments. □

F CHEBYSHEV TYPE-I POLYNOMIALS

An interesting extension of the framework is when using Chebyshev type-I polynomials as basis functions, indeed the approximation error decreases with the number of bases, and adding bases is probably less critical then with basis on the real line. The Chebyshev polynomials of type-I are defined as

$$T^k(\cos\theta) = \cos(k\theta) \tag{35}$$

or $T^k(x) = \cos\left(k\cos^{-1}(x)\right), x \in [-1, 1]$, where we typically map the real axis to the $[-1, 1]$ interval using $z = \tanh(x)$. We can then define the series as

$$\phi(x) = \lim_{n\to\infty}\phi^n(x) \tag{36}$$

$$\phi^n(x) = \sum_{k\in[n]}\theta_k T^k(x) \tag{37}$$

with θ_k trainable parameters. The interesting point of the use of the Chebyshev polynomial is that we can now share the parameters among series and the dependence on the index n is dropped in the parameters. This is due to the Taylor expansion, where we drop dependence on the higher-order polynomials.

Asymmetric window We then introduce the asymmetric window function

$$w_\nu(x) = \left(1 + e^{2(x-\nu)/\gamma}\right)^{-1} \tag{38}$$

$$w_k^n = w_\mu(x_k), \;\; x_k \in [0, \nu + \gamma] \tag{39}$$

which then gives the final form of the trainable ℓ-th KAN layer function

$$\phi_\ell^n(x) = \sum_{k\in[n]} \theta_k^\ell w_k^n T^k(x) \tag{40}$$

with $\boldsymbol{\theta}_\ell = \{\theta_k^\ell\}_{k\in[n]}$ the trainable parameters, while μ_ℓ and γ_ℓ, the variational parameter and hyperparameter of the variational optimization problem.

Property F.1. *If we consider $\varphi_k(x) = T^k(x)$ we have that*

$$\int_{-1}^{1} dx h(x)\varphi_k(x)\varphi_{k'}(x) = \delta_{k-k'}$$

with $h(x) = \frac{1}{\sqrt{(1-x^2)}}$

G FOURIER BASIS AND REPRESENTATION

An alternative basis is the one defined based on the Fourier functions

$$\varphi_k(x) = \frac{1}{\sqrt{T}} e^{i\frac{2\pi}{T}kx}$$

with $T = 2$ and with domain $\Omega = [-1, 1]$. We have that any continuous function of period T can be represented as

$$\phi(x) = \lim_{n\to\infty} \sum_{k=-n}^{n} \phi_k(x) \tag{41}$$

$$= \lim_{n\to\infty} \sum_{k=-n}^{n} \theta_k \varphi_k(x), \tag{42}$$

$$\phi_k(x) = \sum_{k=-n}^{n} \theta_k \varphi_k(x) \tag{43}$$

with $\theta_k \in \mathbb{C}$ complex numbers. It is well known that

Property G.1. *(Fourier complex basis) If we consider $\varphi_k(x) = \frac{1}{\sqrt{T}} e^{i\frac{2\pi}{T}kx}$ we have that*

$$\int_{-T}^{T} dx \varphi_k(x)\varphi_{k'}^*(x) = \delta_{k-k'}$$

with $\varphi_k^*(x)$ the complex conjugate of $\varphi_k^*(x)$.

If we want to use real numbers, then we have two sets of bases

$$\varphi_k(x) = \frac{1}{\sqrt{T}} \cos\left(\frac{2\pi}{T}kx\right), \quad \varphi'_k(x) = \frac{1}{\sqrt{T}} \sin\left(\frac{2\pi}{T}kx\right)$$

if $T = 2$ then

$$\varphi_k(x) = \frac{1}{\sqrt{2}} \cos(\pi kx), \quad \varphi'_k(x) = \frac{1}{\sqrt{2}} \sin(\pi kx)$$

Property G.2. *(Fourier real basis) If we consider $\varphi_k(x) = \frac{1}{\sqrt{2}} \cos(\pi kx), \quad \varphi'_k(x) = \frac{1}{\sqrt{2}} \sin(\pi kx)$ we have that*

$$\int_{-1}^{1} dx \varphi_k(x)\varphi_{k'}(x) = \int_{-1}^{1} dx \varphi_k(x)\varphi'_{k'}(x) \tag{44}$$

$$= \int_{-1}^{1} dx \varphi'_k(x)\varphi'_{k'}(x) = \delta_{k-k'} \tag{45}$$

When can then use the basis for represent any periodic function in the interval $[-T/2, T/2]$, based on the following property.

Property G.3. *(Fourier representation) If we consider* $\varphi_k(x) = \frac{1}{\sqrt{2}}\cos(\pi k x)$, $\varphi'_k(x) = \frac{1}{\sqrt{2}}\sin(\pi k x)$ *we have that*

$$\phi(x) = \lim_{n\to\infty} \phi^n(x), \quad \phi^n(x) = \sum_{k=0}^{n} \theta_k \varphi_k(x) + \sum_{k=1}^{n} \theta'_k \varphi'_k(x)$$

with θ_k, θ'_k *the coefficients of the series.*

H Initialization with Chebyshev polynomials

We first recall that

$$\int dx T^k(x) = \frac{1}{2}\left[\frac{T^{k+1}}{k+1} - \frac{T^{k-1}}{k-1}\right] \tag{46}$$

therefore

$$\int dx (\phi^n_\ell(x))^2 = \left(\int dx \sum_{k\in[n]} \theta^{\ell n}_k w^n_k T^k(x)\right)^2 \tag{47}$$

$$= \left(\sum_{k\in[n]} \theta^{\ell n}_k w^n_k \int dx T^k(x)\right)^2 \tag{48}$$

$$= \left(\sum_{k\in[n]} \theta^{\ell n}_k w^n_k \frac{1}{2}\left[\frac{T^{k+1}}{k+1} - \frac{T^{k-1}}{k-1}\right]\right)^2 \tag{49}$$

$$\leq \sum_{k\in[n]} (\theta^{\ell n}_k)^2 (w^n_k)^2 \frac{1}{4}\left(\left[\frac{T^{k+1}}{k+1} - \frac{T^{k-1}}{k-1}\right]\right)^2 \tag{50}$$

$$\leq \sum_{k\in[n]} (\theta^{\ell n}_k)^2 (w^n_k)^2 \frac{1}{4} \tag{51}$$

$$\tag{52}$$

since $\left|\frac{T^{k+1}}{k+1} - \frac{T^{k-1}}{k-1}\right| \leq 1$. If we want

$$\sum_{k\in[n]} (\theta^{\ell n}_k)^2 (w^n_k)^2 \frac{1}{4} = 1$$

we can either set the variance to

$$\mathbb{E}(\theta^n_k)^2 = \frac{4}{\sum_{k\in[n]}(w^n_k)^2} \approx \frac{4}{n - 5/4},$$

when we assume the parameters to be i.i.d. and zero mean, $\mathbb{E}[\theta^n_k] = 0$. The $5/4$ term is due to the shape of the windows, when $\sigma = 1$, the last sample is ≈ 0 while the before-last sample is $1/2$.

I Analytic gradient

Closed-form gradient. We derive here the closed-form gradient of $\phi^\ell_{qp}(x)$ with respect to ν_ℓ.

Let $\nu \equiv \nu_\ell$ and $K \equiv K_\ell(\nu)$. For a fixed value of K (i.e., away from points where $K_\ell(\nu)$ changes), based on Equation (10) and Equation (11), we write

$$\phi^\ell_{qp}(x) = \sum_{k=1}^{K} w^\nu_k \, \theta^{\ell K}_{qpk} \, \varphi^K_k(x), \tag{53}$$

$$w^\nu_k = \frac{P(k)}{\sum_{j=1}^{K} P(j)}, \tag{54}$$

$$P(k) = F(k+1) - F(k), \qquad F(x) = 1 - e^{-\nu x}. \tag{55}$$

From F, we obtain

$$P(k) = \left(1 - e^{-\nu(k+1)}\right) - \left(1 - e^{-\nu k}\right) = e^{-\nu k}\left(1 - e^{-\nu}\right). \tag{56}$$

Hence

$$\sum_{j=1}^{K} P(j) = \left(1 - e^{-\nu}\right) \sum_{j=1}^{K} e^{-\nu j} = \left(1 - e^{-\nu}\right) \frac{e^{-\nu}\left(1 - e^{-K\nu}\right)}{1 - e^{-\nu}}, \tag{57}$$

and therefore

$$w_k^\nu = \frac{\left(1 - e^{-\nu}\right) e^{-\nu(k-1)}}{1 - e^{-K\nu}}. \tag{58}$$

Now compute $\partial_\nu w_k^\nu$ through the log-derivative:

$$\log w_k^\nu = \log(1 - e^{-\nu}) - \nu(k-1) - \log(1 - e^{-K\nu}), \tag{59}$$

so

$$\frac{\partial_\nu w_k^\nu}{w_k^\nu} = \frac{e^{-\nu}}{1 - e^{-\nu}} - (k-1) - \frac{K e^{-K\nu}}{1 - e^{-K\nu}}. \tag{60}$$

Thus

$$\partial_\nu w_k^\nu = w_k^\nu \left[\frac{e^{-\nu}}{1 - e^{-\nu}} - (k-1) - \frac{K e^{-K\nu}}{1 - e^{-K\nu}}\right]. \tag{61}$$

Assuming $\theta_{qpk}^{\ell K}$ and $\varphi_k^K(x)$ are independent of ν within a fixed-K region, we get

$$\frac{\partial \phi_{qp}^\ell(x)}{\partial \nu_\ell} = \sum_{k=1}^{K_\ell(\nu_\ell)} \theta_{qpk}^{\ell K_\ell(\nu_\ell)}\, \varphi_k^{K_\ell(\nu_\ell)}(x)\, \partial_{\nu_\ell} w_k^{\nu_\ell} \tag{62}$$

$$= \sum_{k=1}^{K_\ell(\nu_\ell)} \theta_{qpk}^{\ell K_\ell(\nu_\ell)}\, \varphi_k^{K_\ell(\nu_\ell)}(x)\, w_k^{\nu_\ell}$$

$$\left[\frac{e^{-\nu_\ell}}{1 - e^{-\nu_\ell}} - (k-1) - \frac{K_\ell(\nu_\ell) e^{-K_\ell(\nu_\ell)\nu_\ell}}{1 - e^{-K_\ell(\nu_\ell)\nu_\ell}}\right]. \tag{63}$$

Remark. Because $K_\ell(\nu_\ell)$ is integer-valued, $\phi_{qp}^\ell(x)$ is piecewise differentiable in ν_ℓ: Equation (63) holds on intervals where $K_\ell(\nu_\ell)$ is constant, while the classical derivative is not defined at transition points where K_ℓ changes. Since we only change K_ℓ according to Figure 3, the gradient is always well defined.

J FURTHER DERIVATIONS

Given the weight w_k^ν defined as:

$$w_k^\nu = \frac{\left(1 - e^{-\nu}\right) e^{-\nu(k-1)}}{1 - e^{-K\nu}}, \quad \text{for } K \in \mathbb{N}_{\geq 1},\ \nu \neq 0,$$

we want to compute:

$$S = \sum_{k=1}^{K} \left(1 - w_k^\nu\right).$$

Using the linearity property of summation, we write:

$$S = \sum_{k=1}^{K} \left(1 - w_k^\nu\right) = \sum_{k=1}^{K} 1 - \sum_{k=1}^{K} w_k^\nu = K - \sum_{k=1}^{K} w_k^\nu$$

For the second term we have that

$$\sum_{k=1}^{K} w_k^{\nu} = \sum_{k=1}^{K} \frac{(1-e^{-\nu})\, e^{-\nu(k-1)}}{1-e^{-K\nu}}$$
$$= \left(\frac{1-e^{-\nu}}{1-e^{-K\nu}}\right) \sum_{k=1}^{K} e^{-\nu(k-1)}.$$

If we define $j = k-1$, we have that

$$\sum_{k=1}^{K} e^{-\nu(k-1)} = \sum_{j=0}^{K-1} \left(e^{-\nu}\right)^j.$$

Using the formula for the sum of a finite geometric series $\sum_{j=0}^{K-1} r^j = \frac{1-r^K}{1-r}$ (for $r \neq 1$):

$$\sum_{j=0}^{K-1} \left(e^{-\nu}\right)^j = \frac{1-\left(e^{-\nu}\right)^K}{1-e^{-\nu}} = \frac{1-e^{-K\nu}}{1-e^{-\nu}}.$$

Thus,

$$\sum_{k=1}^{K} w_k^{\nu} = \left(\frac{1-e^{-\nu}}{1-e^{-K\nu}}\right) \cdot \left(\frac{1-e^{-K\nu}}{1-e^{-\nu}}\right) = 1.$$

This confirms that w_k^{ν} forms a normalized probability distribution over $k \in \{1, 2, \ldots, K\}$. Therefore

$$\sum_{k=1}^{K} (1 - w_k^{\nu}) = K - 1.$$

Note on the boundary case $\nu = 0$: If $\nu \to 0$, taking the limit via L'Hôpital's Rule yields $w_k^0 = \frac{1}{K}$. Thus $\sum_{k=1}^{K} w_k^0 = 1$, and the result $\sum_{k=1}^{K}(1 - w_k^0) = K - 1$ holds universally for all $\nu \in \mathbb{R}$.

K Dataset Characteristics and Hyper-Parameters

Table 4 summarizes the dataset properties, including input dimensionality, output dimensionality, and dataset size (number of observations).

Table 4: Dataset characteristics.

	Samples	Features	Classes
CIFAR10	60000	32x32x3	10
CIFAR100	60000	32x32x3	100
DoubleMoon	5000	2	2
EuroSAT	27000	32x32x3	10
EyeMovements	10936	27	3
FashionMNIST	70000	28x28	10
House	22784	16	2
Jannis	83733	54	4
MNIST	70000	28x28	10
MagicTelescope	19020	10	2
NCI1	4110	37	2
POL	15000	48	2
Phoneme	5404	5	2
MiniBooNE	130064	50	2
REDDIT-B	2000	1	2
Spiral	5000	2	2
SpiralHard	10000	2	2

Hyper-parameter tuning was conducted in a structured and comparable manner across all baselines, while allowing for dataset-specific adaptations on the synthetic benchmarks. For all models and datasets, we fixed a common training protocol in terms of batch size 128, number of epochs 1000, optimizer AdamW with learning rate 0.01 and no weight decay. Model capacity was explored through grid searches over the hidden size $\{2, 5, 16\}$ and the number of layers $\{1, 2\}$ (extended to $\{1, 2, 5\}$ for graph datasets), ensuring fairness across KAN, InfinityKAN, and MLP baselines. On the synthetic datasets (DoubleMoon, Spiral, and SpiralHard), additional hyper-parameters were tuned to capture increasing geometric complexity. For KAN and InfinityKAN models, we varied the number of fixed and starting number bases (KAN: $\{2, 8, 32, 128\}$, iKAN: $\{2, 64\}$), while for InfinityKAN we further explored different bases weighting functions (Symmetric, OneSided, Truncated ERFC, and Truncated Exponential) and activation functions (ReLU, Sigmoid, PReLU, Tanh, GELU, SiLU). These choices were shared across the three synthetic datasets, and we fixed the Truncated Exponential and PReLU for the remaining datasets as they clearly gave more stable and higher performances. By evaluating all approaches on identical datasets with consistent hyperparameter tuning budgets and evaluation protocols, we provide a comprehensive assessment of whether adaptive basis learning offers advantages over both fixed-basis approaches and traditional neural networks across varying problem complexities.

L Ablation studies

We conduct a series of ablation studies to understand the sensitivity of InfinityKAN to key design choices, including the choice of activation function, the weighting distribution, and the prior hyperparameters ν and σ.

Activation functions. We first investigate the impact of different activation functions on the performance of InfinityKAN. Figure 6 presents the validation accuracy across six activation functions (SiLU, GELU, Tanh, PReLU, Sigmoid, and ReLU) on the synthetic datasets. For the simpler DoubleMoon and Spiral tasks, all activation functions achieve near-perfect accuracy with minimal variance, indicating that the choice of activation is not critical for these problems. However,

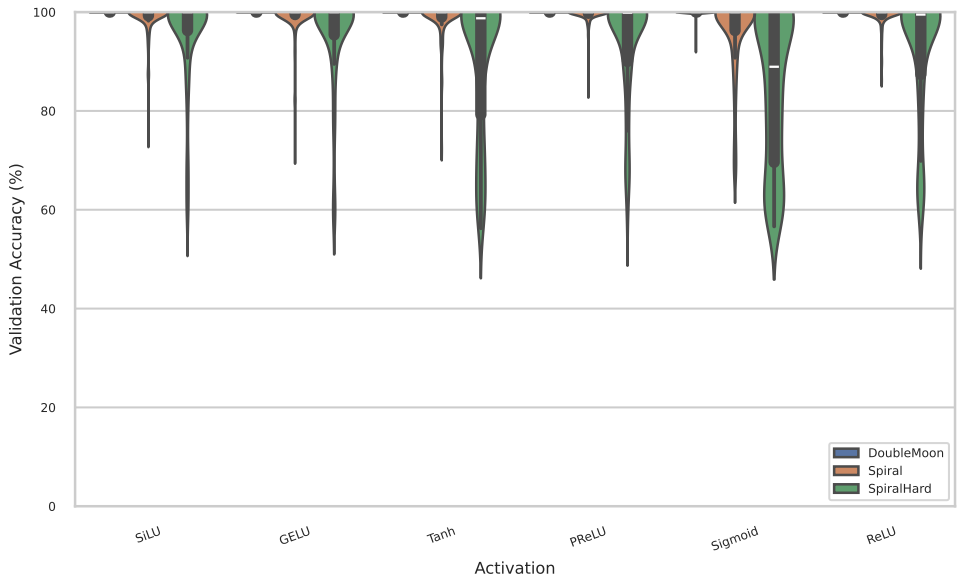


Figure 6: Validation accuracy of INFINITYKAN across different activation functions (SiLU, GELU, Tanh, PReLU, Sigmoid, ReLU) on the synthetic datasets DoubleMoon, Spiral, and SpiralHard. Violin plots show the distribution of scores over multiple runs. All activation functions achieve near-perfect accuracy on DoubleMoon and Spiral, while SpiralHard exhibits greater variability, with Sigmoid showing the lowest and most variable performance.

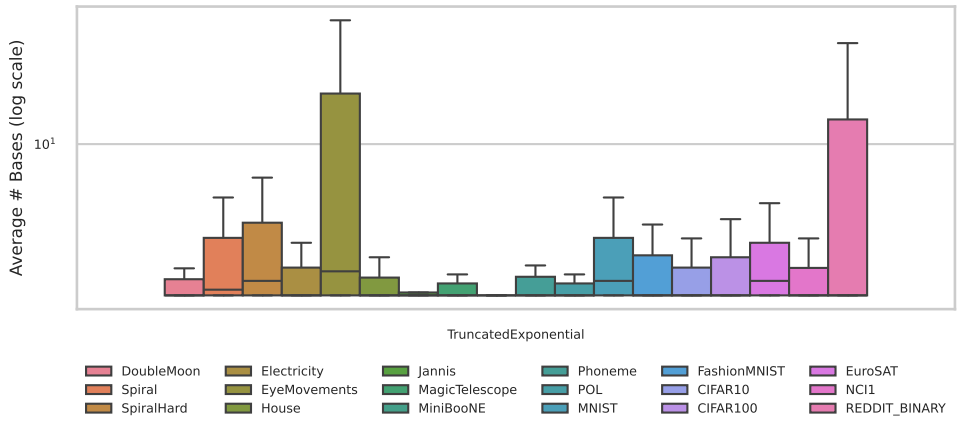


Figure 7: Average number of basis functions learned by INFINITYKAN using the Truncated Exponential weighting distribution across all 18 datasets. The y-axis is shown on a log scale. Error bars indicate standard deviation over 10 runs. Most datasets converge to a small number of bases (below 10), while EyeMovements and REDDIT_BINARY exhibit substantially higher and more variable basis counts, reflecting their greater complexity.

on the more challenging SpiralHard dataset, we observe greater differentiation: while most activations maintain high accuracy, Sigmoid exhibits notably lower and more variable performance. This suggests that for complex decision boundaries, smooth unbounded activations (such as PReLU and ReLU variants) provide more stable learning dynamics. Based on these findings, we selected PReLU as the default activation for subsequent experiments.

Learned number of bases across datasets. Figure 7 shows the average number of basis functions learned by INFINITYKAN using the Truncated Exponential weighting distribution across all 18 datasets. The results reveal that most datasets converge to a compact representation with fewer than 10 basis functions, demonstrating that INFINITYKAN effectively identifies parsimonious models.

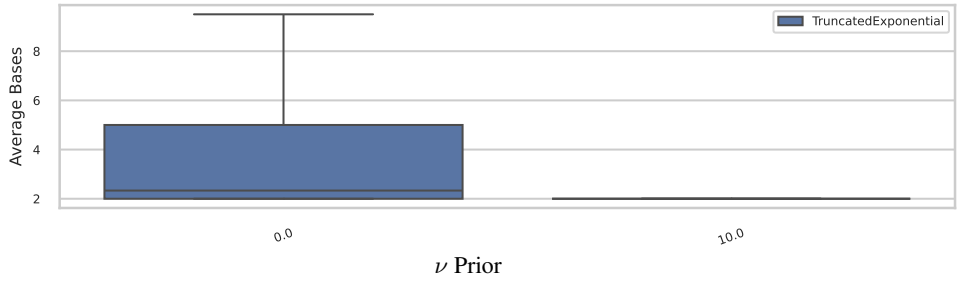


Figure 8: Effect of the ν prior hyperparameter on the average number of basis functions learned by INFINITYKAN using the Truncated Exponential weighting function. With $\nu = 0.0$, the model exhibits high variability in the learned number of bases, while a stronger prior ($\nu = 10.0$) regularizes the model toward fewer bases with reduced variance.

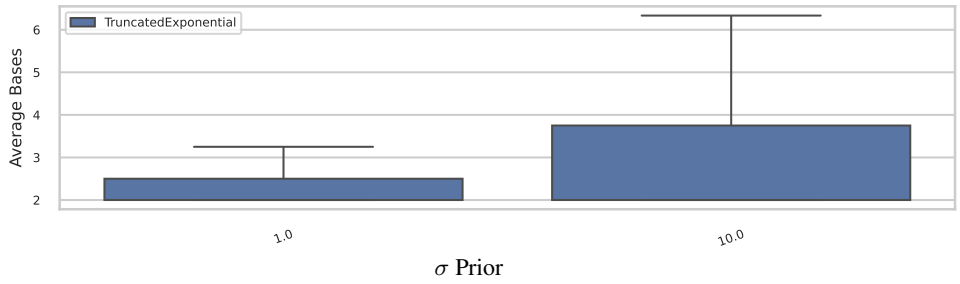


Figure 9: Effect of the σ prior hyperparameter on the average number of basis functions learned by INFINITYKAN using the Truncated Exponential weighting function. A weaker prior ($\sigma = 1.0$) results in fewer bases with lower variability, while a stronger prior ($\sigma = 10.0$) leads to a higher average number of bases with increased variance across runs.

However, certain datasets–particularly EyeMovements and REDDIT_BINARY–exhibit substantially higher and more variable basis counts. This behavior reflects the inherent complexity of these tasks: EyeMovements involves time-series patterns with temporal dependencies, while REDDIT_BINARY captures graph-structural information that may require richer functional representations. The variability in basis counts across runs for these datasets suggests that multiple model configurations can achieve similar performance, highlighting the exploratory nature of the variational optimization.

Effect of prior hyperparameters. We analyze the sensitivity of INFINITYKAN to the prior hyperparameters ν and σ, which control the regularization of the rate parameter and basis coefficients, respectively. Figure 8 shows that the ν prior has a strong regularizing effect on the learned number of bases. With $\nu = 0.0$ (no regularization), the model exhibits high variability in the learned basis count, ranging from approximately 2 to 10 bases. In contrast, setting $\nu = 10.0$ constrains the model to consistently learn around 2 bases with minimal variance. This demonstrates that the ν prior provides an effective mechanism for controlling model complexity when prior knowledge about the desired sparsity is available.

The σ prior exhibits a different behavior, as shown in Figure 9. A weaker prior ($\sigma = 1.0$) results in fewer bases (median around 2.5) with relatively low variability. Increasing the prior strength to $\sigma = 10.0$ leads to a higher average number of bases (median around 4) with substantially increased variance. This counterintuitive effect may arise because a stronger σ prior encourages larger coefficient magnitudes, which in turn requires more basis functions to adequately represent the target function. These ablation results provide practical guidance for practitioners: the ν prior offers direct control over model sparsity, while the σ prior influences the trade-off between basis count and coefficient magnitude.

M LIMITATIONS.

While INFINITYKAN eliminates the need to tune the number of basis functions, it introduces its own hyperparameters (the priors on ν and θ, i.e. η, and σ^2), though our ablation studies suggest these are less sensitive than the basis count and in the large-data limit become unimportant. The variational optimization introduces computational overhead compared to fixed-basis KANs: in our experiments, INFINITYKAN requires approximately $\approx 2\times$ the training time due to the dynamic resizing of parameter tensors. However, note that this overhead is strongly offset by the elimination of hyperparameter search for the number of bases. Additionally, on some datasets (e.g. EyeMovements), INFINITYKAN performs comparably to–but does not exceed–carefully tuned KANs, indicating that INFINITYKAN can replace a full hyperparameter search but may not outperform a well-guided manual tuning effort.

The current first-order approximation of the variational objective represents a compromise between accuracy and speed, but extending it to more sophisticated inference techniques–such as importance-weighted Monte Carlo or higher-order gradients–could improve the quality of the learned posterior over basis functions and lead to better uncertainty quantification, while incurring additional computational cost.

Lastly, the update on the basis strategy may have an impact on performance. High update frequency allows fast basis adaptation but increases the variance of the parameter estimation, while a lower update frequency would have the opposite effect, hindering the learning of the optimal number of bases.

IMPACT STATEMENT

INFINITYKAN's main practical effect is reducing the compute spent on hyperparameter search for basis count, which can lower the energy footprint of training KAN-based models relative to manual tuning. We are not aware of other specific societal risks beyond those common to machine learning research generally.