Training and Verifying robust Kolmogorov-Arnold Networks

Björn Heiderich *
bjoern.heiderich@ipa.fraunhofer.de

Max-Lion Schumacher *
max-lion.schumacher@ipa.fraunhofer.de

Marco F. Huber
marco.huber@ieee.org

Abstract

Kolmogorov–Arnold Networks (KANs) offer strong theoretical representational power but, like MLPs and CNNs, remain vulnerable to adversarial attacks. Benchmarks on Fashion MNIST and CIFAR10 confirm this susceptibility. We introduce GloroKAN, leveraging KANs' B-spline structure to approximate local Lipschitz constants directly in the forward pass, boosting robustness without gradient-based adversarial training and approaching adversarially trained performance. Additionally, we propose a verification method using algebraic geometry to exploit KANs' piecewise polynomial nature. While these findings highlight KANs' potential for robust, interpretable models, further research is needed to realize their full promise.

1 Introduction

Deep neural networks have shown remarkable performance in various domains, from computer vision to natural language processing. Despite their impressive capabilities, these models often lack robustness against adversarial perturbations (Carlini et al., 2019; Croce & Hein, 2020). Small, carefully constructed changes to inputs can drastically alter a model's output, posing significant safety and reliability concerns in real-world applications such as autonomous driving, medical diagnosis, and cybersecurity.

Kolmogorov–Arnold Networks (KANs) constitute a distinct architectural family arising from the classical Kolmogorov–Arnold superposition theorem, which guarantees that any multivariate continuous function can be decomposed into compositions of univariate functions (Lin & Unbehauen, 1993; Köppen, 2002; Montanelli & Yang, 2020). Recently, these networks have gained renewed attention as potentially more interpretable or structured architectures, featuring explicit forms of learnable activation functions realized via B-splines (Liu et al., 2024). Despite promising theoretical guarantees in terms of representational power, the robustness of KANs has not been deeply studied, and existing research indicates they can exhibit similar vulnerabilities to adversarial attacks as conventional deep networks (Alter et al., 2024; Zeng et al., 2024; Shen et al., 2024).

In this work, we take a systematic look at adversarial robustness in KANs. We first benchmark Kolmogorov–Arnold Networks against standard multi-layer perceptrons (MLPs) and convolutional neural networks (CNNs) on two canonical datasets, Fashion MNIST and CIFAR10. Through extensive experiments under various attack methods (Auto Attack (Croce & Hein, 2020) and spatial transformations (Engstrom et al., 2019)), we show how KANs compare in terms of adversarial accuracy and clean accuracy. Beyond empirical findings, we also propose a novel training approach that leverages the structural properties of B-splines in KANs to approximate local Lipschitz constants directly in the forward pass of the network. This strategy yields a new model variant, *GloroKAN*, that can be seen as arbitrarily close to a globally robust classifier for appropriately small neighborhoods around input data—without requiring computationally expensive backward passes for gradient-based adversarial training methods. Furthermore, we propose a new robustness verification method for KANs based on the work of Newton (2023) for neural networks. This approach is based on some theory

[*] These authors contributed equally to this work

from algebraic geometry. Since KAN activations are piecewise polynomials and algebraic geometry deals with sets defined by relations of polynomials, we believe this to be a suitable approach.

Contributions. The key contributions of this paper are:

- **Benchmarking KANs for adversarial robustness.** We provide a thorough empirical comparison of KANs, MLPs, and CNNs with and without adversarial training on Fashion MNIST and CIFAR10, evaluating both l^∞ robustness and spatial robustness.
- **New training method for robustness.** We introduce a novel procedure, *GloroKAN*, that exploits the B-spline structure of KANs to compute an approximate local Lipschitz constant in one forward pass. We prove that GloroKAN approaches a model with guaranteed global robustness in small neighborhoods.
- **Verification via semialgebraic sets.** We show how the piecewise polynomial nature of KAN activations can be harnessed for formal robustness verification via semialgebraic set methods, thus bridging a gap between theoretical guarantees and practical verifiability.

The remainder of this paper is structured as follows. In Sec. 2, we summarize related work on Kolmogorov-Arnold Networks, adversarial robustness, and verification strategies. We then introduce in Sec. 3.2 our new *Robustness Training* method and prove that it yields a model arbitrarily close to a globally robust classifier. Sec. 3.3 details a *Robustness Verification* approach grounded in algebraic geometry for bounding the network's output on polytopic input sets. In Sec. 4.1 we present an empirical *Benchmarking of KANs* by evaluating various architectures on Fashion MNIST and CIFAR10 with different attack methods. Sec. 4.2 analyses the empirical results of our new *Robustness Training* method. The paper closes with conclusions and an outlook to future work.

2 Related Work

The topic of Kolmogorov-Arnold Networks (KANs) has recently gained increased interest through Liu et al. (2024). Previous publications on this topic are Sprecher David A (2002); Köppen (2002); Lin & Unbehauen (1993); Lai & Shen (2024); Fakhoury et al. (2022); Montanelli & Yang (2020); He (2023). Applications of KANs include time series, e.g., Genet & Inzirillo (2024), the medical field, e.g., Tang et al. (2024), computer vision, e.g., Mahara et al. (2024), Graph Neural Networks, e.g., Xu et al. (2024), and Physics Informed Neural Networks, e.g., Abueidda et al. (2025). A recent survey on KANs can be found in Ji et al. (2025).

One of the first publications to mention the existence of adversarial attacks is Goodfellow et al. (2015). A good overview of the topic of adversarial robustness can be found in Bai et al. (2021); Silva & Najafirad (2020). In Carlini et al. (2019) best practices for the evaluation of machine learning models with regard to adversarial robustness are given.

The adversarial robustness of KANs has not yet been comprehensively investigated. To the best of the authors' knowledge, there are only a few publications to refer to that mainly deal with the adversarial robustness analysis of KANs. Publications that can be mentioned are Shen et al. (2024), Alter et al. (2024) and Zeng et al. (2024). As is already known for other deep neural networks, KANs are also susceptible to perturbations Shen et al. (2024). Alter et al. (2024) has already benchmarked different KAN models.

Robustness verification aims to provide formal guarantees that a model's predictions remain consistent under bounded adversarial or natural perturbations. Early methods focused on exact, solver-based approaches. For instance, Reluplex Katz et al. (2017) formulates the problem as an SMT instance to systematically search for adversarial examples or prove their absence in piecewise linear networks. MILP-based approaches such as Tjeng et al. (2019) similarly leverage integer programming formulations to offer exact robustness guarantees, though these approaches often face high computational costs when scaling to large networks. More scalable methods typically rely on abstractions or relaxations that bound the outputs of layers under all possible perturbations. Abstract interpretation Gehr et al. (2018) is a representative framework in this vein, providing approximate yet formally sound robustness certificates. Recently, auto_LiRPA Xu et al. (2020) has emerged as a versatile tool that automates linear relaxation–based bounding methods, making it easier to verify larger and more diverse network architectures.

3 METHODS

3.1 PREREQUISITES

KANs are neural network architectures based on the Kolmogorov–Arnold superposition theorem. For a detailed introduction, we refer the reader to (Liu et al., 2024). We will use the following definition of local robustness of a model $f : \mathbb{R}^m \to \mathbb{R}^n$ with $F(x) := \arg\max f(x)$. F is the class prediction of our model f.

Definition 3.1. *A model f is locally robust around x with true label y and radius r if*

$$||x - x'|| \leq r \Rightarrow y = F(x') \tag{1}$$

In order to describe the global behaviour of a model f, we introduce the following naïve score for global robustness:

Definition 3.2. *Let $\mathcal{D}$ be a probability distribution on $\mathbb{R}^m$ and $l : \mathbb{R}^m \to \{1, \ldots, n\}$ the function that assigns the true label to each data point. Then we define*

$$R(f) := \mathbb{E}_{x \sim \mathcal{D}} \left[\mathbf{1}_{\, l(x) = F(x'), \, \forall x' \, \textit{with} \, ||x - x'|| \leq r \}} \right] \quad . \tag{2}$$

$R(f)$ is called the global robustness score of model f. For realizations $x_1, \ldots, x_l$, $l \in \mathbb{N}$, from $\mathcal{D}$,

$$\tilde{R}(f) := \sum_{i=1}^{l} \left[\mathbf{1}_{\, l(x_i) = F(x'), \, \forall x' \, \textit{with} \, ||x_i - x'|| \leq r \}} \right] \tag{3}$$

is called the empirical global robustness score of model f.

Obviously $0 \leq R(f) \leq 1$ applies. The robustness of model f increases with larger values of $R(f)$.

3.2 ROBUSTNESS TRAINING

The main idea of this section is to use an idea of a training method taken from Leino et al. (2021) and change the computation of Lipschitz constants by taking benefit out of the special structure of KANs. We prove analytically that our new model gets arbitrarily close to a model with guaranteed global robustness without the need to compute gradients with respect to the inputs in a backward pass. More precisely, KANs use a weighted sum of B-splines plus a basis function as learnable activation functions Liu et al. (2024). For $t_0, \ldots, t_m$, $m \in \mathbb{N}$ a B-spline is defined by

$$B_{i,0}(x) := \begin{cases} 1, \text{ if } t_i \leq x \leq t_{i+1} \\ 0 \quad \text{otherwise} \end{cases} \quad .$$

and

$$B_{i,k}(x) := \frac{x - t_i}{t_{i+k} - t_i} B_{i,k-1}(x) + \frac{t_{i+k+1} - x}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(x) \, .$$

for $0 \leq i < m$ and $k \geq 0$, see also de Boor (2002). $(B_{i,k})_{0 \leq i < m}$ is called a B-Spline of order k. For a B-spline of order k, the derivative is given by

$$\frac{\mathrm{d}B_{i,k}(x)}{\mathrm{d}x} = k \left(\frac{B_{i,k-1}(x)}{t_{i+k} - t_i} - \frac{B_{i+1,k-1}(x)}{t_{i+k+1} - t_{i+1}} \right) \, . \tag{4}$$

Due to the recursive definition of the B-splines, the derivative of the B-splines with respect to x can be calculated directly in the forward pass of the KAN model.

To apply the method proposed in Leino et al. (2021), one has to compute the Lipschitz constant for every f_i, $1 \leq i \leq n$, where f_i is the i-th element of f and thus

$$f = (f_1, \ldots, f_n) \, . \tag{5}$$

We propose to focus on local Lipschitz constants instead of a global Lipschitz constant. Computing the exact local Lipschitz constant is NP-hard Katz et al. (2017), so it is necessary to think of approximations to the local Lipschitz constant if one wants to reduce the complexity of the computation. We introduce the definition of a local Lipschitz constant:

Definition 3.3. *Let $h : \mathbb{R}^m \to \mathbb{R}$ be a differentiable function. For each $x \in \mathbb{R}^m$ and $r > 0$, we define*

$$L_{x,r} := \inf\{L \mid |h(y) - h(z)| \leq L\|y - z\|,\ \forall y, z \in U(x, r)\} \ .$$

$U(x, r)$ is the sphere around x with radius r, i.e., $U(x, r) := \{y \mid \|y - x\| \leq r\}$. In the following, $L_{x,r}$ is referred to as the (smallest) local Lipschitz constant for h at the point x with radius r.

The following statement also applies to the smallest local Lipschitz constant. Proofs of the statements can be found in the appendix.

Remark 3.4. *If $h : \mathbb{R}^m \to \mathbb{R}$ is differentiable and Lipschitz-continuous with constant L, then*

$$\max_{1\leq i\leq m} \left|\frac{\mathrm{d}}{\mathrm{d}x} h(x)\right| = \max_{1\leq i\leq m} \left(\left|\frac{\partial}{\partial x_1} h(x)\right|, \ldots, \left|\frac{\partial}{\partial x_m} h(x)\right|\right) \leq L \ .$$

The maximum over all partial derivatives in absolute value is smaller than the Lipschitz constant L. If we consider the local Lipschitz constant instead, the limit value is even equal, as the following lemma shows.

Lemma 3.5. *Let $h : \mathbb{R}^m \to \mathbb{R}$ be continuously differentiable. Then holds*

$$\max_{1\leq i\leq m} \left|\frac{\mathrm{d}}{\mathrm{d}x} h(x)\right| = \max_{1\leq i\leq m} \left(\left|\frac{\partial}{\partial x_1} h(x)\right|, \ldots, \left|\frac{\partial}{\partial x_m} h(x)\right|\right) = \lim_{r\to 0} L_{x,r} \ . \tag{6}$$

Lemma 3.5 implies that in small neighborhoods around a point x, the maximum magnitude of the partial derivatives provides a suitable approximation for the local Lipschitz constant. In our method for training KANs, we will use the left-hand side of (6) as an approximation for the local Lipschitz constant. We will now briefly introduce the idea from Leino et al. (2021), for more details we refer to this source. Let f be as in (5) and and $(K_i)_{1\leq i\leq n} \in \mathbb{R}^n$. Let

$$\tilde{f}(x, (K_i)_{1\leq i\leq n}) := \begin{cases} f_i(x),\ 1 \leq i \leq n \\ \max_{i\neq j}\{y_i + (K_i + K_j)H\tilde{r}\},\ y = f(x),\ j = F(x),\ i = n + 1 \end{cases} \tag{7}$$

for every $\tilde{r} > 0$ and $H > 0$. $\tilde{f}$ obviously depends on $\tilde{r}$, for better readability we do not include the index $\tilde{r}$ in our definition. H is a hyperparameter. $\dim(\tilde{f})$ is one dimension larger than the number of available labels. If $F = n + 1$ at a data point, then the model cannot output a robust prediction at this data point, see Leino et al. (2021). Let L_i be the Lipschitz constant of f_i. We define a model $g(x) := \tilde{f}(x, (L_i)_{1\leq i\leq n})$. We change Theorem 1 from Leino et al. (2021) towards the following statement.

Theorem 3.6. *If $G(x) := \arg\max g(x) \neq n+1$, then $G(x) = F(x)$ and F is locally robust around x for all radii r with $r \leq H\tilde{r}$.*

Theorem 3.6 remains true if we replace L_i for $1 \leq i \leq n$ by the local Lipschitz constant $L_{i,x,r}$ for every f_i. Remark 3.4 and Lemma 3.5 apply to f_i. We therefore define a new model $h_r := \tilde{f}(x, (L_{i,x,r})_{1\leq i\leq n})$, $r \leq H\tilde{r}$, for which Theorem 3.6 is still valid.

Theorem 3.7. *Set*

$$h_{\text{approx}}(x) := \tilde{f}\Big(x, \Big(\max_{1\leq j\leq m} \left|\frac{\mathrm{d}}{\mathrm{d}x} f_i(x)\right|\Big)_{1\leq i\leq n}\Big) \ . \tag{8}$$

Then it follows that

$$h_r \overset{r\to 0}{\rightrightarrows} h_{\text{approx}} \quad .$$

When f is a KAN, then h_{approx} is referred to as GloroKAN, based on the name used in Leino et al. (2021).

This follows directly from Lemma 3.5. Due to the form in which the derivative of B-splines is expressed, see (4), and the definition of GloroKAN, it is possible to calculate GloroKAN together with one forward pass of the model f with the same complexity in terms of the required arithmetic operations. More precisely, if one forward pass of the model f requires N arithmetic operations, then the number of required arithmetic operations of GloroKAN is in $O(N)$. For a small radius r, we know that GloroKAN $\approx h_r$, for which Theorem 3.7 holds, meaning that GloroKAN is arbitrarily close to a guaranteed robust classifier with radius r. The following algorithm outlines the calculation of GloroKAN when the model f is a KAN. f^l denotes the l-th layer of the model f, where f has a total of L layers. $\frac{\mathrm{d}}{\mathrm{d}x} f^l(\text{output})$ gets calculated according to (4) during the calculation of $f^l(\text{output})$.

Algorithm 1: Calculation of GloroKAN

```
Input: x
Output: $h_{\text{approx}}(x)$
output $\leftarrow x$; diff $\leftarrow \boldsymbol{I}_n$ ;
for $l \leftarrow 1$ to $L$ do
    output, diff_layer $\leftarrow f^l$(output), $\frac{d}{dx} f^l$(output) ;
    diff $\leftarrow$ diff $*$ diff_layer
diff$_i \leftarrow \max |\text{diff}_{:,i}|, \forall 1 \leq i \leq n$ ;
$j \leftarrow \arg\max$ output ;
return (output, $\max_{i \neq j}\{\text{output}_i + (\text{diff}_i + \text{diff}_j) H \tilde{r}\}$)
```

3.3 Robustness Verification

In this section we focus on finding a good overapproximation for the image of a rectangular input set under a KAN. Since KAN activations are piecewise polynomials, we make use of tools from algebraic geometry dealing with sets defined by relations of polynomials. The techniques used are based on Newton (2023) and adapted to KANs. Let x^k, $k = 1, \ldots, N$, be the activations at the KAN nodes, where x^0 is the input layer, x^N the output layer and $x^k = (x^k_j)_{j=1,\ldots,M_k}$ denotes the single activations of each layer.

We consider the input constraints

$$x^0 - \underline{x^0} \geq 0 \, , \tag{9}$$

$$\overline{x^0} - x^0 \geq 0 \, . \tag{10}$$

The bounds $\underline{x^0}$ and $\overline{x^0}$ specify the desired size of the input set. As overapproximation sets of the output, we consider polytopes. Consequently, the output constraints have the form

$$c_r^T x^N + d_r \geq 0 \, , \tag{11}$$

where the c_r and d_r, $r = 1, \ldots, R$, specify the shape and size of the polytope.

We consider the problem of finding the smallest polytope containing the output. This can be described as finding

$$d^* = \inf\{d : c^T x^N + d \geq 0\} \, . \tag{12}$$

Our approach is to formulate the KAN in a way such that Equation 12 becomes

$$d^* = \inf\{f(x) : x \in K\} \tag{13}$$

for a suitable polynomial f and semialgebraic set

$$K = \{x \in \mathbb{R}^n : g_j(x) \geq 0, \, h_l(x) = 1, \, j = 1, \ldots u, \, l = 1, \ldots, v\} \, ,$$

g_j and h_l being suitable polynomials. To achieve this, we formulate the KAN in the following way. As above, denote the grid $t_0, \ldots, t_m$. Let $B_{k,j,v,i}$ be the KAN activation from node x^k_l to x^{k+1}_v between node t_i and t_{i+1}. The variable I will be defined such that it represents an indicator function specific to the grid interval, and two connected nodes. Consider the following constraints. All x^k_j and $I_{k,j,i}$ are considered formal variables of polynomials and ε is a small real number.

$$(t_i - x^k_j)(t_{i+1} - x^k_j)(1 - I_{k,j,i})^2 \geq 0 \tag{14}$$

$$I^2_{k,j,i}(x^k_j - t_i - \varepsilon) \geq 0 \tag{15}$$

$$I^2_{k,j,i}(t_{i+1} - \varepsilon - x^k_j) \geq 0 \tag{16}$$

$$x^{k+1}_v - \sum_{\substack{j=0,\ldots,M_k \\ i=0,\ldots,m}} I_{k,j,i} B_{k,j,v,i} = 0 \tag{17}$$

Here we have $i = 0, \ldots, m$, $j = 0, \ldots, m$, $k = 0, \ldots, N-1$ and $v = 0, \ldots, M_{k+1}$. equation 14 makes $I_{k,j,i}$ be equal to 1 inside $[t_i, t_{i+1}]$ and does not constrain $I_{k,j,i}$ outside of $[t_i, t_{i+1}]$. equation 15 makes $I_{k,j,i}$ be equal to 0 for $x^k_j < t_i + \varepsilon$, while Equation 16 does the same for

$x_j^k > t_{i+1} - \varepsilon$. The small number ε has the purpose of avoiding inconsistencies at the interval bounds t_i, t_{i+1}. Finally Equation 17 defines the structure of F. We are now ready to formulate the robustness verification problem in terms of Equation 13:

$$d_r^* = \inf\{d_r : (9), (10), (11), (14) - (17) \text{ are satisfied } \}.$$

From this point on, solvers can be used, such as Wang et al. (2021), making use of chordal and other sparsity patterns, making the problem Equation 13 tractable. Future work will focus on conducting experiments to validate the proposed method.

4 Experiments

4.1 Benchmark

We want to analyse the robustness of KANs as given in Liu et al. (2024). For this purpose, different models are evaluated on the two datasets Fashion MNIST and CIFAR10. Solving condition (1) is NP-hard even for single-layer neural networks Katz et al. (2017), Weng et al. (2018). Instead of solving condition (1), one can use adversarial attack methods to solve (1) approximatively by trying to find the closest element x_{min} for which $F(x) \neq F(x_{\text{min}})$ holds. $||x_{\text{min}} - x|| > r$ applies exactly when condition (1) is fullfilled. We then refer to $\tilde{R}$, the empirical global robustness score, as adversarial accuracy. In order to obtain the best possible approximation to x_{min}, some best practices should be followed, see also Carlini et al. (2019): Different attacks should be used, which should be tested with different hyperparameters. In general, gradient-based attacks are stronger than gradient-free attacks as long as no gradient masking is used. Since the focus here is on the comparison of robustness across different architectures and not on the evaluation of a defence method, not all proposals from Carlini et al. (2019) are applicable. We decided to use the following two attack methods for evaluating the robustness of the models using the implementations of the Adversarial Robustness Toolbox Nicolae et al. (2018):

- Auto Attack, see Croce & Hein (2020)
- Spatial Transformations Attack, see Engstrom et al. (2019)

We only use l_∞ with $l_\infty(x) := \sup_{1 \leq i \leq m} |x_i|$ for $x \in \mathbb{R}^m$ as Norm in (1), (2) and (3) for our benchmarking. Auto attack is an ensemble of different attack methods, which has proven to be a successful evaluation for adversarial robustness Croce & Hein (2020) and does not require extensive hyperparameter tuning. Therefore, it seems to us to be the most suitable procedure to evaluate the adversarial accuracy of different models and to draw a meaningful comparison between them. The Spatial Transformations Attack differs from the attacks in auto attack in that it does not evaluate the robustness with regard to l_∞, but with regard to displacements and rotations Engstrom et al. (2019). Adversarial training is performed by using the FGSM attack Goodfellow et al. (2015). We performed each experiment five times and included the standard deviation in our results. We analysed four different model architectures on Fashion MNIST and CIFAR10: a multi-layer perceptron, short MLP, a KAN, a CNN, i.e., a model consisting of convolution layers followed by linear layers, and a model that has convolution layers in the first layers followed by KAN layers. We refer to the last model as CKAN. All models are evaluated towards four different radii r, where r is as in definition 3.1.

For the adversarial accuracy on Fashion MNIST we get the results as shown in Table 1, 2.

Table 1: adversarial accuracy Fashion MNIST, **NO** adversarial training

	MLP	**KAN**	**CNN**	**CKAN**
$\mathbf{r = 0.3}$	$\mathbf{0.0663 \pm 0.0026}$	0.0647 ± 0.0016	0.0468 ± 0.0026	$\mathbf{0.0644 \pm 0.0026}$
$\mathbf{r = 0.1}$	$\mathbf{0.3805 \pm 0.0057}$	0.2157 ± 0.0104	0.0564 ± 0.0019	$\mathbf{0.0796 \pm 0.0036}$
$\mathbf{r = 0.05}$	$\mathbf{0.6763 \pm 0.0035}$	0.5903 ± 0.0056	0.0772 ± 0.0053	$\mathbf{0.3050 \pm 0.0350}$
$\mathbf{r = 0.01}$	$\mathbf{0.8862 \pm 0.0021}$	0.8753 ± 0.0032	0.8018 ± 0.0088	$\mathbf{0.8101 \pm 0.0112}$

The results for the adversarial accuracy on CIFAR10 are displayed in Table 3, 4. Further results for clean accuracy and spatial robustness can be found in appendix A.3.

Table 2: adversarial accuracy Fashion MNIST, **with** adversarial training

	MLP	**KAN**	**CNN**	**CKAN**
$\mathbf{r = 0.3}$	$\mathbf{0.1833 \pm 0.0102}$	0.1101 ± 0.0172	0.0538 ± 0.0048	$\mathbf{0.0754 \pm 0.0099}$
$\mathbf{r = 0.1}$	0.7800 ± 0.0026	$\mathbf{0.7833 \pm 0.0042}$	0.3681 ± 0.064	$\mathbf{0.7389 \pm 0.0428}$
$\mathbf{r = 0.05}$	0.8379 ± 0.0036	$\mathbf{0.8399 \pm 0.0011}$	0.8062 ± 0.0075	$\mathbf{0.8183 \pm 0.0073}$
$\mathbf{r = 0.01}$	$\mathbf{0.8850 \pm 0.0028}$	0.8731 ± 0.0169	$\mathbf{0.8941 \pm 0.0019}$	0.8785 ± 0.0052

Table 3: adversarial accuracy CIFAR10, **NO** adversarial training

	MLP	**KAN**	**CNN**	**CKAN**
$\mathbf{r = 0.3}$	0.1403 ± 0.0045	$\mathbf{0.1561 \pm 0.0011}$	0.0528 ± 0.0018	$\mathbf{0.0569 \pm 0.0038}$
$\mathbf{r = 0.1}$	$\mathbf{0.2465 \pm 0.0020}$	0.1978 ± 0.0042	0.0924 ± 0.0036	$\mathbf{0.0995 \pm 0.0013}$
$\mathbf{r = 0.05}$	$\mathbf{0.3783 \pm 0.0016}$	0.2746 ± 0.0029	0.2059 ± 0.0083	$\mathbf{0.2287 \pm 0.0102}$
$\mathbf{r = 0.01}$	$\mathbf{0.5387 \pm 0.0030}$	0.4788 ± 0.0025	0.7553 ± 0.0060	$\mathbf{0.7629 \pm 0.0023}$

Table 4: adversarial accuracy CIFAR10, **with** adversarial training

	MLP	**KAN**	**CNN**	**CKAN**
$\mathbf{r = 0.3}$	0.1179 ± 0.0034	$\mathbf{0.2128 \pm 0.0080}$	0.0569 ± 0.0032	$\mathbf{0.0655 \pm 0.0040}$
$\mathbf{r = 0.1}$	$\mathbf{0.3760 \pm 0.0107}$	0.3132 ± 0.0058	$\mathbf{0.4835 \pm 0.0063}$	0.4672 ± 0.0046
$\mathbf{r = 0.05}$	$\mathbf{0.4425 \pm 0.0033}$	0.3388 ± 0.0025	$\mathbf{0.6653 \pm 0.0032}$	0.6423 ± 0.0043
$\mathbf{r = 0.01}$	$\mathbf{0.5096 \pm 0.0034}$	0.4354 ± 0.0034	$\mathbf{0.8362 \pm 0.0012}$	0.8169 ± 0.0027

In the benchmark, the adversarial accuracy of the MLP model is predominantly slightly better than that of the KAN model on both datasets. For the CNN and CKAN without adversarial training, the CKAN model has a slightly better adversarial accuracy than the CNN on both datasets. On Fashion MNIST, the adversarial accuracy of the CKAN with adversarial training is better than that of the CNN with adversarial training on three of the four radii, while on CIFAR10 it is the other way round. Based on these results, it is therefore not possible to claim that models with KAN layers are in general more or less robust compared to models with MLP layers. If we want to summarize the results of the benchmark as briefly as possible, we believe that the statement of comparable robustness between the two model variants is the most accurate. A comparison of MLP and KAN with CNN and CKAN is not meaningful due to the different model complexity and architecture. The complexity of the models we use is in the range between 1.7 million and 9 million trainable model parameters and is therefore more than a quarter smaller than the small models in Alter et al. (2024). Different adversarial attacks were also used to evaluate the robustness of the models. A comparison of the results is therefore difficult and is not carried out here.

4.2 Empirical Results GloroKAN

GloroKAN and a KAN model with and without adversarial training are trained on Fashion MNIST and evaluated for different radii r. Adversarial training was again carried out with FGSM attack Goodfellow et al. (2015). Two runs are carried out for each radius r. All models are retrained and evaluated for each run. The results are averaged. We define clean accuracy as the standard accuracy evaluated on the dataset without any perturbations. For GloroKAN, we set $H = 20$, $\tilde{r} = 0.3$. The larger we choose the hyperparameter H, the more we enforce the adversarial accuracy while potentially decreasing the clean accuracy. We refer to the additional dimension of GloroKAN as abstain prediction, because it indicates that the datapoint is not robust, so no prediction should be done. We measure the robustness score using $\tilde{R}$ from Definition 3.2, where the inner condition from Definition 3.1 is not solved exactly but estimated with Auto Attack. We therefore refer to $\tilde{R}$ as adversarial accuracy. We refer to "GloroKAN with abstain prediction" when the prediction is either the correct label or the abstain prediction. Figure 1 shows our results. We can assume that it is not possible to achieve greater adversarial accuracy than clean accuracy. This means that for

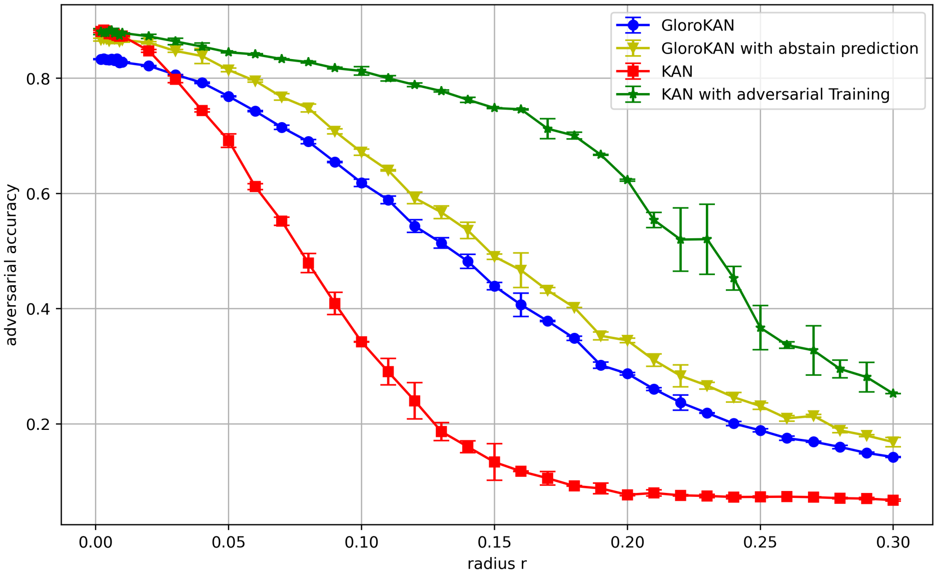


Figure 1: adversarial accuracy

a model that is guaranteed to be robust with respect to a radius r according to Definition 3.1, the adversarial accuracy is equal to the clean accuracy. If we include Figure 2a from the appendix, we see that GloroKAN converges in adversarial accuracy to the value of clean accuracy. In addition, the adversarial accuracy of GloroKAN is significantly higher than that of a KAN model for all radii r considered. Nethertheless, adversarial training is still more effective and achieves a higher adversarial accuracy than GloroKAN. However, GloroKAN with abstain prediction for $r \leq 0.05$ achieves almost as high an adversarial accuracy as a KAN with adversarial training. The clean accuracy without adversarial perturbations of GloroKAN is below that of a KAN model with and without adversarial training.

The additional dimension of the GloroKAN model may worsen the clean accuracy, but we also see that if we still consider the abstain prediction as a good classification, it significantly increases the robustness not only in terms of adversarial accuracy, but also in terms of spatial robustness, see Figure 2b from the appendix. Spatial robustness is measured by perturbing the inputs using the spatial transformation attack Engstrom et al. (2019). This is interesting as it is not the goal of GloroKAN to increase this robustness. Shifts and rotations of the spatial transformation attack lead to inputs that are in a non-robust area for GloroKAN, i.e., the abstain prediction.

5 Conclusion

Our experiments (Sec. 4.1) confirm that Kolmogorov–Arnold Networks (KANs), despite their unique B-spline–based structure, are as susceptible to adversarial perturbations as standard MLPs or CNNs. Both Auto Attack and spatial transformations can reliably degrade performance.

To address these vulnerabilities, we introduced GloroKAN (Sec. 3.2), which leverages the B-spline stucture within KANs to approximate the local Lipschitz constant efficiently during the forward pass of the model. By doing so, GloroKAN is arbitrarily close to a globally robust classifier with a certified and small enough radius. However, standard adversarial training achieves still better adversarial accuracy than GloroKAN (Sec. 4.2).

Finally, we presented an algebraic geometry–based approach (Sec. 3.3) for verifying robustness in KANs by recasting network constraints as a basic semialgebraic set. Though promising, its large-scale feasibility remains to be thoroughly tested. Overall, our findings highlight that KANs are not inherently robust; nonetheless, by exploiting their piecewise polynomial structure for both training and verification, we can move closer to reliable, certifiable performance.

REFERENCES

Diab W. Abueidda, Panos Pantidis, and Mostafa E. Mobasher. Deepokan: Deep operator network based on kolmogorov arnold networks for mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, 436:117699, 2025. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2024.117699. URL https://www.sciencedirect.com/science/article/pii/S0045782524009538.

Tal Alter, Raz Lapid, and Moshe Sipper. On the robustness of kolmogorov-arnold networks: An adversarial perspective, 2024. URL https://arxiv.org/abs/2408.13809.

Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness. In Zhi-Hua Zhou (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 4312–4321. International Joint Conferences on Artificial Intelligence Organization, 8 2021. doi: 10.24963/ijcai.2021/591. URL https://doi.org/10.24963/ijcai.2021/591. Survey Track.

Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness, 2019. URL https://arxiv.org/abs/1902.06705.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, 2020. URL https://arxiv.org/abs/2003.01690.

Carl de Boor. Chapter 6 - spline basics. In Gerald Farin, Josef Hoschek, and Myung-Soo Kim (eds.), *Handbook of Computer Aided Geometric Design*, pp. 141–163. North-Holland, Amsterdam, 2002. ISBN 978-0-444-51104-1. doi: https://doi.org/10.1016/B978-044451104-1/50007-1. URL https://www.sciencedirect.com/science/article/pii/B9780444511041500071.

Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness, 2019. URL https://arxiv.org/abs/1712.02779.

Daniele Fakhoury, Emanuele Fakhoury, and Hendrik Speleers. Exsplinet: An interpretable and expressive spline-based neural network. *Neural Networks*, 152:332–346, 2022. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2022.04.029. URL https://www.sciencedirect.com/science/article/pii/S0893608022001617.

Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, 2018. doi: 10.1109/SP.2018.00058.

Remi Genet and Hugo Inzirillo. Tkan: Temporal kolmogorov-arnold networks, 2024. URL https://arxiv.org/abs/2405.07344.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015. URL https://arxiv.org/abs/1412.6572.

Juncai He. On the optimal expressive power of relu dnns and its application in approximation with kolmogorov superposition theorem, 2023. URL https://arxiv.org/abs/2308.05509.

Tianrui Ji, Yuntian Hou, and Di Zhang. A comprehensive survey on kolmogorov arnold networks (kan), 2025. URL https://arxiv.org/abs/2407.11075.

Guy Katz, Clark Barrett, David Dill, Kyle Julian, and Mykel Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks, 2017. URL https://arxiv.org/abs/1702.01135.

Mario Köppen. On the training of a kolmogorov network. In *International Conference on Artificial Neural Networks*, 2002. URL https://api.semanticscholar.org/CorpusID:39406481.

Ming-Jun Lai and Zhaiming Shen. The kolmogorov superposition theorem can break the curse of dimensionality when approximating high dimensional functions, 2024. URL https://arxiv.org/abs/2112.09963.

Klas Leino, Zifan Wang, and Matt Fredrikson. Globally-robust neural networks, 2021. URL https://arxiv.org/abs/2102.08452.

Ji-Nan Lin and Rolf Unbehauen. On the realization of a kolmogorov network. *Neural Computation*, 5:18–20, 1993. URL https://api.semanticscholar.org/CorpusID:43876736.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks, 2024. URL https://arxiv.org/abs/2404.19756.

Arpan Mahara, Naphtali D. Rishe, and Liangdong Deng. The dawn of kan in image-to-image (i2i) translation: Integrating kolmogorov-arnold networks with gans for unpaired i2i translation, 2024. URL https://arxiv.org/abs/2408.08216.

Hadrien Montanelli and Haizhao Yang. Error bounds for deep relu networks using the kolmogorov–arnold superposition theorem. *Neural Networks*, 129:1–6, 2020. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2019.12.013. URL https://www.sciencedirect.com/science/article/pii/S0893608019304058.

Matthew Newton. Analysis of robust neural networks for control, 2023.

Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, Ian Molloy, and Ben Edwards. Adversarial robustness toolbox v1.2.0. *CoRR*, 1807.01069, 2018. URL https://arxiv.org/pdf/1807.01069.

Haoran Shen, Chen Zeng, Jiahui Wang, and Qiao Wang. Reduced effectiveness of kolmogorov-arnold networks on functions with noise, 2024. URL https://arxiv.org/abs/2407.14882.

Samuel Henrique Silva and Peyman Najafirad. Opportunities and challenges in deep learning adversarial robustness: A survey, 2020. URL https://arxiv.org/abs/2007.00753.

Draghici S Sprecher David A. Space-filling curves and kolmogorov superposition-based neural networks, 2002.

Tianze Tang, Yanbing Chen, and Hai Shu. 3d u-kan implementation for multi-modal mri brain tumor segmentation, 2024. URL https://arxiv.org/abs/2408.00273.

Vincent Tjeng, Kai Y. Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyGIdiRqtm.

Jie Wang, Victor Magron, and Jean-Bernard Lasserre. Chordal-tssos: A moment-sos hierarchy that exploits term sparsity with chordal extension. *SIAM Journal on Optimization*, 31(1):114–141, January 2021. ISSN 1095-7189. doi: 10.1137/20m1323564. URL http://dx.doi.org/10.1137/20M1323564.

Tsui-Wei Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Duane Boning, Inderjit S. Dhillon, and Luca Daniel. Towards fast computation of certified robustness for relu networks, 2018. URL https://arxiv.org/abs/1804.09699.

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, 2024. URL https://arxiv.org/abs/2406.01034.

Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh. Automatic perturbation analysis for scalable certified robustness and beyond. *Advances in Neural Information Processing Systems*, 33, 2020.

Chen Zeng, Jiahui Wang, Haoran Shen, and Qiao Wang. Kan versus mlp on irregular or noisy functions, 2024. URL https://arxiv.org/abs/2408.07906.

A APPENDIX

A.1 PROOFS

proof of Lemma 3.3. Let $y, z \in B(x, r)$. It is

$$\frac{d}{dx}h(x) = \begin{pmatrix} \frac{\partial}{\partial x_1}h(x) \\ \cdot \\ \cdot \\ \cdot \\ \frac{\partial}{\partial x_m}h(x) \end{pmatrix} \quad .$$

With

$$\begin{aligned} |h(y) - h(z)| &= |\int_0^1 \frac{d}{dt}h(y + t(z - y))dt| = |\int_0^1 \frac{d}{dx}h(y + t(z - y))(z - y)dt| \\ &\leq \int_0^1 \max_{0 \leq t \leq 1} \max_{1 \leq i \leq m} |\frac{d}{dx}h(y + t(z - y))| \, ||z - y|| dt \\ &\leq \max_{z \in B(x,r)} \max_{1 \leq i \leq m} (|\frac{\partial}{\partial x_1}h(z)|, \ldots, |\frac{\partial}{\partial x_m}h(z)|) \, ||z - y|| \end{aligned}$$

follows

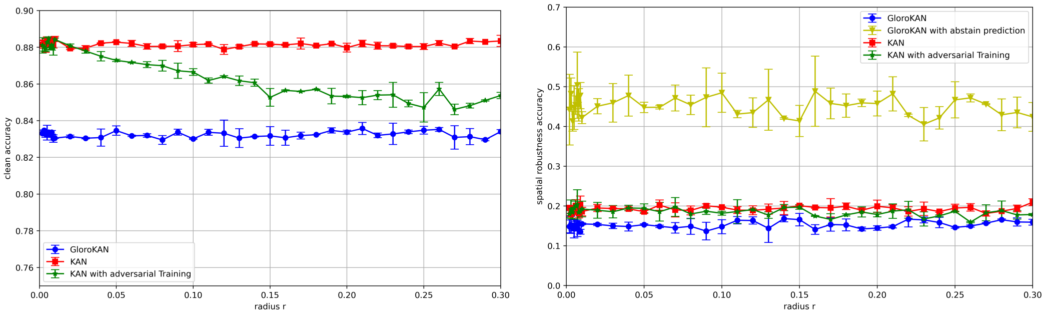
$$L_{x,r} \leq \max_{z \in B(x,r)} \max_{1 \leq i \leq m} (|\frac{\partial}{\partial x_1}h(z)|, \ldots, |\frac{\partial}{\partial x_m}h(z)|) \quad .$$

From

$$\max_{z \in B(x,r)} \max_{1 \leq i \leq m} (|\frac{\partial}{\partial x_1}h(z)|, \ldots, |\frac{\partial}{\partial x_m}h(z)|) \overset{r \to 0}{\to} \max_{1 \leq i \leq m} (|\frac{\partial}{\partial x_1}h(x)|, \ldots, |\frac{\partial}{\partial x_m}h(x)|)$$

follows the claim. □

A.2 FIGURES



(a) (b)

Figure 2: clean and spatial accuracy

A.3 Benchmark Results

Table 5: clean accuracy Fashion MNIST, **NO** adversarial training

	MLP	**KAN**	**CNN**	**CKAN**
r = 0.3	0.8994 ± 0.0011	0.8915 ± 0.0009	0.9165 ± 0.0024	0.8908 ± 0.0019
r = 0.1	0.8980 ± 0.0007	0.8895 ± 0.0010	0.9172 ± 0.0012	0.8919 ± 0.0021
r = 0.05	0.8982 ± 0.0013	0.8905 ± 0.0016	0.9158 ± 0.0013	0.8900 ± 0.0023
r = 0.01	0.9000 ± 0.0009	0.8909 ± 0.0009	0.9169 ± 0.0013	0.8918 ± 0.0035

Table 6: clean accuracy Fashion MNIST, **with** adversarial training

	MLP	**KAN**	**CNN**	**CKAN**
r = 0.3	0.8773 ± 0.0051	0.8716 ± 0.0056	0.9161 ± 0.0047	0.8652 ± 0.0109
r = 0.1	0.8805 ± 0.0025	0.8785 ± 0.0009	0.9134 ± 0.0027	0.8540 ± 0.0064
r = 0.05	0.8901 ± 0.0019	0.8880 ± 0.0021	0.9171 ± 0.0014	0.8746 ± 0.0026
r = 0.01	0.8913 ± 0.0035	0.8881 ± 0.0044	0.9202 ± 0.0014	0.8930 ± 0.0057

Table 7: spatial robustness accuracy Fashion MNIST, **NO** adversarial training

	MLP	**KAN**	**CNN**	**CKAN**
r = 0.3	0.2072 ± 0.0051	0.1816 ± 0.0038	0.1838 ± 0.0124	0.2163 ± 0.0313
r = 0.1	0.2039 ± 0.0095	0.1759 ± 0.0054	0.1765 ± 0.0223	0.2189 ± 0.0216
r = 0.05	0.2031 ± 0.0087	0.1821 ± 0.0089	0.1702 ± 0.016	0.2270 ± 0.0315
r = 0.01	0.2025 ± 0.0034	0.1798 ± 0.0070	0.1768 ± 0.0132	0.2250 ± 0.0088

Table 8: spatial robustness accuracy Fashion MNIST, **with** adversarial training

	MLP	**KAN**	**CNN**	**CKAN**
r = 0.3	0.2763 ± 0.0157	0.1589 ± 0.0159	0.1883 ± 0.0338	0.1978 ± 0.0133
r = 0.1	0.2387 ± 0.0225	0.1695 ± 0.0125	0.2018 ± 0.0155	0.2080 ± 0.0125
r = 0.05	0.2445 ± 0.0183	0.1675 ± 0.0253	0.1931 ± 0.0130	0.2183 ± 0.0160
r = 0.01	0.2425 ± 0.0307	0.1749 ± 0.0169	0.1804 ± 0.0256	0.1923 ± 0.0225

Table 9: clean accuracy CIFAR10, **NO** adversarial training

	MLP	**KAN**	**CNN**	**CKAN**
r = 0.3	0.5497 ± 0.0045	0.5068 ± 0.0022	0.8753 ± 0.0022	0.8692 ± 0.0032
r = 0.1	0.5513 ± 0.0009	0.5090 ± 0.0012	0.8759 ± 0.0013	0.8692 ± 0.0027
r = 0.05	0.5512 ± 0.0036	0.5124 ± 0.0036	0.8741 ± 0.0035	0.8704 ± 0.0017
r = 0.01	0.5545 ± 0.0027	0.5108 ± 0.0039	0.8768 ± 0.0031	0.8680 ± 0.0020

Table 10: clean accuracy CIFAR10, **with** adversarial training

	MLP	**KAN**	**CNN**	**CKAN**
r = 0.3	0.5410 ± 0.0061	0.4252 ± 0.0060	0.8510 ± 0.0024	0.8312 ± 0.0059
r = 0.1	0.5487 ± 0.0049	0.4734 ± 0.0050	0.8062 ± 0.0021	0.7862 ± 0.0037
r = 0.05	0.5515 ± 0.0060	0.4921 ± 0.0046	0.8297 ± 0.0014	0.8141 ± 0.0015
r = 0.01	0.5261 ± 0.0040	0.4869 ± 0.0035	0.8660 ± 0.0043	0.8500 ± 0.0026

Table 11: spatial robustness accuracy CIFAR10, **NO** adversarial training

	MLP	**KAN**	**CNN**	**CKAN**
r = 0.3	0.2499 ± 0.0066	0.2199 ± 0.0046	0.3404 ± 0.0063	0.3347 ± 0.0066
r = 0.1	0.2518 ± 0.0040	0.2164 ± 0.0058	0.3414 ± 0.0056	0.3351 ± 0.0083
r = 0.05	0.2492 ± 0.0034	0.2177 ± 0.0069	0.3372 ± 0.0106	0.3376 ± 0.0040
r = 0.01	0.2502 ± 0.0050	0.2166 ± 0.0053	0.3402 ± 0.0115	0.3311 ± 0.0022

Table 12: spatial robustness accuracy CIFAR10, **with** adversarial training

	MLP	**KAN**	**CNN**	**CKAN**
r = 0.3	0.2537 ± 0.0014	0.2315 ± 0.0084	0.3293 ± 0.113	0.3196 ± 0.0088
r = 0.1	0.2606 ± 0.0174	0.2459 ± 0.0045	0.3206 ± 0.0026	0.3118 ± 0.0058
r = 0.05	0.2439 ± 0.0058	0.2364 ± 0.0012	0.3241 ± 0.0038	0.3141 ± 0.0082
r= 0.01	0.2312 ± 0.0084	0.2133 ± 0.0132	0.3259 ± 0.0060	0.3225 ± 0.0097