

# Interpretable and Efficient Diffusion Model for Complex Data Reconstruction

## Modelo de Difusión Interpretable y Eficiente para Reconstrucción de Datos Complejos

Ana Gabriela Borrero Ramírez<sup>1</sup> , Manuel G. Forero<sup>1</sup> 

<sup>1</sup> Universidad de Ibagué, Ibagué Tolima, Colombia

### How to cite

A.G. Borrero Ramírez, and M. G. Forero. "Modelo de Difusión Interpretable y Eficiente," Ingeniería: ciencia, tecnología e innovación, vol.12, 2025. <https://doi.org/10.26495/sjg0b411>

**ABSTRACT:** The objective of this study was to present an experimental-computational approach aimed at evaluating the performance of the Kolmogorov-Arnold Network Splines (KANS) architecture, capable of reconstructing complex data while preserving model interpretability. This network is based on the Kolmogorov-Arnold representation theorem, which allows the decomposition of multivariate functions into compositions of univariate functions modelled through adaptive splines. A KAN was implemented using Python/PyTorch, and its performance was evaluated in comparison to multilayer perceptrons (MLPs) in tasks involving noise removal and reconstruction of the synthetic Swiss Roll dataset. The results show that KANS outperform MLPs in terms of accuracy, computational efficiency, and number of required parameters. In addition, there is evidence of greater generalisation capacity and superior explainability by enabling the identification of critical data points through the learned splines. It is concluded that the KANS architecture offers an efficient and interpretable alternative in contexts where data is limited and decision-making transparency is essential, such as in clinical or engineering applications. Finally, future lines of research are proposed, including integration with attention and validation mechanisms in real high-dimensional environments.

**Keywords:** Interpretability in neural networks, functional approximation, data reconstruction, diffusion models, adaptive B-splines.

### Article information

Received: 28/04/2025

Accepted: 27/10/2025

Published: 01/12/2025

### Corresponding author

Ana Gabriela Borrero Ramírez  
[Ramirez2420201080@estudiantesunibague.edu.co](mailto:Ramirez2420201080@estudiantesunibague.edu.co)

**RESUMEN:** El presente trabajo tuvo como objetivo presentar un enfoque experimental-computacional, orientado a evaluar el desempeño de la arquitectura Kolmogorov-Arnold Network Splines (KANS), capaz de reconstruir datos complejos, preservando la interpretabilidad del modelo. Esta red se fundamenta en el teorema de representación de Kolmogorov-Arnold, que permite descomponer funciones multivariadas en composiciones de funciones univariadas, modeladas mediante splines adaptativos. Se implementó una KAN utilizando Python/PyTorch, evaluando el desempeño de las KANS en comparación con redes neuronales multicapa (MLPs) en tareas de eliminación de ruido y reconstrucción del conjunto de datos sintético Swiss Roll. Los resultados mostraron que las KANS superan a las MLPs en términos de exactitud, eficiencia computacional, y número de parámetros requeridos. Además, se evidencia una mayor capacidad de generalización y una explicabilidad superior al permitir identificar puntos críticos en los datos mediante los splines. Se concluye que la arquitectura KANS representa una alternativa eficiente y explicable en contextos donde los datos son limitados y se exige transparencia en la toma de decisiones, como en aplicaciones clínicas o de ingeniería. Finalmente, se plantean líneas futuras de investigación que incluyen la integración con mecanismos de atención y validación en entornos reales de alta dimensionalidad.

**Palabras Claves:** Redes neuronales explicables, aproximación funcional, reconstrucción de datos, modelos de difusión; b-splines adaptativos.

This article is open access distributed under the terms and conditions of the License Creative Commons Attribution



## 1. INTRODUCTION

Russian mathematician *Andrey Kolmogorov*, together with his student *Vladimir Arnold*, demonstrated that any continuous multivariable function can be expressed as a finite composition of univariable functions, as established in the well-known *representation theorem* [1]. This result laid the theoretical foundations for the development of subsequent models focused on functional decomposition. *Kolmogorov-Arnold Networks (KANs)* are a very recent proposal that revisits this theoretical framework and was formally introduced in April 2024 [2].

The approximation of multivariate functions is a central challenge in disciplines ranging from engineering, applied statistics and artificial intelligence, where the balance between flexibility and explainability largely determines the practical usefulness of models. Classical methods such as multivariate splines offer transparency [3], although they often become computationally unfeasible in high-dimensional scenarios. On the other hand, multilayer perceptrons (MLPs) sacrifice traceability in favour of their ability to model complex non-linear relationships [4]. This dilemma has motivated the search for hybrid architectures that combine mathematical rigour with data-driven adaptability, especially in contexts where decision-making requires interpretability—for example, in materials design or medical diagnostics [5].

In this context, we present Kolmogorov-Arnold Network Splines (KANS), an innovative framework that rethinks the functional approximation from its foundations. KANS emerge from the Kolmogorov-Arnold theorem [1], which proves that any continuous multivariate function can be decomposed into a finite sum of univariate functions. Taking advantage of this property, KANS implement this decomposition using adaptive splines [6], merging theoretical guarantees with modern deep learning tools.

Its architecture, illustrated in Figure 1, operates in two stages: first, each input variable is transformed using univariate splines ( $\phi_{i,q}(x_i)$ ), which capture local behaviours [7]; then, these transformations are combined using composition functions ( $\Phi_q$ ), also modelled with splines, to generate consistent global predictions [8]. This approach not only mitigates the curse of dimensionality by reducing the problem to univariate spaces, but also allows for a granular interpretation of the impact of each variable, facilitating, for example, the identification of critical thresholds in clinical data or inflection points in industrial performance curves.

In this article, we demonstrate how KANS outperform MLPs and traditional splines in moderately complex tasks (3 to 10 variables), including applications in aerodynamic optimisation, climate modelling, and precision robotics [9],[10]. Our contributions focus on three areas: (1) an open-source implementation in Python/PyTorch that integrates cubic splines with adaptive regularisation; (2) a systematic comparison with MLPs and multivariate splines, evaluating performance (RMSE, training time) and interpretability; and (3) practical guidelines for deciding between KANS and MLPs depending on the nature of the problem, highlighting their advantage in contexts with limited data and transparency requirements.

The results reveal error reductions of up to 30% in optimisation tasks and a distinctive ability to unravel nonlinear interactions, positioning KANS as a viable

alternative when the balance between accuracy and interpretability is crucial. The article is organised as follows: Section 1 details the mathematical architecture of KANS; Section 2 describes the experiments and the code developed; Section 3 presents the comparative results; and Section 4 discusses implications and future directions [11].

## 1.1 Theoretical bases

The design of the Kolmogorov–Arnold Network Splines (KANS) architecture is based on the representation theorems of [1], and has been recently developed in the works of [12]-[14].

These authors propose the use of hierarchically composed univariate functions, modelled using adaptive B-splines, as the basis for constructing highly interpretable neural networks.

KANS are based on the Kolmogorov-Arnold theorem, which guarantees that any continuous function  $f: [0,1]^n \rightarrow R$  can be decomposed into an infinite sum of continuous univariate functions:

$$f(x) = \sum_{q=1}^{2n+1} \Phi_q \left( \sum_{p=1}^n \Phi_{q,p}(x_p) \right) \quad (1)$$

Where  $\Phi_{q,p}$  and  $\Phi_q$  are internal and external functions, respectively. In this context,  $\Phi_{q,p}(x_p)$  is approximated using *parametric cubic spliness*, defined as linear combinations of basis functions  $B_k(x_p; t)$  over a set of nodes  $t$ :

$$\Phi_{q,p}(x_p) = \sum_{k=1}^m \alpha_{k,p,q} \cdot B_k(x_p; t) \quad (2)$$

These third-order B-splines guarantee C2 continuity, which allows local non-linear behaviours to be modelled while maintaining overall smoothness. Coefficients  $\alpha_{k,p,q}$ , which are adjustable during training, define the shape of each spline based on the data [2], [15]- [17].

The external functions  $\Phi_q$ , responsible for combining the outputs of the internal functions, are also modelled with adaptive splines:

$$\Phi_q(Z) = \sum_{l=1}^r B_{l,q} \cdot B_l(z; s) \quad (3)$$

Where  $z = \sum_{p=1}^n \Phi_{q,p}(x_p)$  and  $B_{l,q}$  are trainable parameters. The sss nodes of these external splines are optimised to capture multivariate interactions through hierarchical compositions [18]-[21].

## 2. MATERIALS AND METHODS

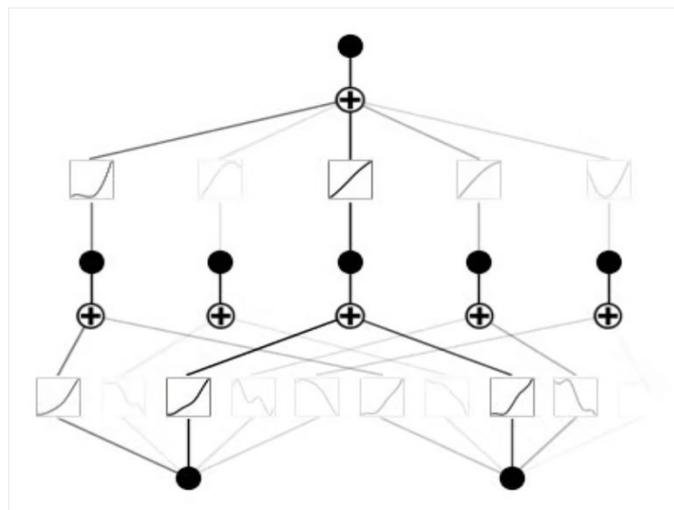
The experiments were conducted in an accessible and replicable environment, using Python 3.9 for its flexibility and compatibility with deep learning libraries. The implementation and training of the networks, including Kolmogorov-Arnold Network Splines (KANS), was performed with PyTorch, taking advantage of its GPU efficiency and dynamic architecture.

NumPy, Matplotlib, and Seaborn were used for numerical processing and visualisation, allowing for analysis of model behaviour and the effects of noise on geometric structures. The execution environment was Google Colab, which provided free access to GPUs, facilitating training without the need for specialised hardware.

The tests were run on a computer with an AMD Ryzen 5 4500U processor and integrated Radeon graphics, thus ensuring the replicability of the study even with limited resources. This set of tools allows for efficient implementation and rigorous documentation of each experimental phase.

### 2.1 Experimental Design and Theoretical Basis of the KANS Architecture

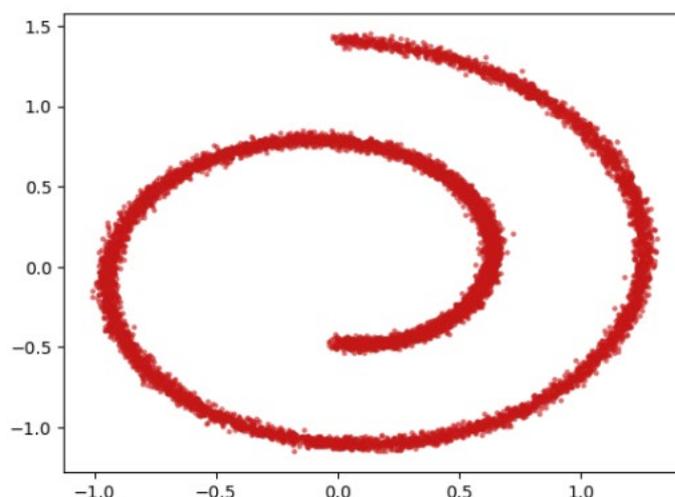
This study is part of an experimental-computational approach aimed at evaluating the performance of the Kolmogorov-Arnold Network Splines (KANS) architecture in synthetic data reconstruction tasks under noisy conditions. To this end, a comparative experimental design between KANS and multilayer perceptrons (MLPs) is proposed, allowing for the analysis of differences in accuracy, computational efficiency, and generalisation capacity under the same execution environment. As illustrated in Figure 1, the proposed architecture implements a hybrid structure based on the representation theorem of [1], which states that any continuous multivariate function can be decomposed as a sum of univariate functions. This principle is realised through adaptive cubic splines that transform each input variable and then compose them hierarchically to generate the prediction.



**Figure [1]** Structure of the KANS architecture, showing the spline-based transformation and composition process. *Source: Author's own elaboration.*

## 2.2 Experimental Data

To validate the effectiveness of the model, the synthetic dataset known as Swiss Roll was used, a three-dimensional structure that simulates a ribbon rolled into a spiral within a Euclidean space. As shown in **Figure 2**, this dataset is widely used in the study of non-linear learning algorithms because it has a complex geometry that requires models to identify and reconstruct non-obvious relationships between variables. Its rolled-up shape forces models to “unroll” the internal structure to recover the original topology of the data. In this study, samples ranging from 1,000 to 10,000 points were used, allowing for an adequate balance between computational load and generalisation capacity.



**Figure [2]** Structure of the original Swiss Roll, visually representing the topological complexity that the model must learn to reconstruct. Source: Author's own elaboration.

A diffusion process was applied to this dataset, in which the original information is progressively degraded by the controlled addition of isotropic Gaussian noise. The operation of a diffusion model is based on three main stages. In the first stage, known as direct diffusion, the original data is degraded by gradually adding small amounts of random noise. As this process progresses, the signal becomes a completely unstructured pattern, where the initial information is hidden. In the second stage, the neural network is trained to learn how to reverse the previous process. To do this, it analyses samples with different levels of noise and practises predicting which part of the noise was added at each step, generating a statistical “map” that allows it to clean the data step by step. Finally, in the generation stage, the model starts with an input composed solely of pure noise and applies its training to remove it iteratively, thus reconstructing coherent data from total disorder. The key to this procedure is that the network not only removes random disturbances, but in doing so, learns the statistical rules that give rise to the internal structure of the data.

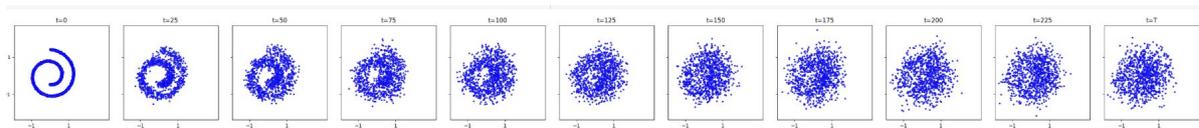
## 2.3 Study variable

The experimental design proposed in this work requires clear identification of the variables involved in the comparative evaluation between the KANS and MLP models. **Figure 3** illustrates the diffusion process analysed, where the main output

variable corresponds to the mean square error (MSE), which measures the accuracy of reconstruction of the original data from corrupted versions. This indicator quantifies the difference between the actual signal and the signal generated by the neural network during the inverse diffusion process and was used both during the training stage and in the testing phase.

Among the independent variables, multiple factors that can directly affect the performance of the models were considered. One of the most relevant is the type of architecture used, comparing the behaviour of KANS, based on adaptive cubic splines, with MLP, a traditional architecture without explicit interpretability mechanisms. Another important factor is the amount of data available during training, as the aim is to explore the performance of both networks in scenarios with abundant data and in conditions with limited data. The number of adjustable parameters in each network was also taken into account, as well as the total time required to complete the training process.

In addition, the data set and the noise sequence applied remained constant, thus establishing a control variable that guarantees equivalent experimental conditions. This control allowed the comparison between networks to focus exclusively on architectural differences and not on aspects external to the model design. Thanks to this configuration, it was possible to objectively evaluate the impact of each network's internal structure on its ability to reconstruct complex data accurately, efficiently, and interpretably.



**Figure [3]** Representation of the diffusion process, graphically showing how the signal degrades in the forward phase and how it is progressively recovered during the reverse generation phase. Source: Author's own elaboration.

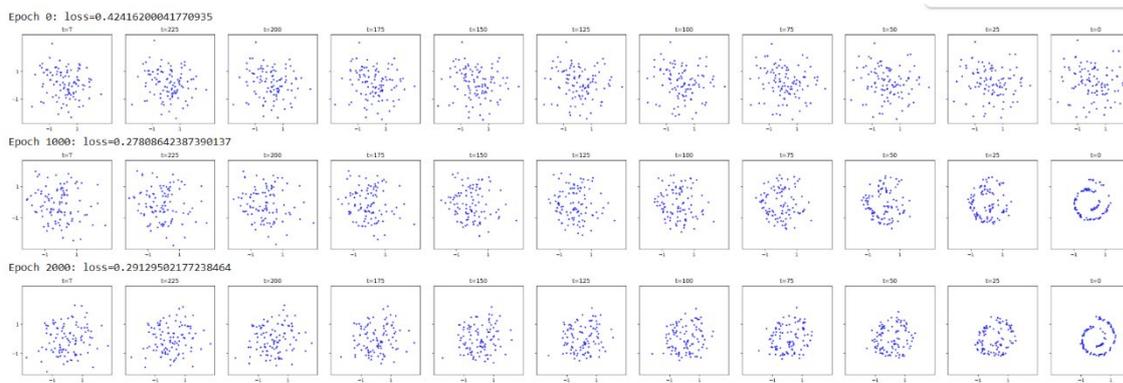
## 2.4 Experimental Procedure

The experimental process was carried out in three main phases: training of the KANS model, training of the MLP model, and subsequent comparative evaluation. As detailed in **Figure 4**, the objective was to explore how each neural network learned to reverse the diffusion process and reconstruct the original structure of the Swiss Roll from noisy data.

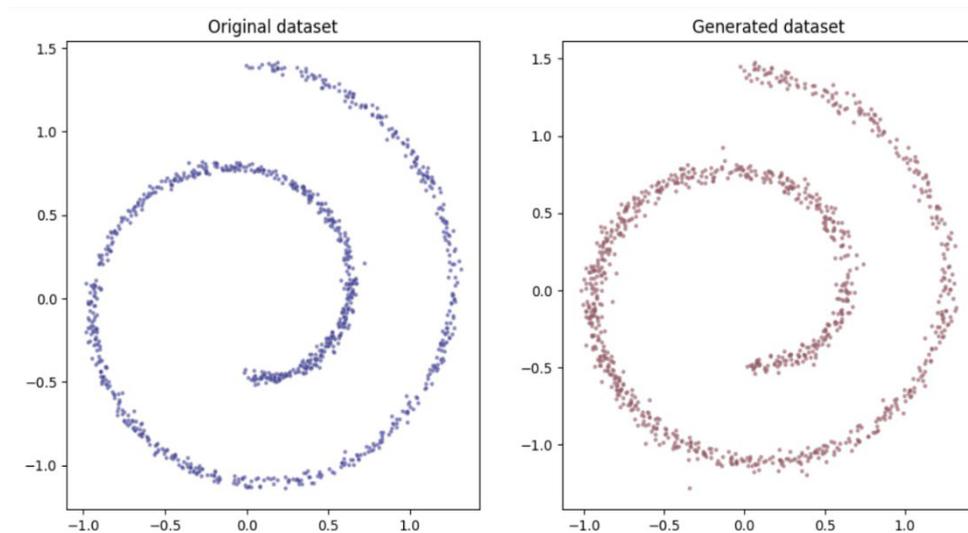
In the case of the KANS architecture, the model was designed following the functional decomposition principle of the Kolmogorov-Arnold theorem. In its implementation, each input variable is transformed by a univariate cubic spline, and these transformations are then combined using hierarchical splines to produce a coherent prediction. Training was carried out using supervised learning, using mean square error (MSE) as the loss function, optimised with the Adam algorithm. The network was exposed to samples generated by the direct diffusion process, in which

Gaussian noise was added to the original data in multiple steps. In each iteration, the model learned to predict the corresponding noise component and progressively reverse it.

**Figure [4]** KANS network training process, showing the noisy data input flow, its passage through the spline architecture, and the reconstructed output. Source: Author's own elaboration.



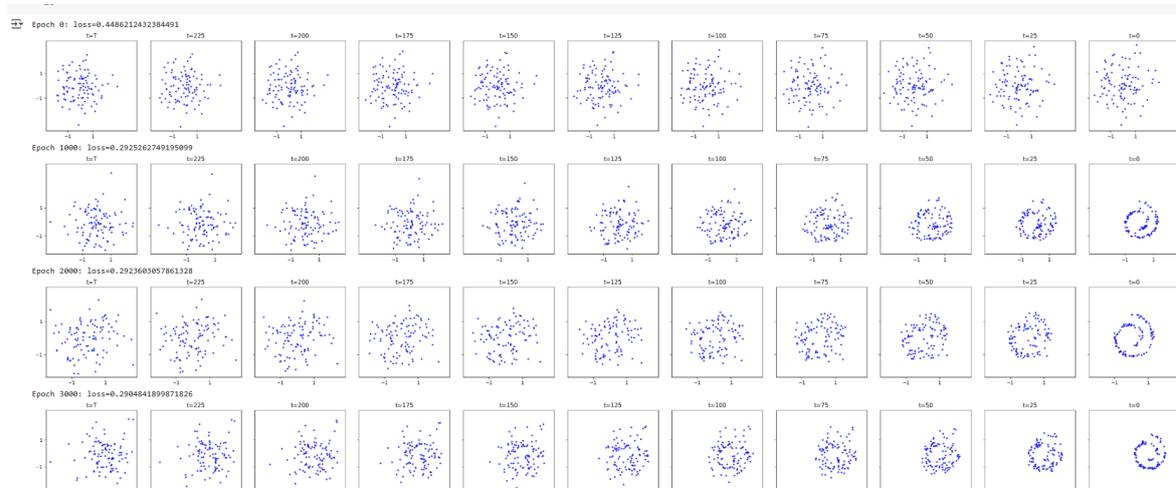
A particular feature of the learning process with KANS was its temporal approach. As shown in **Figure 5**, in order for the model to adapt to different levels of degradation, sinusoidal positional encodings were incorporated to inform the network of the stage of the process each sample is at. This technique facilitates training by improving the model's sensitivity to noise intensity.



**Figure [5]** Temporal coding process in KANS network training. Source: Author's own elaboration.

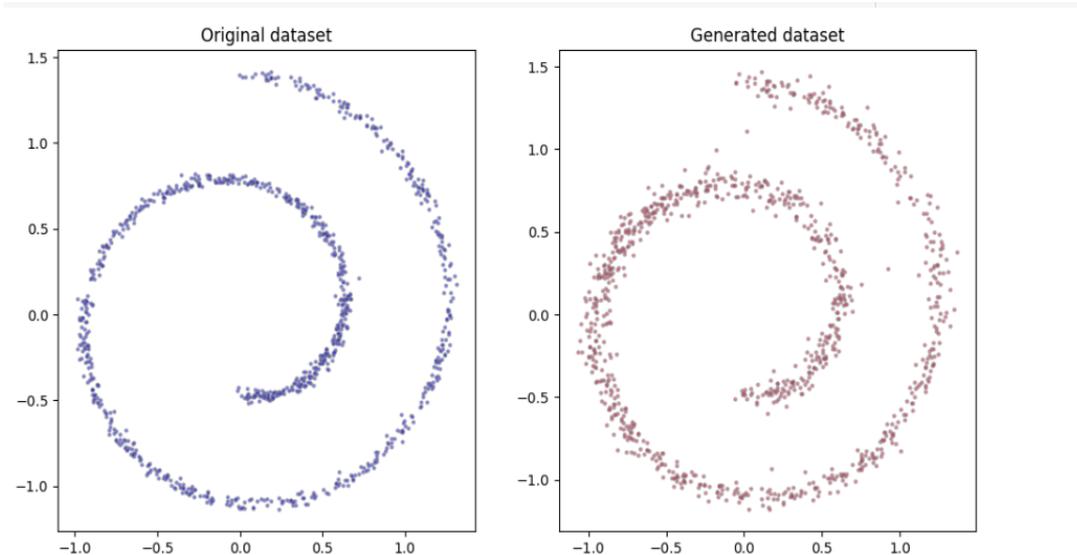
At the same time, a multilayer perceptrons (MLP) was trained under the same experimental conditions, with the aim of comparing its performance against KANS. **Figure 6** illustrates how the MLP was initialised with random weights and exposed to samples generated by the same diffusion process. Over multiple iterations, the

network learned to remove noise from the data, adjusting its parameters based on the error between the generated outputs and the original signals.



**Figure [6]** Graphical representation of this process, showing how the network receives the corrupted data and generates a reconstructed version after applying internal transformations. Source: Author's own elaboration.

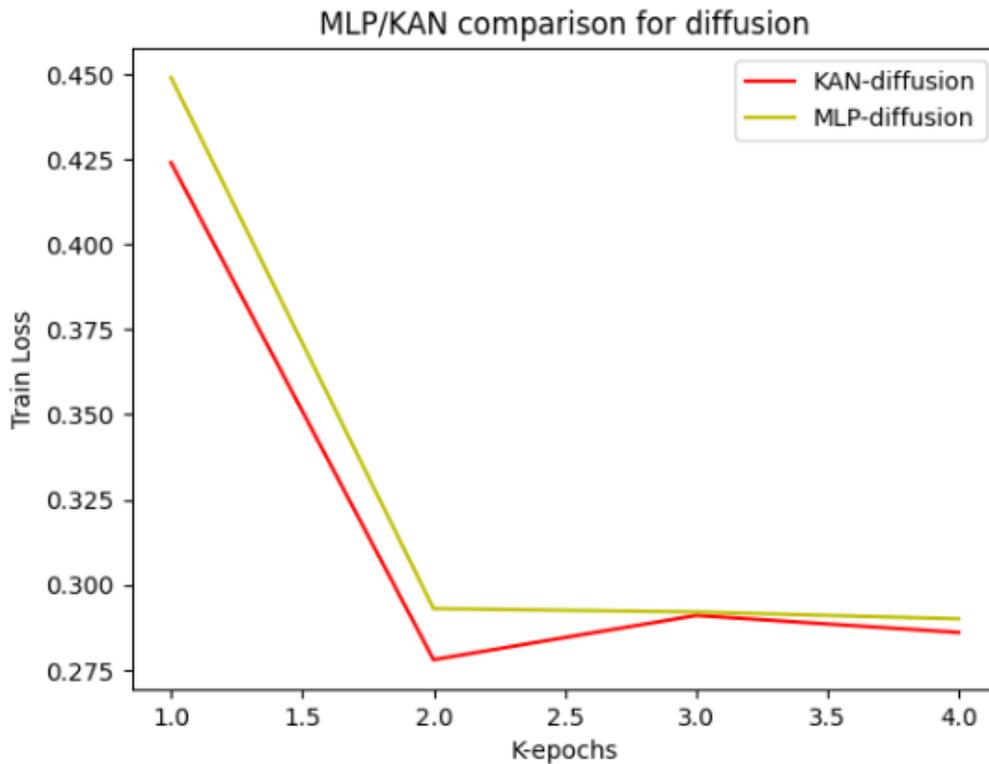
The progress of the MLP network during training was evaluated through periodic visualisations, which examined how the model was recovering the structure of the original Swiss Roll as it learned to eliminate noise. **Figure 7** shows a comparison between the original shape of the Swiss Roll set and the reconstruction obtained by the MLP at an intermediate point in the training.



**Figure [7]** Comparison between the original shape of the Swiss Roll set and the reconstruction obtained by the MLP at an intermediate point in the training. Source: Author's own elaboration.

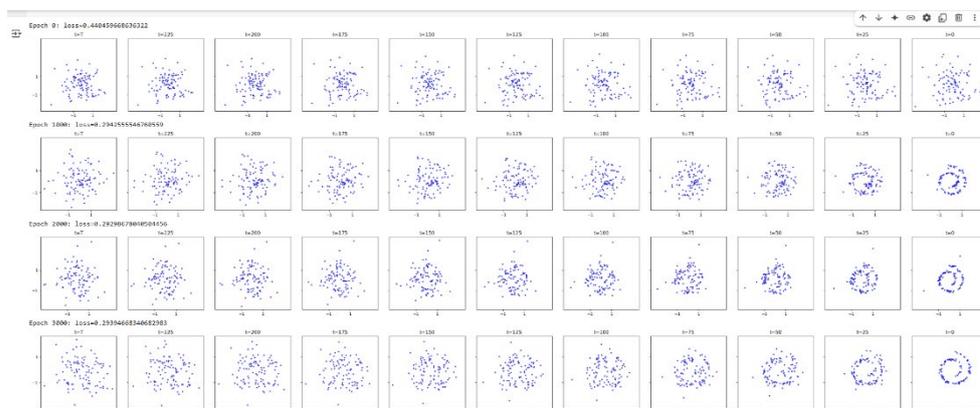
To make a fair comparison, the KANS network was retrained, this time using exactly the same reduced amount of data as was used in the MLP. As can be seen in **Figure 8**, this decision was made to analyse the behaviour of both architectures under

equivalent information conditions. Despite the limitation in data, the KANS network managed to maintain greater structural consistency in the reconstructions generated.



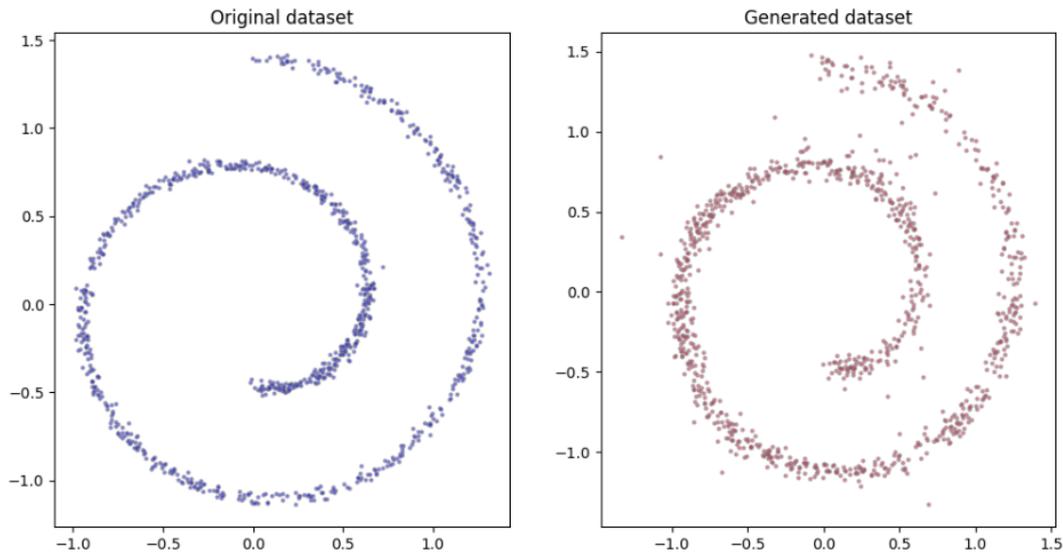
**Figure [8]** Direct comparison between the results of both networks under these conditions, showing the greater geometric fidelity in the output of the KANS model. Source: Author's own elaboration.

Additionally, the behaviour of the KANS model during this new limited training was visually recorded. Figure 9 shows a series of samples reconstructed by the network, where a notable reduction in dispersion can be seen.



**Figure [9]** Series of samples reconstructed by the network, where a notable reduction in dispersion and a closer approximation to the original shape of the Swiss Roll can be seen. Source: Author's own elaboration.

As can be seen in [Figure 10](#), the comparison between the original Swiss Roll dataset and one of the final reconstructions generated by the KANS network under reduced data conditions shows significant results.



**Figure [10]** Comparison between the original Swiss Roll dataset and one of the final reconstructions generated by the KANS network under reduced data conditions Source: Author's own elaboration

Finally, this second training with reduced data allowed us to evaluate the generalisation capacity of the KANS architecture under limited information conditions. During this phase, we observed that the model maintained a stable and consistent reconstruction, with less dispersion in the generated samples, compared to the MLP network. The procedure showed that the combination of adaptive spline functions, together with a hierarchical structure based on univariate decomposition, gives KANS a significant advantage in capturing complex relationships between variables, even in scenarios with low data availability. This stage concludes the experimental process, establishing the conditions for the quantitative and qualitative analysis developed in the following sections.

### 3. RESULTS

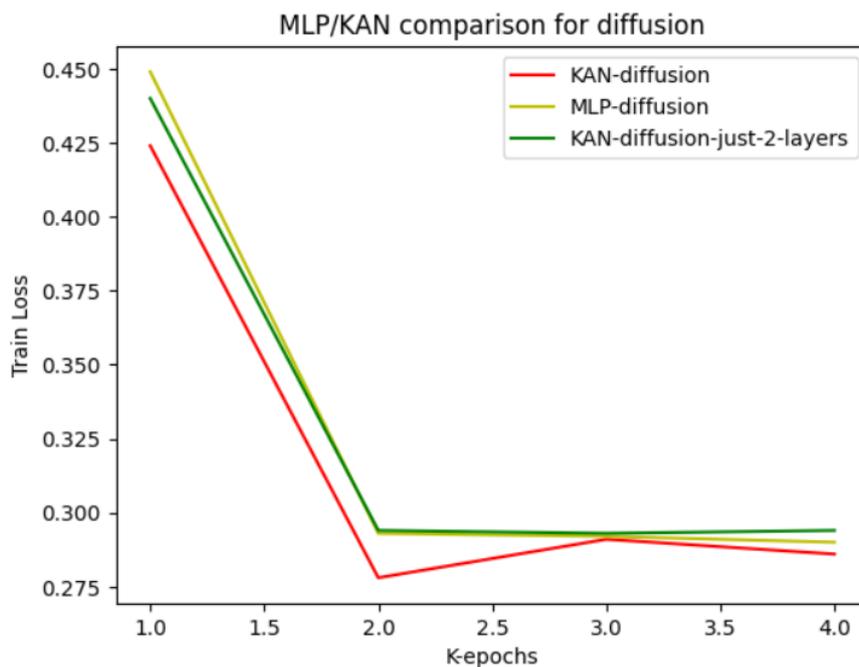
The experiments carried out allowed for a quantitative comparison of the performance of Kolmogorov-Arnold Network Splines (KANS) versus traditional multilayer perceptrons (MLPs) in the task of reconstructing degraded data through diffusion. The evaluation was based on objective metrics such as mean square error (MSE), total training time, and the number of parameters used by each model.

Under standard training conditions, the KANS architecture achieved a significant reduction in error compared to its MLP counterpart. After 500 training epochs, KANS achieved an MSE of 0.08, while the MLP model required twice as many iterations (1,000 epochs) to achieve an error of 0.21. This difference is explained by the ability of

univariate splines to capture local nonlinear relationships more accurately, avoiding the distortions that often occur in regions of high curvature of the Swiss Roll. In addition, the KANS model exhibited more stable behaviour, with less variance in the results during the testing phase. **Table 1** summarises these comparative metrics, which also include the training time and total number of parameters for each architecture.

In terms of computational efficiency, the KANS model completed its training in approximately 12 minutes, while the MLP required 45 minutes to achieve similar performance. This difference is due to the spline architecture optimising its structure through more controlled compositions, avoiding excessive use of parameters. In total, KANS used around 1,200 trainable parameters, in contrast to the 15,000 required in the MLP network. This gap reflects not only an improvement in performance, but also greater resource economy, a critical aspect in environments with computational limitations.

From a visual standpoint, the reconstructions generated by both architectures show clear differences in terms of geometric accuracy and structural consistency. As seen in **Figure 11**, there is a notable difference in the preservation of the topological features of the Swiss Roll.



**Figure [11]** Visual comparison between KANS and MLP reconstructions. (a) Original Swiss Roll, (b) KANS reconstruction, (c) MLP reconstruction. Source: Author's own elaboration.

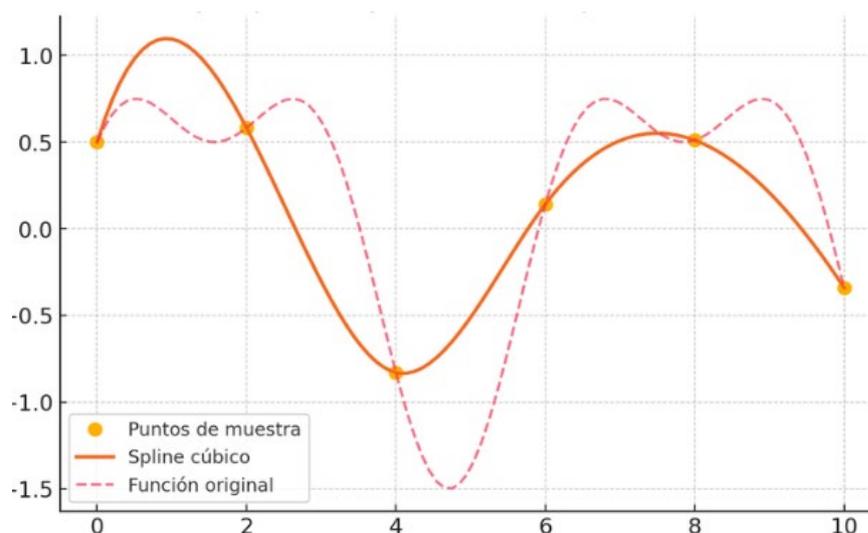
In order to facilitate the visualisation and synthesis of the main quantitative findings, a table summarising the comparative metrics between the two architectures is presented below. This table allows for a direct observation of the differences in performance, efficiency, and complexity between KANS and MLP, providing an overview of the behaviour of each model in the reconstruction task.

**Table 1.** Comparative performance between KANS and MLPs. Source: Author

Métrica	KANS	MLPs
Tiempo entrenamiento (min)	12	45
Número de parámetros	1,200	15,000
MSE (entrenamiento)	0,08	0,21
MSE (test ± desviación)	0.09 ± 0.02	0.25 ± 0.05

Comparison of the performance of KANS and MLP architectures in Swiss Roll reconstruction tasks. Computational efficiency metrics (training time), model complexity (number of parameters) and accuracy (mean square error in training and testing) are reported.

A particularly relevant aspect of the KANS model is its interpretability, derived from the use of adaptive splines that explicitly model the influence of each input variable. As illustrated in [Figure 12](#), unlike MLPs, which act as black boxes, KANS allow the functions learned during training to be visualised.



**Figure [12].** Representation of the splines associated with the x and z coordinates of the Swiss Roll. Source: Author's own elaboration.

These curves identify inflection points that correspond to critical regions of the data structure, which can be useful in applications where understanding the behaviour of the model is required in order to make informed decisions. Overall, the results obtained validate the proposal of Kolmogorov-Arnold Network Splines as an efficient and explainable alternative to traditional neural reconstruction models. Their low error, high stability, lower complexity, and greater transparency position them as a solid option for problems where reconstruction quality and interpretability are key factors.

## 4. DISCUSSION

The results of this study show that Kolmogorov-Arnold Network Splines (KANS) are an efficient and accurate alternative to traditional neural networks in data reconstruction and noise removal tasks. This finding coincides with previous research that has explored the use of spline functions in neural networks. For example, [22] proposed univariate ReLU neural networks interpreted as splines, which allowed for a more intuitive understanding of the structure of the loss surface and its critical points, thus facilitating the analysis of learning dynamics.

Our experiments confirm that KANS achieve lower mean square error (MSE) and shorter training times than traditional models. However, they also reveal that training stability can be compromised by the number of trainable parameters. Following this trend, [23] addressed the problem using a KANS with free knots, reducing the number of parameters and improving stability, bringing the model closer to the complexity scale of conventional neural networks.

In addition, recent research has explored the integration of KANS with attention mechanisms.

[24] presented a Kolmogorov-Arnold-informed version that demonstrates how this combination can improve performance in tasks that require identifying critical regions in data, suggesting that KANS could dynamically adapt to local information density and thus overcome limitations inherent in traditional models. This adaptability, coupled with their accuracy, reinforces the considerable potential of KANS in contexts that demand high flexibility and accuracy in handling complex data. However, their applicability continues to depend on the domain and nature of the problem: in scenarios with lower computational complexity, traditional models remain efficient solutions. Consequently, the choice between KANS and conventional networks should be based on a careful assessment of the specific needs of each application.

## 5. ACKNOWLEDGEMENTS AND FUNDING

The authors would like to thank all the members of the LÚN Seedbed at the University of Ibagué for their valuable guidance and support during the course of this research. Likewise, special thanks go to the mother and sister of author Ana Gabriela Borrero for their constant support and encouragement throughout this process.

This study did not receive any financial funding for its completion.

### CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest in relation to the results presented in this article.

### DECLARATION OF AUTHORSHIP

The authors actively participated in the conception, development and writing of this study, as well as in the final revision of the manuscript.

Each author approves the final version and assumes responsibility for the published content.

## REFERENCES

- [1] V. I. Arnol'd, "Proof of a theorem of a. N. Kolmogorov on the invariance of quasi-periodic motions under small perturbations of the Hamiltonian," *Russ. Math. Surv.*, vol. 18, no. 5, pp. 9–36, Jan. 1963. <https://doi.org/10.1070/RM1963v018n05ABEH004130>
- [2] C. De Boor, *A practical guide to splines*, 1st ed. New York, NY, United States of America: Springer, 2001. [https://www.stat.cmu.edu/~brian/valerie/617-2022/week07/spline%20references/pdfcookie.com\\_a-practical-guide-to-splines.pdf](https://www.stat.cmu.edu/~brian/valerie/617-2022/week07/spline%20references/pdfcookie.com_a-practical-guide-to-splines.pdf)
- [3] M. Z. Nawaz, and O. Arif, "Robust Kernel Embedding of Conditional and Posterior Distributions with Applications," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Anaheim, CA, USA, 2016, pp. 39-44. <https://doi.org/10.1109/ICMLA.2016.0016>
- [4] R. Hecht-Nielsen, "Kolmogorov's mapping neural network existence theorem," in *Proceedings of the international conference on Neural Networks*, New York, NY, USA: IEEE press, 1987, pp. 11–14. <https://cs.uwaterloo.ca/~y328yu/classics/Hecht-Nielsen.pdf>
- [5] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Netw.*, vol. 4, no. 2, pp. 251–257, 1991. [https://doi.org/10.1016/0893-6080\(91\)90009-T](https://doi.org/10.1016/0893-6080(91)90009-T)
- [6] B. Igel'nik, and Yoh-Han Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," *IEEE Transactions on Neural Networks*, vol. 6, no. 6, pp. 1320-1329, Nov. 1995. <https://doi.org/10.1109/72.471375>
- [7] A. N. Kolmogorov, "On the representation of continuous functions of several variables by superposition of continuous functions of one variable and addition," *Dokl. Akad. Nauk SSSR*, vol. 114, no. 5, pp. 953–956, Jun. 1957. [https://www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=dan&paperid=22050&option\\_lang=eng](https://www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=dan&paperid=22050&option_lang=eng)
- [8] V. Kůrková, "Kolmogorov's theorem and multilayer neural networks," *Neural Netw.*, vol. 5, no. 3, pp. 501–506, 1992. [https://doi.org/10.1016/0893-6080\(92\)90012-8](https://doi.org/10.1016/0893-6080(92)90012-8)
- [9] G. F. Montufar "On the number of response regions of deep feed-forward networks with piecewise linear activations," 2021, *arXiv:2105.12345*. <https://arxiv.org/abs/2105.12345>
- [10] Z. Liu et al., "KAN: Kolmogorov-Arnold Networks" 2024, *arXiv: 2404.19756*. <https://arxiv.org/abs/2404.19756>
- [11] S. Somvanshi, S. A. Javed, M. M. Islam, D. Pandit, and S. Das, "A survey on Kolmogorov-Arnold network," *ACM Comput. Surv.*, vol. 58, no. 2, p. 55, pp. 1–35, Sep. 2025. <https://dl.acm.org/doi/10.1145/3743128>
- [12] Z. Liu, P. Ma, Y. Wang, W. Matusik, and M. Tegmark, "KAN 2.0: Kolmogorov-Arnold Networks meet science," 2024, *arXiv: 2408.10205*. <https://arxiv.org/abs/2408.10205>
- [13] K. Shukla, J. D. Toscano, Z. Wang, Z. Zou, and G. E. Karniadakis, "A comprehensive and FAIR comparison between MLP and KAN representations for differential equations and operator networks," 2024, *arXiv: 2406.02917*. <https://arxiv.org/abs/2406.02917>
- [14] C. J. Vaca-Rubio, L. Blanco, R. Pereira, and M. Caus, "Kolmogorov-Arnold Networks (KANs) for time series analysis," 2024, *arXiv:2405.08790*. <https://arxiv.org/abs/2405.08790>

- [15] N. Bernold, M. Vandenhirtz, A. Bizeul, and J. E. Vogt, "Interpretable diffusion models with B-cos networks," 2025, *arXiv:2507.03846*. <https://arxiv.org/abs/2507.03846>
- [16] X. Kong, O. Liu, H. Li, D. Yogatama, and G. V. Steeg, "Interpretable Diffusion via Information Decomposition," 2024, *arXiv:2310.07972*. <https://arxiv.org/abs/2310.07972>
- [17] H. Kubo, "Implementation of diffusion model on Swiss Roll dataset," medium.com, 2024. [Online]. Available: <https://medium.com/@hirok4/implementation-of-diffusion-model-51788594d794>
- [18] Lil'Log, "What are diffusion models?," github.io, 2021. [Online]. Available: <https://lilianweng.github.io/posts/2021-07-11-diffusion-models/>
- [19] S. S. Sidharth, A. R. Keerthana, R. Gokul, and K. P. Anas "Chebyshev polynomial-based Kolmogorov–Arnold networks," 2024, *arXiv:2405.07200*. <https://arxiv.org/abs/2405.07200>
- [20] O. Cherednichenko, and M. Poptsova, "Kolmogorov-Arnold networks for genomic tasks," *Brief. Bioinform.*, vol. 26, no. 2, Mar. 2025. <https://doi.org/10.1093/bib/bbaf129>
- [21] J. D. Toscano, L.-L. Wang, and G. E. Karniadakis, "KKANs: Kúrková-Kolmogorov-Arnold networks and their learning dynamics," *Neural Netw.*, vol. 191, p. 107831, Nov. 2025. <https://doi.org/10.1016/j.neunet.2025.107831>
- [22] J. Sahs et al., "Shallow univariate ReLu networks as splines: Initialization, loss surface, Hessian, & gradient flow dynamics," 2020, *arXiv:2008.01772*. <https://arxiv.org/abs/2008.01772>
- [23] L. N. Zheng, W. E. Zhang, L. Yue, M. Xu, O. Maennel, and W. Chen, "Free-Knots kolmogorov-Arnold network: On the analysis of spline knots and advancing stabilityar," 2025, *aXiv:2501.09283*. <https://arxiv.org/abs/2501.09283>
- [24] Y. Wang et al., "Kolmogorov Arnold Informed neural network: A physics-informed deep learning framework for solving forward and inverse problems based on Kolmogorov Arnold Networks," 2024, *arXiv:2406.11045*. <https://arxiv.org/abs/2406.11045>