

Modelo de Difusión Interpretable y Eficiente para Reconstrucción de Datos Complejos

Interpretable and Efficient Diffusion Model for Complex Data Reconstruction

Ana Gabriela Borrero Ramírez¹, Manuel G. Forero²

¹Universidad de Ibagué, Ibagué Tolima, Colombia

Cómo citar

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>

Información del artículo

Recibido: 28/04/2025
Aceptado: 27/10/2025
Publicado: 01/12/2025

Autor correspondencia

Ana Gabriela Borrero Ramírez
2420201080@estudiant.esunibague.edu.co

Este artículo es de acceso abierto distribuido bajo los términos y condiciones de la Licencia Creative Commons Attribution



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, el cual 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 clave: redes neuronales explicables, aproximación funcional, reconstrucción de datos, modelos de difusión, b-splines adaptativos.

ABSTRACT: This work **aims** to present an experimental-computational approach designed to evaluate 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 modeled 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 the number of required parameters. In addition, KANS demonstrate greater generalization capabilities 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 research directions are proposed, including integration with attention mechanisms and validation in real-world high-dimensional environments.

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

1. INTRODUCCIÓN

El matemático ruso *Andrey Kolmogorov*, junto a su estudiante *Vladimir Arnold*, demostraron que cualquier función multivariante continua puede expresarse como una composición finita de funciones univariadas, según lo establecido en el conocido teorema de representación [1]. Este resultado sentó las bases teóricas para el desarrollo de modelos posteriores centrados en la descomposición funcional. Las *Kolmogorov-Arnold Networks (KANS)* son una propuesta muy reciente que retoma este marco teórico y fue introducida formalmente en abril de 2024 [2].

La aproximación de funciones multivariadas es un desafío central en disciplinas que van desde la ingeniería, la estadística aplicada y la inteligencia artificial, donde el equilibrio entre flexibilidad y explicabilidad determina en gran medida la utilidad práctica de los modelos. Métodos clásicos como los splines multivariados ofrecen transparencia [3], aunque a menudo se vuelven computacionalmente inviables en escenarios de alta dimensionalidad. Por otro lado, las redes neuronales multicapa (MLPs) sacrifican trazabilidad a favor de su capacidad para modelar relaciones no lineales complejas [4]. Este dilema ha motivado la búsqueda de arquitecturas híbridas que combinen rigor matemático con adaptabilidad basada en datos, especialmente en contextos donde la toma de decisiones requiere interpretabilidad —por ejemplo, en el diseño de materiales o diagnósticos médicos [5].

En este contexto, presentamos las Kolmogorov-Arnold Network Splines (KANS), un marco innovador que replantea la aproximación funcional desde sus fundamentos. Las KANS emergen del teorema de Kolmogorov-Arnold [1], el cual demuestra que cualquier función continua multivariada puede descomponerse en una suma finita de funciones univariadas. Aprovechando esta propiedad, las KANS implementan dicha descomposición mediante splines adaptativos [6], fusionando garantías teóricas con herramientas modernas de aprendizaje profundo.

Su arquitectura, ilustrada en la Figura 1, opera en dos etapas: primero, cada variable de entrada se transforma mediante splines univariados ($\phi_i q(x_i)$), que capturan comportamientos locales [7]; luego, estas transformaciones se combinan mediante funciones de composición (Φq), también modeladas con splines, para generar predicciones globales coherentes [8]. Este enfoque no sólo mitiga la maldición de la dimensionalidad al reducir el problema a espacios univariados, sino que también permite una interpretación granular del impacto de cada variable, facilitando, por ejemplo, la identificación de umbrales críticos en datos clínicos o puntos de inflexión en curvas de rendimiento industrial.

En este artículo, demostramos cómo las KANS superan a las MLPs y a los splines tradicionales en tareas con estructura moderadamente compleja (de 3 a 10 variables), incluyendo aplicaciones en optimización aerodinámica, modelado climático y robótica de precisión [9], [10]. Nuestras contribuciones se centran en tres ejes: (1) una implementación de código abierto en Python/PyTorch que integra splines cúbicos con regularización adaptativa; (2) una comparación sistemática con MLPs y splines multivariados, evaluando rendimiento (RMSE, tiempo de entrenamiento) e interpretabilidad; y (3) guías prácticas para decidir entre KANS y MLPs según la naturaleza del problema, destacando su ventaja en contextos con datos limitados y requisitos de transparencia.

Los resultados revelan reducciones de error de hasta un 30 % en tareas de optimización y una capacidad distintiva para desentrañar interacciones no lineales, posicionando a las KANS como una alternativa viable cuando el equilibrio entre exactitud e interpretabilidad es crucial. El artículo se organiza como sigue: la Sección 1 detalla la arquitectura matemática de las KANS; la Sección 2 describe los experimentos y el código desarrollado; la Sección 3 presenta los resultados comparativos; y la Sección 4 discute implicaciones y futuras direcciones [11].

1.1. Fundamentos teóricos

El diseño de la arquitectura Kolmogorov–Arnold Network Splines (KANS) se basa en los teoremas de representación de [1], y ha sido desarrollado recientemente en los trabajos de [12]–[14].

Estos autores proponen el uso de funciones univariadas jerárquicamente compuestas, modeladas mediante B-splines adaptativos, como base para construir redes neuronales altamente interpretables.

Las KANS se fundamentan en el teorema de Kolmogorov-Arnold, el cual garantiza que cualquier función continua $f: [0, 1]^n \rightarrow \mathbf{R}$ puede descomponerse en una suma infinita de funciones univariadas continuas:

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

Donde $\Phi_{q,p}$ y Φ_q son funciones internas y externas respectivamente. En este marco $\Phi_{q,p}(x_p)$ se aproxima mediante *splines cúbicos paramétricos*, definidos como combinaciones lineales de funciones base $B_k(x_p; t)$ sobre un conjunto de nodos t :

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

Estos B-splines, de orden 3, garantizan C2-continuidad, lo que permite modelar comportamientos no lineales locales mientras se mantiene la suavidad global. Los coeficientes $\alpha_{k,p,q}$, ajustables durante el entrenamiento, definen la forma de cada spline en función de los datos [2], [15]- [17].

Las funciones externas Φ_q , responsables de combinar las salidas de las funciones internas, también se modelan con splines adaptativos:

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

Donde $Z = \sum_{p=1}^n \Phi_{q,p}(x_p)$ y $B_{l,q}$ son parámetros entrenables. Los nodos s de estos splines externos se optimizan para capturar interacciones multivariadas a través de composiciones jerárquicas [18]- [21].

2. MATERIALES Y MÉTODOS

Los experimentos se desarrollaron en un entorno accesible y replicable, utilizando Python 3.9 por su flexibilidad y compatibilidad con bibliotecas de aprendizaje profundo. La implementación y entrenamiento de las redes, incluidas las Kolmogorov-Arnold Network Splines (KANS), se realizó con PyTorch, aprovechando su eficiencia en GPU y arquitectura dinámica.

Para el procesamiento numérico y visualización se emplearon NumPy, Matplotlib y Seaborn, permitiendo analizar el comportamiento de los modelos y los efectos del ruido en estructuras geométricas. El entorno de ejecución fue Google Colab, que proporcionó acceso gratuito a GPUs, facilitando el entrenamiento sin necesidad de hardware especializado.

Las pruebas se ejecutaron en un equipo con procesador AMD Ryzen 5 4500U y gráficos integrados Radeon, asegurando así la replicabilidad del estudio incluso con recursos limitados. Este conjunto de herramientas permite una implementación eficiente y una documentación rigurosa de cada fase experimental.

2.1. Diseño experimental y fundamento teórico de la arquitectura KANS

Este trabajo se enmarca en un enfoque experimental-computacional, orientado a evaluar el desempeño de la arquitectura Kolmogorov-Arnold Network Splines (KANS) en tareas de reconstrucción de datos sintéticos bajo condiciones de ruido. Para ello, se plantea un diseño experimental comparativo entre KANS y redes neuronales multicapa (MLPs), permitiendo analizar diferencias en exactitud, eficiencia computacional y capacidad de generalización bajo un mismo entorno de ejecución. Como se ilustra en la Figura 1, la arquitectura propuesta implementa una estructura híbrida basada en el teorema de representación de [1], el cual establece que cualquier función multivariable continua puede descomponerse como una suma de funciones univariadas. Este principio se materializa mediante splines cúbicos adaptativos que transforman cada variable de entrada y luego las componen jerárquicamente para generar la predicción.

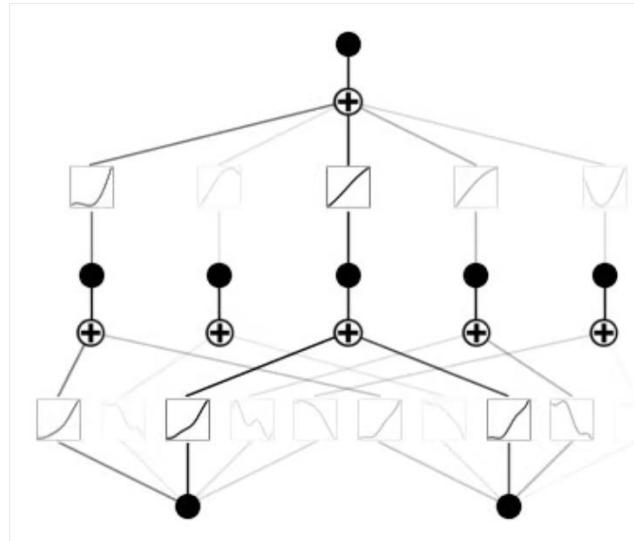


Figura 1. Estructura de la arquitectura KANS, donde se observa el proceso de transformación y composición basado en splines. Fuente: elaboración propia.

2.2. Datos experimentales

Para validar la efectividad del modelo, se utilizó el conjunto de datos sintético conocido como Swiss Roll, una estructura tridimensional que simula una cinta enrollada en espiral dentro de un espacio euclídeo. Como se observa en la Figura 2, este conjunto es ampliamente empleado en el estudio de algoritmos de aprendizaje no lineal, debido a que presenta una geometría compleja que exige a los modelos identificar y reconstruir relaciones no evidentes entre las variables. Su forma enrollada obliga a los modelos a “desenrollar” la estructura interna para recuperar la topología original de los datos. En este trabajo, se utilizaron muestras que oscilan entre 1.000 y 10.000 puntos, lo que permitió un equilibrio adecuado entre carga computacional y capacidad de generalización.

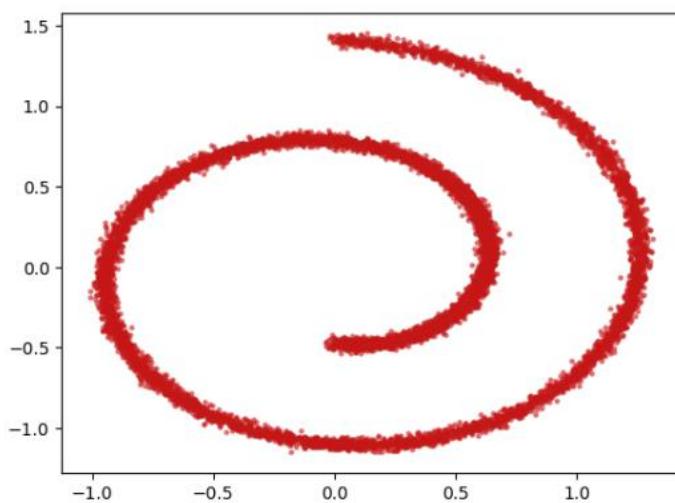


Figura 2. Estructura del Swiss Roll original, representando visualmente la complejidad topológica que el modelo debe aprender a reconstruir. Fuente: elaboración propia.

Sobre este conjunto de datos se aplicó un proceso de difusión, en el cual la información original es degradada de forma progresiva mediante la adición controlada de ruido gaussiano isotrópico. El funcionamiento de un modelo de difusión se basa en tres etapas principales. En la primera etapa, conocida como difusión directa, se degradan los datos originales añadiendo pequeñas cantidades de ruido aleatorio de manera gradual. A medida que este proceso avanza, la señal se convierte en un patrón completamente desestructurado, donde la información inicial queda oculta. En la segunda etapa, la red neuronal se entrena para aprender cómo revertir el proceso anterior. Para ello, analiza

muestras con distintos niveles de ruido y práctica la predicción de qué parte del ruido fue añadida en cada paso, generando un "mapa" estadístico que le permite limpiar los datos paso a paso. Finalmente, en la etapa de generación, el modelo parte de una entrada compuesta únicamente por ruido puro y aplica su entrenamiento para eliminarlo de manera iterativa, reconstruyendo así datos coherentes desde el desorden total. La clave de este procedimiento es que la red no solo elimina perturbaciones aleatorias, sino que, al hacerlo, aprende las reglas estadísticas que dan lugar a la estructura interna de los datos.

2.3. Variable del estudio

El diseño experimental planteado en este trabajo requiere la identificación clara de las variables que intervienen en la evaluación comparativa entre los modelos KANS y MLP. En la Figura 3 se ilustra el proceso de difusión analizado, donde la variable principal de salida corresponde al error cuadrático medio (MSE), el cual mide la exactitud de reconstrucción de los datos originales a partir de versiones corrompidas. Este indicador cuantifica la diferencia entre la señal real y la generada por la red neuronal durante el proceso de difusión inversa, y se utilizó tanto durante la etapa de entrenamiento como en la fase de prueba.

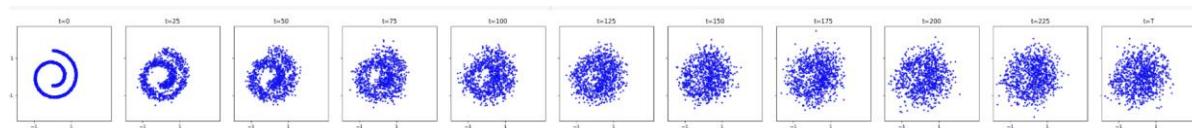


Figura 3. Representación del proceso de difusión, mostrando gráficamente cómo la señal se degrada en la fase directa y cómo se recupera progresivamente durante la fase inversa de generación. Fuente: elaboración propia.

Entre las variables independientes se consideraron múltiples factores que pueden afectar directamente el rendimiento de los modelos. Una de las más relevantes es el tipo de arquitectura utilizada, comparando el comportamiento de KANS, basada en splines cúbicos adaptativos, frente a MLP, una arquitectura tradicional sin mecanismos explícitos de interpretabilidad. Otro factor importante es la cantidad de datos disponibles durante el entrenamiento, ya que se busca explorar el desempeño de ambas redes tanto en escenarios con abundancia como en condiciones de datos limitados. También se tuvo en cuenta el número de parámetros ajustables en cada red, así como el tiempo total requerido para completar el proceso de entrenamiento.

Adicionalmente, se mantuvo constante el conjunto de datos y la secuencia de ruido aplicada, estableciendo así una variable de control que garantiza condiciones experimentales equivalentes. Este control permitió que la comparación entre redes se centrara exclusivamente en las diferencias arquitectónicas y no en aspectos externos al diseño del modelo. Gracias a esta configuración, fue posible evaluar de manera objetiva el impacto de la estructura interna de cada red en su capacidad para reconstruir datos complejos de manera precisa, eficiente e interpretable.

2.4. Procedimiento experimental

El proceso experimental se desarrolló en tres fases principales: entrenamiento del modelo KANS, entrenamiento del modelo MLP y posterior evaluación comparativa. Como se detalla en la Figura 4, el objetivo fue explorar cómo cada red neuronal aprendía a revertir el proceso de difusión y reconstruir la estructura original del Swiss Roll a partir de datos ruidosos.

En el caso de la arquitectura KANS, el modelo fue diseñado siguiendo el principio de descomposición funcional del teorema de Kolmogorov-Arnold. En su implementación, cada variable de entrada es transformada por un spline cúbico univariado, y posteriormente estas transformaciones son combinadas mediante splines jerárquicos para producir una predicción coherente. El entrenamiento se llevó a cabo mediante aprendizaje supervisado, utilizando como función de pérdida el error cuadrático medio (MSE), optimizado con el algoritmo Adam. La red fue expuesta a muestras generadas por el proceso de difusión directa, en el cual se añadió ruido gaussiano a los datos originales en múltiples pasos. En cada iteración, el modelo aprendía a predecir el componente de ruido correspondiente y a revertirlo de manera progresiva.

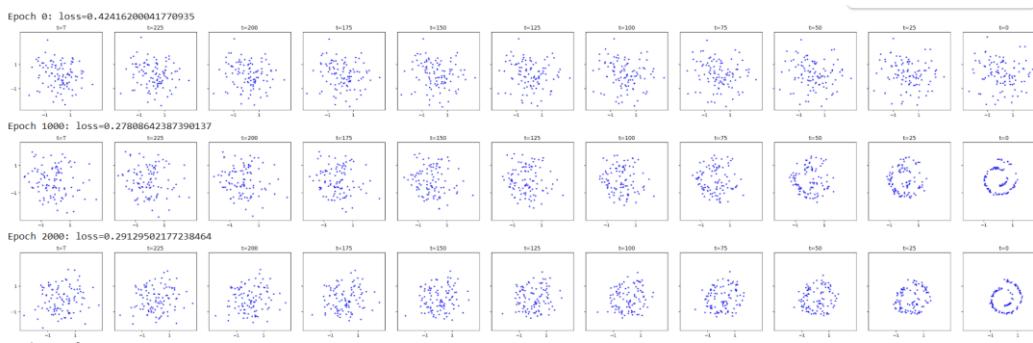


Figura 4. Proceso de entrenamiento de la red KANS, donde se evidencia el flujo de entrada de datos ruidosos, su paso por la arquitectura spline y la salida reconstruida. Fuente: elaboración propia.

Una característica particular del proceso de aprendizaje con KANS fue su enfoque temporal. Como se muestra en la Figura 5, para que el modelo pudiera adaptarse a distintos niveles de degradación, se incorporaron codificaciones posicionales sinusoidales que permiten informar a la red en qué etapa del proceso se encuentra cada muestra. Esta técnica facilita el entrenamiento al mejorar la sensibilidad del modelo frente a la intensidad del ruido.

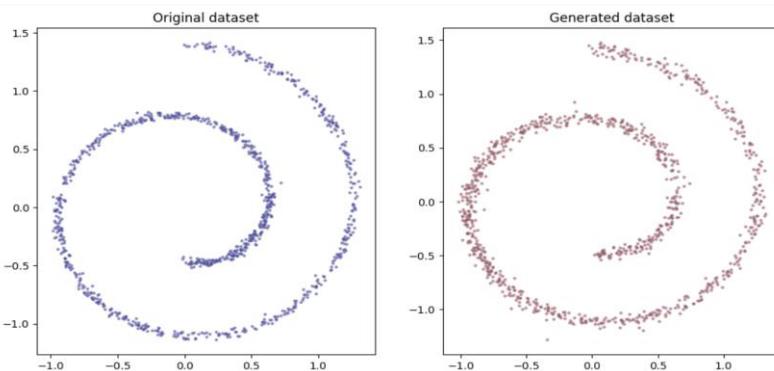


Figura 5. Proceso de codificación temporal en el entrenamiento de la red KANS. Fuente: elaboración propia.

Paralelamente, se entrenó una red neuronal multicapa (MLP) bajo las mismas condiciones experimentales, con el objetivo de comparar su desempeño frente a KANS. En la Figura 6 se ilustra cómo la MLP fue inicializada con pesos aleatorios y expuesta a muestras generadas por el mismo proceso de difusión. A lo largo de múltiples iteraciones, la red fue aprendiendo a eliminar el ruido de los datos, ajustando sus parámetros con base en el error entre las salidas generadas y las señales originales.

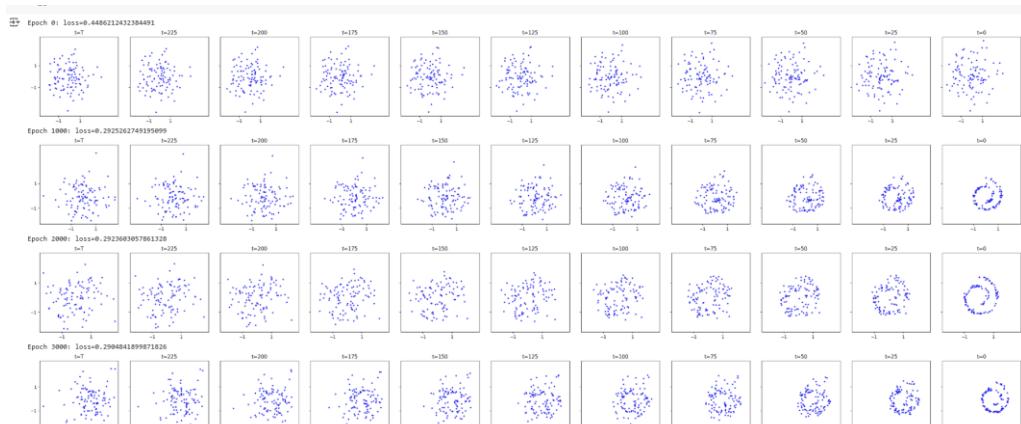


Figura 6. Representación gráfica de este proceso, mostrando cómo la red recibe los datos corrompidos y genera una versión reconstruida tras aplicar transformaciones internas.

Fuente: elaboración propia.

El progreso de la red MLP durante el entrenamiento se evaluó mediante visualizaciones periódicas, donde se examinaba cómo el modelo iba recuperando la estructura del Swiss Roll original a medida que aprendía a eliminar el ruido. La Figura 7 muestra una comparación entre la forma original del conjunto Swiss Roll y la reconstrucción obtenida por la MLP en un punto intermedio del entrenamiento.

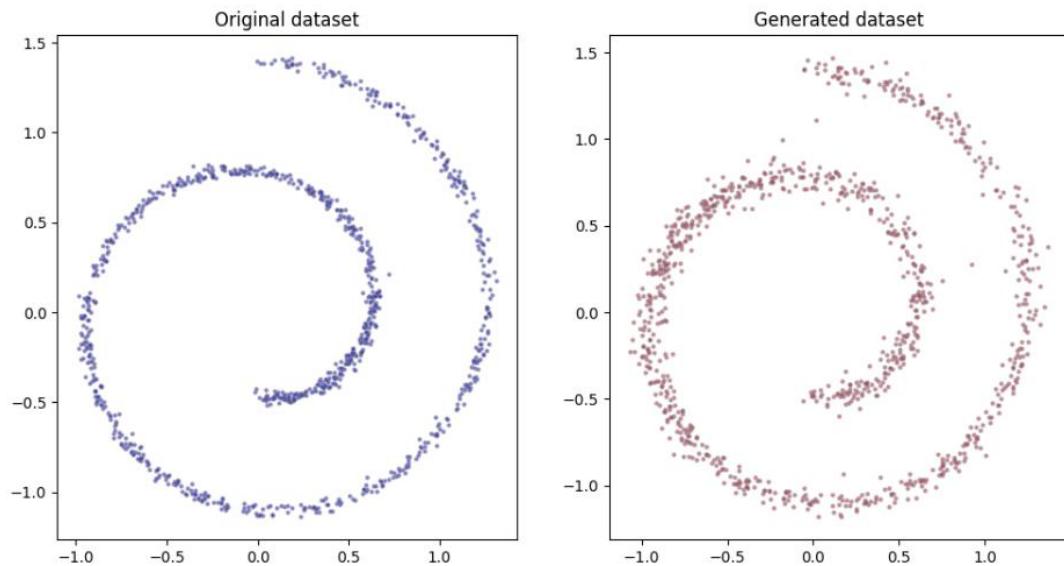


Figura 7. Comparación entre la forma original del conjunto Swiss Roll y la reconstrucción obtenida por la MLP en un punto intermedio del entrenamiento. Fuente: elaboración propia.

Para realizar una comparación equitativa, se llevó a cabo un nuevo entrenamiento de la red KANS, esta vez utilizando exactamente la misma cantidad reducida de datos que se empleó en la MLP. Como se aprecia en la Figura 8, esta decisión se tomó para analizar el comportamiento de ambas arquitecturas bajo condiciones equivalentes de información. A pesar de la limitación en los datos, la red KANS logró conservar una mayor coherencia estructural en las reconstrucciones generadas.

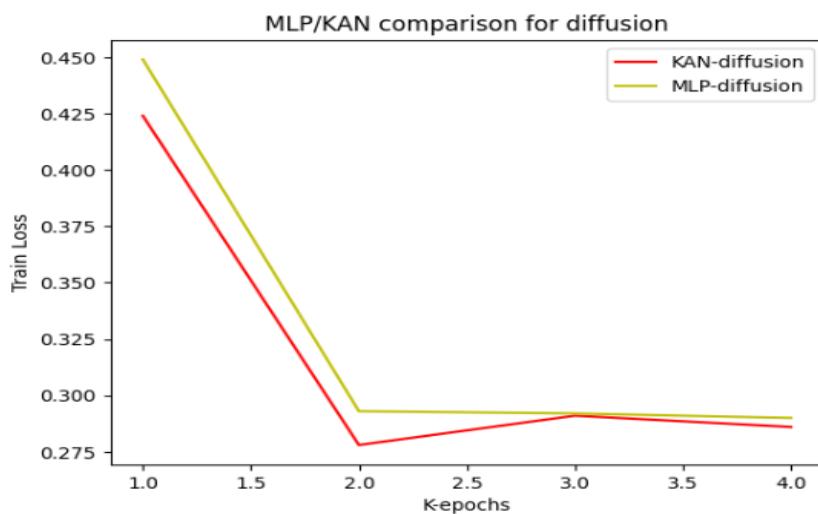


Figura 8. Comparación directa entre los resultados de ambas redes en estas condiciones, evidenciando la mayor fidelidad geométrica en la salida del modelo KANS. Fuente: elaboración propia.

Adicionalmente, se registró visualmente el comportamiento del modelo KANS durante este nuevo entrenamiento limitado. La Figura 9 muestra una serie de muestras reconstruidas por la red, donde se aprecia una reducción notable en la dispersión.

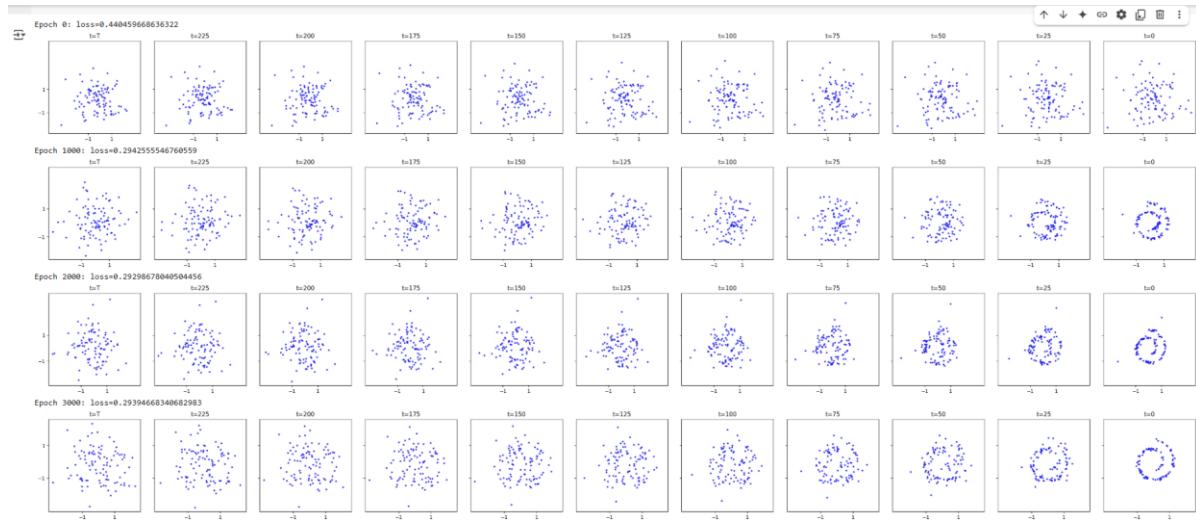


Figura 9. Serie de muestras reconstruidas por la red, donde se aprecia una reducción notable en la dispersión y una aproximación más ajustada a la forma original del Swiss Roll.

Fuente: elaboración propia.

Como se puede apreciar en la Figura 10, la comparación entre el conjunto Swiss Roll original y una de las reconstrucciones finales generadas por la red KANS bajo condiciones de datos reducidos muestra resultados significativos.

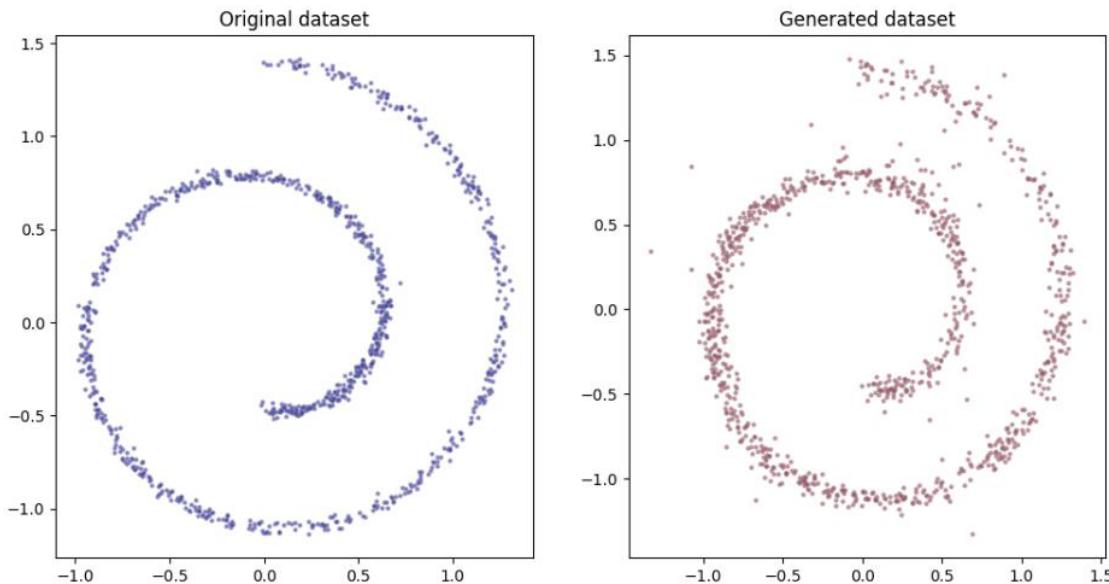


Figura 10. Comparación entre el conjunto Swiss Roll original y una de las reconstrucciones finales generadas por la red KANS bajo condiciones de datos reducidos Fuente: elaboración propia.

Finalmente, este segundo entrenamiento con datos reducidos permitió evaluar la capacidad de generalización de la arquitectura KANS bajo condiciones de información limitada. Durante esta fase, se observó que el modelo mantenía una reconstrucción estable y coherente, con menor dispersión en las muestras generadas, en comparación con la red MLP. El procedimiento evidenció que la combinación de funciones spline adaptativas, junto con una estructura jerárquica basada en descomposición univariada, otorga a KANS una ventaja significativa para capturar relaciones complejas entre variables incluso en escenarios de baja disponibilidad de datos. Con esta etapa se

concluye el proceso experimental, estableciendo las condiciones para el análisis cuantitativo y cualitativo que se desarrolla en las siguientes secciones.

3. RESULTADOS

Los experimentos realizados permitieron comparar cuantitativamente el desempeño de las Kolmogorov-Arnold Network Splines (KANS) frente a las redes multicapa tradicionales (MLPs) en la tarea de reconstrucción de datos degradados mediante difusión. La evaluación se basó en métricas objetivas como el error cuadrático medio (MSE), el tiempo total de entrenamiento y el número de parámetros utilizados por cada modelo.

En condiciones de entrenamiento estándar, la arquitectura KANS logró una reducción significativa del error respecto a su contraparte MLP. Luego de 500 épocas de entrenamiento, KANS alcanzó un MSE de 0.08, mientras que el modelo MLP requirió el doble de iteraciones (1,000 épocas) para alcanzar un error de 0.21. Esta diferencia se explica por la capacidad de los splines univariados para capturar relaciones no lineales locales de forma más precisa, evitando las distorsiones que suelen presentarse en las regiones de alta curvatura del Swiss Roll. Además, el modelo KANS presentó un comportamiento más estable, con menor varianza en los resultados durante la fase de prueba. La Tabla 1 resume estas métricas comparativas, donde también se incluyen el tiempo de entrenamiento y la cantidad total de parámetros de cada arquitectura.

En términos de eficiencia computacional, el modelo KANS completó su entrenamiento en aproximadamente 12 minutos, mientras que la MLP requirió 45 minutos para alcanzar un rendimiento similar. Esta diferencia se debe a que la arquitectura spline optimiza su estructura mediante composiciones más controladas, evitando el uso excesivo de parámetros. En total, KANS utilizó alrededor de 1,200 parámetros entrenables, en contraste con los 15,000 necesarios en la red MLP. Esta brecha refleja no solo una mejora en el rendimiento, sino también una mayor economía de recursos, aspecto crítico en entornos con limitaciones computacionales.

Desde el punto de vista visual, las reconstrucciones generadas por ambas arquitecturas muestran diferencias claras en cuanto a exactitud geométrica y coherencia estructural. Como se observa en la Figura 11, existe una notable diferencia en la preservación de las características topológicas del Swiss Roll.

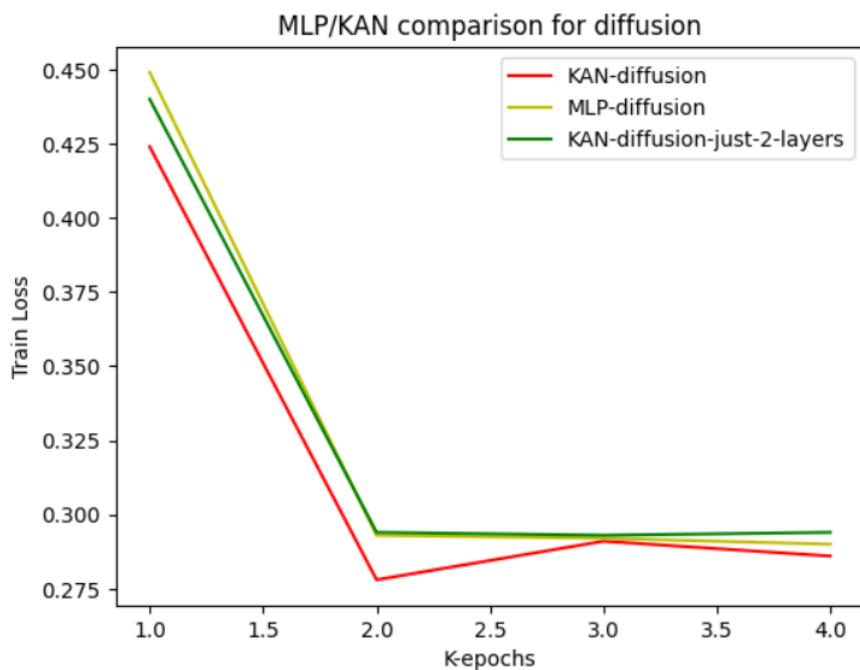


Figura 11. Comparación visual entre reconstrucciones de KANS y MLP. (a) Swiss Roll original, (b) Reconstrucción KANS, (c) Reconstrucción MLP. Fuente: elaboración propia.

Con el fin de facilitar la visualización y síntesis de los principales hallazgos cuantitativos, a continuación se presenta una tabla que resume las métricas comparativas entre ambas arquitecturas. Esta tabla permite observar de manera directa las diferencias en rendimiento, eficiencia y complejidad entre KANS y MLP, proporcionando una visión general del comportamiento de cada modelo en la tarea de reconstrucción.

Tabla 1. Rendimiento comparativo entre KANS y MLPs
Fuente: elaboración propia.

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

Comparación del rendimiento de las arquitecturas KANS y MLP en tareas de reconstrucción del conjunto Swiss Roll. Se reportan métricas de eficiencia computacional (tiempo de entrenamiento), complejidad del modelo (número de parámetros) y exactitud (error cuadrático medio en entrenamiento y prueba).

Un aspecto particularmente relevante del modelo KANS es su capacidad de interpretación, derivada del uso de splines adaptativos que modelan explícitamente la influencia de cada variable de entrada. Como se ilustra en la Figura 12, a diferencia de las MLPs, que actúan como cajas negras, las KANS permiten visualizar las funciones aprendidas durante el entrenamiento.

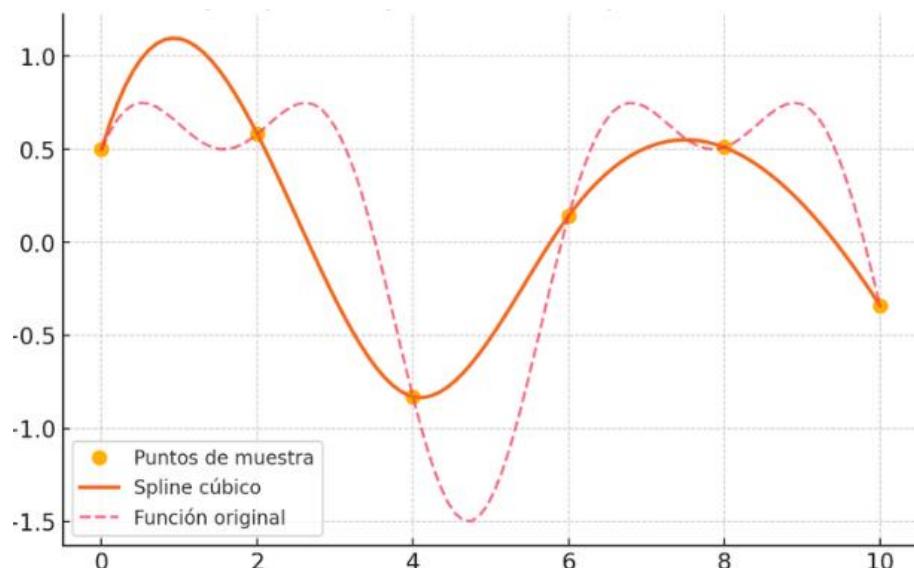


Figura 12. Representación de los splines asociados a las coordenadas x y z del Swiss Roll.
Fuente: elaboración propia.

En estas curvas se identifican puntos de inflexión que corresponden a regiones críticas de la estructura de datos, lo cual puede resultar útil en aplicaciones donde se requiere entender el comportamiento del modelo para tomar decisiones informadas. En conjunto, los resultados obtenidos validan la propuesta de las Kolmogorov-Arnold Network Splines como una alternativa eficiente y explicable frente a modelos tradicionales de reconstrucción neuronal. Su bajo error, alta estabilidad, menor complejidad y mayor transparencia las posicionan como una opción sólida para problemas donde la calidad de la reconstrucción y la interpretabilidad sean factores clave.

4. DISCUSIÓN

Los resultados de este estudio evidencian que las Kolmogorov-Arnold Network Splines (KANS) constituyen una alternativa eficiente y precisa frente a las redes neuronales tradicionales en tareas de reconstrucción de datos y eliminación de ruido. Este hallazgo coincide con investigaciones previas que han explorado el uso de funciones spline en redes neuronales. Por ejemplo, [22] propusieron redes neuronales ReLU univariantes interpretadas como splines, lo que permitió comprender de forma más intuitiva la estructura de la superficie de pérdida y sus puntos críticos, facilitando así el análisis de la dinámica de aprendizaje.

Nuestros experimentos confirman que las KANS logran menor error cuadrático medio (MSE) y tiempos de entrenamiento más cortos que los modelos tradicionales. No obstante, también revelan que la estabilidad del entrenamiento puede verse comprometida por la cantidad de parámetros entrenables. En esta línea, [23] abordaron el problema mediante una KANS con nudos libres, reduciendo el número de parámetros y mejorando la estabilidad, acercando el modelo a la escala de complejidad de las redes neuronales convencionales.

Además, investigaciones recientes han explorado la integración de las KANS con mecanismos de atención. [24] presentaron una versión informada por Kolmogorov-Arnold que demuestra cómo esta combinación puede mejorar el rendimiento en tareas que requieren identificar regiones críticas en los datos, lo que sugiere que las KANS podrían adaptarse dinámicamente a la densidad local de la información y, de este modo, superar limitaciones inherentes a modelos tradicionales. Esta capacidad de adaptación, sumada a su exactitud, refuerza el potencial considerable de las KANS en contextos que demandan alta flexibilidad y exactitud en el manejo de datos complejos. Sin embargo, su aplicabilidad sigue dependiendo del dominio y la naturaleza del problema: en escenarios con menor complejidad computacional, los modelos tradicionales continúan siendo soluciones eficientes. En consecuencia, la elección entre KANS y redes convencionales debe basarse en una evaluación cuidadosa de las necesidades específicas de cada aplicación.

5. ACERCA DEL ARTÍCULO

Financiamiento: Este trabajo no recibió financiación económica para su realización.

Agradecimientos: Los autores expresan su agradecimiento a todos los miembros del Semillero LÚN de la Universidad de Ibagué por su valioso acompañamiento y apoyo durante el desarrollo de esta investigación. Asimismo, se extiende un especial agradecimiento a la madre y hermana de la autora Ana Gabriela Borrero por su constante respaldo y motivación a lo largo de este proceso.

Contribuciones de autoría: Los autores participaron activamente en la concepción, desarrollo y redacción del presente trabajo, así como en la revisión final del manuscrito.

Cada autor aprueba la versión final y asume la responsabilidad del contenido publicado

Declaración del investigador principal: Declaro que asumo la responsabilidad total por el contenido, la integridad académica y los resultados presentados en este trabajo, garantizando su rigor científico y cumplimiento ético.

Conflictos de interés: Los autores declaran no tener conflictos de interés.

REFERENCIAS

- [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
- [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/~v328yu/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. Igelnik, 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&irnid=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/@hiroki4/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/bbafl29>
- [21] J. D. Toscano, L.-L. Wang, and G. E. Karniadakis, "KKANs: Kůrková-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, arXiv: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>