Kolmogorov-Arnold Network Autoencoders for High-Dimensional Data Representation

Anonymous authors
Paper under double-blind review

Abstract

In the era of high-dimensional data, traditional machine learning models often face challenges related to computational complexity, overfitting, and suboptimal feature representations. This paper introduces Kolmogorov-Arnold Network Autoencoders (KANAs), a novel framework that leverages Kolmogorov-Arnold Networks (KANs) to transform dimensionality reduction and data reconstruction. Through experiments across diverse datasets, KANA consistently demonstrates superior reconstruction fidelity and linear probing accuracy, establishing itself as a powerful and versatile tool for high-dimensional data processing. The proposed framework shows strong potential for applications in areas such as scientific modeling and data compression.

1 Introduction

In the age of big data, machine learning algorithms are increasingly tasked with processing and interpreting high-dimensional datasets across a variety of domains LeCun et al. (2015); He et al. (2016); Schmidhuber (2015). The ability to extract meaningful insights from such data has fueled advancements in fields as diverse as genomics, climate science, and financial analytics Hassan et al. (2022); Fathi et al. (2022); Sohangir et al. (2018). While high-dimensional data offers rich informational potential, it also introduces significant challenges, including computational complexity, large storage requirements, and the heightened risk of overfitting Bolón-Canedo et al. (2015); Tufail et al. (2023). These challenges are often exacerbated by the "curse of dimensionality," a phenomenon wherein the effective volume of the feature space grows exponentially with the dimensionality, leading to sparsity and diminishing model performance Verleysen & François (2005).

To mitigate these issues, the machine learning field has seen the development of advanced techniques aimed at reducing dimensionality and learning efficient representations. Among these, autoencoders have emerged as a fundamental tool in unsupervised learning, capable of capturing the underlying structure of data in a compressed form Hinton & Salakhutdinov (2006); Bengio et al. (2013). However, traditional autoencoder architectures often rely on static activation functions such as ReLU or sigmoid, which may limit their adaptability to the complex geometries and dynamics inherent in high-dimensional data.

Recent advancements have sought to address these limitations by innovating network architectures and activation functions. Notably, the success of Kolmogorov-Arnold Networks (KANs) Liu et al. (2025), inspired by the Kolmogorov-Arnold representation theorem, has highlighted the potential of neural networks to approximate high-dimensional mappings with enhanced expressiveness. Building on these innovations, learnable activation functions such as Swish Ramachandran et al. (2017) and Mish Misra (2019) have further expanded the capacity of neural networks to adapt to varying data distributions.

Motivated by these advancements, we introduce Kolmogorov-Arnold Network Autoencoders (KANAs), a framework designed specifically to tackle the complexities of high-dimensional data representation. Unlike conventional models, KANA integrates KANs within the autoencoder architecture, enhancing its ability to capture intricate patterns and adapt to diverse data structures. This dynamic approach enables KANA to achieve superior reconstruction quality and generalization, making it a robust solution for a wide range of applications, from data compression to scientific modeling.

2 Related Work

The study of KANs has gained significant attention in recent years as a promising alternative to traditional multilayer perceptrons (MLPs) Samadi et al. (2024). Unlike conventional MLPs, which employ static linear weights followed by fixed activation functions, KANs replace linear weights with learnable activation functions, allowing for dynamic pattern learning and enhanced adaptability. This architectural innovation has enabled KANs to achieve superior performance with fewer parameters, outperforming larger MLPs in terms of accuracy, scaling efficiency, and interpretability Vaca-Rubio et al. (2024). These characteristics make KANs particularly attractive for applications requiring compact yet expressive models.

KANs have demonstrated significant success across various domains. In graph learning, for example, specialized KAN-based architectures such as the Kolmogorov-Arnold Graph Isomorphism Network (KAGIN) and the Kolmogorov-Arnold Graph Convolution Network (KAGCN) have achieved state-of-the-art results in graph regression tasks by providing more effective node feature updates Bresson et al. (2024). These models leverage the flexibility of learnable activation functions to capture intricate relationships within graph structures, surpassing the capabilities of traditional MLPs. Similarly, KANs have been shown to improve transfer learning frameworks by replacing conventional linear probing layers in ResNet-50 architectures with KAN layers Shen & Younes (2024), significantly enhancing adaptability to complex and diverse data patterns while improving generalization performance.

KAN principles have also been successfully integrated into Convolutional Neural Networks (CNNs). For instance, the Residual KAN (RKAN) incorporates KAN modules into established architectures like ResNet and DenseNet. By using Chebyshev polynomial-based convolutions, RKAN achieves improved feature extraction capabilities while maintaining computational efficiency Yu et al. (2024).

Beyond KANs, the concept of learnable activation functions has emerged as a broader research theme in neural network design. Methods such as Parametric Rectified Linear Unit (PReLU) He et al. (2015) and spline-based activation functions Bohra et al. (2020) share the goal of enhancing network adaptability by allowing activation functions to evolve during training.

Our proposed KANA build upon the theoretical foundations of learnable activation functions. By integrating learnable activation functions into the autoencoder architecture, KANA offers a more adaptable solution for tasks such as dimensionality reduction and data reconstruction.

3 Method

3.1 Architecture

The architecture of KANA is designed to effectively process high-dimensional data by transforming it through a structured sequence of encoding and decoding steps. At its core, KANA consists of two primary components: an encoder and a decoder, which work together to achieve dimensionality reduction and data reconstruction, respectively. The overall structure of KANA is illustrated in Figure 1.

The encoder maps high-dimensional input data $\mathbf{x} \in \mathbb{R}^D$ into a compressed, lower-dimensional latent space $\mathbf{z} \in \mathbb{R}^d$, where $d \ll D$. This is achieved through a composition of functions, each corresponding to a layer in the encoder:

$$\mathbf{z} = (\Phi_{L-1} \circ \Phi_{L-2} \circ \cdots \circ \Phi_0)(\mathbf{x}), \tag{1}$$

where each Φ_i represents a transformation applied at layer i, typically combining a linear transformation, a learnable activation function, and optional normalization techniques. This layered approach progressively encodes the input $\mathbf{x}$ into increasingly abstract representations, culminating in the latent variable $\mathbf{z}$, which captures the most salient features necessary for reconstruction.

The decoder, conversely, reconstructs the original high-dimensional input $\mathbf{x}$ from its latent representation $\mathbf{z}$. This is achieved through another sequence of function compositions, reversing the encoding process:

$$\hat{\mathbf{x}} = (\Psi_{L-1} \circ \Psi_{L-2} \circ \cdots \circ \Psi_0)(\mathbf{z}), \tag{2}$$

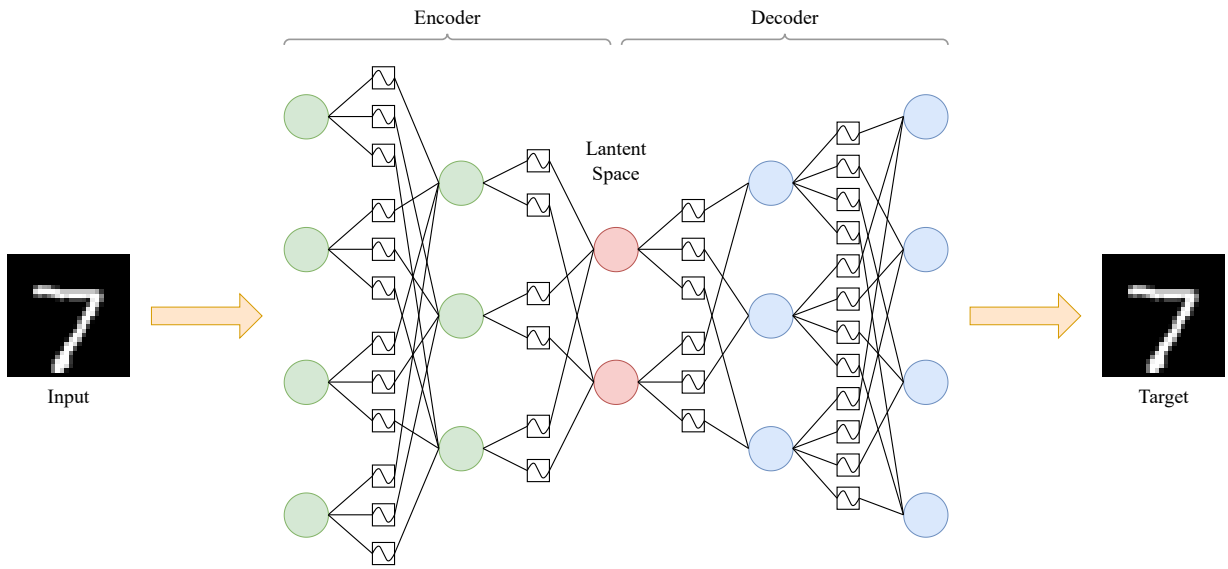

Figure 1: Overview of the KANA architecture.

where each Ψ_i represents a transformation applied at layer i of the decoder. These transformations progressively map the latent representation $\mathbf{z}$ back into the original data space. The output $\hat{\mathbf{x}}$ is the reconstruction of the input data $\mathbf{x}$, with the aim of minimizing the discrepancy between $\hat{\mathbf{x}}$ and $\mathbf{x}$.

This layered architecture, coupled with learnable activation functions within the transformations Φ_i and Ψ_i, enables KANA to adapt dynamically to the underlying structure of the data.

3.2 KAN Layer

The Kolmogorov-Arnold representation theorem establishes that any continuous function mapping from $[0,1]^n$ to $\mathbb{R}$ can be decomposed into a sum of univariate continuous functions combined through addition operations Schmidt-Hieber (2021). This theorem serves as a cornerstone in high-dimensional function approximation and is formally expressed as:

$$f(\mathbf{x}) = f(x_1, \ldots, x_n) = \sum_{q=1}^{2n+1} \psi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right), \tag{3}$$

where $\psi_{q,p} : [0,1] \to \mathbb{R}$ and $\Phi_q : \mathbb{R} \to \mathbb{R}$ are continuous univariate functions.

Inspired by this theorem, the KAN layer in KANA is designed to adapt dynamically during training by employing a collection of trainable activation functions, denoted collectively as Φ. These functions allow the KAN layer to adjust its behavior to the specific patterns and structures present in the input data. However, instead of adhering to the $2n+1$ dimensional framework suggested by the theorem, KANA adopts a more flexible approach. By optimizing the latent dimensionality and learnable activation parameters empirically, we aim to achieve an optimal trade-off between representational capacity and computational efficiency. This deviation is motivated by the observation that the rigid $2n+1$ structure may over-parameterize the activation layer, leading to diminishing returns in performance.

The activation output for an input vector $\mathbf{x}$ is represented as:

$$\Phi(\mathbf{x}) = \sum_{i=1}^{N} \phi_i(x_i), \tag{4}$$

where $\phi_i(x_i)$ represents the learnable activation function corresponding to the i-th input dimension. By tailoring each $\phi_i(x_i)$ to optimize specific attributes of the data, the KAN layer is able to enhance its effectiveness in modeling complex, non-linear relationships.

Each activation function $\phi(x)$ within the KAN layer integrates a spline-based representation with a base activation function to achieve a high degree of flexibility and expressiveness. This is mathematically defined as:

$$\phi(x) = \sum_{i=1}^{l+p} B_{i,p}(x) \cdot w_i + b(x) \cdot w_0, \tag{5}$$

where $B_{i,p}(x)$ are B-spline basis functions of degree p, which provide a localized and adaptable basis for approximating functions. The grid size, l, determines the number of basis functions, and w_i are the trainable coefficients that adjust the contribution of each basis function. The term $b(x)$ represents the base activation function, which is scaled by the trainable coefficient w_0.

3.3 Convolutional KAN Layer

Inspired by the architecture of CNNs, the Convolutional KAN (CKAN) layer introduces shift-invariant properties into neural networks, enabling the model to recognize patterns regardless of their spatial position. This design addresses a key limitation of the standard KAN layer, which assigns a separate set of parameters to each input dimension. By incorporating parameter sharing across different input locations, the KAN layer significantly reduces the number of trainable parameters, enhancing computational efficiency while retaining expressiveness.

The core of the CKAN layer is a kernel composed of learnable activation functions organized in a matrix format:

$$\Phi = \begin{bmatrix} \phi_{1,1} & \phi_{1,2} & \phi_{1,3} & \cdots & \phi_{1,m} \\ \phi_{2,1} & \phi_{2,2} & \phi_{2,3} & \cdots & \phi_{2,m} \\ \phi_{3,1} & \phi_{3,2} & \phi_{3,3} & \cdots & \phi_{3,m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_{n,1} & \phi_{n,2} & \phi_{n,3} & \cdots & \phi_{n,m} \end{bmatrix}, \tag{6}$$

where each element $\phi_{i,j}$ represents a distinct learnable activation function.

The CKAN layer applies the kernel Φ in a convolution-like manner over the input, enabling the activation functions to account for spatial relationships while maintaining shift invariance. This approach mirrors the way CNN kernels learn spatially invariant features but extends the concept to learnable, non-linear activation functions.

4 Experiments

4.1 Experimental Setup

Self-supervised pretraining. During the self-supervised pretraining stage, the model is trained over 50 epochs to learn meaningful and robust representations in the latent space. The AdamW optimizer Loshchilov (2017) is utilized with a learning rate of 1×10^{-3}, a weight decay of 1×10^{-4}, and a cosine annealing learning rate schedule Loshchilov & Hutter (2016). Training is conducted with a batch size of 1024, optimizing the reconstruction quality. Unless otherwise specified, the latent space dimension is set to 16.

Linear probing. The linear probing phase evaluates the quality of the representations learned during pretraining. Here, the pretrained encoder is frozen, and a lightweight linear classifier is trained on the extracted latent features. The training is conducted over 5 epochs using the SGD optimizer Robbins &

Model	Dimension	MNIST		Fashion-MNIST	
		MSE	Accuracy	MSE	Accuracy
Baseline	8	0.0223	87.2%	0.0169	74.4%
	16	0.0172	88.2%	0.0144	78.8%
	32	0.0132	89.4%	0.0120	80.1%
KANA	8	0.0174	88.9%	0.0139	76.4%
	16	0.0108	91.5%	0.0112	79.0%
	32	0.0075	91.0%	0.0096	80.6%

Table 1: Model performance across different latent dimensions on MNIST and Fashion-MNIST datasets.

Monro (1951) with a learning rate of 0.1, momentum Qian (1999) set to 0.9 , and a batch size of 256. This stage provides a quantitative assessment of the discriminative power of the learned features.

4.2 Model Robustness

We evaluated KANA's robustness across different latent space dimensions using the MNIST Deng (2012) and Fashion-MNIST Xiao et al. (2017) datasets. For comparison, our baseline model uses a standard MLP architecture for both encoder and decoder, matching KANA in depth and hidden units. The results, summarized in Table 1, show that KANA consistently outperformed the baseline model. Notably, increasing the latent dimension to 32 for MNIST did not yield further improvements in linear probing classification accuracy, which slightly declined to 91.0%. This indicates that additional latent dimensions may introduce redundancy rather than contributing meaningful new features for this dataset. In contrast, the baseline model, even with a latent dimension of 32, struggled to match KANA's performance at just 16 dimensions, achieving a accuracy of only 89.4% during linear probing. This highlights KANA's ability to learn compact and expressive representations more effectively than traditional methods. The findings underscore KANA's efficiency in balancing latent dimensionality and performance, offering robust results across diverse configurations and datasets.

4.3 Qualitative Analysis

Reconstructed images. Figure 2 compares original and reconstructed images. KANA demonstrates a remarkable ability to reconstruct input data, preserving high-fidelity details even for complex patterns.

Latent space interpolation. To evaluate the quality of the learned latent space, we performed linear interpolations between selected data points in this space. Given two latent vectors, $\mathbf{z}_0 = f(\mathbf{x}_0)$ and $\mathbf{z}_1 = f(\mathbf{x}_1)$, where $f(\cdot)$ is the encoder function, we compute a linear interpolation, $\mathbf{z}_\lambda = (1-\lambda)\mathbf{z}_0 + \lambda\mathbf{z}_1$, where $\lambda \in [0, 1]$. Decoding these interpolated latent vectors with the decoder function $g(\cdot)$ produces smooth transitions between the original data points, $\hat{\mathbf{x}}_\lambda = g(\mathbf{z}_\lambda)$. The results, with λ chosen at equal intervals, shown in Figure 3, demonstrate the model's ability to learn continuous, meaningful, and semantically coherent representations within its latent space.

4.4 Ablation Studies

To understand the impact of key architectural and design choices, we performed a series of ablation studies on the MNIST dataset. These studies focused on encoder and decoder design choices, hidden layer dimensions, and loss functions, with evaluations based on reconstruction quality and linear probing classification accuracy.

Encoder Design. The encoder designs evaluated in our experiments span four distinct configurations, each progressively enhancing representational capabilities. The baseline MLP consists of two fully connected layers with a ReLU activation in between. The convolution-enhanced MLP incorporates a convolutional layer at the start, followed by a ReLU activation and a fully connected linear layer. The standard KAN encoder employs two stacked KAN layers, which dynamically adapt their activation functions to improve feature

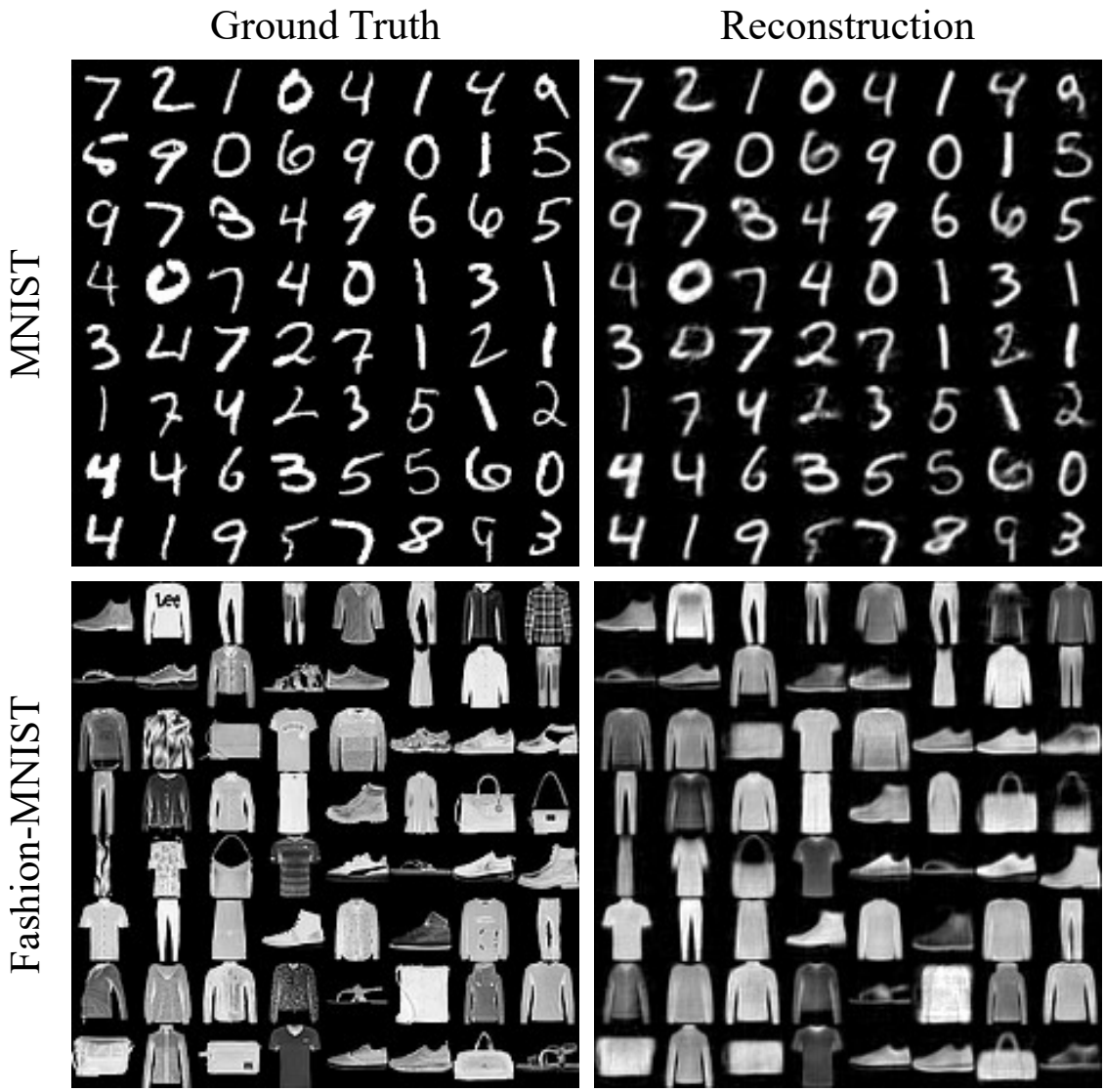

Figure 2: Visual comparison of original and reconstructed images on MNIST and Fashion-MNIST datasets.

Type	MSE	Accuracy
MLP	0.0120	89.1%
Conv	0.0138	87.1%
KAN	**0.0108**	**91.5%**
CKAN	0.0130	87.2%

Table 2: Ablation study on encoder design. Default settings are highlighted in gray.

extraction. Finally, the CKAN encoder combines a CKAN layer with a KAN layer, which learns spatially invariant features and reduce the parameters by weight sharing. The results, summarized in Table 2, the significance of learnable activation functions. Interestingly, while CKAN layers enhance shift invariance, their performance on MNIST suggests a trade-off between parameter sharing and expressiveness.

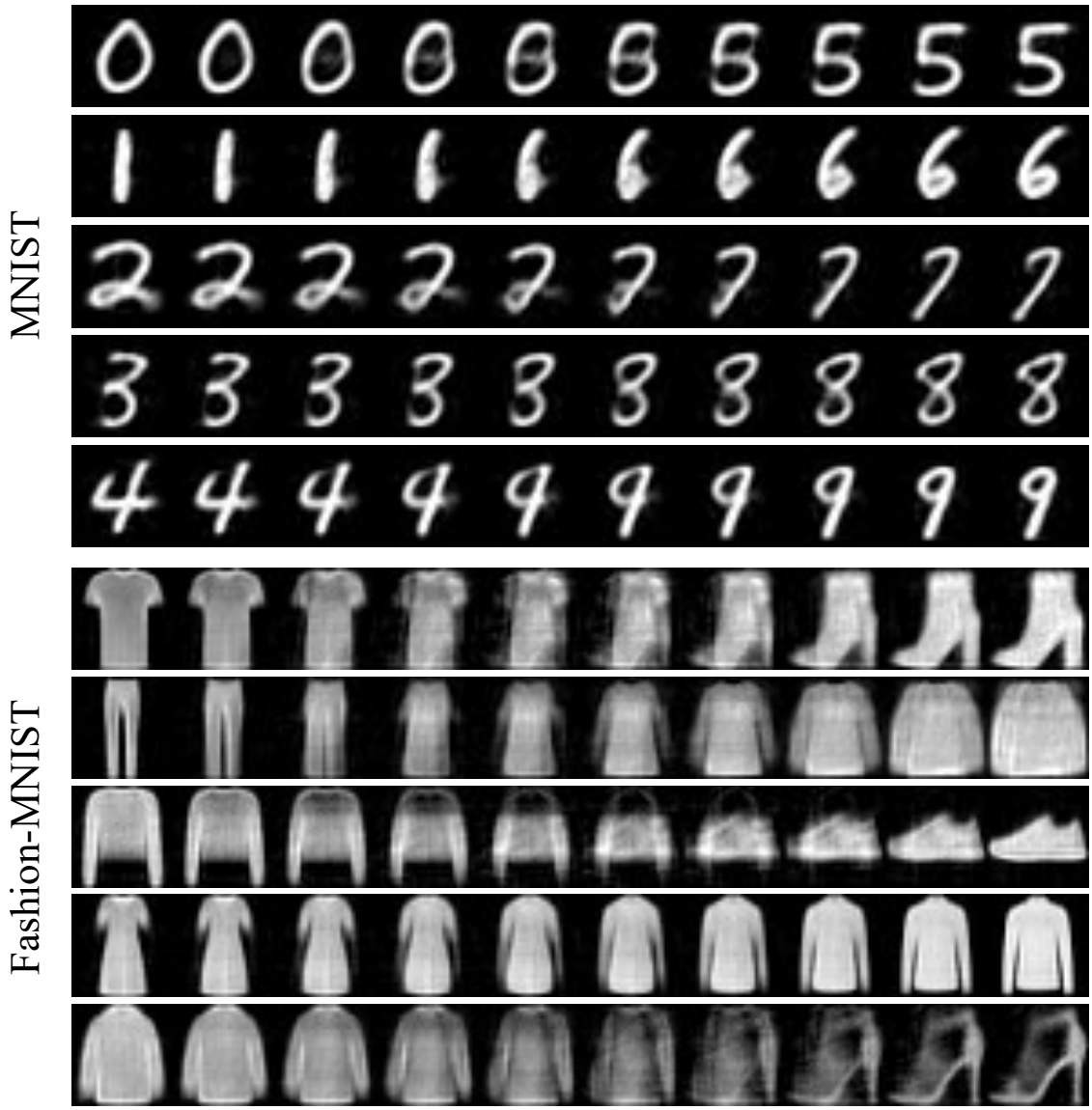

Figure 3: Visualization of linear interpolation between latent representations on MNIST and Fashion-MNIST datasets.

Type	MSE	Accuracy
MLP	0.0160	88.4%
KAN	**0.0108**	**91.5%**

Table 3: Ablation study on decoder design. Default settings are highlighted in gray.

Decoder design. Similarly, we compared MLP-based decoders with those incorporating KAN layers. As shown in Table 3, decoders with KAN layers significantly outperformed their MLP counterparts, achieving lower reconstruction error and higher accuracy. These results indicate that learnable activation mechanisms are not only beneficial in encoding but also enhance decoding processes, likely by better capturing complex data distributions.

Encoder	Decoder	MSE	Accuracy
196	33	0.0167	90.1%
196	196	0.0108	**91.5%**
1569	33	0.0156	89.1%
1569	196	**0.0107**	91.2%

Table 4: Ablation study on hidden layer dimensions. Default settings are highlighted in gray.

Type	MSE	Accuracy
Log-Cosh	**0.0107**	91.0%
MAE	0.0144	91.1%
MSE	0.0108	**91.5%**
MSLE	0.0113	91.1%

Table 5: Ablation study on loss functions. Default settings are highlighted in gray.

Hidden layer dimensions. We explored the relationship between hidden layer dimensionality and model performance by testing various configurations, as shown in Table 4. Contrary to theoretical expectations based on Kolmogorov–Arnold representation ($2n + 1$ dimensions for input dimension n), the best results were achieved with balanced 196-dimensional hidden layers. This configuration struck an optimal balance between parameter efficiency and representational capacity, resulting in low reconstruction error and high classification accuracy during linear probing.

Loss funtions. The choice of loss function plays a crucial role in shaping the model's learning behavior and convergence. We evaluated several loss functions, including Log-Cosh, Mean Absolute Error (MAE), Mean Squared Error (MSE), and Mean Squared Logarithmic Error (MSLE). As shown in Table 5, the MSE loss demonstrated the best balance between reconstruction error and linear probing classification accuracy, confirming its effectiveness for this task.

5 Discussion

Our findings demonstrate that the KANA framework, through its innovative integration of KAN layers, effectively tackles the complex challenges associated with processing high-dimensional data. By enabling dynamic and adaptive activation functions, KANA not only enhances reconstruction quality but also generates latent representations that are both meaningful and semantically coherent. These advancements are substantiated by a comprehensive set of qualitative and quantitative analyses, which highlight the model's ability to capture intricate patterns and dependencies in the data.

However, one limitation of the KANA framework is the computational overhead introduced by its use of spline-based learnable activation functions, particularly when compared to traditional method. The overhead primarily arises from the increased complexity in parameter optimization, as well as the additional computational resources required during both training and inference. While this computational burden may limit KANA's scalability, particularly in applications that involve extremely high-dimensional data or demand real-time processing capabilities, the significant improvements in reconstruction fidelity and latent representation quality make it a highly valuable tool in contexts where precision and accuracy are of utmost importance.

Looking to the future, research should focus on expanding KANA's capabilities by applying it to more complex, multi-layered network architectures. Such expansions would provide a more thorough understanding of the model's performance improvements, as well as the trade-offs that come with increased model complexity, including potential risks of overfitting and increased computational requirements. Exploring efficient approximations of spline-based activations or novel learnable activation designs could help strike a balance

between maintaining model expressiveness and reducing computational demands. Furthermore, it will be critical to assess KANA's scalability and computational efficiency in large-scale deployments, especially in applications involving massive datasets or stringent real-time processing requirements. Lastly, future studies should evaluate KANA's adaptability to a broader range of tasks and datasets, including multi-modal, graph-structured, and domain-specific data, to better understand its robustness, generalization capabilities, and potential across diverse use cases and industries.

6 Conclusion

In conclusion, KANA marks a transformative step forward in the field of high-dimensional data processing. By introducing KAN layers, KANA enhances both the quality of reconstruction and the richness of latent representations. These improvements make KANA a powerful tool in a wide array of machine learning applications, particularly those requiring high accuracy and nuanced understanding of complex data. The ability to dynamically adapt activation functions allows KANA to model intricate dependencies in the data, which can lead to more robust and flexible representations compared to traditional static activations.

Although challenges related to computational efficiency and scalability remain, especially in resource-constrained environments, the results thus far strongly suggest that KANA offers substantial promise for a variety of real-world applications. Its potential to improve performance in fields such as computer vision, natural language processing, and time-series analysis underscores the broad applicability of the framework. As further refinements are made to reduce its computational overhead and optimize its deployment, KANA's impact could extend to even more complex and large-scale tasks, where the need for both accuracy and computational efficiency is critical.

Ultimately, KANA represents not only an important advance in activation function design but also a foundation upon which future innovations in deep learning architectures can be built. The ongoing development of KANA will likely inspire new methods for improving model expressiveness, efficiency, and generalization, making it a valuable asset in pushing the boundaries of machine learning research and application.

References

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Pakshal Bohra, Joaquim Campos, Harshit Gupta, Shayan Aziznejad, and Michael Unser. Learning activation functions in deep (spline) neural networks. *IEEE Open Journal of Signal Processing*, 1:295–309, 2020.

Verónica Bolón-Canedo, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. Recent advances and emerging challenges of feature selection in the context of big data. *Knowledge-based systems*, 86:33–45, 2015.

Roman Bresson, Giannis Nikolentzos, George Panagopoulos, Michail Chatzianastasis, Jun Pang, and Michalis Vazirgiannis. Kagnns: Kolmogorov-arnold networks meet graph learning. *arXiv preprint arXiv:2406.18380*, 2024.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Marzieh Fathi, Mostafa Haghi Kashani, Seyed Mahdi Jameii, and Ebrahim Mahdipour. Big data analytics in weather forecasting: A systematic review. *Archives of Computational Methods in Engineering*, 29(2): 1247–1275, 2022.

Mubashir Hassan, Faryal Mehwish Awan, Anam Naz, Enrique J deAndrés Galiana, Oscar Alvarez, Ana Cernea, Lucas Fernández-Brillet, Juan Luis Fernández-Martínez, and Andrzej Kloczkowski. Innovations in genomics and big data analytics for personalized medicine and health care: A review. *International journal of molecular Sciences*, 23(9):4645, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacic, Thomas Y. Hou, and Max Tegmark. KAN: Kolmogorov–arnold networks. In *The Thirteenth International Conference on Learning Representations*, 2025.

I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Diganta Misra. Mish: A self regularized non-monotonic activation function. *arXiv preprint arXiv:1908.08681*, 2019.

Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999.

Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.

Moein E Samadi, Younes Müller, and Andreas Schuppert. Smooth kolmogorov arnold networks enabling structural knowledge representation. *arXiv preprint arXiv:2405.11318*, 2024.

Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

Johannes Schmidt-Hieber. The kolmogorov–arnold representation theorem revisited. *Neural networks*, 137: 119–126, 2021.

Sheng Shen and Rabih Younes. Reimagining linear probing: Kolmogorov-arnold networks in transfer learning. *arXiv preprint arXiv:2409.07763*, 2024.

Sahar Sohangir, Dingding Wang, Anna Pomeranets, and Taghi M Khoshgoftaar. Big data: Deep learning for financial sentiment analysis. *Journal of Big Data*, 5(1):1–25, 2018.

Shahid Tufail, Hugo Riggs, Mohd Tariq, and Arif I Sarwat. Advancements and challenges in machine learning: A comprehensive review of models, libraries, applications, and algorithms. *Electronics*, 12(8): 1789, 2023.

Cristian J Vaca-Rubio, Luis Blanco, Roberto Pereira, and Màrius Caus. Kolmogorov-arnold networks (kans) for time series analysis. *arXiv preprint arXiv:2405.08790*, 2024.

Michel Verleysen and Damien François. The curse of dimensionality in data mining and time series prediction. In *International work-conference on artificial neural networks*, pp. 758–770. Springer, 2005.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Ray Congrui Yu, Sherry Wu, and Jiang Gui. Residual kolmogorov-arnold network for enhanced deep learning. *arXiv preprint arXiv:2410.05500*, 2024.