Adversarial Experiments on Convolutional Kolmogorov Arnold Networks

In partial fulfillment of the project for CS F425 - Deep Learning

A Vinil
BITS Pilani - Hyderabad Campus

Abhijeet Manoj Varma
BITS Pilani - Hyderabad Campus

Kakade Rohan Bhaskar
BITS Pilani - Hyderabad Campus

ABSTRACT

We conduct adversarial experiments on Convolutional Kolmogorov-Arnold Networks (KAN-CNN) to evaluate their robustness against adversarial perturbations. Adversarial images are generated using the Fast Gradient Sign Method (FGSM), and the models' performance is assessed across varying spline orders and grid sizes, key parameters for KAN's function approximation. Our findings reveal that KAN-CNNs, despite their novel structure, are vulnerable to adversarial attacks and exhibit limited improvements in performance when these parameters are adjusted. Furthermore, while adversarial training improves robustness, the results underscore that hybrid KAN-CNN models remain susceptible to adversarial manipulations, akin to traditional MLP-based CNNs, necessitating further exploration of defense mechanisms.

KEYWORDS

Adversarial Learning, Kolmogorov Arnold Networks, Convolutional Neural Networks

1 INTRODUCTION

Deep learning models, particularly Convolutional Neural Networks (CNNs), have demonstrated remarkable performance in various domains, ranging from image recognition to natural language processing [7, 8]. However, their susceptibility to adversarial attacks has raised significant concerns, particularly in critical applications such as healthcare, finance, and autonomous systems [6, 9]. Adversarial learning, which explores these vulnerabilities and defense mechanisms, has become a crucial research area in modern AI.

Kolmogorov-Arnold Networks (KANs), inspired by the Kolmogorov-Arnold representation theorem, have recently emerged as a promising alternative to traditional Multi-Layer Perceptrons (MLPs)[5]. By employing learnable activation functions and univariate function approximations, KANs offer enhanced interpretability and accuracy, especially in scenarios requiring compact architectures [4]. The integration of KANs with CNNs (KAN-CNNs) extends these advantages to convolutional models. This approach has shown potential for improving adaptability and function approximation in CNNs, while maintaining a comparable level of accuracy but with fewer parameters, as demonstrated in recent empirical studies [2].

Despite their theoretical appeal, the robustness of KAN-CNNs under adversarial conditions remains largely unexplored. This work aims to fill this gap by evaluating the performance of KAN-CNNs in adversarial scenarios using images generated through the Fast Gradient Sign Method (FGSM) [6]. We systematically vary two key KAN parameters—*Spline Order* (*SO*) and *Grid Size* (*GS*)—to assess their impact on model robustness and computational cost. Additionally, we investigate the effectiveness of adversarial training in mitigating these vulnerabilities.

The paper is organized as follows: Section 2 provides background information on Adversarial Learning and Kolmogorov-Arnold Networks. Section 3 describes the experiments. A discussion of the results is presented in Section 4, followed by conclusions and directions for future research in Section 5.

Project in Computer Science (DAT620), IDE, UiS
2018.

2 RELATED WORK

This section discusses background information on Adversarial Learning and its intersection with Kolmogorov-Arnold Networks (KANs).

2.1 Adversarial Learning

Adversarial learning is a critical area in machine learning, examining how models can be manipulated by adversarial examples—inputs intentionally modified to mislead the model into incorrect predictions. This vulnerability has significant implications for security-sensitive applications such as autonomous driving and cybersecurity. Understanding adversarial learning is essential for building robust models that can withstand such attacks.

A common method for generating adversarial examples is the Fast Gradient Sign Method (FGSM), which perturbs the input based on the gradient of the loss function with respect to the input:

$$x_{\text{adv}} = x + \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y)),$$

where ϵ is the step size that controls perturbation intensity. FGSM is computationally efficient and highlights vulnerabilities in neural network models due to their linear decision boundaries.

Defense mechanisms are also an integral part of adversarial learning. Adversarial training, for instance, involves training models with perturbed data to make them more resilient to adversarial examples. The adversarial risk function for such training is defined as:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim D} \left[\max_{\|\delta\| \leq \epsilon} J(\theta, x + \delta, y) \right],$$

where the model is optimized to minimize the expected loss after perturbation. This method encourages robust decision boundaries that reduce sensitivity to perturbations.

2.2 Kolmogorov-Arnold Networks

Kolmogorov-Arnold Networks (KANs) offer an alternative to traditional Multi-Layer Perceptrons (MLPs) by leveraging the Kolmogorov-Arnold representation theorem, which states that any multivariate continuous function can be represented as a finite sum of univariate functions. KANs replace the fixed activation functions of MLPs with

learnable activation functions on edges, enhancing interpretability and computational efficiency [5].

The architecture of KANs utilizes spline-based functions instead of linear weight matrices, represented as:

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

where Φ_q and $\varphi_{q,p}$ are learnable B-splines. This structure simplifies the computation graph and allows KANs to achieve comparable or better performance with fewer parameters, which is beneficial for small-scale tasks.

KANs mitigate the curse of dimensionality (COD) by breaking down high-dimensional functions into simpler, univariate spline components. This approach avoids the exponential growth in parameters typically needed in traditional networks. For example:

$$f(x) = \sum_{i=1}^{L} \sum_{j=1}^{N} \phi_{i,j}(x_j),$$

where $\phi_{i,j}$ are adaptive spline functions.

The scalability of KANs is supported by their improved convergence properties. For KANs using cubic splines ($k = 3$), the scaling exponent α can be as high as 4, indicating faster convergence than MLPs:

$$\|f - (\Phi_{L-1} \circ \Phi_{L-2} \circ \cdots \circ \Phi_0)x\|_{C^m} \leq CG^{-k-1+m},$$

where G is the grid size and k is the spline order.

KANs' dual degrees of freedom, involving the arrangement of nodes (external) and spline flexibility (internal), enable them to capture complex data structures more effectively than MLPs. This makes KANs well-suited for tasks that require interpretability and parameter efficiency.

In summary, KANs provide a scalable and efficient alternative to MLPs by using spline-based function representations, making them ideal for applications that require high accuracy, interpretability, and reduced computational resources.

2.3 KAN and Adversarial Learning

Kolmogorov-Arnold Networks (KANs) have emerged as a promising approach for function approximation, leveraging their unique structure to provide adaptive, spline-based activation functions on edges rather than fixed node activations found in traditional MLPs. This architecture has shown potential in enhancing neural network robustness under adversarial conditions. Despite this promise, the robustness of KANs, particularly in the context of adversarial learning, remains an under-explored area. Empirical studies have demonstrated that both smaller and larger KAN models can exhibit vulnerabilities to adversarial attacks[1].

3 EXPERIMENTS ON CNN-KAN MODELS

In this section, we perform two experiments on a CNN-KAN model developed by [3].

Each experiment involves training three variations of the model: $KAN\text{-}CNN_{\text{original}}$, $KAN\text{-}CNN_{\text{manipulated}}$, and $KAN\text{-}CNN_{\text{secure}}$. The $KAN\text{-}CNN_{\text{original}}$ model is trained and tested on original images, the $KAN\text{-}CNN_{\text{manipulated}}$ model is trained on original images and tested on adversarial images, and the $KAN\text{-}CNN_{\text{secure}}$ model is trained and tested on a combination of original and adversarial images. All three baseline models are evaluated using the F-1 score metric, with the results presented in Table 3 and Table 4. The adversarial images are generated using the Fast Gradient Sign Method (FGSM) [6].

For these experiments, we vary the **grid size (*GS*)** and **spline order (*SO*)** parameters of the KAN model. These parameters are critical for function approximation.

- **Grid Size (*GS*)**: This parameter determines the number of intervals or points for the B-spline functions, influencing the granularity of the approximation. A finer grid allows the model to capture more details but increases computational costs.
- **Spline Order (*SO*)**: This parameter specifies the degree of the B-splines, enabling smoother and more flexible approximations. Higher spline orders enhance expressivity but can lead to overfitting and increased computational complexity.

By systematically varying *GS* and *SO*, we analyze their impact on model performance across the experiments.

3.1 Experiment 1: MNIST

This experiment is conducted on the MNIST Handwritten Digits dataset [8]. The architecture of the KAN-CNN model is detailed in Table 1, while the initialization function of the KAN Linear layer is provided in Table 2.

We begin by training and testing the models and recording the corresponding F-1 scores, which are presented in Table 3. Each row in the table represents a single run of the experiment. The first run serves as the baseline for subsequent runs.

In the second run, the grid size (*GS*) of the KAN Linear layer is increased from 5 to 100. For the third run, the spline order (*SO*) is increased from 3 to 5, while the grid size is kept at 5, as in the baseline model. Finally, in the fourth run, both *GS* and *SO* are increased, and the corresponding F-1 scores are recorded.

3.2 Experiment 2: CIFAR-10

This experiment is conducted on the CIFAR-10 dataset [7]. We use the same KAN-CNN model, whose architecture is detailed in Table 1. The initialization function of the KAN Linear layer is described in Table 2. Note the boldfaced parameters *grid size* (*GS*) and *spline order* (*SO*), which are varied to perform the experiments.

The steps followed in this experiment are identical to those described in Experiment 3.1, with the additional measure of recording the training time for each run and for each model. The training times are reported in column 6 of Table 4.

4 RESULTS AND DISCUSSION

This section briefly discusses the results of the experiments performed in Section 3. It is important to note that we used the implementation of the KAN-CNN model provided by [4] without increasing the depth of layers or adding additional layers.

Table 3 presents the F-1 scores of the KAN-CNN model on the MNIST dataset. Being a relatively simple dataset, the F-1 scores for all three models—$KAN\text{-}CNN_{\text{original}}$, $KAN\text{-}CNN_{\text{manipulated}}$, and $KAN\text{-}CNN_{\text{secure}}$—are very high.

Table 1: Architecture of the KAN-CNN Model

Layer	Type	Input Channels	Output Channels	Kernel Size	Output Size
1	Convolution	1	32	3	$32 \times 28 \times 28$
2	Max Pooling	32	32	2	$32 \times 14 \times 14$
3	Convolution	32	64	3	$64 \times 14 \times 14$
4	Max Pooling	64	64	2	$64 \times 7 \times 7$
5	KAN Linear	$64 \times 7 \times 7$	256	-	256
6	KAN Linear	256	10	-	10

Table 2: Description of the init Method in the KANLinear Class

Parameter	Description
`in_features`	Number of input features.
`out_features`	Number of output features.
`grid_size`	Size of the grid for B-spline basis functions (default: 5).
`spline_order`	Order of the spline used (default: 3).
`scale_noise`	Scale for noise added to the spline weights (default: 0.1).
`scale_base`	Scale for the base weights (default: 1.0).
`scale_spline`	Scale for the spline weights (default: 1.0).
`enable_standalone_scale_spline`	Enables separate scaling for spline weights (default: True).
`base_activation`	Activation function applied to the base layer (default: `SiLU`).
`grid_eps`	Small value to modify grid computations (default: 0.02).
`grid_range`	Range for the grid used in B-spline basis computation (default: [-1, 1]).

Table 3: KAN-CNN model performance on MNIST Dataset

F1-scores			Hyperparameters		Remarks
KAN-CNN$_{original}$	**KAN-CNN$_{manipulated}$**	**KAN-CNN$_{secure}$**	**Grid Size (GS)**	**Spline Order (SO)**	
0.9909	0.9636	**0.9831**	5	3	**(Reference)**
0.9866	0.9519	**0.9763**	100	3	GS increased
0.9901	0.9635	**0.9812**	5	5	SO increased
0.9880	0.9580	**0.9746**	100	5	GS and SO increased

Table 4: KAN-CNN model performance on CIFAR10 Dataset

KAN-CNNoriginal	KAN-CNNmanipulated	KAN-CNNsecure	GS	SO	Training Time (in sec)	Remarks
0.7278	0.2168	**0.6311**	5	3	(27.93, 35.59, 36.54)	**(Reference)**
0.7088	0.2142	**0.6122**	100	3	(154.51, 157.87, 157.12)	GS increased
0.7204	0.2366	**0.6431**	5	5	(34.76, 37.94, 37.31)	SO increased
0.7107	0.2128	**0.6165**	100	5	(258.27, 261.26, 260.43)	GS and SO increased

However, $KAN\text{-}CNN_{\text{manipulated}}$, trained on manipulated data generated using FGSM, consistently shows lower performance compared to $KAN\text{-}CNN_{\text{secure}}$. Even when the grid size (*GS*) and spline order (*SO*) are increased, $KAN\text{-}CNN_{\text{manipulated}}$ continues to underperform. This indicates that KAN-based models remain vulnerable to simple FGSM-based adversarial attacks and do not exhibit significantly superior performance in this scenario.

Similarly, for Experiment 3.2, the performance of $KAN\text{-}CNN_{\text{manipulated}}$ is worse, and even $KAN\text{-}CNN_{\text{secure}}$ fails to outperform $KAN\text{-}CNN_{\text{original}}$. An improvement in the F-1 score for $KAN\text{-}CNN_{\text{secure}}$ is observed when the spline order (*SO*) is increased from 3 to 5, while keeping the grid size (*GS*) fixed at 5 or 100. However, increasing either *GS* or *SO* results in longer training times. These findings highlight a trade-off between model performance and computational efficiency when adjusting *GS* and *SO*.

The results further demonstrate that hybrid KAN-CNN models, similar to MLP-based CNNs, are susceptible to adversarial manipulations. Nonetheless, their performance can be improved through adversarial learning, suggesting that incorporating adversarial training strategies remains essential for enhancing robustness in such hybrid models.

5 CONCLUSION

This study investigates the robustness of Convolutional Kolmogorov-Arnold Networks (KAN-CNNs) against adversarial perturbations by conducting comprehensive experiments using the Fast Gradient Sign Method (FGSM). Our findings indicate that KAN-CNNs exhibit vulnerabilities under adversarial conditions. Adjustments to key parameters such as spline order and grid size had limited impact on enhancing the models' resistance to attacks. These results highlight that while KANs hold promise for function approximation and general model flexibility, their application in adversarial settings requires further exploration. Future research should focus on developing more sophisticated defense strategies and hybrid architectures that leverage KANs' strengths while mitigating their weaknesses against adversarial threats. By continuing to refine these approaches, it will be possible to build models that are both powerful in function approximation and resilient in the face of adversarial challenges.

REFERENCES

[1] Tal Alter, Raz Lapid, and Moshe Sipper. 2024. On the Robustness of Kolmogorov-Arnold Networks: An Adversarial Perspective. *arXiv preprint arXiv:2408.13809* (2024). https://arxiv.org/abs/2408.13809

[2] Alexander Dylan Bodner, Antonio Santiago Tepsich, Jack Natan Spolski, and Santiago Pourteau. 2024. Convolutional Kolmogorov-Arnold Networks. arXiv:cs.CV/2406.13155 https://arxiv.org/abs/2406.13155

[3] J. Eamon. 2024. CNN-KAN. https://github.com/jakariaemon/CNN-KAN. https://github.com/jakariaemon/CNN-KAN.

[4] Jakaria Emom. 2020. CNN-KAN. https://github.com/jakariaemon/CNN-KAN. Accessed: 2024-10-31.

[5] Z. Liu et. al. 2024. Kolmogorov–Arnold Networks: A New Approach for Small-Scale AI and Science Applications. arXiv:cs.LG/2404.19756v4 https://arxiv.org/abs/2404.19756v4

[6] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. arXiv:stat.ML/1412.6572 https://arxiv.org/abs/1412.6572

[7] Alex Krizhevsky. 2009. Learning Multiple Layers of Features from Tiny Images. (2009). Technical Report.

[8] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. 1998. The MNIST Database of Handwritten Digits. http://yann.lecun.com/exdb/mnist/. Retrieved October 18, 2024.

[9] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. 2015. The Limitations of Deep Learning in Adversarial Settings. arXiv:cs.CR/1511.07528 https://arxiv.org/abs/1511.07528