Conv-KAN under Corruption: A Mentor-Based Study

7113095019 LIU YI-HSI

***Abstract*—In this work, we explore the robustness of the Convolutional Kolmogorov-Arnold Network (Conv-KAN), a novel architecture based on functional decomposition. We evaluate its performance on the CIFAR-100 dataset and its corrupted variant, CIFAR-100C. Furthermore, we incorporate a mentor-based oracle model that predicts whether the classifier's predictions are correct, based on confidence and entropy features. We compare Conv-KAN against ResNet-50 and analyze the reliability of both models under distribution shift, revealing the trade-off between expressiveness and robustness.**

I. Introduction

Deep convolutional neural networks (CNNs) have achieved remarkable performance on standard datasets. However, they are often vulnerable to distributional shifts such as corruptions or adversarial perturbations. In this study, we focus on the recently proposed Conv-KAN architecture and evaluate its robustness and reliability under corruption. We also employ a mentor oracle model that acts as a second-level evaluator, predicting whether the classifier's output is likely to be correct, enhancing error awareness.

II. Method

A. Kolmogorov-Arnold Function Representation in KAN

The core of Kolmogorov-Arnold Networks is based on the Kolmogorov-Arnold representation theorem [1], which states that any multivariate continuous function $f(x_1, \ldots, x_n)$ can be written as a superposition of univariate functions:

$$f(x_1, \ldots, x_n) = \sum_{q=0}^{2n} \phi_q \left(\sum_{p=1}^{n} \psi_{q,p}(x_p) \right)$$

In the context of neural networks, KAN models parameterize each ϕ_q and $\psi_{q,p}$ as **trainable univariate functions** rather than fixed weights. This allows the network to learn complex non-linear transformations in a highly interpretable and modular way.

Each KAN layer contains a set of such univariate functions:

$$\Phi = \{\phi_{q,p}\}, \quad p = 1, 2, \ldots, n_{\text{in}}, \quad q = 1, 2, \ldots, n_{\text{out}}$$

where n_{in} is the number of input dimensions (channels) and n_{out} is the number of output dimensions.

Each univariate function $\phi_{q,p}(x)$ is implemented using **B-spline basis functions** as:

$$\phi(x) = \sum_i c_i B_i(x)$$

where:

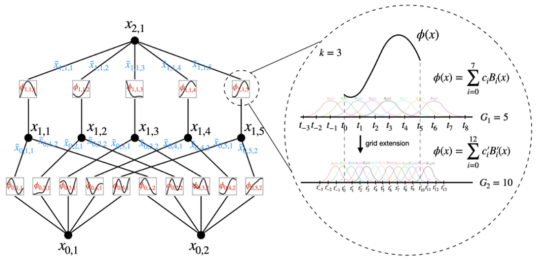

Fig. 1. Left: Notations of activations that flow through the network. Right: an activation function is parameterized as a B-spline, which allows switching between coarse-grained and fine-grained grids.

- $B_i(x)$ are fixed spline basis functions (e.g., cubic B-splines)
- c_i are learnable coefficients

This spline-based formulation enables smooth and flexible function approximation, with interpretability due to the limited degrees of freedom in each univariate transformation.

To ensure model simplicity and prevent overfitting, regularization terms are typically applied:

- **Smoothness regularization:** penalizes rapid changes between spline coefficients.
- **Sparsity constraints:** encourages minimal basis function activation, typically via L1 penalties on c_i.

Instead of using fixed linear filters in convolution, Conv-KAN replaces weights with learnable univariate functions $\phi_{c,mn}(\cdot)$ for each input channel c and spatial kernel position (m, n). The convolution operation becomes:

$$y_{i,j} = \sum_{c=1}^{C} \sum_{(m,n)\in K} \phi_{c,mn}(x_{i+m,j+n}^{(c)})$$

Each ϕ is parameterized via spline or piecewise-linear bases, and trained end-to-end with smoothness and sparsity regularization.

B. Convolutional Kolmogorov-Arnold Network (Conv-KAN)

Conv-KAN [2] is inspired by the Kolmogorov-Arnold representation theorem, which states that any multivariate function can be decomposed into a combination of univariate functions:

We implement a compact Conv-KAN model named `KANC_MLP_Big` designed for CIFAR-100 classification. It consists of two KAN-based convolutional layers followed by a max-pooling operation and a fully connected multi-layer perceptron (MLP) head.

The architecture is defined as follows (Fig 2):

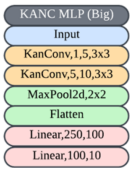

Fig. 2. KAN Architectures used in our CKAN model. The two convolutional layers employ univariate function parameterization.

- **KAN Convolution Layer 1:** Converts 1-channel input to 5 channels using a (3×3) kernel, stride 1, and padding 1.
- **Max Pooling:** (2×2) pooling to reduce spatial resolution.
- **KAN Convolution Layer 2:** Further maps 5 channels to 10 channels with the same kernel and stride.
- **Flatten + MLP:** Feature maps are flattened to a vector and passed to a fully connected layer (300 hidden units), followed by ReLU and a final classification layer of size 100.

This CKAN variant is relatively shallow but leverages univariate function parameterizations in the convolutional layers to enhance representational flexibility.

C. Mentor Oracle for Prediction Failure Detection

Inspired by the oracle-based reliability detection framework [3], we train a mentor model to classify prediction correctness.

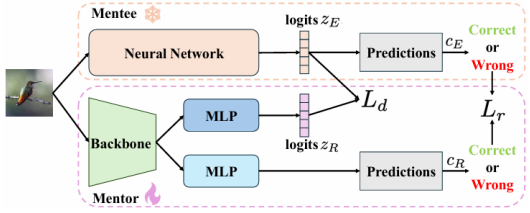

Fig. 3. Mentor model architecture: a binary classifier that predicts whether the base model's prediction is correct.

To assess the reliability of the base model (e.g., CKAN or ResNet-50), we introduce a secondary model called the **mentor**. This model acts as an oracle to estimate whether a given prediction is likely to be correct or not, by analyzing internal features of the base model.

Inputs to the Mentor: For each input sample x, the mentor receives the following as input:

- The logits vector $z \in \mathbb{R}^C$ produced by the base model before softmax.
- The maximum softmax confidence: $\max_i p(y = i|x)$.
- The entropy of the prediction distribution:

$$H(x) = -\sum_{i=1}^{C} p(y = i|x) \log p(y = i|x)$$

These features are concatenated into a single input vector for the mentor classifier.

Labels for Supervision: The target label for the mentor is binary:

$$\text{Label}(x) = \begin{cases} 1 & \text{if } \arg\max f(x) = y_{\text{true}} \\ 0 & \text{otherwise} \end{cases}$$

The mentor is trained using binary cross-entropy loss to distinguish correct vs. incorrect predictions made by the base model.

Architecture and Training: The mentor can be a lightweight MLP or CNN-based classifier. In our experiments, we use a simple two-layer fully connected network:

- Input dimension: C (logits) + 2 (confidence + entropy)
- Hidden layer: 64 units with ReLU activation
- Output: Single sigmoid unit for binary prediction

Optimization is performed using Adam with a learning rate of $1e{-}3$ for 10 epochs.

Objective: The mentor provides a signal on when to trust the model's predictions. In deployment, this can be used to trigger fallbacks or abstain decisions when the prediction is deemed unreliable.

III. Experiment

A. Datasets

We conduct experiments on:

- **CIFAR-100** [4]: 50,000 training and 10,000 clean test images across 100 classes.
- **CIFAR-100C** [5]: A corrupted test set with 15 types of common image corruptions (e.g., fog, blur, noise).

B. Models

We compare:

- **Conv-KAN**: Trained on CIFAR-100 from scratch.
- **ResNet-50**: Standard CNN baseline, trained with identical pipeline.

C. Evaluation Metrics

- **Top-1 Accuracy**: On CIFAR-100 train/test and CIFAR-100C.
- **Mentor Accuracy (BCA)**: Binary correctness accuracy on predicting whether model predictions are correct.

IV. Results and Analysis

TABLE I
Performance on CIFAR-100 and CIFAR-100C with Mentor Evaluation

Model	Train Acc	Test Acc	CIFAR-100C	Mentor
ResNet-50	94.88%	89.43%	33.89%	53.03%
CKAN	**96.01%**	83.69%	16.99%	26.07%

As shown in Table I, CKAN achieves higher training accuracy than ResNet-50, indicating stronger expressivity on clean data. However, its performance on CIFAR-100C drops significantly, suggesting poor robustness under distributional shift.

The mentor model performs significantly better for ResNet-50 than CKAN. A Binary Correctness Accuracy (BCA) of 53.03% for ResNet-50 vs. 26.07% for CKAN suggests that CKAN's errors are less structured and harder to predict, possibly due to less calibrated confidence.

V. Conclusion

We analyzed the robustness and error predictability of Conv-KAN on standard and corrupted datasets. While Conv-KAN demonstrates strong learning capacity, it suffers from poor generalization and unreliable confidence under corruption. The mentor-based oracle system reveals that its prediction errors are harder to detect, calling for future improvements such as confidence calibration or robust training strategies.

References

[1] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacic, Thomas Y. Hou, Max Tegmark:KAN: Kolmogorov-Arnold Networks. CoRR abs/2404.19756, 2024.

[2] Alexander Dylan Bodner, Antonio Santiago Tepsich, Jack Natan Spolski, Santiago Pourteau: Convolutional Kolmogorov-Arnold Networks. CoRR abs/2406.13155, 2024.

[3] Shuangpeng Han, Mengmi Zhang:Unveiling AI's Blind Spots: An Oracle for In-Domain, Out-of-Domain, and Adversarial Errors. CoRR abs/2410.02384, 2024.

[4] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical Report, University of Toronto, 2009.

[5] Dan Hendrycks and Thomas Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In ICLR, 2019.