Catastrophic Forgetting in Kolmogorov-Arnold Networks

Mohammad Marufur Rahman[1], Guanchu Wang[2], Kaixiong Zhou[3], Minghan Chen[1], Fan Yang[1]

[1]Department of Computer Science, Wake Forest University
[2]Department of Computer Science, University of North Carolina at Charlotte
[3]Department of Electrical and Computer Engineering, North Carolina State University
rahmm224@wfu.edu, gwang16@charlotte.edu, kzhou22@ncsu.edu, chenm@wfu.edu, yangfan@wfu.edu

Abstract

Catastrophic forgetting is a longstanding challenge in continual learning, where models lose knowledge from earlier tasks when learning new ones. While various mitigation strategies have been proposed for Multi-Layer Perceptrons (MLPs), recent architectural advances like Kolmogorov-Arnold Networks (KANs) have been suggested to offer intrinsic resistance to forgetting by leveraging localized spline-based activations. However, the practical behavior of KANs under continual learning remains unclear, and their limitations are not well understood. To address this, we present a comprehensive study of catastrophic forgetting in KANs and develop a theoretical framework that links forgetting to *activation support overlap* and *intrinsic data dimension*. We validate these analyses through systematic experiments on synthetic and vision tasks, measuring forgetting dynamics under varying model configurations and data complexity. Further, we introduce KAN-LoRA, a novel adapter design for parameter-efficient continual fine-tuning of language models, and evaluate its effectiveness in knowledge editing tasks. Our findings reveal that while KANs exhibit promising retention in low-dimensional algorithmic settings, they remain vulnerable to forgetting in high-dimensional domains such as image classification and language modeling. These results advance the understanding of KANs' strengths and limitations, offering practical insights for continual learning system design.

Code — https://github.com/marufur-cs/AAAI26
Full version — https://arxiv.org/abs/2511.12828

Introduction

Catastrophic forgetting, also known as catastrophic interference (McCloskey and Cohen 1989), a fundamental challenge in machine learning, occurs when a neural network loses previously acquired information while learning from new data. This phenomenon is central to the field of continual learning, where models are trained incrementally on non-stationary data distributions (Ven, Soures, and Kudithipudi 2024; Kemker et al. 2017). Moreover, it is prevalent in a wide range of research fields such as meta-learning (Spigler 2020), domain adaptation (Xu et al. 2020), foundation models (Luo et al. 2025), and reinforcement learning (Zhang et al. 2023), where the retention of prior knowledge is critical for generalization and stability.

Multi-Layer Perceptrons (MLPs) are inherently prone to catastrophic forgetting (Zenke, Poole, and Ganguli 2017). Several techniques have been proposed to overcome catastrophic forgetting in MLPs (Wang et al. 2025; De Lange et al. 2022). Regularization-based techniques (Kirkpatrick et al. 2017; Kong et al. 2024) impose restrictions on the network's weight adjustments, hence reducing the likelihood of interference with previously acquired knowledge. Architecture-based methods (Yoon et al. 2018; Mirzadeh et al. 2022) mitigate forgetting by modifying the network's architecture to accommodate new information. Rehearsal-based methods (Buzzega et al. 2020; Riemer et al. 2019) aim to preserve prior information by including data samples from earlier learning sessions during the current session. Although catastrophic forgetting has been extensively studied in MLPs, it remains relatively underexplored in emerging fundamental neural architectures such as Kolmogorov-Arnold Networks (KANs) (Liu et al. 2025).

KANs, inspired by the Kolmogorov-Arnold representation theorem (Kolmogorov 1961), have emerged as a promising alternative neural network architecture to traditional MLPs. KANs were introduced to address several fundamental limitations of MLPs. Unlike MLPs, which rely on fixed activation functions, KANs utilize learnable one-dimensional activation functions (spline) along the edges of the network. Splines can be easily adjusted locally and are accurate for low-dimensional functions, giving KANs the potential to avoid forgetting. As spline bases are local, a data sample affects only a few related spline coefficients, leaving other coefficients unaltered. This unique architecture enables KANs to learn non-linear relations more effectively and to be more robust against catastrophic forgetting in continual learning scenarios (Lee et al. 2025). KANs have been successfully applied in various domains (Yang and Wang 2025; Abd Elaziz, Ahmed Fares, and Aseeri 2024), yet studies around their effectiveness in mitigating catastrophic forgetting in continual learning are still quite limited.

Only a few pioneer works have studied the catastrophic forgetting phenomenon in KANs under the continual learning settings. Lee et al. recently proposed a simple and heuristic strategy, WiseKAN, which allocates distinct parameter subspaces to different tasks to mitigate catastrophic forgetting in KANs. Liu et al. demonstrated robustness of KANs against catastrophic forgetting using synthetic data on regression tasks. Furthermore, studies proposed modified KANs to gain robust retention in specific domains, such as classification (Hu et al. 2025) and face forgery detection (Zhang et al. 2025). Despite these initial efforts, a comprehensive understanding of forgetting in KANs remains elusive, particularly in terms of theoretical characterization and empirical evaluation on practical real-world tasks.

To bridge the gap, we first develop a theoretical framework for understanding catastrophic forgetting in KANs by formulating several key factors such as *activation support overlap* and *intrinsic data dimension*. Our analysis reveals that forgetting in KANs scales linearly with activation support overlap and grows exponentially with the intrinsic dimensionality of the task manifold, offering a principled explanation for KANs' robustness in simple tasks and vulnerability in complex domains. Building on these insights, we then conduct extensive empirical experiments comparing KANs with MLPs across a spectrum of tasks, including the low-dimensional synthetic addition and the high-dimensional image classification. Furthermore, we design a novel LoRA (Hu et al. 2022) adapter based on KAN, termed KAN-LoRA, to enable continual fine-tuning of language models (LMs) for sequential knowledge editing. Across all experimental settings, our results consistently corroborate the theoretical analysis, illustrating that while KANs achieve strong retention in structured and low-dimensional tasks, they remain susceptible to forgetting in high-dimensional domains, thereby highlighting both the strengths and limitations of KANs in practical continual learning scenarios. Our main contributions are summarized as below:

- We develop a theoretical framework for catastrophic forgetting in KANs, deriving formal retention bounds based on activation support overlap and intrinsic data dimension, and characterizing how forgetting evolves;
- We validate the theoretical analysis through empirical experiments on synthetic and image data, demonstrating strong alignment between the support overlap, task complexity, and the observed forgetting behavior;
- We introduce KAN-LoRA, a novel KAN-based adapter for continual fine-tuning of LMs, and evaluate its performance in sequential knowledge editing, highlighting both the strengths and limitations of KANs in practice.

Preliminary

Catastrophic Interference

Neural networks learn the non-linear mapping between input and output spaces by finding a region in the parameter space where the network achieves expected behavior (Bishop 1994). When the neural network is trained on new data, the network's parameter space shifts accordingly to capture the mapping between new input and output space. As a result, performance degrades on prior data. This phenomenon was termed as *catastrophic interference* by McCloskey and Cohen. It was observed in many machine learning models such as support vector machine (Ayad 2014), but is particularly pronounced in connectionist models (e.g., MLPs) due to their dense and globally updated parameterizations (French 1999). Standard neural training algorithms typically lack the capacity to progressively learn new tasks without overwriting previous knowledge (Aleixo et al. 2023), making them especially vulnerable to catastrophic interference. Such limitation has motivated continual learning studies (De Lange et al. 2022) to develop algorithms and architectures that enable models to acquire new knowledge incrementally while preserving performance on learned tasks.

Kolmogorov-Arnold Networks

KANs are inspired by the Kolmogorov-Arnold representation theorem, which states that a finite sum of continuous univariate functions and the binary addition operation can represent any multivariate continuous function $f(\mathbf{x})$ in a specified bounded domain (Kolmogorov 1961). Based on the theorem, function $f(\mathbf{x})$ can be represented as

$$f(\mathbf{x}) = f(x_1, x_2, ..., x_n) = \sum_{q=1}^{2n+1} \Psi_q(\sum_{p=1}^{n} \psi_{p,q}(x_p)),$$

where n is the number of input variables, $\psi_{p,q} : [0,1] \to \mathbb{R}$, and $\Psi_q : \mathbb{R} \to \mathbb{R}$. This equation indicates that a 2-layer network with n inputs and $(2n+1)$ outputs is sufficient to represent $f(\mathbf{x})$ by the sums of univariate functions. However, 1-D function ψ can be fractal and non-smooth, making it unlearnable (Girosi and Poggio 1989) in practice. KANs solve this issue by generalizing the theorem to multiple layers with arbitrary width. Formally, KANs consisting L layers can be indicated by

$$f(x) = (\Phi_{L-1} \circ \Phi_{L-2} \circ \cdots \circ \Phi_1 \circ \Phi_0)(x),$$

$$\Phi_\ell = \begin{bmatrix} \phi_{\ell,1,1} & \phi_{\ell,2,1} & \cdots & \phi_{\ell,d_\ell,1} \\ \phi_{\ell,1,2} & \phi_{\ell,2,2} & \cdots & \phi_{\ell,d_\ell,2} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{\ell,1,N_\ell} & \phi_{\ell,2,N_\ell} & \cdots & \phi_{\ell,d_\ell,N_\ell} \end{bmatrix},$$

where $\circ$ indicates matrix multiplication, Φ_ℓ is the function matrix that corresponds to the ℓ-th layer, d_ℓ and N_ℓ are the number of input coordinates and univariate branches respectively. The univariate function ϕ is defined as the weighted sum of a base and a spline function (Liu et al. 2025).

Forgetting Analysis

We first introduce a formal measure of *forgetting* and the notation needed to analyze how KAN's local activations give rise to both perfect retention and task interference. Let $f^{(t)}$ denote the KAN obtained after sequentially training on tasks $1, 2, \ldots, t$, and define

$$F_i = L\big(f^{(T)}, \mathcal{D}_i\big) - L\big(f^{(i)}, \mathcal{D}_i\big)$$

as the *forgetting* on task i, where $L(f, \mathcal{D})$ is the expected loss under data distribution $\mathcal{D}$. We index layers by $\ell \in \{1, \ldots, L\}$, and within each layer we number the input coordinates (pre-activations) by $p \in \{1, \ldots, d_\ell\}$ and the individual univariate branches by $q \in \{1, \ldots, N_\ell\}$.

To capture where each unit actually "turns on", we define the *activation support* of branch $\phi_{\ell,p,q}$ for task i as

$$S^{(i)}_{\ell,p,q} = \{\, z \in \mathbb{R} : \phi_{\ell,p,q}(z) \neq 0\},$$

the subset of real inputs on which that branch contributes non-zero output. We measure the size of these one-dimensional sets by the Lebesgue measure $\mu(\cdot)$[1]. With these setups, we can represent the maximum one-dimensional overlap between any single activation for tasks i and j as

$$\Delta_{i,j} = \max_{\ell,p,q} \; \mu\big(S^{(i)}_{\ell,p,q} \cap S^{(j)}_{\ell,p,q}\big),$$

which will serve as the key link between KAN's architectural locality and the bounds on catastrophic forgetting. [2]

Bounded Retention

We now precisely characterize when KAN achieves perfect retention and how any residual overlap translates into bounded forgetting. Overall, we demonstrate that KAN's local-support activations act as task-specific feature detectors: if their "on" regions never coincide across tasks, earlier knowledge remains untouched, and when they do overlap, forgetting grows in direct proportion to that overlap.

Lemma 1 (Zero-Overlap Retention)**.** *Suppose for an earlier task i and every later task $j > i$ the maximal support-overlap satisfies $\Delta_{i,j} = 0$. Then*

$$F_i \;=\; L\big(f^{(T)}, \mathcal{D}_i\big) \;-\; L\big(f^{(i)}, \mathcal{D}_i\big) \;=\; 0.$$

Remark on Lemma 1

When no branch ever activates on both tasks, gradient updates for new tasks cannot affect the parameters responsible for task i, guaranteeing *exact* retention. This lemma formalizes the intuition that **truly disjoint representations cannot interfere**.

Theorem 1 (Retention Bound via Overlap)**.** *Under the additional assumptions that each branch $\phi_{\ell,p,q}$ is L_ℓ-Lipschitz[3] and the loss is bounded by C, for any $j > i$ we have*

$$F_i \;\le\; C \sum_{\ell=1}^{L} N_\ell \, L_\ell \, \Delta_{i,j}.$$

Remark on Theorem 1

This bound reveals that any forgetting in KANs **scales linearly with the one-dimensional overlap** $\Delta_{i,j}$ and the network's size parameters. Importantly, when $\Delta_{i,j} = 0$, it collapses to $F_i \le 0$, recovering exact retention as a special case and shows that small overlaps incur proportionally small forgetting.

[1] Lebesgue measure generalizes the length to a broader class of sets. Here, it corresponds to the total length of the activation region.

[2] Detailed derivations for all theorems are in the full version.

[3] ϕ is L-Lipschitz if $|\phi(z_1) - \phi(z_2)| \le L|z_1 - z_2|$ for all $z_1, z_2 \in \mathbb{R}$. Here, L_ℓ quantifies the spline smoothness in layer ℓ.

Cumulative Forgetting

While Theorem 1 guarantees zero or bounded forgetting on a per-task basis, real continual learning involves sequences of overlapping tasks whose supports may intersect in complex ways. To capture the deeper dynamics of forgetting in KANs, we further analyze at the branch level and consider cumulative contributions and effects.

Theorem 2 (Branch-wise Cumulative Forgetting)**.** *Under the Lipschitz and bounded-loss assumptions of Theorem 1, the forgetting on task i after training on all subsequent tasks $i+1, \ldots, T$ can be decomposed as*

$$F_i \;\le\; C \sum_{\ell=1}^{L} \sum_{p=1}^{d_\ell} \sum_{q=1}^{N_\ell} L_\ell \,\Big[\sum_{j=i+1}^{T} \mu\big(S^{(i)}_{\ell,p,q} \cap S^{(j)}_{\ell,p,q}\big)\Big].$$

Remark on Theorem 2

Forgetting in KANs is driven not only by the largest single overlap but also by the **total overlap each branch experiences across tasks**. Branches with frequent cross-task activation contribute disproportionately to forgetting, suggesting that sparsifying or diversifying supports could mitigate interference.

Corollary 1 (Expected Forgetting under Random Supports)**.** *If each branch's supports for task j are independently drawn as length-s_j intervals in $[0,1]$, then in expectation*

$$\mathbb{E}[F_i] \;\le\; C \sum\nolimits_{\ell=1}^{L} N_\ell L_\ell \; \sum\nolimits_{j=i+1}^{T} s_i \, s_j.$$

Remark on Corollary 1

Forgetting in KANs grows with the pairwise products of support sizes: a **difficult task** (*with large s_j*) can retroactively erode performance on earlier tasks, and **longer task sequences** amplify this effect.

Corollary 2 (Saturation via Union-Bound)**.** *Let*

$$U^{(i)}_{\ell,p,q} = \bigcup_{j=i+1}^{T} \big(S^{(i)}_{\ell,p,q} \cap S^{(j)}_{\ell,p,q}\big)$$

be the union of all overlaps for branch (ℓ, p, q). Then

$$F_i \;\le\; C \sum_{\ell=1}^{L} \sum_{p=1}^{d_\ell} \sum_{q=1}^{N_\ell} L_\ell \; \mu\big(U^{(i)}_{\ell,p,q}\big),$$

with $\mu(U^{(i)}_{\ell,p,q}) \le \min\big(\sum_{j=i+1}^{T} \Delta_{i,j},\; \mu(S^{(i)}_{\ell,p,q})\big)$.

Remark on Corollary 2

Forgetting in KANs will **saturate**, once a branch's **full activation support is covered by overlaps**. After enough highly overlapping tasks, further tasks cannot increase forgetting beyond that support size.

Complexity-Induced Forgetting

Beyond mere pairwise overlap, we further conduct theoretical analysis by examining how intrinsic task complexity drives forgetting in KANs. In particular, we show that when tasks live on data manifolds of differing intrinsic dimensions, the degree of forgetting can change dramatically. This complements our earlier results by linking forgetting directly to geometric measures of task difficulty.

Theorem 3 (Intrinsic-Dimension Forgetting Rate)**.** *Suppose task t generates data concentrated on a compact submanifold $\mathcal{M}_t \subset [0,1]^n$ of intrinsic dimension d_t, and each univariate branch's activation support can be enclosed within an r-ball in the pre-activation domain. Then, for any earlier task i and later task j, the expected support overlap satisfies*

$$\mathbb{E}\big[\mu(S^{(i)}_{\ell,p,q} \cap S^{(j)}_{\ell,p,q})\big] = O\big(r^{\,d_i+d_j}\big),$$

and hence the forgetting on task i obeys

$$F_i \;=\; O\Big(\sum_{j=i+1}^{T} N_{\text{tot}}\,\bar{L}\,r^{\,d_i+d_j}\Big),$$

where $N_{\text{tot}} = \sum_\ell N_\ell$ counts the total number of univariate branches and $\bar{L}$ is an average Lipschitz constant.

Remark on Theorem 3

Tasks with **higher intrinsic dimension produce exponentially smaller "gaps"** in their activation partitions, so even modest support radius r could incur large overlaps and thus substantial forgetting. Conversely, low-dimensional tasks enjoy near-zero overlap and stable retention in continuous learning.

Corollary 3 (Retention for Low-Dimensional Tasks)**.** *If every subsequent task j has intrinsic dimension $d_j \le D$, then*

$$F_i = O\big(T\,N_{\text{tot}}\,\bar{L}\,r^{\,d_i+D}\big),$$

which becomes negligible when $d_i + D$ is sufficiently small.

Remark on Corollary 3

When both the original task and all new tasks inhabit **low-dimensional manifolds**, their **activation overlaps shrink exponentially** in dimension, protecting against forgetting even over long task sequences.

Corollary 4 (Fragmentation Mitigates Complexity)**.** *If each branch's support for task t is split into k_t disjoint intervals (effective radius r/k_t), then Theorem 3's rate improves to*

$$F_i = O\Big(\sum_{j=i+1}^{T} N_{\text{tot}}\,\bar{L}\,\big(r/k_i\big)^{d_i}\big(r/k_j\big)^{d_j}\Big).$$

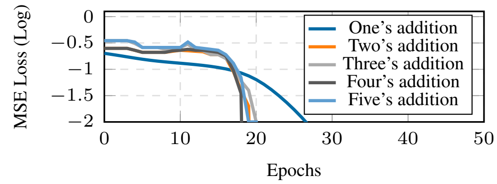

Figure 1: MSE loss in logarithmic scale for five different binary addition tasks during training on *one's addition* task.

Remark on Corollary 4

KANs can **sharply reduce forgetting** on high-dimensional tasks **by increasing support fragmentation**, which effectively refines each branch's receptive field and trades off coarser representation granularity for higher retention fidelity.

Overall, Theorem 3 and its corollaries illuminate how KAN's forgetting depends on the deeper geometric complexity of task data and the combinatorial structure of activation supports. This perspective provides actionable guidance for designing KAN architectures and pruning strategies.

Experiments

We conduct a series of experiments to empirically validate our theoretical findings and assess KANs' forgetting behavior across diverse settings. Starting with low-dimensional synthetic tasks, we analyze retention under binary and decimal addition. We then evaluate KANs on high-dimensional image classification benchmarks and finally test KAN-LoRA for continual knowledge editing in LMs. These experiments effectively illustrate how model architecture and task complexity shape the forgetting in KANs. [4]

Binary and Decimal Addition

Experimental Setup We construct five synthetic tasks under a continual setting. Each task is defined by fixing one of the operands in a two-digit addition problem. Specifically, Task 1 involves *one's addition*, where the digit 1 is added to every digit from 1 to 9. Task 2 is *two's addition*, and so forth up to *five's addition* in Task 5. We apply this construction for both binary and decimal representations of digits. This setup enables us to systematically evaluate forgetting across increasingly overlapping arithmetic patterns.

Model Configuration Our KAN model is configured with three input nodes, two hidden neurons, and two output neurons to perform addition of two 4-bit binary numbers. At each step, the model receives two input bits (one from each number) along with a carry bit, and outputs the corresponding sum bit and the carry bit for the next step. The univariate functions in KANs are modeled using B-splines (Prautzsch, Boehm, and Paluszny 2002), where the grid size determines the number of intervals in the spline. A larger grid size provides greater flexibility, allowing the splines to capture more

[4]More details on the experiments are in the full version.

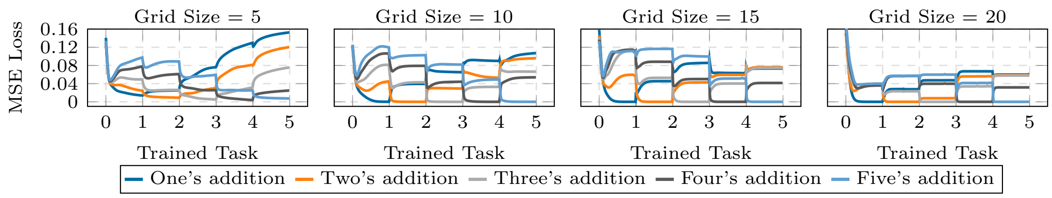

Figure 2: MSE loss during sequential training on five different decimal addition tasks, from *one's* to *five's addition* facts.

complex functions (Liu et al. 2025). For binary addition tasks, we use a grid size of 5. For decimal addition tasks, we evaluate KANs with grid sizes of 5, 10, 15, and 20. As a baseline, we compare the KAN's performance on binary addition to a specialized MLP architecture (Ruiz-Garcia 2022) designed to learn binary addition rules in a continual learning setting without catastrophic forgetting.

Evaluation Results. The KAN model is sequentially trained on five binary addition tasks. Figure 1 shows the Mean Squared Error (MSE) loss (log scale) for all five tasks during training on the *one's addition* task. Notably, even during training on the first task, the losses for subsequent tasks also decrease significantly, indicating strong positive correlation. As training progresses, the model maintains stable performance on earlier tasks, with overall forgetting remaining below 1×10^{-6} after all five sessions, showing a strong resilience to catastrophic forgetting. The KAN outperforms the specialized MLP model designed for binary addition. While the MLP requires sequential training on both *one's* and *two's addition* tasks to succeed, the KAN model generalizes effectively after learning just the *one's addition* task.

Similarly, for the decimal addition, the KAN is trained sequentially over five tasks. Unlike the binary setting, the model does not fully learn subsequent tasks during training on the first one. As training progresses, clear signs of catastrophic forgetting emerge. Figure 2 shows that learning a new task leads to a noticeable decline in performance on previous tasks. However, the severity of forgetting decreases as the grid size increases, suggesting that finer spline resolution improves retention. After completing all five tasks, a clear forgetting pattern appears: performance deteriorates more significantly for tasks that are farther in time from the most recent training, indicating that earlier tasks suffer more from forgetting. These observations empirically support our analysis in Corollary 1. On one hand, increasing the grid size reduces each spline's support length, thereby decreasing pairwise overlaps and mitigating the forgetting. On the other hand, later tasks, which have larger effective support sizes due to increased digit variability, lead to greater cumulative interference, consistent with the $s_i s_j$ dependence.

Tables 1 and 2 further present empirical evidences supporting Theorems 1 and 2. In Table 1, for each pair of tasks selected from the five decimal addition tasks, the ratio $F_i/\Delta_{i,j}$ remains approximately *constant*, suggesting that the forgetting F_i scales linearly with the support overlap $\Delta_{i,j}$ between tasks i and j. Similarly, Table 2 shows that the ratio between the observed forgetting and the cumulative support overlap[5] $\sum_{i+1}^{T} \mu(S^{(i)} \cap S^{(j)})$ is also nearly *constant*, indicating a linear dependence. Additionally, this ratio becomes more stable (i.e., exhibits lower variance) as the grid size of KANs increases, revealing that finer-grained spline meshes promote more consistent forgetting behavior.

Task (i)	Task (j)	Grid 10		Grid 15		Grid 20	
		F_i	$\frac{F_i}{\Delta_{i,j}}$	F_i	$\frac{F_i}{\Delta_{i,j}}$	F_i	$\frac{F_i}{\Delta_{i,j}}$
1	2	0.46	**0.74**	0.45	**0.74**	0.32	**0.61**
2	3	0.45	**0.73**	0.40	**0.67**	0.34	**0.64**
3	4	0.52	**0.77**	0.46	**0.74**	0.32	**0.63**
4	5	0.44	**0.72**	0.42	**0.68**	0.32	**0.64**

Table 1: Retention bounds across KANs and tasks.

Task (i)	Task (j)	Grid 10		Grid 15		Grid 20	
		F_i	$\frac{F_i}{\sum \mu^{ij}}$	F_i	$\frac{F_i}{\sum \mu^{ij}}$	F_i	$\frac{F_i}{\sum \mu^{ij}}$
1	2, 3, 4, 5	0.68	**0.15**	0.62	**0.15**	0.57	**0.16**
2	3, 4, 5	0.67	**0.16**	0.51	**0.15**	0.44	**0.16**
3	4, 5	0.39	**0.16**	0.39	**0.16**	0.29	**0.16**
4	5	0.25	**0.18**	0.19	**0.17**	0.16	**0.17**

Table 2: Cumulative bounds across KANs and tasks.

Image Classification

Experimental Setup To evaluate KANs in real-world settings, we assess their forgetting behavior with continual learning using CIFAR-10, Tiny-ImageNet, and MNIST datasets. CIFAR-10 consists $(32 * 32)$-pixel images of 10 evenly distributed classes. To simulate a class-incremental continual-learning scenario, the dataset is divided into five sequential tasks, each containing images from two different classes. A similar five-task setup is constructed for Tiny-ImageNet dataset, by selecting 10 classes from 200 different classes of $(64 * 64)$-pixel images. MNIST, $(28 * 28)$ pixels, is likewise divided into five sequential tasks. These three datasets vary in intrinsic dimensionality, where MNIST has the lowest while Tiny-ImageNet has the highest dimension.

Model Configuration We adopt a Transformer-based architecture for image classification, in which all MLP layers are replaced with KAN layers, resulting in the KAN-Transformer model (Yang and Wang 2025). This modification intends to utilize the adaptive capacity of KANs within

[5]Simplify to $\sum \mu^{ij}$ in Table 2 notations.

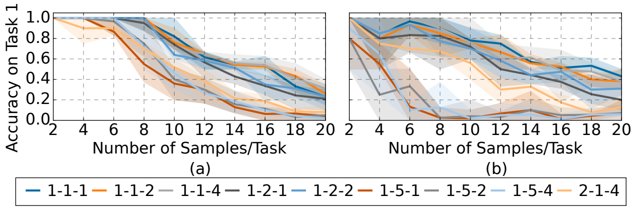

Figure 3: Task 1 accuracy after sequential training on task 1 and 2 from CIFAR-10, comparing (a) KAN-Transformer and (b) MLP-Transformer. Model configuration is labeled as (#classification *layers* – #encoder *blocks* – #attention *heads*).

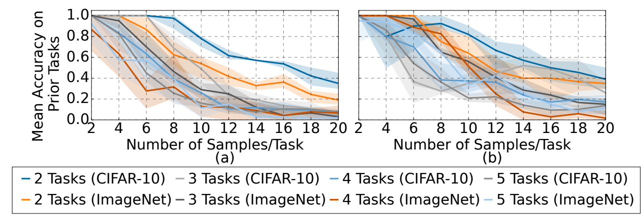

Figure 4: Average accuracy on previously learned tasks after training on 2 to 5 tasks with varying sample sizes from CIFAR-10 and Tiny-ImageNet datasets. Sub-figures show results for (a) KAN-Transformer and (b) MLP-Transformer.

MNIST			CIFAR-10			Tiny-ImageNet		
Quantize label (Q)	Shape (S)	$\frac{\log(F_i)}{d_i}$	Quantize label (Q)	Shape (S)	$\frac{\log(F_i)}{d_i}$	Quantize label (Q)	Shape (S)	$\frac{\log(F_i)}{d_i}$
2	8×8	**0.074**	8	8×8	**0.046**	8	16×16	**0.052**
2	16×16	**0.071**	8	16×16	**0.053**	8	32×32	**0.054**
2	28×28	**0.074**	8	32×32	**0.046**	8	64×64	**0.052**
4	28×28	**0.071**	16	32×32	**0.053**	16	64×64	**0.051**
8	28×28	**0.075**	32	32×32	**0.050**	32	64×64	**0.049**
16	28×28	**0.073**	64	32×32	**0.048**	64	64×64	**0.048**
32	28×28	**0.075**	128	32×32	**0.047**	128	64×64	**0.047**

Table 3: Forgetting rate for varied intrinsic dimensions.

the Transformer framework for continual learning scenarios. To provide a fair and competitive baseline, we also design an MLP-based transformer model augmented with the EWC (Kirkpatrick et al. 2017) regularization technique.

Evaluation Results. Figure 3 illustrates the accuracy on task 1 after sequential training of the KAN (with grid size 10) and the MLP model on tasks 1 and 2 from CIFAR-10, evaluated across various model configurations and increasing sample sizes per task. Both architectures retain high accuracy in shallow settings with a single encoder block, attention head, and classification layer. Notably, the KAN model demonstrates superior retention, maintaining 100% accuracy on task 1 up to 8 samples per task, whereas the MLP model drops to around 80%. As the number of encoder blocks and classification layers increases, performance declines sharply, particularly in MLPs, which suggests deeper networks are more susceptible to catastrophic forgetting.

Figure 4 summarizes the impact of varying the number of samples per task during continual learning, evaluated across different task counts (ranging from 2 to 5) for both CIFAR-10 and Tiny-ImageNet. All models use one single encoder block, attention head, and classification layer. On CIFAR-10, KAN models exhibit better retention than their MLP counterparts, particularly when trained on a smaller number of tasks. In contrast, MLP models outperform KANs on the more challenging Tiny-ImageNet dataset. These results underscore the increasing difficulty of continual learning in KANs as both the number of tasks and the underlying data complexity grow. Moreover, the performance curves in Figure 4a suggest a clear saturation effect: after a certain number of highly overlapping tasks, additional training yields diminishing increases in forgetting, consistent with the bounded cumulative interference described in Corollary 2, where support unions eventually stabilize.

Table 3 further presents empirical evidence supporting Theorem 3. Forgetting F_i is measured on task 1 after sequential training on all five tasks from MNIST, CIFAR-10, and Tiny-ImageNet. To vary the intrinsic dimension d_i, the images are quantized using different label sets (Q) and resized to different spatial resolutions (S), where $d_i = \log_2(Q \times S)$. Across datasets and configurations, the ratio $\log(F_i)/d_i$ remains approximately *constant*, providing strong support for the exponential relationship between forgetting and task complexity as captured by intrinsic dimension. This behavior reflects a geometric constraint where increasing intrinsic dimension entangles KANs' localized supports, highlighting the need for dimensionality-aware tuning or support fragmentation (as in Corollary 4) to sustain better retention.

Knowledge Editing for LMs

Experimental Setup LMs require continual knowledge editing to replace outdated information and integrate new facts. To evaluate the forgetting behavior of KANs and MLPs in such high-dimensional editing scenarios, five consecutive tasks are curated from the CounterFact (Meng et al. 2023) and ZsRE (Levy et al. 2017) benchmarks.

Model Configuration LoRA (Hu et al. 2022) is a parameter-efficient fine-tuning technique that adapts LMs by freezing pre-trained weights and training lightweight adapters, substantially reducing memory usage and computational cost compared to full fine-tuning (Biderman et al. 2024). To explore the use of KAN as a LoRA adapter for continual fine-tuning, we design a modified adapter architecture. In the standard LoRA setup, the frozen weight matrix $W_0 \in \mathbb{R}^{a\times b}$ is augmented by a trainable low-rank residual matrix $\Delta W \in \mathbb{R}^{a\times b}$, factorized as $\Delta W = BA$ with $B \in \mathbb{R}^{a\times c}$ and $A \in \mathbb{R}^{c\times b}$, where rank $c \ll \min(a, b)$. The adapter's output is $h = W_0x + BAx$. In our design, both A and B are parameterized using KANs. This KAN-based variant, referred as KAN-LoRA, is integrated into the final two layers of Llama2-7B and Llama2-13B (Touvron et al. 2023). We apply EWC regularization during continual fine-tuning, using the preceding task as memory. For a fair comparison, we develop an MLP-LoRA adapter with identical EWC settings. For all KAN-LoRA experiments, we use a grid size of 5 to balance capacity and efficiency.

Model	Dataset	Samples per Task	KAN LoRA				MLP LoRA			
			# Trained tasks				# Trained tasks			
			2	3	4	5	2	3	4	5
Llama 2-7B	CounterFact	2	100	65	50	45	100	85	57	60
		3	100	93	80	48	100	90	67	57
		4	100	88	78	53	100	90	65	42
		5	100	88	80	57	100	98	77	66
	ZsRE	2	100	80	67	60	100	95	97	87
		3	100	87	71	58	100	100	91	78
		4	100	85	70	55	100	82	63	46
		5	100	76	64	60	100	86	73	57
Llama 2-13B	CounterFact	2	100	75	60	50	100	70	50	43
		3	100	93	71	60	100	93	62	53
		4	100	97	60	44	100	97	77	56
		5	100	76	73	57	100	84	83	63
	ZsRE	2	100	100	83	72	100	80	53	52
		3	100	97	78	75	100	89	62	60
		4	100	75	66	58	100	92	81	55
		5	100	85	80	67	100	83	73	59

(a) KAN-LoRA and MLP-LoRA adapters with rank 8.

Model	Dataset	Samples per Task	KAN LoRA				MLP LoRA			
			# Trained tasks				# Trained tasks			
			2	3	4	5	2	3	4	5
Llama 2-7B	CounterFact	2	100	75	47	45	100	55	30	43
		3	100	77	62	55	100	80	53	45
		4	100	85	57	41	100	75	60	43
		5	100	70	60	49	100	76	59	51
	ZsRE	2	100	90	50	45	100	55	43	37
		3	100	77	58	57	100	70	49	27
		4	100	70	65	48	100	70	58	39
		5	100	74	63	49	100	86	59	39
Llama 2-13B	CounterFact	2	100	65	50	60	100	55	43	40
		3	100	80	69	57	100	77	56	45
		4	100	88	62	39	100	85	50	34
		5	100	64	61	48	100	84	65	45
	ZsRE	2	100	70	53	45	100	55	47	40
		3	100	80	60	57	100	73	47	40
		4	100	70	63	41	100	82	62	48
		5	100	72	59	50	100	76	63	44

(b) KAN-LoRA and MLP-LoRA adapters with rank 16.

Table 4: Mean accuracy (%) on previously edited tasks during continual fine-tuning of Llama 2-7B and Llama 2-13B models equipped with KAN-LoRA and MLP-LoRA adapters. Performance is reported across five consecutive tasks for each dataset.

Model	Adapter	Trainable parameters	Training time (s/epoch)	Inference time (s/sample)
Llama 2-7B	KAN LoRA	2.6M	0.57	0.13
	MLP LoRA	0.28M	0.54	0.12
Llama 2-13B	KAN LoRA	3.2M	1.05	0.23
	MLP LoRA	0.35M	1.01	0.21

Table 5: Comparison of trainable parameters, training, and inference time for KAN-LoRA and MLP-LoRA adapters.

Evaluation Results The modified Llama models equipped with KAN-LoRA and MLP-LoRA adapters are continually fine-tuned across multiple tasks. Tables 4a and 4b report the mean accuracy on previously edited tasks after sequential edits of varying lengths, for adapter ranks 8 and 16 respectively, highlighting the extent of forgetting during the continual fine-tuning process. Increasing the adapter rank leads to greater forgetting in both KAN and MLP variants. However, KAN adapters consistently outperform their MLP counterparts at rank 16 and in low-sample (per task) regimes. Notably, the KAN adapter shows reduced forgetting in Llama2-13B, while the MLP adapter displays the opposite trend. In small-sample settings, KAN achieves consistently higher retention in Llama2-13B compared to MLP. These results suggest that KAN adapters are more resilient to forgetting in large-scale LMs, especially at higher ranks and under limited task supervision.

Table 5 further compares the computational and parameter efficiency of KAN-LoRA and MLP-LoRA adapters, using a grid size of 5 and an adapter rank at 8. KAN introduces significantly more trainable parameters than MLP, approximately $10\times$ more for both Llama2-7B and Llama2-13B models. Training and inference times are measured on the CounterFact dataset with 5 samples per task. While the KAN adapter incurs higher computational cost than the MLP variant, the overhead remains moderate relative to the observed gain in model capacity and retention performance.

Conclusion & Discussion

In this work, we present the first comprehensive study of catastrophic forgetting in KANs under continual learning settings. We develop a theoretical framework that connects forgetting dynamics to the architectural locality of spline activations and the intrinsic dimensionality of task data. Our analysis yields formal retention bounds and characterizes the cumulative and geometry-driven nature of forgetting in KANs. To validate these insights, we conduct systematic experiments across synthetic arithmetic tasks and real-world image classification benchmarks. Empirical results strongly corroborate the our analysis, revealing a clear linear relationship between forgetting and activation overlap, and an exponential increase in forgetting as task dimensionality rises. We further introduce KAN-LoRA, a novel adapter design for continual fine-tuning of LMs in model editing tasks, and demonstrate its retention superiority compared to MLP-based alternatives. Our findings establish both the strengths and limitations of KANs for continual learning.

Stepping further, we believe this work opens up several unconventional directions for advancing KANs in continual learning. First, the evolution of spline activations across tasks suggests a *new dynamic view* of learning, where forgetting reflects adaptation pressures on local function regions. Designing KANs with mechanisms that support controlled specialization or even lifecycle-based pruning of splines may enhance long-term retention. Second, our analysis of support overlap points to the possibility of *distributed memory encoding*. Rather than eliminating interference, future models could intentionally overlap supports to store multiple tasks in a compressed fashion that allows task-specific retrieval through decoding strategies. Third, forgetting itself may also function as a *beneficial inductive bias*. Selective decay in high-dimensional regions could suppress redundant or unstable features, reduce overfitting, and improve generalization. These visions can largely reframe forgetting not as a flaw to be eliminated, but as a property to be shaped, positioning KANs as a flexible and interpretable foundation for memory-aware continual learning systems.

Acknowledgements

The authors thank the anonymous reviewers for comments. This research was supported by the NSF IIS2451480.

References

Abd Elaziz, M.; Ahmed Fares, I.; and Aseeri, A. O. 2024. CKAN: Convolutional Kolmogorov–Arnold Networks Model for Intrusion Detection in IoT Environment. *IEEE Access*, 12: 134837–134851.

Aleixo, E. L.; Colonna, J. G.; Cristo, M.; and Fernandes, E. 2023. Catastrophic Forgetting in Deep Learning: A Comprehensive Taxonomy.

Ayad, O. 2014. Learning under Concept Drift with Support Vector Machines. In *Artificial Neural Networks and Machine Learning – ICANN 2014*, 587–594. Cham: Springer International Publishing.

Biderman, D.; Portes, J.; Ortiz, J. J. G.; Paul, M.; Greengard, P.; et al. 2024. LoRA Learns Less and Forgets Less.

Bishop, C. M. 1994. Neural networks and their applications. *Review of scientific instruments*, 65(6): 1803–1832.

Buzzega, P.; Boschini, M.; Porrello, A.; Abati, D.; and Calderara, S. 2020. Dark Experience for General Continual Learning: a Strong, Simple Baseline.

De Lange, M.; Aljundi, R.; Masana, M.; Parisot, S.; et al. 2022. A Continual Learning Survey: Defying Forgetting in Classification Tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7): 3366–3385.

French, R. M. 1999. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4): 128–135.

Girosi, F.; and Poggio, T. 1989. Representation Properties of Networks: Kolmogorov's Theorem Is Irrelevant. *Neural Computation*, 1(4): 465–469.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2): 3.

Hu, Y.; Liang, Z.; Yang, F.; Hou, Q.; et al. 2025. KAC: Kolmogorov-Arnold Classifier for Continual Learning.

Kemker, R.; McClure, M.; Abitino, A.; et al. 2017. Measuring Catastrophic Forgetting in Neural Networks.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; Hassabis, D.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13): 3521–3526.

Kolmogorov, A. N. 1961. On the Representation of Continuous Functions of Several Variables by Superpositions of Continuous Functions of a Smaller Number of Variables. *American Mathematical Society*.

Kong, Y.; Liu, L.; Chen, H.; Kacprzyk, J.; and Tao, D. 2024. Overcoming Catastrophic Forgetting in Continual Learning by Exploring Eigenvalues of Hessian Matrix. *IEEE Transactions on Neural Networks and Learning Systems*, 35(11).

Lee, A.; Gomes, H. M.; Zhang, Y.; and Kleijn, W. B. 2025. Kolmogorov-Arnold Networks Still Catastrophically Forget but Differently from MLP. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(17): 18053–18061.

Levy, O.; Seo, M.; Choi, E.; and Zettlemoyer, L. 2017. Zero-Shot Relation Extraction via Reading Comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning*, 333–342. Association for Computational Linguistics.

Liu, Z.; Wang, Y.; Vaidya, S.; Ruehle, F.; Halverson, J.; Soljačić, M.; Hou, T. Y.; and Tegmark, M. 2025. KAN: Kolmogorov-Arnold Networks.

Luo, Y.; Yang, Z.; Meng, F.; Li, Y.; Zhou, J.; and Zhang, Y. 2025. An Empirical Study of Catastrophic Forgetting in Large Language Models During Continual Fine-tuning.

McCloskey, M.; and Cohen, N. J. 1989. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. In *Psychology of Learning and Motivation*, volume 24, 109–165. Academic Press.

Meng, K.; Bau, D.; Andonian, A.; and Belinkov, Y. 2023. Locating and Editing Factual Associations in GPT.

Mirzadeh, S. I.; Chaudhry, A.; Yin, D.; et al. 2022. Architecture Matters in Continual Learning.

Prautzsch, H.; Boehm, W.; and Paluszny, M. 2002. *Bézier and B-spline techniques*. Springer Science Business Media.

Riemer, M.; Cases, I.; Ajemian, R.; Liu, M.; Rish, I.; Tu, Y.; and Tesauro, G. 2019. Learning to Learn without Forgetting by Maximizing Transfer and Minimizing Interference.

Ruiz-Garcia, M. 2022. Model architecture can transform catastrophic forgetting into positive transfer.

Spigler, G. 2020. Meta-learnt priors slow down catastrophic forgetting in neural networks.

Touvron, H.; Lavril, T.; Izacard, G.; et al. 2023. LLaMA: Open and Efficient Foundation Language Models.

Ven, G. M. V. D.; Soures, N.; and Kudithipudi, D. 2024. Continual Learning and Catastrophic Forgetting.

Wang, Z.; Yang, E.; Shen, L.; and Huang, H. 2025. A Comprehensive Survey of Forgetting in Deep Learning Beyond Continual Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(3): 1464–1483.

Xu, Y.; Zhong, X.; Yepes, A. J. J.; and Lau, J. H. 2020. Forget Me Not: Reducing Catastrophic Forgetting for Domain Adaptation in Reading Comprehension.

Yang, X.; and Wang, X. 2025. Kolmogorov-Arnold Transformer. In *The Thirteenth International Conference on Learning Representations*.

Yoon, J.; Yang, E.; Lee, J.; and Hwang, S. J. 2018. Lifelong Learning with Dynamically Expandable Networks.

Zenke, F.; Poole, B.; and Ganguli, S. 2017. Continual Learning Through Synaptic Intelligence. *Proceedings of machine learning research*, 70: 3987—3995.

Zhang, T.; Peng, S.; Gao, L.; et al. 2025. Unifying Locality of KANs and Feature Drift Compensation Projection for Data-free Replay based Continual Face Forgery Detection.

Zhang, T.; Wang, X.; Liang, B.; and Yuan, B. 2023. Catastrophic Interference in Reinforcement Learning: A Solution Based on Context Division and Knowledge Distillation. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12): 9925–9939.