Progressively Overcoming Catastrophic Forgetting in Kolmogorov–Arnold Networks

Evgenii Ostanin
Toronto Metropolitan University
Toronto, Canada
email: eostanin@torontomu.ca

Nebojsa Djosic
Toronto Metropolitan University
Toronto, Canada
email: nebojsa.djosic@torontomu.ca

Fatima Hussain
Toronto Metropolitan University
Toronto, Canada
email: fatima.hussain@torontomu.ca

Salah Sharieh
Toronto Metropolitan University
Toronto, Canada
email: salah.sharieh@torontomu.ca

Alexander Ferworn
Toronto Metropolitan University
Toronto, Canada
email: aferworn@torontomu.ca

Malek Sharieh
Holy Trinity School
Richmond Hill, Canada
email: malek.sharieh@hts.on.ca

***Abstract*—Catastrophic forgetting remains a major challenge in continuous learning, particularly for architectures not explicitly designed for knowledge retention. This paper explores Kolmogorov–Arnold networks as an alternative to multilayer perceptrons in such settings. We introduce two freezing strategies: tensor-level spline freezing and point-level control freezing, that exploit the spline-based structure of Kolmogorov–Arnold networks to preserve knowledge from earlier tasks. Experiments on Modified National Institute of Standards and Technology (MNIST) handwritten digit dataset show that both methods yield modest but consistent improvements when paired with replay techniques. The best configurations improve total accuracy by up to 2.2% and reduce forgetting by 5.4% over the no-freeze baseline. These findings reveal a new direction for mitigating forgetting through the selective control of spline parameters specific for the Kolmogorov-Arnold networks. Future work will explore a deeper integration with regularization and expansion methods to further enhance knowledge retention in continual learning.**

Keywords-Continual Learning; Catastrophic Forgetting; Kolmogorov–Arnold Networks; KAN; Spline Freezing; Memory Retention; Experience Replay; Progressive Freezing.

I. Introduction

Continual learning remains a central challenge in modern Machine Learning (ML), particularly in contexts where models must incrementally adapt to new information without catastrophic degradation of previously acquired knowledge [1]. Traditional deep learning models, including Multi-Layer Perceptrons (MLPs), often suffer from *catastrophic forgetting* [2], where performance on earlier tasks deteriorates as new data is introduced. While various techniques such as regularization, dynamic expansion, and rehearsal have been proposed to address this problem [3]–[6], the search for architectures that naturally lend themselves to incremental learning continues.

Kolmogorov–Arnold Networks (KANs), a recent innovation based on the Kolmogorov–Arnold representation theorem [7], have been proposed as interpretable and adaptable neural networks that may address some limitations of fixed-activation architectures. In KANs, traditional scalar weights are replaced by univariate, learnable activation functions (typically splines), enabling fine-grained, input-dependent transformations. Each spline activation has its own parameter set, so during sequential training, only the splines relevant to a new task are updated while the others remain fixed, thus naturally preserving previously acquired knowledge.

In this paper, we evaluate the suitability of KAN for continual learning by comparing their retention capabilities to MLPs under task-incremental training scenarios. Specifically, we adopt the *Split-MNIST* protocol [8] which partitions the MNIST [9] (Modified National Institute of Standards and Technology) handwritten-digit dataset into two sequential training tasks on digits 0–4 and 5–9.

Building on our previous analysis of KANs under adversarial threats [11], [12], this study extends the evaluation to continual learning scenarios, introducing a broader set of robustness indicators. We compare results across architectures and freezing strategies, focusing on metrics of **accuracy**, **retention**, and **forgetting**. Notably, we observe that freezing improves knowledge retention in settings with conventional replay but does not provide consistent benefits when replay is class-balanced. These findings highlight the nuanced interactions between architecture, training dynamics, and memory retention, opening new directions for lifelong learning research.

Main Contributions:

- A comprehensive comparison of KANs and MLP in continual learning using the Split-MNIST benchmark.
- Systematic testing of replay and balanced replay buffer strategies for mitigating forgetting in both model types, using such methods as experience replay (random sampling) and stratified (class-balanced) replay respectively.
- Introduction and evaluation of two novel KAN-specific freezing techniques, targeting spline control points and entire spline tensors.
- Empirical findings showing that KANs benefit from freezing strategies primarily when used in conjunction with naive replay mechanisms.

The remainder of this paper is organized as follows: Section II reviews related work on continual learning and memory retention in neural networks. Section III details the experimental design, including dataset splits, architecture configurations, and freezing protocols. Section IV presents the results of our evaluations, with a comparative analysis of accuracy and forgetting. Section V discusses conclusions and future work directions.

II. Related Work

A. Continual and Lifelong Learning

Continual learning, also referred to as lifelong or incremental learning [1], focuses on enabling models to learn from a stream of tasks without suffering from catastrophic forgetting. This challenge arises when models trained on new data overwrite previously learned information, leading to severe performance drops on older tasks [2], [3]. To systematically evaluate continual learning capabilities, benchmarks such as Split-MNIST [8], [9], [13] are widely adopted. These benchmarks divide a dataset into separate subsets of classes forcing the model to incrementally adapt without access to previous task data during training.

While much of the foundational work in this area has focused on regularization-based methods (e.g., Elastic Weight Consolidation [14]) and architectural adaptation (e.g, dynamically growing networks [15]), rehearsal-based strategies have recently gained prominence for their effectiveness and simplicity. However, many of these techniques are designed around conventional neural architectures, such as MLPs, and their applicability to novel, more interpretable models remains largely unexplored. This paper contributes to bridging this gap by evaluating continual learning in KANs.

B. Replay and Balanced Replay

Replay mechanism attempt to alleviate forgetting by introducing a rehearsal buffer that store and replays a subset of previously encountered samples. The simplest approach, *experience replay*, samples examples uniformly at random from a memory buffer [16], while *balanced replay* aims to ensure class-wise uniformity, especially critical when data from previous tasks are imbalanced or limited [6], [10]. These methods are often paired with online learning or streaming data scenarios, where maintaining compact yet representative memory is crucial.

Despite their widespread adoption in conventional deep learning, replay-based techniques have not been systematically evaluated in emerging network paradigms such as KANs. Given KANs' fundamentally different parameterization, where edge functions are learnable instead of weights alone, it is not immediately clear, whether replay would behave similarly. Our study investigates this open question and quantifies the impact of replay versus balanced replay in KAN training regimes.

C. Kolmogorov-Arnold Networks and Interpretability

KANs [7] represent a significant shift in neural network architecture. Originally proposed by Andrey Kolmogorov in 1957 and later extended by Vladimir Arnold in 1963, the Kolmogorov–Arnold Representation Theorem, also known as the superposition theorem, states that any continuous multivariate function $f(x_1, ..., x_n)$ defined on a bounded domain can be expressed as a finite composition of continuous univariate functions, typically formulated as:

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

In (1) $\phi_{q,p} : [0, 1] \to \mathbb{R}$ are continuous inner functions, and $\Phi_q : \mathbb{R} \to \mathbb{R}$ represent continuous outer functions.

Inspired by the Kolmogorov–Arnold representation theorem, KANs replace scalar edge weights with univariate, learnable spline functions. In KANs each connection from neuron p to q now applies its own spline function $\phi_{q,p}(x_p)$ to the incoming activation x_p, rather than simply multiplying by a constant weight as in MLPs. This change enables each connection to perform a data-driven, nonlinear transformation, offering both functional richness and a degree of interpretability rarely found in traditional models. Rather than stacking fixed nonlinearities at the nodes as in MLPs, KANs achieve expressivity through these adaptable spline-based edge functions.

Although KANs are relatively new, their potential has already sparked interest across many domains. Prior studies have explored their application to time series [17], [18], robustness under adversarial attacks [11], [12], [19], [20], and noise resilience [21], [22]. For instance, [7] demonstrated that KANs can approximate complex mappings with far fewer parameters while retaining interpretability via their control-point structures. However, continual learning in KANs remains underexplored. No prior work has directly evaluated their memory retention across tasks or how spline parameterization interacts with long-term adaptation.

Our study positions itself as one of the first to investigate KANs in a continual learning context, motivated by their spline-based design, which naturally partitions the model into independent, learnable components, and by the opportunity this provides for targeted freezing mechanisms. Figures 1 and 2 highlight the architectural distinction between KAN and MLP, which motivates the design of two freezing strategies: control point-level freezing and tensor-level spline freezing. These mechanisms exploit the hierarchical structure of spline parameters and enable selective locking of the model's knowledge, a novel direction for lifelong learning research.

D. Freezing Mechanisms in Incremental Learning

Weight freezing has been used historically to preserve important parameters while learning new tasks. Techniques like Learning without Forgetting (LwF) [23] and PackNet [24] selectively retain task-specific neurons or weights to reduce interference. More recently, methods such as Progressive Neural Networks [25] have explored architectural partitioning where frozen components are reused or extended.

In the context of KANs, we introduce two distinct types of freezing. First, control **point-level freezing** targets high-importance spline parameters based on magnitude or gradient scores. Second, **tensor-level spline freezing** locks entire univariate transformation functions. These techniques take advantage of the KAN flexible design, where splines are modular and easy to adjust in a detailed way. Our experiments demonstrated that spline-level freezing in KANs offers a new dimension of control not available in traditional ML models, and its interaction with replay dynamics presents novel trade-offs between plasticity and stability.

III. METHODOLOGY

A. Architectures

To compare robustness under continual learning, we consider two types of models: a standard MLP and a KAN. The MLP serves as a baseline, while the KAN explores the performance of spline-based representations in a task-incremental settings.

The MLP consists of two linear layers with a Rectified Linear Unit (ReLU) activation in between. The first linear layer maps a flattened 28×28 MNIST image (784-dimensional input) to a 128-dimensional hidden layer. The second (output) layer produces a 10-dimensional output corresponding to the MNIST class logits. Figure 1 demonstrates the MLP architecture used in the experiments.

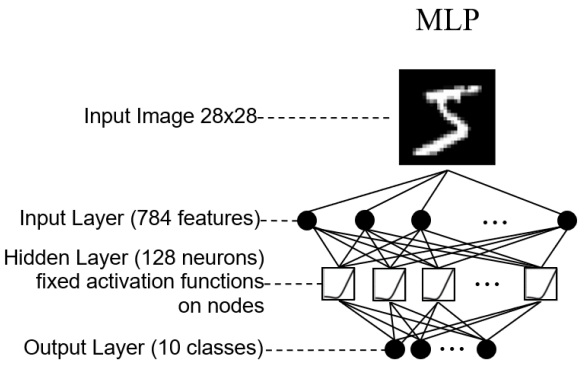

Figure 1. MLP Architecture.

The KAN has a similar structure, with a linear layer projecting inputs to a 128-dimensional hidden space, followed by another linear layer to predict class logits. However, between the input and hidden layers, KAN includes a matrix of learnable spline control weights (128×784), which are not used for direct computation but are integrated into the computation graph. These spline weights can be interpreted as representing local functional transformations and can be subjected to regularization or freezing. KAN architecture is shown in Figure 2.

B. Continual Learning Setup

To simulate continual learning under the Split-MNIST protocol [8], [9], we split the MNIST dataset into two tasks: Task A, containing digits 0 through 4, and Task B, containing digits 5 through 9. The model is first trained on Task A for three epochs, then trained on Task B for three more epochs. Catastrophic forgetting is quantified as the drop in Task A accuracy after Task B training. All experiments use the AdamW optimizer with learning rate 1×10^{-3}, cross-entropy loss, and batch size of 64 images per mini-batch.

C. Replay and Balanced Replay

To mitigate forgetting, we implement two forms of experience replay. In both, we store a subset of Task A examples and mix them into the mini-batches during Task B training.

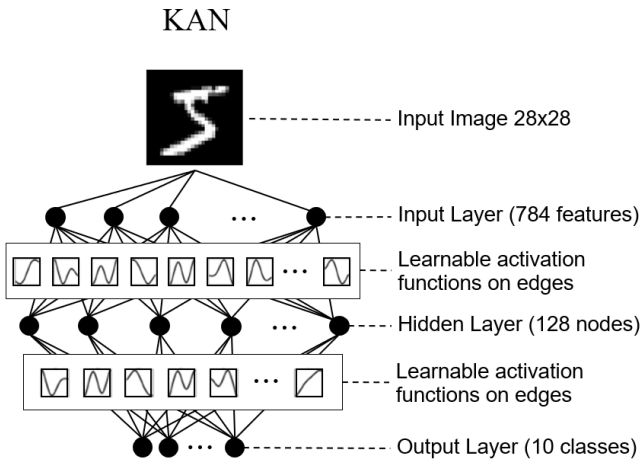

Figure 2. KAN Architecture.

In the first variant, **replay**, we randomly sample a buffer of Task A examples. In the second, **balanced replay**, we sample the replay buffer in a stratified fashion to ensure class balance across the five Task A classes. To avoid confusion with Task B and for brevity, we will refer to balanced replay as stratified replay (s-replay) throughout the paper.

We tested replay buffer sizes of 50, 100, and 500, where the buffer size denotes the number of data samples retained for the next training round. We chose buffer sizes to represent low, medium, and high replay capacities, so we could observe how freezing performs under different conditions. As expected, larger buffers led to better retention. Replay with 50 examples provided moderate improvements, while 500 nearly eliminated forgetting. However, our objective is to explore whether spline freezing can further improve retention. Thus, we selected buffer size 100 for all subsequent experiments. This setting provides a middle ground. It significantly improves performance over the baseline, but leaves room for further gains. Using 50 samples could underestimate the impact of freezing, while 500 would saturate the model's retention capacity, potentially masking the effects of freezing mechanisms.

D. Spline Freezing Strategies

A unique feature of KANs is the presence of interpretable and modular spline parameters. Building on parameter-isolation and architectural-partitioning approaches [24], [25], we evaluate two types of freezing techniques to investigate their effect on continual learning:

(1) Tensor-level (entire spline) freezing: In this strategy, we compute a score for each of the 128 spline rows (or neurons) and freeze the top $k\%$ rows. Three scoring methods are evaluated using such approaches as :

- **weight**: mean absolute value of the weights in each row
- **grad**: mean absolute gradient magnitude per row (requires a gradient pass)
- **weight_grad**: a combination of both (with $\alpha = 0.5$)

(2) Point-level (individual control point) freezing: Here, the same scoring methods are applied to individual elements (control points) in the spline weight matrix. The top $k\%$ of all elements are then frozen, regardless of their row or neuron association.

For both strategies, we test $k \in \{0.05, 0.1, 0.25, 0.5, 0.75\}$, spanning from minimal to aggressive freezing intensities. This range lets us assess how varying degrees of parameter k affect retention under both replay=100 and s_replay=100, yielding 30 experiments per technique. These strategies are visualized in Figure 3. In each case, frozen parameters are excluded from optimization updates by masking their gradients before applying the optimizer step.

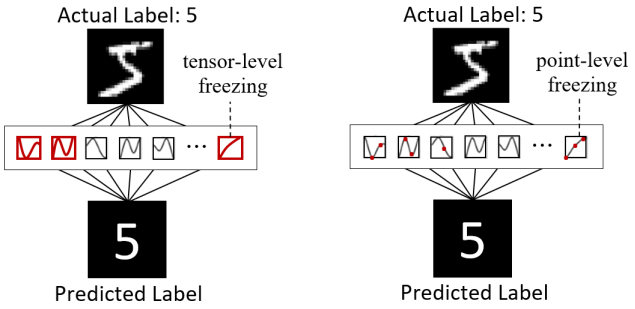

Figure 3. Tensor-level (left) and Point-level (right) freezing strategy.

E. Experimental Pipeline

Each experiment proceeds as follows. The model is initialized and trained on Task A. After evaluating and recording the initial accuracy, the freezing mechanism (if any) is applied using a single gradient pass (when necessary). The model is then trained on Task B, incorporating replayed samples into each batch, as specified. After training, we compute and report the accuracy on Task B (new task), accuracy on Task A (after forgetting), and total accuracy across both tasks. We also compute forgetting as the drop in Task A accuracy before and after Task B training.

The following section presents the experimental results. We first validate the replay strategies across different buffer size and architectures, then evaluate the impact of spline freezing techniques. The aim is to determine whether freezing entire spline or individual components can improve retention in continual learning settings, and whether the choice of scoring strategy or replay method impacts this effect.

IV. Results

A. Baseline Performance and Forgetting

Figure 4 and Table I summarize the performance of MLP and KAN under different training scenarios. The clean setting refers to training on all MNIST classes simultaneously, serving as an upper-bound reference. Both models achieve high accuracy on the full MNIST task (88.8% and 88.6%, respectively), but suffer from severe catastrophic forgetting when trained sequentially on separated tasks. The baseline reflects continual training without any mitigation, revealing the severity of catastrophic forgetting and establishing a comparison point for subsequent interventions. Figure 4 also shows the improvements achieved through replay and s-replay (stratified replay) before any spline or tensor freezing techniques are applied. Table I additionally reports the forgetting metric, which quantifies the reduction in accuracy on Task A after training on Task B. The reported accuracy corresponds to the model's total accuracy after both training phases. For example, in the baseline scenario, Task A accuracy drops by nearly 96%, resulting in an overall accuracy of just 43.4% for MLP and 43.3% for KAN, underscoring the impact of forgetting in continual learning.

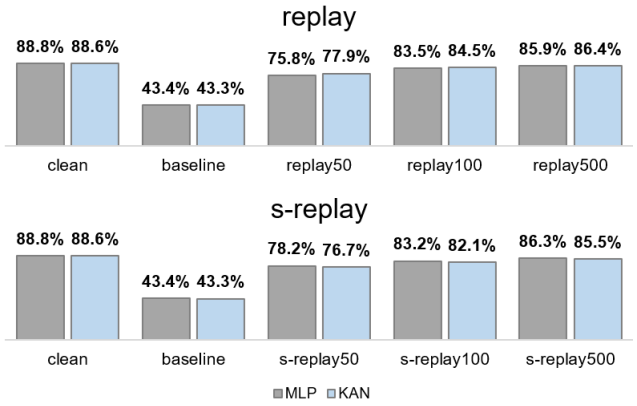

Figure 4. Baseline accuracy and replay effectiveness for MLP and KAN.

TABLE I
BASELINE ACCURACY AND FORGETTING FOR MLP AND KAN.

Scenario	MLP Acc.	KAN Acc.	MLP Forget.	KAN Forget.
Clean	0.888	0.886	-	-
Baseline	0.434	0.433	0.958	0.964
Replay 50	0.758	0.779	0.322	0.275
Replay 100	0.835	0.845	0.149	0.137
Replay 500	0.859	0.864	0.075	0.071
s-Replay 50	0.782	0.767	0.261	0.305
s-Replay 100	0.832	0.821	0.147	0.187
s-Replay 500	0.863	0.855	0.068	0.077

B. Replay and Stratified (Balanced) Replay

We evaluated replay-based strategies with varying buffer sizes. Standard replay (random) and s-replay both improve accuracy and retention, as seen in Figure 4. With replay buffer size 100, MLP and KAN reach 83.5% and 84.5% accuracy, respectively, while s-replay achieves 83.2% for MLP and 82.1% for KAN.

These configurations reduce forgetting substantially, as seen in Table I. The choice of buffer size 100 offers a middle ground between replay_50, which yielded lower gains, and replay_500, which almost eliminated forgetting. We selected 100 for subsequent experiments, as it maintained measurable room for improvement while ensuring sufficient retention to validate the impact of freezing methods.

C. Point-Level Freezing

Point-Freezing (pf) methods, shown in Figure 5, Table II, and Table III, freeze the top-$k\%$ of control points using heuristics based on weights (w), gradients (g), or a weighted average (wg). For s-replay, the best configuration is `pf_g_s-replay100` at $k = 25\%$, which achieved 84.3% accuracy and reduced forgetting to 0.133, outperforming the no-freeze baseline of 82.1% (+2.2%) accuracy and 18.7% (-5.4%) forgetting. In the replay setup, the best pf result was `pf_wg_replay100` at $k = 25\%$, with 84.2% accuracy and 0.133 forgetting.

Across all experiments, s-replay consistently outperformed standard replay in both baseline accuracy and forgetting, even before freezing was applied. Moreover, pf under s-replay remained effective across multiple k values, with most configurations improving over the no-freeze baseline.

These results suggest that s-replay provides a stronger foundation for knowledge retention, likely due to its class-balanced sampling, which ensures more uniform coverage of prior task classes during rehearsal. When combined with pf, this structure appears to help selectively consolidate important spline parameters, leading to synergistic gains in both accuracy and forgetting. The consistent effectiveness of point-level freezing under s-replay highlights its value as a complementary mechanism for continual learning with KANs. Despite these gains, the best s-replay + pf configuration still incurs a 13.3% forgetting rate, underscoring the need to explore additional forgetting mitigation strategies in future work.

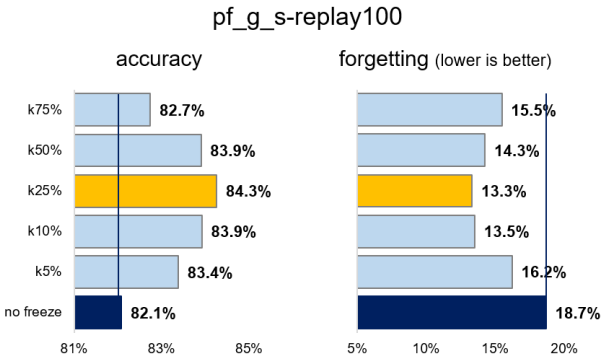

Figure 5. Best KAN scenario with Point-Level Freezing (pf).

D. Tensor-Level Freezing

Tensor-Level Freezing (tf), shown in Figure 6, Table II, and Table III, disables entire spline rows and can yield strong improvements, though it introduces more variance compared to point-level freezing. The best replay configuration is `tf_wg_replay100` at $k = 75\%$, which achieved 85.2% accuracy (+0.7%) and reduced forgetting to 0.101 (-3.6%). For s-replay, the best result is `tf_g_s-replay100` at $k = 75\%$, with 84.3% accuracy and 0.127 forgetting.

Although some configurations (e.g., $k = 10\%$ for `tf_w_replay100`) resulted in noticeable performance drops, the top-performing setups confirm that tensor-freezing can outperform point-freezing in certain cases when appropriately tuned. These gains are most evident at higher freezing thresholds ($k = 50\%$–75%), suggesting that the disabling of larger sets of spline transformations can help stabilize representations after task shifts, particularly when combined with structured replay. However, the broader range of outcomes highlights that tensor freezing is more sensitive to the choice of k and scoring strategy, reinforcing the need for careful calibration.

TABLE II
KAN ACCURACY UNDER REPLAY AND S-REPLAY FOR TENSOR (TF) AND POINT (PF) FREEZING.

Method	no freeze	k5%	k10%	k25%	k50%	k75%
pf_w_replay100	0.845	0.817	0.841	0.830	0.837	0.816
pf_g_replay100	0.845	0.833	0.835	0.822	0.845	0.824
pf_wg_replay100	0.845	0.838	0.832	0.842	0.824	0.844
pf_w_s-replay100	0.821	0.816	0.828	0.831	0.833	0.829
pf_g_s-replay100	0.821	0.834	0.839	0.843	0.839	0.827
pf_wg_s-replay100	0.821	0.825	0.838	0.840	0.838	0.829
tf_w_replay100	0.845	0.851	0.813	0.834	0.844	0.821
tf_g_replay100	0.845	0.840	0.846	0.830	0.834	0.843
tf_wg_replay100	0.845	0.818	0.834	0.833	0.832	0.852
tf_w_s-replay100	0.821	0.837	0.826	0.826	0.835	0.829
tf_g_s-replay100	0.821	0.831	0.832	0.834	0.842	0.843
tf_wg_s-replay100	0.821	0.851	0.841	0.841	0.841	0.837

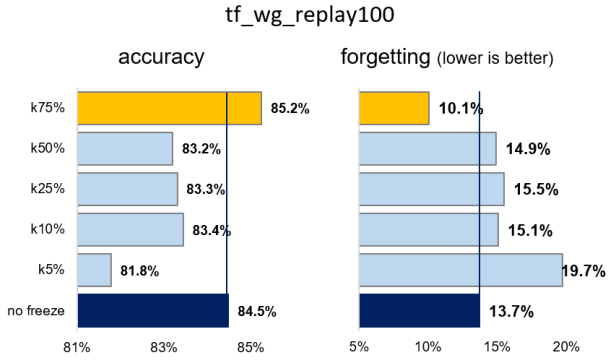

Figure 6. Best KAN scenario with tensor-level freezing (tf).

TABLE III
KAN FORGETTING UNDER REPLAY AND S-REPLAY FOR TENSOR (TF) AND POINT (PF) FREEZING.

Method	no freeze	k5%	k10%	k25%	k50%	k75%
pf_w_replay100	0.137	0.185	0.133	0.154	0.137	0.178
pf_g_replay100	0.137	0.166	0.153	0.178	0.123	0.157
pf_wg_replay100	0.137	0.141	0.144	0.133	0.166	0.112
pf_w_s-replay100	0.187	0.182	0.166	0.160	0.155	0.157
pf_g_s-replay100	0.187	0.162	0.135	0.133	0.143	0.155
pf_wg_s-replay100	0.187	0.173	0.135	0.148	0.130	0.152
tf_w_replay100	0.137	0.116	0.200	0.153	0.133	0.163
tf_g_replay100	0.137	0.121	0.133	0.165	0.158	0.116
tf_wg_replay100	0.137	0.197	0.151	0.155	0.149	0.101
tf_w_s-replay100	0.187	0.141	0.168	0.174	0.130	0.152
tf_g_s-replay100	0.187	0.143	0.164	0.152	0.137	0.127
tf_wg_s-replay100	0.187	0.123	0.134	0.132	0.137	0.136

E. Freezing Strategies: Comparative Effectiveness

Our evaluation of spline freezing strategies shows that both tf and pf methods improve continual learning when paired with replay mechanisms. While both enhance accuracy and retention, their effectiveness depends on the configuration.

pf offers consistent gains, especially under s-replay. Most k values outperform the no-freeze baseline, with the best configuration (`pf_g_s-replay100` at $k = 25\%$) improving accuracy by +2.2% and reducing forgetting by 5.4%. This suggests that fine-grained control over spline weights helps preserve prior task knowledge without impairing new learning.

tf, which locks full spline rows, shows greater variability but also higher potential. The best configuration (`tf_wg_replay100` at $k = 75\%$) yielded the top accuracy overall (+1.0% vs no-freeze) and reduced forgetting by 3.6%. However, tf performance is more sensitive to k and the scoring strategy, and can degrade if freezing is too aggressive.

Figures 5 and 6 summarize the top-performing pf and tf setups. While pf freezing is more robust across scenarios, tf freezing offers a higher ceiling when properly tuned. These complementary traits highlight the adaptability of KANs for continual learning applications.

V. Conclusion and Future Work

This paper investigated KANs in continual learning, demonstrating that both tensor-level and point-level spline freezing consistently improve retention in Split-MNIST when paired with simple replay (up to +2.2 % overall accuracy and a 5.4 % reduction in forgetting). While the absolute improvements are moderate, these KAN-specific freezing strategies leverage the spline structure to preserve prior task knowledge without impeding new learning, opening a promising direction for more targeted retention strategies.

Future work will explore freezing in deeper KANs, integration with regularization and dynamic expansion methods, and testing on more complex benchmarks beyond MNIST. Additionally, we aim to develop adaptive freezing and unfreezing strategies, drawing inspiration from biological learning and synaptic plasticity. With these enhancements, we expect to achieve higher retention and greater robustness in continual learning tasks, further unlocking the potential of KANs for long-term knowledge consolidation.

Acknowledgment

We acknowledge the use of various general-purpose online and cloud-based tools, including those with AI-driven features, during the preparation of this work.

References

[1] B. Liu, “Lifelong machine learning: A paradigm for continuous learning”, Frontiers of Computer Science, vol. 11, no. 3, pp. 359–361, 2017.

[2] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, “Continual lifelong learning with neural networks: A review”, Neural Networks, vol. 113, pp. 54–71, 2019, [Online]. Available: https://doi.org/10.1016/j.neunet.2019.01.012. Accessed: 14 May 2025.

[3] M. Delange et al., “A continual learning survey: Defying forgetting in classification tasks”, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1–1, 2021.

[4] J. Zhang, Y. Fu, Z. Peng, D. Yao, and K. He, ”CORE: Mitigating catastrophic forgetting in continual learning through cognitive replay”, 2024. [Online]. Available: https://arxiv.org/abs/2402.01348. Accessed: 14 May 2025.

[5] A. Krawczyk and A. Gepperth, “An analysis of best-practice strategies for replay and rehearsal in continual learning”, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 4196–4204.

[6] L. Liu, L. Liu, and Y. Cui, ”Prior-free balanced replay: Uncertainty-guided reservoir sampling for long-tailed continual learning”, 2024. [Online]. Available: https://arxiv.org/abs/2408.14976. Accessed: 14 May 2025.

[7] Z. Liu et al., “KAN: Kolmogorov-Arnold networks”, Apr. 2024, [Online]. Available: http://arxiv.org/abs/2404.19756. Accessed: 14 May 2025.

[8] F. Zenke, B. Poole, and S. Ganguli, ”Continual learning through synaptic intelligence”, 2017. [Online]. Available: https://arxiv.org/abs/1703.04200. Accessed: 14 May 2025.

[9] L. Deng, “The MNIST database of handwritten digit images for machine learning research”, IEEE Signal Processing Magazine, vol. 29, pp. 141–142, Jun. 2012.

[10] A. Chaudhry et al., ”On tiny episodic memories in continual learning”, 2019. [Online]. Available: https://arxiv.org/abs/1902.10486. Accessed: 14 May 2025.

[11] E. Ostanin, N. Djosic, F. Hussain, S. Sharieh, and A. Ferworn, “Evaluating the robustness of Kolmogorov-Arnold networks against noise and adversarial attacks”, in Proceedings of the SECURWARE 2024, The Eighteenth International Conference on Emerging Security Information, Systems and Technologies, Nov. 2024, pp. 11–16.

[12] N. Djosic, E. Ostanin, F. Hussain, S. Sharieh, and A. Ferworn, “KAN vs KAN: Examining Kolmogorov-Arnold networks (KAN) performance under adversarial attacks”, in Proceedings of the SECURWARE 2024, The Eighteenth International Conference on Emerging Security Information, Systems and Technologies, Nov. 2024, pp. 17–22.

[13] M. Farajtabar, N. Azizan, A. Mott, and A. Li, ”Orthogonal gradient descent for continual learning”, 2019. [Online]. Available: https://arxiv.org/abs/1910.07104. Accessed: 14 May 2025.

[14] J. Kirkpatrick et al., “Overcoming catastrophic forgetting in neural networks”, Proceedings of the National Academy of Sciences, vol. 114, no. 13, pp. 3521–3526, Mar. 2017. [Online]. Available: http://dx.doi.org/10.1073/pnas.1611835114. Accessed: 14 May 2025.

[15] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, ”Lifelong learning with dynamically expandable networks”, 2018. [Online]. Available: https://arxiv.org/abs/1708.01547. Accessed: 14 May 2025.

[16] D. Rolnick, A. Ahuja, J. Schwarz, T. P. Lillicrap, and G. Wayne, ”Experience replay for continual learning”, 2019. [Online]. Available: https://arxiv.org/abs/1811.11682. Accessed: 14 May 2025.

[17] K. Xu, L. Chen, and S. Wang, “Kolmogorov-Arnold networks for time series: Bridging predictive power and interpretability”, Jun. 2024, [Online]. Available: http://arxiv.org/abs/2406.02496. Accessed: 14 May 2025.

[18] C. J. Vaca-Rubio, L. Blanco, R. Pereira, and M. Caus, “Kolmogorov-Arnold networks (KANs) for time series analysis”, May 2024, [Online]. Available: http://arxiv.org/abs/2405.08790. Accessed: 14 May 2025.

[19] A. D. M. Ibrahum, Z. Shang, and J.- E. Hong, “How resilient are Kolmogorov–Arnold networks in classification tasks? A robustness investigation”, Applied Sciences, vol. 14, no. 22, 2024

[20] T. Alter, R. Lapid, and M. Sipper, “On the robustness of Kolmogorov-Arnold networks: An adversarial perspective”, 2024. [Online]. Available: https://arxiv.org/abs/2408.13809. Accessed: 14 May 2025.

[21] C. Zeng, J. Wang, H. Shen, and Q. Wang, “KAN versus MLP on irregular or noisy functions”, 2024, [Online]. Available: https://arxiv.org/abs/2408.07906. Accessed: 14 May 2025.

[22] H. Shen, C. Zeng, J. Wang, and Q. Wang, “Reduced effective-ness of Kolmogorov-Arnold networks on functions with noise”, Jul. 2024, [Online]. Available: http://arxiv.org/abs/2407.14882. Accessed: 14 May 2025.

[23] Z. Li and D. Hoiem, ”Learning without forgetting”, 2017. [Online]. Available: https://arxiv.org/abs/1606.09282. Accessed: 14 May 2025.

[24] A. Mallya, D. Davis, and S. Lazebnik, ”Piggyback: Adapting a single network to multiple tasks by learning to mask weights”, 2018. [Online]. Available: https://arxiv.org/abs/1801.06519. Accessed: 14 May 2025.

[25] A. A. Rusu et al., ”Progressive neural networks”, 2022. [Online]. Available: https://arxiv.org/abs/1606.04671. Accessed: 14 May 2025.