KAN Control Framework for Continual Learning

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

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

***Abstract*—This study extends the concept of progressively reducing catastrophic forgetting in Kolmogorov-Arnold Networks (KAN) by introducing the KAN Control Framework (KCF). In KCF, the stability-plasticity trade-off is described through a control vector that separates memory-related controls (replay policy, buffer size M, replay intensity ρ) from plasticity-related controls (freezing granularity, freeze ratio k, and importance scoring). Using a two-task Split-MNIST protocol with multi-seed evaluation, replay-only behavior is first mapped over (M, ρ) for standard and balanced replay, where retention improvements are largely driven by rehearsal strength. Progressive spline freezing is then studied as an additional control, and the largest gains are observed in constrained replay scenarios, where memory or replay throughput may be limited, while sufficiently strong replay-only settings can still achieve the lowest forgetting overall. Finally, the way in which the freezing granularity (point-level vs. tensor-level) and scoring rules shape the retention-adaptation trade-off are analyzed and practical guidance is provided for selecting KCF settings under common resource limits. Overall, KCF is intended to make CF configurations in KANs more explicit and reproducible.**

Keywords-Continual Learning; Catastrophic Forgetting; Kolmogorov-Arnold Networks; KAN Control Framework; Spline Freezing; Memory Retention.

I. Introduction

This paper extends a previous publication dedicated to progressive spline freezing for reducing Catastrophic Forgetting (CF) in Kolmogorov-Arnold Networks (KAN) [1]. Continual Learning (CL) is commonly framed as learning from a stream of tasks or shifting data distributions without repeatedly retraining from scratch [2], [3]. In many practical settings, CF can be observed, where after optimization and training on new data, performance in earlier tasks can drop substantially [4]–[6]. This effect is often more problematic when shift is expected (e.g., evolving cyber threats, changing user behavior, or non-stationary sensor streams) and when storage, retraining cadence, or latency constraints make full replay or repeated re-optimization difficult.

A broad family of CL approaches has been proposed to soften interference by limiting updates or reintroducing past information during training. Regularization-based approaches (e.g., Elastic Weight Consolidation and related synaptic-importance methods) penalize changes to parameters estimated to be important for earlier tasks [4], [5]. Gradient-projection and orthogonalization methods restrict the update direction so that interference between tasks is reduced [7]. Architectural strategies add capacity or isolate task-specific subnetworks so that earlier mappings are preserved more directly [8], [9]. Another group of replay-based approaches retain a small buffer of past samples and re-use them with new data so that representations may remain preserved over time [10], [11]. Replay is frequently effective, although its impact tends to depend on the memory budget and on how the buffer is sampled; balanced and uncertainty-guided variants have recently been explored to improve stability when buffers are small [12].

At the same time, KANs have emerged as a structurally distinct alternative to Multilayer Perceptrons (MLPs), replacing fixed node activations with learnable univariate functions placed on edges, commonly parameterized as splines [13]. This shift in the location of the capacity has been associated with favorable trade-offs between accuracy and interpretability in some tasks [13]. At the same time, it also changes where and how plasticity is expressed during sequential training.

Recent studies have started to examine the behavior of KANs under noise and adversarial perturbations, where both robustness opportunities and sensitivity patterns have been reported [14]–[18]. Compared to robustness analysis, however, CL for KANs is still less explored, particularly when reliable retention is needed under limited memory and limited replay throughput. Our prior work introduced progressive freezing strategies for KANs in a CL setting, where subsets of spline parameters are selectively frozen based on importance scores and can be combined with replay. It was observed that freezing can reduce forgetting, but the interaction with replay is not always straightforward: when replay is abundant, freezing may provide only marginal benefit, whereas under constrained replay budgets it can offer more noticeable improvements [1]. These observations motivated a broader framing in which freezing is treated as one of several knobs that may be used to manage stability-plasticity trade-offs.

This extended paper develops that framing into the KAN Control Framework (KCF) for controlled adaptation. The central idea is to represent CL in KANs as a controlled update

process governed by a small set of explicit, configurable controls such as replay policy and budget, freezing granularity, freezing fraction, and importance scoring (among others). By making these controls explicit, KCF is intended to support systematic exploration of trade-offs, repeatable comparisons under matched constraints, and decision guidance for selecting configurations when memory and computation are limited. A controlled Split-MNIST benchmark [5] is used as a testbed, where the MNIST [19] (Modified National Institute of Standards and Technology) handwritten-digit dataset is partitioned into two sequential tasks comprising digits 0-4 and 5-9. The multi-seed evaluation is reported so that not only mean behavior but also variability can be inspected. This matters because small differences in configuration may lead to noticeably different retention outcomes, especially in the low-budget regimes that often motivate CL in the first place.

As compared to the original paper [1], this extended paper makes the following contributions:

- KCF: CL in KANs is formalized as a controlled adaptation problem, and a compact control vector is specified so that replay and freezing choices can be expressed in a unified way.
- Reliability under constraints: multi-seed, matched-budget experiments are provided to characterize when progressive freezing is likely to help, and when replay alone may be sufficient, so that freezing can be interpreted as a constraint-aware control rather than a universal improvement.
- Granularity-aware analysis: freezing is compared at different granularities (point-level vs. curve/tensor-level) and across importance scoring choices, highlighting stability-plasticity patterns that appear to be specific to spline-parameterized KAN layers.
- Actionable selection guidance: the empirical findings are translated into practical recommendations for choosing KCF settings when memory and replay throughput are constrained.

This paper is organized in the following way: Section II reviews CL mechanisms and KAN fundamentals outlined in the related work. Section III presents the KCF and the controlled adaptation formulation. Section IV describes datasets, models, and evaluation protocol. Section V reports results and analyses, and Sections VI and VII discuss implications, limitations, and directions for controlled adaptation in non-stationary settings, and provide conclusions and future work directions. With this background in place, related work is reviewed next to position KCF within the broader CL landscape.

II. Related Work

A. CL and CF

CL studies how models may be trained on a sequence of tasks or data distributions while previously learned capabilities are preserved [2], [3], [20]. A recurring difficulty is CF, where optimization on new data introduces interference that degrades performance in earlier tasks [6]. The literature is often summarized into (i) regularization and importance-based constraints, (ii) gradient-constraint or projection methods, (iii) architectural and parameter-isolation strategies, and (iv) replay-based stabilization, with combinations being common [3], [20].

Regularization-based methods reduce forgetting by discouraging changes in parameters that appear to matter the most for the prior tasks. Elastic Weight Consolidation approximates the importance of the parameters via a Fisher-information estimate and penalizes updates along directions that are treated as critical [4]. Related approaches such as Synaptic Intelligence track the contributions of parameters during training and then protect weights that seem to carry prior knowledge [5]. In addition to regularization, gradient-projection techniques aim to prevent destructive interference by constraining the update direction. For example, this can be achieved by enforcing the orthogonality of gradients with respect to stored task-relevant directions [7].

Architectural and parameter-isolation approaches preserve prior knowledge by allocating or masking parameters for new tasks. Progressive networks add new capacity per task, while earlier columns remain frozen [8]. Other methods learn task-conditioned masks over a shared network, which can allow reuse while interference is limited [9], [21]. Across these directions, a common theme is the value of selective update control. Restricting plasticity in the right place may result in a more reliable knowledge retention.

B. Replay and memory-constrained rehearsal

Replay-based methods mitigate forgetting by reusing examples (or features) from previous tasks while training on the current task [10], [11]. Because storage is often limited in practical settings, substantial attention has been paid to how replay effectiveness scales with the memory budget and to sampling strategies that may improve stability under tight constraints [10], [11]. Recent work also explores replay variants that balance representativeness and uncertainty. For example, this is demonstrated through balanced replay combined with uncertainty-guided selection [12]. These results also hint that replay may be more naturally viewed as a configurable control (budget plus policy) than as a single, fixed technique.

C. KANs and stability under distributional stress

KANs introduce a structural alternative to MLPs by placing learnable univariate nonlinearities on edges and aggregating them at nodes, commonly implemented as spline-parameterized functions [13]. Since representational capacity is shifted toward structured function parameters, it is plausible that interference during sequential adaptation will present somewhat differently than in standard MLPs. In parallel with rapid adoption, several studies have examined KAN stability under perturbation-based stressors, including adversarial attacks and noise, to characterize robustness patterns and implementation-dependent effects [14]–[18]. While these studies focus on input perturbations and threat models, they also suggest that KAN behavior can vary meaningfully with

architectural and implementation choices, which strengthens the case for explicit adaptation controls in non-stationary settings.

D. Controlled adaptation and selective plasticity for KANs

Despite growing interest in KAN robustness, CL behavior for KANs remains comparatively underexplored. Our prior work investigated CF in KANs on Split-MNIST [5], [19] and introduced progressive spline-freezing strategies, where subsets of spline parameters are frozen based on importance scores and may be combined with replay [1]. The observed pattern is consistent with broader CL experience: replay can be effective, but when replay is constrained, additional update controls may become valuable for retention [1]. This extended paper builds on that foundation by positioning freezing and replay within a unified KCF, where replay policy/budget and freezing granularity/fraction are treated as explicit controls that can be tuned to manage the stability-plasticity trade-off. Because CL systems are increasingly deployed in regulated socio-technical environments, it may also be useful to view controlled adaptation through a broader compliance and governance lens. Saidane and Al-Sharieh provide a compliance-driven framework for privacy and security in highly regulated e-government environments, emphasizing structured controls and accountability [22]. Although the focus here remains on technical controls for model adaptation, that governance-oriented view supports the design of mechanisms that are explicit, configurable, and auditable - qualities that align directly with KCF's emphasis on controllable, inspectable adaptation.

III. KAN Control Framework

In this section, KCF is introduced as a unifying view of CL in KANs as a controlled adaptation process. Building on our earlier progressive freezing results [1] and the spline-parameterized KAN layer structure [13], the stability-plasticity trade-off is made more explicit by exposing a small set of tunable controls that govern (i) what past information is retained and replayed, and (ii) how strongly model plasticity is restricted during adaptation. The aim is not to propose a single best setting, but to provide a compact vocabulary for describing and comparing settings under matched constraints.

A. Controlled adaptation view

A two-phase CL protocol (Task A followed by Task B) is considered, and controlled adaptation is framed as choosing a control vector that determines replay behavior and plasticity restriction during Task B training. Figure 1 summarizes the resulting pipeline. After training on Task A, samples may be stored into a bounded memory buffer. During Task B training, Task B minibatches are interleaved with replayed samples at a prescribed intensity, while plasticity may be restricted by freezing a subset of spline parameters according to an importance score. In the evaluation phase, Task A retention after Task B, Task B acquisition, and aggregate performance are measured.

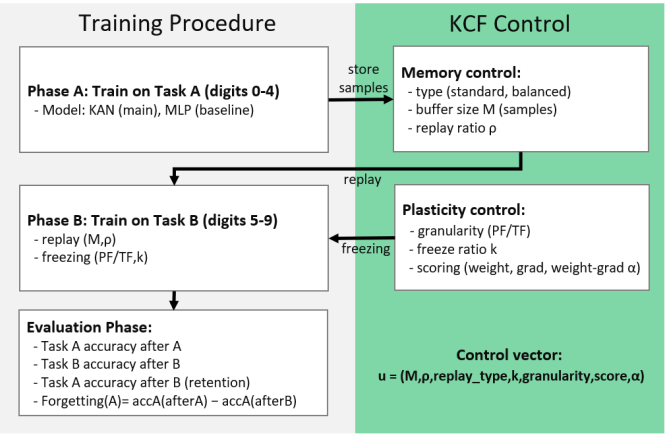

Figure 1. KCF controlled adaptation pipeline for Split-MNIST. KCF separates memory control (replay type, buffer size M, replay ratio ρ) from plasticity control (freezing granularity, freeze ratio k, scoring, and weighting α), yielding an explicit control vector $\mathbf{u}$.

B. Control vector

The KCF control vector is defined as

$$\mathbf{u} = (M, \rho, \texttt{replay\_type}, k, \texttt{granularity}, \texttt{score}, \alpha), \quad (1)$$

where M bounds the replay buffer size, ρ specifies replay intensity, `replay_type` selects the sampling policy, and $(k, \texttt{granularity}, \texttt{score}, \alpha)$ governs progressive freezing. In the present setting, freezing targets the spline-parameter components of KAN layers, i.e., the parameters that define the learned edge functions. Two granularities are considered: (i) Point-level Freezing (PF), which freezes individual spline coefficients, and (ii) Tensor-level Freezing (TF), which freezes entire coefficient vectors associated with an input-output feature pair, as introduced in our prior work [1]. The freeze ratio $k \in [0, 1]$ specifies what fraction of coefficients (PF) or curves (TF) are frozen according to an importance score, and α (when used) controls how multiple score components are combined or weighted within the scoring rule.

C. KCF trade-off intuition

KCF is organized around a practical observation: reducing CF typically requires either more rehearsal (larger M and/or higher ρ) or stronger restrictions on plasticity (larger k), and both options come with different cost profiles. Figure 2 illustrates this intuition. High memory and high replay intensity can reduce forgetting but increase storage and training-time overhead; strong plasticity restriction can also prevent forgetting but may limit new-task learning or lead to under-adaptation. In KCF, the goal is often an efficient retention regime (a "sweet spot") where modest replay budgets combined with targeted freezing yield strong retention without relying on large buffers or overly restrictive freezing.

D. Control knobs and operational meaning

Table I summarizes the KCF knobs, their practical meaning, expected effects on forgetting and adaptation, and the main

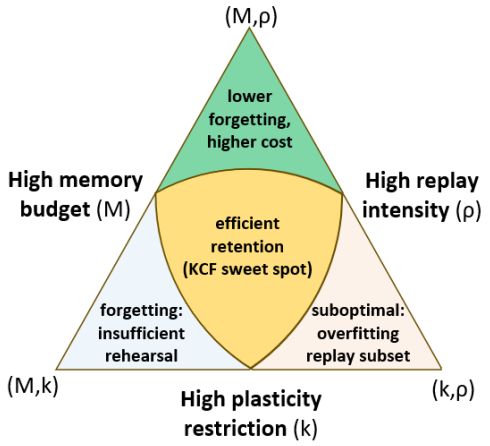

Figure 2. Conceptual KCF trade-off between memory budget (M), replay intensity (ρ), and plasticity restriction (k). The goal is efficient retention: low forgetting at controlled cost.

cost implications. This explicit mapping is central to KCF. CL configuration is treated as a constrained design choice, where $\mathbf{u}$ is selected to balance retention, learning, and operational cost.

E. Outcome measures for control selection

Given $\mathbf{u}$, controlled adaptation is evaluated using retention and acquisition measures that are standard in CL [3], [20]. In the two-task Split-MNIST protocol [5], [19], Task A accuracy after Task A (initial learning), Task B accuracy after Task B (new-task acquisition), Task A accuracy after Task B (retention), and forgetting (the drop in Task A performance after learning Task B) are reported [1]. In later sections, these metrics are used to compare KCF configurations under matched constraints (fixed M and ρ) and to identify regimes where progressive freezing may provide the largest benefit relative to replay alone.

IV. Methodology

A. Task protocol: Split-MNIST sequential learning

CL is evaluated on the Split-MNIST two-task protocol [5], [19]. Task A contains digits $\{0, 1, 2, 3, 4\}$ and Task B contains digits $\{5, 6, 7, 8, 9\}$. Training proceeds in two rounds, and the same model instance is carried forward between rounds (i.e., parameters are updated continuously rather than reinitialized).

In Phase A (the first training round), the model is trained only on Task A until convergence under the chosen training setup, and Task A accuracy after Phase A ($A1$) is recorded. Phase B (the second training round) then continues training on Task B under a selected KCF control vector. After Phase B, Task A accuracy is re-evaluated ($A2$) to quantify retention, and Task B accuracy is also recorded to capture post-adaptation performance on the new task. Forgetting is measured as the drop in Task A performance induced by Phase B training, computed as $A1 - A2$ (so larger values indicate greater forgetting, while values closer to zero indicate stronger retention).

TABLE I
KCF control knobs and meaning.

Param.	Meaning	Impact
M	Replay buffer size (stored samples)	Retention: more coverage → less forgetting (diminishing returns at high M). Cost: storage + I/O overhead.
ρ	Replay intensity during Task B	Retention: more rehearsal → less forgetting. Cost: extra training steps → higher time/compute.
`type`	Replay policy (standard vs. balanced)	Retention: balanced sampling may improve class coverage when M is small. Cost: minor sampling/stratification overhead.
k	Freeze ratio (fraction frozen)	Retention: higher k can reduce interference. Adaptation: too high k may limit Task B learning. Cost: low runtime overhead (mainly masking).
`gran.`	Freezing granularity (PF vs. TF)	Retention/adaptation: PF protects fine-grained coefficients; TF preserves structured curves; trade-offs can be regime-dependent. Cost: similar overhead for both.
`score`	Importance scoring (w, g, w×g)	Behavior: determines which parameters/curves are protected; may affect stability and variance. Cost: small bookkeeping to compute scores.
α	Score mixing weight (if used)	Behavior: tunes sensitivity to magnitude vs. gradient signal (stability vs. plasticity). Cost: negligible.

Figure 3 illustrates the baseline sequential protocol in which Phase B training is performed on Task B without replay and without any freezing. In this setting, the dashed arrow denotes that the same model continues to be optimized on the new task, but Task A samples are no longer available during Phase B. The metric $A1$ is computed immediately after Phase A, and $A2$ is computed after completing Phase B; the difference $A1 - A2$ summarizes the retention loss attributable to task-incremental training.

Figure 4 illustrates replay-based CL. After Phase A, a bounded replay buffer is populated with at most M Task A samples (selected according to the replay policy used in the experiment, e.g., standard or balanced replay). During Phase B, Task B optimization is performed while replayed Task A samples are mixed with Task B minibatches at intensity ρ. In this view, (M, ρ) jointly control the amount of past information reintroduced during adaptation: M limits the available memory of prior data, while ρ determines how strongly that memory is rehearsed during Phase B updates.

Finally, Figure 5 shows the replay+freezing variant used for controlled adaptation in KANs. In addition to replay with buffer size M and intensity ρ, Phase B updates are restricted by a freeze mask that reduces plasticity by preventing updates

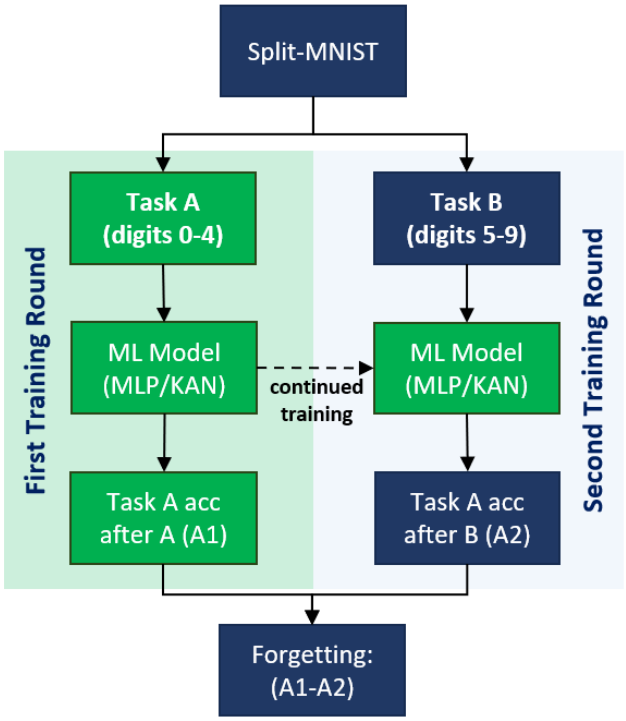

Figure 3. Baseline CL protocol on Split-MNIST (no replay, no freezing). Forgetting is measured as the drop in Task A performance after training on Task B ($A1 - A2$).

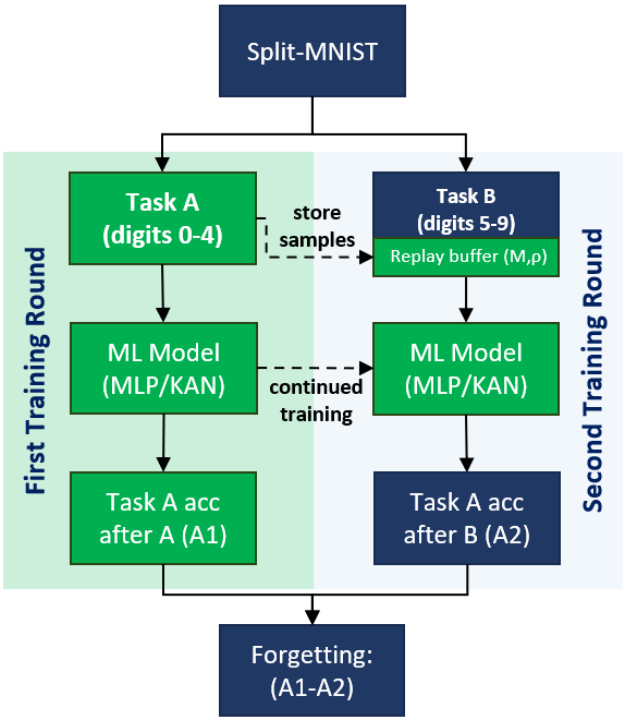

Figure 4. Replay-based CL. After Phase A, samples are stored into a bounded buffer (M). During Phase B, replayed samples are interleaved with Task B training at intensity ρ.

to a selected subset of spline parameters. The freeze ratio k specifies the fraction of parameters to be frozen, and the freezing granularity (PF/TF) determines whether freezing is applied at the point level or at a coarser tensor/curve level as discussed later. In this configuration, replay contributes to retention by reintroducing Task A samples, while freezing constrains which parts of the model are allowed to change during Task B training, thereby shaping the retention-adaptation trade-off.

B. Models: MLP baseline and KAN main model

A standard MLP baseline is compared against a KAN model used as the primary architecture under KCF. Both models operate on 28×28 grayscale images flattened to 784 input features and output 10 logits. The focus is not on optimizing architectures, but on comparing adaptation behavior under fixed, simple models.

a) MLP baseline: The MLP uses a single hidden layer with 128 neurons and fixed activations on nodes (Figure 6). This baseline represents the standard fully-connected feed-forward setting that is often used to study CF in simple benchmarks.

b) KAN model: The KAN replaces fixed node activations with learnable univariate nonlinear functions on edges (Figure 7), parameterized as spline-based functions as in prior KAN formulations [13]. The architecture is configured as [784, 128, 10]: one hidden layer of 128 nodes between 784 flattened inputs and 10 output logits, using cubic B-spline edge functions (spline order $p = 3$, grid size $G = 5$) with a SiLU residual activation, yielding $G + p = 8$ spline coefficients per input–output feature pair and approximately 1.02M parameters in total. In this CL setting, plasticity is primarily governed through the spline parameterization, which allows targeted freezing strategies [1].

C. Memory control (replay control)

As outlined in Table I, replay is governed by $(M, \rho, \texttt{replay\_type})$:

- Buffer size M: the maximum number of samples stored after Phase A.
- Replay intensity ρ: the relative rate at which replayed samples are reused during Phase B (i.e., the fraction/ratio of replayed batches or samples injected alongside Task B training).
- Replay type: standard (uniform sampling from the buffer) versus balanced (class-balanced sampling *within the replay buffer*), plus none (no replay).

Task B minibatches are sampled normally; `standard` vs. `balanced` affects only how replay samples are drawn from the buffer. These controls reflect a common operational constraint: retention improvements may need to be achieved under strict limits on stored data and replay throughput. In that scenario, the same change in ρ can sometimes matter more than a modest change in M, since ρ directly increases how often older data is revisited during optimization.

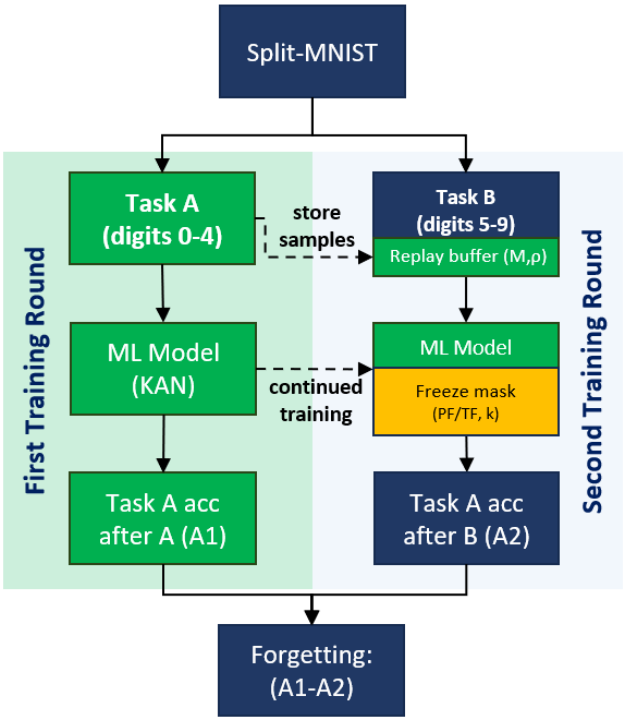

Figure 5. Replay + progressive freezing. During Phase B, replay is combined with a freeze mask that restricts plasticity by freezing a fraction k of spline parameters at a selected granularity (PF/TF).

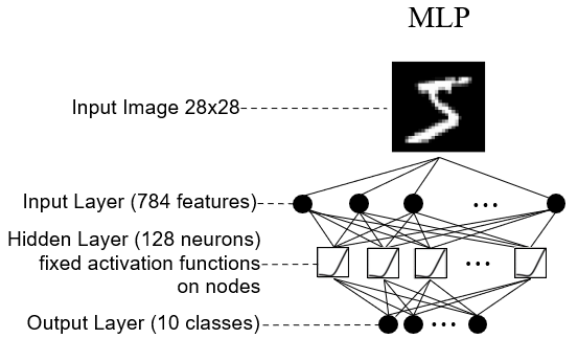

Figure 6. MLP baseline schematic. Fixed activation functions are applied at nodes in a standard fully-connected network.

D. Plasticity control (progressive freezing)

Plasticity control is implemented by freezing a subset of spline parameters during Phase B. Two freezing granularities are considered (Figure 8):

- TF: whole spline "curves" are frozen (blocks associated with an input-output feature pair).
- PF: individual spline coefficients are frozen (fine-grained points) within curves.

A freeze ratio $k \in [0, 1]$ specifies the fraction of curves (TF) or coefficients (PF) to freeze, selected by an importance score. Multiple scoring rules are evaluated: weight, grad, and

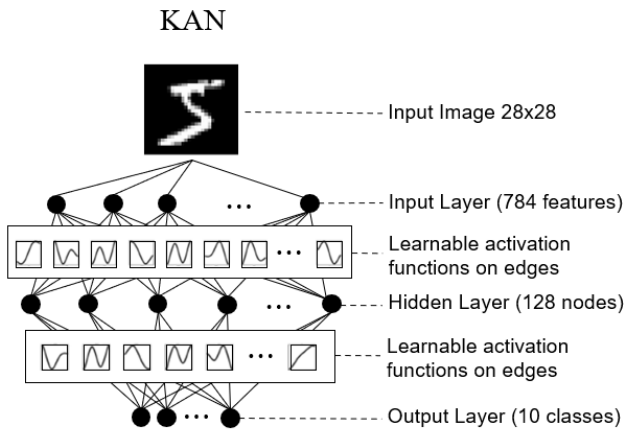

Figure 7. KAN architecture schematic. Learnable activation functions are placed on edges, enabling structured control of functional parameters.

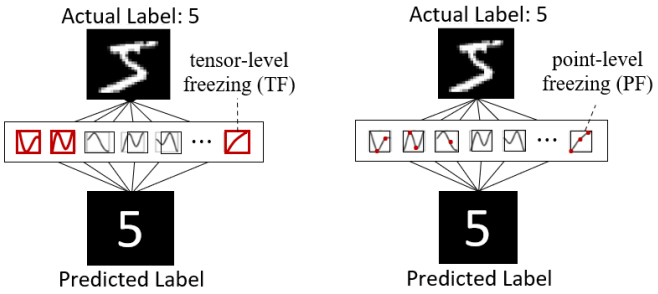

Figure 8. Freezing granularity intuition. TF freezes whole spline curves (structured blocks), while PF freezes individual spline coefficients (fine-grained points) within curves.

weight_grad (weight-gradient product), consistent with the progressive freezing formulation in [1]. TF tends to enforce structured restrictions that preserve entire functional subcomponents, while PF provides finer control that may protect local parameter importance at higher resolution. Neither option is assumed to be universally better; rather, their behavior is treated as part of the control space.

E. Training configuration and evaluation metrics

All configurations are trained for a fixed number of epochs in Phase A and Phase B with the same optimizer and learning rate, and are repeated across multiple random seeds. Task A accuracy after Phase A ($A1$), Task A accuracy after Phase B ($A2$), Task B accuracy after Phase B, and forgetting are reported:

$$\text{Forgetting}(A) = A1 - A2. \quad (2)$$

Aggregate accuracy after Phase B is also reported when overall performance comparisons are needed.

TABLE II
FULL EXPERIMENT GRID (CONTROLS AND VALUES).

Control	Values
M	{50, 100, 200, 500}
ρ	{0.05, 0.1, 0.2, 0.3}
replay_type	{none, standard, balanced}
granularity	{TF, PF}
k	{0.0, 0.05, 0.1, 0.25, 0.5, 0.75}
score	{weight, grad, weight_grad}
seeds	{42, 27, 101}

F. Full experiment grid

Table II lists the complete configuration grid explored in this work, comprising 270 experiments in total (including three random seeds per configuration). The grid spans replay budgets (M), replay intensities (ρ), replay policies, freezing granularities and ratios, and importance scoring rules. To keep a controlled, grid-level study of KCF knobs under a fixed computational budget, a deliberately low-epoch training protocol is used (3 epochs for Phase A and 3 epochs for Phase B) across all configurations. This setting is not intended to maximize absolute accuracy. Instead, it tends to amplify interference effects and provides a repeatable stress-test where differences attributable to replay intensity (M, ρ) and plasticity restriction $(k, \texttt{granularity}, \texttt{score})$ can be seen more clearly under matched training time. Using the same epoch budget for every run also helps reduce the chance that retention gains are simply artifacts of longer optimization, and it enables a fairer comparison between replay-only and replay+freezing regimes when replay throughput is itself a cost driver. The conclusions in this paper are therefore framed as trends within this controlled Split-MNIST protocol [5], [19]: replay remains the dominant lever for retention, while progressive freezing tends to provide its largest benefit in constrained replay regimes by improving retention efficiency under fixed compute.

V. RESULTS

Multi-seed results (mean $\pm$ std) are reported for the Split-MNIST CL protocol described in Section IV. The objective is to evaluate KCF under matched constraints and to characterize when progressive freezing appears to provide reliable retention gains beyond replay alone. Some variability across seeds is expected, so emphasis is placed on consistent trends rather than on single-run outcomes.

A. Baseline catastrophic forgetting without controls

The baseline sequential training setting (no replay, no freezing) is considered first. In Figure 9, both KAN and MLP learn Task A well after Phase A, and Task B is learned after Phase B, yet Task A performance collapses after Phase B, yielding near-complete forgetting (Task A accuracy after B ≈ 0). The two-task split therefore induces strong interference and can serve as a compact stress-test for controlled adaptation.

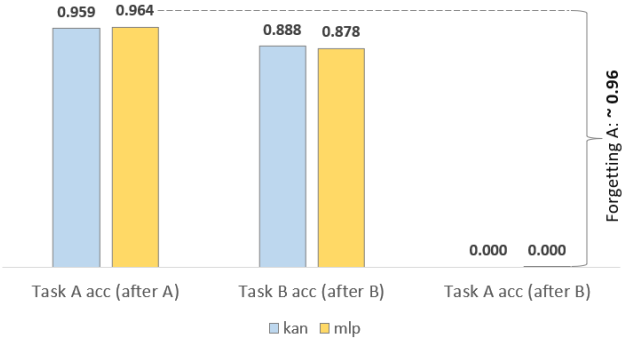

Figure 9. Baseline sequential training (no replay, no freezing). Both models learn Task A and Task B in isolation, but Task A accuracy after training on Task B collapses, indicating near-complete CF.

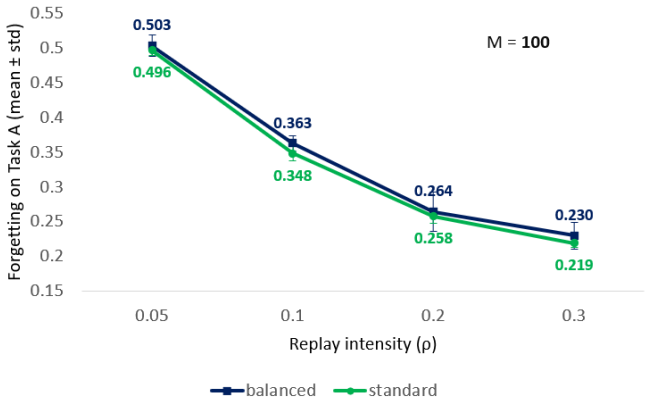

Figure 10. Replay-only results at $M = 100$: forgetting on Task A decreases as replay intensity ρ increases. Standard and balanced replay show similar trends, with small differences that depend on ρ.

B. Replay establishes a controllable retention baseline

Replay-based rehearsal provides the primary memory control in KCF. In Figure 10, increasing replay intensity ρ substantially reduces forgetting for a fixed memory budget ($M = 100$). Across the tested range, forgetting decreases as replay intensity increases, which suggests that rehearsal strength is a major driver of retention in this benchmark scenario. Differences between replay policies (standard vs. balanced) are comparatively modest at this budget, although small regime-dependent differences can be seen at particular ρ values.

C. Replay performance landscape across budgets and intensities

To better understand replay-only behavior, the retention and forgetting landscape over (M, ρ) is visualized for both replay policies. Figures 11–14 summarize replay-only outcomes as heatmaps. Two patterns are observed repeatedly. First, increasing either M or ρ improves retention (Task A accuracy after Phase B) and reduces forgetting, with the best

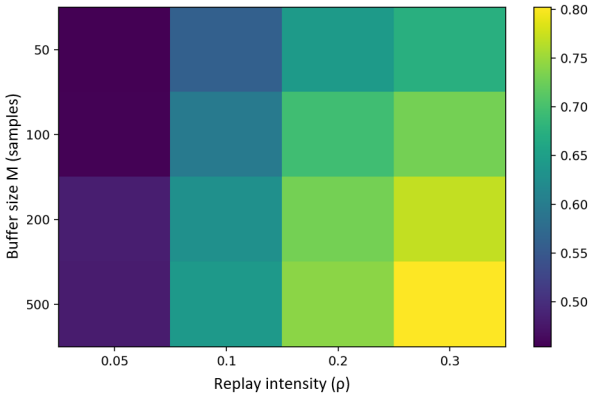

Figure 11. Replay-only retention landscape (balanced replay): Task A accuracy after Phase B (mean) as a function of memory budget M and replay intensity ρ.

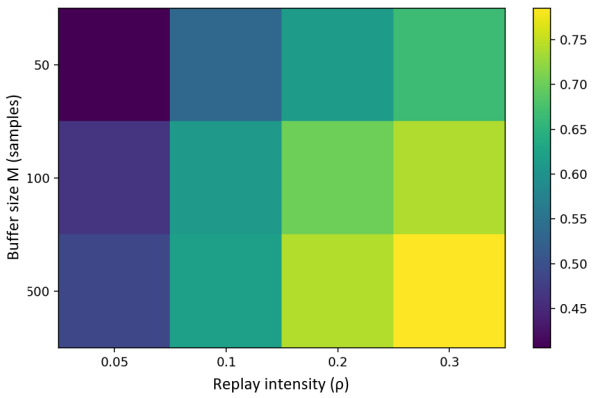

Figure 12. Replay-only retention landscape (standard replay): Task A accuracy after Phase B (mean) as a function of M and ρ.

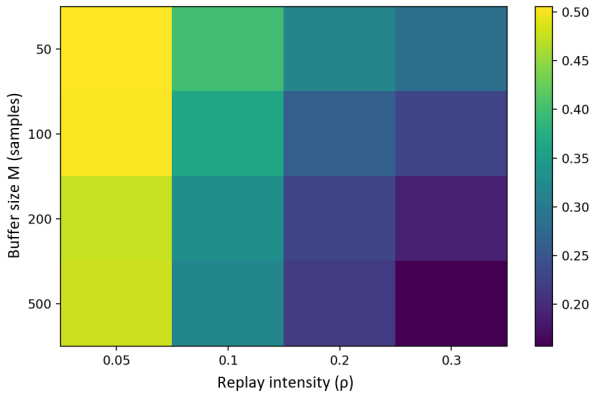

Figure 13. Replay-only forgetting landscape (balanced replay): forgetting on Task A (mean) as a function of M and ρ. Lower values indicate better retention.

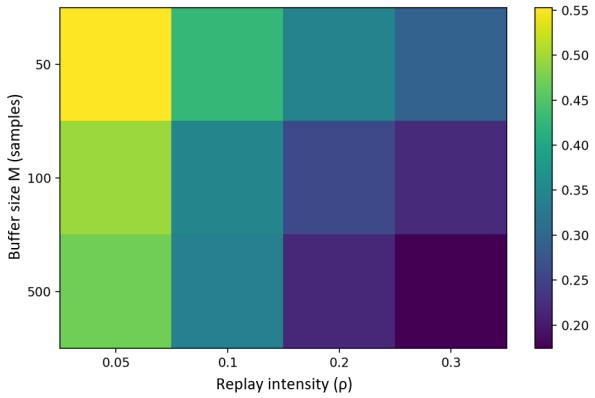

Figure 14. Replay-only forgetting landscape (standard replay): forgetting on Task A (mean) as a function of M and ρ. Lower values indicate better retention.

outcomes occurring when both are large. Second, for a fixed M, increasing ρ often yields a stronger improvement than increasing M by a comparable step, which is consistent with replay throughput being a particularly influential lever in this setting.

Balanced and standard replay lead to similar qualitative landscapes, although small differences can appear in low-budget corners. When both M and ρ are small, both policies remain in a high-forgetting regime. As budgets increase, retention improves and the two policies tend to converge. These corners are also where complementary controls, such as freezing, may matter most.

D. KCF gains under constraints: freezing helps when replay is limited

A central hypothesis of KCF is that progressive freezing may be most useful under constrained replay budgets, where replay alone does not fully prevent forgetting. Figure 15 compares replay-only and replay+KCF (replay plus progressive freezing) in two constrained operating points, alongside an unconstrained reference where replay-only is allowed to choose the best (M, ρ). In the constrained settings, adding KCF reduces forgetting relative to replay-only, yielding a consistent retention gain. In contrast, the unconstrained replay-only best-case provides the lowest forgetting overall, which suggests that freezing is not necessarily meant to beat "unlimited replay". Instead, it may help improve retention more efficiently when memory or replay throughput is capped.

E. Effect of plasticity restriction k and importance scoring

Next, the plasticity restriction knob k is examined together with the importance scoring rule. Figures 16 and 17 report forgetting as a function of k for TF and PF under a fixed replay configuration (standard replay, $M = 100$, $\rho = 0.2$). Across both granularities, increasing k tends to reduce forgetting, although the magnitude of improvement and stability across seeds can depend on the scoring rule. In this regime, gradient-based and combined (weight$\times$grad) scoring generally yield more favorable forgetting profiles than weight-only scoring, which suggests that update-sensitivity signals may be helpful when selecting parameters to protect during adaptation.

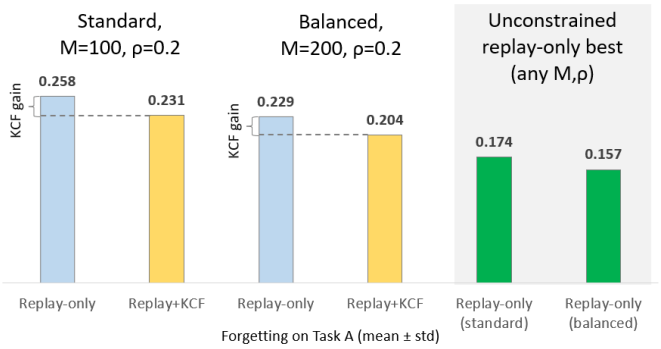

Figure 15. Constrained vs. unconstrained regimes. Under constrained replay budgets, adding KCF (replay + progressive freezing) reduces forgetting relative to replay-only. Under unconstrained replay (best (M, ρ)), replay-only achieves the lowest forgetting, supporting KCF's role as a constraint-aware control.

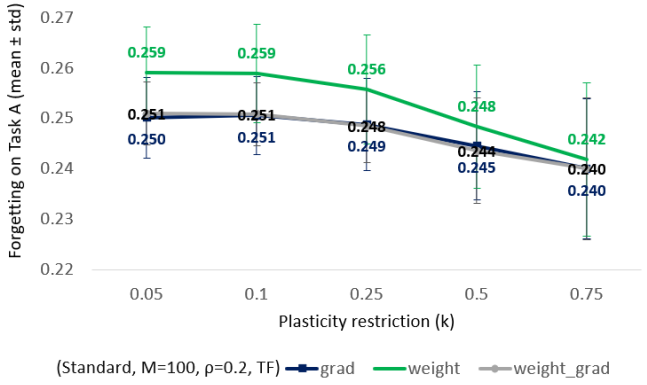

Figure 16. Forgetting vs. plasticity restriction k under TF (standard replay, $M = 100$, $\rho = 0.2$). The scoring rule influences the retention trajectory; gradient-informed scoring yields consistently lower forgetting in this regime.

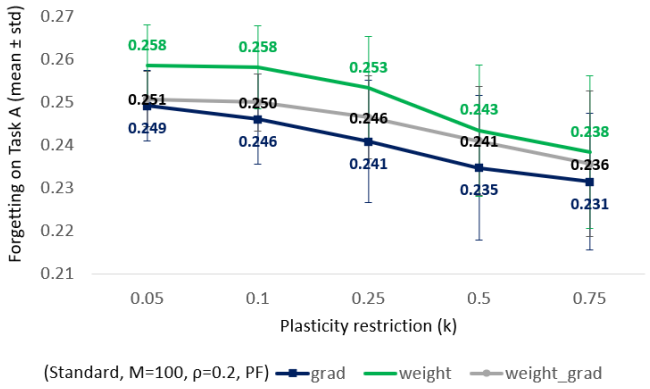

Figure 17. Forgetting vs. plasticity restriction k under PF (standard replay, $M = 100$, $\rho = 0.2$). PF shows a similar monotonic forgetting reduction with larger k, with differences across scoring rules.

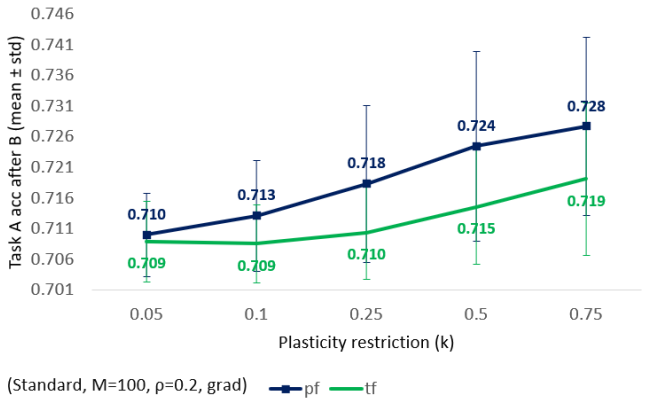

Figure 18. Task A retention (accuracy after Phase B) vs. plasticity restriction k under standard replay ($M = 100$, $\rho = 0.2$, grad). PF and TF both improve retention with larger k, with PF showing a modest advantage at higher k in this regime.

F. Granularity trade-offs: PF vs. TF impacts retention and new-task learning

KCF includes freezing granularity as a structural control: TF freezes whole spline "curves", whereas PF freezes individual spline coefficients. This trade-off is evaluated by comparing retention (Task A accuracy after Phase B) and new-task acquisition (Task B accuracy after Phase B) as k varies.

Figures 18 and 19 report Task A retention after Phase B for standard ($M = 100$) and balanced ($M = 200$) replay settings (both with $\rho = 0.2$ and gradient-based scoring). In both regimes, retention improves with larger k, and PF often yields slightly higher Task A retention than TF at larger k. This is consistent with PF providing finer-grained protection for previously learned function components, although the size of the gap varies across regimes.

Stronger plasticity restriction can also reduce Task B acquisition. Figures 20 and 21 show Task B accuracy after Phase B as k increases under the same settings as above. A mild but consistent downward trend is visible as k increases, which reflects the expected stability–plasticity trade-off: protecting prior knowledge may limit how fully the model adapts to the new task. In some regimes, TF preserves Task B performance slightly better than PF at higher k, which is consistent with TF applying more structured constraints than freezing many individual points.

G. Summary of empirical takeaways

Across the full experiment grid, four recurring patterns can be summarized: (i) without controls, both KAN and MLP exhibit near-complete CF; (ii) replay strength (through M and especially ρ) is the dominant driver of forgetting reduction and defines a smooth performance landscape over (M, ρ); (iii) progressive freezing contributes most under constrained replay budgets, where it can deliver more efficient retention gains relative to replay-only; and (iv) the KCF knobs expose a concrete stability–plasticity trade-off: increasing k tends to improve retention but can mildly reduce new-task acquisition, while PF/TF and scoring rules shift where a favorable balance may occur for a given constraint regime. These trends moti-

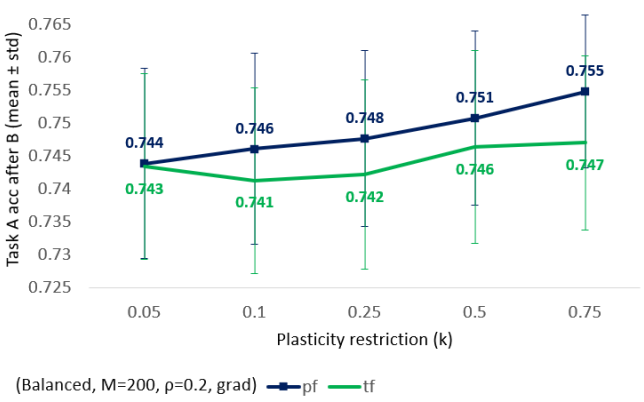

Figure 19. Task A retention vs. plasticity restriction k under balanced replay ($M = 200$, $\rho = 0.2$, grad). Retention increases with k for both PF and TF; the relative gap depends on k and the replay regime.

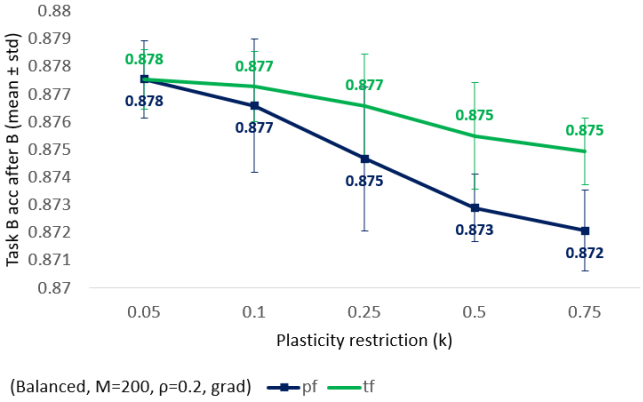

Figure 21. Task B accuracy after Phase B vs. k under balanced replay ($M = 200$, $\rho = 0.2$, grad). As k increases, Task B accuracy decreases modestly, highlighting the stability–plasticity trade-off under stronger freezing.

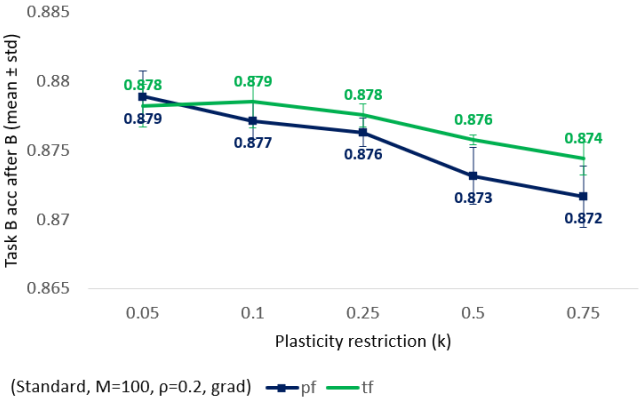

Figure 20. Task B accuracy after Phase B vs. k under standard replay ($M = 100$, $\rho = 0.2$, grad). Increasing k restricts plasticity and can slightly reduce new-task performance; PF and TF exhibit different sensitivity profiles.

vate the selection guidance and cost-of-adaptation discussion developed in the next section.

VI. Discussion

In KCF, the stability-plasticity trade-off is made operational: rather than treating replay or freezing as standalone techniques, a compact control vector is exposed that can be tuned under explicit resource constraints. The Split-MNIST experiments provide a small but fairly transparent setting in which these controls can be inspected, and several implications can be drawn for both CL research and for practitioners who may be considering KANs in dynamically changing environments, where data distributions may shift over time.

In many settings, sufficiently strong replay may be effective on its own. When replay is limited, other controls may become more important, and the practical question shifts to which choices offer the best return under a given budget.

Across the replay-only landscapes, increasing rehearsal strength through (M, ρ) consistently reduces forgetting. This is in line with rehearsal-based CL and supports the view that replay defines a strong baseline in this benchmark regime [10], [11]. Replay, however, is not free. Storage costs scale with M, training-time overhead tends to scale with replay throughput ρ, and operational policies may also place limits on what can be retained (data minimization) or how long it can be stored (retention windows). KCF is therefore best interpreted as a constraint-aware mechanism: when replay cannot be increased arbitrarily, progressive freezing can deliver measurable retention gains without requiring large buffers or high replay intensity. In that sense, replay is not replaced; rather, it may be made more efficient in the low-budget corners where CF is otherwise severe.

The plasticity control k exhibits the expected stability–plasticity behavior: stronger freezing reduces forgetting but can mildly reduce Task B performance. This aligns with a general CL principle that limiting representational drift may help retention at the cost of flexibility [3]–[5]. The contribution of KCF is mainly practical. Instead of optimizing only for minimal forgetting, k (together with replay controls) can be chosen to meet an operational objective. In some deployments, a small drop in new-task accuracy may be acceptable if retention is improved substantially. In others, fast adaptation may be prioritized and a smaller k (or replay-only) might be preferred. Neither choice is automatically correct, and it is often the constraint profile that decides.

A KAN-specific point is that freezing granularity can influence the retention/adaptation balance. TF preserves structured function components, while PF enables fine-grained protection of individual spline coefficients. These controls are not merely implementation details. They correspond to different assumptions about how knowledge is stored in KAN edge functions: PF protects localized parameter importance, while TF preserves entire functional substructures that may be reused across tasks. The observed differences suggest that CL for KANs may benefit from controls that operate at the level of learned functions rather than only at the level of conventional weights [13]. It is also plausible that these effects may depend on the task stream and the type of shift; the present benchmark is intentionally simple, so more complex regimes will be

needed for stronger claims.

The scoring rule (`weight`, `grad`, `weight_grad`) shapes which spline parameters are protected. In the reported experiments, gradient-informed scoring generally yields more favorable forgetting profiles than weight-only scoring in the tested regime. One possible interpretation is that gradient statistics capture update sensitivity during adaptation, providing a useful proxy for parameters likely to be affected by interference. More broadly, scoring can be seen as connecting classical importance-based CL (e.g., Elastic Weight Consolidation or synaptic-importance methods) with KAN-specific freezing: both aim to identify and protect knowledge-bearing components, but KCF applies this principle to spline-function parameters with tunable granularity [1], [4], [5].

The results support a simple selection heuristic:

- If replay budgets are generous (large M and/or high ρ), replay-only may be sufficient and will often be the simplest deployment choice.
- If replay is constrained, adding plasticity control may help: a moderate k with a gradient-informed scoring rule can provide retention gains, and PF/TF can be selected depending on whether retention or new-task learning is prioritized.
- KCF can be used as a design tool: configurations can be compared under matched budgets (fixed M and ρ) to identify an "efficient retention" operating point rather than a single global optimum.

This heuristic is consistent with the KCF "sweet spot" intuition: efficient retention is often obtained by combining modest rehearsal with targeted restrictions, rather than by maximizing a single knob.

This study uses Split-MNIST [5], [19] as a controlled two-task benchmark chosen deliberately to isolate CF dynamics under a transparent, reproducible protocol. This is a recognized limitation: Split-MNIST is a relatively simple benchmark, and the extent to which KCF's benefits generalize to high-dimensional data (e.g., Split-CIFAR, ImageNet subsets) or natural language processing tasks remains to be established. Extending KCF evaluation to multi-task class-incremental streams, domain-shift benchmarks, and non-vision settings is therefore the most important empirical next step, and the control-vector formulation is designed to transfer without modification to those settings.

A second limitation concerns the static nature of the freeze ratio k. In the current formulation, k is pre-configured before Phase B begins and does not change in response to observed drift or task difficulty. An adaptive scheduling strategy—one that conditions k on drift-detection signals or on running estimates of forgetting severity—could provide a more favorable retention/adaptation balance when distributional change is gradual or irregular. This is a concrete planned extension of KCF.

Third, all experiments use a two-task sequential protocol. Evaluating KCF over longer task sequences (e.g., 10+ tasks under class-incremental or domain-incremental protocols) is needed to characterize how accumulated freezing constraints interact with ongoing plasticity demands. The present results should therefore be interpreted as characterizing KCF behavior in constrained, short-horizon regimes; stronger claims about long-horizon stability require dedicated multi-task experiments.

Finally, while the present paper emphasizes CL, KCF is motivated as a broader abstraction for controlled adaptation in dynamically changing environments. A longer-term direction is to integrate KCF with drift detection and governance mechanisms so that when and how models adapt can be made explicit and auditable in deployed systems [22].

VII. Conclusion

In this paper, the KCF was presented as a controlled adaptation view of CL in KANs. Extending our prior work on progressive spline freezing [1], this paper makes four novel contributions relative to that work: (i) KCF is formalized as a compact, unified control vector that separates memory and plasticity controls into an explicit, reproducible specification; (ii) a multi-seed reliability characterization is provided that exposes variance patterns absent from single-run comparisons in our previous work; (iii) a granularity-aware analysis is introduced, distinguishing point-level (PF) and tensor-level (TF) freezing behaviors and their regime-dependent trade-offs; and (iv) actionable selection guidance is derived for common resource-constrained operating scenarios.

Through multi-seed experiments on Split-MNIST [5], [19], it was found that replay establishes a strong retention baseline and defines a smooth performance landscape over (M, ρ), while progressive freezing tends to provide its largest benefits under constrained replay budgets, where it can improve retention efficiency without relying on maximal replay. Overall, the results support KCF as a practical lens for configuring KAN continual learning: matched-budget comparisons are enabled, the stability–plasticity trade-off is clarified, and selection guidance can be derived for common resource constraints. Future work may benefit from longer task streams and more realistic shifts, from adaptive control schedules, and from tighter integration of controlled adaptation with governance and compliance objectives in deployed systems.

Acknowledgment

General-purpose online and cloud-based tools, including AI-assisted writing assistants, were used during the preparation of this work solely for language editing and text refinement; they were not used for experiment design, data collection, statistical analysis, or generation of results, figures, or tables. All scientific contributions, experimental findings, and conclusions reflect the authors' own work.

References

[1] E. Ostanin, N. Djosic, F. Hussain, S. Sharieh, A. Ferworn, and M. Sharieh, "Progressively Overcoming Catastrophic Forgetting in Kolmogorov–Arnold Networks", in *IARIA Congress 2025: The 2025 IARIA Annual Congress on Frontiers in Science, Technology, Services, and Applications*, Jul. 2025, pp. 138-143.

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

[3] 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.
[4] 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.
[5] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence", in *Proc. 34th Int. Conf. on Machine Learning (ICML)*, vol. 70, pp. 3987–3995, 2017.
[6] Z. Li and D. Hoiem, "Learning without forgetting", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935–2947, 2018.
[7] M. Farajtabar, N. Azizan, A. Mott, and A. Li, "Orthogonal gradient descent for continual learning", in *Proc. 23rd Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, vol. 108, pp. 3762–3773, 2020.
[8] A. A. Rusu et al., "Progressive neural networks", *arXiv preprint arXiv:1606.04671*, 2016.
[9] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong learning with dynamically expandable networks", in *Proc. 6th Int. Conf. on Learning Representations (ICLR)*, 2018.
[10] A. Chaudhry et al., "On tiny episodic memories in continual learning", *arXiv preprint arXiv:1902.10486*, 2019.
[11] D. Rolnick, A. Ahuja, J. Schwarz, T. P. Lillicrap, and G. Wayne, "Experience replay for continual learning", in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.
[12] L. Liu, L. Liu, and Y. Cui, "Prior-free balanced replay: Uncertainty-guided reservoir sampling for long-tailed continual learning", *arXiv preprint arXiv:2408.14976*, 2024.
[13] Z. Liu et al., "KAN: Kolmogorov-Arnold networks", in *Proc. 13th Int. Conf. on Learning Representations (ICLR)*, 2025.
[14] 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.
[15] 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.
[16] E. Ostanin, N. Djosic, F. Hussain, S. Sharieh, and A. Ferworn, "From Theory to Practice: Evaluating and Enhancing Kolmogorov-Arnold Networks (KAN) Robustness Under Adversarial Conditions", *International Journal on Advances in Security*, vol. 18, no. 1 & 2, pp. 77-91, Jun. 2025.
[17] 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, article 10173, 2024.
[18] T. Alter, R. Lapid, and M. Sipper, "On the robustness of Kolmogorov-Arnold networks: An adversarial perspective", *arXiv preprint arXiv:2408.13809*, 2024.
[19] L. Deng, "The MNIST database of handwritten digit images for machine learning research", *IEEE Signal Processing Magazine*, vol. 29, pp. 141–142, Jun. 2012.
[20] M. Delange et al., "A continual learning survey: Defying forgetting in classification tasks", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3366–3385, 2022.
[21] A. Mallya, D. Davis, and S. Lazebnik, "Piggyback: Adapting a single network to multiple tasks by learning to mask weights", in *Proc. European Conference on Computer Vision (ECCV)*, pp. 67–82, 2018.
[22] A. Saidane and S. Al-Sharieh, "A compliance-driven framework for privacy and security in highly regulated socio-technical environments: an e-government case study", in *Research Anthology on Privatizing and Securing Data*, IGI Global, pp. 933–962, 2021.