BEYOND POINTWISE REPLAY: PATHWISE KNOWLEDGE CONSOLIDATION FOR CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual learning (CL) aims to acquire new knowledge without forgetting what was learned before. Replay-based methods mitigate forgetting by revisiting a small memory buffer, but they supervise each replayed sample only at isolated targets, its label or its stored logits. Such pointwise supervision leaves the model underconstrained in function space and permits sharp, localized updates that cause catastrophic forgetting. To address this limitation, we propose Pathwise Knowledge Consolidation (PKC), which turns each replayed sample into a continuum of prediction tasks indexed by a corruption level: the label is geodesically corrupted toward the stored logits, and a lightweight training-only conditioner produces a prediction for each corrupted label. This connects the two previously isolated replay targets by a continuous trajectory of supervision, so the model preserves not only what to predict but also how its predictions evolve. A variational objective couples target fidelity along the path with path smoothness; its Euler–Lagrange characterization balances target fitting against path curvature, and a Hölder-type bound shows that the smoothness energy controls prediction drift across corruption levels. Extensive experiments demonstrate that PKC improves CL accuracy over strong replay methods while substantially reducing forgetting at no additional inference cost. PKC further outperforms recent prompt-based methods.

1 INTRODUCTION

Continual learning studies how a model can continuously incorporate new knowledge without sacrificing performance on previously learned tasks. In practical applications, data often arrive sequentially, and storing all historical data is typically infeasible due to privacy, memory, or computational constraints. Under this setting, naively training on new tasks often causes catastrophic forgetting (McCloskey & Cohen, 1989; Ratcliff, 1990), where the parameters adapt to recent data at the expense of earlier knowledge. Therefore, the key objective of continual learning is to develop mechanisms that support continual adaptation while maintaining long-term knowledge retention.

Memory replay is one of the most effective strategies for addressing this challenge. By storing a small episodic buffer and revisiting past examples during training, replay-based methods anchor the model to earlier tasks and substantially reduce forgetting. Yet, despite their empirical success, most existing replay methods remain fundamentally *pointwise*. They preserve past knowledge only at isolated replay samples, typically through replayed labels, stored logits, or pointwise regularization. As a result, they do not explicitly constrain the local prediction structure surrounding each memory item, leaving room for substantial drift between replay states. In this paper, we propose a *pathwise knowledge consolidation* (PKC) perspective for replay-based continual learning. Instead of treating each memory sample as a single supervision point, we regard it as inducing a family of related prediction states and enforce smooth, consistent knowledge preservation along this path. This simple shift from pointwise rehearsal to pathwise consolidation leads to a richer and more robust mechanism for retaining past knowledge under fixed memory budgets. The overall learning process is illustrated in Figure 1.

Our theoretical analysis shows that the optimal prediction trajectory satisfies a second-order differential equation balancing target fitting and curvature minimization. While the gradient term encourages accurate predictions, the second-order term penalizes rapid changes along the corruption path, effectively constraining the model to follow low-curvature trajectories in function space. This is

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

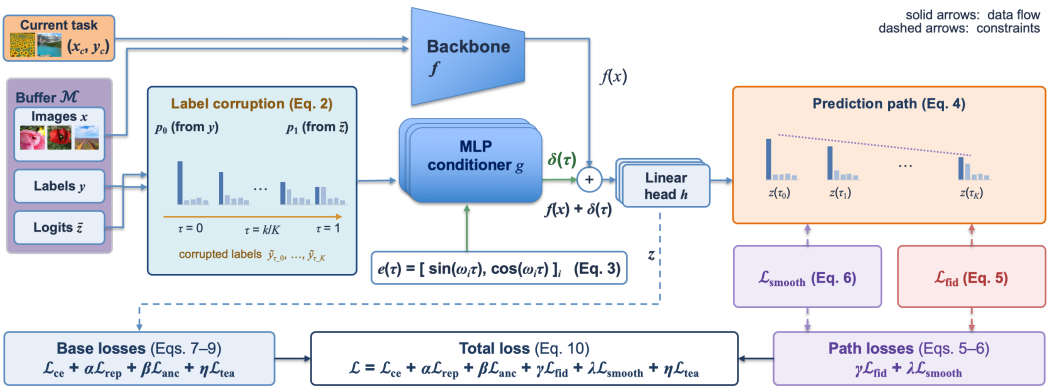

Figure 1: **The PKC training pipeline.** (a) A replayed triple $(x, y, \bar{z})$ from the buffer $\mathcal{M}$ and the current batch (x_c, y_c). The label and the stored logits are geodesically mixed into $K+1$ corrupted labels $\tilde{y}_{\tau_0}, \ldots, \tilde{y}_{\tau_K}$. (b) Images pass once through the unmodified backbone f. Each corrupted label, together with the sinusoidal embedding $e(\tau)$ of its level, is mapped by the zero-initialized conditioner g to a feature offset $\delta(\tau)$ that is added before the shared head h, giving the prediction path $z(\tau_0), \ldots, z(\tau_K)$. (c) Two constraints act on the path. $\mathcal{L}_{\text{fid}}$ trains every $z(\tau_k)$ to output y, and $\mathcal{L}_{\text{smooth}}$ keeps adjacent predictions close. Bottom row: the plain logits z feed the base losses, the path feeds $\gamma\mathcal{L}_{\text{fid}} + \lambda\mathcal{L}_{\text{smooth}}$, and one backward pass on the total loss updates f, g and h. At test time only $h(f(x))$ is used.

critical for continual learning: catastrophic forgetting often arises from sharp, localized deformations of the decision boundary induced by new-task updates. By explicitly discouraging high-curvature solutions, our method enforces smooth evolution of predictions, preventing abrupt boundary shifts and preserving previously learned knowledge.

Extensive experiments with a ResNet-18 trained from scratch, and on pretrained ViT backbones in the online Si-Blurry protocol, show that PKC improves CL accuracy over strong replay baselines on every benchmark, significantly reduces forgetting relative to pointwise replay, and surpasses recent prompt-based methods, at no additional inference cost during test time.

Our contributions can be summarized as follow:

- We propose a novel memory replay framework through *pathwise knowledge consolidation*: each buffered sample induces a corruption-indexed family of prediction tasks interpolating between its label and its stored logits, turning two isolated targets into one supervised path.
- A variational analysis shows the optimal trajectory obeys a second-order Euler–Lagrange equation penalizing curvature, and a Hölder-type bound shows the smoothness term directly controls prediction drift across corruption levels.
- With a ResNet-18 trained from scratch, our PKC obtains the highest Class-IL and Task-IL accuracy among replay methods and significantly reduces forgetting on every dataset; with a pretrained ViT, PKC also surpasses strong baselines in the Si-Blurry protocol.

2 Related Works

Continual learning (CL) has evolved into a diverse set of methodological paradigms, each addressing catastrophic forgetting from a different perspective. A commonly adopted taxonomy organizes existing approaches into several representative categories. (1) regularization-based methods aim to stabilize learning by restricting parameter updates so that previously acquired knowledge is not significantly altered. These approaches typically introduce penalties or constraints that encode the importance of parameters for past tasks, thereby discouraging destructive updates (Kirkpatrick et al., 2017; Schwarz et al., 2018; Zenke et al., 2017; Rebuffi et al., 2017; Chaudhry et al., 2018; Aljundi et al., 2018; Hou et al., 2019; Cha et al., 2021; Liu et al., 2024; Lewandowski et al., 2025). (2) memory-based (replay) methods explicitly maintain a subset of past data in a buffer and interleave these stored samples with current-task training. By revisiting prior examples, the model can reinforce earlier knowledge and reduce forgetting through rehearsal-like mechanisms (Lopez-Paz & Ranzato, 2017; Riemer et al., 2019; Chaudhry et al., 2019b; Buzzega et al., 2020; Pham et al., 2021; Verwimp

et al., 2021; Arani et al., 2022; Caccia et al., 2022; Wang et al., 2022a; Bonicelli et al., 2022; Sarfraz et al., 2023; Wang et al., 2023; Yang et al., 2023; Liang & Li, 2023; Wang et al., 2024; Wen et al., 2024; Dang et al., 2025; Tong et al., 2025). (3) Bayesian continual learning approaches adopt a probabilistic view by modeling uncertainty over parameters or functions. These methods update posterior distributions as new data arrives, enabling principled trade-offs between retaining prior knowledge and incorporating new information (Nguyen et al., 2018; Ritter et al., 2018; Kao et al., 2021; Henning et al., 2021; Pan et al., 2020; Titsias et al., 2020; Rudner et al., 2022; Li et al., 2025). (4) architecture-based strategies mitigate interference by dynamically modifying the model structure, such as expanding capacity, allocating task-specific subnetworks, or selectively reusing components across tasks (Mallya & Lazebnik, 2018; Serra et al., 2018; Li et al., 2019; Hung et al., 2019; Yu et al., 2025). (5) prompt-based approaches have recently emerged, particularly in the context of large-scale pretrained models. These methods encode task-specific knowledge in learnable prompts or embeddings, allowing the model to adapt to new tasks while largely preserving shared backbone parameters (Wang et al., 2022c;b; Kang et al., 2025).

Existing replay-based continual learning methods are largely pointwise. They mitigate forgetting by imposing replay constraints at discrete memory states, for example through supervised rehearsal on buffered samples (Chaudhry et al., 2019b), logit matching on replayed examples (Buzzega et al., 2020), loss redesign (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2021; Caccia et al., 2022) or sample selection (Aljundi et al., 2019; Tong et al., 2025) at the replay point itself. In contrast, our method is pathwise. Rather than treating each replay sample as a single supervision anchor, we associate each memory item with a corruption-indexed prediction trajectory and optimize a replay objective that enforces both target fidelity and smoothness along this trajectory. From this perspective, our method extends replay from pointwise consolidation to pathwise consolidation.

3 METHOD

3.1 PROBLEM DEFINITION

Let $x \in \mathcal{X}$ denote an input and $y \in \mathcal{Y} = \{1, \dots, C\}$ its label. A continual learner learns a sequence of tasks with datasets $\mathcal{D}_1, \dots, \mathcal{D}_N$, one at a time, and must learn a network f_θ on the whole sequence without forgetting the earlier tasks. The network consists of a backbone f producing features $f(x)$ and a linear head h producing logits $z = h(f(x)) \in \mathbb{R}^C$; we write θ for all trainable parameters. Replay-based methods keep a small memory buffer $\mathcal{M}$ of past samples, filled by reservoir sampling and revisited during training, and optimize

$$\min_\theta \; \mathbb{E}_{(x,y)\sim\mathcal{D}_t}\, \ell(f_\theta(x), y) \;+\; \alpha\, \mathbb{E}_{(x,y)\sim\mathcal{M}}\, \ell(f_\theta(x), y), \tag{1}$$

where ℓ is the cross-entropy loss. Each buffered point contributes a single supervision term, so previous knowledge is preserved only at discrete targets, the one-hot label or the logits stored with the sample (Buzzega et al., 2020). We call this pointwise replay.

3.2 A CONTINUATION VIEW OF PREDICTIVE CONSISTENCY

For each corruption level $t \in [0, 1]$ we form a corrupted target $\tilde{y}_t$, with larger t meaning stronger corruption; Section 3.3 gives the exact construction (Eq. 2: $\tilde{y}_t$ moves geodesically from the one-hot label at t=0 to the stored logits at t=1). The model receives the input together with the corrupted target and its level, and we write its prediction at level t as

$$\hat{y}_t := f_\theta(x, \tilde{y}_t, t).$$

Evaluating the same sample x at every corruption level traces a curve in prediction space, the *prediction trajectory* $\phi_\theta(t; x, y) := \hat{y}_t$, $t \in [0, 1]$; in practice we evaluate it at K+1 levels $t_k = k/K$, $k = 0, \dots, K$, which is what produces the path of Figure 1(c). Instead of a single mapping from x to y, learning is now the construction of a stable trajectory over corruption levels, and two requirements follow: predictions should remain accurate at every t, and they should not change abruptly as t varies. Together they give

$$\mathcal{J}(\theta) = \mathbb{E}_{(x,y)}\left[\int_0^1 \ell\big(f_\theta(x, \tilde{y}_t, t),\, y\big)\, dt + \lambda \int_0^1 \big\|\partial_t f_\theta(x, \tilde{y}_t, t)\big\|^2 dt\right], \qquad \phi_\theta(t; x, y) := f_\theta(x, \tilde{y}_t, t).$$

The corruption enters through the input of the network: at level t the model is fed the corrupted target $\tilde{y}_t$ and must still predict the clean label y, so the first term asks for accuracy under every degree of corruption and the second term asks that the prediction change smoothly as the corruption level varies from 0 to 1; $\ell(\cdot, y)$ is the same cross-entropy loss as in Eq. 1 and $\lambda > 0$ controls the strength of the path regularization. Taken together, the objective encourages the model to solve a continuum of related subproblems while maintaining a coherent solution path. We instantiate the corrupted targets $\tilde{y}_t$ in Section 3.3.

To connect this formulation with a discrete consistency objective, consider a partition

$$0 = t_0 < t_1 < \cdots < t_K = 1, \qquad \Delta t_k := t_{k+1} - t_k.$$

A natural discretization is

$$\mathcal{J}_K(\theta) = \mathbb{E}_{(x,y)} \left[\sum_{k=0}^{K} \ell\big(\phi_\theta(t_k; x, y), y\big)\, \Delta t_k + \lambda \sum_{k=0}^{K-1} \frac{\|\phi_\theta(t_{k+1}; x, y) - \phi_\theta(t_k; x, y)\|^2}{\Delta t_k} \right].$$

The first summation is a Riemann approximation to the integrated loss term. For the second summation, note that

$$\frac{\|\phi_\theta(t_{k+1}; x, y) - \phi_\theta(t_k; x, y)\|^2}{\Delta t_k} = \left\| \frac{\phi_\theta(t_{k+1}; x, y) - \phi_\theta(t_k; x, y)}{\Delta t_k} \right\|^2 \Delta t_k.$$

Hence, as the partition becomes finer, the discrete consistency term converges to

$$\int_0^1 \|\partial_t \phi_\theta(t; x, y)\|^2 \, dt.$$

This shows that the continuous formulation can be understood as the limit of pairwise consistency regularization across nearby corruption levels. Next, we state two propositions and a theorem; proofs are given in Appendix A.

Proposition 1. *Assume that for every (x, y), the map $t \mapsto \phi_\theta(t; x, y)$ is continuously differentiable on $[0, 1]$, and that $t \mapsto \ell(\phi_\theta(t; x, y), y)$ is continuous. Then, as $\max_k \Delta t_k \to 0$, the discrete objective $\mathcal{J}_K(\theta)$ converges to*

$$\mathcal{J}(\theta) = \mathbb{E}_{(x,y)} \left[\int_0^1 \ell\big(\phi_\theta(t; x, y), y\big)\, dt + \lambda \int_0^1 \|\partial_t \phi_\theta(t; x, y)\|^2 \, dt \right].$$

For a fixed pair (x, y), define the path energy

$$\mathcal{E}[\phi] = \int_0^1 \left(\ell(\phi(t), y) + \lambda \|\phi'(t)\|^2 \right) dt.$$

The next result gives the stationary condition for this functional.

Proposition 2. *Suppose $\ell(\cdot, y)$ is differentiable. If $\phi^\star$ is a stationary point of $\mathcal{E}$, then $\phi^\star$ satisfies*

$$\nabla_\phi \ell(\phi^\star(t), y) - 2\lambda \phi^{\star\prime\prime}(t) = 0, \qquad t \in (0, 1).$$

Proposition 2 gives a useful interpretation. The gradient term $\nabla_\phi \ell(\phi(t), y)$ pulls the trajectory toward the target, while the second-derivative term penalizes excessive curvature along the corruption path. Thus, the learned trajectory is encouraged to be both accurate and smooth.

The next theorem makes the role of the regularizer more explicit.

Theorem 1. *Assume that $\phi_\theta(\cdot; x, y)$ is absolutely continuous on $[0, 1]$. Then for any $s, t \in [0, 1]$,*

$$\|\phi_\theta(t; x, y) - \phi_\theta(s; x, y)\| \le |t - s|^{1/2} \left(\int_0^1 \|\partial_\tau \phi_\theta(\tau; x, y)\|^2 d\tau \right)^{1/2}.$$

The theorem shows that the path regularizer directly controls how much the prediction can change as the corruption level varies. In particular, when the regularization term is small, predictions at nearby corruption levels must remain close. This gives a simple stability interpretation of the objective: it encourages a coherent continuation from easier low-corruption subproblems to harder high-corruption

ones, rather than allowing the model to fit each level independently. The full procedure is given in Algorithm 1.

Algorithm 1 Pathwise Knowledge Consolidation (PKC)

Require: tasks $\{\mathcal{D}_1, \ldots, \mathcal{D}_N\}$; memory buffer $\mathcal{M}$ (capacity M, reservoir sampling); path length K with levels $\tau_k = k/K$; loss weights $\alpha, \beta, \gamma, \lambda, \eta$ (Eq. 10); sharpness s; temperature T; label-drop rate p_{drop}; anchor refresh rate ρ_a

initialize $\theta = (f, g, h)$ with the last layer of g set to zero; $\mathcal{M} \leftarrow \varnothing$; teacher $\theta_T \leftarrow \theta$
for task $t = 1, \ldots, N$ **do**
  for each iteration **do**
    sample a current batch $B_c = \{(x_c, y_c)\} \sim \mathcal{D}_t$ and a replay batch $B_m = \{(x_m, y_m, \bar{z}_m)\} \sim \mathcal{M}$
    plain forward (no corrupted label): $z_c = h(f(x_c))$, $z_m = h(f(x_m))$
    compute $\mathcal{L}_{\text{ce}}, \mathcal{L}_{\text{rep}}, \mathcal{L}_{\text{anc}}, \mathcal{L}_{\text{tea}}$ from z_c, z_m and the teacher $\theta_T = (f_T, h_T)$ (Eqs. 7–9)
    endpoints: $p_0 = \text{softmax}(s\,\text{onehot}(y_m))$; $p_1 = \text{softmax}(\bar{z}_m/T)$
    for $k = 0, \ldots, K$ **do**
      $\tilde{y}_{\tau_k}(c) = p_0(c)^{1-\tau_k}\, p_1(c)^{\tau_k} \big/ \sum_{c'} p_0(c')^{1-\tau_k}\, p_1(c')^{\tau_k}$ for $c = 1, \ldots, C$ ▷ Eq. 2
      with probability p_{drop} replace $\tilde{y}_{\tau_k}$ by the uniform distribution
      $z(\tau_k) = h\big(f(x_m) + g([\tilde{y}_{\tau_k};\, e(\tau_k)])\big)$ ▷ $e(\cdot)$: Eq. 3; one shared $f(x_m)$
    end for
    $\mathcal{L}_{\text{fid}}$ and $\mathcal{L}_{\text{smooth}}$ from $\{z(\tau_k)\}_{k=0}^{K}$ (Eqs. 5, 6)
    $\mathcal{L} = \mathcal{L}_{\text{ce}} + \alpha\mathcal{L}_{\text{rep}} + \beta\mathcal{L}_{\text{anc}} + \gamma\mathcal{L}_{\text{fid}} + \lambda\mathcal{L}_{\text{smooth}} + \eta\mathcal{L}_{\text{tea}}$; one backward step on θ
    reservoir-insert each (x_c, y_c, z_c) of B_c into $\mathcal{M}$ ▷ z_c stored as a constant
  end for
  task boundary: $\theta_T \leftarrow \theta$; $\bar{z} \leftarrow (1 - \rho_a)\bar{z} + \rho_a\, h_T(f_T(x))$ for all $(x, \cdot, \bar{z}) \in \mathcal{M}$
end for
inference: $\hat{y} = \arg\max h(f(x))$

3.3 Instantiating the path: label-space geodesic corruption

Each buffered sample is a triple $(x, y, \bar{z})$, where $\bar{z} \in \mathbb{R}^C$ is the vector of logits over all C classes that the model produced for x at the moment the sample entered the buffer. We store logits rather than only the label because they encode the old model's belief about *every* class, the dark knowledge such as inter-class similarity that a one-hot label discards; this is the same signal that DER++ (Buzzega et al., 2020) anchors to. We express both supervision signals as distributions over classes: a sharpened *label endpoint* $p_0 = \text{softmax}\big(s \cdot \text{onehot}(y)\big)$ and a softened *belief endpoint* $p_1 = \text{softmax}(\bar{z}/T)$, with sharpness s and temperature T. The corruption path is their geodesic in the probability simplex, evaluated at $K+1$ levels $\tau_k = k/K$:

$$\tilde{y}_\tau(c) \;=\; \frac{p_0(c)^{\,1-\tau}\, p_1(c)^{\,\tau}}{\sum_{c'=1}^{C} p_0(c')^{\,1-\tau}\, p_1(c')^{\,\tau}}, \qquad c = 1, \ldots, C, \quad \tau \in \{\tau_0, \ldots, \tau_K\}, \tag{2}$$

where $p_0(c)$ and $p_1(c)$ are the c-th components of the two endpoint distributions, so that $\log \tilde{y}_\tau(c) = (1-\tau)\log p_0(c) + \tau \log p_1(c)$ up to a constant independent of c: the path is linear interpolation in the log domain followed by renormalization. Because $s < \infty$ keeps every coordinate of p_0 strictly positive, the path is well defined at every level, and as τ grows the true-class mass melts while the structure of the old belief emerges. The corrupted labels are constants for the optimizer (no gradient flows into them) and each is independently replaced by the uniform distribution with probability p_{drop}. Corrupting toward $\bar{z}$ rather than toward an uninformative distribution is essential: the path then consolidates the old model's actual knowledge and stays consistent with the anchor loss, whereas corrupting toward noise turns the path into label-noise augmentation and in our early experiments reduced Class-IL accuracy.

3.4 Conditioned prediction head

As shown in Figure 1(b), the backbone is untouched and sees only the image, producing features $f(x)$ in a single pass. Conditioning happens entirely after the backbone. The level τ is encoded by a

fixed sinusoidal embedding of dimension d=64,

$$e(\tau) = \big[\sin(\omega_i \tau),\, \cos(\omega_i \tau)\big]_{i=1}^{d/2}, \qquad \omega_i = 1000 \cdot 10000^{-2i/d}, \tag{3}$$

that is, $d/2 = 32$ frequencies ω_i, each contributing one sine and one cosine, so that $e(\tau) \in \mathbb{R}^d$; the embedding has no learnable parameters and follows the positional encoding of transformers. Each corrupted label, concatenated with the embedding of its level, is mapped by a small MLP g (two hidden layers of width 256, SiLU activations) to an offset $\delta(\tau)$ of the same dimension as $f(x)$, and the shared linear head evaluates

$$z(\tau) \;=\; h\big(f(x) + g([\tilde{y}_\tau; e(\tau)])\big). \tag{4}$$

The last layer of g is zero-initialized, so $\delta \equiv 0$ at the start of training and the model coincides exactly with the plain network; the path structure grows from zero, and the additive form guarantees that removing g recovers the baseline architecture. The conditioner is the only addition: about 0.22M parameters, 2% of ResNet-18, used during training only. The cost is one backbone pass plus $K{+}1$ evaluations of the linear head per replay batch; at test time the corrupted-label branch is switched off and prediction reduces to $h(f(x))$ (the online Si-Blurry configuration is given in Appendix B).

3.5 Training objective

Two losses shape the path. *Fidelity* requires every node to recover the true class,

$$\mathcal{L}_{\text{fid}} = \frac{\sum_{k=0}^{K} w(\tau_k)\, \mathrm{CE}\big(z(\tau_k), y\big)}{\sum_{k=0}^{K} w(\tau_k)}, \tag{5}$$

with Min-SNR weights $w(\tau)$ (Hang et al., 2023) computed from the corruption schedule (numeric values in the appendix). The weights are large near τ=0, where the corrupted label is essentially the clean label and a hard cross-entropy is appropriate, and small near τ=1, where the corrupted label is the softened stored belief p_1. A light fidelity term at the belief end keeps y as the top class without destroying the soft structure that the anchor preserves.

Smoothness penalizes differences between adjacent nodes,

$$\mathcal{L}_{\text{smooth}} = \sum_{k=0}^{K-1} \phi\big(\|p(\tau_{k+1}) - p(\tau_k)\|_2^2\big), \qquad p(\tau) = \operatorname{softmax} z(\tau), \tag{6}$$

where ϕ is the pseudo-Huber function; per Proposition 1 this discretizes $\int \|\partial_\tau p\|^2 d\tau$ (norms are averaged over the class dimension so the scale is dataset-independent). These two terms are added to the base objective of the replay methods we build on. Let $B_c = \{(x_c, y_c)\} \sim \mathcal{D}_t$ be the current batch and $B_m = \{(x_m, y_m, \bar{z}_m)\} \sim \mathcal{M}$ the replay batch, and let $z_c = h(f(x_c))$ and $z_m = h(f(x_m))$ be the plain logits (no corrupted label) of the two batches. The base terms are

$$\mathcal{L}_{\text{ce}} = \mathrm{CE}_{\text{new}}(z_c, y_c) \;+\; \rho\, \mathcal{L}_{\text{group}}(z_c, y_c), \tag{7}$$

$$\mathcal{L}_{\text{rep}} = \mathrm{CE}(z_m, y_m), \qquad \mathcal{L}_{\text{anc}} = \|z_m - \bar{z}_m\|_2^2, \tag{8}$$

$$\mathcal{L}_{\text{tea}} = \mathrm{KL}\big(\operatorname{softmax}(h_T(f_T(x_m))/\kappa) \,\|\, \operatorname{softmax}(z_m/\kappa)\big) \text{ restricted to old classes.} \tag{9}$$

$\mathcal{L}_{\text{ce}}$ is the loss-decoupled current-task loss of LODE (Liang & Li, 2023): a cross-entropy restricted to the classes of the current task plus a group term, weighted by ρ, that balances old against new classes and prevents the new-class logits from overwhelming the old ones. $\mathcal{L}_{\text{rep}}$ and $\mathcal{L}_{\text{anc}}$ are the replay cross-entropy and the logit anchor of DER++ (Buzzega et al., 2020), both computed on the plain forward pass of the replay batch. $\mathcal{L}_{\text{tea}}$ is a knowledge-distillation term (Hinton et al., 2015) toward a teacher $\theta_T = (f_T, h_T)$, a frozen copy of the model taken at the last task boundary, with temperature κ and applied only to old-class logits. The full objective is

$$\mathcal{L} = \mathcal{L}_{\text{ce}} + \alpha \mathcal{L}_{\text{rep}} + \beta \mathcal{L}_{\text{anc}} + \gamma \mathcal{L}_{\text{fid}} + \lambda \mathcal{L}_{\text{smooth}} + \eta \mathcal{L}_{\text{tea}}. \tag{10}$$

Besides the loss weights γ and λ, PKC has four schedule constants (K, s, T, p_{drop}) that are fixed across all datasets and backbones; their selection is described in Appendix B. The others are inherited with their published values (α and β from DER++, ρ from LODE, η and κ from standard distillation) and are not tuned. Of the two loss weights, results are insensitive to λ over two orders of magnitude (Section 4.2), so in practice γ is the only weight that has to be set.

4 EXPERIMENTS

Datasets We validate the performance of PKC under both task-incremental (Task-IL) and class-incremental (Class-IL) settings using four benchmarks: CIFAR-10, CIFAR-100 Krizhevsky et al. (2009), Tiny-ImageNet Wu et al. (2017), and ImageNet-R Hendrycks et al. (2021). The task partitioning is configured as follows: 1) CIFAR-10: 5 tasks, 2 classes per task; 2) CIFAR-100: 10 tasks, 10 classes per task; 3) Tiny-ImageNet & ImageNet-R: 10 tasks, 20 classes per task.

Baselines For ResNet baselines, our approach is benchmarked against a broad spectrum of state-of-the-art CL methods, encompassing five distinct categories: (1) Regularization-based methods, including oEWC (Schwarz et al., 2018), SI (Zenke et al., 2017), LwF (Li & Hoiem, 2018), CPR (Cha et al., 2021) and GPM (Saha et al., 2021). (2) Bayesian-based methods, such as NCL (Kao et al., 2021). (3) Architecture-based methods like HAT (Serra et al., 2018). (4) Memory-based methods, including ER (Chaudhry et al., 2019b), A-GEM (Chaudhry et al., 2019a), HAL (Chaudhry et al., 2021), DER++ (Buzzega et al., 2020), ER-ACE (Caccia et al., 2022), LODE-DER++ (Liang & Li, 2023), CILA (Wen et al., 2024), FNC^2-HSD (Dang et al., 2025). (5) Coreset-based approaches: GSS (Aljundi et al., 2019), CSReL-LODE-DER++ (Tong et al., 2025). **For ViT baseline comparisons**, we compare PKC with the rehearsal-based methods ER (Chaudhry et al., 2019b), DER++ (Buzzega et al., 2020), ER-ACE (Caccia et al., 2022), RM (Bang et al., 2021), and CLIB (Koh et al., 2022), as well as the prompt-based methods MVP (Moon et al., 2023) and MISA (Kang et al., 2025).

Evaluation Metrics with ResNet as Backbone Following standard continual-learning evaluation protocols in the ResNet setting (Buzzega et al., 2020), we report average accuracy (Acc) and backward transfer (BWT) over the entire sequence of tasks. Results are reported under both task-incremental (Task-IL) and class-incremental (Class-IL) settings. **Evaluation Metrics with ViT as Backbone.** Following the evaluation setting adopted in prior ViT backbone setting (Moon et al., 2023), we assess model performance primarily using A_{AUC} and A_{Last}. Specifically, A_{AUC} (Koh et al., 2022) summarizes predictive performance throughout the learning process under an anytime inference setting, where task identities are unavailable at inference time, consistent with the "no test-time oracle" principle of GCL. In contrast, A_{Last} evaluates the model after completing the entire learning sequence and corresponds to the commonly used final average accuracy in conventional continual learning. To further quantify the degradation of previously acquired knowledge, which constitutes a major challenge for existing approaches, we additionally report F_{Last}, measuring the average amount of forgetting observed after training. More implementation details are provided in Appendix B.

4.1 MAIN RESULTS

From-scratch ResNet-18. Table 1 reports accuracy and Table 3 backward transfer. PKC obtains the highest Class-IL and Task-IL accuracy on all three datasets; the margin is largest on CIFAR-100, where it improves over CSReL-LODE-DER++, the strongest prior method, by 3.2 points and over DER++ by 8.8. The BWT shows where the accuracy comes from: relative to DER++, whose stored signals PKC consumes without modification, Class-IL BWT moves from -22.4 to -10.9 on CIFAR-10, from -53.9 to -26.3 on CIFAR-100 and from -64.6 to -27.6 on Tiny-ImageNet, while accuracy rises at the same time. Under Task-IL, PKC is also the most accurate method on all three datasets, with smaller margins ($+0.3$ over CILA on CIFAR-10, $+3.1$ and $+0.4$ over LODE-DER++ on CIFAR-100 and Tiny-ImageNet), and it has the best backward transfer in every column of Table 3.

Pretrained ViT in the online Si-Blurry protocol. Table 2 shows that PKC surpasses the prompt-based state of the art in the protocol these methods were designed for: with a 500-sample memory it improves A_{Last} over MISA by 7.3 points on Tiny-ImageNet and 11.4 on ImageNet-R, and with 2,000 samples by 3.9 and 8.6; it also has the lowest final forgetting at both budgets and trains in about half the wall-clock time of MISA and MVP, which run a prompt-selection pass in addition to the backbone (Tables 6 and 7 in Appendix D). The comparison with pointwise replay is the informative one: DER++ with the same 2,000-sample memory reaches an A_{Last} of 56.8 on Tiny-ImageNet against 80.3 for PKC.

Table 1: Task-IL and Class-IL overall accuracy with **ResNet** on CIFAR10, CIFAR-100, and Tiny-ImageNet (memory size = 500). '—' indicates unavailable or inapplicable results.

Method	CIFAR-10		CIFAR-100		Tiny-ImageNet	
	Class-IL	Task-IL	Class-IL	Task-IL	Class-IL	Task-IL
fine-tuning	19.62 ± 0.05	61.02 ± 3.33	9.29 ± 0.33	33.78 ± 0.42	7.92 ± 0.26	18.31 ± 0.68
Joint train	92.20 ± 0.15	98.31 ± 0.12	71.32 ± 0.21	91.31 ± 0.17	59.99 ± 0.19	82.04 ± 0.10
SI	19.48 ± 0.17	68.05 ± 5.91	9.41 ± 0.24	31.08 ± 1.65	6.58 ± 0.31	36.32 ± 0.13
LwF	19.61 ± 0.05	63.29 ± 2.35	9.70 ± 0.23	28.07 ± 1.96	8.46 ± 0.22	15.85 ± 0.58
NCL	19.53 ± 0.32	64.49 ± 4.06	8.12 ± 0.28	20.92 ± 2.32	7.56 ± 0.36	16.29 ± 0.87
UCB	—	79.28 ± 1.87	—	57.15 ± 1.67	—	—
HAT	—	92.56 ± 0.78	—	72.06 ± 0.50	—	—
A-GEM	22.67 ± 0.57	89.48 ± 1.45	9.30 ± 0.32	48.06 ± 0.57	8.06 ± 0.04	25.33 ± 0.49
GSS	49.73 ± 4.78	91.02 ± 1.57	13.60 ± 2.98	57.50 ± 1.93	—	—
HAL	41.79 ± 4.46	84.54 ± 2.36	9.05 ± 2.76	42.94 ± 1.80	—	—
oEWC	19.49 ± 0.12	64.31 ± 4.31	8.24 ± 0.21	21.2 ± 2.08	7.42 ± 0.31	15.19 ± 0.82
CPR(EWC)	19.61 ± 3.67	65.23 ± 3.87	8.42 ± 0.37	21.43 ± 2.57	7.67 ± 0.23	15.58 ± 0.91
GPM	—	—	—	72.48 ± 0.40	—	30.72 ± 0.27
ER	57.74 ± 0.27	93.61 ± 0.27	27.69 ± 0.58	71.56 ± 0.57	9.99 ± 0.29	48.64 ± 0.46
DER++	72.70 ± 1.36	93.88 ± 0.50	36.37 ± 0.85	75.64 ± 0.60	19.38 ± 1.41	51.91 ± 0.68
ER-ACE	71.83 ± 1.42	94.12 ± 0.61	37.05 ± 0.36	75.97 ± 0.69	20.43 ± 0.97	52.59 ± 0.75
LODE-DER++	75.45 ± 0.90	94.41 ± 0.22	38.95 ± 0.93	78.92 ± 0.67	22.01 ± 0.79	57.26 ± 0.63
CILA	76.03 ± 0.79	96.40 ± 0.21	—	—	20.64 ± 0.59	54.13 ± 0.72
FNC^2-HSD	75.51 ± 0.52	96.14 ± 0.25	40.25 ± 0.58	65.85 ± 0.44	20.31 ± 0.34	53.46 ± 0.59
CSReL-LODE-DER++	74.32 ± 1.23	—	41.96 ± 0.78	—	22.83 ± 0.23	—
PKC (Ours)	$\mathbf{76.45 \pm 0.68}$	$\mathbf{96.74 \pm 0.31}$	$\mathbf{45.16 \pm 0.56}$	$\mathbf{82.05 \pm 0.77}$	$\mathbf{25.56 \pm 0.46}$	$\mathbf{57.64 \pm 0.56}$

Table 2: Si-Blurry online continual learning with a **ViT-B/16** backbone on Tiny-ImageNet and ImageNet-R (five sessions; mean $\pm$ std over five seeds). A_{AUC} and A_{Last} in %; higher is better. Baseline numbers are those reported by Kang et al. (2025) under the same protocol; bold marks the best entry per memory budget.

Buffer	Method	Tiny-ImageNet		ImageNet-R	
		$A_{\mathrm{AUC}} \uparrow$	$A_{\mathrm{Last}} \uparrow$	$A_{\mathrm{AUC}} \uparrow$	$A_{\mathrm{Last}} \uparrow$
500	ER	59.46 ± 1.81	40.60 ± 2.71	40.31 ± 1.33	28.85 ± 1.43
	DER++	61.67 ± 1.19	46.03 ± 1.00	40.32 ± 1.08	31.53 ± 1.57
	ER-ACE	64.52 ± 0.78	56.82 ± 0.67	41.06 ± 1.32	36.59 ± 0.52
	RM	31.96 ± 0.80	7.43 ± 0.27	18.31 ± 1.09	4.14 ± 0.18
	CLIB	60.11 ± 1.53	48.97 ± 1.48	37.18 ± 1.52	29.51 ± 0.98
	MVP	76.52 ± 0.73	65.19 ± 0.58	49.07 ± 1.47	44.17 ± 1.72
	MISA	79.08 ± 0.60	69.91 ± 0.52	54.72 ± 1.15	47.48 ± 0.57
	PKC (Ours)	$\mathbf{81.99 \pm 0.82}$	$\mathbf{77.20 \pm 0.36}$	$\mathbf{54.89 \pm 2.82}$	$\mathbf{58.88 \pm 0.56}$
2,000	ER	66.75 ± 1.13	55.07 ± 1.28	45.74 ± 1.35	38.13 ± 0.32
	DER++	66.58 ± 0.88	56.81 ± 0.65	42.79 ± 1.31	36.06 ± 1.04
	ER-ACE	66.86 ± 0.84	58.40 ± 0.38	43.62 ± 1.31	40.49 ± 0.22
	RM	47.26 ± 1.13	44.55 ± 0.37	27.88 ± 1.29	24.25 ± 0.99
	CLIB	65.47 ± 0.76	56.87 ± 0.54	42.69 ± 1.30	35.43 ± 0.38
	MVP	80.67 ± 0.75	74.34 ± 0.32	52.47 ± 1.45	50.54 ± 2.08
	MISA	82.91 ± 0.47	76.41 ± 0.33	57.67 ± 0.72	53.62 ± 0.68
	PKC (Ours)	$\mathbf{83.60 \pm 0.86}$	$\mathbf{80.32 \pm 0.25}$	$\mathbf{57.80 \pm 2.56}$	$\mathbf{62.19 \pm 0.31}$

Table 3: **Backward Transfer** of different CL methods with memory size 500.

Method	CIFAR10		CIFAR100		Tiny-ImageNet	
	Class-IL	Task-IL	Class-IL	Task-IL	Class-IL	Task-IL
finetuning	-96.39 ± 0.12	-46.24 ± 2.12	-89.68 ± 0.96	-62.46 ± 0.78	-78.94 ± 0.81	-67.34 ± 0.79
AGEM	-94.01 ± 1.16	-14.26 ± 1.18	-88.50 ± 1.56	-45.43 ± 2.32	-78.03 ± 0.78	-59.28 ± 1.08
GSS	-62.88 ± 2.67	-7.73 ± 3.99	-82.17 ± 4.16	-33.98 ± 1.54	–	–
HAL	-62.21 ± 4.34	-5.41 ± 1.10	-49.29 ± 2.82	-13.60 ± 1.04	–	–
ER	-45.35 ± 0.07	-3.54 ± 0.35	-74.84 ± 1.38	-16.81 ± 0.97	-75.24 ± 0.76	-31.98 ± 1.35
DER++	-22.38 ± 4.41	-4.66 ± 1.15	-53.89 ± 1.85	-14.72 ± 0.96	-64.60 ± 0.56	-27.21 ± 1.23
ER-ACE	-13.64 ± 0.95	-3.28 ± 0.83	-39.51 ± 1.23	-14.57 ± 0.39	-46.07 ± 0.83	-28.35 ± 0.16
LODE-DER++	-16.37 ± 0.67	-2.93 ± 0.19	-53.23 ± 1.72	-15.24 ± 0.76	-55.89 ± 0.98	-19.13 ± 0.56
PKC (Ours)	$\mathbf{-10.91 \pm 1.16}$	$\mathbf{-1.32 \pm 0.37}$	$\mathbf{-26.32 \pm 1.04}$	$\mathbf{-9.05 \pm 0.45}$	$\mathbf{-27.59 \pm 3.40}$	$\mathbf{-19.06 \pm 0.53}$

4.2 Ablation Study

Table 4 removes one component at a time with ResNet-18 on CIFAR-100. Removing the fidelity term or the interior nodes of the path ($K{=}1$, which leaves only the two endpoints and is pointwise replay with a conditioner) roughly doubles forgetting and lowers accuracy by 2 to 5 points: fidelity along the interior of the path is what limits forgetting. Uniform node weights instead of Min-SNR reduce forgetting but cost 4.8 points of accuracy, because the belief end of the path is then supervised as hard as the label end, so the lower BWT of that row is bought with plasticity rather than earned. Removing the smoothness term costs about one point of accuracy and two points of backward transfer, while any $\lambda > 0$ in the range of Appendix C performs alike.

Table 4: Ablation with ResNet-18 on CIFAR-100 (Class-IL, buffer 500; mean $\pm$ std over three seeds). Each row changes one design choice from the default configuration of Table 1.

Variant	Acc. (%) ↑	BWT (%) ↑
PKC (default, $K{=}5$, Min-SNR, $\gamma{=}0.05$, $\lambda{=}0.005$)	45.16 ± 0.56	-26.32 ± 1.04
w/o $\mathcal{L}_{\text{fid}}$ ($\gamma{=}0$)	40.36 ± 0.87	-47.32 ± 1.02
w/o $\mathcal{L}_{\text{smooth}}$ ($\lambda{=}0$)	43.93 ± 0.64	-28.11 ± 0.79
$K{=}1$ (endpoints only)	42.81 ± 0.91	-42.94 ± 1.32
w/o Min-SNR weights ($w \equiv 1$)	40.36 ± 1.79	-19.36 ± 1.73

Sensitivity to λ. The smoothness weight is a robustness knob rather than the source of the gain: on CIFAR-100 accuracy varies by about 0.5 for λ between 0.001 and 0.1, and only removing the term entirely ($\lambda{=}0$) costs about one point (Appendix C). The path is smooth by parameterization, so the explicit penalty only guards against the degenerate solutions ruled out by Theorem 1; we keep $\lambda{=}0.005$ in all reported runs.

Efficiency Comparison Table 7 (Appendix D) compares wall-clock training time on Si-Blurry Tiny-ImageNet under the same memory budget: PKC trains in 40 minutes against 87 for MISA and 79 for MVP, so the gains of Table 2 come with less than half the training time. On ResNet-18 the only addition is the conditioner (0.22M parameters, used during training only) and $K{+}1$ evaluations of the linear head per replay batch; inference during test time has no additional computation cost.

5 Conclusion

We introduce pathwise knowledge consolidation, which advances replay from relying on two isolated targets to utilizing a corruption-indexed prediction path that connects each sample's label to its stored logits. A lightweight conditioner realizes this path on an unmodified backbone, incurring negligible training overhead and no additional inference cost during test time. Furthermore, our variational analysis demonstrates that the objective encourages low-curvature trajectories in function space, while extensive experiments confirm that our method achieves state-of-the-art accuracy among replay-based approaches in both Class-IL and Task-IL scenarios.

AI use statement

In this work, we used generative AI tools to aid and polish the writing. We have not used generative AI tools for research ideation, for generating synthetic data, or for proving mathematical claims; retrieval and discovery of related work. Coding and experiment execution were carried out by the authors. We have reviewed all AI-assisted text, every reported number was produced by our own experiments, and we take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

REFERENCES

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision*, pp. 139–154, 2018.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems 30*, 2019.

Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. Learning fast, learning slow: A general continual learning method based on complementary learning system. In *International Conference on Learning Representations*, 2022.

Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 8214–8223. IEEE, 2021.

Lorenzo Bonicelli, Matteo Boschini, Angelo Porrello, Concetto Spampinato, and Simone Calderara. On the effectiveness of lipschitz-driven rehearsal in continual learning. *Advances in Neural Information Processing Systems*, 35:31886–31901, 2022.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. In *International Conference on Learning Representations*, 2022.

Sungmin Cha, Hsiang Hsu, Taebaek Hwang, Flavio Calmon, and Taesup Moon. Cpr: Classifier-projection regularization for continual learning. In *International Conference on Learning Representations*, 2021.

Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision*, pp. 532–547, 2018.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *Proceedings of the International Conference on Learning Representations*, 2019a.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K. Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. Continual learning with tiny episodic memories. *https://arxiv.org/abs/1902.10486*, 2019b.

Arslan Chaudhry, Albert Gordo, Puneet K Dokania, Philip HS Torr, and David Lopez-Paz. Using hindsight to anchor past knowledge in continual learning. *Association for the Advancement of Artificial Intelligence*, 2021.

Trung-Anh Dang, Vincent Nguyen, Ngoc-Son Vu, and Christel Vrain. Memory-efficient continual learning with neural collapse contrastive. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 7950–7959. IEEE, 2025.

Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-SNR weighting strategy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7441–7451, 2023.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8340–8349, 2021.

Christian Henning, Maria Cervera, Francesco D'Angelo, Johannes Von Oswald, Regina Traber, Benjamin Ehret, Seijin Kobayashi, Benjamin F Grewe, and João Sacramento. Posterior meta-replay for continual learning. *Advances in Neural Information Processing Systems*, 34:14135–14149, 2021.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 831–839, 2019.

Ching-Yi Hung, Cheng-Hao Tu, Cheng-En Wu, Chien-Hung Chen, Yi-Ming Chan, and Chu-Song Chen. Compacting, picking and growing for unforgetting continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Zhiqi Kang, Liyuan Wang, Xingxing Zhang, and Karteek Alahari. Advancing prompt-based methods for replay-independent general continual learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ta-Chu Kao, Kristopher Jensen, Gido van de Ven, Alberto Bernacchia, and Guillaume Hennequin. Natural continual learning: success is a journey, not (just) a destination. *Advances in neural information processing systems*, 34:28067–28079, 2021.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114 (13):3521–3526, 2017.

Hyunseo Koh, Dahyun Kim, Jung-Woo Ha, and Jonghyun Choi. Online continual learning on class incremental blurry task configuration with anytime inference. In *International Conference on Learning Representations*, 2022.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical Report, University of Toronto*, 2009.

Alex Lewandowski, Michał Bortkiewicz, Saurabh Kumar, András György, Dale Schuurmans, Mateusz Ostaszewski, and Marlos C. Machado. Learning continually by spectral regularization. In *The Thirteenth International Conference on Learning Representations*, 2025.

Mei Li, Yuxiang Lu, Qinyan Dai, Suizhi Huang, Yue Ding, and Hongtao Lu. BECAME: Bayesian continual learning with adaptive model merging. In *Forty-second International Conference on Machine Learning*, 2025.

Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *International Conference on Machine Learning*, pp. 3925–3934. PMLR, 2019.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

Yan-Shuo Liang and Wu-Jun Li. Loss decoupling for task-agnostic continual learning. *Advances in Neural Information Processing Systems*, 36, 2023.

Chenxi Liu, Zhenyi Wang, Tianyi Xiong, Ruibo Chen, Yihan Wu, Junfeng Guo, and Heng Huang. Few-shot class incremental learning with attention-aware self-adaptive prompt. In *European Conference on Computer Vision*, 2024.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Jun-Yeong Moon, Keon-Hee Park, Jung Uk Kim, and Gyeong-Moon Park. Online class incremental learning on stochastic blurry task boundary via mask and visual prompt tuning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11731–11741, 2023.

Cuong V. Nguyen, Yingzhen Li, Thang D. Bui, and Richard E. Turner. Variational continual learning. In *International Conference on Learning Representations*, 2018.

Pingbo Pan, Siddharth Swaroop, Alexander Immer, Runa Eschenhagen, Richard Turner, and Mohammad Emtiyaz E Khan. Continual deep learning by functional regularisation of memorable past. *Advances in Neural Information Processing Systems*, 33:4453–4464, 2020.

Quang Pham, Chenghao Liu, and Steven HOI. Dualnet: Continual learning, fast and slow. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021.

Roger Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285, 1990.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, , and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *International Conference on Learning Representations*, 2019.

Hippolyt Ritter, Aleksandar Botev, and David Barber. Online structured laplace approximations for overcoming catastrophic forgetting. *Advances in Neural Information Processing Systems*, 31, 2018.

Tim GJ Rudner, Freddie Bickford Smith, Qixuan Feng, Yee Whye Teh, and Yarin Gal. Continual learning via sequential function-space variational inference. In *International Conference on Machine Learning*, pp. 18871–18887. PMLR, 2022.

Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations*, 2021.

Fahad Sarfraz, Elahe Arani, and Bahram Zonooz. Error sensitivity modulation based experience replay: Mitigating abrupt representation drift in continual learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Jonathan Schwarz, Jelena Luketina, Wojciech M. Czarnecki, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress and compress: A scalable framework for continual learning. In *Proceedings of the International Conference on Machine Learning*, 2018.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International conference on machine learning*, pp. 4548–4557. PMLR, 2018.

Michalis K. Titsias, Jonathan Schwarz, Alexander G. de G. Matthews, Razvan Pascanu, and Yee Whye Teh. Functional regularisation for continual learning with gaussian processes. In *International Conference on Learning Representations*, 2020.

Ruilin Tong, Yuhang Liu, Javen Qinfeng Shi, and Dong Gong. Coreset selection via reducible loss in continual learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Eli Verwimp, Matthias De Lange, and Tinne Tuytelaars. Rehearsal revealed: The limits and merits of revisiting samples in continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9385–9394, 2021.

Zhenyi Wang, Li Shen, Le Fang, Qiuling Suo, Tiehang Duan, and Mingchen Gao. Improving task-free continual learning by distributionally robust memory evolution. In *International Conference on Machine Learning*, pp. 22985–22998. PMLR, 2022a.

Zhenyi Wang, Yan Li, Li Shen, and Heng Huang. A unified and general framework for continual learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European conference on computer vision*, pp. 631–648. Springer, 2022b.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 139–149, 2022c.

Zifeng Wang, Zheng Zhan, Yifan Gong, Yucai Shao, Stratis Ioannidis, Yanzhi Wang, and Jennifer Dy. Dualhsic: Hsic-bottleneck and alignment for continual learning. In *International Conference on Machine Learning*, 2023.

Yichen Wen, Zhiquan Tan, Kaipeng Zheng, Chuanlong Xie, and Weiran Huang. Provable contrastive continual learning. In *International Conference on Machine Learning*, pp. 52819–52838. PMLR, 2024.

Jiayu Wu, Qixiang Zhang, and Guoxi Xu. Tiny imagenet challenge. *Technical report*, 2017.

Enneng Yang, Li Shen, Zhenyi Wang, Tongliang Liu, and Guibing Guo. An efficient dataset condensation plugin and its application to continual learning. *Advances in Neural Information Processing Systems*, 36, 2023.

Ruilong Yu, Mingyan Liu, Fei Ye, Adrian G. Bors, Rongyao Hu, Jingling sun, and shijie zhou. Learning expandable and adaptable representations for continual learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pp. 3987–3995. PMLR, 2017.

REPRODUCIBILITY STATEMENT

All experiments run in the Mammoth framework with the standard task splits; hyperparameters are listed in Section 4 and Appendix B, the algorithm is given in full in Algorithm 1, and proofs are in Appendix A. Code and configuration files will be released.

A PROOFS

Proposition (Proposition 1, restated)**.** Assume that for every (x, y), the map $t \mapsto \phi_\theta(t; x, y)$ is continuously differentiable on $[0, 1]$, and that $t \mapsto \ell(\phi_\theta(t; x, y), y)$ is continuous. Then, as $\max_k \Delta t_k \to 0$, the discrete objective $\mathcal{J}_K(\theta)$ converges to

$$\mathcal{J}(\theta) = \mathbb{E}_{(x,y)} \left[\int_0^1 \ell\big(\phi_\theta(t; x, y), y\big)\, dt + \lambda \int_0^1 \|\partial_t \phi_\theta(t; x, y)\|^2\, dt \right].$$

Proof. The first term in $\mathcal{J}_K(\theta)$ is a standard Riemann sum for

$$\int_0^1 \ell\big(\phi_\theta(t; x, y), y\big)\, dt.$$

For the second term, the continuous differentiability of ϕ_θ implies that the finite-difference quotient

$$\frac{\phi_\theta(t_{k+1}; x, y) - \phi_\theta(t_k; x, y)}{\Delta t_k}$$

converges to $\partial_t \phi_\theta(t; x, y)$ as $\Delta t_k \to 0$. Therefore,

$$\sum_{k=0}^{K-1} \frac{\|\phi_\theta(t_{k+1}; x, y) - \phi_\theta(t_k; x, y)\|^2}{\Delta t_k}$$

is a Riemann sum for

$$\int_0^1 \|\partial_t \phi_\theta(t; x, y)\|^2\, dt.$$

Taking the expectation over (x, y) gives the result, by dominated convergence, assuming the integrands are dominated by an integrable function. □

For a fixed pair (x, y), define the path energy

$$\mathcal{E}[\phi] = \int_0^1 \big(\ell(\phi(t), y) + \lambda \|\phi'(t)\|^2\big)\, dt.$$

The next result gives the stationary condition for this functional.

Proposition (Proposition 2, restated)**.** Suppose $\ell(\cdot, y)$ is differentiable. If $\phi^\star$ is a stationary point of $\mathcal{E}$, then $\phi^\star$ satisfies

$$\nabla_\phi \ell(\phi^\star(t), y) - 2\lambda \phi^{\star\prime\prime}(t) = 0, \qquad t \in (0, 1).$$

Proof. Let

$$L(\phi, \phi', t) = \ell(\phi(t), y) + \lambda \|\phi'(t)\|^2.$$

Applying the Euler–Lagrange equation,

$$\frac{\partial L}{\partial \phi} - \frac{d}{dt}\left(\frac{\partial L}{\partial \phi'}\right) = 0,$$

we obtain

$$\nabla_\phi \ell(\phi(t), y) - \frac{d}{dt}\big(2\lambda \phi'(t)\big) = 0.$$

This simplifies to

$$\nabla_\phi \ell(\phi(t), y) - 2\lambda \phi''(t) = 0,$$

which proves the claim. □

The previous proposition gives a useful interpretation. The gradient term $\nabla_\phi \ell(\phi(t), y)$ pulls the trajectory toward the target, while the second-derivative term penalizes excessive curvature along the corruption path. Thus, the learned trajectory is encouraged to be both accurate and smooth.

The next theorem makes the role of the regularizer more explicit.

Theorem (Theorem 1, restated)**.** Assume that $\phi_\theta(\cdot; x, y)$ is absolutely continuous on $[0, 1]$. Then for any $s, t \in [0, 1]$,

$$\|\phi_\theta(t; x, y) - \phi_\theta(s; x, y)\| \le |t - s|^{1/2} \left(\int_0^1 \|\partial_\tau \phi_\theta(\tau; x, y)\|^2 d\tau \right)^{1/2}.$$

Proof. By the fundamental theorem of calculus,

$$\phi_\theta(t; x, y) - \phi_\theta(s; x, y) = \int_s^t \partial_\tau \phi_\theta(\tau; x, y)\, d\tau.$$

Taking norms and applying the Cauchy–Schwarz inequality gives

$$\|\phi_\theta(t; x, y) - \phi_\theta(s; x, y)\| \le \int_s^t \|\partial_\tau \phi_\theta(\tau; x, y)\|\, d\tau \le |t - s|^{1/2} \left(\int_s^t \|\partial_\tau \phi_\theta(\tau; x, y)\|^2 d\tau \right)^{1/2}.$$

Since

$$\int_s^t \|\partial_\tau \phi_\theta(\tau; x, y)\|^2 d\tau \le \int_0^1 \|\partial_\tau \phi_\theta(\tau; x, y)\|^2 d\tau,$$

the stated bound follows. □

B Implementation Details

From-scratch experiments use ResNet-18 trained with SGD (learning rate 0.1, momentum 0.9), batch size 32, 50 epochs per task and a 500-sample reservoir buffer, following the Mammoth protocol; K=5, s=3, T=5, p_{drop}=0.2, γ=0.05, λ=0.005 and gradient clipping at norm 5 on all datasets. The replay-side weights are inherited from DER++ and LODE and were set per dataset on the first tasks: CIFAR-100 uses β=0.1 with the anchor averaged over classes and ρ=0.1; CIFAR-10 uses β=0.01, the teacher term restricted to past classes and ρ=0; Tiny-ImageNet uses β=0.25, ρ=0.1 and a task-boundary re-scoring of buffered samples.

Choice of the schedule constants. The path length K, the endpoint sharpness s, the temperature T and the drop rate p_{drop} listed above were chosen once during development by a small manual search on a single benchmark, then frozen and used unchanged in all other experiments; none of them was re-tuned on any other dataset, so the results of Table 1 are not obtained by per-dataset tuning. The distillation temperature κ takes the standard value used in knowledge distillation.

Min-SNR node weights. With $\mathrm{SNR}(\tau) = (1 - \tau + \epsilon)/(\tau + \epsilon)$, $\epsilon = 0.05$, clipped at 5 and normalized to mean one, the weights for K=5 are $w = [2.76,\ 1.88,\ 0.80,\ 0.38,\ 0.16,\ 0.03]$.

Si-Blurry configuration. The ViT experiments in Tables 2, 6 and 7 follow the protocol of Moon et al. (2023) and Kang et al. (2025): five sessions, stochastic blurry task boundaries with recurring classes, a single pass over the stream, ViT-B/16 pretrained on ImageNet-21k, memory budgets of 500 and 2,000 samples, five seeds; baseline numbers are those reported by Kang et al. (2025). The backbone is kept frozen and the classifier head and the conditioner are trained with the objective of Section 3, using balanced replay from a reservoir buffer, AdamW (learning rate 10^{-3}, weight decay 0.05), stream batches of 32 samples plus 32 replayed samples, and α=0.5, β=0.005, γ=0.05, λ=0.005, s=8. The evaluation protocol is identical to that of MVP and MISA: five sessions with a disjoint-class ratio of N=50% and a blurry-sample ratio of M=10%, stochastic splits generated from seeds 1–5, three online iterations per stream batch, and anytime evaluation every 1,000 stream samples.

C Sensitivity to the Smoothness Weight

Table 5 varies λ around its default of 0.005 with ResNet-18 on CIFAR-100. For λ between 0.001 and 0.1 accuracy varies by about half a point and backward transfer by less than one, while removing the term entirely (λ=0) costs about one point of accuracy and two points of backward transfer. This is what the construction predicts: the corrupted labels follow a geodesic, the level embedding is sinusoidal and the conditioner is a smooth network, so $z(\tau)$ is smooth in τ by parameterization, and fidelity at every node already keeps neighbouring predictions close; the penalty of Eq. 6 guards against the degenerate solutions that Theorem 1 rules out. We keep λ=0.005 in all reported runs for consistency.

Table 5: Sensitivity to the smoothness weight λ with ResNet-18 on CIFAR-100 (Class-IL, buffer 500; mean $\pm$ std over three seeds).

λ	0	0.001	**0.005** (default)	0.02	0.1
Acc. (%) ↑	43.93 ± 0.64	44.87 ± 1.02	45.16 ± 0.56	44.72 ± 1.00	44.61 ± 0.51
BWT (%) ↑	−28.11 ± 0.79	−26.43 ± 0.42	−26.32 ± 1.04	−26.49 ± 0.66	−26.97 ± 1.01

D Additional Si-Blurry Results

Table 6 reports final forgetting and Table 7 wall-clock training time for the Si-Blurry experiments of Table 2.

Table 6: Final forgetting F_{Last} (%) on five-session Si-Blurry (mean $\pm$ std over five seeds; lower is better). Bold marks the best entry per memory budget.

Method	**Buffer**	**Tiny-ImageNet**	**ImageNet-R**
MISA	0	14.77 ± 2.77	19.91 ± 4.21
MVP	0	50.01 ± 6.17	40.48 ± 4.30
MISA	500	26.78 ± 6.38	25.97 ± 5.81
MVP	500	31.43 ± 6.68	26.48 ± 3.95
PKC (Ours)	500	**14.57 ± 2.94**	**14.57 ± 2.12**
MISA	2,000	16.43 ± 3.72	17.56 ± 3.68
MVP	2,000	18.56 ± 4.38	18.33 ± 2.27
PKC (Ours)	2,000	**9.80 ± 2.09**	**8.88 ± 2.02**

Table 7: Wall-clock training time (minutes) on Si-Blurry Tiny-ImageNet (mean $\pm$ std over five seeds; lower is better).

Buffer	**MISA**	**MVP**	**PKC (Ours)**
500	86.99 ± 1.69	79.38 ± 0.72	**40.27 ± 1.51**
2,000	84.67 ± 0.38	79.27 ± 1.35	**42.41 ± 1.35**