# NOSTALGIA: CONTINUAL LEARNING THROUGH CONSTRAINED OPTIMISATION IN SMALL MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Pre-trained transformer models are typically adapted to downstream tasks via supervised fine-tuning, but sequential adaptation often leads to *catastrophic forgetting*, where performance on earlier tasks degrades. Existing mitigation strategies commonly rely on data replay, architectural modifications, or heuristic regularization, limiting their scalability. We introduce **Nostalgia**, a past-sample-free, optimization-based method for continual supervised fine-tuning that mitigates forgetting by constraining parameter updates to directions that minimally interfere with previously learned tasks. Our approach is grounded in a local quadratic approximation of task losses and operationalizes this principle through a tractable low-rank approximation of the average Hessian, enabling efficient projection-based updates compatible with modern parameter-efficient fine-tuning.

## 1 INTRODUCTION

Sequential fine-tuning is a common way to adapt a shared model to a stream of supervised tasks when retraining jointly on all tasks is undesirable or when past data cannot be retained. This setting is relevant to compact models because their smaller memory and compute requirements make repeated local adaptation practical, while a limited parameter budget can leave less redundancy for accommodating successive tasks. The present work studies this setting at a controlled experimental scale: a small convolutional network trained on a three-task vision sequence. We use this setting to test the proposed optimization mechanism, rather than to claim validation on the broader range of sub-billion-parameter transformers.

This incremental adaptation regime can produce catastrophic forgetting [McCloskey & Cohen (1989); French (1999)]: as the model is fine-tuned on a new task, gradient updates may overwrite parameters important to earlier tasks. Replay-based methods such as DER [Buzzega et al. (2020)] reduce forgetting by retaining past samples, but this is unsuitable when storage or data-retention constraints apply. Architectural isolation approaches allocate task-specific capacity, causing memory use to grow with the number of tasks. Regularization methods such as EWC [Kirkpatrick et al. (2017)] and L2-SP [Li et al. (2020)] instead penalize parameter movement. EWC is attractive because it is past-sample-free during subsequent tasks, but its diagonal Fisher approximation discards interactions between parameters. This motivates a regularizer that retains EWC's soft constraint while using a structured estimate of past-task curvature. Here, *past-sample-free* means that samples from previously learned tasks are not retained.

A principled curvature model follows from local continual learning [Lanzillotta et al. (2025)]: under a local quadratic approximation, updates in the null space of the average past-task Hessian incur no forgetting. Because the full Hessian is intractable, we utilise the fact that most of the eigen-vectors are close to zero, and hence can be ignored in the spectral decomposition.

We introduce two methods **Nostalgia-EWC** and **Nostalgia**, which use this low-rank curvature memory in an EWC-style quadratic anchor and as hard projection respectively. Both require $O(nk)$ storage and update cost for $n$ trainable parameters and rank $k$ of the spectral decomposition we consider, without retaining past examples. We evaluate the methods on a ResNet-10 three-task sequence and a ViT-B/16 with LoRA on subsampled six DomainNet domains. Across the reported runs, the soft anchor provides stronger retention, while hard projection preserves greater plasticity.

Our contributions are as follows:



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.

1. **Low-rank curvature projection and regularization**: We introduce Nostalgia and Nostalgia-EWC. While Nostalgia acts as a past-sample-free hard projection based method, Nostalgia-EWC is a past-sample-free quadratic regularizer that replaces EWC's diagonal Fisher with a rank-$k$ average-Hessian memory, preserving cross-parameter curvature structure at $O(nk)$ storage and update cost.
2. **Efficient spectral memory**: We also provide an efficient method to estimate and accumulate dominant curvature directions using Hessian Vector Products and factored updates, without materializing a full parameter-space matrix or retaining past data.
3. **Evaluation across architectures**: In the ResNet-10 experiment [He et al. (2016)], Nostalgia-EWC provides the strongest earlier-task retention among the evaluated methods. On DomainNet with ViT-B/16 and LoRA [Dosovitskiy et al. (2021); Hu et al. (2022)], it achieves the lowest average forgetting and highest average final accuracy among Nostalgia, GPM, and Nostalgia-EWC in the reported single-seed runs, while exposing the trade-off between retention and plasticity.

## 2 Related Works

**Regularization and Parameter-Constrained Approaches:** A classical line of work introduces explicit regularization terms to keep fine-tuned parameters close to their pre-trained initialization. Methods such as L2-SP and Elastic Weight Consolidation (EWC) [Kirkpatrick et al. (2017); Kang et al. (2019)] constrain updates using $\mathcal{L}_2$ or Fisher-weighted penalties, discouraging deviation from important pre-trained weights. Other extensions—such as Selective Projection Decay or Laplace-based priors—attempt to model parameter importance more precisely. While these methods are conceptually simple, their performance is highly sensitive to regularization strength, and they often underperform on large-scale multimodal models, where the correspondence between parameter distance and functional similarity is weak. Synaptic Intelligence (SI) estimates parameter importance from the contribution of individual parameters to learning over a task [Zenke et al. (2017)]. The term below represents the loss function $\mathcal{L}_{l2sp}(\theta)$ for a parameter $\theta$ as the loss function of the task, $\mathcal{L}_{task}(\theta)$, and a regulariser representing the *l2-norm* based distance from $\theta_0$ the parameter weights after pre-training:

$$\mathcal{L}_{l2sp}(\theta) = \mathcal{L}_{task}(\theta) + \frac{\lambda}{2}||\theta - \theta_0||_2^2$$

A more selective formulation is *Elastic Weight Consolidation(EWC)*, which estimates the importance of each parameter based on the Fisher Information Matrix $F$ computed during pretraining. The fine-tuning loss is augmented as:

$$\mathcal{L}_{\text{EWC}}(\theta) = \mathcal{L}_{\text{task}}(\theta) + \frac{\lambda}{2}\sum_i F_i(\theta_i - \theta_{0,i})^2,$$

**Parameter Isolation Methods**: Adapters [Houlsby et al. (2019)] introduce small task-specific modules while keeping most backbone parameters frozen. Variants such as AdaptFormer and AdapterFusion improve their placement or combine independently trained adapters [Chen et al. (2022); Pfeiffer et al. (2021)]. HALRP [Li et al. (2024)] instead uses Hessian information to represent task-specific parameter changes through low-rank transformations. These methods reduce interference by allocating separate capacity, but their storage generally grows as new task-specific components are added.

**Rehearsal Methods**: Rehearsal-based strategies mitigate catastrophic forgetting by replaying samples from previous tasks during new task training. In language model SFT, this involves mixing pretraining data with downstream batches to preserve general knowledge and prevent domain overfitting. Approaches vary in buffer maintenance, sampling balance, and selective replay, offering a scalable, practical alternative to costly second-order or parameter-constrained methods for continual learning and SFT retention. Popular methods for the same are Dark Experience Replay (DER, DER++), Meta-Experience Replay (MER), and Hindsight Anchor Learning [Boschini et al. (2023); Buzzega et al. (2020); Riemer et al. (2018); Chaudhry et al. (2021)].

**Projection based methods**: GPM [Saha et al. (2021)] mitigates catastrophic forgetting by constraining updates for new tasks to lie in the orthogonal complement of gradient subspaces deemed important for previously learned tasks. GPM identifies these subspaces from the network's task-specific

representations using singular value decomposition (SVD), and stores their bases as a memory for subsequent gradient projection. While our method similarly employs projection-based constraints on parameter updates, the two methods differ in how the protected subspace is constructed. GPM identifies important directions through the representational structure of past-task activations, relying on the relationship between input and gradient spaces

**Ensembling**: [Kleiman et al. (2025); Li et al. (2025); Qiu et al. (2025)] Weight interpolation ("model soups") averages parameters across fine-tuned and pre-trained models, often improving robustness without retraining. Output ensembling aggregates predictions from different checkpoints to balance specialization and generality. Although effective in mid-scale settings (e.g., ViT, CLIP, LLaMA-7B), these techniques scale poorly to frontier models due to memory and alignment constraints. These methods have shown resilience even in other modalities such as Language.

## 3 Problem Setup: SFT for Continual Learning

Our problem considers the setting of continual learning as the sequential acquisition of multiple supervised tasks $\tau_0, \tau_1, \tau_2, \ldots, \tau_T$, where the data for task $\tau_t$ is denoted $D_t = \{(x, y) \mid x \in \mathcal{X}_t, y \in \mathcal{Y}_t\}$. $\tau_0$ is considered as the pre-training task. We define forgetting for task $\tau_t$ as:

$$E_t(\theta) = L_t(\theta) - L_t(\theta_t) \tag{1}$$

where $L_t(\theta)$ is the loss function for $\tau_t$ evaluated for model parameters $\theta$, and $\theta_t$ are the parameters at the completion of learning task $\tau_t$. The goal of continual learning is to incrementally update the model on each new task $\tau_t$ using only $D_t$ and possibly an external memory $M_t$ (e.g., a buffer), while retaining performance on previously learned tasks. Performance is measured by minimizing both forgetting $E_t(\theta)$ and the following multi-task loss:

$$L_t^{MT}(\theta) = \frac{1}{t}\sum_{i=1}^{t} L_i(\theta) \tag{2}$$

The continual learning objective is to find update rules so that, after each step, the updated parameters approximately minimize the multi-task loss using only current task data, $D_t$ and memory, $M_t$:

$$\min_{\theta_t} \; obj_t(D_t, M_t) \approx L_t^{MT}(\theta_t) \tag{3}$$

The central challenge is that direct access to past task data $D_o$, for $o < t$, is unavailable. Thus, the algorithm must approximate the true multi-task loss with whatever information is stored or accessible. Previous work addresses this by using polynomial approximations for the loss function near each task optimum. Empirically and theoretically, loss surfaces tend to be well-behaved and locally convex near minima, so quadratic approximations are widely adopted:

$$\hat{L}_t(\theta) \approx L_t(\theta_t) + (\theta - \theta_t)^T \nabla L_t(\theta_t) + \frac{1}{2}(\theta - \theta_t)^T H_t(\theta_t)(\theta - \theta_t) \tag{4}$$

where $H_t(\theta_t)$ is the Hessian of $L_t$ evaluated at $\theta_t$. This lets us redefine the forgetting term as

$$E_t(\theta) = (\theta - \theta_t)^T \nabla L_t(\theta_t) + \frac{1}{2}(\theta - \theta_t)^T H_t(\theta_t)(\theta - \theta_t) \tag{5}$$

Lanzillotta et al. (2025) primarily classify continual learning algorithms into two categories — local and global — based on their approach to multi-task loss approximation:

**Definition 3.1** (Local and global task loss approximations). Let I(X; Y) denote the mutual information of the pair of random variables (X, Y). We say that the task loss's second order polynomial approximation $\hat{L}_t(\theta)$ is *local* when $I(\hat{L}_t(\theta); \theta_t) > 0 \; \forall \; \theta \in \Theta$, and that it is global when $I(\hat{L}(\theta); \theta_t) = 0 \; \forall \; \theta \in \Theta - \{\theta_t\}$.

Local algorithms, however, do provide a strong theoretical justification for searching the space from which we can get zero forgetting, as stated in the following lemma [Lanzillotta et al. (2025)]:

**Lemma** [Lanzillotta et al. (2025)]: *(Optimal quadratic local continual learning) For any continual learning algorithm producing a sequence of parameters $\theta_1, \theta_2, \ldots \theta_t$ such that $\theta_i$ is the local minima*

*of $L_i$ and $\sup_{\theta_i,\theta_k} ||\theta_i - \theta_k||^3 < \epsilon$, the following relationship holds:*

$$E(1), \dots E(t-1) = 0 \implies E(t) = \frac{1}{2}\Delta_t^T \left( \frac{1}{t} \sum_{j=1}^{t-1} H_j^* \right) \Delta_t \geq 0$$

*Moreover, if $E(1), \dots E(t-1) = 0$ the optimal learning objective for task t is:*

$$\min_{\Delta_t \in \Theta} L_t(\theta_{t-1} + \Delta_t) \quad s.t. \quad \Delta_t^T \left( \frac{1}{t} \sum_{j=1}^{t-1} H_j^* \right) \Delta_t = 0$$

Note that the constraints described through the lemma force the update steps to lie in the *null-space* of the average *Hessian matrix*. Intuitively, when a quadratic approximation to the loss is accurate enough, and each task solution is a local minimum, the parameter updates must be taken along directions where the multi-task loss landscape is (absolutely) flat to prevent forgetting.

# 4 METHOD: NOSTALGIA

We consider the standard continual learning setup in which a model with parameters $\theta \in \mathbb{R}^n$ is trained sequentially on tasks $\tau_1, \dots, \tau_T$. Following prior work [Lanzillotta et al. (2025)], we approximate each past task loss $L_{\tau_i}$ by a second-order Taylor expansion around its solution $\theta^*_{\tau_i}$. Within this approximation, the average Hessian identifies how parameter movement increases past-task loss. Formally, after completing tasks $1, \dots, t-1$, zero quadratic forgetting requires an update $\Delta_t$ satisfying

$$\Delta_t^\top \bar{H}_{t-1} \Delta_t = 0, \text{ where } \bar{H}_{t-1} = \frac{1}{t-1} \sum_{i=1}^{t-1} H_{\tau_i}(\theta^*_{\tau_i}). \tag{6}$$

This condition suggests a direct strategy: prevent each new-task update from entering directions of positive past-task curvature. We call this hard-constraint method **Nostalgia**. We first construct a compact spectral memory of those directions, then use it to project gradients. Finally, we show how the limitations of this hard constraint lead naturally to **Nostalgia-EWC**, a soft curvature-weighted relaxation.

## 4.1 LOW-RANK HESSIAN APPROXIMATION

Storing or inverting an $n \times n$ Hessian is infeasible for modern neural networks. Empirically, however, task Hessians are effectively low-rank: most eigenvalues are near zero and only a small subspace carries significant curvature [Sagun et al. (2017); Papyan (2019); Ghorbani et al. (2019); Li et al. (2024)]. We therefore approximate the average Hessian by its top-$k$ eigendecomposition,

$$\bar{H}_{t-1} \approx Q_{t-1} \Lambda_{t-1} Q_{t-1}^\top, \tag{7}$$

where $Q_{t-1} \in \mathbb{R}^{n \times k}$ has orthonormal columns and $\Lambda_{t-1} \in \mathbb{R}_+^{k \times k}$ is diagonal and $k$ is a hyper-parameter that we use to represent the number of eigen-values that we consider. The pair $(Q_{t-1}, \Lambda_{t-1})$ forms the shared spectral memory used by both Nostalgia and Nostalgia-EWC.

**Single-task eigenspace estimation:** For a completed task $\tau$, we estimate its dominant eigenspace without forming the Hessian explicitly. Over $E_a$ accumulation rounds, a Lanczos routine driven by Hessian–vector products returns a stochastic rank-$k$ approximation

$$H_\tau^{(e)} \approx Q_\tau^{(e)} \Lambda_\tau^{(e)} \big(Q_\tau^{(e)}\big)^\top. \tag{8}$$

We keep only the non-negative spectral part and form the weighted factor $F_\tau^{(e)} = Q_\tau^{(e)} \big(\Lambda_\tau^{(e)}\big)_+^{1/2}$. Concatenating and renormalizing across rounds gives

$$F_\tau^{\text{loc}} = \frac{1}{\sqrt{E_a}} \Big[ F_\tau^{(1)} \ \cdots \ F_\tau^{(E_a)} \Big], \tag{9}$$

so that $F_\tau^{\text{loc}} (F_\tau^{\text{loc}})^\top \approx \frac{1}{E_a} \sum_{e=1}^{E_a} H_\tau^{(e)}$. In a distributed setting, workers never exchange parameter-space vectors; they instead average the small Gram matrix $G_\tau = (F_\tau^{\text{loc}})^\top F_\tau^{\text{loc}}$, compute its leading

**Algorithm 1** NOSTALGIA sequential training

**Input:** $\theta_0$, tasks $\{\tau_t\}_{t=1}^{T}$, rank $k$, regularization strength $\lambda$
**Initialize:** $(Q_0, \Lambda_0) \leftarrow (\varnothing, \varnothing)$, $\theta^* \leftarrow \theta_0$
**for** $t = 1$ **to** $T$ **do**
  **for** batches $B \subset D_{\tau_t}$ **do**
    $g_t \leftarrow \nabla_\theta \ell_t(\theta; B)$
    **if** $Q_{t-1} \neq \varnothing$ **then**
      $g_t \leftarrow g_t + \lambda Q_{t-1}\Lambda_{t-1}Q_{t-1}^\top(\theta - \theta^*)$
    **end if**
    $\theta \leftarrow \theta - \eta_t g_t$
  **end for**
  $\theta^* \leftarrow \theta$
  $(Q_{\text{new}}, \Lambda_{\text{new}}) \leftarrow$ SINGLEDOMAINEIGENSPACE$(\tau_t, k)$
  $(Q_t, \Lambda_t) \leftarrow$ ACCUMULATEPASTDOMAINS$(Q_{t-1}, \Lambda_{t-1}, Q_{\text{new}}, \Lambda_{\text{new}}, t, k)$
**end for**

eigenpairs $G_\tau V_\tau = V_\tau \Sigma_\tau$, and reconstruct a local orthonormal basis $Q_\tau = F_\tau^{\text{loc}} V_\tau \Sigma_\tau^{-1/2}$ with eigenvalues $\Lambda_\tau = \Sigma_\tau$.

**Accumulating curvature across tasks:** After task $t$ finishes, we rebuild a memory $(Q_{\text{mem}}, \Lambda_{\text{mem}})$ that approximates the running average Hessian $\bar{H}_t = \frac{t-1}{t}\bar{H}_{t-1} + \frac{1}{t}H_{\tau_t}$. Given the previous memory $Q_{\text{mem}}\Lambda_{\text{mem}}Q_{\text{mem}}^\top$ and the new task summary $Q_t\Lambda_t Q_t^\top$, we form weighted factors

$$F_{\text{old}} = \sqrt{\frac{t-1}{t}}\, Q_{\text{mem}}\Lambda_{\text{mem}}^{1/2}, \qquad F_{\text{new}} = \sqrt{\frac{1}{t}}\, Q_t\Lambda_t^{1/2}, \tag{10}$$

concatenate $F = [F_{\text{old}}\ F_{\text{new}}]$, and recover the top-$k$ eigenspace of $FF^\top$. This merging is performed with a stable factor-SVD procedure (Algorithm 2) and avoids ever materializing an $n \times n$ matrix.

**Nostalgia: enforcing the null-space constraint:** Nostalgia implements equation 6 as directly as the low-rank approximation permits. During task $t$, it projects each raw gradient $g_t$ onto the orthogonal complement of the retained curvature basis:

$$\tilde{g}_t = \big(I - Q_{t-1}Q_{t-1}^\top\big)g_t. \tag{11}$$

Because $Q_{t-1}$ is fixed while task $t$ is learned, the accumulated displacement remains orthogonal to every protected direction. Nostalgia therefore offers a simple, past-sample-free approximation to null-space training: past examples are replaced by the spectral memory, and every optimization step is restricted before it is applied.

**From hard projection to Nostalgia-EWC:** However the limitations of using hard projections in non-convex and high dimensional loss landscapes motivate replacing the hard constraint with a graded penalty. Standard EWC [Kirkpatrick et al. (2017)] anchors parameters using a diagonal Fisher importance estimate. Nostalgia-EWC retains the same principle but substitutes the accumulated low-rank Hessian memory, thereby preserving interactions among parameters and weighting each protected direction by its estimated curvature. For task $t$, we optimize the current-task loss together with

$$R_{\text{NE}}(\theta) = \frac{\lambda}{2}\,(\theta - \theta^*)^\top Q_{t-1}\Lambda_{t-1}Q_{t-1}^\top(\theta - \theta^*), \tag{12}$$

where $\theta^*$ is the solution after the most recently completed task. The corresponding gradient contribution is

$$g_{\text{NE}} = \lambda\, Q_{t-1}\Lambda_{t-1}Q_{t-1}^\top(\theta - \theta^*). \tag{13}$$

Nostalgia-EWC thus softens Nostalgia in two ways: movement in the protected subspace remains possible, and its cost is scaled by the estimated curvature of each direction. The coefficient $\lambda$ explicitly controls the resulting retention–plasticity trade-off. Algorithm 1 summarizes sequential training with this soft objective; both variants use the same curvature-accumulation procedure in Algorithm 2.

**Algorithm 2** ACCUMULATEPASTDOMAINS

**Input:** previous memory $(Q_{t-1}, \Lambda_{t-1})$, new-task summary $(Q_{\text{new}}, \Lambda_{\text{new}})$, task index $t$, rank $k$
**Output:** Hessian memory $(Q_{\text{mem}}, \Lambda_{\text{mem}})$
**if** $t = 1$ **then**
  $(Q_{\text{mem}}, \Lambda_{\text{mem}}) \leftarrow (Q_{\text{new}}, \Lambda_{\text{new}})$
**else**
  $F_{\text{old}} \leftarrow \sqrt{\frac{t-1}{t}}\, Q_{t-1}\Lambda_{t-1}^{1/2}$
  $F_{\text{new}} \leftarrow \sqrt{\frac{1}{t}}\, Q_{\text{new}}\Lambda_{\text{new}}^{1/2}$
  $F \leftarrow [F_{\text{old}}\ F_{\text{new}}]$
  $(Q_{\text{mem}}, \Lambda_{\text{mem}}) \leftarrow$ TOPEIGENPAIRSFROMFACTOR$(F, k)$
**end if**
**return** $(Q_{\text{mem}}, \Lambda_{\text{mem}})$

### 4.2 COMPLEXITY

For $n$ trainable parameters and memory rank $k$, Nostalgia stores $O(nk)$ values in $Q$ and requires $O(nk)$ operations per projected update. Nostalgia-EWC additionally stores the $k$ eigenvalues in $\Lambda$ and has the same $O(nk)$ asymptotic update cost through equation 13. Lanczos estimation costs $O(k \cdot \text{HVP})$ per task, where each Hessian–vector product has roughly the cost of one gradient evaluation. Restricting $n$ through parameter-efficient fine-tuning, such as LoRA, reduces these costs in the DomainNet experiment described in Section 5.2.

## 5 EXPERIMENTS

We evaluate Nostalgia-EWC and Nostalgia in two sequential vision-classification settings. The first is a controlled three-task sequence using a compact ResNet-10 [He et al. (2016)], which isolates retention and plasticity across heterogeneous datasets. The second is a six-task domain-incremental experiment on DomainNet using a ViT-B/16 [Dosovitskiy et al. (2021)] with LoRA, which tests the method under distribution shifts within a shared label space. Hard-projection Nostalgia is included as an ablation of how the shared Hessian memory is used. Unless explicitly stated otherwise, results are based on single runs and should therefore be interpreted descriptively rather than as estimates of statistically significant differences.

### 5.1 THREE-TASK CLASSIFICATION SETUP

We train a ResNet-10 [He et al. (2016)] sequentially on CIFAR-10 [Krizhevsky (2009)], MNIST [LeCun et al. (1998)], and CIFAR-100 [Krizhevsky (2009)]. This sequence varies visual complexity and label cardinality, producing substantial forgetting under naive fine-tuning. Complete training and implementation details are given in Appendix A.1.

**Methods and baselines.** We evaluate NOSTALGIA-EWC as the primary method and use hard-projection NOSTALGIA as a curvature-mechanism ablation. We compare them against NAIVE ADAM, EWC [Kirkpatrick et al. (2017)], GPM [Saha et al. (2021)], and A-GEM [Chaudhry et al. (2019)]. All methods share the same AdamW base optimizer and the same per-task training budget.

### 5.2 DOMAINNET: CONTINUAL DOMAIN-INCREMENTAL CLASSIFICATION

We evaluate on DOMAINNET [Peng et al. (2019)], whose six domains share 345 classes but exhibit substantial distribution shift. The domains are learned sequentially from clipart to sketch without task identity at test time. We adapt a ViT-B/16 [Dosovitskiy et al. (2021)] with LoRA [Hu et al. (2022)]; complete training and subsampling details are provided in Appendix A.2.

We report average forgetting over the first five domains, average final accuracy over all six, and plasticity as accuracy on the final domain. Forgetting is the difference between a domain's accuracy immediately after learning it and its accuracy at the end of the sequence; negative values indicate backward transfer.

| Method | Avg. forgetting ↓ | Avg. final acc. ↑ | Plasticity ↑ |
|---|---|---|---|
| NOSTALGIA | 0.046 | 0.596 | 0.672 |
| NOSTALGIA-EWC | **0.023** | **0.605** | 0.661 |
| GPM | 0.047 | 0.596 | 0.667 |

Table 1: Continual domain-incremental classification on DOMAINNET (6 domains, ViT-B/16 + LoRA $r$=8, single seed). Lower forgetting is better (↓), higher accuracy is better (↑). Avg. final acc. is averaged over all six tasks; forgetting over the five forgettable tasks.

| Method | Clipart | | Infograph | | Painting | | Quickdraw | | Real | | Sketch |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Acc ↑ | For ↓ | Acc ↑ | For ↓ | Acc ↑ | For ↓ | Acc ↑ | For ↓ | Acc ↑ | For ↓ | Acc ↑ |
| Naive (AdamW) | 0.715 | 0.023 | 0.265 | −0.021 | 0.688 | 0.016 | 0.575 | 0.213 | 0.884 | 0.003 | 0.671 |
| NOSTALGIA | 0.718 | 0.021 | 0.265 | −0.021 | 0.692 | 0.018 | 0.575 | 0.206 | 0.883 | 0.004 | **0.672** |
| NOSTALGIA-EWC | **0.719** | **0.012** | **0.267** | −0.039 | 0.687 | **0.006** | 0.529 | **0.132** | 0.882 | **0.002** | 0.661 |
| GPM | 0.718 | 0.019 | 0.266 | −0.022 | **0.694** | 0.018 | **0.579** | 0.213 | **0.886** | 0.005 | 0.667 |

Table 2: Per-domain retention on DOMAINNET. (**Acc**) accuracy at end of that domain's own fine-tuning block; (**For**) $= \mathrm{Acc}_{\mathrm{own}} - \mathrm{Acc}_{\mathrm{final}}$, positive = forgotten, negative = backward transfer. Sketch is last, forgetting undefined. ViT-B/16 + LoRA $r$=8, single seed.

### 5.2.1 RESULTS

Table 1 summarises retention and plasticity for the two NOSTALGIA variants and the GPM baseline, while Table 5.2.1 reports forgetting for each domain. All three methods retain clipart, infograph, painting and real almost perfectly, while forgetting is concentrated primarily in **quickdraw**; infograph exhibits backward transfer for every method.

Per-domain retention on DOMAINNET. Domains are presented left to right in the order the model sees them. For each domain we report the accuracy achieved at the end of that domain's own fine-tuning block (**Acc**) and the subsequent forgetting $\mathrm{Acc}_{\mathrm{own}} - \mathrm{Acc}_{\mathrm{final}}$ (**For**); positive forgetting indicates degradation, negative values indicate backward transfer. Sketch is the final domain and cannot be forgotten, so its forgetting is not defined (–). All runs use ViT-B/16 with LoRA $r$=8 and a single seed.

Two observations. First, adding the low-rank Hessian penalty of NOSTALGIA to EWC roughly halves average forgetting (0.046 → 0.023) with *no* plasticity cost (0.672 → 0.661), and yields the highest average final accuracy (0.605). Second, GPM — which shares the null-space projection machinery but builds its subspace from gradients rather than curvature — lands within noise of plain NOSTALGIA. That pair of results is consistent with the interpretation that the benefit comes from constraining updates to a protected subspace, and that replacing the gradient-based subspace with a curvature-based one yields a comparably-sized protected subspace without changing the qualitative behaviour.

**Limitations.** These numbers are a single seed and we therefore do not attach significance to differences of the size reported here (0.023 vs 0.046); our rank ablation required three seeds to resolve a larger effect. The comparison set is also partial (naive fine-tuning, EWC, A-GEM and SDFT are reported separately). Multi-seed results for the full baseline suite are deferred to the revision.

## 5.3 THREE-TASK CLASSIFICATION RESULTS

Table 3 reports the final validation accuracy of each method on each task after the full sequential training. The grid in Figure 2 shows validation accuracy as a function of global training step for all three tasks, with the task boundary marked by the visible accuracy drop on the first task and the corresponding rise on the newly introduced task.

**Nostalgia-EWC** provides the strongest retention of the two earlier tasks among the evaluated methods. In particular, it substantially limits the degradation of CIFAR-10 and preserves MNIST

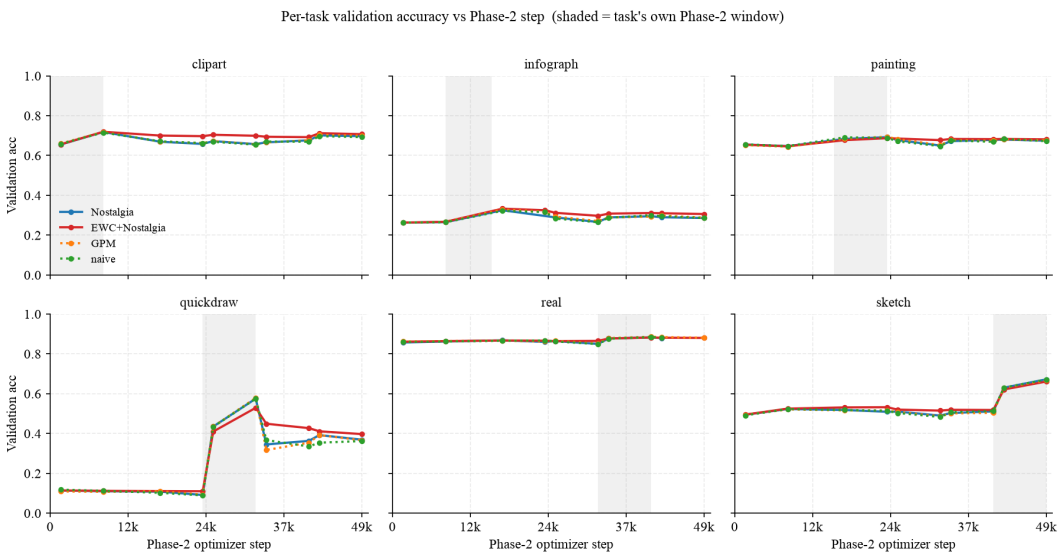



Figure 1: Retention and final accuracy across the six DOMAINNET domains (local runs, single seed). Forgetting is dominated by quickdraw for every method.

| Method | MNIST | | CIFAR-10 | | CIFAR-100 |
|---|---|---|---|---|---|
| | Acc ↑ | Forget ↓ | Acc ↑ | Forget ↓ | Acc ↑ |
| NAIVE ADAM | 0.557 | 0.438 | 0.549 | 0.281 | 0.543 |
| EWC | 0.760 | 0.234 | 0.648 | 0.182 | 0.532 |
| GPM | 0.649 | 0.346 | 0.554 | 0.276 | 0.537 |
| A-GEM | 0.317 | 0.677 | 0.224 | 0.606 | 0.470 |
| NOSTALGIA | 0.594 | 0.402 | 0.583 | 0.247 | **0.543** |
| NOSTALGIA-EWC | **0.827** | **0.163** | **0.804** | **0.026** | 0.466 |

Table 3: Validation accuracy and forgetting per task after sequential training on CIFAR-10 → MNIST → CIFAR-100. **Acc** is the final validation accuracy at the end of all three tasks; **Forget** is the drop from peak accuracy on that task ($\max_t$ acc − final acc). Means and standard deviations are reported over three seeds for Nostalgia; other methods use a single seed. Arrows indicate the desired direction. The final-task forgetting is identically zero for every method because no subsequent task is trained after CIFAR-100.

more effectively than standard EWC, suggesting that the low-rank Hessian anchor captures past-task dependencies that a diagonal importance estimate misses. This stability comes at the cost of lower performance on CIFAR-100, the final task in the sequence. The result therefore illustrates a clear retention–plasticity trade-off: stronger protection of high-curvature directions preserves earlier knowledge, but restricts the model's ability to adapt fully to the newest task.

**Hard-projection ablation.** Unprotected sequential Adam reaches 0.543 on the most recent task but loses substantial accuracy on the two earlier tasks. Hard-projection Nostalgia preserves the same reported CIFAR-100 accuracy and improves final accuracy by 0.034 on CIFAR-10 and 0.037 on MNIST. It retains less earlier-task accuracy than both Nostalgia-EWC and standard EWC, but its stronger final-task accuracy illustrates the greater plasticity of projection without a parameter anchor.

**Rank sensitivity of hard projection.** Unlike the main ResNet-10 experiment, the ablation in Figure 3 uses a ViT fine-tuned with LoRA rank 8. Varying the protected Hessian rank from $k = 8$ to $k = 64$ progressively reduces average forgetting from approximately 0.139 to 0.103, while final-task accuracy stays nearly constant at about 0.81. This result supports the usefulness of the shared

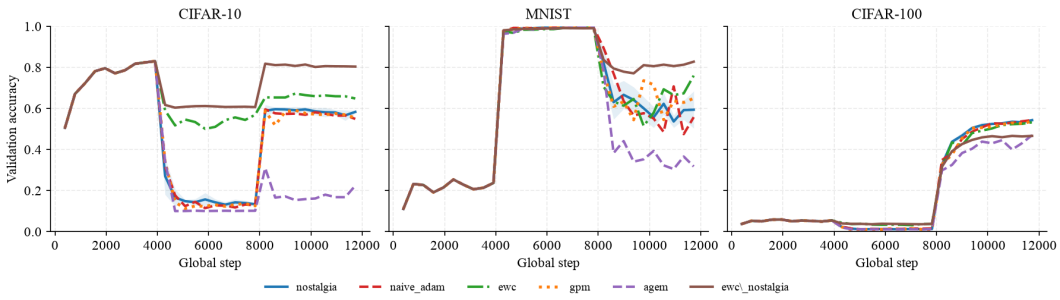



Figure 2: Validation accuracy curves across the full training trajectory. Vertical jumps in each panel mark the transition to a new task.

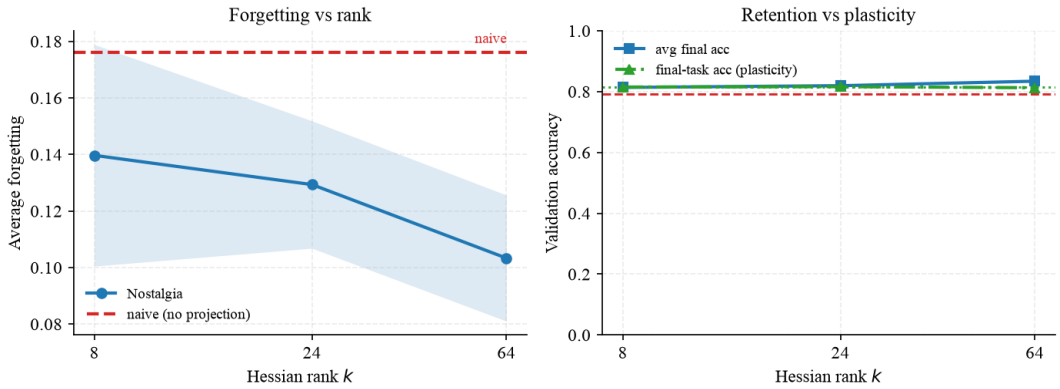



Figure 3: Rank sensitivity of the hard-projection Nostalgia ablation on a ViT fine-tuned with LoRA rank 8; this experiment is separate from the ResNet-10 evaluation in Table 3. Left: average forgetting, with the dashed line denoting naive training without projection. Right: average final validation accuracy and final-task accuracy, used as measures of overall performance and plasticity, respectively. The shaded region shows the variation across Nostalgia runs.

spectral memory in this ViT–LoRA setting, but it is not directly comparable to the ResNet-10 results and does not characterize the sensitivity of Nostalgia-EWC to $k$ or to its regularization coefficient $\lambda$.

### 5.4 Discussion

The central result is that Nostalgia-EWC converts the low-rank average-Hessian estimate into substantially stronger protection of earlier tasks than either diagonal-Fisher EWC or hard projection in the reported runs. This supports the proposed use of structured curvature as an EWC importance matrix. However, the lower CIFAR-100 accuracy shows that strong retention is purchased with reduced plasticity, and the single Nostalgia-EWC run is insufficient to establish a stable ranking over the baselines. The contrast with hard-projection Nostalgia clarifies the role of the proposed objective. Projection produces modest retention gains with little apparent loss of final-task accuracy. Nostalgia-EWC additionally penalizes cumulative displacement from the previous solution in proportion to curvature, producing much stronger stability but less plasticity. This difference motivates future calibration of $\lambda$ and rank $k$ rather than treating the two mechanisms as interchangeable.

**Limitations and scope.** In this work we don't compare newer methods for continual learning in NLP, RL and other sequence related problems. Methods such as SDFT [Shenfeld et al. (2026)] and it's RL variants such as SDPO [Hübotter et al. (2026)] while they for newer methods they remain limited to NLP and we haven't found an equivalent for Vision. NLP applications are considered out of the scope of this paper as we only consider classicaly CL algorithms.

## 6 AI USAGE

In this paper, we have used AI tools for certain parts of the paper, including the implementation and planning of the experiments. It was also used to validate certain ideas, which were human-thought-out initially, such as extending our nostalgia method with regularisation leading to nostalgia EWC. Some details for improving the computations related to Lanczos was done with the help of LLMs. Some parts of the paper's writing were refined by the use of AI, this includes grammar, and grounding our paper's claims. We also use AI's assistance for expanding our related works to ensure that our work is truely novel. The authors take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

## REFERENCES

Matteo Boschini, Lorenzo Bonicelli, Pietro Buzzega, Angelo Porrello, and Simone Calderara. Class-incremental continual learning into the extended der-verse. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(5):5497–5512, 2023. doi: 10.1109/TPAMI.2022.3206549. URL https://doi.org/10.1109/TPAMI.2022.3206549.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and SIMONE CALDERARA. Dark experience for general continual learning: a strong, simple baseline. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 15920–15930. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/b704ea2c39778f07c617f6b7ce480e9e-Paper.pdf.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-GEM. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Hkf2_sC5FX.

Arslan Chaudhry, Albert Gordo, Puneet K. Dokania, Philip Torr, and David Lopez-Paz. Using hindsight to anchor past knowledge in continual learning, 2021. URL https://arxiv.org/abs/2002.08165.

Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *ArXiv*, abs/2205.13535, 2022. URL https://api.semanticscholar.org/CorpusID:249097890.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Robert M. French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3 4:128–135, 1999. URL https://api.semanticscholar.org/CorpusID:2691726.

Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2232–2241. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/ghorbani19b.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

Zhao Kang, Xiao Lu, Yiwei Lu, Chong Peng, Wen-Yu Chen, and Zenglin Xu. Structure learning with similarity preserving. *Neural networks : the official journal of the International Neural Network Society*, 129:138–148, 2019. URL https://api.semanticscholar.org/CorpusID:208547679.

Shmuel Kaniel. Estimates for some computational techniques in linear algebra. *Mathematics of Computation*, 20(95):369–378, 1966.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13): 3521–3526, March 2017. ISSN 1091-6490. doi: 10.1073/pnas.1611835114. URL http://dx.doi.org/10.1073/pnas.1611835114.

Anat Kleiman, Gintare Karolina Dziugaite, Jonathan Frankle, Sham M. Kakade, and Mansheej Paul. Soup to go: mitigating forgetting during continual learning with model averaging, 2025. URL https://openreview.net/forum?id=n2EU4PUrJP.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45(4): 255–282, 1950.

Giulia Lanzillotta, Sidak Pal Singh, Benjamin F Grewe, and Thomas Hofmann. Local vs global continual learning. In Vincenzo Lomonaco, Stefano Melacci, Tinne Tuytelaars, Sarath Chandar, and Razvan Pascanu (eds.), *Proceedings of The 3rd Conference on Lifelong Learning Agents*, volume 274 of *Proceedings of Machine Learning Research*, pp. 121–143. PMLR, 29 Jul–01 Aug 2025. URL https://proceedings.mlr.press/v274/lanzillotta25a.html.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Jiaqi Li, Yuanhao Lai, Rui Wang, Changjian Shui, Sabyasachi Sahoo, Charles X Ling, Shichun Yang, Boyu Wang, Christian Gagné, and Fan Zhou. Hessian aware low-rank perturbation for order-robust continual learning. *IEEE Transactions on Knowledge and Data Engineering*, 36 (11):6385–6396, 2024. doi: 10.1109/TKDE.2024.3419449. URL https://doi.org/10.1109/TKDE.2024.3419449.

Mei Li, Yuxiang Lu, Qinyan Dai, Suizhi Huang, Yue Ding, and Hongtao Lu. BECAME: Bayesian continual learning with adaptive model merging. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 35481–35501. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/li25bk.html.

Xuhong Li, Yves Grandvalet, and Franck Davoine. A baseline regularization scheme for transfer learning with convolutional neural networks. *Pattern Recognition*, 98:107049, 2020. ISSN 0031-3203. doi: 10.1016/j.patcog.2019.107049.

Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pp. 109–165. Academic Press, 1989. doi: https://doi.org/10.1016/S0079-7421(08)60536-8. URL https://www.sciencedirect.com/science/article/pii/S0079742108605368.

Christopher Conway Paige. *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices*. PhD thesis, University of London, 1971.

Vardan Papyan. Measurements of three-level hierarchical structure in the outliers in the spectrum of deepnet hessians. In *International Conference on Machine Learning*, 2019. URL https://api.semanticscholar.org/CorpusID:59222724.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Barak Pearlmutter. Fast exact multiplication by the hessian. *Neural Computation*, 6:147–160, 01 1994. doi: 10.1162/neco.1994.6.1.147.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1406–1415, 2019.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. AdapterFusion: Non-destructive task composition for transfer learning. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty (eds.), *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 487–503, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.39. URL https://aclanthology.org/2021.eacl-main.39/.

Haomiao Qiu, Miao Zhang, Ziyue Qiao, and Liqiang Nie. Train with perturbation, infer after merging: A two-stage framework for continual learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=lqm1qJ43Sw.

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. *CoRR*, abs/1810.11910, 2018. URL http://arxiv.org/abs/1810.11910.

Levent Sagun, Leon Bottou, and Yann LeCun. Eigenvalues of the hessian in deep learning: Singularity and beyond, 2017. URL https://arxiv.org/abs/1611.07476.

Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=3AOj0RCNC2.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning, 2026. URL https://arxiv.org/abs/2601.19897.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 3987–3995. JMLR.org, 2017.

# A Additional Analysis

## A.1 Three-task classification experimental details

The ResNet-10 is trained end-to-end on an M4 Pro with 48 GB of unified memory. Each task is trained for 10 epochs with batch size 128, backbone learning rate $10^{-3}$, weight decay $10^{-4}$, and 500 linear-warmup steps. Validation is performed after every epoch, and method-specific state is computed after completing each task. We implement the methods in PyTorch [Paszke et al. (2019)]. Lanczos iterations use double-backward Hessian–vector products over flattened backbone parameters, with gradient projection applied separately to each parameter group.

### A.2 DomainNet experimental details

DomainNet images are resized to $224 \times 224$. We adapt the ViT-B/16 backbone with LoRA rank $r = 8$, scaling $\alpha = 16$, and dropout $0.05$, restricting the second-order computations to approximately $2.9 \times 10^5$ adapter parameters. Domains follow the fixed sequence clipart $\rightarrow$ infograph $\rightarrow$ painting $\rightarrow$ quickdraw $\rightarrow$ real $\rightarrow$ sketch. Each domain uses three head-alignment epochs followed by five LoRA fine-tuning epochs. We subsample the number of training points of about $\approx 3.4 \times 10^3$ images per class-domain pair.

**Note on AdamW optimizer:** We use AdamW as the common base optimizer across all methods for a controlled comparison. For Nostalgia, however, we do not apply AdamW's momentum and adaptive coordinate-wise preconditioning to the projected update, since these transformations do not preserve the hard projection constraint. Instead, the projection is enforced directly on the update used to modify the parameters. Thus, "AdamW" here refers to the common optimization/training configuration, while the Nostalgia update is explicitly constrained to remain in the orthogonal complement of the protected subspace.

### A.3 Krylov Subspaces and Computations

The central computational obstacle in the single-task eigenspace estimation step of Section 4 is that we need the dominant eigenspace of a task Hessian $H_\tau \in \mathbb{R}^{n \times n}$ without ever forming $H_\tau$ explicitly. The only primitive we are willing to use is a *Hessian–vector product* (HVP) oracle

$$v \longmapsto H_\tau v, \qquad v \in \mathbb{R}^n,$$

which can be computed by a double-backpropagation procedure [Pearlmutter (1994)] at roughly the cost of one gradient evaluation, with no $O(n^2)$ storage. Krylov subspace methods are exactly the family of algorithms designed to extract spectral information from a matrix using only such matrix–vector products.

**Definition 1** (Krylov subspace)**.** Given a symmetric matrix $H \in \mathbb{R}^{n \times n}$, a starting vector $v_1 \in \mathbb{R}^n$ with $\|v_1\|_2 = 1$, and an order $m \leq n$, the order-$m$ Krylov subspace generated by $H$ and $v_1$ is

$$\mathcal{K}_m(H, v_1) \;=\; \operatorname{span}\left\{v_1,\, Hv_1,\, H^2 v_1,\, \ldots,\, H^{m-1} v_1\right\}.$$

Two properties of $\mathcal{K}_m$ make it the right object for our setting.

**(i) It is built entirely from HVPs.** Each additional basis vector $H^j v_1$ is obtained by applying the HVP oracle to the previous one, so constructing $\mathcal{K}_m$ costs $m$ oracle calls and $O(nm)$ memory — never an $n \times n$ matrix.

**(ii) It concentrates rapidly along dominant eigendirections.** Writing $H = U \Sigma U^\top$ for the (unknown) eigendecomposition of $H$ and expanding $v_1 = \sum_{i=1}^n c_i u_i$ in the eigenbasis, we have

$$H^j v_1 \;=\; \sum_{i=1}^{n} c_i\, \sigma_i^{\,j}\, u_i.$$

As $j$ grows, the terms with the largest $|\sigma_i|$ dominate the sum geometrically — this is the same mechanism that drives power iteration, but $\mathcal{K}_m$ retains *all* $m$ iterates simultaneously rather than only the last one, which is what lets a single Krylov space of modest order $m$ resolve several leading eigenpairs at once rather than just the top one. This is precisely why the empirical Hessian spectra reported in prior work [Sagun et al. (2017); Papyan (2019); Ghorbani et al. (2019)]—a few large outlier eigenvalues sitting above a bulk near zero — are favorable for Krylov methods: a Krylov space of order $m \ll n$ already contains near-exact directions for the outliers we care about.

Optimizing a Rayleigh quotient over $\mathcal{K}_m$ rather than over all of $\mathbb{R}^n$ gives the classical variational characterization of Krylov approximations: the best rank-$k$ ($k < m$) symmetric approximation extractable from $\mathcal{K}_m$ is obtained by restricting $H$ to this subspace and diagonalizing the (small) restricted operator. Formally, if $V_m = [v_1 \,|\, v_2 \,|\, \cdots \,|\, v_m] \in \mathbb{R}^{n \times m}$ is any orthonormal basis of $\mathcal{K}_m(H, v_1)$, the *Ritz pairs* $(\theta_i, V_m y_i)$ obtained from the eigenpairs $(\theta_i, y_i)$ of the compressed matrix

$$T_m \;=\; V_m^\top H V_m \;\in\; \mathbb{R}^{m \times m}$$

are the best available estimates, within $\mathcal{K}_m$, of the true eigenpairs of $H$: $\theta_i$ and $V_m y_i$ satisfy $\|H(V_m y_i) - \theta_i(V_m y_i)\|_2 = \min$ over unit vectors in $\mathcal{K}_m$. The remaining question — answered in Section A.4 — is how to build an orthonormal $V_m$ and the compressed operator $T_m$ without ever forming $\mathcal{K}_m$'s basis by brute-force Gram–Schmidt on $\{v_1, Hv_1, \dots\}$, which is numerically unstable once $m$ is more than a handful of steps.

### A.4 Lanczos Tridiagonalization and the Low-Rank Estimate $Q\Lambda Q^\top$

For symmetric $H$, the Lanczos algorithm produces the orthonormal basis $V_m$ of $\mathcal{K}_m(H, v_1)$ implicitly and stably, via a three-term recurrence, and $T_m = V_m^\top H V_m$ comes out *tridiagonal* for free [Lanczos (1950)]—this is the property used to construct the single-round estimate in equation 8.

**The recurrence.** Starting from $v_1$ (unit norm, $\beta_1 := 0$, $v_0 := 0$), Lanczos generates $v_2, v_3, \dots, v_m$ by

$$w_j = Hv_j - \beta_j v_{j-1}, \tag{14}$$

$$\alpha_j = v_j^\top w_j, \tag{15}$$

$$\tilde{w}_j = w_j - \alpha_j v_j, \tag{16}$$

$$\beta_{j+1} = \|\tilde{w}_j\|_2, \qquad v_{j+1} = \tilde{w}_j / \beta_{j+1}. \tag{17}$$

Every step touches $H$ only through the single HVP $Hv_j$ in equation 14, matching the cost model of Section 4.2: each Lanczos step is one HVP, i.e. one extra gradient-scale computation, plus $O(n)$ vector arithmetic. Collecting the scalars $\alpha_j, \beta_j$ produced along the way gives a tridiagonal matrix

$$T_m = \begin{pmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \beta_3 & \\ & \beta_3 & \ddots & \ddots \\ & & \ddots & \alpha_m \end{pmatrix} \in \mathbb{R}^{m \times m},$$

and one can verify directly from equation 14–equation 17 that, in exact arithmetic, $HV_m = V_m T_m + \beta_{m+1} v_{m+1} e_m^\top$, i.e. $T_m = V_m^\top H V_m$ exactly, with the residual term measuring how far $\mathcal{K}_m$ is from being $H$-invariant.

**From $T_m$ to the rank-$k$ estimate.** Because $T_m$ is tridiagonal and only $m \times m$ (with $m$ a small constant, independent of $n$), its full eigendecomposition $T_m = Y \Theta Y^\top$ is obtained with a standard $O(m^3)$ symmetric eigensolver, which is negligible next to the $O(nm)$ cost of forming $V_m$. The Ritz pairs $\{(\theta_i, V_m y_i)\}_{i=1}^m$ are then sorted by $|\theta_i|$, and we retain the top-$k$ pairs corresponding to the largest-magnitude Ritz values as the single-round eigenspace estimate

$$H_\tau^{(e)} \;\approx\; Q_\tau^{(e)} \Lambda_\tau^{(e)} \big(Q_\tau^{(e)}\big)^\top, \qquad Q_\tau^{(e)} = V_m Y_{:,1:k} \in \mathbb{R}^{n \times k}, \;\; \Lambda_\tau^{(e)} = \mathrm{diag}(\theta_1, \dots, \theta_k),$$

which is the object introduced in equation 8. As stated in the single-task eigenspace estimation step of Section 4, only the non-negative Ritz values are kept when forming the weighted factor $F_\tau^{(e)} = Q_\tau^{(e)} (\Lambda_\tau^{(e)})_+^{1/2}$, since negative curvature directions do not correspond to a valid positive-semidefinite curvature memory for the null-space constraint in equation 6.

**Why this converges quickly for our use case.** The classical Kaniel–Paige bound [Kaniel (1966); Paige (1971)] quantifies the intuition from Section A.3: for the largest eigenvalue $\sigma_1$ with gap $\Delta = \sigma_1 - \sigma_2$ to the next eigenvalue, the Ritz value $\theta_1$ obtained after $m$ Lanczos steps satisfies

$$0 \;\le\; \sigma_1 - \theta_1 \;\le\; (\sigma_1 - \sigma_n) \left( \frac{\tan \phi_1}{T_{m-1}(1 + 2\Delta/(\sigma_1 - \sigma_n))} \right)^2,$$

where $\phi_1$ is the angle between $v_1$ and the true top eigenvector and $T_{m-1}$ is a Chebyshev polynomial, so convergence is geometric in $m$ whenever $\Delta > 0$ is non-negligible relative to the spectral range. Analogous bounds hold for the subsequent eigenpairs. This is precisely what makes rank-$k$ truncation with a small $k$ a good approximation of $H_\tau$ in practice: the outlier-dominated spectra reported for overparameterized networks give a large effective gap between the top $k$ directions and the bulk, so $m$ need only modestly exceed $k$ (in our runs, $m \approx 2$–$3k$) before the retained Ritz pairs are accurate to within numerical tolerance, keeping the accumulation-round estimation described in Section 4 to $O(k \cdot \mathrm{HVP})$ cost per task, as claimed in Section 4.2.

### A.5 Forgetting Bounds for Hard Projection

We analyze the hard-projection variant in equation 11, using the task sequence, displacement $\Delta_t$, average Hessian $\bar{H}_{t-1}$, and spectral memory $Q_{t-1}$ already defined in Sections 3 and 4. The only additional notation is the eigendecomposition

$$\bar{H}_{t-1} = \sum_{i=1}^{n} \sigma_i u_i u_i^\top, \qquad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0,$$

where the positive-semidefinite assumption matches the non-negative spectral memory constructed by Algorithm 2. Define $U_k = [u_1, \ldots, u_k]$ and $\mathcal{U}_k = \mathrm{span}(U_k)$.

**Lemma 2** (Projection invariant)**.** *If every update on task $t$ uses equation 11 while $Q_{t-1}$ is held fixed, then $Q_{t-1}^\top \Delta_t = 0$.*

*Proof.* $\Delta_t = -\sum_s \eta_s \tilde{g}_s$, where each projected gradient $\tilde{g}_s = (I - Q_{t-1}Q_{t-1}^\top) g_s$ satisfies $Q_{t-1}^\top \tilde{g}_s = 0$ by equation 11. Linearity of $Q_{t-1}^\top(\cdot)$ over the sum gives the claim. □

Under the local quadratic assumptions developed in prior work [Lanzillotta et al. (2025)], reviewed in Section 3, and conditional on zero forgetting through task $t-1$, the forgetting incurred by $\Delta_t$ is

$$E_t(\theta_t) = \tfrac{1}{2} \Delta_t^\top \bar{H}_{t-1} \Delta_t. \tag{18}$$

**Theorem 3** (Exact top-$k$ projection)**.** *Suppose $\mathrm{span}(Q_{t-1}) = \mathcal{U}_k$, i.e. the spectral memory exactly recovers the true top-$k$ eigenspace of $\bar{H}_{t-1}$. Then, under equation 18,*

$$E_t(\theta_t) \;\leq\; \tfrac{1}{2}\, \sigma_{k+1} \, \|\Delta_t\|^2.$$

*Proof.* Split $\bar{H}_{t-1} = H_{\mathrm{top}} + H_{\mathrm{tail}}$, where $H_{\mathrm{top}} = \sum_{i \leq k} \sigma_i u_i u_i^\top$ and $H_{\mathrm{tail}} = \sum_{i>k} \sigma_i u_i u_i^\top$. By Lemma 2 and $\mathrm{span}(Q_{t-1}) = \mathcal{U}_k$, $\Delta_t \perp \mathcal{U}_k$, hence $\Delta_t^\top H_{\mathrm{top}} \Delta_t = 0$. Since $H_{\mathrm{tail}} \preceq \sigma_{k+1} I$, $\Delta_t^\top H_{\mathrm{tail}} \Delta_t \leq \sigma_{k+1} \|\Delta_t\|^2$. Substituting into equation 18 gives the bound. □

**Definition 4** (Largest principal angle)**.** For $k$-dimensional subspaces $S_1, S_2 \subset \mathbb{R}^n$ with orthonormal bases $B_1, B_2$, the largest principal angle $\Theta_{\max}(S_1, S_2)$ satisfies

$$\sin \Theta_{\max}(S_1, S_2) \;=\; \max_{x \in S_1^\perp,\, \|x\|=1} \|P_{S_2} x\|,$$

where $P_{S_2}$ is orthogonal projection onto $S_2$.

**Theorem 5** (Forgetting under an imperfect subspace estimate)**.** *Let $\varepsilon = \sin \Theta_{\max}(\mathrm{span}(Q_{t-1}), \mathcal{U}_k)$. Then, under equation 18,*

$$E_t(\theta_t) \;\leq\; \tfrac{1}{2} \left(\sigma_{k+1} + \varepsilon^2 \sigma_1\right) \|\Delta_t\|^2.$$

*Proof.* As in Theorem 3, split $\bar{H}_{t-1} = H_{\mathrm{top}} + H_{\mathrm{tail}}$ and bound $\Delta_t^\top H_{\mathrm{tail}} \Delta_t \leq \sigma_{k+1} \|\Delta_t\|^2$; this step uses no assumption on $Q_{t-1}$. For the top term, write $H_{\mathrm{top}} = U_k U_k^\top H_{\mathrm{top}} U_k U_k^\top$ (it acts only within $\mathcal{U}_k$), so with $P_U := U_k U_k^\top$,

$$\Delta_t^\top H_{\mathrm{top}} \Delta_t = (P_U \Delta_t)^\top H_{\mathrm{top}} (P_U \Delta_t) \;\leq\; \sigma_1 \|P_U \Delta_t\|^2.$$

By Lemma 2, $\Delta_t \in \mathrm{span}(Q_{t-1})^\perp$, so by definition of the largest principal angle, $\|P_U \Delta_t\| \leq \varepsilon \|\Delta_t\|$. Combining the two terms and substituting into equation 18 gives the bound. □

*Remark* 6. Theorem 5 recovers Theorem 3 exactly when $\varepsilon = 0$, confirming the decomposition is consistent. The two terms separate forgetting into a component controlled by the choice of rank $k$ (via $\sigma_{k+1}$, which can decay quickly under the outlier-dominated Hessian spectra observed in prior work [Sagun et al. (2017); Papyan (2019); Ghorbani et al. (2019)]) and a component due purely to imperfect estimation of the top-$k$ eigenspace ($\varepsilon^2 \sigma_1$).

**Stability note.** In finite-precision arithmetic, the vectors $v_j$ produced by equation 14–equation 17 slowly lose orthogonality once Ritz values start converging, which can manufacture spurious duplicate copies of already-converged eigenpairs. We mitigate this with periodic full reorthogonalization of $v_{j+1}$ against $v_1, \ldots, v_j$ every few steps, which costs an additional $O(nm)$ but keeps $V_m^\top V_m \approx I$ to machine precision — cheap relative to the HVP cost that dominates the overall budget.