LEARNING WHILE FORGETTING: SPARSE ADAPTER CONSOLIDATION FOR MEMORY-BOUNDED CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual learning under a fixed retained-state budget eventually forces a choice about what to forget. Retaining one adaptation per task grows storage with the stream, while consolidating adaptations into shared memory makes older information progressively harder to recover. We introduce PALIMPSEST, a write–recover–gate learner that treats consolidation and forgetting as coupled problems. Each task's low-rank adapter is sparsely represented and superposed, with decay, into a shared fixed-size history store. Joint sparse recovery reconstructs candidate historical adapters. A support-recovery analysis motivates an age-dependent reliability horizon, while a calibrated load correction turns this into a deployment forecast that determines when to serve a recovered adapter and when to fall back to the frozen backbone. Reusing the ImageNet-R calibration without refitting predicts the recovery horizon of a running class-incremental learner with $R^2 = 0.914$. Gate interventions show that fallback recovers 3.1–14.1 percentage points of mean old-task accuracy after the predicted horizon under forced forgetting, while also exposing conservative cases in which useful adaptation is discarded. Under retained-state caps of 5M parameters on ImageNet-R and 15M on OmniBench-1K, PALIMPSEST leads both reported accuracy metrics at the longest evaluated horizons; at 200 tasks on OmniBench-1K, it exceeds the strongest capped baseline by 7.18 points in average accuracy and 6.90 points in final accuracy. The advantage emerges as memory becomes limiting, supporting sparse consolidation with explicit, forecast-guided forgetting as a principled approach to memory-bounded continual learning.

1 INTRODUCTION

Continual learning (CL) aims to learn a sequence of tasks while retaining knowledge from those learned before (Parisi et al., 2019; De Lange et al., 2022; Wang et al., 2024). Its central challenge is catastrophic forgetting, in which adapting to new tasks degrades performance on earlier ones (McCloskey & Cohen, 1989; French, 1999; Kirkpatrick et al., 2017). Early work trained models from scratch and countered forgetting by regularizing important parameters (Kirkpatrick et al., 2017; Zenke et al., 2017), replaying stored exemplars (Rebuffi et al., 2017; Chaudhry et al., 2019), or expanding the architecture for each new task (Rusu et al., 2016; Yan et al., 2021). Recently, pre-trained models combined with parameter-efficient fine-tuning have become the dominant approach (Zhou et al., 2024a; Houlsby et al., 2019; Hu et al., 2022): the backbone stays frozen and only small modules are learned for each new task. However, a small per-task footprint can still accumulate substantial storage over a long stream. We study the regime in which retained state beyond the frozen backbone must fit a fixed budget, including adapters, dictionaries, and classifier statistics.

Most recent methods keep a separate lightweight component for every task. EASE trains a task-specific adapter subspace (Zhou et al., 2024b); TUNA combines task-specific adapters with a merged universal adapter (Wang et al., 2025); InfLoRA confines each new LoRA branch to a subspace that does not interfere with earlier tasks (Liang & Li, 2024); and SD-LoRA decouples the magnitude and direction of each task's low-rank update (Wu et al., 2025). Isolating tasks in this way sharply reduces interference and gives strong accuracy on short streams. The cost is storage: every task leaves behind its own component, so retained state grows linearly with the number of tasks. Figure 1(a) shows this growth, which eventually exceeds any fixed budget and forces the learner to evict, merge, or reject

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.

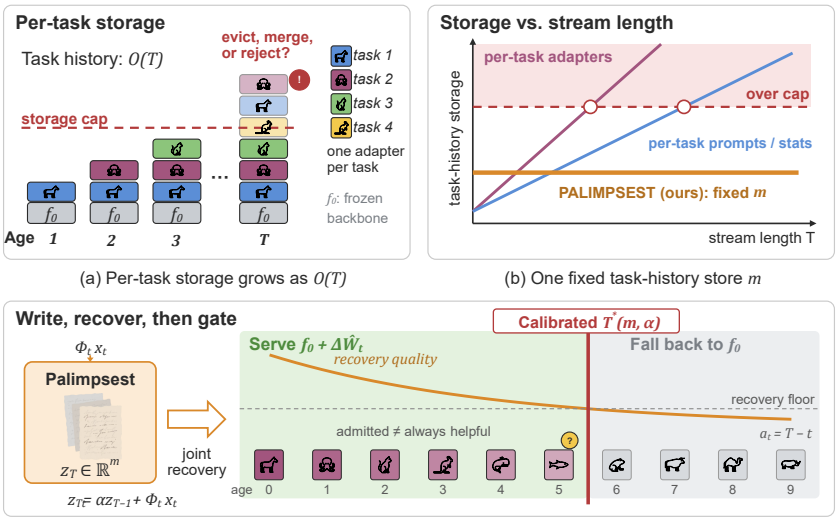


Figure 1: **Long-horizon continual learning under fixed memory. (a)** Retaining separate task state increases storage with the stream length T. **(b)** PALIMPSEST consolidates task adaptations into a store of fixed dimension m. **(c)** A calibrated horizon $T^*(m, \alpha)$ decides which recovered adapters are admissible; older tasks fall back to the frozen backbone (§4.6).

old state. Long streams therefore call for a learner whose retained state is fixed in advance, yet which stays accurate as the number of tasks keeps growing.

Reaching that goal raises two challenges. First, a fixed store must hold an unbounded history: new adaptations cannot open new slots, so they must share one bounded representation, and older information will inevitably become harder to recover as the stream grows. Second, the learner must know when a compressed adaptation is no longer trustworthy: a reconstructed adapter can cease to help, and even hurt, before the learner stops using it. In our gate experiments, reverting to frozen predictions improves mean old-task accuracy by 3–14 percentage points in settings where recovered adapters have become harmful (§4.6). Forgetting is thus unavoidable under a fixed budget, and the question is whether it can be made predictable and controlled rather than implicit.

Sparse representation and sparse recovery provide a natural analytical framework for this construction: many sparse signals superposed through incoherent measurements can be recovered jointly, and the same analysis predicts the load at which recovery breaks down (Donoho, 2006; Candès et al., 2006; Tropp, 2006). Building on this insight, we introduce PALIMPSEST, a continual learner that turns forgetting from an accident into a design choice. It compresses every task into one shared store of fixed size, reads past tasks back out when needed, and forecasts in advance how far into the past those reconstructions can be trusted, falling back to the frozen backbone beyond that point (Figure 1(b–c)). Storage stays fixed, and older tasks are forgotten gradually and predictably rather than all at once.

Extensive experiments show that, under a fixed storage budget, PALIMPSEST achieves state-of-the-art accuracy on long task streams. Its advantage grows with the stream length: it leads both metrics at the longest ImageNet-R horizon and at every OmniBench-1K horizon from 50 to 200 tasks. At 200 tasks it surpasses the strongest baseline by about 7 points in both average and final accuracy, while its own retained state stays within the budget, and the recovery horizon transfers without refitting to a running class-incremental learner ($R^2 = 0.914$). Our contributions are twofold:

- **Long-horizon continual learning under a fixed storage budget.** We introduce PALIMPSEST, whose write, recover, and gate modules keep task history in one store of fixed size. Under a fixed storage budget it achieves state-of-the-art accuracy on long task streams, and without a budget it remains competitive with the strongest baselines while retaining a fraction of their storage.
- **Predictable forgetting.** We turn forgetting into an explicit, forecastable quantity: a recovery horizon, motivated by compressed-sensing theory and computed before any data arrive, predicts

how far into the past recovery remains reliable; gate interventions show when acting on it prevents harmful adapter use and when it is too conservative.

2 Related Work

2.1 Learning setting and memory budgets

Continual learning asks a model to absorb a sequence of tasks without catastrophic forgetting, the loss of earlier competence when later tasks overwrite shared parameters (McCloskey & Cohen, 1989; French, 1999; McClelland et al., 1995). Surveys group the responses into three families (Parisi et al., 2019; De Lange et al., 2022; Wang et al., 2024; Zhou et al., 2024c). Regularization methods penalize drift in parameters deemed important for earlier tasks, as in EWC, SI, MAS, and LwF (Kirkpatrick et al., 2017; Zenke et al., 2017; Aljundi et al., 2018; Li & Hoiem, 2016). Rehearsal methods keep a buffer of past examples and replay them, from iCaRL to tiny episodic memories and gradient-based sample selection (Rebuffi et al., 2017; Chaudhry et al., 2019; Aljundi et al., 2019). Architecture methods allocate new capacity per task, whether whole columns, weight masks, expandable representations, or tokens (Rusu et al., 2016; Mallya et al., 2018; Yan et al., 2021; Douillard et al., 2022). With pre-trained transformers (Dosovitskiy et al., 2021; Zhou et al., 2024a), the field moved to parameter-efficient tuning: adapters, prompt and prefix tuning, visual prompts, and AdaptFormer add small trainable modules to a frozen backbone (Houlsby et al., 2019; Lester et al., 2021; Li & Liang, 2021; Jia et al., 2022; Chen et al., 2022). We study class-incremental learning in this regime: tasks with disjoint classes arrive in sequence, and after each one the model classifies among all classes seen so far without task identity; we report average and final incremental accuracy. Because parameter-efficient, exemplar-free learners still accumulate per-task modules, following memory-aware evaluation (Zhou et al., 2023) we count all state retained beyond the frozen backbone, including adapters, dictionaries, prompts, heads, prototypes, and statistics.

2.2 Continual learning with pre-trained models

Most pre-trained continual learners keep the backbone frozen and retain per-task state (Zhou et al., 2024a). SimpleCIL (Zhou et al., 2025) shows that frozen features with a prototype classifier are already competitive, and SLCA, RanPAC, FeCAM, and analytic readouts (Zhang et al., 2023; McDonnell et al., 2023; Goswami et al., 2023; Zhuang et al., 2022) improve the classifier with retained statistics. L2P, DualPrompt, CODA-Prompt, HiDe-Prompt, and LAE (Wang et al., 2022b;a; Smith et al., 2023; Wang et al., 2023a; Gao et al., 2023) learn prompt pools or lightweight modules selected per query. EASE, TUNA, DIA, MOS, and SSIAT (Zhou et al., 2024b; Wang et al., 2025; Li et al., 2025; Sun et al., 2025; Tan et al., 2024) retain or integrate one adapter per task, and CaRE routes among task-specific experts to scale to hundreds of tasks (Lou et al., 2026). The low-rank family builds on LoRA (Hu et al., 2022) and on gradient projection, which confines updates to subspaces orthogonal to earlier tasks (Zeng et al., 2019; Farajtabar et al., 2020; Saha et al., 2021; Lin et al., 2022; Chaudhry et al., 2020): O-LoRA, InfLoRA, SD-LoRA, GR-LoRA, and SplitLoRA (Wang et al., 2023b; Liang & Li, 2024; Wu et al., 2025; Lin et al., 2026; Qiu et al., 2026) control interference through subspace constraints, decoupled magnitudes, or gradient recycling and splitting. Every method above keeps a component per task or a pool that grows with the stream; we instead consolidate task updates in one store of fixed size.

2.3 Consolidation and sparse recovery

Merging many fine-tuned updates into one set of weights is the closest precedent for a store that does not grow. Model soups average fine-tuned weights, task arithmetic adds and subtracts task vectors, and Fisher-weighted averaging weights the merge by parameter importance (Wortsman et al., 2022; Ilharco et al., 2023; Matena & Raffel, 2022); their continual-learning descendants, task arithmetic with LoRA, CoMA, CoFiMA, and MagMax (Chitale et al., 2023; Marouf et al., 2024; Marczak et al., 2024), merge compatible updates into a fixed number of parameters as tasks arrive. Compressed sensing shows that a sparse signal can be recovered from far fewer linear measurements than its dimension, with basis pursuit succeeding under the restricted isometry property and failing sharply beyond a phase transition (Donoho, 2006; Candès & Tao, 2005; Candès et al., 2006; Baraniuk et al., 2008; Donoho & Tanner, 2009); simultaneous sparse approximation extends recovery to several

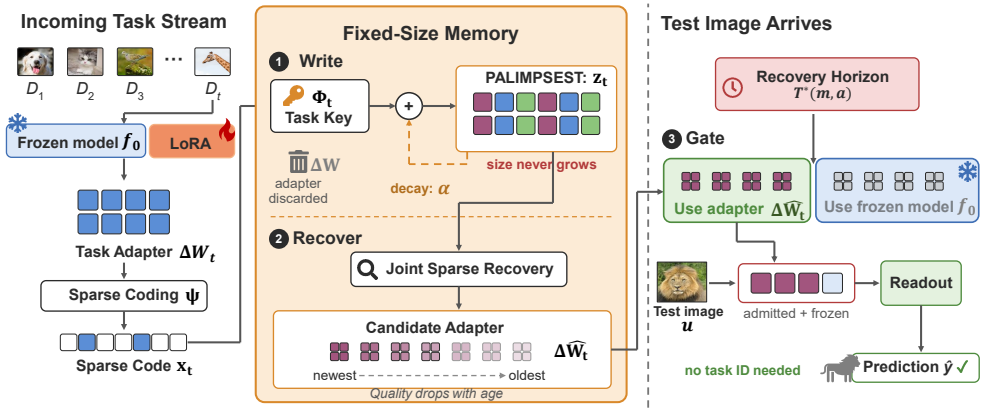


Figure 2: **Bounded-memory continual adaptation.** WRITE: each incoming task produces an adapter ΔW_t, which is encoded as a sparse code x_t, keyed by Φ_t, and accumulated into one fixed-size memory $z_t \in \mathbb{R}^m$; the full adapter is then discarded. RECOVER: the bounded state is jointly decoded into candidate historical adapters. GATE: a calibrated horizon forecasts how far into the past recovery remains reliable; older reconstructions are never served, and those tasks fall back to the frozen backbone. A shared nearest-prototype readout produces $\hat{y}$.

signals that share measurements (Tropp, 2006; Tropp & Gilbert, 2007). Sparse coding and dictionary learning supply the representation, from learned sparse codes for natural images to K-SVD and convolutional sparse coding (Olshausen & Field, 1996; Aharon et al., 2006; Papyan et al., 2017). Merging methods keep one blended weight and lose the ability to tell tasks apart. Our store keeps superposed sparse codes that a joint recovery can separate on demand, and the recovery theory gives an explicit horizon for when that separation stops being reliable, which becomes a policy for admitting adapters or falling back to frozen features.

3 METHOD: PALIMPSEST

3.1 OVERVIEW

We consider continual adaptation under a fixed budget: a stream of tasks $\mathcal{D}_1, \ldots, \mathcal{D}_T$ arrives sequentially at a frozen backbone f_0, learning task t produces a lightweight adapter ΔW_t, and a conventional task-specific learner keeps one adapter per task, so its adaptation storage grows with T. PALIMPSEST instead keeps a task-history store $z_t \in \mathbb{R}^m$ of fixed dimension m, an append-only dictionary Ψ of adapter atoms, and one prototype set per class. Its two phases are

Training:	adapter $\Delta W_t \Rightarrow$ sparse code $x_t \Rightarrow$ write $z \leftarrow \alpha z + \Phi_t x_t \Rightarrow$ discard ΔW_t
Testing:	recover $\hat{x}_t$ from $z_T \Rightarrow$ gate by age $a_t \leq T^* \Rightarrow$ embed query $u \Rightarrow$ predict $\hat{y}$

Training (Figure 2, step 1) sparse-codes each task's adapter, adds its keyed trace to the decayed store, registers the prototypes of its classes, and discards the adapter. Testing (steps 2 and 3) runs one joint sparse-recovery solve per memory state to read back candidate adapters, admits those younger than the recovery horizon $T^*(m, \alpha)$, and classifies each query by nearest prototype over all classes, without task identity. The store never grows and the dictionary adds only k atoms per layer per task; both count toward the retained-state budget B (§4.1), and Algorithm 1 lists the full procedure.

3.2 PALIMPSEST WRITING: COMPRESS EACH TASK INTO THE STORE

Writing turns an adapter into a few numbers in the shared store, in two steps: the adapter becomes a sparse code, and the code's keyed trace is added to the decayed memory.

After learning task t, its LoRA adapter $\Delta W_t \in \mathbb{R}^{d \times d}$ is represented in the shared dictionary Ψ as an approximately sparse code $\Delta W_t \approx \Psi x_t$, $x_t \in \mathbb{R}^p$ (the *sparse coding* box in Figure 2). Ψ is an *append-only* pooled-SVD dictionary: task t contributes the top-k singular components of its own

adapter as new atoms, and existing atoms are never changed. A code written at time t therefore decodes at any later time against exactly the atoms it was written with, so no discarded adapter is ever needed (§3.3). The growing atom count is charged to the budget B, and $k \approx 2$ atoms per layer already retain 98% of the adapter's gain over frozen (§4.7). The code is then keyed by a seeded random Gaussian map $\Phi_t \in \mathbb{R}^{m\times p}$ (the *task key*) and added to the decayed memory, as the dashed decay loop in Figure 2 shows, through the recursive write update

$$z_t = \alpha z_{t-1} + \Phi_t x_t, \qquad z_0 = 0, \qquad \text{so that} \qquad z_T = \sum_{t=1}^{T} \alpha^{T-t} \Phi_t x_t \in \mathbb{R}^m. \tag{1}$$

The memory is m numbers for any stream length: new tasks are superimposed onto the same z, and a task of age a survives with weight α^a, so each write gradually forgets the oldest tasks.

3.3 PALIMPSEST RECOVER: READ THE HISTORY BACK OUT

At inference, with no test-task identity, reading the memory means undoing the superposition (step 2 of Figure 2). Stacking the keys as $A = [\Phi_1|\cdots|\Phi_T]$ and the decay-weighted codes as $w^\star = (\alpha^{T-1}x_1; \ldots; x_T)$ gives the linear inverse problem $z_T = Aw^\star$, which we solve jointly with basis pursuit over all task blocks at once, yielding every candidate in a single solve:

$$\hat{w} = \arg\min_{w} \|w\|_1 \quad \text{s.t.} \quad Aw = z_T, \tag{2}$$

or its BPDN relaxation under noise. Block t is then rescaled by its age $a_t = T - t$ and decoded through the same dictionary used at write time, $\hat{x}_t = \alpha^{-a_t}\hat{w}_{[t]}$ and $\Delta\hat{W}_t = \Psi\hat{x}_t$.

One solve produces candidate reconstructions for all task blocks, the *candidate adapter* row of Figure 2 whose quality fades with age; it runs once per memory state and never sees the test image. Recovery only proposes candidates; the gate decides which to serve.

3.4 GATE: DECIDE WHICH CANDIDATES TO SERVE

How far back should recovered adapters be used? Reconstruction quality deteriorates as decay attenuates older traces and interference grows, and support-recovery analysis makes this explicit.

Under the sparse-code and Gaussian measurement assumptions of Appendix A.1, and conditioning on A satisfying the restricted isometry property at order $2K$, basis pursuit obeys $\|\hat{w} - w^\star\|_2 \le \varepsilon_K := C\,\|w^\star - w^\star_K\|_1/\sqrt{K}$, where $w^\star_K$ keeps the largest K coefficients. Since every coordinate error is then at most ε_K, the support S_t of block t is recovered exactly whenever $\alpha^{a_t} x_{\min} > 2\varepsilon_K$, with $x_{\min}$ the smallest nonzero code magnitude: decay shrinks older signals toward the error floor. This certifies support, not accuracy, and motivates rather than establishes the deployed horizon ($0 < \alpha < 1$):

$$T^* \;=\; \max\Big\{\,0,\;\; \min\big(a_{\text{cap}},\, T_{\text{sig}}\big)\,\Big(1 - \frac{L}{a_{\text{cap}}}\Big)_+\Big\}, \tag{3}$$

with $a_{\text{cap}} = C_1 m$, $L = (1-\alpha)^{-1}$, and $T_{\text{sig}} = (\tfrac{1}{2}\log m + C_2)/\log(1/\alpha)$. The capacity a_{cap} grows linearly with m, T_{sig} is the age at which decayed signals reach the recovery floor, and $(1 - L/a_{\text{cap}})_+$ is the headroom left as the live-task load L approaches capacity. Fast decay suppresses old traces and slow decay raises interference, so T^* has an interior optimum in α. Equation (3) is a calibrated forecast, not a bound: (C_1, C_2) are fitted once per domain, after which T^* is computed from (m, α) before the stream begins (the *recovery horizon* box in Figure 2); its link to accuracy is tested in §4.5–4.6, and Appendix A separates what is proved from what is calibrated.

The forecast defines the serving policy, step 3 of Figure 2: task t is served by $f_0 + \Delta\hat{W}_t$ if $a_t \le T^*$ and by the frozen f_0 otherwise, so recovery proposes candidates and the gate decides which are used. Each class keeps two prototypes, computed at registration under $f_0 + \Delta W_t$ and under f_0 (Alg. 1, line 6). At test time the image u is embedded under f_0 and each admissible candidate, the *admitted + frozen* row of Figure 2; every class is scored in the embedding its task is currently served under, and $\hat{y}$ is one nearest-prototype decision over the union of all classes, with normalized distances keeping scores comparable. Linking an adapter to its classes is training-time bookkeeping, as in expandable-component methods (Zhou et al., 2024b; Wang et al., 2025; Sun et al., 2025); age belongs to the stored block, not the query, so test-task identity is never used.

The gate therefore guarantees exactly one thing: no reconstruction older than T^* is ever applied. It does not guarantee that every served adapter beats the backbone, nor that a conservative forecast never discards a useful one; both cases are measured in §4.6.

Algorithm 1 Write–recover–gate continual learning under a fixed store

Require: frozen backbone f_0; store size m; decay α; calibrated constants (C_1, C_2)

$T^* \leftarrow \mathrm{horizon}(m, \alpha; C_1, C_2)$; $z \leftarrow 0$ ▷ Eq. (3), fixed before any data

Training: write, once per task

for task $t = 1, \ldots, T$ **do**
  fit LoRA adapter ΔW_t on $\mathcal{D}_t$ with f_0 frozen
  append the top-k singular components of ΔW_t to Ψ; sparse-code x_t ▷ $\Delta W_t \approx \Psi x_t$, §3.2
  $z \leftarrow \alpha z + \Phi_t x_t$; discard ΔW_t ▷ Eq. (1)
  register prototypes of the new classes under $f_0 + \Delta W_t$ and under f_0
end for

Testing: recover, once per memory state

$\hat{w} \leftarrow \arg\min \|w\|_1$ s.t. $[\Phi_1 | \cdots | \Phi_T]\, w = z$; $\Delta\hat{W}_t \leftarrow \Psi\big(\alpha^{-(T-t)}\hat{w}_{[t]}\big)$ for all t ▷ Eq. (2), §3.3

Testing: gate and read out, per image

$\mathcal{C} \leftarrow \{\, t : T - t \le T^* \,\}$ ▷ admissible by age, §3.4

for test image u **do**
  embed u under f_0 and under $f_0 + \Delta\hat{W}_t$ for each $t \in \mathcal{C}$
  $\hat{y} \leftarrow$ nearest prototype over the union of all classes
end for

Concretely, we use a frozen ViT-B/16 (ImageNet-21k) with rank-10 LoRA on the query and value projections. The joint solve is amortized over all queries, and recovered adapters are decoded one at a time, so peak working memory beyond the persistent state is one 0.37M-parameter adapter plus a feature buffer. Per image, the age-only rule admits $n_{\mathrm{adm}} = \min\{T, \lfloor T^* \rfloor + 1\}$ adapters, so each query costs $n_{\mathrm{adm}}+1$ backbone passes, bounded by the horizon rather than the stream length.

4 Experiments

4.1 Evaluation protocol

The main comparisons use ImageNet-R (Hendrycks et al., 2021), with 200 classes divided into $T \in \{10, 20, 40\}$ equal tasks, and OmniBenchmark-1K (Lou et al., 2026), a 1000-class benchmark built from OmniBenchmark (Zhang et al., 2022) for very long task sequences, divided into $T \in \{50, 100, 125, 200\}$ tasks; it is not the 300-class OmniBenchmark split used by EASE and TUNA.

Increasing T lengthens the stream without adding classes. The class order uses seed 1993 following Rebuffi et al. (2017). Recovery studies also use CIFAR-100 (Krizhevsky, 2009) and FGVC-Aircraft (Maji et al., 2013); the gate diagnostic uses ImageNet-R with 10 tasks under four random class orders, and the readout ablation uses OmniBench-1K at T=50.

Tables 1 and 3 compare LwF (Li & Hoiem, 2016); prompt methods L2P (Wang et al., 2022b), DualPrompt (Wang et al., 2022a), and CODA-Prompt (Smith et al., 2023); and LAE (Gao et al., 2023), InfLoRA (Liang & Li, 2024), EASE (Zhou et al., 2024b), SD-LoRA (Wu et al., 2025), TUNA (Wang et al., 2025), GR-LoRA (Lin et al., 2026), and SplitLoRA (Qiu et al., 2026), in the LAMDA-PILOT toolbox (Sun et al., 2023) on a ViT-B/16-IN21K backbone with released settings.

Two storage regimes are reported. The fixed-budget comparison caps all state retained beyond the frozen backbone at B=5M parameters on ImageNet-R and B=15M on OmniBench-1K. Growing-state baselines then use rejection (P3): they train normally until the cap is reached and store no new task state afterwards, so their capped and native results coincide until then (§4.3). Our retained state (3.1M and 8.2M, Table 3) is below both caps, so our configuration is identical in both regimes. Entries are $\mathcal{A}_{Avg}/\mathcal{A}_{Last}$ in percent, point estimates with each method's own readout; forecast quality is the R^2 between predicted and measured horizons, with reference calibration $(C_1, C_2) = (0.0184, -1.253)$ reused unchanged under "no refit" (Appendix B.1 lists all remaining conventions).

Table 1: **Accuracy under fixed storage:** 5M parameters for ImageNet-R and 15M for OmniBench-1K. Entries are $\mathcal{A}_{Avg}/\mathcal{A}_{Last}$ (%): average / final accuracy. Bold: best per metric and horizon; "–": unavailable. †: native state fits the cap (SplitLoRA on ImageNet-R only at T=10, in low-rank form); n/f: cannot fit the cap; (P3): new task state rejected at the cap (§4.3).

Method	Venue	ImageNet-R: $\mathcal{A}_{Avg}/\mathcal{A}_{Last}$ (B=5M)			OmniBench-1K: $\mathcal{A}_{Avg}/\mathcal{A}_{Last}$ (B=15M)			
		T=10	T=20	T=40	T=50	T=100	T=125	T=200
LwF†	ECCV'16	38.12 / 16.78	22.84 / 7.35	12.42 / 4.03	– / –	– / –	– / –	– / –
DualPrompt†	ECCV'22	75.08 / 69.23	72.65 / 66.27	67.18 / 59.47	62.70 / 50.38	60.10 / 47.18	58.40 / 45.24	53.00 / 36.28
L2P†	CVPR'22	77.96 / 72.35	75.75 / 70.12	71.99 / 65.12	61.70 / 49.57	59.20 / 47.40	58.60 / 46.04	55.10 / 43.43
LAE†	ICCV'23	76.27 / 69.73	71.59 / 64.52	60.93 / 50.67	62.50 / 48.80	55.40 / 40.87	52.30 / 36.64	42.19 / 25.99
CODA-Prompt†	CVPR'23	78.47 / 73.40	75.97 / 70.30	67.37 / 62.00	58.03 / 53.90	55.71 / 50.50	52.80 / 47.95	45.82 / 40.36
InfLoRA (P3)	CVPR'24	78.45 / 72.80	72.83 / 67.80	63.20 / 58.60	58.10 / 53.00	50.97 / 46.50	49.27 / 44.30	43.37 / 39.24
EASE-21k (P3)	CVPR'24	77.29 / 72.00	73.92 / 69.73	70.29 / 65.41	60.02 / 55.25	57.73 / 53.02	55.06 / 50.25	54.06 / 48.24
SD-LoRA (P3)	ICLR'25	77.03 / 70.20	72.11 / 65.90	61.42 / 54.60	52.79 / 47.96	49.93 / 44.27	46.63 / 41.96	43.09 / 39.24
TUNA (P3)	ICCV'25	80.02 / 75.23	75.90 / 71.85	72.13 / 67.32	64.79 / 60.98	58.50 / 53.52	55.28 / 50.66	49.27 / 43.18
GR-LoRA (P3)	ICML'26	85.93 / 78.04	**82.72** / 71.78	76.47 / 68.30	55.23 / 31.00	47.01 / 17.00	40.31 / 15.33	31.25 / 7.89
SplitLoRA†	ICLR'26	**86.45 / 81.07**	n/f	n/f	69.50 / 57.59	67.13 / 55.11	66.20 / 54.25	65.13 / 54.37
Ours	—	81.31 / 73.17	79.05 / **72.77**	**77.26 / 70.39**	**73.30 / 62.70**	**73.42 / 62.47**	**72.87 / 62.64**	**72.31 / 61.27**

Table 2: **Closure-policy ablation on ImageNet-R under the 5M-parameter cap.** Final accuracy $\mathcal{A}_{Last}$ (%) for eviction (P1), merging (P2), and rejection of new task state (P3). P3 is best for each tested baseline at both horizons; ours requires no external closure. Bold marks the best baseline per policy and the overall best result at each horizon.

Method	T=20			T=40		
	P1	P2	P3	P1	P2	P3
EASE-21k (K=3)	12.52	55.28	69.73	7.63	55.20	65.41
GR-LoRA (K=6)	64.87	64.52	71.78	61.73	62.17	**68.30**
InfLoRA (K=13)	57.65	59.00	67.80	39.25	38.50	58.60
SD-LoRA (K=13)	52.57	53.02	65.90	32.55	32.78	54.60
TUNA (K=16)	**71.27**	**71.00**	**71.85**	**65.12**	**66.70**	67.32
Ours		**72.77**			**70.39**	

4.2 Long-horizon performance under fixed storage

Table 1 tests performance when retained state cannot grow beyond the cap. On ImageNet-R, the advantage emerges as the stream lengthens. SplitLoRA leads both metrics at T=10 (86.45/81.07%). At T=20, ours leads in final accuracy (72.77%, 0.92 above TUNA) while GR-LoRA leads in average accuracy; by T=40, ours leads both at 77.26/70.39%, 0.79/2.09 points above GR-LoRA. From T=10 to 40, ours loses 4.05/2.78 points, against final-accuracy declines of 7.91 for TUNA and 9.74 for GR-LoRA, so the short-horizon ranking reverses; SplitLoRA cannot fit the 5M cap beyond T=10.

On OmniBench-1K, the lead persists through 200 tasks: ours leads both metrics at every horizon. At T=200, it reaches 72.31/61.27%, 7.18/6.90 points above SplitLoRA, the strongest baseline. Its final accuracy spans only 1.43 points across all four horizons, against SplitLoRA's 3.22-point decline, and its final-accuracy margin over the best baseline widens from 1.72 at T=50 to 7.36–8.39 at T=100–125 and 6.90 at T=200. As the task partition changes with T, these compare separate stream lengths rather than tracing retention along a single stream.

4.3 Sensitivity to the baseline closure policy

A capped comparison depends on what a growing-state method does when its budget is exhausted. Table 2 isolates this choice on ImageNet-R using final accuracy. Rejection (P3) is strongest for all five baselines at both horizons, but by method-dependent margins: at T=40, InfLoRA moves from 39.25% (eviction) and 38.50% (merging) to 58.60%, whereas TUNA moves from 65.12 and 66.70 to 67.32%. Ours still exceeds the best baseline-policy combination by 0.92 points at T=20 and 2.09 at T=40, so its lead does not rest on a weak closure choice.

Table 3: **Accuracy with native, unconstrained storage.** Entries are $\mathcal{A}_{Avg}/\mathcal{A}_{Last}$ (%): average / final accuracy. Bold marks the best for each metric at each horizon; "–" denotes unavailable results. Store is retained inference-time parameters (M; lower is better), excluding training-only buffers such as projection bases, at T=40 for ImageNet-R and T=200 for OmniBench-1K. For ours it counts the store (m=15,000), the dictionary atoms, and one prototype set; the OmniBench-1K T=200 entry uses k=1 atom per layer, while all other horizons use k=2.

Method	Venue	ImageNet-R: $\mathcal{A}_{Avg}/\mathcal{A}_{Last}$				OmniBench-1K: $\mathcal{A}_{Avg}/\mathcal{A}_{Last}$				
		T=10	T=20	T=40	Store↓	T=50	T=100	T=125	T=200	Store↓
LwF	ECCV'16	38.12 / 16.78	22.84 / 7.35	12.42 / 4.03	∼0	– / –	– / –	– / –	– / –	∼0
DualPrompt	ECCV'22	75.08 / 69.23	72.65 / 66.27	67.18 / 59.47	0.9M	62.68 / 50.38	60.08 / 47.18	58.44 / 45.24	52.98 / 36.28	0.9M
L2P	CVPR'22	77.96 / 72.35	75.75 / 70.12	71.99 / 65.12	0.2M	61.67 / 49.57	59.20 / 47.40	58.64 / 46.04	55.13 / 43.43	0.2M
LAE	ICCV'23	76.27 / 69.73	71.59 / 64.52	60.93 / 50.67	0.5M	62.50 / 48.80	55.37 / 40.87	52.34 / 36.64	42.19 / 25.99	0.5M
CODA-Prompt	CVPR'23	78.47 / 73.40	75.97 / 70.30	67.37 / 62.00	4.0M	58.03 / 53.90	55.71 / 50.50	52.80 / 47.95	45.82 / 40.36	4.5M
InfLoRA	CVPR'24	80.42 / 74.80	75.11 / 70.20	69.02 / 63.20	14.9M	57.89 / 53.07	51.33 / 46.39	49.73 / 43.27	43.83 / 39.21	74.5M
EASE-21k	CVPR'24	81.23 / 75.90	79.29 / 74.40	76.11 / 70.20	53.7M	61.85 / 53.73	59.66 / 52.90	58.82 / 51.10	58.03 / 49.47	∼196M
SD-LoRA	ICLR'25	77.03 / 70.20	68.41 / 60.80	57.53 / 47.10	14.9M	55.93 / 45.82	50.27 / 34.34	38.33 / 27.79	29.93 / 13.24	74.5M
TUNA	ICCV'25	84.17 / 79.40	82.30 / 77.20	80.01 / 75.60	12.6M	**75.87** / 61.16	72.15 / 59.75	72.32 / 59.30	70.03 / 57.81	∼41.3M
GR-LoRA	ICML'26	**86.76 / 81.80**	**84.77 / 79.43**	**82.45 / 76.10**	35.2M	75.05 / 60.01	**74.26** / 58.06	**73.08** / 56.90	71.39 / 53.64	∼73.3M
SplitLoRA	ICLR'26	86.45 / 81.07	83.92 / 77.90	79.99 / 72.20	14.3M	69.50 / 57.59	67.13 / 55.11	66.20 / 54.25	65.13 / 54.37	14.9M
Ours	—	81.31 / 73.17	79.05 / 72.77	77.26 / 70.39	3.1M	73.30 / **62.70**	73.42 / **62.47**	72.87 / **62.64**	**72.31 / 61.27**	8.2M

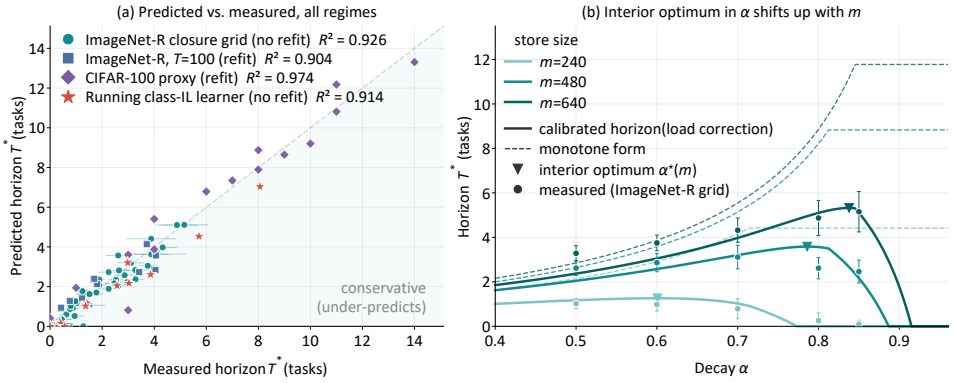


Figure 3: **Recovery-horizon validation and the effect of decay. (a)** Predicted versus measured forgetting horizons for the ImageNet-R closure grid, the T=100 ImageNet-R stream, a CIFAR-100 proxy, and the running class-incremental learner; the legend specifies calibration and R^2 for each setting. The diagonal denotes exact agreement; points below it are conservative forecasts. **(b)** Horizon versus decay α for $m \in \{240, 480, 640\}$. The calibrated horizon (solid lines) captures an interior optimum that shifts toward larger α as memory grows, whereas the monotone form (dashed lines) does not. Circles show measured ImageNet-R horizons, and triangles mark the forecast's optima.

4.4 Accuracy–Storage Trade-offs Without a Cap

Removing the cap separates the benefit of bounded memory from the peak accuracy attainable with more state; Figure 4 plots average accuracy against storage. On ImageNet-R, GR-LoRA leads both metrics at all horizons, reaching $82.45/76.10\%$ at T=40 with 35.2M parameters against $77.26/70.39\%$ with 3.1M for ours, a 5.19-point gain at $11\times$ the storage; TUNA gains 2.75 points at $4\times$. This is a storage–accuracy trade-off rather than unconstrained state of the art. On OmniBench-1K, TUNA leads average accuracy at T=50 and GR-LoRA at T=100 and 125, while ours leads final accuracy throughout and both metrics at T=200 (0.92 above GR-LoRA in average and 3.46 above TUNA in final accuracy) with 8.2M parameters against 41.3M and 73.3M. Our entries are the same runs as in Table 1: the cap changes the baselines' results, not ours.

4.5 Forecast Accuracy, Calibration, and Transfer

We test whether the calibrated horizon predicts recovery deterioration as memory, decay, and setting change (Figure 3; all results in Table 5). The load correction captures the decay optimum. On the ImageNet-R calibration grid, the corrected form attains $R^2 = 0.985$, against 0.940 for the monotone

alternative, and only it rises and then falls with α, with an optimum that shifts toward slower decay as m grows (Figure 3(b)); the monotone form captures neither effect.

The forecast also transfers across settings. Reusing the ImageNet-R constants without refitting gives $R^2 = 0.926$ on the eight-order closure grid and 0.914 on the running class-incremental learner; the refitted series reach 0.904 on the T=100 ImageNet-R stream and 0.974 on the CIFAR-100 proxy, while prompt-pool fits remain weak ($R^2 \approx 0.11$). The error-to-accuracy bridge is only a diagnostic, since discarded codes are unavailable at deployment; the deployed policy uses T^* alone.

4.6 Does acting on the forecast prevent harmful adapter use?

The age-only gate raises post-horizon accuracy by 14.1, 10.9, and 3.1 points under forced forgetting ($\alpha = 0.5$; $m = 120, 240, 480$) on a 10-task ImageNet-R learner over four random class orders. Under gentle decay ($m = 800$, $\alpha = 0.85$) it discards useful adapters at ages 8–9, and some admitted adapters underperform frozen, so it guarantees fallback, not accuracy (Appendix B.4).

4.7 Component and readout ablations

Table 4: **Readout ablation on OmniBench-1K at** $T = 50$, $\mathcal{A}_{Avg}/\mathcal{A}_{Last}$ (%).

Frozen NCM	$63.57/52.60$	Oracle task-ID NCM	$75.89/64.53$
Ours: ID-free union NCM	$73.30/62.70$	Joint linear probe (upper bound)	$78.57/67.23$

We first vary the readout, holding the consolidated adapters fixed (Table 4). The deployed ID-free union NCM, the Table 1 configuration, gains $9.73/10.10$ points over frozen NCM; an oracle task-ID NCM adds a further $2.59/1.83$ and an offline joint linear probe $5.27/4.53$, neither available to the deployed learner, so the readout headroom is smaller than the gain from consolidation itself.

The remaining ablations isolate the sparse code and joint recovery. At about two atoms per layer, the pooled-SVD dictionary retains 98% of the adapter's gain over frozen, against 0% for a random dictionary of the same size (Appendix A.7); on a synthetic grid, the joint-ℓ_1 horizon reaches 12 tasks as m grows, whereas a matched-filter decoder stays at 2–3 tasks (Appendix A.5).

5 Conclusion

Continual learners built on pre-trained backbones usually pay for stability with storage, keeping one adapter per task until the total grows with the length of the stream. We asked what happens when that total is not allowed to grow, and answered with PALIMPSEST: each task's adapter is sparse-coded and written, with decay, into one store of fixed size; a single joint sparse-recovery solve reads candidate adapters back out; and a calibrated horizon, motivated by support-recovery analysis, decides how far into the past those candidates can be trusted before the learner falls back to the frozen backbone. Under fixed storage caps, this design leads both accuracy metrics at the longest horizons on ImageNet-R and OmniBench-1K, by about seven points at 200 tasks, while its own retained state stays below the caps. The horizon forecast transfers to a running learner without refitting, and gate interventions show that acting on it recovers 3 to 14 points of old-task accuracy where reconstructions have become harmful, and where it costs accuracy by being too conservative. Forgetting is therefore bounded by construction, predictable from the store size and decay before any data arrive, and explicit in the serving policy. We see this as a step toward continual learners whose memory is set once and whose behavior at its edge can be reasoned about in advance.

6 Limitations

The main comparisons use one class order and at most 200 tasks, and the capped baselines depend on our closure policies. Separate task state wins at short horizons. The horizon relies on domain-calibrated constants and does not transfer to prompt-pool settings, and the gate guarantees fallback rather than an accuracy floor. The fixed store bounds task history only: the dictionary and prototypes still grow slowly, and at 200 tasks we keep one dictionary atom per layer.

REPRODUCIBILITY STATEMENT

All experiments run in the LAMDA-PILOT toolbox on a frozen ViT-B/16-IN21K backbone with the splits, seeds and closure policies of §4.1 and §4.3. The forecast's constants, the closure-grid data, and the gate-evaluation data behind Figure 5 are released with the code. Appendix A contains the proof sketches and states which parts of the recovery model are calibrated rather than proved.

USE OF AI STATEMENT

Large language models were used to edit the prose for clarity and grammar, to format tables, and to assist with scripts for figure generation. They were not used to design the method or experiments, to produce or analyze results, or to derive proofs. All results, claims, and proofs were verified by the authors.

ETHICS STATEMENT

This work studies memory-bounded continual learning on standard image benchmarks and raises no ethical concerns beyond those of the benchmarks themselves.

REFERENCES

Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.*, 54(11):4311–4322, 2006.

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, 2018.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *NeurIPS*, 2019.

Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin. A simple proof of the restricted isometry property for random matrices. *Constr. Approx.*, 28(3):253–263, 2008.

Emmanuel J. Candès and Terence Tao. Decoding by linear programming. *IEEE Trans. Inf. Theory*, 51(12):4203–4215, 2005.

Emmanuel J. Candès, Justin K. Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Commun. Pure Appl. Math.*, 59(8):1207–1223, 2006.

Venkat Chandrasekaran, Benjamin Recht, Pablo A. Parrilo, and Alan S. Willsky. The convex geometry of linear inverse problems. *Found. Comput. Math.*, 12(6):805–849, 2012.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K. Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

Arslan Chaudhry, Naeemullah Khan, Puneet Dokania, and Philip Torr. Continual learning in low-rank orthogonal subspaces. In *NeurIPS*, 2020.

Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. AdaptFormer: Adapting vision transformers for scalable visual recognition. In *NeurIPS*, 2022.

Rajas Chitale, Ankit Vaidya, Aditya Kane, and Archana Ghotkar. Task arithmetic with LoRA for continual learning. *arXiv preprint arXiv:2311.02428*, 2023.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE TPAMI*, 44(7):3366–3385, 2022.

David Donoho and Jared Tanner. Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing. *Phil. Trans. R. Soc. A*, 367(1906):4273–4293, 2009.

David L. Donoho. Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306, 2006.

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 *ICLR*, 2021.

Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. DyTox: Transformers for continual learning with dynamic token expansion. In *CVPR*, 2022.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *AISTATS*, 2020.

Robert M. French. Catastrophic forgetting in connectionist networks. *Trends Cogn. Sci.*, 3(4): 128–135, 1999.

Qiankun Gao, Chen Zhao, Yifan Sun, Teng Xi, Gang Zhang, Bernard Ghanem, and Jian Zhang. A unified continual learning framework with general parameter-efficient tuning. In *ICCV*, 2023.

Dipam Goswami, Yuyang Liu, Bartłomiej Twardowski, and Joost van de Weijer. FeCAM: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. In *NeurIPS*, 2023.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, 2021.

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 *ICML*, 2019.

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 *ICLR*, 2022.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *ICLR*, 2023.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, 2022.

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. *PNAS*, 114(13):3521–3526, 2017.

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

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *EMNLP*, 2021.

Jiashuo Li, Shaokun Wang, Bo Qian, Yuhang He, Xing Wei, Qiang Wang, and Yihong Gong. Dynamic integration of task-specific adapters for class incremental learning. In *CVPR*, 2025.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL*, 2021.

Zhizhong Li and Derek Hoiem. Learning without forgetting. In *ECCV*, 2016.

Yan-Shuo Liang and Wu-Jun Li. InfLoRA: Interference-free low-rank adaptation for continual learning. In *CVPR*, 2024.

Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. TRGP: Trust region gradient projection for continual learning. In *ICLR*, 2022.

Yipeng Lin, Fengqiang Wan, and Yang Yang. GR-LoRA: Gradient-recycling low-rank adaptation for class-incremental learning. In *ICML*, 2026.

Meng Lou, Yunxiang Fu, and Yizhou Yu. Scaling continual learning to 300+ tasks with bi-level routing mixture-of-experts. *arXiv preprint arXiv:2602.03473*, 2026.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *ECCV*, 2018.

Daniel Marczak, Bartłomiej Twardowski, Tomasz Trzciński, and Sebastian Cygert. MagMax: Leveraging model merging for seamless continual learning. In *ECCV*, 2024.

Imad Eddine Marouf, Subhankar Roy, Enzo Tartaglione, and Stéphane Lathuilière. Weighted ensemble models are strong continual learners. In *ECCV*, 2024.

Michael S. Matena and Colin A. Raffel. Merging models with fisher-weighted averaging. In *NeurIPS*, 2022.

James L. McClelland, Bruce L. McNaughton, and Randall C. O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychol. Rev.*, 102(3):419–457, 1995.

Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychol. Learn. Motiv.*, 24:109–165, 1989.

Mark D. McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton van den Hengel. RanPAC: Random projections and pre-trained models for continual learning. In *NeurIPS*, 2023.

Bruno A. Olshausen and David J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

Vardan Papyan, Yaniv Romano, and Michael Elad. Convolutional neural networks analyzed via convolutional sparse coding. *JMLR*, 18(83):1–52, 2017.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Netw.*, 113:54–71, 2019.

Haomiao Qiu, Miao Zhang, Ziyue Qiao, Weili Guan, Min Zhang, and Liqiang Nie. SplitLoRA: Balancing stability and plasticity in continual learning through gradient space splitting. In *ICLR*, 2026.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In *CVPR*, 2017.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *ICLR*, 2021.

James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. CODA-Prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *CVPR*, 2023.

Hai-Long Sun, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. PILOT: A pre-trained model-based continual learning toolbox. *arXiv preprint arXiv:2309.07117*, 2023.

Hai-Long Sun, Da-Wei Zhou, Hanbin Zhao, Le Gan, De-Chuan Zhan, and Han-Jia Ye. MOS: Model surgery for pre-trained model-based class-incremental learning. In *AAAI*, 2025.

Yuwen Tan, Qinhao Zhou, Xiang Xiang, Ke Wang, Yuchuan Wu, and Yongbin Li. Semantically-shifted incremental adapter-tuning is a continual ViTransformer. In *CVPR*, 2024.

Joel A. Tropp. Algorithms for simultaneous sparse approximation. part ii: Convex relaxation. *Signal Process.*, 86(3):589–602, 2006.

Joel A. Tropp and Anna C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Inf. Theory*, 53(12):4655–4666, 2007.

Liyuan Wang, Jingyi Xie, Xingxing Zhang, Mingyi Huang, Hang Su, and Jun Zhu. Hierarchical decomposition of prompt-based continual learning: Rethinking obscured sub-optimality. In *NeurIPS*, 2023a.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE TPAMI*, 46(8):5362–5383, 2024.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning. In *Findings of EMNLP*, 2023b.

Yan Wang, Da-Wei Zhou, and Han-Jia Ye. Integrating task-specific and universal adapters for pre-trained model-based class-incremental learning. In *ICCV*, 2025.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. DualPrompt: Complementary prompting for rehearsal-free continual learning. In *ECCV*, 2022a.

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 *CVPR*, 2022b.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *ICML*, 2022.

Yichen Wu, Hongming Piao, Long-Kai Huang, Renzhen Wang, Wanhua Li, Hanspeter Pfister, Deyu Meng, Kede Ma, and Ying Wei. SD-LoRA: Scalable decoupled low-rank adaptation for class incremental learning. In *ICLR*, 2025.

Shipeng Yan, Jiangwei Xie, and Xuming He. DER: Dynamically expandable representation for class incremental learning. In *CVPR*, 2021.

Guanxiong Zeng, Yang Chen, Bo Cui, and Shan Yu. Continual learning of context-dependent processing in neural networks. *Nat. Mach. Intell.*, 1(8):364–372, 2019.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, 2017.

Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. SLCA: Slow learner with classifier alignment for continual learning on a pre-trained model. In *ICCV*, 2023.

Yuanhan Zhang, Zhenfei Yin, Jing Shao, and Ziwei Liu. Benchmarking omni-vision representation through the lens of visual realms. In *ECCV*, 2022.

Da-Wei Zhou, Qi-Wei Wang, Han-Jia Ye, and De-Chuan Zhan. A model or 603 exemplars: Towards memory-efficient class-incremental learning. In *ICLR*, 2023.

Da-Wei Zhou, Hai-Long Sun, Jingyi Ning, Han-Jia Ye, and De-Chuan Zhan. Continual learning with pre-trained models: A survey. In *IJCAI*, 2024a.

Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *CVPR*, 2024b.

Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Class-incremental learning: A survey. *IEEE TPAMI*, 46(12):9851–9873, 2024c.

Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *IJCV*, 133: 1012–1032, 2025.

Huiping Zhuang, Zhenyu Weng, Hongxin Wei, Renchunzi Xie, Kar-Ann Toh, and Zhiping Lin. ACIL: Analytic class-incremental learning with absolute memorization and privacy protection. In *NeurIPS*, 2022.

A Full theory: assumptions, theorems, and proof sketches

This appendix gives the rigorous statements and proof sketches behind the calibrated horizon of §3, Eq. (3); the main text states the forms, here we derive them and separate, at every step, what is a theorem (the *certified* horizon of §A.4) from what is a calibrated model.

A.1 Setup: continual learning as recovery from a bounded decay store

We write $[n] = \{1, \dots, n\}$, x_S for the restriction of x to $S \subseteq [p]$, and $\mathrm{supp}(x) = \{i : x_i \neq 0\}$. $\log_{1/\alpha}(\cdot)$ is logarithm to base $1/\alpha$.

Objects. A task $t \in [T]$ has a code $x_t \in \mathbb{R}^p$ that is s-sparse in a dictionary Ψ; its *core* is $S_t = \mathrm{supp}(x_t)$, $|S_t| = s$. Consolidation of task t is a linear measurement $\Phi_t \in \mathbb{R}^{m \times p}$. The *age* of task t at time T is $a_t := T - t$ (most recent task has age 0).

Assumption 1 (Signal model)**.** Supports $S_1, \dots, S_T$ are drawn i.i.d. uniformly among s-subsets of $[p]$. Nonzeros obey $\min_{i \in S_t} |x_{t,i}| \geq x_{\min} > 0$ and $\|x_t\|_\infty \leq 1$. Unless stated, $\Psi = I$.

Assumption 2 (Consolidation operator)**.** $\Phi_1, \dots, \Phi_T$ have i.i.d. $\mathcal{N}(0, 1/m)$ entries and are mutually independent (random consolidation; the structured/learned case is in §A.8).

Assumption 3 (Bounded decay store — the fixed budget)**.** Tasks are written online into a store of fixed dimension m with leak $\alpha \in (0, 1]$:

$$z_t = \alpha\, z_{t-1} + \Phi_t x_t, \quad z_0 = 0 \qquad \Longrightarrow \qquad z := z_T = \sum_{t=1}^{T} \alpha^{\,a_t}\, \Phi_t x_t \;\in\; \mathbb{R}^m.$$

The budget is m numbers *regardless of* T: this is the deployment regime. Forgetting is forced jointly by the finite m and the decay $\alpha \leq 1$.

Assumption 4 (Recovery by certificate)**.** Let $A = [\Phi_1 | \cdots | \Phi_T] \in \mathbb{R}^{m \times pT}$ and $w^\star = (\alpha^{a_1} x_1; \dots; \alpha^{a_T} x_T) \in \mathbb{R}^{pT}$, so that $z = Aw^\star$. The certificate decoder solves basis pursuit (joint ℓ_1)

$$\hat{w} = \arg\min_{w} \|w\|_1 \;\; \text{s.t.} \;\; Aw = z, \tag{4}$$

(or its BPDN/LASSO relaxation under noise) and reads off $\hat{x}_t = \hat{w}_{[t]}$, the t-th length-p block; $\widehat{S}_t$ is the support of its largest-s entries.

Remark 1 (The reduction)**.** (i) With Φ_t independent Gaussian, A is an $m \times N$ Gaussian matrix, $N := pT$. (ii) The decay folds into the coefficients: $w^\star$ has a *recency-ordered, geometrically decaying* magnitude profile (block t scaled by α^{a_t}), so it is *compressible* rather than exactly sparse. Bounded-memory continual learning is therefore a *single* compressed-sensing problem with ambient dimension N, a recency profile, and m measurements; all continual-learning content is in *which blocks are recoverable* as a function of $(m, s, p, \alpha, T, x_{\min})$.

Real instantiation (for calibration). Frozen ViT-B/16 (ImageNet-21k); each task fits an independent rank-r LoRA adapter $\Delta W_t = B_t A_t$ on the q/v projections, playing the role of the per-task operator; ΔW_t is coded into a pooled-SVD dictionary ($k \approx 2$ atoms/layer) and read out as in §3. Empirically these per-task operators are near-orthogonal (operator overlap $\rho_{\mathrm{op}} \approx 0.017$, $\sim 4\times$ chance), so the incoherence the certificate needs holds on real adapters.

A.2 THE SINGLE-TASK CERTIFICATE

Definition 1 (RIP). A has the restricted isometry property of order k with constant δ_k if $(1-\delta_k)\,\|v\|_2^2 \le \|Av\|_2^2 \le (1+\delta_k)\,\|v\|_2^2$ for all k-sparse v.

Theorem 1 (Single-task safe-forgetting rate). *Let $T=1, \alpha=1$ and Φ as in Assumption 2. There are universal $C, c>0$ such that if $m \ge C\,s\log(p/s)$, then with probability $\ge 1-e^{-cm}$, $\delta_{2s}(\Phi) < \sqrt{2}-1$ and basis pursuit (4) recovers x exactly ($\widehat{S}=S$). Conversely, if $m \le c\,s\log(p/s)$, exact recovery fails w.p. $\to 1$. In CL terms: a task with core size s is safely forgettable down to any store $m \ge C\,s\log(p/s)$, and not below.*

Proof sketch. Order-$2s$ RIP with the stated constant for an $m \times p$ Gaussian matrix holds w.h.p. once $m \gtrsim s\log(p/s)$ (Gaussian-width bound; Baraniuk et al., 2008; Chandrasekaran et al., 2012). $\delta_{2s} < \sqrt{2}-1$ implies exact BP recovery (Candès & Tao, 2005). The converse is the ℓ_1 phase transition (Donoho & Tanner, 2009). The constant is the ℓ_1 value; OMP would need $m \gtrsim s\log p$ (Tropp & Gilbert, 2007), so the operator is fixed as ℓ_1/BP throughout. □

A.3 THE STABLE-RECOVERY CERTIFICATE

We restate the classical stable-recovery guarantee in the sequential setting; it is the mathematical core from which the horizon follows.

Lemma 1 (Stable recovery of the recency-weighted code). *Let $N=pT$. Since the blocks Φ_t are independent Gaussian, $A \in \mathbb{R}^{m\times N}$ is Gaussian, so with probability $\ge 1-N^{-c}$ it satisfies RIP up to sparsity*

$$k_{\max} \asymp \frac{m}{\log(N/k_{\max})} \approx \frac{m}{\log(pT)}. \tag{5}$$

Then for every $k \le k_{\max}$, basis pursuit (4) obeys the stable-recovery bound

$$\|\hat{w}-w^\star\|_2 \le C\,\frac{\|w^\star - w_k^\star\|_1}{\sqrt{k}}, \tag{6}$$

where $w_k^\star$ is the best k-term approximation of $w^\star$.

Proof sketch. RIP of order $2k$ with $\delta_{2k} < \sqrt{2}-1$ for the $m \times N$ Gaussian matrix holds w.h.p. for $k \lesssim m/\log(N/k)$ (Baraniuk et al., 2008); (6) is then the stable-recovery bound of Candès et al. (2006) for basis pursuit. □

Equation (6) is the certificate: recovery quality is set by the *tail mass* left after the largest k coefficients are kept. Because $w^\star$ is recency-ordered (Remark 1), keeping the largest coefficients means keeping the most recent task blocks, and the tail is the geometric mass of the older ones.

A.4 THE CERTIFIED SAFE-FORGETTING HORIZON

A single basis-pursuit solve operates at *one* error floor, set by the tail beyond the budget's capacity $k_{\max}$, and *every* block is judged against that same floor. This is what makes the horizon both well-posed and non-circular.

Definition 2 (Capacity, the recovery floor, the certified horizon). Let $k_{\max}$ be the RIP capacity (5) and write $a_{\text{cap}} := k_{\max}/s$. Define the single *post-recovery floor* of the bounded store as the stable-recovery error at the operating capacity,

$$\varepsilon^\star := C\,\frac{\left\|w^\star - w^\star_{k_{\max}}\right\|_1}{\sqrt{k_{\max}}}. \tag{7}$$

The *certified safe-forgetting horizon* is

$$\boxed{T^* := \max\left\{\, a \ge 0 \;:\; s\,(a+1) \le k_{\max} \text{ and } \alpha^a\, x_{\min} > 2\varepsilon^\star \,\right\}.} \tag{8}$$

Remark 2 (Why (8) is non-circular)**.** The floor $\varepsilon^\star$ depends only on the capacity $k_{\max}$ (a property of A, hence of m, p, T) and the decay profile of $w^\star$; it does *not* reference T^*. For each candidate age a we check two primitive conditions — a *capacity* condition (the $a{+}1$ most recent tasks fit under $k_{\max}$ coordinates) and a *floor* condition (the age-a trace clears $2\varepsilon^\star$) — and take the largest a passing both. Crucially the floor is evaluated at the fixed operating capacity $k_{\max}$, *not* adaptively per age: the decoder runs one BP solve, so all blocks see the same $\varepsilon^\star$, which is what gives the correct "older = harder" ordering. The capacity condition $s(a+1) \le k_{\max}$ is not a second floor but the *validity regime* of Lemma 1: once more than $k_{\max}$ coordinates are significant the RIP premise behind $\varepsilon^\star$ fails, so the horizon is capped at $a_{\text{cap}} = k_{\max}/s$ regardless of the floor.

Proposition 1 (Certified support recovery)**.** *Condition on the RIP event of Lemma 1 at order $2k_{\max}$. If task t satisfies the floor condition $\alpha^{a_t} x_{\min} > 2\varepsilon^\star$, then its core is recovered by the largest-s selection on $\hat{w}_{[t]}$, i.e. $\widehat{S}_t = S_t$.*

Proof. By (6) at $k = k_{\max}$, $\|\hat{w} - w^\star\|_2 \le \varepsilon^\star$, hence $\|\hat{w} - w^\star\|_\infty \le \varepsilon^\star$. For $i \in S_t$, $\left|\hat{w}_{[t],i}\right| \ge \alpha^{a_t} x_{\min} - \varepsilon^\star$; for $j \notin S_t$, $\left|\hat{w}_{[t],j}\right| \le \varepsilon^\star$. Every true coordinate ranks above every false one whenever $\alpha^{a_t} x_{\min} - \varepsilon^\star > \varepsilon^\star$, i.e. $\alpha^{a_t} x_{\min} > 2\varepsilon^\star$. □

This is a *sufficient* condition; the $\ell_\infty \le \ell_2$ step makes it conservative (§A.6, §A.8). The capacity condition $s(a_t + 1) \le k_{\max}$ does *not* enter this per-block argument: it belongs to the *definition* (8) of T^*, where it both caps the horizon at a_{cap} and marks the regime in which $\varepsilon^\star$ is a valid floor (Remark 2).

A.4.1 Two-regime approximation

Corollary 1 (Two-regime form)**.** *The recency ordering gives $\left\|w^\star - w^\star_{k_{\max}}\right\|_1 \lesssim s\,\alpha^{a_{\text{cap}}}/(1-\alpha)$ (the geometric tail of blocks older than a_{cap}), so*

$$\varepsilon^\star \asymp \frac{C\sqrt{s}\;\alpha^{a_{\text{cap}}}}{(1-\alpha)\sqrt{a_{\text{cap}}}}, \qquad a_{\text{cap}} \asymp \frac{c_1\, m}{s\,\log(pT)}. \tag{9}$$

The capacity condition in (8) caps a at $a_{\text{cap}} - 1$; the floor condition $\alpha^a x_{\min} > 2\varepsilon^\star$ caps a at

$$a_{\text{dec}} \;\approx\; \log_{1/\alpha} \frac{x_{\min}}{2\varepsilon^\star} \;=\; a_{\text{cap}} + \frac{1}{\log(1/\alpha)} \log \frac{x_{\min}(1-\alpha)\sqrt{a_{\text{cap}}}}{2C\sqrt{s}}. \tag{10}$$

Hence

$$T^* \;\approx\; \min\{a_{\text{cap}},\, a_{\text{dec}}\}. \tag{11}$$

Capacity-limited *($a_{\text{dec}} \ge a_{\text{cap}}$): $T^* \asymp a_{\text{cap}} \asymp m/(s \log pT)$, linear in m up to the log.* Decay-limited *($a_{\text{dec}} < a_{\text{cap}}$): $T^* \asymp a_{\text{dec}} < a_{\text{cap}}$.*

Equation (11) should be read as a closed-form *summary* of the fixed-point (8), not as an independent theorem; the rigorous object is the maximization, and $a_{\text{cap}}, a_{\text{dec}}$ are derived quantities, both functions of the primitives only.

A.4.2 The no-decay case is not zero horizon

Remark 3 ($\alpha = 1$: the age-horizon notion is the wrong object)**.** When $\alpha = 1$ every block has equal magnitude, so $w^\star$ has *no* recency ordering and there is no age-ordered forgetting frontier. The floor condition in (8) is then not the binding object: if total sparsity is below capacity, $sT \lesssim m/\log(pT)$, all task codes may be recovered simultaneously; if it exceeds capacity, recovery can fail, but the failure is *not* ordered by age. Thus at $\alpha = 1$ the horizon is not "$T^* = 0$"; rather the age-based horizon is inapplicable and the relevant statement is the global capacity condition

$$sT \;\lesssim\; \frac{m}{\log(pT)}.$$

Decay $\alpha < 1$ is precisely what converts the unordered capacity limit into an *ordered*, predictable, oldest-first frontier.

Remark 4 (Calibration)**.** The synthetic grid ($p = 128, s = 3, \alpha = 0.8$) gave horizons $\{1, 2, 3, 5, 7, 9, 12\}$ over $m \in \{96, \ldots, 320\}$ — monotone and super-linear, the predicted $a_{\text{dec}} \to a_{\text{cap}}$ crossover; the $\alpha = 1$ run was age-*independent*, exactly the regime of Remark 3. On real ViT features the a_{cap} branch transfers with the synthetic-calibrated constant ($c_1 \approx 0.85$ vs. 1.19), but the a_{dec} floor *under*-predicts measured horizons — the $\ell_\infty \leq \ell_2$ looseness made visible, and the motivation for §A.6.

A.5 Separation from a capacity-blind decoder

The horizon theorem says capacity *can* be converted into recoverable age range by joint recovery. We now show this conversion is the certificate's doing: a decoder that ignores cross-task structure has a much shorter horizon, and the gap grows with the budget. We are careful about what the separation does and does not say.

Definition 3 (Capacity-blind (matched-filter) decoder)**.** The naive decoder back-projects, $\tilde{x}_t = \Phi_t^\top z$, and returns the top-$s$ support of $\tilde{x}_t$ — it never solves the joint problem (4).

Proposition 2 (Blind horizon is only logarithmic in m)**.** *Under Assumptions 1–3, the matched-filter decoder recovers task t iff $\alpha^{a_t} x_{\min} \gtrsim \eta_{\text{MF}} := \sqrt{s \log p/(m(1-\alpha^2))}$, and this is sufficient up to constants. Hence its horizon is*

$$a_{\text{MF}} \asymp \frac{1}{2\log(1/\alpha)} \log\Big(\frac{x_{\min}^2\, m\,(1-\alpha^2)}{s \log p}\Big) = \Theta(\log m).$$

Proof sketch. $\Phi_t^\top z = \Phi_t^\top \Phi_t(\alpha^{a_t} x_t) + \sum_{t' \neq t} \alpha^{a_{t'}} \Phi_t^\top \Phi_{t'} x_{t'}$. The first term is $\alpha^{a_t} x_t$ plus an $O(\sqrt{s/m})$ Gram perturbation; the cross terms are mean-zero with per-coordinate variance $\leq s/(m(1-\alpha^2))$, so their maximum over p coordinates is $\Theta(\eta_{\text{MF}})$. Top-s selection succeeds iff the signal exceeds it. □

Proposition 3 (Certified margin grows with the budget)**.** *In the capacity-limited regime $a_{\text{cap}} \leq a_{\text{dec}}$, the recoverable age range of the certificate exceeds that of the matched filter by*

$$T^* - a_{\text{MF}} \asymp \frac{c_1 m}{s \log pT} - \Theta(\log m) \xrightarrow[m\to\infty]{} \infty, \qquad \frac{T^*}{a_{\text{MF}}} \asymp \frac{m/(s \log pT)}{\log m} \to \infty.$$

This margin is a function of $(m, \alpha, s, p, T, x_{\min})$ only and is therefore computable a priori.

Proof. Subtract Proposition 2 from Corollary 1; in the capacity-limited regime $T^* \asymp a_{\text{cap}} \asymp m/(s \log pT)$ while $a_{\text{MF}} = \Theta(\log m)$. Every quantity entering both horizons is a primitive. □

Remark 5 (What the separation does *not* claim)**.** We do *not* claim the matched filter loses all old tasks *simultaneously*. Under decay $\alpha < 1$ the blind decoder still has an age-ordered threshold, because the signal α^{a_t} decreases with age: it forgets gracefully too, just from a much shorter horizon $a_{\text{MF}} = \Theta(\log m)$. The formal conclusion is the *margin* of Proposition 3: joint recovery keeps a recoverable age range that outgrows the blind decoder's as the budget grows. We also stress the scope: the matched filter is a pedagogical *blind* baseline; Propositions 2–3 establish the value of *joint recovery over back-projection*, and are *not* a model of, or a performance claim against, deployed rivals (InfLoRA, SD-LoRA, prompt pools).

Remark 6 (Where genuinely catastrophic collapse comes from — motivation, not theorem)**.** Truly *unordered* (catastrophic) collapse arises for a different decoder class: a *hard-allocation* method that assigns each task a fixed, disjoint sub-budget (a slot of a prompt pool, an orthogonal subspace) and carries *no* recency weighting. Once the allocated sub-budget is spent, the overflow tasks share one undifferentiated failure mode with no magnitude ordering to peel, so they degrade together rather than oldest-first. Our model (decayed superposition + joint recovery) does not produce this collapse; demonstrating it for deployed allocation methods (prompt pools, orthogonal-subspace LoRA) on a long, capacity-stressing stream is an *empirical* target, complementary to the theory here. The theory's role is the predictable graceful frontier T^*; the experiment's role is to exhibit the rivals' wall.

Remark 7 (Calibration)**.** The synthetic grid measured the budget-growing margin of Proposition 3: at $\alpha = 0.8$ the matched-filter horizon stayed pinned at $\approx$ 2–3 across $m \in \{96, \ldots, 320\}$ while the ℓ_1 horizon rose to 12; the gap grew monotonically $\{+1, +1, +2, +3, +5, +7, +9\}$.

A.6 LOAD-COLLAPSE REFINEMENT: THE INTERIOR OPTIMUM IN α (CALIBRATED)

The v0 decay branch (10) is *monotone decreasing* in α: the capacity term a_{cap} is α-independent, and the correction $\frac{1}{\log(1/\alpha)} \log \frac{x_{\min}(1-\alpha)\sqrt{a_{\text{cap}}}}{2C\sqrt{s}}$ runs from 0 (as $\alpha \to 0$) to $-\infty$ (as $\alpha \to 1$, where $(1-\alpha) \to 0$). This is the form that attains $R^2 = 0.94$ in Remark 8, and being monotone it *cannot* reproduce a phenomenon seen on both synthetic and real grids: the horizon *rises then falls* with α, with an interior optimum $\alpha^\star \in (0,1)$ that *shifts upward with* m. We give a calibrated physical model and its mechanism; this is explicitly *not* a theorem at the rigor of §A.4 (see §A.8).

(a) Signal-recency horizon (rises with α). Slower decay sees further back. With a head-limited floor $\eta \approx c_0\sqrt{s/m}$ (the per-coordinate spread of an s-sparse signal measured into m dimensions, before the over-capacity regime), the recovery condition $\alpha^k x_{\min} > \eta$ gives

$$T_{\text{sig}}(\alpha, m) := \frac{1}{\log(1/\alpha)} \log \frac{x_{\min}\sqrt{m/s}}{c_0}, \tag{12}$$

increasing in α (as $\log(1/\alpha) \downarrow 0$).

(b) Over-capacity collapse (falls as $\alpha \to 1$). The number of *live* (non-negligibly decayed) tasks is the e-folding age of $\alpha^k = e^{-k(1-\alpha)+O((1-\alpha)^2)}$, namely

$$L := \frac{1}{1-\alpha} \quad \text{(effective live-task load)}. \tag{13}$$

Each live task injects s significant coordinates, so the store carries $\approx sL$ significant coordinates against capacity $k_{\max} = s\, a_{\text{cap}}$. Recovery is well-posed only while $L \le a_{\text{cap}}$; as $L \uparrow a_{\text{cap}}$ the effective oversampling ratio $\to 1$ and the store becomes non-invertible. The fractional headroom is $(1 - L/a_{\text{cap}})_+$.

Proposition 4 (Refined horizon with interior optimum (calibrated))**.** *Combining (a)–(b),*

$$\boxed{T^*_{\text{fit}}(m, \alpha) = \max\Big(0,\ \min\{a_{\text{cap}},\ T_{\text{sig}}(\alpha, m)\} \cdot \big(1 - \tfrac{L}{a_{\text{cap}}}\big)_+\Big)} \tag{14}$$

with $a_{\text{cap}} = c_1 m/(s \log pT)$, T_{sig} from (12), $L = 1/(1-\alpha)$, and two domain-calibrated constants (c_1, c_0). As $\alpha \to 1$, $T_{\text{sig}} \to \infty$ but the load factor $\to 0$; as $\alpha \to 0$, $T_{\text{sig}} \to 0$. Hence T^*_{fit} has an interior maximum $\alpha^\star(m) \in (0,1)$, and since $a_{\text{cap}} \propto m$ raises the collapse threshold $L \le a_{\text{cap}}$, $\alpha^\star(m)$ increases with m.

Derivation (heuristic). T_{sig} solves the head-limited condition $\alpha^k x_{\min} = \eta$, $\eta = c_0\sqrt{s/m}$, monotone increasing in α. The load factor is the well-posedness headroom $L \le a_{\text{cap}}$, vanishing at $L = a_{\text{cap}}$. The product of a factor diverging as $\alpha \to 1$ and one vanishing there — the latter's zero crossing $\alpha = 1 - 1/a_{\text{cap}}$ pushed toward 1 as $a_{\text{cap}} \propto m$ grows — yields an interior maximum whose location increases with m. The capacity ceiling $\min\{a_{\text{cap}}, \cdot\}$ caps the rising branch. □

Remark 8 (Calibration — the load-bearing empirical fit)**.** On the measured (m, α) recovery grid the monotone v0 floor (10) attains $R^2 = 0.94$ and cannot produce the interior peak; the refined form (14) attains $R^2 = 0.985$ and reproduces both the interior optimum and its upward m-shift ($\alpha^\star : 0.70 \to 0.80$ across m). Ordering-averaged over 8 task orderings (removing the age-vs-identity confound) it holds at $R^2 = 0.926$, including the two-regime capacity crossover.

Constants and parameterization. The empirical fit uses a *reduced* parameterization $a_{\text{cap}} = C_1 m$ and $T_{\text{sig}} = (\frac{1}{2}\log m + C_2)/\log(1/\alpha)$ — equivalent to (12) with $C_2 = \log x_{\min} - \frac{1}{2}\log s - \log c_0$ bundling $(x_{\min}, s, c_0)$ into one additive signal-floor offset — giving $(C_1, C_2) = (0.0184, -1.253)$. Two cautions on notation: (a) the empirical $C_1 = 0.0184$ (in $a_{\text{cap}} = C_1 m$) is *not* the theory constant $c_1 \approx 0.85$ of (9) (in $a_{\text{cap}} = c_1 m/(s \log pT)$); they differ by the $s \log pT$ factor absorbed into C_1. (b) The second empirical constant is the *additive* offset $C_2 = -1.253$ *inside* T_{sig}, not a multiplicative c_0; in particular $c_0 = x_{\min}\, s^{-1/2} e^{-C_2} > 0$, and there is no sign contradiction with the -1.253 reported elsewhere.

A.7 Recovery → accuracy bridge and the forgetting policy (calibrated)

Sections A.4–A.5 certify *support recovery*; the core claim is about *accuracy*. We close the gap with a calibrated bridge, turning the recovery horizon into an accuracy horizon and the certified margin into a recovered-accuracy margin.

Proposition 5 (Recovery→accuracy bridge (calibrated)). *Let rel-err$_t := \|\hat{x}_t - x_t\|_2 / \|x_t\|_2$ and let $g_t \in [0,1]$ be the fraction of task t's classification gain (over the frozen backbone) retained when the recovered operator $\Delta\hat{W}_t$ is used. There is a domain-calibrated threshold τ such that $g_t \geq \frac{1}{2}$ iff rel-err$_t \lesssim \tau$, with τ in a tight band $[0.77, 0.85]$ (mean ≈ 0.80),* below *the norm-failure point* 1.0. *Consequently the accuracy horizon coincides with T^* evaluated at τ, and the margin of Proposition 3 transfers to a recovered-accuracy margin.*

Remark 9 (Why $\tau < 1$, and calibration). $\tau < 1$ because store leakage is *structured* interference concentrated in the discriminative directions (rank-1 truncation retains 88% of the gain, rank-2 95%), so accuracy fails at a milder recovery error than norm-failure. The band $\tau \approx 0.80$ holds across the (m, α) grid and is the same for the frozen task-IL analysis and the running class-IL learner. End-to-end on the running learner, the frozen-calibrated constants transfer with *no refit*: Arrow 1 (forecast→recovery horizon) $R^2 = 0.914$; Arrow 2 (bridge) accuracy knee at rel-err $= 0.80 \pm 0.06$. The pooled-SVD dictionary is load-bearing: it codes adapters at $k \approx 2$ recovering 98% of the gain, a random dictionary 0%.

Corollary 2 (Reach the ceiling; recover the margin a blind decoder loses). *Combine Proposition 3 with Proposition 5. The certificate* attains *a recoverable — hence, via the bridge, accuracy — horizon up to a_{cap} (Corollary 1, an* achievability *result); whether a_{cap} is also an* upper *bound for an arbitrary fixed-budget decoder is a* converse *we conjecture but do not prove here (deferred, §A.8). The certificate-governed policy — recover by (4), and for a queried task of age a use the recovered operator iff the certificate predicts $a \leq T^*$ (equivalently rel-err $\leq \tau$), else fall back to the frozen readout — attains accuracy up to a_{cap} and degrades smoothly along $T^*(m, \alpha)$. Against a capacity-blind readout it recovers the tasks of age in $(a_{\text{MF}}, T^*]$ by a margin $\asymp m/(s \log pT) - \Theta(\log m)$ that the certificate computes in advance.*

Remark 10 (The gate is the policy). The "use-recovered-iff-$a \leq T^*$" rule is the deployable forgetting policy. Empirically (Figure 5), gating on the calibrated horizon recovers old-task accuracy by $+3$ to $+14$pp on average past T^* in the forced-forgetting regime (strong decay, where unguarded recovery returns harmful noise below the frozen floor), and closes to ≈ 0 where decay is gentle and old operators still beat frozen — the policy helps precisely where the forecast says it should. Gating on the rel-err $\leq \tau$ band rather than raw age removes the gentle-decay over-drop.

A.8 What is proved, what is calibrated, what remains

Proved (up to standard CS results, at proof-sketch rigor).

1. The bounded decay store is a single compressed-sensing problem $z = Aw^\star$ (§A.1).
2. Single-task rate (Theorem 1) and stable recovery $\|\hat{w} - w^\star\|_2 \leq C \|w^\star - w^\star_k\|_1 / \sqrt{k}$ (Lemma 1).
3. Sufficient condition $\alpha^{a_t} x_{\min} > 2\varepsilon^\star$ for support recovery (Proposition 1), and the non-circular certified horizon (8), summarized by $T^* \approx \min\{a_{\text{cap}}, a_{\text{dec}}\}$ (Corollary 1).
4. The matched filter has a $\Theta(\log m)$ horizon (Proposition 2), so the certificate's recoverable age range exceeds it by a budget-growing, a-priori-computable margin (Proposition 3) — *without* claiming simultaneous collapse (Remark 5).
5. At $\alpha = 1$ the age-horizon is inapplicable; the binding object is the global capacity $sT \lesssim m/\log(pT)$ (Remark 3).

Calibrated models (fit to data, not derived from first principles). The load-collapse interior optimum (Proposition 4; $R^2 = 0.985$ on the recovery grid) and the recovery→accuracy bridge (Proposition 5; $\tau \approx 0.80$, end-to-end $R^2 = 0.914$ no-refit). These carry the empirical weight of the paper and are labeled as calibrated.

B ADDITIONAL PROTOCOL DETAILS, TABLES, AND FIGURES

B.1 PROTOCOL DETAILS

Storage accounting. The fixed-budget comparison imposes B=5M retained parameters on ImageNet-R and B=15M on OmniBench-1K. Accounting includes all state beyond the frozen backbone that is needed at inference: adapters, prompts, dictionaries, classifier heads, prototypes, and retained statistics. Training-only buffers are excluded for every method, including InfLoRA's DualGPM bases and SplitLoRA's projection matrices; SplitLoRA's per-task LoRA factors are also excluded because they are already folded into its dense merged update. The store dimension m in recovery experiments is distinct from this total parameter budget B. Without a cap, methods retain their native state, with storage reported at T=40 and T=200 for the two benchmarks. Our configuration is identical in both regimes: its retained state is below both caps, so the cap binds only the baselines.

Reporting conventions. Paired accuracy entries are average / final incremental accuracy, $\mathcal{A}_{Avg}/\mathcal{A}_{Last}$, in percent; all rankings and differences compare like metrics. Missing entries are not treated as zero, and the tables report point estimates rather than significance tests. The main comparisons retain each method's readout; §4.7 varies ours explicitly and separates task-ID-free evaluation from oracle task-ID and offline diagnostics. Forecast quality is measured by R^2 between predicted and measured horizons, not classification accuracy. The reference ImageNet-R calibration is $(C_1, C_2) = (0.0184, -1.253)$. "No refit" keeps these constants fixed; "refit" estimates constants for the stated evaluation setting.

Capacity planning as a consequence of the forecast. For fixed decay and calibrated constants, the horizon defines a candidate store size for a target age: $m^\star = \min\{m : T^*(m, \alpha) \geq T_{\text{target}}\}$, over admissible store sizes. This selection requires no additional learner runs once calibration is available. It is a model-based sizing rule, not an experimentally established minimum-memory guarantee.

B.2 STORAGE–ACCURACY FRONTIER

Figure 4 plots the native results of Table 3 against retained storage, as discussed in §4.4.

B.3 ADDITIONAL FORECAST CALIBRATION AND TRANSFER RESULTS

Table 5 supplements the main-text validation in §4.5 and Figure 3.

Transfer versus refitting. Without refitting, the ImageNet-R constants give $R^2 = 0.926$ on the ImageNet-R closure grid and 0.914 on the running class-incremental learner (Figure 3a), 0.83–0.94 on CIFAR-100 and Aircraft grids, and 0.79–0.85 on T=100 streams. The two refitted series in Figure 3(a) reach 0.904 on the T=100 ImageNet-R stream and 0.974 on the CIFAR-100 proxy. Prompt-pool settings remain weak even after fitting ($R^2 \approx 0.11$), delimiting the model's empirical scope. Points below the diagonal in Figure 3(a) are conservative forecasts.

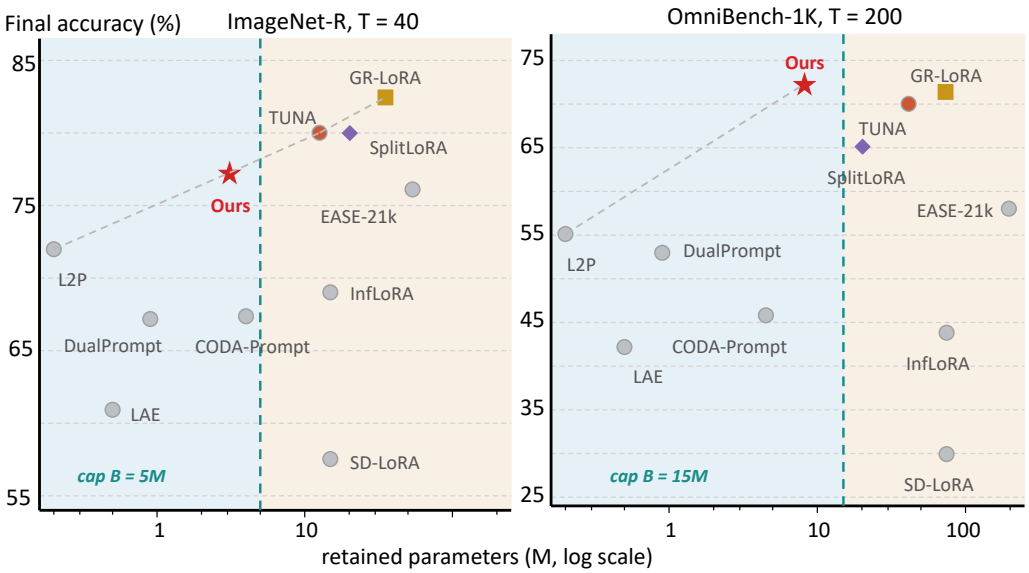


Figure 4: **Storage–accuracy trade-off with native, unconstrained state.** Average accuracy $\mathcal{A}_{Avg}$ from Table 3 is plotted against retained parameters (millions, logarithmic scale) at T=40 on ImageNet-R (left) and T=200 on OmniBench-1K (right). Red stars denote ours; gray dashed lines trace the Pareto frontiers among the plotted methods. Blue and orange regions mark storage below and above the reference caps of 5M and 15M, respectively; these are unconstrained results, not budget-capped evaluations. Ours lies on both frontiers, reaching 77.26% with 3.1M parameters on ImageNet-R and the highest plotted accuracy, 72.31%, with 8.2M on OmniBench-1K.

Table 5: **Forecast accuracy with and without recalibration.** R^2 between predicted and measured horizons. Figure 3(a) displays a subset of these evaluations; the CIFAR-100 proxy and adapter-grid studies are listed separately. "No refit" reuses the ImageNet-R calibration; refitted results assess fit within the stated setting.

Evaluation	Calibration	R^2
ImageNet-R grid (corrected / monotone)	fit	0.985 / 0.940
ImageNet-R closure grid, 8-order average	no refit	0.926
CIFAR-100 / Aircraft grids	no refit	0.83–0.94
Long stream (T=100)	no refit	0.79–0.85
Running learner (end-to-end)	no refit	0.914
ImageNet-R, T=100 (Figure 3a)	refit	0.904
CIFAR-100 proxy (Figure 3a)	refit	0.974
CIFAR-100 adapter grid	refit	0.93–0.94
Prompt-pool capacity	fit	$\approx$ 0.11

B.4 GATE INTERVENTIONS UNDER FORCED FORGETTING AND GENTLE DECAY

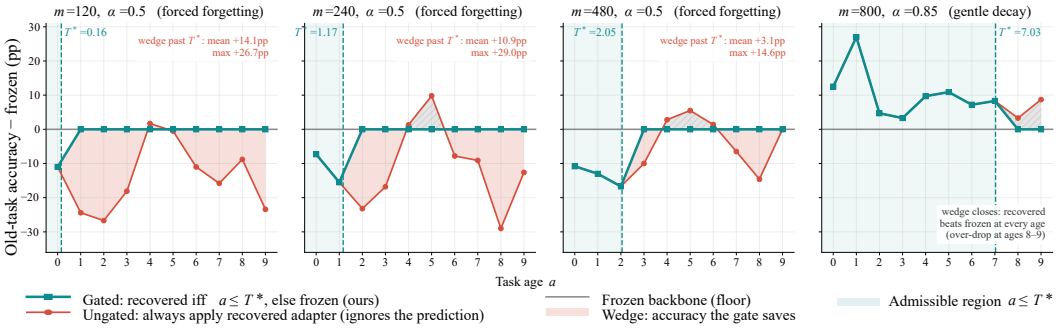


Figure 5: **Prediction-driven gating under forced forgetting and gentle decay.** Old-task accuracy minus frozen-backbone accuracy (percentage points) versus task age on a running 10-task ImageNet-R learner over four random class orders. Teal squares show the gated policy, which serves the recovered adapter for $a \leq T^*$ and otherwise falls back to frozen; red circles always apply the recovered adapter. Dashed vertical lines mark T^*, and pale blue shading marks $a \leq T^*$. In the first three panels ($\alpha = 0.5$, $m = 120, 240, 480$), the pink regions show losses avoided by fallback, with mean gains past T^* of 14.1, 10.9, and 3.1 points. Under gentle decay (right; $m = 800$, $\alpha = 0.85$), recovered adapters outperform frozen at every age, so the age-only gate switches prematurely at ages 8–9.

Gate-on versus gate-off. Figure 5 holds recovery fixed and changes only whether the age threshold is obeyed. In the forced-forgetting settings ($\alpha = 0.5$, $m = 120, 240, 480$), the predicted horizons are 0.16, 1.17, and 2.05 tasks. Beyond these thresholds, the gated policy returns to frozen, while always applying the recovered adapter can depress accuracy by as much as 29 points. The reported mean post-horizon gains from gating are 14.1, 10.9, and 3.1 points, respectively. This isolates the value of deciding *when to stop using* an adapter, rather than improving its reconstruction.

Conservative forecasts also have a cost. With $m = 800$ and $\alpha = 0.85$, recovered adapters outperform frozen at every evaluated age, yet $T^* = 7.03$ still triggers fallback at ages 8–9, discarding useful adaptation. Nor does the gate guarantee the frozen floor inside the trusted region: some served adapters in the first three panels are already below frozen. A reconstruction-error criterion could address this mismatch, but requires the original code or an error estimator; neither is used by the deployed age-only rule. Results use four random class orders on ImageNet-R, and ages within a stream are not independent replicates. A capacity-planning corollary of the forecast is given in Appendix B.1.