SCALING GRADIENT PROJECTION FOR CONTINUAL LEARNING OVER LONG TASK SEQUENCES VIA SPECTRAL CONSOLIDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Gradient-projection methods address catastrophic forgetting in continual learning (CL) by retaining important activation subspaces and restricting subsequent parameter updates to minimally interfere with learned tasks. While these methods are competitive on standard CL benchmarks, their protection mechanisms become overly restrictive over long task sequences. We observe that in fixed-capacity models, protected subspaces formed by accumulating directions from successive tasks eventually saturate and stop admitting new directions. Even when constraints on parameter updates are relaxed, as in soft-projection variants, accumulated importance can ossify initially soft protection into effectively hard constraints. To address these limitations, we introduce Spectral Consolidation Gradient Projection Memory (SC-GPM), which combines activation usage across tasks in a bounded per-layer operator and recomputes the protected directions and their strengths after each task. We formulate the trade-off between protecting past tasks and reserving capacity for new learning as a convex optimization problem and derive an exact spectral solution. The solution reserves a specified amount of capacity for new learning by relaxing protection on less-used directions. Across DomainNet, OmniBenchmark, and ImageNet-1K streams of 100–160 tasks, SC-GPM improves final average accuracy by 2.18–3.64 percentage points over the strongest evaluated gradient-projection baseline on each dataset, with higher learning accuracy on all three. On domain-incremental Permuted MNIST with 200 tasks, SC-GPM outperforms the strongest evaluated gradient-projection baselines by 2.48 percentage points in final average accuracy and 1.24 points in accuracy averaged throughout learning (average anytime accuracy).

1 INTRODUCTION

Continual learning requires a model to preserve earlier knowledge while remaining able to learn from new streams of data (Wang et al., 2024). Unless special care is taken, training on new tasks overwrites previously learned information, a failure known as catastrophic forgetting (McCloskey & Cohen, 1989). Gradient-projection methods address this problem by retaining important activation directions from earlier tasks and restricting subsequent updates along them (Zeng et al., 2019; Saha et al., 2021; Wang et al., 2021).

As more tasks are learned, the protection accumulated for earlier tasks also determines how much the model can change to accommodate later ones. Existing Gradient Projection Memory (GPM)-based methods provide various mechanisms to protect and reuse representations, but are typically evaluated over only 20–40 tasks (Table 1). Over longer sequences, methods that retain previously selected directions have less room for new additions as their protected bases grow. Later tasks inherit these earlier choices, while accumulated importance can ossify initially soft constraints into effectively hard ones (Figure 1(a)). Recomputing the protected directions allows protection to adapt to changing task usage. However, a fixed rule that assigns stronger protection as usage accumulates can still increasingly restrict later updates (Figure 1(b)). This reflects the stability–plasticity trade-off, since protecting earlier tasks can reduce the ability to learn subsequent ones. This raises a central question for gradient-projection methods: how should protection evolve as tasks accumulate so that earlier knowledge is preserved while the model retains sufficient plasticity for later learning?

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.

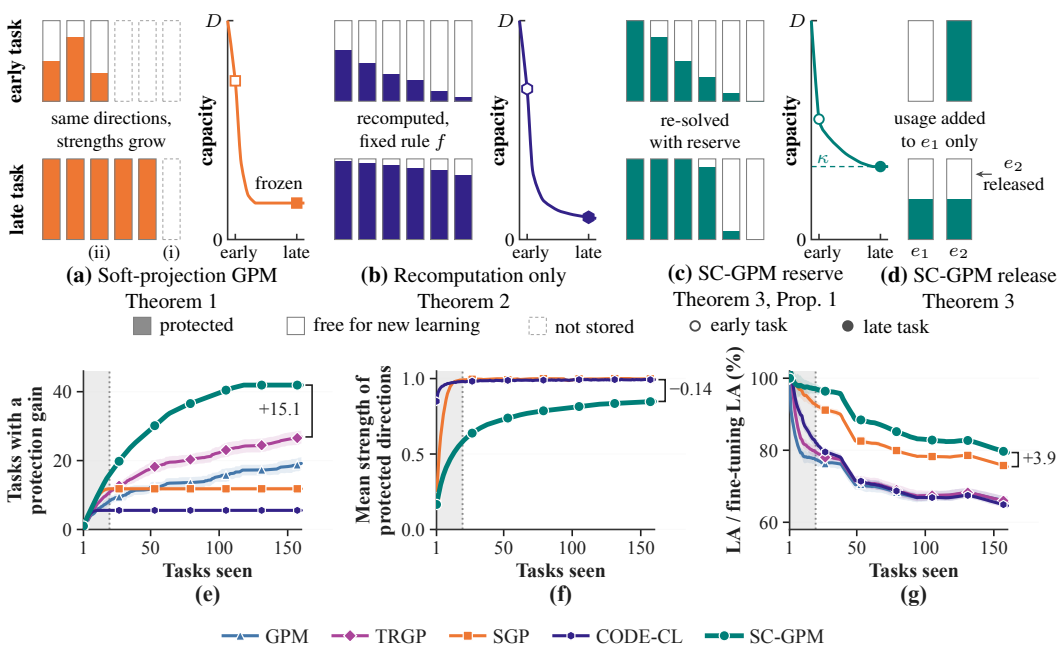


Figure 1: Stability–plasticity of gradient projection over long task sequences. Schematics illustrate (a) stopped basis growth and hardening soft protection, (b) capacity loss under a fixed spectral rule despite recomputed directions, and (c)–(d) SC-GPM's capacity reserve and release of protection. Associated theorems are indicated alongside each subfigure. Panels (e)–(g) show results for OmniBenchmark-160 with means and sample standard deviations across three seeds and the first 20 tasks shaded. (e) cumulative count of tasks with $\Delta \operatorname{tr} P \geq 1$, averaged over 20 layers; (f) mean strength over protected directions; (g) cumulative learning accuracy relative to fine-tuning on same seeds.

To answer this question, we study GPM-based continual learning over image streams of 100–160 tasks and a domain-incremental stress test extending to 200 tasks. Figure 1(e)–(g) shows how protection and learning evolve on OmniBenchmark-160. SGP and CODE-CL stop gaining protected dimensions within the first 20 tasks, their strengths approach hard constraints, and their learning accuracy falls further below that of unconstrained fine-tuning. SC-GPM's protection instead keeps growing, and its learning accuracy stays closest to fine-tuning. The restrictions sketched in Figure 1(a)–(b) therefore arise on a real benchmark. Section 3 formalizes when these mechanisms restrict subsequent updates.

To address these limitations, we propose *Spectral Consolidation Gradient Projection Memory* (SC-GPM), which combines activation usage across tasks in a fixed-size per-layer operator. After each task, SC-GPM uses this accumulated usage to jointly recompute the protected directions and their strengths. Our contributions are:

- **We develop theory to characterize the limits of projection-based methods.** For append-only memories, we establish when the memory stops admitting new directions and when its accumulated strengths reach full protection. For methods that recompute protected directions, we establish when accumulated usage reduces gradient capacity under a fixed spectral rule.
- **We empirically validate our theoretical predictions over long task sequences.** We evaluate five projection baselines on three image streams of 100–160 tasks and domain-incremental Permuted MNIST extending to 200 tasks. On the image streams, append-only memories admit no new directions at 76–90% of tasks, averaged over layers. A leading recompute-based method leaves most layers with less than 1% available gradient capacity by the end of each stream and learning accuracy falling 18.60–19.85 percentage points below unconstrained fine-tuning.
- **We introduce SC-GPM to adapt protection over long task sequences, addressing these limitations.** We formulate the joint adaptation of protected directions and their strengths as a convex optimization problem and derive an exact spectral solution that reserves a specified amount of gradient capacity. On all three image streams, SC-GPM achieves the highest final average accuracy among the evaluated sequential methods, with improvements of 2.18–3.64 percentage

points over the strongest projection baseline. Across configurations on Permuted MNIST at 200 tasks, it improves final average accuracy and average anytime accuracy by 2.48 and 1.24 points, respectively, over the strongest projection baseline for each metric.

2 Background and Related Work

2.1 Continual Learning

Prior continual-learning methods preserve earlier knowledge through parameter regularization (Kirkpatrick et al., 2017; Zenke et al., 2017; Aljundi et al., 2018), distillation (Li & Hoiem, 2016), rehearsal (Chaudhry et al., 2019b; Buzzega et al., 2020), or allocating model capacity (Rusu et al., 2016; Mallya & Lazebnik, 2018; Serrà et al., 2018). Memory requirements become particularly important over long task sequences (Lou et al., 2026). We study task- and domain-incremental learning under a fixed auxiliary-memory budget, examining activation-space protection over long task sequences.

Table 1: Related projection-based methods grouped by protection mechanism.

Protection rule and methods	T
Appended directions	
Hard: GPM (Saha et al., 2021), DFGP (Yang et al., 2023a), PGP (Qiao et al., 2024)	20
Clipped accumulation: SGP (Saha & Roy, 2023)	20
Hard base + extension: TRGP[‡] (Lin et al., 2022b), CUBER[‡] (Lin et al., 2022a), API[†] (Liang & Li, 2023), SD (Zhao et al., 2023), ROGO (Yang et al., 2023b), HLOP (Xiao et al., 2024), BECAME (Li et al., 2025), QGPM (Kim et al., 2026)	25
Replay-refreshed directions	
Learned, releasable: FS-DGPM (Deng et al., 2021)	40
Recomputed directions	
Fixed spectral scale: CODE-CL[‡] (Apolinario et al., 2025)	20
Relative or scheduled scale: Adam-NSCL (Wang et al., 2021), AdNS (Kong et al., 2022), GPCNS (Yang et al., 2024)	25
Joint allocation + reserve: SC-GPM (ours)	**200**

T: longest task sequence in original evaluation. [‡]Per-task state grows with T. [†]Expands the network.

2.2 Gradient Projection

Activation-space protection restricts changes along input directions that were important to earlier tasks (Zeng et al., 2019; Saha et al., 2021). For a layer $W \in \mathbb{R}^{d\times D}$ with input $x \in \mathbb{R}^D$, an update ΔW changes preactivations by $\Delta W x$. To limit its impact on learned tasks, the layer filters a proposed update G through a symmetric protection matrix P_t formed after task t:

$$\Delta W = GQ_t, \qquad Q_t = I - P_t, \qquad 0 \preceq P_t \preceq I. \tag{1}$$

For a stored orthonormal basis M_t, hard protection uses $P_t = M_t M_t^\top$, removing update components along its columns. Soft protection instead uses $P_t = M_t \operatorname{diag}(\gamma_t) M_t^\top$, with $\gamma_{i,t} \in [0, 1]$. Each stored direction then transmits a fraction $1 - \gamma_{i,t}$ of the proposed update along it. The choice of directions and their strengths together determines which changes remain available to later tasks.

Table 1 groups protection mechanisms and evaluation horizons. Appendix B.4 details their theoretical scope, and Appendix D compares persistent state and computational costs.

Append-only memories. GPM appends dominant residual activation directions into an auxiliary state and blocks updates along these directions to preserve earlier knowledge (Saha et al., 2021). SGP uses the same memory construction and accumulates importance across tasks to set nonuniform soft protection strengths (Saha & Roy, 2023). TRGP and CUBER use task correlations to enable transfer, while ROGO constructs a bounded relaxing space (Lin et al., 2022b;a; Yang et al., 2023b).

Recomputed protection. Building on the insight that protected directions can change as new tasks are learned, several methods recompute protection from accumulated activation statistics. OWM filters updates through a regularized inverse of accumulated input correlation (Zeng et al., 2019). CODE-CL uses conceptor filtering for soft protection (Jaeger, 2014; He & Jaeger, 2018) and supports shared-subspace transfer (Apolinario et al., 2025). Both use a fixed spectral scale. Adam-NSCL and

GPCNS instead set protection relative to the current spectrum (Wang et al., 2021; Yang et al., 2024), while AdNS adapts its null space and schedules the constraint strength (Kong et al., 2022). These relative and scheduled rules differ from the fixed spectral rules analyzed in Section 3. SC-GPM uses the accumulated spectrum to revise directions and strengths jointly while reserving gradient capacity.

3 Limitations of Accumulated Gradient Protection

We examine how protected directions and their strengths evolve as tasks accumulate. Neural networks can lose plasticity during prolonged training (Dohare et al., 2024). Here, we analyze how accumulated gradient protection can restrict the updates available to later tasks.

Gradient capacity. For hard protection, gradient capacity counts the unprotected input directions. Soft protection also permits partial updates along protected directions. We account for both through the *gradient capacity*, $C_t = \operatorname{tr} Q_t$, the sum of the factors by which Q_t transmits updates along its eigenvectors. We omit the layer index in the analysis and use $\|\cdot\|_2$ and $\|\cdot\|_F$ for spectral and Frobenius norms. We provide details on our notation and assumptions in Appendix A.

3.1 Finite Basis Allocation and Strength Saturation

Methods in this family protect earlier tasks along two axes: allocating new basis directions and scaling the protection strength along them (Figure 1(a)). Consider an append-only memory that retains its existing orthonormal columns and only appends new directions, so the protected subspace $S_t = \operatorname{span}(M_t)$ expands monotonically ($S_{t-1} \subseteq S_t$). This construction underlies GPM, SGP, and several extensions (Appendix B.4). Because a layer with input dimension D can accommodate only finitely many mutually orthogonal additions, basis growth must eventually halt, leaving later tasks to share a fixed basis (Figure 1(a)(i)).

Each retained direction carries a nondecreasing protection strength $\gamma_{i,t} \in [0, 1]$. Hard protection fixes every strength at one. SGP instead accumulates importance through the clipped recursion $\gamma_{i,t} = \min\{1, \gamma_{i,t-1} + \delta_{i,t}\}$, with $\delta_{i,t} \geq 0$ (Saha et al., 2021; Saha & Roy, 2023). Once basis growth stops and every retained strength reaches one, protection becomes a fixed hard projector that blocks updates along these directions (Figure 1(a)(ii)). Theorem 1 formalizes both restrictions.

Theorem 1 (Finite allocation and strength saturation)**.** *Let an append-only orthonormal memory have rank at most $R \leq D$, with $k_t = \dim S_t - \dim S_{t-1}$. **(i) Finite allocation:** The number of tasks with basis additions is bounded by $\#\{t : k_t > 0\} \leq R - \dim S_0$. Consequently, additions cease after a finite task, leaving the stored basis fixed thereafter.*

***(ii) Monotone strengths:** If retained strengths never decrease, the protection matrix satisfies $P_{t+1} \succeq P_t$. A direction first stored at task τ_i follows the clipped recursion with nonnegative increments $\delta_{i,s}$:*

$$\gamma_{i,t} = \min\left\{1,\ \gamma_{i,\tau_i} + \sum_{s=\tau_i+1}^{t} \delta_{i,s}\right\}. \tag{2}$$

If, for every retained column, these increments sum to at least $1 - \gamma_{i,\tau_i}$ at a finite index, then beyond a finite task $P_t = \Pi$ is a fixed hard projector. Every subsequent update satisfies $\Delta W = \Delta W(I - \Pi)$, restricting updates to the remaining unprotected input directions.

In geometric terms, once protection hardens, updates can act only on the input directions orthogonal to the protected subspace. For a fixed input second-moment matrix $\Sigma = \mathbb{E}[xx^\top]$, the resulting change in layer preactivation satisfies $\mathbb{E}\|\Delta W x\|^2 \leq \|\Delta W\|_2^2 \operatorname{tr}[(I - \Pi)\Sigma]$. In other words, if the protected directions capture an ε fraction of the input energy, only the remaining $(1 - \varepsilon)$ fraction can contribute to the direct layer-response change.

We observe these restrictions in practice. **Basis additions become rare:** Across the three image streams, GPM, TRGP, and SGP add no new basis directions at 76–90% of tasks, averaged over layers (Appendix Figure 4). **Protection becomes nearly hard:** On OmniBenchmark-160, SGP's mean protection strength exceeds .99 by task 20 (Figure 1(f)), showing that its initially soft protection quickly approaches the hard-protection regime on average. Appendix F.1 gives the measurements and Appendix B the proofs.

3.2 Gradient Capacity under Fixed Spectral Protection

Rather than freezing a basis over time, methods such as OWM and CODE-CL allow their protected directions to change as activation statistics accumulate (Zeng et al., 2019; Apolinario et al., 2025). Let $R_0 \succeq 0$ and $R_t = R_{t-1} + B_t$, where $B_t \succeq 0$ reflects newly observed feature activations. Recomputing eigenvectors lets the protected basis follow changes in data representations.

However, recomputing directions alone does not guarantee the ability to learn future tasks. In these methods, a fixed spectral transfer function $f : [0, \infty) \to [0, 1]$ maps the eigenvalues of R_t to protection strengths, with $f(0) = 0$ and $f(\lambda) \to 1$ as $\lambda \to \infty$. The resulting protection matrix is $P_t = f(R_t)$, with $Q_t = I - f(R_t)$ filtering subsequent updates.

For example, the rational form $f_a(\lambda) = \lambda/(\lambda + a)$ with fixed scale $a > 0$ yields the regularized pass-through operator $Q_t = a(R_t + aI)^{-1}$. As activation energy sums across tasks without decay, the eigenvalues of R_t cannot decrease. Because f is nondecreasing, larger eigenvalues assign stronger protection, preventing any increase in the gradient capacity $\operatorname{tr} Q_t$. As a result, the protected basis can continually rotate while total gradient capacity falls toward zero (Figure 1(b)). Theorem 2 establishes the conditions under which this capacity vanishes.

Theorem 2 (Saturation of cumulative spectral protection)**.** *Under this fixed spectral rule,* $\operatorname{tr} Q_{t+1} \leq \operatorname{tr} Q_t$*. If* f *also preserves PSD order (is operator monotone), as* f_a *does, then* $P_{t+1} \succeq P_t$*. The directions with zero accumulated usage eventually form a fixed subspace of dimension* n*. If every positive eigenvalue grows without bound, then* $\operatorname{tr} Q_t \to n$*. In particular, an eventually full-rank state with diverging least eigenvalue has* $\operatorname{tr} Q_t \to 0$*.*

In other words, the protected directions can change even as the capacity available for later updates decreases (Figure 1(b)). On OmniBenchmark-160, CODE-CL's mean protection strength reaches .98 by task 20 (Figure 1(f)), displaying the same rapid saturation observed in static-basis methods.

Together, Theorems 1 and 2 identify how accumulated protection can restrict later updates. These results motivate SC-GPM, which jointly revises protected directions and their strengths while reserving gradient capacity (Section 4).

4 Spectral Consolidation Gradient Projection Memory

4.1 Accumulated Task Usage

After learning task t, we compute the uncentered empirical activation correlation matrix $\widehat{\Sigma}_t = \frac{1}{N} \sum x x^\top$ and its eigendecomposition $V_t \operatorname{diag}(s_{t,i}) V_t^\top$, where the eigenvalues $s_{t,1} \geq \cdots \geq s_{t,D} \geq 0$ quantify the signal energy along each orthogonal direction $v_{t,i}$. We use uncentered PCA because the layer acts on the raw inputs x. For a target energy threshold $0 < \varepsilon \leq 1$, we retain the leading K_t eigenvectors capturing an ε fraction of the total trace energy.

For nonzero moments, we normalize by the leading eigenvalue, $x_{t,i} = s_{t,i}/s_{t,1}$, and define

$$m_\alpha(x) = \frac{(\alpha + 1)x}{1 + \alpha x}, \quad U_t = \sum_{i=1}^{K_t} m_\alpha(x_{t,i}) v_{t,i} v_{t,i}^\top, \quad \Omega_t = \Omega_{t-1} + U_t, \tag{3}$$

where $\alpha \geq 0$, $\Omega_0 = 0$, and $U_t = 0$ for a zero moment. We call $m_\alpha(x_{t,i})$ the *task usage spectrum*. Since $m_\alpha(1) = 1$, each nonzero task contributes peak usage one, independently of activation scale. At $\alpha = 0$, $m_0(x) = x$ preserves relative energy. Increasing α gives greater weight to weaker activation directions within the retained set, making their weights more equal. For a unit direction v, $v^\top \Omega_t v$ sums the normalized and weighted activation energy along that direction across tasks. After each task, SC-GPM uses Ω_t to adapt protected directions and their strengths.

4.2 Capacity-Constrained Protection Allocation

We allocate protection by balancing two competing geometric objectives: preserving the proposed parameter update against avoiding changes along high-energy (dominant) activation directions. Let G denote the proposed gradient update, and let GQ be the projected update applied to the layer.

To limit changes to layer outputs on past tasks, we use the penalty $\mathrm{tr}(GQ\Omega QG^\top)$, motivated by a layerwise Taylor expansion of past-task losses (Appendix C.1). We measure deviation from the unconstrained update by $\|G - GQ\|_F^2$. For update distributions satisfying $\mathbb{E}[G^\top G] \preceq \sigma^2 I$, the expected penalties are bounded by $\sigma^2 \,\mathrm{tr}(\Omega Q^2)$ and $\sigma^2\|I - Q\|_F^2$, respectively. Since $0 \preceq Q \preceq I$ implies $Q^2 \preceq Q$, we further bound $\mathrm{tr}(\Omega Q^2)$ by $\mathrm{tr}(\Omega Q)$. Combining these bounds with a gradient capacity reserve $0 < \kappa \leq D$ gives the following convex surrogate:

$$\min_{0\preceq Q\preceq I \ \mathrm{tr}\, Q\geq\kappa} F_\Omega(Q) = \tfrac{1}{2}\|I - Q\|_F^2 + \mathrm{tr}(\Omega Q). \tag{4}$$

The first term penalizes distortion of the proposed update, while the second suppresses updates along directions strongly represented in earlier tasks' activations. The constraint $\mathrm{tr}\, Q \geq \kappa$ reserves at least κ effective units of gradient capacity for later updates (Figure 1(c)). Appendix C provides the complete derivation and optimality conditions.

Theorem 3 (Exact spectral allocation)**.** *Let $\Omega = V\,\mathrm{diag}(\lambda_i)V^\top \succeq 0$. The unique minimizer $Q^\star$ of (4) shares the eigenbasis V of Ω: $Q^\star = V\,\mathrm{diag}(q_1^\star, \ldots, q_D^\star)V^\top$. Each $q_i^\star$ determines the fraction of the proposed update retained along direction v_i. A shared threshold $c \geq 0$ determines these fractions:*

$$q_i^\star = \mathrm{clip}_{[0,1]}(1 + c - \lambda_i). \tag{5}$$

Clipping keeps each transmission factor between zero and one. Set $c = 0$ if the resulting allocation already provides sufficient gradient capacity, $\sum_i \mathrm{clip}_{[0,1]}(1 - \lambda_i) \geq \kappa$. Otherwise, choose $c > 0$ to relax protection until total gradient capacity meets the reserve exactly, $\sum_i q_i^\star = \kappa$.

SC-GPM selects protection $P^\star = I - Q^\star$ closest to the accumulated activation statistics Ω in Frobenius norm, subject to the capacity reserve (Figure 1(c)). As later tasks contribute new activation statistics, both the directions and their strengths can change, allowing previously blocked directions to become available for updates (Figure 1(d)).

Proposition 1 (Monotone, bounded total protection)**.** *For PSD accumulation and fixed κ, $Q_0^\star = I$ and $\mathrm{tr}\, Q_t^\star = \max\left\{\kappa, \sum_i \mathrm{clip}_{[0,1]}(1 - \lambda_i(t))\right\}$. Thus total protection is nondecreasing and bounded by $D - \kappa$, while gradient capacity never falls below κ.*

A fixed basis allows protection strengths to change, but restricts the protection directions.

Proposition 2 (Cost of an inherited basis)**.** *Let $P^\star = I - Q^\star$, and let $M \in \mathbb{R}^{D\times D}$ be an orthonormal basis. Every feasible protection P that is diagonal in this basis satisfies $F_\Omega(I - P) - F_\Omega(Q^\star) \geq \frac{1}{2}\|\,\mathrm{off}(M^\top P^\star M)\|_F^2$, where* off *sets the diagonal entries to zero.*

Here M can extend a stored partial basis and adjusting protection strengths changes only the diagonal entries in this basis. The off-diagonal entries of $M^\top P^\star M$ quantify the mismatch between the inherited basis and the optimal protection. Proposition 2 gives a lower bound on the resulting cost in (4), even after optimizing the protection strengths. Recomputing the basis from Ω removes this restriction. Now, the strengths along the new directions can be chosen to obtain $P^\star$. If adjusting strengths in the inherited basis can already produce $P^\star$, the lower bound is zero.

Computing and Memory Overheads. During task t, we hold P_{t-1} fixed and filter layer updates through $I - P_{t-1}$. After each task, we compute U_t, update Ω_t, and obtain P_t using Theorem 3. We retain one $D \times D$ matrix of accumulated activation statistics per protected layer and reconstruct protection from this state. This requires $O(\sum_l D_l^2)$ storage, independent of task count and matching GPM's worst-case storage order. The eigendecomposition operates in each layer's input activation space and, together with scalar allocation, requires $O(\sum_l D_l^3)$ time at each task boundary. In practice, this adds minimal overhead: on ImageNet-1K-100, SC-GPM takes approximately 2% longer than GPM/SGP, and 4% longer than fine-tuning (Table 9). Appendix D provides the details.

5 Experimental Evaluation

5.1 Experimental Setup

Following the task-incremental protocol of GPM and SGP (Saha et al., 2021; Saha & Roy, 2023), we evaluate DomainNet-101 (Peng et al., 2019), OmniBenchmark-160 (Zhang et al., 2022), and

Table 2: Accuracy and transfer on the three long image streams.

Method	DomainNet-101 ACC	DomainNet-101 LA	DomainNet-101 BWT	DomainNet-101 FWT	OmniBenchmark-160 ACC	OmniBenchmark-160 LA	OmniBenchmark-160 BWT	OmniBenchmark-160 FWT	ImageNet-1K-100 ACC	ImageNet-1K-100 LA	ImageNet-1K-100 BWT	ImageNet-1K-100 FWT
Finetune	10.25 ±0.35	**81.73** ±0.10	−72.19 ±0.27	**6.63** ±0.47	8.06 ±0.21	**56.03** ±0.68	−48.27 ±0.53	**9.31** ±0.34	14.91 ±0.38	**75.16** ±0.13	−60.86 ±0.34	**4.84** ±0.30
ER	**30.60** ±0.64	78.30 ±0.35	**−48.18** ±0.78	3.20 ±0.65	**14.10** ±0.12	51.24 ±0.26	**−37.38** ±0.31	4.53 ±0.12	**31.93** ±0.68	73.43 ±0.28	**−41.92** ±0.75	3.11 ±0.23
GPM	58.76 ±1.72	63.49 ±0.57	−4.78 ±1.17	−11.61 ±0.60	33.40 ±0.71	36.87 ±0.28	−3.49 ±0.52	−9.85 ±0.62	46.87 ±0.90	53.96 ±0.52	−7.16 ±1.32	−16.35 ±0.70
TRGP	58.37 ±1.97	63.22 ±0.60	−4.89 ±1.40	−11.89 ±0.67	33.39 ±0.29	36.80 ±0.36	−3.44 ±0.10	−9.91 ±0.68	47.23 ±0.88	53.43 ±0.62	−6.26 ±0.27	−16.89 ±0.80
Adam-NSCL	56.78 ±1.30	71.38 ±0.83	−14.74 ±1.00	−3.73 ±0.40	27.87 ±0.59	43.12 ±1.35	−15.35 ±1.91	−3.60 ±1.31	45.73 ±0.28	62.99 ±0.58	−17.44 ±0.30	−7.32 ±0.39
SGP	66.49 ±0.55	71.68 ±0.49	−5.25 ±0.08	−3.42 ±0.06	39.25 ±0.69	42.27 ±0.38	−3.04 ±0.41	−4.44 ±0.14	53.90 ±0.63	61.60 ±0.56	−7.77 ±0.17	−8.72 ±0.68
CODE-CL	63.29 ±0.60	63.13 ±0.52	**0.17** ±0.11	−11.98 ±0.55	36.46 ±0.19	36.18 ±0.15	**0.27** ±0.09	−10.53 ±0.33	55.01 ±0.44	55.48 ±0.41	**−0.47** ±0.18	−14.83 ±0.60
SC-GPM	**70.13** ±0.30	**75.81** ±0.41	−5.74 ±0.13	**0.71** ±0.25	**41.53** ±0.70	**44.46** ±0.65	−2.95 ±0.05	**−2.25** ±0.32	**57.19** ±0.68	**66.32** ±0.24	−9.23 ±0.48	**−3.99** ±0.39
Independent	Acc.: 75.10 ± 0.55				Acc.: 46.71 ± 0.34				Acc.: 70.32 ± 0.22			
Joint (offline)	Acc.: 77.06 ± 0.74				Acc.: 51.85 ± 0.01				Acc.: 74.38 ± 0.12			

Acc. and LA are percentages; BWT and FWT are percentage points. Entries show three-seed means with sample SD. Bold marks the best mean within the non-projection and projection groups.

ImageNet-1K-100 (Deng et al., 2009) (suffixes indicate task counts). All use a reduced ResNet-18. Each task sequence has two phases that differ in domain, class granularity, or semantic group. Task identity selects the output head in this task-incremental setting (van de Ven et al., 2022). A separate domain-incremental Permuted MNIST stress test (Goodfellow et al., 2013) follows the single-head protocol of GPM: a two-hidden-layer MLP with one shared ten-way head, and no task identity at inference. Fresh pixel permutations vary the horizon from 50 to 200 tasks while holding the recognition problem fixed. As in GPM, all methods train with SGD, except Adam-NSCL, whose Adam optimizer is part of the method. Appendix E gives a comprehensive accounting of these.

We compare SC-GPM with GPM, SGP, TRGP, Adam-NSCL and CODE-CL (Table 1). Fine-tuning measures unconstrained acquisition. Independent and Joint provide reference accuracies: Independent trains a separate model on each task, while Joint trains one model offline on all tasks. For the sequential methods, we bound persistent auxiliary memory using a base budget equivalent to one $D_l \times D_l$ matrix per protected layer (see Appendix Table 7). This budget covers projection state or stored replay examples; task-specific output heads are accounted for separately. GPM, SGP, Adam-NSCL and SC-GPM stay within this memory constraint, and experience replay (ER) (Chaudhry et al., 2019b) uses it for stored examples. TRGP and CODE-CL retain additional task-specific state and receive a nominal allowance of twice the base budget. Further allocations stop once this allowance is reached. CODE-CL allocates state for whole tasks, so its final allocation brings its image-stream total to 2.13 times the base budget (Appendix D.2). Comparisons report three-seed means and sample SD. On the image streams, baseline hyperparameters follow published settings, with CODE-CL evaluated at two published apertures (8 and 16) and reported at the better setting. SC-GPM uses the same retained-energy threshold as GPM, SGP and TRGP ($\varepsilon = .985$) and a fixed reserve $\kappa_l = 1$; only α is selected, from four values on DomainNet validation data using a seed disjoint from evaluation. This configuration is then used unchanged on all three image streams. Single-seed checks on DomainNet favor the published GPM and SGP settings over the tested alternatives (Appendix E). On PMNIST, we conduct broader method-specific searches (Section 5.4).

Evaluation metrics. We report final average accuracy (Acc./ACC), learning accuracy (LA), backward transfer (BWT), and forward transfer (FWT). We evaluate average anytime accuracy (AAA) at task boundaries to measure performance throughout learning. Appendix E provides definitions.

5.2 Results on Long Image Streams

SC-GPM achieves the highest final accuracy among the evaluated sequential methods on all three image streams (Table 2). Relative to the strongest projection baseline, its gains are 3.64 points over

SGP on DomainNet, 2.28 over SGP on OmniBenchmark, and 2.18 over CODE-CL on ImageNet-1K. Its learning accuracy is higher by 4.13, 2.19 and 10.84 points, respectively. On OmniBenchmark, SC-GPM improves both acquisition and BWT over every append-only baseline. CODE-CL, the most recent baseline, nearly eliminates forgetting, but its learning accuracy is 8.28–12.68 points below SC-GPM's. Consistent with Theorem 2, CODE-CL's fixed-aperture protection leaving under 1% free capacity in most layers by the end of each stream (Appendix F.1). Under the same auxiliary-memory budget, SC-GPM exceeds ER by 25.26–39.53 percentage points in final accuracy and 14.85–23.01 points in AAA on the three image streams (Table 2; Appendix Table 12). ER achieves higher LA but forgets substantially more.

Although the best projection baseline exceeds SC-GPM on each image stream at $T = 20$, SC-GPM has the highest mean Acc among the evaluated projection methods from tasks 40, 27 and 53 onwards on DomainNet, OmniBenchmark and ImageNet-1K, respectively. Appendix Figure 8 offers a detailed breakdown of the behavior over task horizons. Appendix Figure 9 and the trajectories in Appendix F.2 separate acquisition, retention and the early cost of the eventual advantage.

5.3 Ablations

Table 3 compares GPM, memory consolidation alone, allocation within GPM's basis, and joint revision. Joint spectral consolidation improves final accuracy by 8.13–11.37 points over GPM, and the gain comes through learning accuracy, which rises by 7.59–12.36 points. Within the consolidated representation, optimizing the protection profile contributes 7.54–10.71 points, whereas consolidating the memory under hard protection improves final accuracy by only 0.40–0.66 points. The allocation within GPM's appended basis improves final accuracy over GPM by 2.72–3.59 points and gives the highest LA in the table, but its BWT is 6.54–8.52 points below SC-GPM's and its final accuracy trails SC-GPM's by 5.33–7.78 points. Revising strengths alone thus releases capacity that the inherited basis cannot protect, as Proposition 2 predicts, so directions and strengths must be revised together.

Table 3: Ablation of memory consolidation and protection allocation on the three image streams

	DomainNet-101			OmniBenchmark-160			ImageNet-1K-100		
Construction	ACC	LA	BWT	ACC	LA	BWT	ACC	LA	BWT
GPM	58.76 ±1.72	63.49 ±0.57	**−4.78** ±1.17	33.40 ±0.71	36.87 ±0.28	−3.49 ±0.52	46.87 ±0.90	53.96 ±0.52	**−7.16** ±1.32
Consolidated + hard	59.42 ±1.59	65.17 ±0.62	−5.81 ±0.99	33.99 ±0.33	38.09 ±0.12	−4.12 ±0.31	47.27 ±1.61	56.11 ±0.60	−8.92 ±2.16
Appended + allocation	62.35 ±1.65	**76.48** ±0.18	−14.26 ±1.55	36.20 ±0.62	**45.63** ±0.56	−9.49 ±0.39	49.59 ±1.04	**66.98** ±0.39	−17.57 ±1.43
SC-GPM	**70.13** ±0.30	75.81 ±0.41	−5.74 ±0.13	**41.53** ±0.70	44.46 ±0.65	**−2.95** ±0.05	**57.19** ±0.68	66.32 ±0.24	−9.23 ±0.48

Three-seed means and sample SD. Bold marks the best mean in each column.

5.4 Domain-Incremental PMNIST

We evaluate PMNIST over 50–200 permutations with a shared output head and broader hyperparameter searches, selecting each projection method's configuration separately at each horizon. GPM, SGP, TRGP and SC-GPM share the retained-energy grid, with method-specific searches described in Appendix E. SC-GPM matches CODE-CL at 50 tasks and leads projection methods in final accuracy from 100 tasks onward, reaching $18.50 \pm 1.22\%$ at 200 tasks versus $16.02 \pm 0.22\%$ for SGP (Figure 2(a)). At 200 tasks, SC-GPM's best configuration at every shared threshold outperforms the corresponding configurations of the other threshold-based methods and the best searched configurations of CODE-CL and Adam-NSCL (Figure 2(c)). SC-GPM, SGP, GPM and TRGP achieve similar learning accuracy near 97%, so SC-GPM's higher final accuracy reflects less forgetting. CODE-CL's learning accuracy declines as its protection saturates. We also vary the number of training examples used to estimate projection statistics from 64 to 4,000 per task. Across this range, SC-GPM's final accuracy at 200 tasks exceeds the best of GPM, SGP, TRGP and CODE-CL by 1.08–2.81 percentage points (Appendix F.3). With each method at its configuration selected for AAA, SC-GPM also has the highest AAA among projection methods at every PMNIST horizon, while on the image streams CODE-CL's early retention gives it the highest value on DomainNet and ImageNet-1K (Table 12).

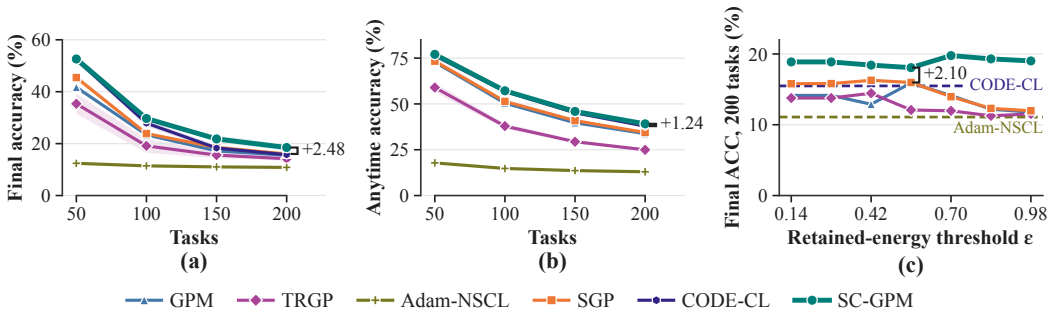


Figure 2: (a) Final accuracy and (b) average anytime accuracy, with configurations selected separately for each method, metric and horizon. (c) Final accuracy at 200 tasks versus ε.

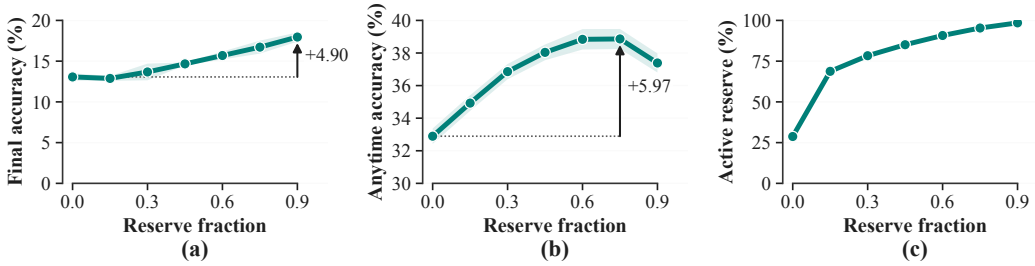


Figure 3: A larger capacity reserve engages more often and raises final accuracy. (a) Final accuracy, (b) average anytime accuracy and (c) the share of layer–task boundaries where the reserve changes protection, as functions of the reserve fraction on PMNIST-200 with $\varepsilon = .98$ and $\alpha = 20$. A fraction of zero denotes the minimum reserve $\kappa_l = 1$. Three-seed means and sample SD.

Replay remains stronger on this shared-head problem, and Appendix F.3 reports its complete results together with fine-tuning and fixed-configuration trajectories.

Impact of the Capacity Reserve. Even at the minimum setting, the reserve is active during 10.2 to 13.9% of layer and task boundaries across the nine runs on the image streams. This occurs only in the three shortcut convolutions with $D = 10, 20, 40$, where capacity stays at the floor of $\operatorname{tr} Q = 1$. In contrast, SGP drops to zero gradient capacity ($\operatorname{tr} Q = 0$) in the $D = 10$ shortcut in eight of the nine runs (Appendix F.1). On PMNIST-200 with fixed $(\varepsilon, \alpha) = (.98, 20)$, we vary the reserve fraction to evaluate its impact on performance (Figure 3). Raising the reserve to .90 increases final accuracy from 13.06% to 17.96%. However, AAA reaches its highest value of 38.86% with a .75 reserve, compared to 32.89% at the minimum setting. The optimal reserve level therefore differs between final accuracy and anytime accuracy. Appendix F.4 shows how α controls the balance between early accuracy and capacity for later tasks.

6 Conclusion

We establish that accumulated protection restricts later learning in GPM-style methods, and we validate these predictions empirically over long task sequences. To address these limitations, we introduce SC-GPM and derive its protection rule from a convex optimization problem that uses accumulated task usage to guide protection while reserving capacity for later updates. Ablations show that combining memory consolidation and protection allocation improves final accuracy over either alone. Using bounded auxiliary memory, SC-GPM achieves the highest final accuracy among the evaluated sequential methods on all three image streams of 100–160 tasks. It exceeds the strongest projection baseline by 2.18–3.64 percentage points. At 200 tasks on domain-incremental PMNIST, SC-GPM improves upon the strongest projection baselines by 2.48 points in final accuracy and 1.24 in AAA. **Limitations:** SC-GPM fixes each layer's minimum gradient capacity before training. However, while this mechanism guarantees capacity, it is unable to guarantee alignment with future gradients or optimal generalization or transfer. Adapting the reserve to task demands and relating gradient capacity to these outcomes remain directions for future work.

Reproducibility Statement

Appendix A states the assumptions and notation, Appendices B–C provide complete proofs, and Appendix D specifies the algorithm. Appendix E describes data construction, training, reference runs and parameter selection. Code to reproduce all experiments is available at https://anonymous.4open.science/r/SC-GPM-2027. The empirical appendices report three-seed results, fixed-configuration trajectories, sensitivity analyses and the memory accounting in Appendix D.2.

Ethics Statement

This work studies continual learning on existing image-classification benchmarks and does not involve a new human-subject study. Aggregate benchmark accuracy does not establish fairness, privacy or suitability for deployment, and the learned models can inherit biases in the source data. Our evaluation concerns sustained learning under the stated benchmark protocols.

AI Use Statement

In this work, we used generative AI tools to implement methods. The theoretical results and the experimental design are the authors' own. Additionally, we used generative AI tools to create and edit software code, create scientific figures, identify relevant literature, draft parts of the paper and edit it for readability. We have reviewed all AI-assisted work: the authors verified and tested the AI-assisted code for correctness, checked every figure against the logged results, checked every cited work, and revised the drafted text. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

References

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Computer Vision – ECCV 2018*, pp. 144–161. Springer International Publishing, 2018. doi: 10.1007/978-3-030-01219-9_9.

Marco P. E. Apolinario, Sakshi Choudhary, and Kaushik Roy. CODE-CL: Conceptor-based gradient projection for deep continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 775–784, 2025.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. doi: 10.1017/CBO9780511804441.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: A strong, simple baseline. In *Advances in Neural Information Processing Systems*, volume 33, pp. 15920–15930, 2020.

Massimo Caccia, Pau Rodriguez, Oleksiy Ostapenko, Fabrice Normandin, Min Lin, Lucas Page-Caccia, Issam Hadj Laradji, Irina Rish, Alexandre Lacoste, David Vázquez, and Laurent Charlin. Online fast adaptation and knowledge accumulation (OSAKA): A new approach to continual learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 16532–16545, 2020.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *International Conference on Learning Representations*, 2019a.

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

Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. RandAugment: Practical automated data augmentation with a reduced search space. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18613–18624, 2020.

Danruo Deng, Guangyong Chen, Jianye Hao, Qiong Wang, and Pheng-Ann Heng. Flattening sharpness for dynamic gradient projection memory benefits continual learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18710–18721, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE, 2009. doi: 10.1109/CVPR.2009.5206848.

Shibhansh Dohare, J. Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A. Rupam Mahmood, and Richard S. Sutton. Loss of plasticity in deep continual learning. *Nature*, 632(8026):768–774, 2024. doi: 10.1038/s41586-024-07711-7.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 3762–3773. PMLR, 2020.

Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Ian J. Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

Xu He and Herbert Jaeger. Overcoming catastrophic interference using conceptor-aided backpropagation. In *International Conference on Learning Representations*, 2018.

Herbert Jaeger. Controlling recurrent neural networks by conceptors. *arXiv preprint arXiv:1403.3369*, 2014.

Dongjun Kim, Seohyeon Cha, Huancheng Chen, Chianing Wang, and Haris Vikalo. Quantized gradient projection for memory-efficient continual learning. In *International Conference on Learning Representations*, pp. 21920–21945, 2026.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

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, 2017. doi: 10.1073/pnas.1611835114.

Yajing Kong, Liu Liu, Zhen Wang, and Dacheng Tao. Balancing stability and plasticity through advanced null space in continual learning. In *Computer Vision – ECCV 2022*, pp. 219–236. Springer Nature Switzerland, 2022. doi: 10.1007/978-3-031-19809-0_13.

Mei Li, Yuxiang Lu, Qinyan Dai, Suizhi Huang, Yue Ding, and Hongtao Lu. BECAME: Bayesian continual learning with adaptive model merging. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 35481–35501. PMLR, 2025.

Zhizhong Li and Derek Hoiem. Learning without forgetting. In *Computer Vision – ECCV 2016*, pp. 614–629. Springer International Publishing, 2016. doi: 10.1007/978-3-319-46493-0_37.

Yan-Shuo Liang and Wu-Jun Li. Adaptive plasticity improvement for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7816–7825, 2023.

Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. Beyond not-forgetting: Continual learning with backward knowledge transfer. In *Advances in Neural Information Processing Systems*, volume 35, pp. 16165–16177, 2022a. doi: 10.52202/068431-1176.

Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. TRGP: Trust region gradient projection for continual learning. In *International Conference on Learning Representations*, 2022b.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

Meng Lou, Yunxiang Fu, and Yizhou Yu. Scaling continual learning to 300+ tasks with bi-level routing mixture-of-experts. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Arun Mallya and Svetlana Lazebnik. PackNet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

James Martens and Roger Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2408–2417. PMLR, 2015.

Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, pp. 109–165. Elsevier, 1989. doi: 10.1016/S0079-7421(08)60536-8.

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/CVF International Conference on Computer Vision (ICCV)*, 2019.

Jingyang Qiao, Zhizhong Zhang, Xin Tan, Chengwei Chen, Yanyun Qu, Yong Peng, and Yuan Xie. Prompt gradient projection for continual learning. In *International Conference on Learning Representations*, pp. 16294–16315, 2024.

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 and Kaushik Roy. Continual learning with scaled gradient projection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8):9677–9685, 2023. doi: 10.1609/aaai.v37i8.26157.

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

Joan Serrà, Dídac Surís, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4548–4557. PMLR, 2018.

Gido M. van de Ven, Tinne Tuytelaars, and Andreas S. Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 4(12):1185–1197, 2022. doi: 10.1038/s42256-022-00568-3.

Tom Veniat, Ludovic Denoyer, and Marc'Aurelio Ranzato. Efficient continual learning with modular networks and task-driven priors. In *International Conference on Learning Representations*, 2021.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5362–5383, 2024. doi: 10.1109/TPAMI.2024.3367329.

Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of feature covariance for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 184–193, 2021.

Mingqing Xiao, Qingyan Meng, Zongpeng Zhang, Di He, and Zhouchen Lin. Hebbian learning based orthogonal projection for continual learning of spiking neural networks. In *International Conference on Learning Representations*, pp. 16271–16293, 2024.

Chengyi Yang, Mingda Dong, Xiaoyue Zhang, Jiayin Qi, and Aimin Zhou. Introducing common null space of gradients for gradient projection methods in continual learning. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 5489–5497. ACM, 2024. doi: 10.1145/3664647.3680605.

Enneng Yang, Li Shen, Zhenyi Wang, Shiwei Liu, Guibing Guo, and Xingwei Wang. Data augmented flatness-aware gradient projection for continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5630–5639, 2023a.

Zeyuan Yang, Zonghan Yang, Yichen Liu, Peng Li, and Yang Liu. Restricted orthogonal gradient projection for continual learning. *AI Open*, 4:98–110, 2023b. doi: 10.1016/j.aiopen.2023.08.010.

Jinsoo Yoo, Yunpeng Liu, Frank Wood, and Geoff Pleiss. Layerwise proximal replay: A proximal point method for online continual learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57199–57216. PMLR, 2024.

Guanxiong Zeng, Yang Chen, Bo Cui, and Shan Yu. Continual learning of context-dependent processing in neural networks. *Nature Machine Intelligence*, 1(8):364–372, 2019. doi: 10.1038/s42256-019-0080-x.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3987–3995. PMLR, 2017.

Yuanhan Zhang, Zhenfei Yin, Jing Shao, and Ziwei Liu. Benchmarking omni-vision representation through the lens of visual realms. In *Computer Vision – ECCV 2022*, pp. 594–611. Springer Nature Switzerland, 2022. doi: 10.1007/978-3-031-20071-7_35.

Zhen Zhao, Zhizhong Zhang, Xin Tan, Jun Liu, Yanyun Qu, Yuan Xie, and Lizhuang Ma. Rethinking gradient projection continual learning: Stability / plasticity feature space decoupling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3718–3727, 2023.

A Notation and Assumptions

We index tasks by $t = 1, \ldots, T$ and analyze one protected layer at a time. The implementation solves the same allocation independently at each protected layer. Matrices compared with $\preceq$ are symmetric: $A \preceq B$ means that $B - A$ is positive semidefinite (PSD). A symmetric PSD contraction has eigenvalues in $[0, 1]$. For a matrix A, $\mathrm{off}(A)$ keeps its off-diagonal entries and sets its diagonal to zero. Table 4 collects the notation, and Table 5 states the assumptions behind the loss-based motivation.

Table 4: Notation used in the analysis.

Symbol	Meaning
$W \in \mathbb{R}^{d\times D}$, $x \in \mathbb{R}^D$	Layer weights and input, with D fixed in the finite-allocation theorem.
G, $\Delta W = GQ$	Proposed and applied weight displacements.
P, $Q = I - P$	Protection and gradient-transmission matrices, with $0 \preceq P, Q \preceq I$.
$C = \mathrm{tr}\, Q$, κ	Total gradient capacity and its prescribed minimum.
M_t, S_t, $\gamma_{i,t}$	Stored orthonormal basis, its span, and per-direction protection strengths.
$\widehat{\Sigma}_t$, $s_{t,i}$	Task input second moment and its ordered eigenvalues.
ε, K_t, α	Retained energy fraction, retained rank, and task-spectrum shape parameter.
U_t, $\Omega_t = \sum_{\tau\le t} U_\tau$	Normalized task usage and cumulative usage operator.
λ_i, c	Eigenvalues of Ω and the capacity-constraint multiplier.
$\langle A, B\rangle_F$	Frobenius inner product $\mathrm{tr}(A^\top B)$.

Table 5: Core modeling assumptions for the loss-based motivation. These local assumptions justify the surrogate, whereas the exact allocation theorem requires only its stated PSD state and feasible capacity constraint.

Assumption	Formal statement	Purpose and scope
Local layerwise analysis	Vary W with inputs x_τ, other layers and normalization scales fixed.	Isolates the response controlled by one projection operator. Simultaneous multi-layer changes introduce mixed curvature terms.
Bounded local curvature	$\nabla_o^2 \ell_\tau(o + u\Delta W x_\tau, z) \preceq LI$ for $u \in [0, 1]$, with finite input second moments.	Bounds the Taylor remainder by $(L/2)\|\Delta W x_\tau\|^2$ on a smooth update segment. No stationarity of past tasks is assumed.
Bounded update second moment	Conditional on fixed Q, Ω, $0 \preceq K = \mathbb{E}[G^\top G] \preceq \sigma^2 I$.	A distribution-agnostic scale bound gives worst-case distortion and response penalties. Isotropy attains the bound but is not assumed of actual gradients.

SC-GPM uses past activation usage to limit changes to earlier tasks' responses. The layerwise loss bound in Appendix C.1 motivates this penalty and accounts for differences between learn-time usage and current responses, including discarded spectral energy. Appendix C.7 connects the capacity reserve to the transmitted update.

B Proofs of the Limitation Theorems

B.1 Proof of Theorem 1

Each basis addition uses a new input-space direction, so a fixed layer dimension permits only finitely many additions. Once additions stop, accumulated strength determines whether protection along the retained directions becomes hard.

Proof. **Finite allocation.** Write $r_t = \dim S_t$. Nested spans give nonnegative integer increments $k_t = r_t - r_{t-1}$, so $\mathbf{1}\{k_t > 0\} \le k_t$. For every N,

$$\sum_{t=1}^{N} \mathbf{1}\{k_t > 0\} \le \sum_{t=1}^{N} (r_t - r_{t-1}) = r_N - r_0 \le R - r_0.$$

The left side increases to the total number of addition events as $N \to \infty$, proving the bound in Theorem 1(i). A finite nonempty set of addition indices has a maximum t_0, and if there are no additions we set $t_0 = 0$. For $t \ge t_0$, nestedness and equal dimensions imply $S_t = S_{t_0}$. Because old columns are retained unchanged, the stored basis itself is fixed.

Monotone protection. Write $P_t = \sum_{i=1}^{r_t} \gamma_{i,t} m_i m_i^\top$. Retained directions are unchanged and every strength increment is nonnegative. Therefore

$$P_{t+1} - P_t = \sum_{i=1}^{r_t} (\gamma_{i,t+1} - \gamma_{i,t}) m_i m_i^\top + \sum_{i=r_t+1}^{r_{t+1}} \gamma_{i,t+1} m_i m_i^\top \succeq 0.$$

Each summand is a nonnegative multiple of a PSD rank-one matrix. This conclusion needs no assumption that strengths reach one.

Strength saturation. For $a, b \ge 0$, $\min\{1, \min\{1, a\} + b\} = \min\{1, a + b\}$: if $a \le 1$ the inner minimum is a; if $a > 1$ both sides equal one. At the allocation time τ_i, (2) holds with an empty sum. Applying this identity to the recursion extends it from t to $t + 1$, proving the formula by induction. Define the first time the accumulated strength reaches one,

$$T_i = \inf \left\{ t \ge \tau_i : \gamma_{i,\tau_i} + \sum_{s=\tau_i+1}^{t} \delta_{i,s} \ge 1 \right\},$$

with $\inf \varnothing = \infty$. If $T_i < \infty$, nonnegative increments ensure $\gamma_{i,t} = 1$ for every $t \ge T_i$. For example, increments at least $b_i > 0$ on $N_i \ge \lceil (1 - \gamma_{i,\tau_i})/b_i \rceil$ later occasions suffice, because their sum is at least $1 - \gamma_{i,\tau_i}$.

After t_0 there are finitely many retained columns. If all their T_i are finite, $T_\star = \max\{t_0, T_1, \ldots, T_{r_{t_0}}\}$ is finite. For $t \ge T_\star$,

$$P_t = \sum_{i=1}^{r_{t_0}} m_i m_i^\top = \Pi_{S_{t_0}}, \qquad Q_t = I - \Pi_{S_{t_0}}.$$

Orthonormality makes $\Pi_{S_{t_0}}$ an orthogonal projector. Thus $\Delta W (I - \Pi_{S_{t_0}}) = G(I - \Pi_{S_{t_0}})^2 = \Delta W$, so each update row lies in the same orthogonal complement of dimension $D - r_{t_0}$. □

Allocation and saturation times. The bound limits the number of additions, not when the last one occurs. In $\mathbb{R}^2$, a memory can retain e_1 at task 1 and add e_2 at an arbitrarily late task N. Monotone strengths also need not reach one at a finite task. Starting from zero, increments $\delta_{i,\tau_i+k} = 2^{-k}$ give strength $1 - 2^{-k} < 1$ after k increments. A summable sequence can instead converge below one.

B.2 Layer Response under Gradient Protection

Hard protection confines an applied update to the complement of the protected span. Let Π be its orthogonal projector and suppose $\Delta W \Pi = 0$. Then $\Delta W = \Delta W (I - \Pi)$, so only the covariance in that complement contributes to the direct response. Set $K = (I - \Pi)\Sigma(I - \Pi) \succeq 0$. Then

$$\begin{aligned} \mathbb{E}\|\Delta W x\|^2 &= \operatorname{tr}(\Delta W \Sigma \Delta W^\top) = \operatorname{tr}(\Delta W K \Delta W^\top) \\ &\le \|\Delta W\|_2^2 \operatorname{tr} K = \|\Delta W\|_2^2 \{\operatorname{tr} \Sigma - \operatorname{tr}(\Pi\Sigma)\}. \end{aligned}$$

The inequality follows from $\Delta W^\top \Delta W \preceq \|\Delta W\|_2^2 I$ and trace cyclicity. Because $(I - \Pi)^2 = I - \Pi$, we also have $\operatorname{tr} K = \operatorname{tr}[\Sigma(I - \Pi)]$. This proves the preactivation bound in Section 3.1. If $\operatorname{tr}(\Pi\Sigma) \ge \varepsilon \operatorname{tr} \Sigma$, the uncovered energy gives the stated specialization. For soft protection, let the eigenvectors of the operator span an invariant subspace S. If $0 \le \delta \le 1$, $p_i \ge 1 - \delta$ on S, and inputs are supported in S, then $\|Qx\| \le \delta \|x\|$. Consequently,

$$\mathbb{E}\|GQx\|^2 \le \delta^2 \|G\|_2^2 \operatorname{tr} \Sigma.$$

Thus near-saturation also contracts responses continuously before the exact hard limit.

GPM selects new directions from the part of a task representation that its existing memory does not cover. For a representation matrix X and $\Sigma = XX^\top/N$, it forms $X_\perp = (I - \Pi_{t-1})X$ and appends leading left singular vectors of this residual. Each appended vector m_i with positive singular value lies in $S_{t-1}^\perp$. The old and new projections therefore have zero Frobenius inner product, and $\|m_i^\top X\|_2^2 = \|m_i^\top X_\perp\|_2^2 = \sigma_i(X_\perp)^2$. Pythagorean addition gives

$$\|\Pi_t X\|_F^2 = \|\Pi_{t-1} X\|_F^2 + \sum_{i=1}^{k_t} \sigma_i(X_\perp)^2.$$

Choosing k_t so that the right side reaches $\varepsilon\|X\|_F^2$ establishes empirical coverage, provided the rank cap permits those additions. The same argument applies to a union that contains each task's retained leading eigenspace. For a fixed statistic, a larger energy threshold requires a larger prefix; training can also change the statistic itself.

B.3 Proof of Theorem 2

Under PSD accumulation, the ordered eigenvalues of the historical state are nondecreasing. Applying a fixed nondecreasing scalar rule therefore makes total gradient capacity nonincreasing. Matrix order requires the stronger operator-monotone condition, while the limiting capacity also depends on divergence of the positive eigenvalues.

Proof. **Total capacity.** Order eigenvalues increasingly. Since $R_{t+1} - R_t = B_{t+1} \succeq 0$, the min–max characterization gives $\lambda_i(R_{t+1}) \geq \lambda_i(R_t)$ for every i. Applying the nondecreasing scalar function f term by term yields

$$\operatorname{tr} Q_{t+1} = \sum_{i=1}^{D} [1 - f(\lambda_i(R_{t+1}))] \leq \sum_{i=1}^{D} [1 - f(\lambda_i(R_t))] = \operatorname{tr} Q_t.$$

This argument compares eigenvalues and does not require the two matrices to share eigenvectors.

Matrix monotonicity. An operator-monotone f preserves PSD order by definition, so $P_{t+1} = f(R_{t+1}) \succeq f(R_t) = P_t$. For the regularized rule f_a, set $A = R_t + aI \succ 0$ and $H = A^{-1/2} B_{t+1} A^{-1/2} \succeq 0$. Then

$$(R_{t+1} + aI)^{-1} = A^{-1/2}(I + H)^{-1} A^{-1/2} \preceq A^{-1},$$

since each eigenvalue of $(I + H)^{-1}$ is at most one. Multiplying by a proves $Q_{t+1} \preceq Q_t$ and hence $P_{t+1} \succeq P_t$.

Limiting capacity. If $x \in \ker R_{t+1}$, then $0 = x^\top R_{t+1} x = x^\top R_t x + x^\top B_{t+1} x$. Both terms are nonnegative. Their vanishing gives $R_t^{1/2} x = 0$ and thus $x \in \ker R_t$. The kernels form a nested decreasing sequence. Every strict decrease lowers their integer dimension, so there are at most D such events. After the last one the kernel is fixed, and we write its dimension as n. Thereafter the first n eigenvalues are zero. Because $f(0) = 0$,

$$\operatorname{tr} Q_t = n + \sum_{i=n+1}^{D} [1 - f(\lambda_i(R_t))].$$

If each of the remaining eigenvalues diverges, each summand tends to zero by $f(\lambda) \to 1$. The sum has finitely many terms, proving $\operatorname{tr} Q_t \to n$. The full-rank case has $n = 0$. □

Scalar and matrix monotonicity. Consider $f(\lambda) = \mathbf{1}\{\lambda \geq 1\}$, $R = \operatorname{diag}(1.5, .5)$ and $B = .2vv^\top$ with $v = (1, 1)^\top/\sqrt{2}$. Both R and $R + B$ have one eigenvalue above one and one below it, so capacity is unchanged. Their leading eigenvectors differ because $R + B$ has a nonzero off-diagonal entry. Thus $f(R + B) - f(R)$ is a nonzero symmetric matrix of trace zero and must have both positive and negative eigenvalues. The trace conclusion holds, but PSD monotonicity does not.

The accumulated-state identity applies to an untruncated conceptor union at a fixed regularization scale. Write $C(R) = R(R + aI)^{-1}$. For positive-definite R,

$$(I - C(R))^{-1} = I + R/a.$$

Using the conceptor OR operation yields

$$\begin{aligned} C(R_1) \vee C(R_2) &= I - \left[(I - C(R_1))^{-1} + (I - C(R_2))^{-1} - I\right]^{-1} \\ &= I - [I + (R_1 + R_2)/a]^{-1} = C(R_1 + R_2). \end{aligned}$$

The identity extends to PSD inputs by continuity. It characterizes the accumulated protection component when the regularization scale a is fixed. The conceptor literature parameterizes a as the inverse squared aperture. For the regularized OWM expression, let A_t collect historical input vectors as columns. The identity $(A_t A_t^\top + aI)A_t = A_t(A_t^\top A_t + aI)$ implies

$$I - A_t(A_t^\top A_t + aI)^{-1}A_t^\top = I - (A_t A_t^\top + aI)^{-1}A_t A_t^\top = a(A_t A_t^\top + aI)^{-1}.$$

Thus the fixed-a expression has the required additive state $R_t = A_t A_t^\top$. This identity characterizes the normalized regularized operator of OWM, whose authors also give recursive implementations with separate initialization and scaling parameters (Zeng et al., 2019). The same authors observed empirically that the capacity of this projector decreases as tasks accumulate, and Theorem 2 proves this for every fixed spectral rule.

B.4 Applicability of the Limitation Theorems

Other gradient-based constraints. Gradient-based continual learning also includes replay-derived loss constraints (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019a) and output-gradient memories (Farajtabar et al., 2020). These methods use different information to constrain subsequent updates. Our analysis concerns activation-space protection, where stored input directions and their assigned strengths determine the restriction applied at each layer. The following discussion identifies the projection components to which the assumptions of our limitation theorems apply.

Retaining earlier activation directions limits how often a memory can grow, while a fixed strength rule can progressively restrict updates as usage accumulates. Theorems 1 and 2 establish these results at fixed layer dimension. Table 6 identifies the conditions for each mechanism in Table 1 and for regularized OWM. A nested span bounds the number of rank increases; fixing the stored basis also requires its columns to remain unchanged. With a changing basis, nondecreasing scalar strengths alone do not imply monotone protection matrices.

Relation to existing methods. FS-DGPM refreshes its memory using replay and learns releasable strengths (Deng et al., 2021). Adam-NSCL selects eigenvalues relative to a changing spectral scale, and AdNS adapts the null-space construction (Wang et al., 2021; Kong et al., 2022). GPCNS recomputes, in its Eq. (10),

$$\gamma_{t,i} = \frac{(\alpha + 1)\sigma_{t,i}}{\alpha\sigma_{t,i} + \max_j \sigma_{t,j}},$$

The current singular values determine this strength, followed by the projection in its Eq. (11) (Yang et al., 2024). The rule uses SGP's weighting function but recomputes strengths instead of clipping increments to old importance. Put $s_{t,i} = \sigma_{t,i}/\max_j \sigma_{t,j}$. Then $\gamma_{t,i} = (\alpha + 1)s_{t,i}/(\alpha s_{t,i} + 1)$ depends only on the normalized spectrum. Multiplying every singular value by the same constant leaves each strength, the relative energy cutoff, and $\operatorname{tr} Q$ unchanged. A change in spectral shape can still alter the strengths and selected span: at $\alpha = 1$, spectra $(2, 1)$ and $(4, 1)$ assign the second direction $2/3$ and $2/5$. Thus accumulation alone cannot drive this rule toward hard protection. Theorem 2 assumes a fixed rule applied to a PSD-additive state, and this construction supplies neither condition. Whether its capacity decays depends on the stream. Collaborative variants that add a GPM, TRGP, or SGP memory retain that memory's separate scope under Theorem 1.

For CODE-CL, ideal untruncated fixed-aperture conceptor OR gives the PSD-additive state proved above. Its deployed singular-value truncation and shared-space transfer lie outside this exact identity (Apolinario et al., 2025). These algebraic conditions, rather than method names, determine the scope of Theorem 2.

Table 6: Scope of the limitation theorems at fixed layer dimension. Each result concerns the memory or protection component named in the first column, not the complete method.

Method / component	Applicable result	Why the condition holds, or what must be checked
Append-only memories		
GPM; DFGP; PGP	Theorem 1(i), (ii) for hard protection	Residual directions are appended to retained orthonormal columns. Unit strengths give hard protection from allocation. DFGP changes training, and PGP applies the construction to prompt-related inputs.
SGP	Theorem 1(i), (ii)	The basis is appended, and old strengths receive nonnegative increments clipped at one. Finite hard saturation requires the stated cumulative-increment condition.
TRGP; CUBER; ROGO	Theorem 1(i) for the historical union	Retained task spaces or the GPM memory accumulate directions. Transfer scalings, correlation-based relaxation and a relaxing subspace respectively modify the applied update, so (ii) concerns the stored span rather than the deployed operator.
API	Theorem 1(i) at fixed D	Its GPM memory appends directions, while architectural expansion changes D and is outside a single fixed-dimensional bound.
SD; BECAME	Theorem 1(i) for a retained GPM component	SD retains a historical feature union but recomputes shared/residual constraints. BECAME inherits its base projection memory and adds model merging. Their complete protection is base-dependent.
HLOP; QGPM	Theorem 1(i) for the exact nested-span construction	HLOP consolidates old lateral connections and learns additional subspaces. QGPM compresses an appended GPM basis. Exact orthonormality and projector conclusions require the ideal subspace representation, not finite Hebbian error or quantization error.
Recomputed directions under a fixed spectral rule		
Ideal cumulative conceptor; fixed-scale regularized OWM	Theorem 2	The exact untruncated operators use $f_a(\lambda) = \lambda/(\lambda + a)$ with a PSD-additive state. The limiting claim requires diverging positive eigenvalues. Deployed CODE-CL includes truncation and shared-space transfer. Its runs end with under 1% free capacity in most layers (Appendix F.1).
Fixed absolute spectral threshold; SC-GPM without a reserve	Theorem 2, trace conclusion	Respectively $f(\lambda) = \mathbf{1}\{\lambda \geq \theta\}$, $\theta > 0$, and $f(\lambda) = \min\{\lambda, 1\}$. Neither rule generally preserves PSD order.
Rules that accumulation alone does not saturate		
FS-DGPM	Outside both theorems	Replay refreshes the basis and learned importance can decrease, so its protection is already releasable. The retained-column and monotone-strength premises do not apply. SC-GPM releases protection without a replay buffer.
Adam-NSCL; AdNS	Outside both theorems	Adam-NSCL selects directions by a threshold relative to the current spectrum and AdNS adapts the null-space construction, so the deployed rule is not fixed across tasks, and Adam-NSCL's rule is unchanged by rescaling the historical state. Accumulation alone therefore does not saturate protection. Adam-NSCL represents this design empirically.
GPCNS	Outside both theorems	It reselects singular vectors and sets strengths from the normalized spectrum $\sigma_{t,i}/\max_j \sigma_{t,j}$ of the current statistics, so rescaling those statistics leaves the deployed operator unchanged. This differs from SGP's clipped accumulation, and an added GPM, TRGP or SGP memory keeps its own scope above.
SC-GPM with reserve	Proposition 1	The state-dependent multiplier enforces $\operatorname{tr} Q \geq \kappa$ while allowing directional release, so the deployed spectral rule is not fixed.

C Analysis of Spectral Allocation

C.1 Derivation of the Response Penalty

To relate changes in layer responses to past-task loss, we vary one layer while keeping its current input representations and all other weights fixed. For past task τ, let $x_\tau(z)$ be the layer input and $\ell_\tau(o, z)$ its loss as a function of preactivation $o = Wx_\tau(z)$. Define

$$\Sigma_\tau^{\mathrm{cur}} = \mathbb{E}[x_\tau x_\tau^\top], \qquad s_\tau^{\mathrm{cur}} = \|\Sigma_\tau^{\mathrm{cur}}\|_2, \qquad \Gamma^{\mathrm{cur}} = \sum_{\tau: s_\tau^{\mathrm{cur}} > 0} \frac{\Sigma_\tau^{\mathrm{cur}}}{s_\tau^{\mathrm{cur}}}.$$

Hold these scales fixed during the local calculation and normalize each task by its activation scale: $\mathcal{J}_{\mathrm{past}}(W) = \sum_{\tau: s_\tau^{\mathrm{cur}} > 0} \mathcal{L}_\tau(W)/s_\tau^{\mathrm{cur}}$. Suppose ℓ_τ is twice differentiable along the preactivation segment produced by the update and satisfies $\nabla_o^2 \ell_\tau \preceq LI$ there. With $h = \Delta W x_\tau$, the integral form of Taylor's theorem gives

$$\begin{aligned} \ell_\tau(o + h, z) - \ell_\tau(o, z) &= \nabla_o \ell_\tau(o, z)^\top h + \int_0^1 (1 - u) h^\top \nabla_o^2 \ell_\tau(o + uh, z) h \, du \\ &\le \nabla_o \ell_\tau(o, z)^\top h + \frac{L}{2} \|h\|^2. \end{aligned} \tag{6}$$

Taking expectations, normalizing and summing yields

$$\Delta \mathcal{J}_{\mathrm{past}} \le \langle g_{\mathrm{past}}, \Delta W \rangle_F + \frac{L}{2} \operatorname{tr}(\Delta W \Gamma^{\mathrm{cur}} \Delta W^\top), \qquad g_{\mathrm{past}} = \nabla_W \mathcal{J}_{\mathrm{past}}. \tag{7}$$

The first-order term remains because earlier tasks need not be stationary at the current model. For piecewise-smooth networks, the bound applies within a fixed smooth region. The quadratic term measures the response to the layer update through current past-task representations, as in layerwise curvature models (Martens & Grosse, 2015).

Approximation with stored task usage. SC-GPM records activation usage when each task is learned. Let $\Sigma_\tau^{\mathrm{learn}}$ denote the moment recorded at task boundary τ and write $\Gamma^{\mathrm{learn}} = \sum_\tau \Sigma_\tau^{\mathrm{learn}} / \|\Sigma_\tau^{\mathrm{learn}}\|_2$, omitting zero moments. If E_τ is the normalized discarded spectral tail, $m_\alpha(x) \ge x$ gives

$$\Gamma^{\mathrm{learn}} \preceq \Omega + \sum_\tau E_\tau, \qquad E_\tau \succeq 0, \quad \operatorname{tr} E_\tau \le (1 - \varepsilon) \frac{\operatorname{tr} \Sigma_\tau^{\mathrm{learn}}}{\|\Sigma_\tau^{\mathrm{learn}}\|_2}. \tag{8}$$

The usage operator retains at least the normalized weight assigned to each selected direction. The discarded tail accounts for the omitted spectral energy. To compare this stored state with responses at the current model, set $\mathcal{E} = \Gamma^{\mathrm{cur}} - \Omega$. Then

$$\begin{aligned} \operatorname{tr}(\Delta W \Gamma^{\mathrm{cur}} \Delta W^\top) &= \operatorname{tr}(\Delta W \Omega \Delta W^\top) + \operatorname{tr}(\mathcal{E} \Delta W^\top \Delta W) \\ &\le \operatorname{tr}(\Delta W \Omega \Delta W^\top) + \|\mathcal{E}\|_2 \|\Delta W\|_F^2. \end{aligned} \tag{9}$$

SC-GPM computes the allocation from stored usage, without retaining past examples or recomputing their gradients. Equations (7)–(9) give the corresponding response penalty and replacement error.

Simultaneous updates across layers also produce mixed curvature terms $\sum_{l<m} \langle \Delta W_l, H_{lm}[\Delta W_m] \rangle_F$, alongside $\frac{1}{2} \sum_l \langle \Delta W_l, H_{ll}[\Delta W_l] \rangle_F$ and the first-order term. We optimize a layerwise surrogate that balances update distortion with a linear upper bound on the past-response penalty. The spectral solution is exact for this surrogate.

C.2 Worst-Case Bounds and Linear Relaxation

We penalize both changes to the proposed update and changes to responses along previously used activation directions. For a proposed update G, put $K = \mathbb{E}[G^\top G]$. Trace cyclicity gives

$$\mathbb{E}\|G - GQ\|_F^2 = \operatorname{tr}[K(I - Q)^2], \qquad \mathbb{E} \operatorname{tr}(GQ\Omega QG^\top) = \operatorname{tr}(KQ\Omega Q).$$

For every $A \succeq 0$ and $0 \preceq K \preceq \sigma^2 I$,

$$\operatorname{tr}(KA) \le \sigma^2 \operatorname{tr} A,$$

with equality at $K = \sigma^2 I$. Both $(I-Q)^2$ and $Q\Omega Q$ are PSD, so this proves the two worst-case identities in the main text. It also gives the joint identity

$$\sup_{0 \preceq K \preceq \sigma^2 I} \left\{ \tfrac{1}{2} \operatorname{tr}[K(I-Q)^2] + \operatorname{tr}(KQ\Omega Q) \right\} = \sigma^2 \left\{ \tfrac{1}{2} \|I-Q\|_F^2 + \operatorname{tr}(\Omega Q^2) \right\}.$$

The second-moment class is attainable, for example by a random rank-one G whose single nonzero row is $\sigma\sqrt{D}\, e_i^\top$ with i uniform. Finally,

$$\operatorname{tr}(\Omega Q) - \operatorname{tr}(\Omega Q^2) = \operatorname{tr}[\Omega^{1/2}(Q - Q^2)\Omega^{1/2}] \ge 0,$$

This nonnegative difference justifies the linear upper bound. Its slack depends on Q and vanishes for hard projectors. Dividing by σ^2 and setting the relative coefficient to one gives (4). We optimize this upper bound to obtain a threshold that can fully protect high-usage directions and release low-usage directions. Theorem 3 gives its exact minimizer.

C.3 Proof of Theorem 3

We first show that the minimizer shares an eigenbasis with Ω, then solve one scalar gain per direction under the trace constraint. For $0 \le \kappa \le D$, the feasible set in (4) contains I and is closed and bounded. The continuous, strictly convex objective therefore has a unique minimizer. Write $H = V^\top Q V$ in a usage eigenbasis. Each diagonal entry $h_i = H_{ii}$ lies in $[0,1]$, and $\sum_i h_i = \operatorname{tr} Q \ge \kappa$. Thus $\operatorname{diag}(h_i)$ is feasible. Moreover,

$$F_\Omega(Q) = \sum_i \left[\frac{1}{2}(1-h_i)^2 + \lambda_i h_i \right] + \sum_{i<j} H_{ij}^2. \tag{10}$$

The last term is strictly positive if any off-diagonal entry is nonzero. The minimizer must therefore be diagonal in this basis.

For $0 \le \kappa < D$, a vector with every coordinate in $(\kappa/D, 1)$ gives strict feasibility. The scalar Karush–Kuhn–Tucker (KKT) conditions are necessary and sufficient (Boyd & Vandenberghe, 2004). Let $c \ge 0$ multiply $\kappa - \sum_i q_i$, and $a_i, b_i \ge 0$ multiply $-q_i$ and $q_i - 1$, respectively:

$$q_i - 1 + \lambda_i - c - a_i + b_i = 0,$$

$$a_i q_i = 0, \quad b_i(q_i - 1) = 0, \quad c\left(\kappa - \sum_i q_i\right) = 0. \tag{11}$$

For fixed c, each box-constrained coordinate minimizes $\frac{1}{2}(1-q_i)^2 + (\lambda_i - c)q_i$. Its unconstrained minimizer is $1 + c - \lambda_i$, and restricting it to $[0,1]$ gives $q_i = \operatorname{clip}_{[0,1]}(1 + c - \lambda_i)$. If $c = 0$ is feasible, it is the unconstrained box minimizer and therefore solves the problem. Otherwise define

$$f(c) = \sum_i \operatorname{clip}_{[0,1]}(1 + c - \lambda_i).$$

This function is continuous and nondecreasing, with $f(0) < \kappa$ and $f(\lambda_{\max}) = D$. Thus a positive multiplier satisfying $f(c) = \kappa$ exists. If several multipliers attain that trace, their coordinatewise nondecreasing gain vectors have the same sum, so every coordinate agrees. The operator remains unique.

For $\kappa = D$, feasibility forces $Q = I$ and any $c \ge \lambda_{\max}$ yields that solution. For $\kappa = 0$, the unconstrained box solution is always feasible. Repeated eigenvalues receive equal gains, so the result is invariant under arbitrary rotations within a repeated-eigenvalue block. This completes Theorem 3.

C.4 Projection Formulation and Sensitivity

We can also obtain optimal protection by projecting accumulated usage onto the feasible set. Define $\mathcal{C}_\kappa = \{P = P^\top : 0 \preceq P \preceq I,\ \operatorname{tr} P \le D - \kappa\}$. Substituting $Q = I - P$ gives

$$F_\Omega(I-P) = \frac{1}{2}\|P\|_F^2 - \operatorname{tr}(\Omega P) + \operatorname{tr}\Omega = \frac{1}{2}\|P - \Omega\|_F^2 + \operatorname{tr}\Omega - \frac{1}{2}\|\Omega\|_F^2.$$

The final two terms do not depend on P, proving $P^\star = \Pi_{\mathcal{C}_\kappa}(\Omega)$.

Euclidean projection is nonexpansive (Boyd & Vandenberghe, 2004): at fixed κ, a change in the usage operator cannot produce a larger Frobenius-norm change in optimal protection.

C.5 Total Protection and Directional Release

Under PSD task additions, the ordered eigenvalues of Ω_t are nondecreasing: $\lambda_i(t+1) \geq \lambda_i(t)$. Hence unconstrained capacity $C_t^{\mathrm{unc}} = \sum_i \mathrm{clip}_{[0,1]}(1 - \lambda_i(t))$ satisfies $C_{t+1}^{\mathrm{unc}} \leq C_t^{\mathrm{unc}}$. While it exceeds κ, the unconstrained profile is optimal; once it would fall below κ, the KKT solution fixes total capacity at κ. This proves Proposition 1, including the initialization $\Omega_0 = 0$.

Directional release with an inactive reserve. Let $\kappa = .25$, and

$$A = \begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix}, \qquad B = A + \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

Both states are reserve-inactive: $\operatorname{tr} Q^\star(A) = 1$ and $\operatorname{tr} Q^\star(B) = \sqrt{2} - 1 > .25$. The saturated rule gives $e_2^\top P^\star(A) e_2 = 1$ but $e_2^\top P^\star(B) e_2 = 2 - \frac{3\sqrt{2}}{4} < 1$. Protection along e_2 therefore decreases under a PSD state addition, even though total protection increases. The allocation redistributes protection as the joint spectrum changes, which is why Proposition 1 is stated for total protection. Before any saturation and with an inactive reserve, $P = \Omega$ and PSD order is preserved.

Directional release with an active reserve. For $\kappa = 1$, take

$$\Omega_a = \mathrm{diag}(2, 4), \qquad c_a = 2, \qquad Q_a^\star = \mathrm{diag}(1, 0).$$

After adding $\mathrm{diag}(2, 0) \succeq 0$,

$$\Omega_b = \mathrm{diag}(4, 4), \qquad c_b = 3.5, \qquad Q_b^\star = \mathrm{diag}(.5, .5).$$

Both operators satisfy the KKT conditions and have trace one. Thus gain can move to a previously blocked direction under additive historical usage.

C.6 Proof of Proposition 2

We bound the cost of keeping an inherited basis by the part of optimal protection that it cannot represent. Projection optimality converts this difference into an objective gap.

Proof. Let $A = \Omega$ and $P^\star = \Pi_{\mathcal{C}_\kappa}(A)$. For feasible P, projection optimality gives $\langle P^\star - A, P - P^\star \rangle_F \geq 0$. Completing the square yields

$$\begin{aligned} F_\Omega(I - P) - F_\Omega(I - P^\star) &= \tfrac{1}{2}\|P - A\|_F^2 - \tfrac{1}{2}\|P^\star - A\|_F^2 \\ &= \tfrac{1}{2}\|P - P^\star\|_F^2 + \langle P^\star - A, P - P^\star \rangle_F \\ &\geq \tfrac{1}{2}\|P - P^\star\|_F^2. \end{aligned}$$

If $M^\top P M$ is diagonal, orthogonal invariance and the orthogonality of diagonal and off-diagonal matrix entries give

$$\begin{aligned} \|P - P^\star\|_F^2 &= \|M^\top P M - \mathrm{diag}(\mathrm{diag}(M^\top P^\star M))\|_F^2 + \|\operatorname{off}(M^\top P^\star M)\|_F^2 \\ &\geq \|\operatorname{off}(M^\top P^\star M)\|_F^2. \end{aligned}$$

Combining the two inequalities proves the claim. □

A stored partial basis can be extended to an orthonormal basis. The result then applies to candidates whose protection is diagonal in that extension.

C.7 Task Usage and Gradient Capacity

Task-spectrum weighting and the hard-protection limit. For $x \in [0, 1]$,

$$\frac{\partial m_\alpha(x)}{\partial \alpha} = \frac{x(1-x)}{(1+\alpha x)^2} \geq 0, \qquad x \leq m_\alpha(x) \leq 1, \qquad m_\alpha(1) = 1.$$

For a nonzero retained moment, U_t is PSD with $\|U_t\|_2 = 1$, so $0 \preceq \Omega_t \preceq tI$. Rescaling activations multiplies every $s_{t,i}$ by the same positive factor and leaves both the energy prefix and $x_{t,i}$ unchanged. As $\alpha \to \infty$, task usage approaches the orthogonal projector onto its retained positive-energy span. With zero reserve and a single projector state, the optimizer becomes its hard complement. This recovers hard projection as a limiting case. Across tasks, nonorthogonal projectors can sum to eigenvalues below one, and the reserve can release protection; the allocation is therefore more general than GPM's update rule.

Gradient capacity and applied updates. A positive reserve leaves at least one direction available for updates: with $0 \preceq Q \preceq I$, $\operatorname{tr} Q \geq \kappa > 0$ ensures a nonzero transmission eigenvalue. Cauchy–Schwarz gives $\operatorname{tr} Q^2 \geq \kappa^2/D$. Under the isotropic second-moment model $\mathbb{E}[G^\top G] = \sigma^2 I$, this yields $\mathbb{E}\|GQ\|_F^2 \geq \sigma^2\kappa^2/D$. A particular gradient can still lie in $\ker Q$. For a gradient step $G = -h\nabla_W \mathcal{L}_{\text{new}}$ with $h > 0$, its first-order loss change is

$$\langle \nabla_W \mathcal{L}_{\text{new}}, GQ \rangle_F = -h \operatorname{tr}[(\nabla_W \mathcal{L}_{\text{new}})^\top \nabla_W \mathcal{L}_{\text{new}} Q] \leq 0.$$

Under the isotropic second-moment assumption above, its expectation equals $-(\sigma^2/h)\operatorname{tr} Q \leq -\sigma^2\kappa/h$. The reserve therefore guarantees an expected first-order decrease in new-task loss of at least $\sigma^2\kappa/h$ under this model, and positive gradient capacity holds unconditionally.

D Algorithm, Memory, and Computational Cost

D.1 Algorithm

For each protected layer choose $0 < \varepsilon \leq 1$, $\alpha \geq 0$, and $\kappa_l \in (0, D_l]$. Initialize $\Omega_l = P_l = 0$.

1. Train the current task with cross-entropy, using $G_l(I - P_l)$ at protected layers and keeping P_l fixed within the task.
2. Collect current-task input statistics with the evaluation transform. Compute the retained energy prefix and $U_{t,l}$ from (3).
3. Set $\Omega_l \leftarrow \Omega_l + U_{t,l}$ and diagonalize $\Omega_l = V_l \operatorname{diag}(\lambda_{l,i}) V_l^\top$.
4. Set $c_l = 0$ and form $q_{l,i} = \operatorname{clip}_{[0,1]}(1 - \lambda_{l,i})$. If $\sum_i q_{l,i} < \kappa_l$, bisect $c_l \in [0, \lambda_{l,\max}]$ until $\sum_i \operatorname{clip}_{[0,1]}(1 + c_l - \lambda_{l,i}) = \kappa_l$ to numerical tolerance.
5. Update $q_{l,i} = \operatorname{clip}_{[0,1]}(1 + c_l - \lambda_{l,i})$, then set $P_l = V_l \operatorname{diag}(1 - q_{l,i}) V_l^\top$ and continue to the next task.

The scalar sum is continuous, nondecreasing, and D_l-Lipschitz. Returning the upper bisection endpoint preserves the reserve inequality. Layerwise input dimensions for convolutions refer to unfolded patches.

The operator formulation acts on weight displacements. Momentum is reset at each task boundary. Fixed linear filtering within a task commutes with the linear accumulation of momentum when the other displacement terms are filtered consistently. Applied-displacement diagnostics in Figure 6 measure the actual resulting updates. Adam-NSCL retains its own Adam-based update procedure: its implemented transform is $Q = NN^\top/\sqrt{r}$ for a selected null-space basis of rank $r > 0$, and zero when $r = 0$. It therefore combines a hard subspace restriction with uniform scaling inside that subspace. Its recorded transform trace is $\operatorname{tr} Q$, whereas the other methods record $\operatorname{tr} P$. Figure 6 converts it to the common protection convention.

D.2 Memory Accounting

The memory comparison counts persistent auxiliary state. It excludes backbone parameters, growing task heads, optimizer state, reconstructible caches, and diagnostic records. One double-precision $D_l \times D_l$ matrix per protected layer gives the worst-case gradient-projection budget $B = 8\sum_l D_l^2$ bytes. SC-GPM stores one usage matrix per layer. It recomputes the support basis and deployed operator at each task boundary; these require working memory of the same order but no additional persistent state. Table 7 identifies the counted state for each method.

Table 7: Persistent state of the evaluated protection mechanisms, complementing the grouping in Table 1. D is the layer input dimension and T the evaluated sequence length reported for each method.

Method	Core state	Directions	Strength / adaptation	T
GPM (2021)	Basis, $O(D^2)$	Appended	Hard protection	20
SGP (2023)	Basis + weights, $O(D^2)$	Appended	Clipped accumulation	20
TRGP (2022)	Basis + task state	Appended	Hard base + trust-region transfer	20
Adam-NSCL (2021)	Covariance, $O(D^2)$	Recomputed	Hard subspace + normalization	25
CODE-CL (2025)	Conceptor + task state	Recomputed	Fixed-scale soft rule + shared-space transfer	20
SC-GPM	Usage matrix, $O(D^2)$	Recomputed	Joint allocation with capacity reserve	200

GPM's compact basis uses $8\sum_l D_l r_l$ persistent bytes; SGP also stores one importance value per retained column. Adam-NSCL stores cumulative moments of order B. Replay uses the same derived byte budget for 477 image examples or 7,824 PMNIST examples. TRGP stores a shared basis and historical task-specific spaces and scalings. These growing task-specific states stop receiving new allocations at the nominal twofold budget while keeping earlier information. Since TRGP's shared basis stays within B, its total stays at or below $2B$.

CODE-CL stores one conceptor per layer, exactly B, plus $O(TDK + TK^2)$ task-transfer state for K shared directions per task. It applies the same nominal limit to per-task trust regions. Each task is granted as a whole, so its final grant brings the image-stream total to 2.13 times B. On PMNIST, the configurations selected for final accuracy use low retained-energy thresholds, and GPM and SGP store fewer directions than at $\varepsilon = .985$ on images. Small conceptor apertures also leave too few directions for CODE-CL's overlap test to grant a trust region, so CODE-CL retains only the conceptor there. Table 8 reports the resulting persistent state at the end of each stream. CODE-CL also passes over current-task inputs before training to form the conceptor for its overlap test: one batch on image streams and 2,000 examples on PMNIST. Task-specific output heads grow with the number of image tasks, so the bounded-memory claim concerns auxiliary shared-backbone state. PMNIST keeps one shared head throughout.

Table 8: Persistent spectral state at the end of each stream, as a percentage of the budget $B = 8\sum_l D_l^2$ bytes, with the budget in decimal MB. GPM and SGP are reconstructed from one seed's recorded ranks, and SGP includes its stored importance. TRGP is the declared protocol bound, and CODE-CL is measured from its recorded trust regions.

Stream	Budget (MB)	GPM	SGP	Adam-NSCL	SC-GPM	TRGP	CODE-CL
DomainNet	17.97	88.9	90.1	100	100	≤ 200	213
OmniBenchmark	17.97	91.3	90.8	100	100	≤ 200	213
ImageNet-1K	17.97	92.4	93.3	100	100	≤ 200	213
PMNIST	6.20	18.9	20.8	100	100	≤ 200	100

D.3 Computational Cost

A task-boundary eigendecomposition costs $O(\sum_l D_l^3)$; dense application of protection costs $O(\sum_l d_l D_l^2)$ per update. Table 9 measures the resulting ImageNet-1K-100 runtime. SC-GPM

is 4% slower per task than fine-tuning, within 2% of GPM and SGP, and faster than TRGP, CODE-CL, and replay.

Table 9: Wall-clock training time per task on ImageNet-1K-100, averaged over the 100 tasks of the stream at the configurations of Table 2. Every run used one NVIDIA Quadro RTX 6000 GPU in the same software environment. Each task includes training and the collection of its statistics. Three-seed means and sample SD.

Method	Minutes per task	Relative to fine-tuning
Finetune	12.78 ± 0.03	1.00
ER	23.54 ± 0.04	1.84
GPM	13.04 ± 0.14	1.02
TRGP	14.68 ± 0.04	1.15
Adam-NSCL	12.92 ± 0.01	1.01
SGP	13.02 ± 0.06	1.02
CODE-CL	13.63 ± 0.10	1.07
SC-GPM	13.28 ± 0.11	1.04

E Experimental Setup

E.1 Evaluation Metrics

For a sequence of T tasks, let $A(t, j)$ denote accuracy on task j after learning task t. Following standard accuracy and transfer evaluation (Lopez-Paz & Ranzato, 2017; Saha & Roy, 2023), we define final average accuracy $\mathrm{ACC} = T^{-1} \sum_{j=1}^{T} A(T, j)$, learning accuracy $\mathrm{LA} = T^{-1} \sum_{j=1}^{T} A(j, j)$, and backward transfer $\mathrm{BWT} = (T-1)^{-1} \sum_{j=1}^{T-1} [A(T, j) - A(j, j)]$. They satisfy

$$\mathrm{ACC} = \mathrm{LA} + \frac{T-1}{T} \mathrm{BWT}. \tag{12}$$

Average anytime accuracy (Caccia et al., 2020; Yoo et al., 2024), evaluated here at task boundaries, is $\mathrm{AAA} = T^{-1} \sum_{t=1}^{T} \mathrm{ACC}_t$, where $\mathrm{ACC}_t = t^{-1} \sum_{j=1}^{t} A(t, j)$ is the average accuracy on the tasks learned so far after learning task t. Forward transfer compares each task's learning accuracy with the accuracy b_j of the Independent reference, trained on task j alone with the same seed and class split: $\mathrm{FWT} = T^{-1} \sum_{j=1}^{T} [A(j, j) - b_j]$. We follow the convention of Veniat et al. (2021), applied across tasks as in Lin et al. (2022b); Saha & Roy (2023).

E.2 Datasets and Training

The fixed phase orders place a domain, granularity, or semantic shift after substantial prior learning. DomainNet has 101 twenty-class tasks: 66 from real, clipart, painting, and infograph, followed by 35 from quickdraw and sketch. OmniBenchmark has 160 twenty-class tasks: 130 spanning creation, activity, aircraft, structure, material, car, vascular plant, region, and geological formation, followed by 30 bird tasks. ImageNet-1K has 100 ten-class tasks: 61 from 602 non-animal and eight animal classes, followed by 39 containing the remaining 390 animal classes. These shifts test whether protection permits adaptation late in the stream.

Images are stored at 112 pixels. Training uses random resized crops, horizontal flips, and RandAugment (Cubuk et al., 2020) with two operations and magnitude nine; statistics use evaluation transforms. DomainNet and OmniBenchmark reserve 10% for validation and 20% for test, with minimum class counts of 20 and 50 examples, respectively. PMNIST reserves 10% for validation and 10% for test. ImageNet-1K reserves 10% of training data for validation and uses the corpus's labeled validation split for testing. Image class splits and PMNIST permutations vary with seed.

Table 10 summarizes the training protocol. The image streams use a reduced ResNet-18 of width 10, following the smaller ResNet setting of GEM (Lopez-Paz & Ranzato, 2017), and task-specific cosine classifiers (Gidaris & Komodakis, 2018). Each task uses cosine learning-rate decay (Loshchilov & Hutter, 2017). Adam-NSCL retains its method-defining Adam optimizer (Kingma & Ba, 2015).

Reference methods and forward transfer. Joint trains one model on the union of tasks with the same total sample presentations as the sequential schedule. Independent trains each task alone from the shared initialization for the full per-task budget. Both references use seeds 0, 1, and 2 on every

Table 10: Training protocol. All sequential methods use the same per-task training budget.

Field	Image streams	PMNIST
Backbone	Reduced ResNet-18, width 10	$784 \to 400 \to 400$, no biases
Output	Task-specific cosine heads	Shared ten-way head
Protected layers	20 convolutions	Two hidden layers
Epochs / batch size	150 / 128	5 / 64
Optimizer	SGD, momentum .9	SGD, momentum .9
Adam-NSCL optimizer	Adam	Adam
Learning rate	.01 to .0001, cosine	.01 to .0001, cosine
Label smoothing	.1	.1
Statistics examples/task	2,048	2,000
Input	84×84 crops	Flattened 28×28

image stream and match the corresponding sequential run's class split, architecture, and training budget. We compute FWT within each seed, then report its mean and sample SD. Independent starts with fresh optimizer state and trains batch-normalization affine parameters for each standalone task. PMNIST uses independent random permutations of the same digits and has little structure for cross-task transfer, so we omit FWT and the Joint and Independent references there.

E.3 Hyperparameter Selection

Hyperparameters for the image streams. On the image streams, GPM, SGP, and TRGP use $\varepsilon = .985$; SGP uses scale one. TRGP uses top-$k = 2$ with correlation threshold .5, and Adam-NSCL uses threshold parameter ten. CODE-CL uses $K = 80$ shared directions, overlap threshold .5, and singular-value floor .2. The better of its two published apertures, 8 and 16, is 8 on every stream. SC-GPM uses $\varepsilon = .985$, $\alpha = 10$, and $\kappa_l = 1$. We select α from $\{1, 5, 10, 15\}$ on DomainNet validation data using a randomly sampled seed disjoint from evaluation seeds 0, 1, and 2. For the reserve intervention in Figure 3, a fraction r maps to each layer by

$$\kappa_l = \max\{1, \min\{D_l - 1, \mathrm{round}(rD_l)\}\}. \tag{13}$$

SC-GPM uses this selected setting unchanged on all three image streams. Baseline hyperparameters follow the authors' published values. A single-seed DomainNet check favors those settings: GPM reaches 60.28% ACC at $\varepsilon = .985$ versus 56.08% at .95; SGP reaches 66.28% at $(\varepsilon, \mathrm{scale}) = (.985, 1)$ versus 62.08% at $(.95, 1)$ and 65.71% at $(.985, 3)$.

PMNIST hyperparameter search. Data order is matched across methods for each experiment seed. Every projection method is tuned on one grid. SC-GPM, GPM, SGP, and TRGP share $\varepsilon \in \{.14, .28, .42, .56, .70, .84, .98\}$. SC-GPM's α, SGP's scale, and CODE-CL's aperture use $\{0, 1, 2, 5, 10, 20\}$; CODE-CL omits zero because its conceptor is undefined there. SC-GPM varies the reserve fraction over $r \in \{0, .15, \ldots, .90\}$ and converts it using (13). TRGP keeps top-$k = 2$. CODE-CL keeps $K = 80$, overlap threshold .5, and singular-value floor .2. Adam-NSCL uses threshold 1 because thresholds 1, 100, and 10^4 give the same result (Figure 15).

For each horizon, the configuration with the highest final accuracy on the selection seed determines final accuracy, learning accuracy, and BWT. A separate configuration maximizes average anytime accuracy. Each selected configuration is then evaluated over three seeds; Table 11 lists both selections. Because selection uses one seed, an unselected configuration can have a higher three-seed mean, as the 200-task α sweep in Appendix F.4 shows. Appendix F.3 varies the number of statistics examples without changing the selected 200-task configurations.

Ablation controls. The hard consolidated control isolates the memory construction from the protection profile. It uses SC-GPM's normalized task contributions, then hard-protects an ε-energy prefix of the accumulated usage spectrum. This keeps GPM's retained-energy criterion but replaces its appended residual basis with weighted joint usage. Comparing this control with GPM tests that state construction; comparing it with SC-GPM tests allocation applied to the same normalized-usage construction. Table 3 reports final accuracy, learning accuracy, and BWT for each image stream. The appended-allocation control keeps GPM's residual-SVD basis and solves (4) over protections

Table 11: Selected PMNIST configurations at each horizon. For each method, the ACC row gives the configuration selected for final accuracy and the AAA row the configuration selected for average anytime accuracy. The reserve κ_l is listed for the first and second protected layers.

Method	Metric	$T = 50$	$T = 100$	$T = 150$	$T = 200$
SC-GPM	ACC	$\varepsilon = .98$, $\alpha = 10$, $\kappa_l = (353, 180)$	$\varepsilon = .84$, $\alpha = 5$, $\kappa_l = (470, 240)$	$\varepsilon = .84$, $\alpha = 0$, $\kappa_l = (588, 300)$	$\varepsilon = .70$, $\alpha = 2$, $\kappa_l = (706, 360)$
	AAA	$\varepsilon = .98$, $\alpha = 20$, $\kappa_l = (470, 240)$	$\varepsilon = .98$, $\alpha = 20$, $\kappa_l = (470, 240)$	$\varepsilon = .98$, $\alpha = 20$, $\kappa_l = (470, 240)$	$\varepsilon = .98$, $\alpha = 10$, $\kappa_l = (588, 300)$
CODE-CL	ACC	aperture 1	aperture 1	aperture 1	aperture 2
	AAA	aperture 2	aperture 1	aperture 1	aperture 1
SGP	ACC	$\varepsilon = .70$, scale 5	$\varepsilon = .56$, scale 10	$\varepsilon = .56$, scale 2	$\varepsilon = .42$, scale 10
	AAA	$\varepsilon = .84$, scale 10	$\varepsilon = .70$, scale 20	$\varepsilon = .70$, scale 20	$\varepsilon = .70$, scale 20
GPM	ACC	$\varepsilon = .84$	$\varepsilon = .70$	$\varepsilon = .56$	$\varepsilon = .56$
	AAA	$\varepsilon = .84$	$\varepsilon = .84$	$\varepsilon = .70$	$\varepsilon = .70$
TRGP	ACC	$\varepsilon = .56$, top-$k = 2$, correlation .5	$\varepsilon = .14$, top-$k = 2$, correlation .5	$\varepsilon = .14$, top-$k = 2$, correlation .5	$\varepsilon = .42$, top-$k = 2$, correlation .5
	AAA	$\varepsilon = .56$, top-$k = 2$, correlation .5	$\varepsilon = .56$, top-$k = 2$, correlation .5	$\varepsilon = .56$, top-$k = 2$, correlation .5	$\varepsilon = .56$, top-$k = 2$, correlation .5
Adam-NSCL	ACC	threshold 1	threshold 1	threshold 1	threshold 1
	AAA	threshold 1	threshold 1	threshold 1	threshold 1

diagonal in that basis, assigning each stored direction m_i the strength of Theorem 3 with its usage $\omega_i = m_i^\top \Omega m_i$ in place of λ_i.

F Additional Empirical Results

F.1 Protection and Gradient Capacity

The reserve binds only in the three 1×1 shortcut convolutions ($D = 10, 20, 40$), as Section 5.4 reports. Without a reserve, SGP leaves the $D = 10$ shortcut with zero free capacity at 62–156 boundaries in eight of nine runs; updates to that residual transformation are then blocked. The reserve keeps at least one unit of gradient capacity in every shortcut (Proposition 1). Whether a later task can use that capacity depends on the direction of its gradient.

At the end of OmniBenchmark, dimension-weighted free capacity $100 \sum_l \operatorname{tr} Q_l / \sum_l D_l$ averages 13.21% for SC-GPM over three seeds, compared with 9.44% for GPM, 10.51% for SGP, and 0.29% for CODE-CL. At the end of each image stream, CODE-CL leaves under 1% free capacity in 17 to 18 of its 20 layers on average over seeds. This finite observation is consistent with the fixed-rule restriction analyzed in Theorem 2. Figure 4 measures basis growth directly. Averaged over layers, GPM, TRGP, and SGP add no direction at 76–90% of tasks by each image stream's end; additions become rare after the first 20 tasks. Figure 5 repeats Figure 1(e)–(g) on DomainNet-101 and ImageNet-1K-100. On both, SGP and CODE-CL protection strengths exceed .97 by task 20, while SC-GPM's total protection keeps growing and its learning accuracy stays closest to fine-tuning. Trace protection sums the strengths assigned to the activation directions. Figure 1(f) divides $\operatorname{tr} P$ by the number of directions with nonzero protection. SC-GPM's usage support can span the full input space while soft protection still transmits gradients. Figure 6 reports the normalized weighted overlap of the applied update with P.

F.2 Accuracy and Retention on Long Image Streams

We examine learning and retention throughout the stream to identify when SC-GPM's final accuracy gains develop. Figure 7 shows average accuracy, LA and BWT after each task.

In Figure 8, the sustained mean lead begins at task 40 on DomainNet, 27 on OmniBenchmark, 53 on ImageNet-1K and 71 on PMNIST-200, where CODE-CL leads early in the stream.

Figure 9 plots learning accuracy against BWT for every method. Table 2 reports FWT, defined in Appendix E. SC-GPM has the highest FWT among projection methods on every image stream. Its final ACC reaches 91.0%, 80.1%, and 76.9% of the offline Joint reference on DomainNet, OmniBenchmark, and ImageNet-1K, respectively, as ratios of three-seed means.

Table 12 reports average anytime accuracy on the image streams and at each PMNIST horizon.

Retention of earlier tasks. The first-cohort curves (Figure 10) separate the initial adjustment from later retention. CODE-CL's near-complete retention coincides with high measured protection and lower learning accuracy (Table 2).

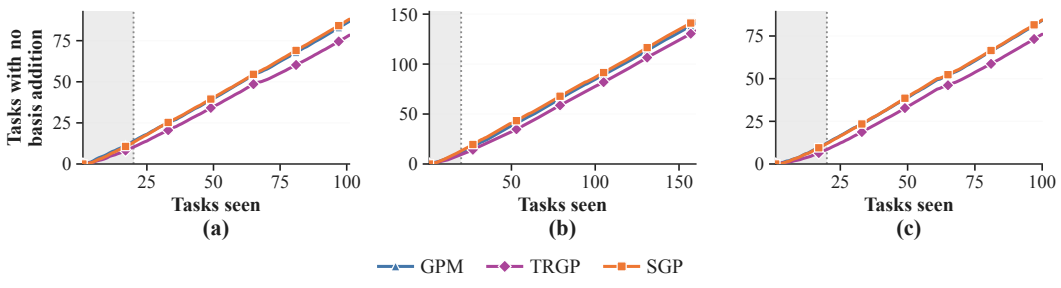


Figure 4: Cumulative number of tasks that add no basis direction to the appended memories of GPM, TRGP and SGP, averaged over the 20 protected layers, on (a) DomainNet-101, (b) OmniBenchmark-160 and (c) ImageNet-1K-100. Three-seed means and sample SD, and shading marks the first 20 tasks.

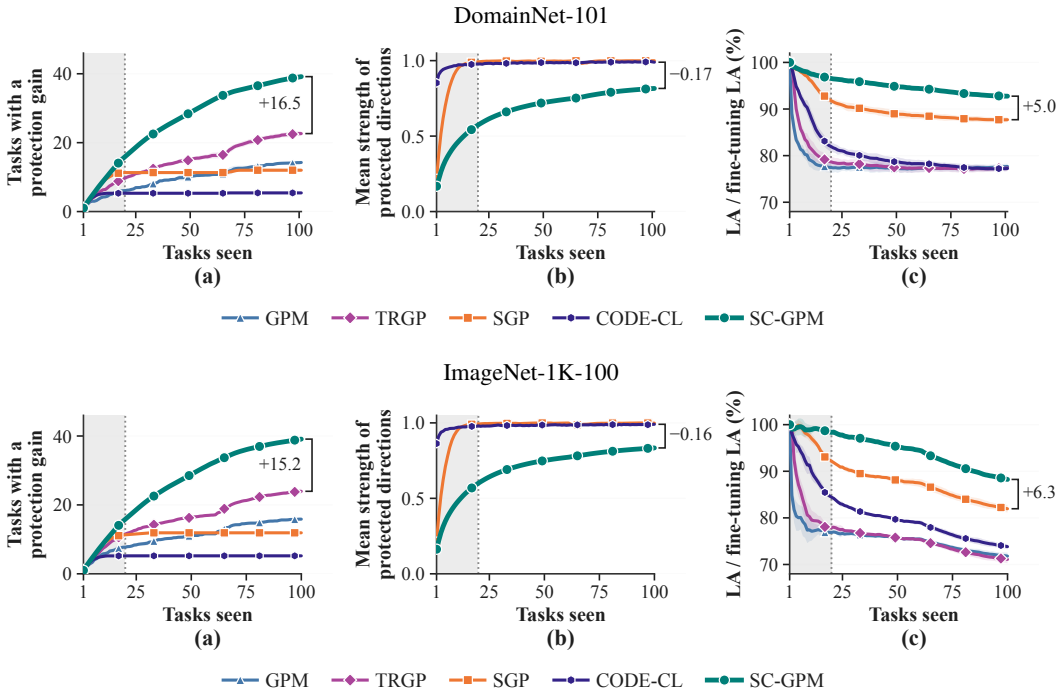


Figure 5: Protection and acquisition on DomainNet-101 (top) and ImageNet-1K-100 (bottom). Panels (a)–(c) in each row are defined as in Figure 1(e)–(g).

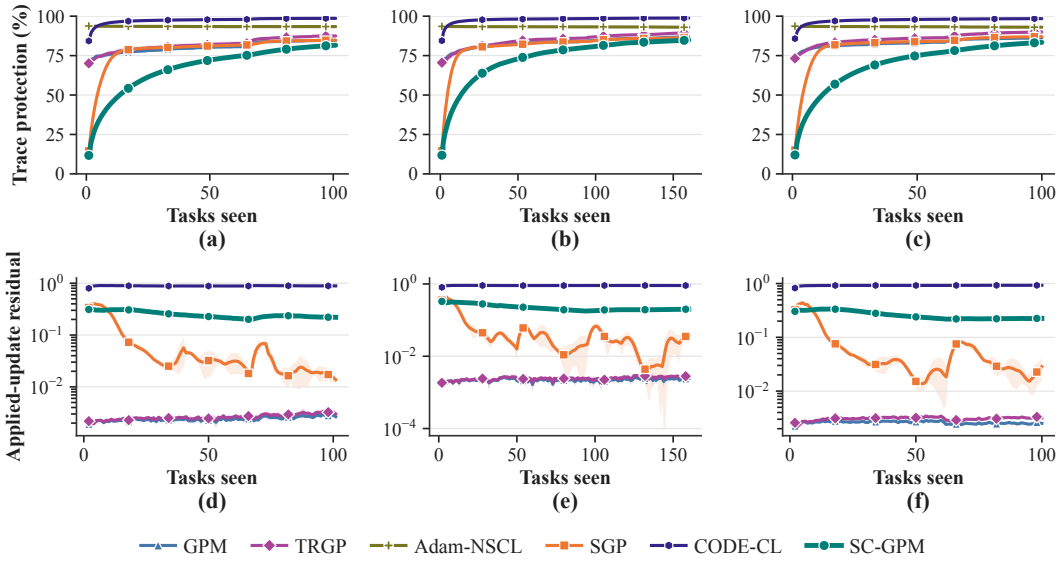


Figure 6: Trace protection and the residual of applied updates. (a–c) The protected share of each layer's trace, $100\,\mathrm{tr}(P)/D$, after each task on DomainNet-101, OmniBenchmark-160 and ImageNet-1K-100, averaged over protected layers. (d–f) The ratio $\|\Delta W P\|_F / \|\Delta W\|_F$ for the operator deployed while learning each task, averaged over epochs and layers, on a shared logarithmic scale. For Adam-NSCL, $P = I - Q$ comes from its recorded update transform, and no comparable residual is recorded. GPM and TRGP keep the residual near zero and SGP approaches that regime, CODE-CL protects nearly the whole trace and has high weighted overlap of its applied update with the protection operator, and SC-GPM retains a larger residual at lower protection. Three-seed means and sample SD over the tasks all seeds record.

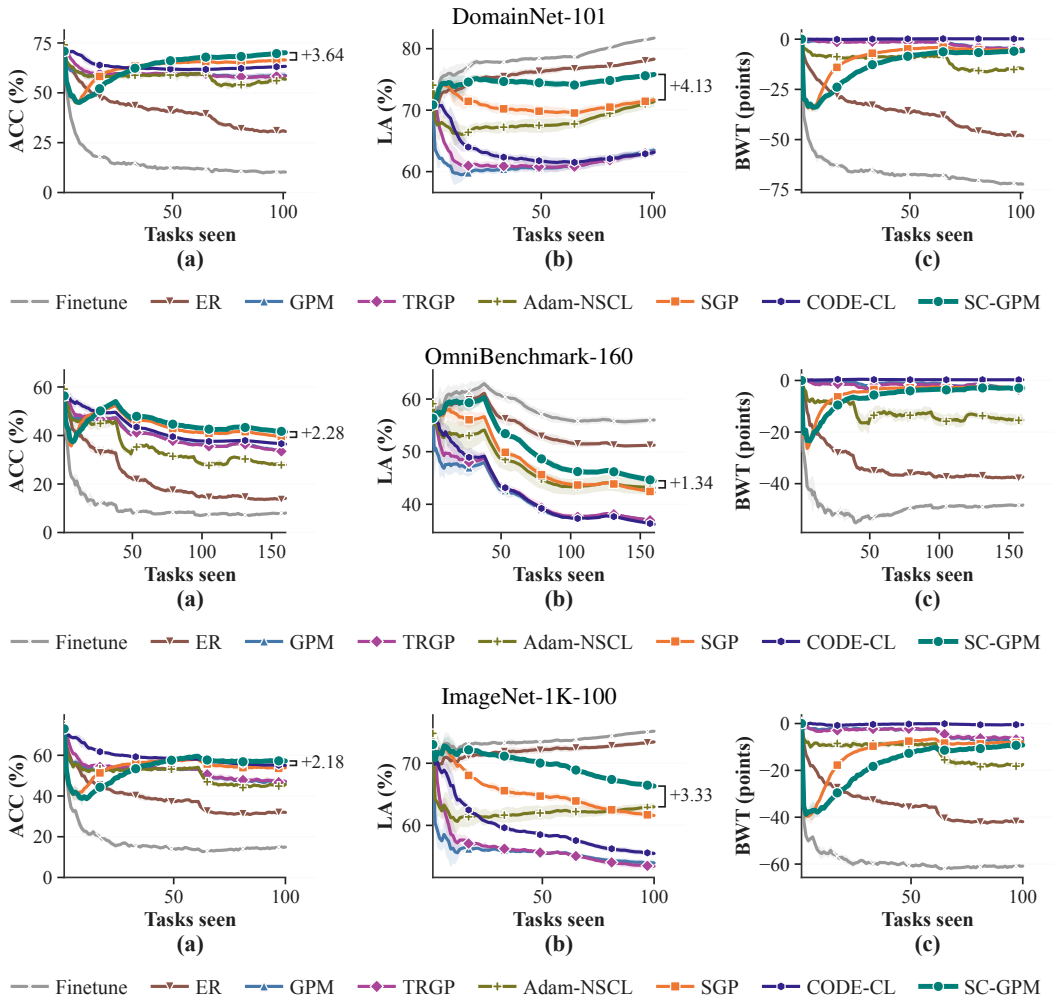


Figure 7: Image-stream trajectories at the configurations of Table 2. In each row, (a) average accuracy, (b) LA and (c) BWT after each task, as three-seed means and sample SD.

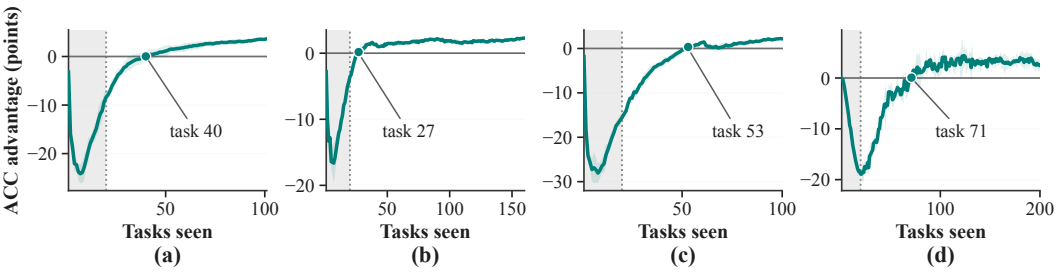


Figure 8: SC-GPM's accuracy advantage emerges beyond short horizons. Each panel shows SC-GPM's mean ACC minus the largest mean ACC among GPM, TRGP, SGP, Adam-NSCL and CODE-CL after each task, on (a) DomainNet-101, (b) OmniBenchmark-160, (c) ImageNet-1K-100 and (d) PMNIST at its 200-task configurations. Bands give the sample SD of same-seed differences, markers show the first task after which the advantage stays positive, and shading marks the first 20 tasks.

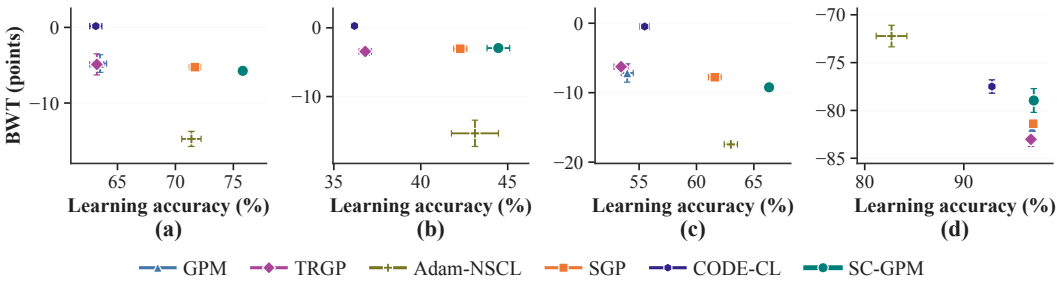


Figure 9: The balance between acquisition and retention on (a) DomainNet-101, (b) OmniBenchmark-160, (c) ImageNet-1K-100 and (d) PMNIST-200. On the image streams, SC-GPM has the highest learning accuracy and CODE-CL the highest BWT at lower learning accuracy. Three-seed means with sample-SD error bars.

Table 12: Average anytime accuracy. The image streams use the configurations of Table 2, and PMNIST uses each method's configuration selected for this metric at each horizon (Table 11). Bold marks the best mean among non-projection methods and among projection methods. Three-seed means and sample SD.

	Image streams			PMNIST			
Method	DomainNet	OmniBenchmark	ImageNet-1K	$T = 50$	$T = 100$	$T = 150$	$T = 200$
Finetune	15.15±0.21	10.24±0.14	17.34±0.23	42.32±0.60	31.59±0.91	26.52±0.82	23.49±0.61
ER	**41.18**±0.50	**22.02**±0.54	**38.82**±0.34	**93.10**±0.03	**88.80**±0.13	**84.51**±0.21	**80.54**±0.28
GPM	59.25±0.98	39.92±0.64	51.86±0.28	72.84±0.78	50.41±0.49	39.72±0.53	33.77±0.39
TRGP	59.70±1.02	40.03±0.30	52.61±0.53	58.95±2.54	37.88±1.45	29.38±0.96	24.96±0.73
Adam-NSCL	57.95±1.29	34.80±0.51	50.68±0.65	17.82±0.26	14.78±0.14	13.61±0.09	12.97±0.09
SGP	62.57±0.84	43.74±0.57	54.05±0.70	73.34±1.42	51.38±0.98	40.90±0.70	34.40±0.60
CODE-CL	**63.29**±0.69	42.35±0.35	**59.15**±0.06	76.40±1.63	56.81±0.68	45.14±0.66	37.89±0.51
SC-GPM	62.96±0.33	**45.03**±0.66	53.67±0.56	**77.05**±1.13	**57.17**±1.14	**45.91**±0.87	**39.13**±0.47

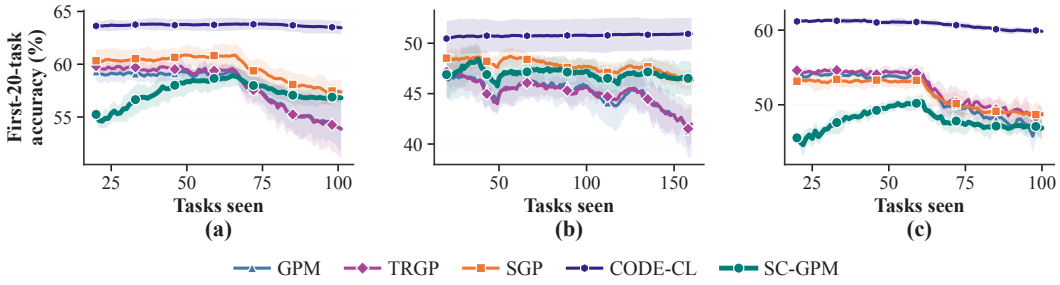


Figure 10: Accuracy on the first twenty tasks as later tasks are learned, on (a) DomainNet-101, (b) OmniBenchmark-160 and (c) ImageNet-1K-100. Three-seed means and sample SD.

F.3 Performance on Long PMNIST Sequences

Figure 11 compares all methods across PMNIST horizons, and Tables 13 and 14 give the values. SC-GPM has the highest final accuracy and the largest retained share of learning accuracy among projection methods at every horizon. The shared ten-way head changes with every new permutation, while projection constrains the hidden layers. Replay can rehearse stored examples through that head. At 200 tasks, Adam-NSCL has higher BWT than SC-GPM (-72.22 versus -78.96), but its acquisition is lower (82.70% versus 97.06% LA); their final average accuracies are 10.84% and 18.50%, respectively. At its selected 200-task configuration, CODE-CL's learning accuracy falls from 96.1% over the first 50 tasks to 90.4% over the last 50, while SC-GPM stays near 97% (Figure 12).

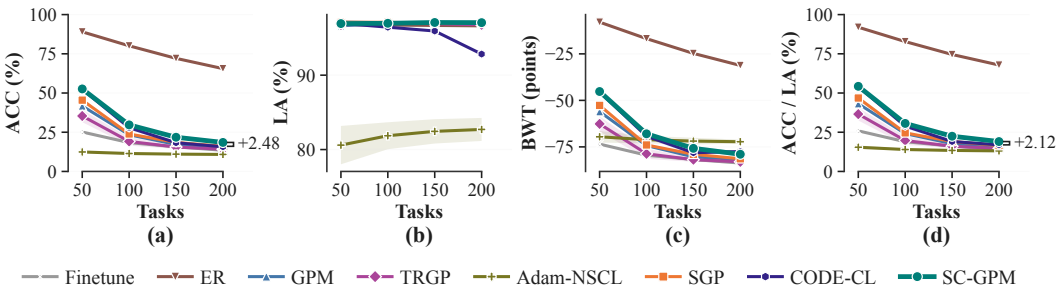


Figure 11: All methods across PMNIST horizons. (a) Final accuracy, (b) learning accuracy, (c) BWT and (d) the ratio of final to learning accuracy, each at the configuration selected for final accuracy at that horizon. Three-seed means and sample SD, with the ratio computed per seed.

Table 13: PMNIST final accuracy at each horizon, with each method at its configuration selected for final accuracy. Bold marks the best mean among non-projection methods and among projection methods. Three-seed means and sample SD.

Method	$T = 50$	$T = 100$	$T = 150$	$T = 200$
Finetune	25.14±0.21	18.38±0.78	15.66±0.86	13.51±0.65
ER	**89.01**±0.21	**80.09**±0.29	**72.10**±0.54	**65.56**±0.52
GPM	41.71±0.76	23.38±0.65	17.20±0.25	15.50±0.46
TRGP	35.36±3.94	19.13±2.72	15.58±1.16	14.17±0.79
Adam-NSCL	12.42±0.60	11.43±0.24	11.03±0.42	10.84±0.45
SGP	45.45±1.21	23.88±0.60	18.52±0.58	16.02±0.22
CODE-CL	52.39±1.09	27.89±1.21	18.31±1.03	15.73±0.69
SC-GPM	**52.61**±1.57	**29.68**±0.06	**21.80**±1.26	**18.50**±1.22

Table 14: PMNIST-200 results, with each method at its configuration selected for final accuracy. ACC/LA is the share of learning accuracy retained at the end. Bold marks the best mean among non-projection methods and among projection methods. Three-seed means and sample SD.

Method	ACC	LA	BWT	ACC/LA
Finetune	13.51±0.65	**96.79**±0.01	−83.70±0.67	13.96±0.67
ER	**65.56**±0.52	96.63±0.01	**−31.23**±0.54	**67.84**±0.55
GPM	15.50±0.46	96.90±0.02	−81.82±0.46	15.99±0.47
TRGP	14.17±0.79	96.78±0.05	−83.02±0.75	14.64±0.81
Adam-NSCL	10.84±0.45	82.70±1.55	**−72.22**±1.13	13.11±0.33
SGP	16.02±0.22	97.00±0.01	−81.39±0.23	16.51±0.23
CODE-CL	15.73±0.69	92.84±0.07	−77.50±0.70	16.94±0.74
SC-GPM	**18.50**±1.22	**97.06**±0.01	−78.96±1.25	**19.06**±1.26

Effect of the statistics sample size. The PMNIST protocol uses 2,000 training examples per task to estimate projection statistics. To test sensitivity to this choice, we vary $n \in \{64, 125, 250, 500, 1000, 2000, 4000\}$ on PMNIST-200 for GPM, SGP, TRGP, CODE-CL, and SC-GPM. Each method keeps its configuration selected for final accuracy at 200 tasks (Table 11); none is re-tuned. Statistics examples come from the task's training split, while every method trains on the full task. The first-layer MNIST input covariance has rank below 700. For $n \leq 500$, the estimated covariance has rank at most n; $n = 64$ is one training batch.

Figure 13 shows that SC-GPM remains the most accurate of these five methods at every n. Across the tested sample sizes of 64 to 4,000 examples, its final accuracy ranges from 17.49% to 18.73%, with a 1.08–2.81-point lead over the strongest other method. Learning accuracy stays at 97.1%, and the reserve fixes the same total capacity at every n; the sample changes the geometry of accumulated usage. Among the baselines, TRGP alone degrades with scarce statistics: from 14.17% at $n = 2000$

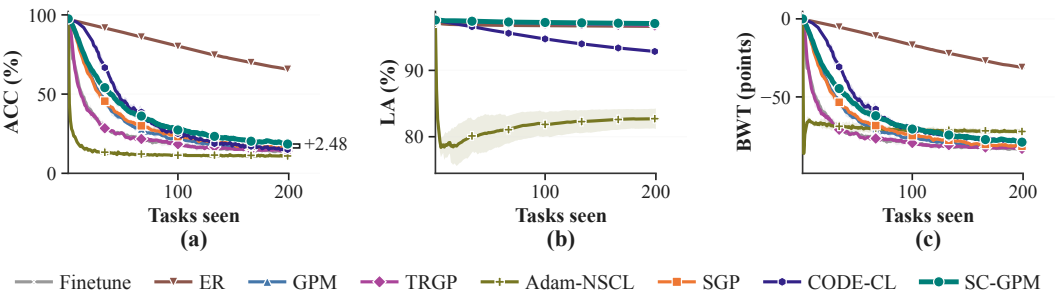


Figure 12: PMNIST trajectories at one fixed configuration per method, the one selected for final accuracy at 200 tasks. (a) average accuracy, (b) LA and (c) BWT after each task, as three-seed means and sample SD. The shorter horizons in Figure 2 use their own selected configurations.

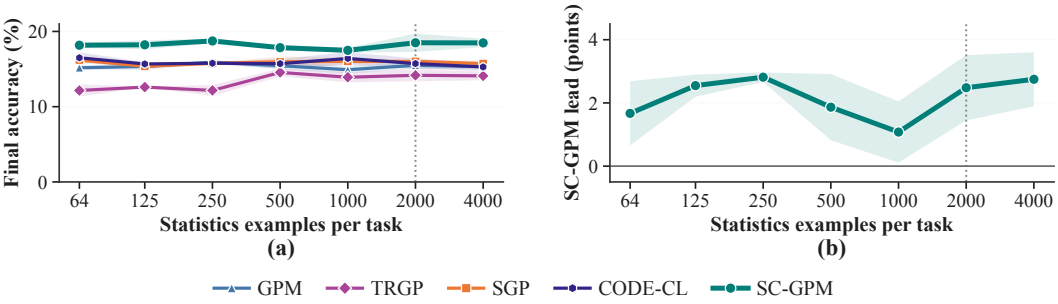


Figure 13: SC-GPM's lead on PMNIST-200 does not depend on the number of examples used for the projection statistics. (a) Final accuracy at 200 tasks as a function of the number n of examples per task, with each method at its configuration selected for final accuracy at 200 tasks. (b) SC-GPM's final accuracy minus that of the strongest other method, with the sample SD of same-seed differences. The dotted line marks the protocol's $n = 2000$, whose runs include those of the selection seed. Three-seed means and sample SD.

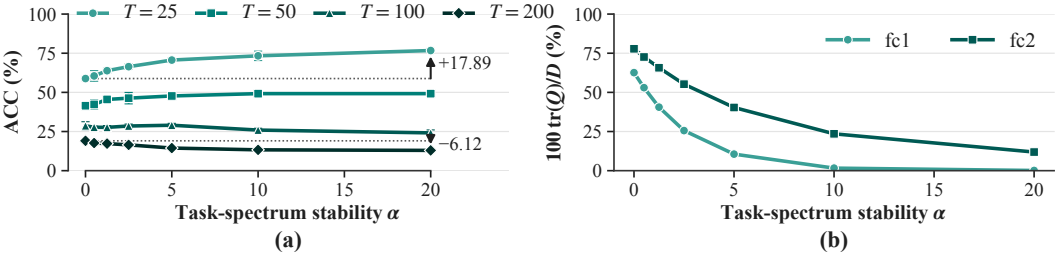


Figure 14: Task-spectrum stability controls the timing of protection. (a) Final accuracy at four horizons, read from the same 200-task runs, as a function of α. (b) Free capacity at the end of the stream in each hidden layer, normalized by its input dimension ($D = 784$ for fc1 and $D = 400$ for fc2). PMNIST with $\varepsilon = .84$ and $\kappa_l = 1$. Three-seed means and sample SD.

to 12.16–12.60% at $n \leq 250$. GPM, SGP, and CODE-CL stay within one point of their $n = 2000$ values.

Table 15: Endpoint metrics for the fixed PMNIST α sweep. Three-seed mean and sample SD.

α	ACC	AAA	BWT
0	19.07 ± 0.42	35.21 ± 0.60	-78.29 ± 0.44
0.5	17.65 ± 1.36	35.43 ± 0.78	-79.65 ± 1.38
1.25	17.29 ± 0.26	36.20 ± 0.50	-79.90 ± 0.26
2.5	16.47 ± 0.27	36.92 ± 0.91	-80.51 ± 0.27
5	14.44 ± 0.27	37.01 ± 0.48	-82.12 ± 0.27
10	13.30 ± 0.29	35.91 ± 0.43	-82.44 ± 0.28
20	12.95 ± 0.59	35.38 ± 0.50	-81.38 ± 0.68

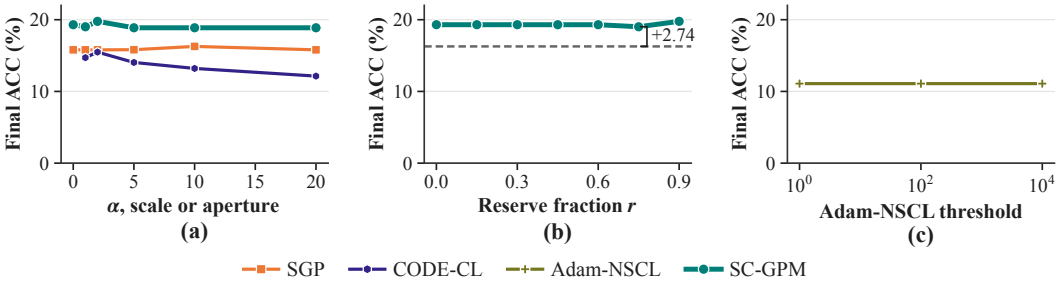


Figure 15: The remaining axes of the shared PMNIST grid at 200 tasks. (a) The best final accuracy at each value of SC-GPM's α, SGP's scale and CODE-CL's aperture, maximized over each method's other hyperparameters. (b) SC-GPM's best final accuracy at each reserve fraction, with a dashed line at the best configuration of any baseline. (c) Adam-NSCL's final accuracy across its threshold, which is identical at every value because its accumulated covariance is rank-deficient and each threshold selects the same null space. All values are from the selection seed.

F.4 Hyperparameter Sensitivity

Figure 14 and Table 15 report a fixed α sweep at $\varepsilon = .84$. Increasing α raises each retained direction's normalized task usage. Its preferred value changes with the horizon: among those tested, mean average accuracy is highest at $\alpha = 20, 10, 5, 0$ for $T = 25, 50, 100, 200$, respectively. Comparing $\alpha = 0$ and 20, the larger value raises average accuracy at task 25 from 58.83 ± 1.33 to $76.72 \pm 1.31\%$, but lowers final average accuracy from 19.07 ± 0.42 to $12.95 \pm 0.59\%$. Their mean average-accuracy curves cross permanently at task 83. Mean endpoint capacity in fc2 also falls from 311.30 to 47.57 out of 400. Thus the useful weighting depends on the horizon and criterion. The image setting $\alpha = 10$ and PMNIST's selected values at its longest horizons ($\alpha = 0$ at 150 tasks and $\alpha = 2$ at 200) represent different operating points. The reserve intervention in Figure 3 holds the task-spectrum construction fixed. Its engagement rate counts task boundaries where the unconstrained allocation would violate the specified capacity and the multiplier c in Theorem 3 becomes positive.