RETHINKING DIRECTIONAL CONSTRAINTS IN CONTINUAL LEARNING THROUGH SPECTRAL CAPACITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Subspace-based continual learning methods constrain update directions in seemingly contradictory ways: some isolate task updates in orthogonal subspaces to reduce interference, whereas others encourage subspace sharing to promote transfer. Despite these opposing designs, both families exhibit similar update-spectrum patterns that differ markedly from unconstrained fine-tuning. To isolate this spectral effect, we replace each method's directional constraint with a training-time constraint that matches the spectrum recorded from its updates while leaving update directions unconstrained. This spectrum-only intervention matches or even surpasses the original direction-constrained methods. We further find that precise matching of the full spectrum is unnecessary: controlling only update spectral capacity—measured by spectral energy and effective rank—through simple schedules of training factors such as LoRA rank already substantially improves continual learning. Building on this insight, we propose SPAR (Spectral Probing for Adaptive Rank), which uses a few probing updates to estimate each task's demand for update spectral capacity and allocates its LoRA rank accordingly. Across diverse benchmarks, this simple scheduler achieves competitive or superior performance without elaborate directional constraints or auxiliary mechanisms.

1 INTRODUCTION

Continual learning (CL) asks a model to acquire a sequence of tasks without losing knowledge required by earlier tasks. The central difficulty is the stability–plasticity dilemma: an update must be expressive enough to learn the current task, yet restrained enough to preserve previous behavior. Subspace constraints have become a dominant way to negotiate this trade-off, but their prescriptions conflict. Separation-oriented methods project gradients or reserve nearly orthogonal task subspaces to reduce interference (Farajtabar et al., 2020; Saha et al., 2021; Li et al., 2026). Sharing-oriented methods instead reuse and optimize shared parameter directions or adapters to promote cross-task transfer and enhance plasticity (He et al., 2025; Kaushik et al., 2026; He et al., 2026). Despite their seemingly opposing designs, both families include methods that achieve state-of-the-art performance. This raises a fundamental question: what common mechanism underlies their success?

To understand why these seemingly opposing approaches succeed, we compare their parameter updates with those of unconstrained sequential fine-tuning through the SVD $\Delta W_t = U_t \Sigma_t V_t^\top$, where the singular vectors specify update directions and the singular values define the update spectrum. Both separation- and sharing-oriented methods produce updates with lower spectral energy and more concentrated spectra, and these differences often become more pronounced along the task sequence. To isolate this spectral effect from the prescribed directions, we record each method's task-wise spectra and use them as targets to constrain otherwise unconstrained fine-tuning while leaving the singular vectors free. This spectrum-replay intervention recovers or even surpasses the performance of the original methods (Table 1), suggesting that suitable spectral-capacity control allows optimization to learn effective update directions without explicit directional constraints.

Although spectrum replay demonstrates the effectiveness of a suitable spectral constraint, its target spectra must be supplied by an existing continual-learning method. We therefore ask whether spectral capacity can instead be controlled directly through key optimization factors. Guided by the observed task-wise spectral trends, we design simple decay schedules for learning rate, optimization steps, and LoRA rank. All three effectively reshape the update spectra and substantially improve

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.

continual performance over unconstrained sequential fine-tuning, establishing spectral-capacity control as a practical design principle. Building on these results, we propose SPAR (Spectral Probing for Adaptive Rank), a simple adaptive rank scheduler that estimates task demand from the spectra induced by brief probing updates and assigns LoRA rank accordingly. We analyze when capacity decay is justified and evaluate the effectiveness of this demand-aware scheduler empirically.

Our main contributions are:

- We disentangle direction from spectrum in representative subspace-based CL methods and show that spectrum matching alone recovers or surpasses their performance, indicating that update-spectrum control can serve as an effective alternative to directional constraints.
- We show that simple schedules of learning rate, optimization steps, or LoRA rank can effectively control spectral capacity and improve continual performance, without precisely matching the full spectrum or introducing elaborate directional constraints.
- Building on this principle, we introduce SPAR, which estimates task demand from brief probe-update spectra and adaptively allocates LoRA rank. We provide theoretical motivation and demonstrate competitive or superior performance across diverse benchmarks.

2 Related Work

Replay and reference-data regularization. Replay methods preserve previous knowledge by revisiting stored examples, features, or predictions. iCaRL maintains exemplar sets for old classes (Rebuffi et al., 2017); GEM and A-GEM use episodic memories to constrain current gradients (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019); Experience Replay mixes buffered experience with the current stream (Rolnick et al., 2019); and Dark Experience Replay distills logits stored along the training trajectory (Buzzega et al., 2020). For pre-trained vision-language models, ZSCL instead anchors the representation to the initial model through distillation on an auxiliary reference dataset (Zheng et al., 2023). Although effective, these approaches require storing past data or sourcing representative external data, while adding memory, data-management, and training costs. Their dependence on accessible and well-matched reference data can further limit practical deployment.

Subspace and directional constraints. Replay-free methods instead regulate the geometry of parameter updates. OGD and GPM project gradients away from subspaces important to previous tasks (Farajtabar et al., 2020; Saha et al., 2021), while O-LoRA and InfLoRA isolate low-rank updates through orthogonal subspaces (Wang et al., 2023; Liang & Li, 2024). BiLoRA, KeepLoRA, and E^2-LoRA further organize updates through almost-orthogonal, residual, or energy-ordered subspaces (Zhu et al., 2025; Luo et al., 2026; Li et al., 2026). An apparently opposing line promotes reuse: SD-LoRA encourages convergence toward overlapping low-loss regions (Wu et al., 2025), Share maintains an evolving shared LoRA subspace (Kaushik et al., 2026), and LoDA decomposes the update space to balance task-specific isolation with cross-task sharing (He et al., 2026). These methods prescribe different directional geometries, whereas our work investigates spectral-capacity control as an alternative and potential complement to explicit directional constraints.

3 From Direction Geometry to Spectral Capacity

3.1 Task Updates and Spectral Capacity

Let a pretrained model contain matrices $W_0^{(\ell)} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ for layers $\ell = 1, \ldots, L$. Tasks arrive sequentially as datasets $\mathcal{D}_1, \ldots, \mathcal{D}_T$. We define the incremental update at task t as

$$\Delta W_t^{(\ell)} = W_t^{(\ell)} - W_{t-1}^{(\ell)}. \tag{1}$$

This definition isolates the parameters consumed by the current task from the cumulative displacement $W_t - W_0$. For LoRA, $\Delta W_t^{(\ell)} = B_t^{(\ell)} A_t^{(\ell)}$. We decompose each task update as

$$\Delta W_t^{(\ell)} = U_t^{(\ell)} \Sigma_t^{(\ell)} (V_t^{(\ell)})^\top. \tag{2}$$

The singular vectors specify the update subspace, while the diagonal spectrum $\Sigma_t^{(\ell)}$ specifies how much capacity is placed along its spectral directions. We use *update spectral capacity* specifically

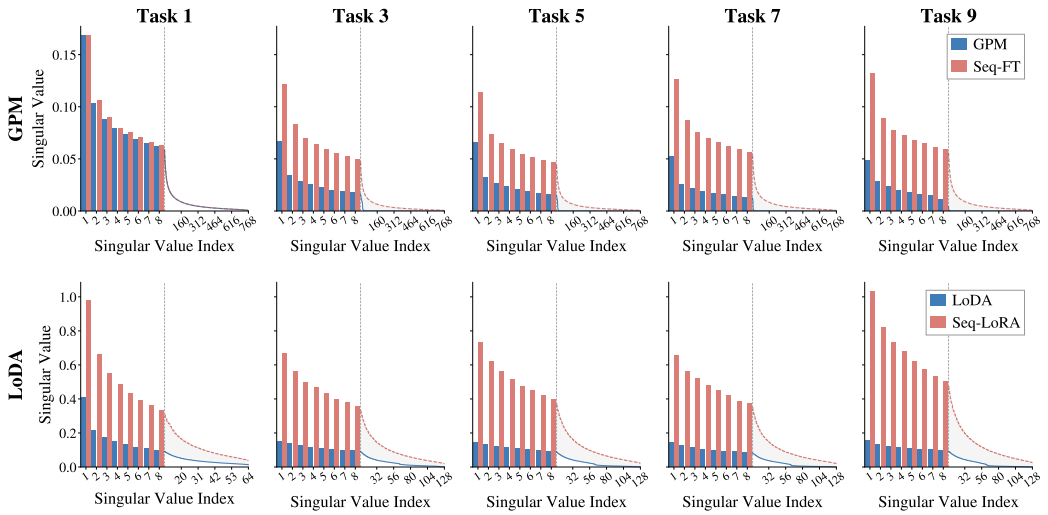


Figure 1: Task-wise singular-value spectra of updates from GPM, LoDA, and sequential fine-tuning, representing separation-oriented, sharing-oriented, and unconstrained updates, respectively.

for the magnitude and effective dimensionality of this task-update spectrum. We characterize update spectral capacity with two complementary quantities: spectral energy, measured by the Frobenius norm, and effective rank, which measures the number of effective update directions.

For singular values $\{\sigma_{t,i}^{(\ell)}\}$, spectral energy is measured by the Frobenius norm:

$$\mathcal{F}_t^{(\ell)} = \|\Delta W_t^{(\ell)}\|_F = \left[\sum_i \left(\sigma_{t,i}^{(\ell)}\right)^2\right]^{1/2}. \tag{3}$$

We normalize squared singular values as

$$p_{t,i}^{(\ell)} = \frac{(\sigma_{t,i}^{(\ell)})^2}{\sum_j (\sigma_{t,j}^{(\ell)})^2}, \tag{4}$$

Following the entropy-based construction of Roy & Vetterli (2007), we define spectral entropy and the corresponding entropy-based effective rank using normalized spectral energies:

$$H_t^{(\ell)} = -\sum_i p_{t,i}^{(\ell)} \log p_{t,i}^{(\ell)}, \qquad r_{\text{eff},t}^{(\ell)} = \exp(H_t^{(\ell)}). \tag{5}$$

Together, the Frobenius norm and effective rank distinguish how far an update moves from how many effective directions it uses. We use them to compare the spectral capacity of direction-constrained methods with that of unconstrained sequential fine-tuning. This provides a common lens for analyzing the seemingly opposing separation- and sharing-oriented methods below.

3.2 Spectrum Matching without Prescribed Directions

We compare the task-wise update spectra of direction-constrained learning methods with unconstrained sequential fine-tuning. We focus on GPM, which restricts updates to directions orthogonal to previous-task subspaces (Saha et al., 2021), and LoDA, which also optimizes shared directions to balance stability and plasticity (He et al., 2026). Additional methods are analyzed in Appendix Table 6. For each method, we evaluate three variants and record their update spectra and performance:

1. **Directional method:** the original method with its prescribed directional constraints.
2. **Directional constraints removed:** sequential fine-tuning without directional constraints.
3. **Directional constraints removed + spectrum matched:** sequential fine-tuning with freely learned directions, but with update singular values matched to those recorded from the original direction-constrained learning method for the corresponding task and layer.

Table 1: Last and incremental accuracy for directional, unconstrained, and spectrum-only variants of GPM and LoDA.

Method	Last-Acc	Inc-Acc
GPM (Saha et al., 2021)	73.95	78.72
– direction (Seq-FT)	$67.23_{(-6.72)}$	$74.76_{(-3.96)}$
– direction + spectrum	$73.80_{(-0.15)}$	$79.18_{(+0.46)}$
LoDA (He et al., 2026)	81.88	86.99
– direction (Seq-LoRA)	$67.37_{(-14.51)}$	$79.40_{(-7.59)}$
– direction + spectrum	$81.55_{(-0.33)}$	$86.90_{(-0.09)}$

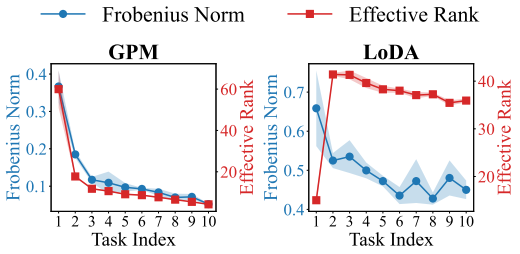


Figure 2: Task-wise Frobenius norm and effective rank for GPM and LoDA.

Figure 1 compares the spectra of the first two variants. Despite their different directional designs, GPM and LoDA both yield lower-energy, more concentrated spectra than unconstrained fine-tuning. For the third variant, we apply spectral correction several times during each task's training. At each correction, we compute the SVD of the current parameter update and replace its singular values with the target spectrum Σ_t^{dir} recorded from the original method for the same task and layer. Retaining the current singular vectors $U_t^{\text{free}}, V_t^{\text{free}}$, we reconstruct the update as

$$\Delta W_t^{\text{spec}} = U_t^{\text{free}} \Sigma_t^{\text{dir}} (V_t^{\text{free}})^\top. \tag{6}$$

We then resume training with the corrected update. Each correction matches the update spectrum to that of the direction-constrained method while leaving the singular vectors unconstrained.

Table 1 shows that removing directional constraints substantially degrades performance for both methods. Restoring their update spectra recovers comparable performance, suggesting that suitable spectral-capacity control allows optimization to learn effective continual-learning directions without the original directional constraints. Appendix A.2 discusses the scope of this inference.

Implicit spectral-capacity decay. Figure 2 shows that both $\mathcal{F}_t$ and $r_{\text{eff},t}$ tend to decrease along the task sequence. To interpret this trend, let c_t denote a scalar capacity proxy, $g_t(c_t)$ the current-task benefit, and $h_{k,t}(c_t)$ the degradation on previous task k. Under an additive, separable surrogate for final performance, maximizing the equally weighted average over all tasks reduces to maximizing

$$J_t(c_t) = g_t(c_t) - H_t(c_t), \qquad H_t(c) = \sum_{k<t} h_{k,t}(c). \tag{7}$$

The functions $h_{k,t}$ allow different tasks to suffer different amounts of interference. If tasks have identical capacity–benefit curves $g_t = g$ and the additional interference cost $H_{t+1}(c) - H_t(c)$ is non-decreasing in c, there exists a non-increasing optimal capacity allocation. Balanced class-incremental splits motivate approximately homogeneous demands as a reference case, although equal class counts do not imply identical benefit curves. These are sufficient conditions for non-increasing capacity, not necessary ones; a harder later task can instead benefit from greater capacity. Appendix B.1 gives the assumptions, proof, and counterexamples. This sufficient-condition analysis motivates capacity decay as a reference policy and task demand as a reason to depart from it.

3.3 Controlling Spectral Capacity

Spectrum replay establishes the effectiveness of spectral-capacity control, but requires target spectra from an existing continual-learning method. We next ask whether standard training factors can provide such control directly, without reference spectra or prescribed update directions. We examine three factors: learning rate, optimization steps, and LoRA rank.

Learning rate and optimization steps regulate the extent of adaptation, while LoRA rank directly bounds update dimensionality. Guided by the observed spectral-capacity decay, we apply task-wise power-law schedules $x_t = x_1/t^\gamma$, where x_t denotes the selected factor and t starts at one. We vary one factor at a time and compare three decay exponents against sequential fine-tuning with constant settings, measuring both the resulting spectral capacity and final average accuracy.

Figure 3 shows that all three scheduling mechanisms reshape the spectral-capacity trajectories and improve continual-learning performance over unconstrained sequential fine-tuning. Appendix A.3

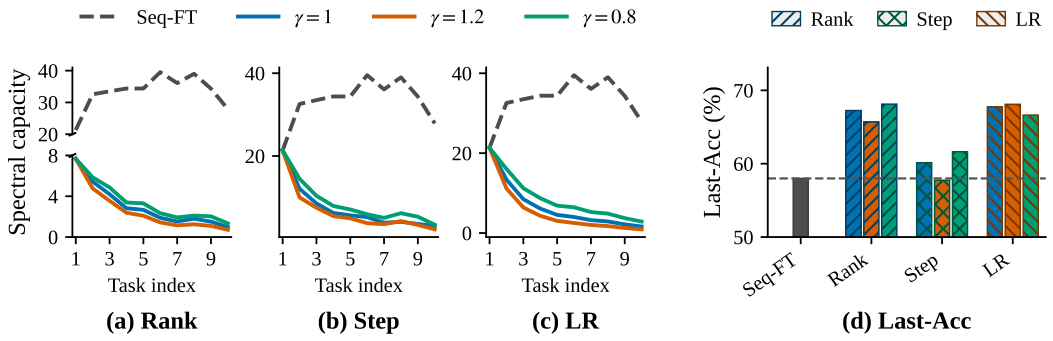


Figure 3: Controlling spectral capacity through training-factor schedules. Panels (a)–(c) show the task-wise spectral-capacity proxy $\mathcal{F}_t r_{\text{eff},t}$ when scheduling LoRA rank, optimization steps, or learning rate. Each compares three power-law decay exponents with constant-setting sequential fine-tuning (Seq-FT). Panel (d) reports final average accuracy (Last-Acc) for all configurations.

reports the corresponding Frobenius norms and effective ranks separately. These results establish training-factor scheduling as a practical way to control spectral capacity without replaying a reference method's spectra. We focus on rank because it directly limits update dimensionality while leaving directions to optimization. Fixed decay schedules, however, depend only on task position and implicitly assume comparable adaptation demands. When tasks require different update dimensionalities or spectral capacities, their appropriate allocations need not follow a common decay trajectory (Figure 9). This motivates estimating current-task demand and adapting rank accordingly.

4 SPAR: Spectral Probing for Adaptive Rank

SPAR addresses this heterogeneity by estimating task demand from brief probing updates and using their Frobenius norm and effective rank to determine the LoRA rank for subsequent adaptation.

4.1 Obtaining Probe Updates

We use one epoch by default to probe task demand: on near-balanced splits, similar task sizes give comparable update counts at a fixed batch size. When task t arrives, we save W_{t-1} and directly fine-tune the designated matrices. For layer ℓ, the probe update is

$$\Delta W_{t,\text{probe}}^{(\ell)} = W_{t,\text{probe}}^{(\ell)} - W_{t-1}^{(\ell)}. \tag{8}$$

We compute its Frobenius norm $\mathcal{F}_t^{(\ell)}$ and effective rank $r_{\text{eff},t}^{(\ell)}$ using Equations 3–5. The probe is then discarded and W_{t-1} is restored before the actual task training.

4.2 Spectral Capacity as a Demand Score

We use the probe's spectral capacity as a proxy for adaptation demand, combining update magnitude and effective dimensionality into a layer-wise score:

$$s_t^{(\ell)} = \mathcal{F}_t^{(\ell)} r_{\text{eff},t}^{(\ell)}. \tag{9}$$

The score guides allocation for continual performance rather than specifying a minimum rank for reconstructing the probe. Multiplication gives equal proportional sensitivity to magnitude and effective dimensionality without additional exponents to tune. This is a modeling choice, not a uniquely optimal measure of demand. Appendix B.2 distinguishes this allocation signal from reconstruction criteria; Table 5 evaluates its utility empirically, and Appendix A.6 compares the resulting allocations with accuracy-guided rank search. Scores are compared across tasks within each layer.

Algorithm 1 SPAR: Spectral Probing for Adaptive Rank

Input: Pretrained parameters W_0; task sequence $\mathcal{D}_1, \ldots, \mathcal{D}_T$
Output: Final adapted parameters W_T

```
Set cumulative scores S^(ℓ) ← 0 for all layers
for t = 1, ..., T do
    Probe W_{t-1} on D_t for one epoch to obtain W_{t,probe}
    for ℓ = 1, ..., L do
        Compute spectral demand s_t^(ℓ) from W_{t,probe}^(ℓ) − W_{t-1}^(ℓ) using Equation 9
        S^(ℓ) ← S^(ℓ) + s_t^(ℓ);   ρ_t^(ℓ) ← s_t^(ℓ) / S^(ℓ)
        Allocate rank r_t^(ℓ) using Equation 10
    end for
    Restore W_{t-1}; initialize and train LoRA with ranks {r_t^(ℓ)}_{ℓ=1}^L on D_t
    Merge W_t^(ℓ) ← W_{t-1}^(ℓ) + B_t^(ℓ) A_t^(ℓ) for all adapted layers
end for
```

4.3 Demand-Aware Rank Allocation

Let $r_{\max}^{(\ell)}$ be the rank cap for each task in layer ℓ, set to $d_{\text{out}}^{(\ell)}$ by default, with $1 \le r_{\min} \le r_{\max}^{(\ell)}$. It is not a total budget shared across tasks or layers. We normalize the current demand by the cumulative demand observed in the same layer and map this share to the rank interval $[r_{\min}, r_{\max}^{(\ell)}]$:

$$\rho_t^{(\ell)} = \frac{s_t^{(\ell)}}{\sum_{i=1}^{t} s_i^{(\ell)}}, \qquad r_t^{(\ell)} = \left\lfloor r_{\min} + (r_{\max}^{(\ell)} - r_{\min})\rho_t^{(\ell)} \right\rceil. \tag{10}$$

Here $\lfloor \cdot \rceil$ denotes nearest-integer rounding. Let $\bar{s}_t^{(\ell)} = t^{-1} \sum_{i=1}^{t} s_i^{(\ell)}$ be the running mean. Then

$$\rho_t^{(\ell)} = \frac{1}{t} \frac{s_t^{(\ell)}}{\bar{s}_t^{(\ell)}}. \tag{11}$$

Thus SPAR combines inverse-task decay with a relative-demand correction: equal scores recover $1/t$, while above-average demand increases allocation and sufficiently large shifts can reverse the decay. The floor $r_{\min}$ preserves a minimum adaptation capacity, and LoRA rank directly controls update dimensionality. Algorithm 1 summarizes the procedure, which retains only one cumulative scalar per layer and no historical singular vectors. Appendix B.3 provides complementary interpretations through a streaming weighted mean and an online quadratic surrogate.

5 Experiments

5.1 Experimental Setup

Benchmarks and backbones. We evaluate class-incremental learning (CIL) on ImageNet-R (Hendrycks et al., 2021), CIFAR-100 (Krizhevsky et al., 2009), CUB-200 (Wah et al., 2011), and Cars-196 (Krause et al., 2013), and domain-incremental learning (DIL) on Office-Home (Venkateswara et al., 2017) and DomainNet (Peng et al., 2019). CIL introduces new classes sequentially, whereas DIL introduces new domains with a shared label space. We use ViT-B/16-IN21K for CIL and ViT-B/16-IN1K for DIL. To examine heterogeneous task demands, we additionally construct imbalanced class-count sequences on Cars-196 and ImageNet-R, detailed in Tables 4 and 5. Additional results on six GLUE (Wang et al., 2018) tasks appear in Appendix A.5.

Baselines and implementation. We compare against prompt-based, representation-based, and LoRA-based baselines, separating methods with and without replay. Unless otherwise specified, we set $r_{\min} = 8$ and $r_{\max}^{(\ell)} = d_{\text{out}}^{(\ell)}$. SPAR uses a one-epoch probe to allocate LoRA ranks, then trains LoRA and a bias-free cosine classifier using cross-entropy (CE) loss. The learned LoRA are then merged into the backbone. No historical update directions are stored or explicitly constrained.

Evaluation metrics. Let A_t denote accuracy over all tasks seen after learning task t. We report final accuracy, Last-Acc $= A_T$, and average incremental accuracy, Inc-Acc $= T^{-1} \sum_{t=1}^{T} A_t$, both in percent. The main benchmark tables report means and standard deviations.

Table 2: Class-incremental learning results. "Replay" denotes feature replay; "CA" denotes classifier alignment. Best and second-best results are highlighted within each replay group.

Method	Replay	ImageNet-R	CIFAR-100	CUB-200	Cars-196
Last-Acc					
L2P (Wang et al., 2022b)	✗	$66.49_{\pm 0.40}$	$82.76_{\pm 1.17}$	$62.21_{\pm 1.92}$	$38.18_{\pm 2.33}$
DualPrompt (Wang et al., 2022a)	✗	$68.50_{\pm 0.52}$	$85.56_{\pm 0.33}$	$66.00_{\pm 0.57}$	$40.14_{\pm 2.36}$
EASE (Zhou et al., 2024)	✗	$77.75_{\pm 0.18}$	$86.54_{\pm 0.29}$	$79.90_{\pm 0.29}$	$35.46_{\pm 1.50}$
SD-LoRA (Wu et al., 2025)	✗	$77.34_{\pm 0.35}$	$88.01_{\pm 0.31}$	$77.48_{\pm 0.20}$	–
BiLoRA (Zhu et al., 2025)	✗	$77.95_{\pm 0.14}$	$87.46_{\pm 0.76}$	–	–
LoDA (He et al., 2026)	✗	$\mathbf{81.93}_{\pm 0.20}$	$\mathbf{90.47}_{\pm 0.06}$	$\underline{81.74}_{\pm 0.78}$	$\underline{66.94}_{\pm 0.90}$
SPAR (Ours)	✗	$\underline{81.66}_{\pm 0.15}$	$\underline{90.05}_{\pm 0.68}$	$\mathbf{82.53}_{\pm 0.50}$	$\mathbf{67.84}_{\pm 0.28}$
SLCA (Zhang et al., 2023a)	✓	$77.00_{\pm 0.33}$	$91.53_{\pm 0.28}$	$84.71_{\pm 0.40}$	$67.73_{\pm 0.85}$
SSIAT (Tan et al., 2024)	✓	$79.38_{\pm 0.59}$	$91.35_{\pm 0.26}$	$88.75_{\pm 0.38}$	$71.02_{\pm 0.69}$
MOS (Sun et al., 2025)	✓	$78.10_{\pm 0.12}$	$90.53_{\pm 0.35}$	$89.91_{\pm 0.25}$	$67.76_{\pm 1.28}$
TUNA (Wang et al., 2025)	✓	$79.44_{\pm 0.38}$	$91.79_{\pm 0.15}$	$88.40_{\pm 0.42}$	$69.46_{\pm 0.35}$
LoDA (He et al., 2026) + CA	✓	$82.55_{\pm 0.55}$	$\mathbf{92.15}_{\pm 0.21}$	$\mathbf{90.67}_{\pm 0.17}$	$\underline{76.52}_{\pm 0.34}$
E^2-LoRA (Li et al., 2026)	✓	$\underline{82.77}_{\pm 0.10}$	$\underline{92.13}_{\pm 0.19}$	$89.77_{\pm 0.16}$	$75.82_{\pm 0.28}$
SPAR (Ours) + CA	✓	$\mathbf{83.10}_{\pm 0.06}$	$\underline{92.13}_{\pm 0.15}$	$\underline{90.34}_{\pm 0.25}$	$\mathbf{78.61}_{\pm 0.44}$
Inc-Acc					
L2P (Wang et al., 2022b)	✗	$72.83_{\pm 0.56}$	$88.48_{\pm 0.83}$	$73.83_{\pm 1.67}$	$51.79_{\pm 4.19}$
DualPrompt (Wang et al., 2022a)	✗	$72.59_{\pm 0.24}$	$90.33_{\pm 0.33}$	$77.92_{\pm 0.50}$	$56.74_{\pm 1.78}$
EASE (Zhou et al., 2024)	✗	$83.87_{\pm 0.78}$	$91.68_{\pm 0.18}$	$86.65_{\pm 0.63}$	$48.96_{\pm 0.96}$
SD-LoRA (Wu et al., 2025)	✗	$82.04_{\pm 0.24}$	$92.54_{\pm 0.18}$	$85.59_{\pm 0.44}$	–
BiLoRA (Zhu et al., 2025)	✗	$81.52_{\pm 0.26}$	$92.50_{\pm 0.62}$	–	–
LoDA (He et al., 2026)	✗	$\mathbf{86.90}_{\pm 0.40}$	$\underline{93.46}_{\pm 1.42}$	$\mathbf{89.35}_{\pm 0.98}$	$\underline{76.04}_{\pm 0.79}$
SPAR (Ours)	✗	$\underline{86.62}_{\pm 0.44}$	$\mathbf{93.89}_{\pm 0.33}$	$\underline{88.65}_{\pm 0.76}$	$\mathbf{76.90}_{\pm 0.65}$
SLCA (Zhang et al., 2023a)	✓	$81.17_{\pm 0.64}$	$94.09_{\pm 0.87}$	$90.94_{\pm 0.68}$	$76.93_{\pm 1.21}$
SSIAT (Tan et al., 2024)	✓	$83.63_{\pm 0.43}$	$94.35_{\pm 0.60}$	$93.00_{\pm 0.90}$	$76.93_{\pm 0.62}$
MOS (Sun et al., 2025)	✓	$83.61_{\pm 0.40}$	$94.09_{\pm 0.91}$	$92.87_{\pm 0.56}$	$74.38_{\pm 1.13}$
TUNA (Wang et al., 2025)	✓	$84.80_{\pm 0.37}$	$94.88_{\pm 0.06}$	$92.01_{\pm 0.68}$	$76.95_{\pm 0.39}$
LoDA (He et al., 2026) + CA	✓	$\underline{87.18}_{\pm 0.42}$	$94.70_{\pm 1.16}$	$\mathbf{93.97}_{\pm 0.56}$	$\underline{82.80}_{\pm 0.23}$
E^2-LoRA (Li et al., 2026)	✓	$\underline{87.18}_{\pm 0.12}$	$\underline{95.01}_{\pm 0.02}$	$92.68_{\pm 0.48}$	$80.90_{\pm 0.73}$
SPAR (Ours) + CA	✓	$\mathbf{87.73}_{\pm 0.23}$	$\mathbf{95.03}_{\pm 0.29}$	$\underline{93.18}_{\pm 0.55}$	$\mathbf{84.16}_{\pm 0.13}$

5.2 Main Results

Class-incremental learning. Table 2 compares class-incremental learning performance on four benchmarks, with and without feature replay for classifier alignment (CA). Despite using only a brief probe and rank scheduling, SPAR matches or exceeds state-of-the-art methods such as LoDA and E^2-LoRA under the corresponding replay settings. It therefore provides a strong simple baseline for class-incremental learning without explicit direction storage, search, or constraints.

Domain-incremental learning. In Table 3, replay-free SPAR outperforms all listed baselines on both metrics. Its Last-Acc reaches 88.59% on Office-Home and 70.80% on DomainNet, exceeding the strongest replay-free baseline, DUCT, by 3.17 and 3.79 points. It also surpasses E^2-LoRA, which uses replay. Adding CA further improves both metrics on both datasets. Thus, adaptive spectral capacity remains effective when the distribution shifts across domains rather than classes.

Imbalanced task complexity. Table 4 alternates large and small tasks on Cars-196, testing adaptation to varying class counts rather than task position alone. SPAR achieves the best Last-Acc and Inc-Acc at all three imbalance ratios. Its Last-Acc advantage over LoDA widens from 1.54 points at $3{:}1$ to 5.55 at $6{:}1$ and 8.26 at $13{:}1$. Fixed-rank LoRA performs substantially worse across settings. These comparisons support demand-aware allocation under heterogeneous tasks.

Table 3: Domain-incremental learning results. "Replay" denotes exemplar or feature replay; "CA" denotes classifier alignment. Best and second-best results are highlighted within each replay group.

Method	Replay	Office-Home		DomainNet	
		Last-Acc	Inc-Acc	Last-Acc	Inc-Acc
L2P (Wang et al., 2022b)	✗	$80.03_{\pm 1.29}$	$79.72_{\pm 4.19}$	$48.72_{\pm 2.83}$	$50.45_{\pm 4.10}$
DualPrompt (Wang et al., 2022a)	✗	$80.85_{\pm 0.14}$	$80.20_{\pm 3.81}$	$50.46_{\pm 3.17}$	$52.28_{\pm 3.35}$
CODA-Prompt (Smith et al., 2023)	✗	$85.07_{\pm 0.34}$	$\underline{84.70}_{\pm 2.94}$	$59.99_{\pm 0.88}$	$59.85_{\pm 4.49}$
RanPAC (McDonnell et al., 2023)	✗	$82.28_{\pm 0.07}$	$82.30_{\pm 3.34}$	$54.80_{\pm 0.36}$	$55.20_{\pm 3.93}$
EASE (Zhou et al., 2024)	✗	$76.33_{\pm 2.16}$	$81.16_{\pm 3.52}$	$43.72_{\pm 1.70}$	$50.50_{\pm 2.27}$
SimpleCIL (Zhou et al., 2025a)	✗	$75.72_{\pm 0.00}$	$75.69_{\pm 5.03}$	$44.08_{\pm 0.00}$	$42.95_{\pm 4.84}$
DUCT (Zhou et al., 2025b)	✗	$\underline{85.42}_{\pm 0.33}$	$81.28_{\pm 0.31}$	$\underline{67.01}_{\pm 1.35}$	$\underline{67.16}_{\pm 3.75}$
SPAR (Ours)	✗	$\mathbf{88.59}_{\pm 0.29}$	$\mathbf{86.18}_{\pm 0.19}$	$\mathbf{70.80}_{\pm 0.15}$	$\mathbf{70.18}_{\pm 0.09}$
MEMO (Zhou et al., 2023)	✓	$63.09_{\pm 1.80}$	$71.18_{\pm 2.76}$	$58.41_{\pm 3.20}$	$61.92_{\pm 5.39}$
DCE (Li et al., 2025)	✓	$84.40_{\pm 0.20}$	$84.60_{\pm 3.00}$	$63.50_{\pm 0.50}$	$64.30_{\pm 6.00}$
E^2-LoRA (Li et al., 2026)	✓	$\underline{88.25}_{\pm 0.25}$	$\underline{84.94}_{\pm 0.21}$	$\underline{69.63}_{\pm 0.05}$	$\underline{68.69}_{\pm 0.04}$
SPAR (Ours) + CA	✓	$\mathbf{89.25}_{\pm 0.14}$	$\mathbf{86.70}_{\pm 0.08}$	$\mathbf{70.88}_{\pm 0.04}$	$\mathbf{70.41}_{\pm 0.04}$

Table 4: Performance under imbalanced task complexity on Cars-196. Each sequence alternates between large and small tasks for 14 tasks. The three settings use $[21, 7]\times 7$, $[24, 4]\times 7$, and $[26, 2]\times 7$ classes per task, corresponding to large-to-small task ratios of 3:1, 6:1, and 13:1, respectively.

Method	3:1 ($[21, 7] \times 7$)		6:1 ($[24, 4] \times 7$)		13:1 ($[26, 2] \times 7$)	
	Last-Acc	Inc-Acc	Last-Acc	Inc-Acc	Last-Acc	Inc-Acc
Seq-FT	39.96	61.34	35.63	55.29	30.97	63.63
Seq-LoRA ($r = 32$) (Hu et al., 2022)	42.20	54.93	33.89	43.83	33.96	43.56
Seq-LoRA ($r = 64$) (Hu et al., 2022)	43.32	56.46	35.77	42.35	43.48	53.51
Seq-LoRA ($r = 128$) (Hu et al., 2022)	44.43	57.78	31.24	43.10	40.35	59.95
LoDA (He et al., 2026) + CA	75.61	82.34	72.94	80.25	71.80	80.35
SPAR (Ours) + CA	**77.15**	**83.72**	**78.49**	**84.16**	**80.06**	**86.01**

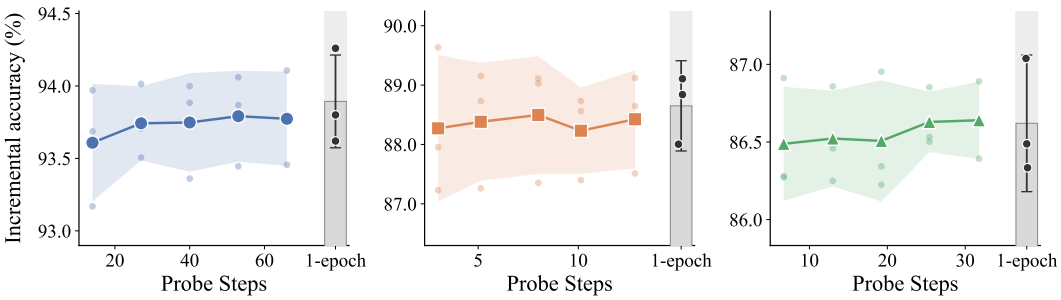
Table 5: Task-demand score ablation on imbalanced ImageNet-R. Each sequence has five small tasks followed by five large tasks, with $4/36$, $5/35$, or $8/32$ classes per small/large task for imbalance ratios of 9:1, 7:1, and 4:1, respectively. The per-layer, per-task rank cap is fixed at 256 across scores.

Score s_t	9:1		7:1		4:1	
	Last-Acc	Inc-Acc	Last-Acc	Inc-Acc	Last-Acc	Inc-Acc
Frobenius norm: $\mathcal{F}_t$	$\underline{77.60}$	$\underline{86.12}$	77.62	86.46	77.72	85.05
Effective rank: $r_{\mathrm{eff},t}$	77.28	85.77	$\underline{78.07}$	**86.69**	$\underline{77.78}$	$\underline{85.45}$
Equal scores: $s_t = 1$ ($\rho_t = 1/t$)	76.48	85.70	76.47	85.66	76.93	84.93
Inverse score: $1/(\mathcal{F}_t r_{\mathrm{eff},t})$	74.88	84.43	74.50	84.55	75.93	83.98
SPAR (Ours): $\mathcal{F}_t r_{\mathrm{eff},t}$	**78.90**	**86.88**	**78.63**	$\underline{86.50}$	**78.50**	**86.11**

5.3 Ablation and Analysis

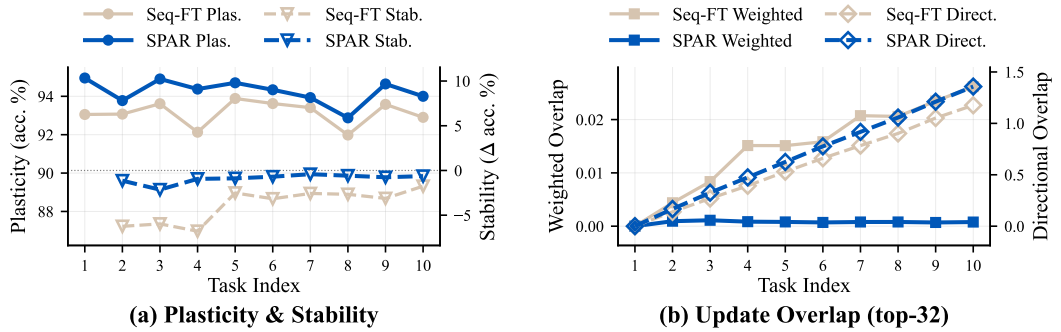
Effect of the demand score. Table 5 compares five task-demand scores. The product $\mathcal{F}_t r_{\mathrm{eff},t}$ gives the best Last-Acc in all settings and the best Inc-Acc in two of three. Relative to equal scores, which reduce the demand share to $1/t$, the product improves Last-Acc by 1.57–2.42 points. Inverting the score performs worst throughout. Together, these results favor combining magnitude and dimensionality and assigning greater allocation weight to larger probe scores.

Sensitivity to probe length. Figure 4 varies the probe steps on CIFAR-100, CUB-200, and ImageNet-R while keeping the subsequent training protocol unchanged. Short probes produce Inc-Acc close to the one-epoch default, with no consistent improvement from longer probes. This suggests that an early spectral estimate is sufficient for useful rank allocation and that fewer probing steps can reduce overhead without a large accuracy penalty in these settings.



(a) CIFAR-100 **(b) CUB-200** **(c) ImageNet-R**

Figure 4: Sensitivity to spectral-probe length on CIFAR-100, CUB-200, and ImageNet-R. The gray bar in each panel denotes the default one-epoch probe. Short probes achieve comparable accuracy.



(a) Plasticity & Stability **(b) Update Overlap (top-32)**

Figure 5: Plasticity, stability, and update overlap on ImageNet-R. (a) Plasticity is current-task accuracy after training; stability is the mean accuracy change on previous tasks caused by the current update. (b) Energy-weighted overlap and directional overlap, using the top-32 left singular vectors of each task update and summing over all previous tasks and the updated parameter matrices.

Plasticity, stability, and update overlap. Figure 5(a) shows that SPAR achieves higher current-task accuracy and less forgetting than Seq-FT. Panel 5(b) compares two overlap measures using the top-k left singular vectors U_j and corresponding singular values s_j of each task update ($k = 32$). Directional overlap, $\|U_i^\top U_j\|_F^2/k$, measures subspace alignment independently of update magnitude; energy-weighted overlap, $\|U_i^\top U_j \operatorname{diag}(s_j)\|_F^2$, measures the current update energy projected onto a previous task's leading subspace. Both are summed over recorded layers and all previous tasks $i < j$. Despite comparable or higher directional overlap, SPAR has much lower energy-weighted overlap than Seq-FT, whose weighted overlap grows along the sequence. These results suggest that retention depends not only on directional alignment but also on the energy carried by overlapping directions, supporting spectral-capacity control as a key factor in balancing stability and plasticity.

6 Conclusion

This work highlights spectral-capacity control as an effective alternative to explicit directional constraints in continual learning. Our spectrum-matching interventions recover or improve upon the corresponding directional methods in the evaluated settings. These results do not imply that update directions are unimportant. Rather, they suggest that, under suitable spectral-capacity control, optimization can learn directions that support new-task learning while retaining previous knowledge, without explicit directional constraints. SPAR provides a simple instantiation of this principle by probing task demand and adapting LoRA rank. More broadly, this perspective offers a practical design principle for continual learning: scheduling rank, learning rate, optimization steps, and other factors that control update spectra to achieve effective stability–plasticity trade-offs.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript editing, including language polishing and improving clarity and readability. All experiments were conducted by the authors. The authors are responsible for verifying all AI-assisted edits and take full responsibility for the final content of this work, including its text, claims, and artifacts.

REPRODUCIBILITY STATEMENT

The supplementary material includes code for reproducing our experiments, including the implementation of SPAR and the spectrum-matching experiments underlying our core analysis. The appendix provides detailed experimental settings, implementation details, additional results and ablations, as well as further explanations of the theoretical assumptions and results. Together, these materials document the procedures needed to reproduce and verify our main findings.

REFERENCES

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 (NeurIPS)*, 2020.

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

Ning Ding, Xingtai Lv, Qiaosen Wang, Yulin Chen, Bowen Zhou, Zhiyuan Liu, and Maosong Sun. Sparse low-rank adaptation of pre-trained language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Mehrdad Farajtabar, Navid Azizan, Alexander Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

Jiangpeng He, Zhihao Duan, and Fengqing Zhu. Cl-lora: Continual low-rank adaptation for rehearsal-free class-incremental learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Lingfeng He, De Cheng, Huaijie Wang, Xi Yang, Nannan Wang, and Xinbo Gao. Task-driven subspace decomposition for knowledge sharing and isolation in lora-based continual learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

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

Prakhar Kaushik, Ankit Vaidya, Shravan Chaudhari, Rama Chellappa, and Alan Yuille. Shared lora subspaces for almost strict continual learning. *arXiv preprint arXiv:2602.06043*, 2026.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the International Conference on Computer Vision Workshops (ICCVW)*, 2013.

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

Lan Li, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Addressing imbalanced domain-incremental learning through dual-balance collaborative experts. *arXiv preprint arXiv:2507.07100*, 2025.

Longhua Li, Lei Qi, Qi Tian, and Xin Geng. Energy-structured low-rank adaptation for continual learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026.

Yan-Shuo Liang and Wu-Jun Li. InfLoRA: Interference-free low-rank adaptation for continual learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

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

Mao-Lin Luo, Zi-Hao Zhou, Yi-Lin Zhang, Yuanyu Wan, Tong Wei, and Min-Ling Zhang. KeepLoRA: Continual learning with residual gradient adaptation. In *International Conference on Learning Representations (ICLR)*, 2026.

Mark D. McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton van den Hengel. Ranpac: Random projections and pre-trained models for continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

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

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Greg Wayne. Experience replay for continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *European Signal Processing Conference (EUSIPCO)*, 2007.

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

James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Hai-Long Sun, Da-Wei Zhou, Hanbin Zhao, Le Gan, De-Chuan Zhan, and Han-Jia Ye. Mos: Model surgery for pre-trained model-based class-incremental learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2025.

Yuwen Tan, Qinhao Zhou, Xiang Xiang, Ke Wang, Yuchuan Wu, and Yongbin Li. Semantically-shifted incremental adapter-tuning is a continual vitransformer. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical report, California Institute of Technology, 2011.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2018.

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

Yan Wang, Da-Wei Zhou, and Han-Jia Ye. Integrating task-specific and universal adapters for pre-trained model-based class-incremental learning. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2025.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022a.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022b.

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

Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. Slca: Slow learner with classifier alignment for continual learning on a pre-trained model. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023a.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations (ICLR)*, 2023b.

Ruiyi Zhang, Rushi Qiang, Sai Ashish Somayajula, and Pengtao Xie. AutoLoRA: Automatically tuning matrix ranks in low-rank adaptation based on meta learning. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2024.

Zangwei Zheng, Mingyuan Ma, Kai Wang, Ziheng Qin, Xiangyu Yue, and Yang You. Preventing zero-shot transfer degradation in continual learning of vision-language models. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023.

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

Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *International Journal of Computer Vision (IJCV)*, 133(3):1012–1032, 2025a.

Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, Lijun Zhang, and De-Chuan Zhan. Dual consolidation for pre-trained model-based domain-incremental learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025b.

Hao Zhu, Yifei Zhang, Junhao Dong, and Piotr Koniusz. Bilora: Almost-orthogonal parameter spaces for continual learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

A Additional Experimental Details and Results

A.1 Implementation Details

Spectrum replay. We first train the original continual-learning method and, after each task t, record the singular values of its incremental update at every adapted layer ℓ:

$$W_{t,\mathrm{ref}}^{(\ell)} - W_{t-1,\mathrm{ref}}^{(\ell)} = U_{t,\mathrm{ref}}^{(\ell)} \Sigma_{t,\mathrm{ref}}^{(\ell)} (V_{t,\mathrm{ref}}^{(\ell)})^{\top}. \tag{12}$$

Only the layer-wise target spectra $\Sigma_{t,\mathrm{ref}}^{(\ell)}$ are retained for replay, not the reference singular vectors. Each target is recorded at task completion and remains fixed throughout replay training of the corresponding task.

During spectrum-replay training, we apply a correction every K optimization steps. Suppressing the layer index, let $W_{t,0}$ denote the replay model's weight matrix immediately before task t. At correction step s, the current task update is

$$\Delta W_{t,s} = \begin{cases} W_{t,s} - W_{t,0}, & \text{direct fine-tuning,} \\ B_{t,s} A_{t,s}, & \text{LoRA,} \end{cases} \qquad \Delta W_{t,s} = U_{t,s} \Sigma_{t,s} V_{t,s}^{\top}. \tag{13}$$

Thus the update is measured relative to the task's starting model, not the preceding correction. We replace the current singular values with the recorded target while retaining the current singular vectors:

$$\widehat{\Delta W}_{t,s} = U_{t,s} \Sigma_{t,\mathrm{ref}} V_{t,s}^{\top}. \tag{14}$$

For direct fine-tuning, we write back $W_{t,s} \leftarrow W_{t,0} + \widehat{\Delta W}_{t,s}$. For LoRA, using the convention $\Delta W = BA$, we reconstruct the factors as

$$B_{t,s} \leftarrow U_{t,s} \Sigma_{t,\mathrm{ref}}^{1/2}, \qquad A_{t,s} \leftarrow \Sigma_{t,\mathrm{ref}}^{1/2} V_{t,s}^{\top}. \tag{15}$$

The LoRA reconstruction uses an SVD representation compatible with the adapter rank; exact matching requires that this rank can represent all nonzero target singular values. Training then resumes from the corrected parameters. The spectra are therefore matched at correction steps, while update directions remain free to evolve under optimization.

LoRA target modules. For vision Transformers, we apply LoRA to the `attn.qkv` and `mlp.fc1` linear layers. For RoBERTa in the NLP experiments, we apply LoRA to the `query` and `value` linear projections in the attention modules.

Spectral probing. The probe directly fine-tunes all entries of the LoRA-targeted matrices for one epoch, without LoRA. On near-balanced benchmark splits, comparable per-task sample counts and a fixed batch size within each benchmark make this an approximately step-matched protocol, limiting differences in training effort when estimating task demand. This approximation does not apply to the deliberately imbalanced sequences, where one epoch entails different update counts. We compute spectra from the differences between the probed and pre-probe matrices, then restore the pre-probe parameters before training with the allocated LoRA ranks.

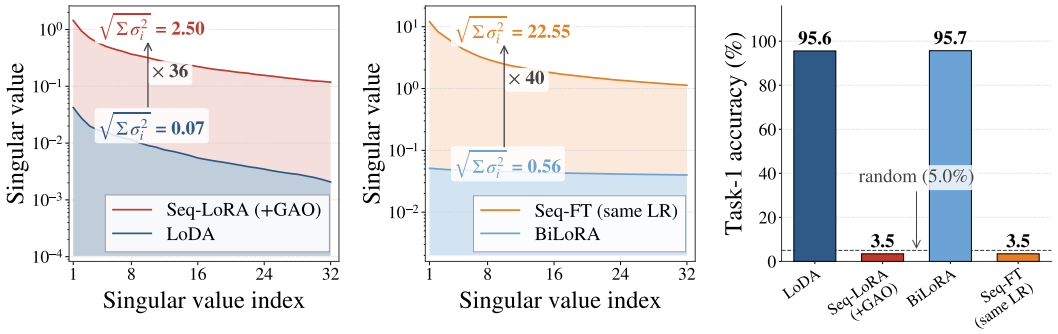
Random seeds. Unless otherwise specified, we run experiments with three random seeds: 1993, 1994, and 1995. We report the mean and standard deviation across these three runs.

A.2 Spectrum Replay across Continual-Learning Methods

Table 6 extends spectrum replay to two class-incremental benchmarks and a broader set of continual-learning methods, including SLCA and BiLoRA, whose designs are not reducible to directional constraints alone. For each method, we compare the original method, its sequential fine-tuning counterpart under the ablations specified in the table, and a spectrum-replay variant that matches the original method's task-wise update spectra while leaving the singular vectors free. Spectrum replay substantially improves performance over the corresponding sequential baselines, and in several cases matches or surpasses the original methods. These results extend the evidence for spectral-capacity control beyond purely directional interventions: spectra induced by different continual-learning mechanisms can serve as effective constraints for sequential fine-tuning without reproducing those mechanisms in full.

Table 6: Spectrum replay across continual-learning methods. Each group compares the original method with sequential baselines and a variant using the original method's update spectra without prescribed singular vectors. Subscripts show changes in percentage points relative to the corresponding original method; green and red backgrounds indicate increases and decreases, respectively.

Method	ImageNet-R		CUB-200	
	Last-Acc	Inc-Acc	Last-Acc	Inc-Acc
GPM (Saha et al., 2021)	73.95	78.72	77.10	82.06
− direction (Seq-FT)	$67.23_{(-6.72)}$	$74.76_{(-3.96)}$	$62.34_{(-14.76)}$	$73.43_{(-8.63)}$
− direction + spectrum	$73.80_{(-0.15)}$	$79.18_{(+0.46)}$	$76.63_{(-0.47)}$	$81.64_{(-0.42)}$
SLCA (Zhang et al., 2023a)	78.60	83.67	83.76	90.42
− slow learner (Seq-FT)	$29.52_{(-49.08)}$	$42.75_{(-40.92)}$	$30.87_{(-52.89)}$	$54.48_{(-35.94)}$
− slow learner + spectrum	$76.78_{(-1.82)}$	$82.97_{(-0.70)}$	$86.30_{(+2.54)}$	$91.34_{(+0.92)}$
BiLoRA (Zhu et al., 2025)	75.92	80.26	78.20	83.18
− bilinear fourier (Seq-FT, same LR)	$1.97_{(-73.95)}$	$5.42_{(-74.84)}$	$3.01_{(-75.19)}$	$4.33_{(-78.85)}$
− bilinear fourier (Seq-FT, 0.1× LR)	$35.87_{(-40.05)}$	$55.85_{(-24.41)}$	$33.50_{(-44.70)}$	$58.65_{(-24.53)}$
− bilinear fourier + spectrum	$70.32_{(-5.60)}$	$76.33_{(-3.93)}$	$80.32_{(+2.12)}$	$84.08_{(+0.90)}$
LoDA (He et al., 2026)	81.88	86.99	81.04	88.68
− direction (Seq-LoRA + GAO)	$0.85_{(-81.03)}$	$3.40_{(-83.59)}$	$0.93_{(-80.11)}$	$1.59_{(-87.09)}$
− direction − GAO (Seq-LoRA)	$67.37_{(-14.51)}$	$79.40_{(-7.59)}$	$64.80_{(-16.24)}$	$79.87_{(-8.81)}$
− direction + spectrum	$81.55_{(-0.33)}$	$86.90_{(-0.09)}$	$83.25_{(+2.21)}$	$89.94_{(+1.26)}$



(a) LoDA vs Seq-LoRA (b) BiLoRA vs Seq-FT (c) First-task accuracy

Figure 6: First-task collapse after removing core continual-learning mechanisms. (a) LoDA versus Seq-LoRA with its directional constraints removed but GAO retained. (b) BiLoRA versus Seq-FT with its core mechanism removed and the original learning rate retained. (c) First-task accuracy for all four variants. Both ablated variants exhibit substantially greater update energy and collapse already on the first task, before forgetting previous tasks can occur.

First-task collapse and recovery through spectrum replay. The LoDA and BiLoRA ablations require particular care because directly removing their core mechanisms causes collapse before continual forgetting can even arise. Removing LoDA's directional constraints leaves its meta-learning component, GAO, in place, yielding Seq-LoRA with GAO. For BiLoRA, removing its core mechanism and applying sequential fine-tuning at the original learning rate also leads to collapse. Figure 6 shows that the Frobenius norms of the first-task parameter update matrices, i.e., the spectral energy used here, $\|\Delta W\|_F = \sqrt{\sum \sigma_i^2}$, are approximately $36\times$ and $40\times$ those of LoDA and BiLoRA, respectively. Their task-1 accuracy collapses to 3.5%, compared with 95.6% for LoDA and 95.7% for BiLoRA. Since no earlier tasks need to be retained at this point, this failure cannot be attributed to forgetting previous tasks; instead, it is consistent with excessive update energy disrupting the pretrained model and preventing successful first-task adaptation.

To obtain viable sequential baselines, we therefore additionally remove GAO from the LoDA-derived variant and reduce the learning rate to $0.1\times$ its original value for the BiLoRA-derived vari-

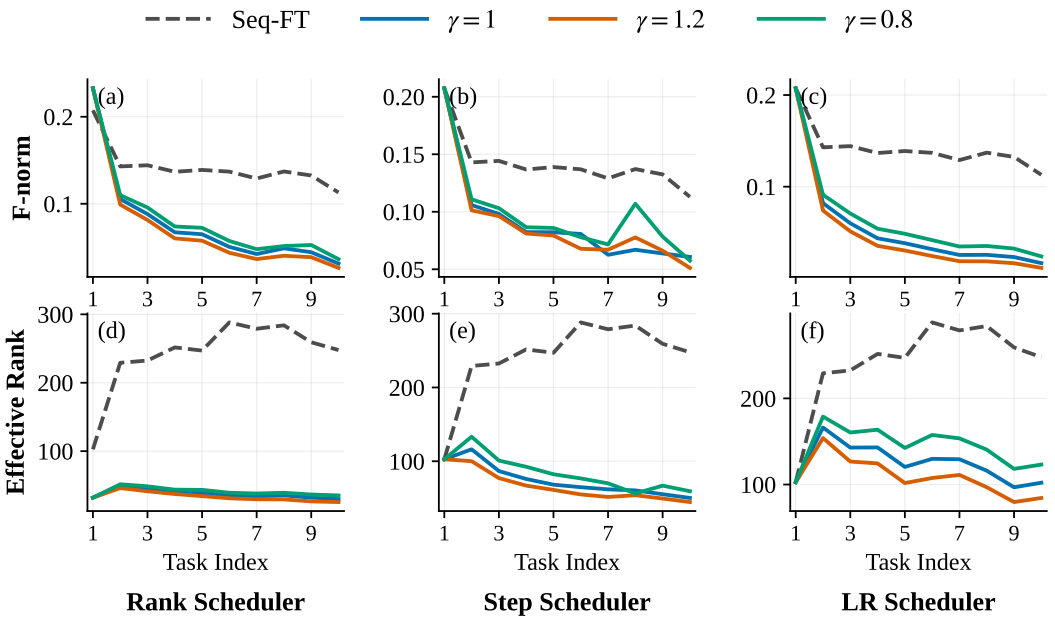


Figure 7: Detailed spectral statistics for the schedules in Figure 3 on Cars-196. Columns correspond to LoRA-rank, optimization-step, and learning-rate scheduling, respectively. Panels (a)–(c) report the Frobenius norm $\mathcal{F}_t$, and panels (d)–(f) report effective rank $r_{\text{eff},t}$. Each panel compares constant-setting Seq-FT with power-law decay exponents $\gamma \in \{0.8, 1.0, 1.2\}$.

ant. These adjustments alleviate collapse but leave substantial performance gaps (Table 6). The adjusted LoDA baseline remains 14.51 and 16.24 percentage points below the original method in Last-Acc on ImageNet-R and CUB-200, respectively; the corresponding deficits for the adjusted BiLoRA baseline are 40.05 and 44.70 points. Thus, simply removing GAO or lowering the learning rate does not recover the original methods' continual-learning performance.

In contrast, constraining sequential updates with the original methods' target spectra avoids collapse and largely recovers their performance without restoring the removed core mechanisms. With LoDA's spectra, Last-Acc reaches 81.55% on ImageNet-R, compared with the original 81.88%, and exceeds the original result on CUB-200 (83.25% versus 81.04%). With BiLoRA's spectra, Last-Acc recovers to 70.32% on ImageNet-R, although a 5.60-point gap remains, and surpasses the original result on CUB-200 (80.32% versus 78.20%). These results indicate that the spectra induced by the original methods provide effective control of update capacity.

Interpretation and scope of spectrum replay. Our central claim is that suitable spectral-capacity control can improve continual-learning performance, not that update directions are unimportant. Replay imposes a strong constraint on the entire singular-value profile, while optimization still learns directions from task data. Its success shows that the recorded spectra provide useful control without reproducing the original directional prescriptions in the evaluated settings; it does not imply that arbitrary or random directions would work. Replay alone also cannot identify whether the full profile, magnitude, or effective dimensionality is essential. The training-factor experiments in Section 3.3 provide complementary evidence: simpler controls improve performance without matching a reference spectrum. Together, these results support spectral-capacity control as a useful design principle, rather than spectral dominance over direction.

A.3 Detailed Spectral Statistics for Training-Factor Schedules

Figure 7 complements Figure 3 by reporting the Frobenius norm $\mathcal{F}_t$ and effective rank $r_{\text{eff},t}$ separately on Cars-196, rather than only their product. Compared with constant-setting Seq-FT, decaying LoRA rank, optimization steps, or learning rate reduces both statistics over the task sequence. The

Table 7: Adaptive-rank LoRA comparison on class- and domain-incremental benchmarks.

Method	ImageNet-R	CIFAR-100	CUB-200	Office-Home
Last-Acc				
AdaLoRA (Zhang et al., 2023b)	$72.43_{\pm 0.17}$	$85.28_{\pm 0.13}$	$79.80_{\pm 0.91}$	$83.69_{\pm 0.35}$
SoRA (Ding et al., 2023)	$76.18_{\pm 0.20}$	$87.58_{\pm 0.88}$	$81.08_{\pm 0.24}$	$85.90_{\pm 0.14}$
AutoLoRA (Zhang et al., 2024)	$76.08_{\pm 0.32}$	$87.69_{\pm 0.24}$	$80.68_{\pm 0.59}$	$85.70_{\pm 0.26}$
SPAR (Ours)	$\mathbf{81.66}_{\pm 0.15}$	$\mathbf{90.05}_{\pm 0.68}$	$\mathbf{82.53}_{\pm 0.50}$	$\mathbf{88.59}_{\pm 0.29}$
Inc-Acc				
AdaLoRA (Zhang et al., 2023b)	$78.53_{\pm 1.14}$	$90.09_{\pm 0.14}$	$86.69_{\pm 1.94}$	$81.03_{\pm 0.62}$
SoRA (Ding et al., 2023)	$82.05_{\pm 0.45}$	$91.88_{\pm 0.22}$	$87.61_{\pm 1.02}$	$83.45_{\pm 0.09}$
AutoLoRA (Zhang et al., 2024)	$81.74_{\pm 0.52}$	$91.94_{\pm 0.29}$	$87.39_{\pm 1.18}$	$83.16_{\pm 0.08}$
SPAR (Ours)	$\mathbf{86.62}_{\pm 0.44}$	$\mathbf{93.89}_{\pm 0.33}$	$\mathbf{88.65}_{\pm 0.76}$	$\mathbf{86.18}_{\pm 0.19}$

Table 8: Final scores on the GLUE benchmark after completing the full task sequence.

Method	CoLA	MRPC	RTE	STS-B	QNLI	SST-2	Avg.
Seq-FT	13.0	68.1	54.9	65.5	83.9	**94.7**	63.4
Seq-LoRA ($r = 32$) (Hu et al., 2022)	28.4	69.1	52.7	62.6	78.1	93.3	64.0
LoDA (He et al., 2026)	28.6	**75.8**	61.2	78.3	76.0	90.8	68.5
E^2-LoRA (Li et al., 2026)	29.0	72.7	62.8	**81.6**	88.3	92.7	71.2
SPAR (Ours)	**30.1**	71.9	**63.1**	80.7	**89.2**	93.3	**71.4**

reduction in the spectral-capacity proxy therefore reflects both smaller update magnitudes and more concentrated spectra, rather than a change in only one component. Notably, step and learning-rate decay also reduce effective rank without explicitly imposing a tighter rank bound, while rank decay reduces the Frobenius norm as well as effective dimensionality. These observations support the use of all three training factors to regulate spectral capacity without prescribing update directions.

A.4 Comparison with Adaptive-Rank LoRA Methods

Our main contribution is the finding that spectral-capacity control can recover the performance of direction-constrained continual learning, rather than adaptive rank allocation itself. This finding motivates multiple designs, including learning-rate, optimization-step, and rank schedules; SPAR is a simple demand-aware instantiation. Adaptive LoRA methods replace uniform ranks with data-dependent allocations for parameter-efficient fine-tuning. AdaLoRA allocates a parameter budget across weight matrices by importance-based pruning of an SVD-parameterized update (Zhang et al., 2023b); SoRA learns sparse gates with proximal gradient updates to adjust adapter ranks (Ding et al., 2023); and AutoLoRA uses meta-learned selection variables to choose layer-wise ranks (Zhang et al., 2024). Their original formulations target downstream adaptation quality and parameter efficiency, rather than retention across a sequential task stream. In contrast, SPAR uses current and accumulated probe scores within each layer to control capacity across tasks, targeting the stability–plasticity trade-off rather than only the performance–parameter-efficiency trade-off.

Table 7 compares these methods on four continual learning benchmarks. SPAR achieves the highest mean Last-Acc and Inc-Acc on all four datasets, supporting the effectiveness of its history-aware capacity control in these continual-learning settings.

A.5 Continual Learning on GLUE

Table 8 reports final task scores on six GLUE (Wang et al., 2018) tasks. CoLA is evaluated using the Matthews correlation coefficient (MCC), STS-B using the Pearson correlation coefficient, and the remaining datasets using accuracy. SPAR achieves the highest average score among the compared methods (71.4), improving over Seq-FT and fixed-rank Seq-LoRA and slightly exceeding E^2-LoRA (71.2). It leads on CoLA, RTE, and QNLI, although LoDA performs better on MRPC and E^2-LoRA on STS-B. These results support the applicability of demand-aware rank allocation beyond vision:

Table 9: Comparison with accuracy-guided rank search. Each dataset's shared Spearman coefficient measures agreement between the searched ranks $\widehat{r}_t^\star$ and the ranks allocated by SPAR.

Method	ImageNet-R			CUB-200		
	Spearman	Last-Acc	Inc-Acc	Spearman	Last-Acc	Inc-Acc
Rank Search	0.9871	$81.43_{\pm 0.32}$	$86.79_{\pm 0.42}$	0.9885	$82.89_{\pm 0.26}$	$88.81_{\pm 1.07}$
SPAR (Ours)		$81.66_{\pm 0.15}$	$86.62_{\pm 0.44}$		$82.53_{\pm 0.50}$	$88.65_{\pm 0.76}$

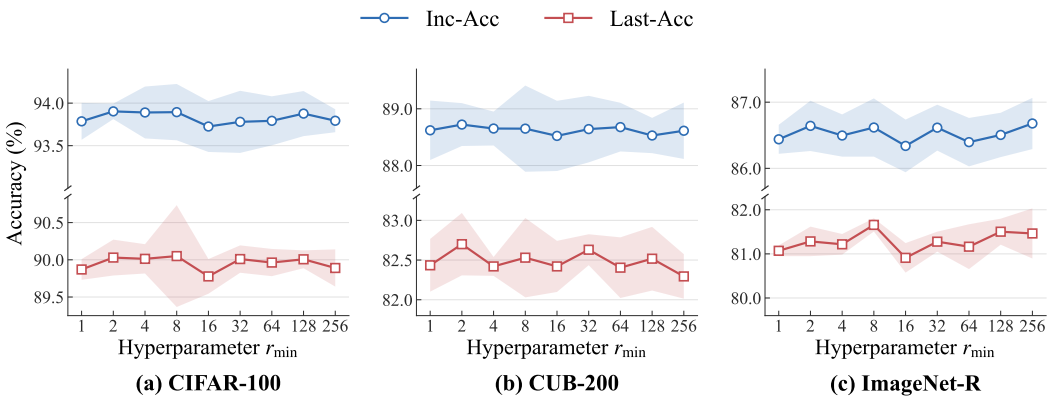


Figure 8: Sensitivity to the minimum LoRA rank $r_{\min}$ on (a) CIFAR-100, (b) CUB-200, and (c) ImageNet-R. Blue circles denote Inc-Acc and red squares denote Last-Acc. Both metrics remain broadly stable as the rank floor varies from 1 to 256.

it improves average continual-learning performance without explicit directional constraints, while retaining task-dependent trade-offs rather than uniformly dominating every task.

A.6 Validation against Accuracy-Guided Rank Search

To assess whether spectral probing yields appropriate rank allocations beyond improving overall accuracy, we compare SPAR with an accuracy-guided rank-search reference on ImageNet-R and CUB-200. Suppressing the layer index, let $\mathrm{Acc}_t(r)$ denote the accuracy criterion used to select rank r at task t, and define the grid oracle as

$$r_t^\star \in \arg\max_{r \in \mathcal{R}} \mathrm{Acc}_t(r), \qquad \mathcal{R} = \{8k : k = 1, \ldots, \lfloor d_{\mathrm{out}}/8 \rfloor\}. \tag{16}$$

We approximate this oracle by discrete ternary search over ranks from 8 to d_{out} in increments of 8, retaining the best evaluated rank $\widehat{r}_t^\star$. This is a search-based reference rather than a certified global optimum, since accuracy need not be unimodal in rank. Table 9 reports Spearman correlation between the searched ranks and SPAR's allocated ranks, together with Last-Acc and Inc-Acc for each method. Rank agreement tests whether the probe-driven scheduler tracks accuracy-preferred capacity, while the two accuracy metrics assess the performance cost of replacing rank search with spectral probing. This comparison evaluates the score together with its online normalization; direct calibration of the raw score would additionally require $\mathrm{Spearman}(s_t, \widehat{r}_t^\star)$.

A.7 Sensitivity to the Minimum Rank

Figure 8 evaluates the sensitivity of SPAR to the minimum LoRA rank $r_{\min}$ on CIFAR-100, CUB-200, and ImageNet-R, sweeping $r_{\min} \in \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$. Both Last-Acc and Inc-Acc remain broadly stable across this range, with no consistent improvement from increasing the rank floor. ImageNet-R exhibits somewhat larger fluctuations in Last-Acc, but these changes are non-monotonic. These results suggest that SPAR does not require precise tuning of $r_{\min}$ within the tested range and support $r_{\min} = 8$ as a practical default rather than a universally optimal choice.

Table 10: Total training time (hh:mm:ss) on the class-incremental benchmarks.

Method	ImageNet-R	CIFAR-100	CUB-200	Cars-196
LoDA (He et al., 2026)	00:45:47	01:02:18	00:22:37	02:07:55
E^2-LoRA (Li et al., 2026)	00:31:24	00:45:37	00:16:54	01:49:56
SPAR (Ours)	00:17:15	00:30:53	00:13:27	01:17:46

A.8 Training Time

Table 10 reports total wall-clock training time over the complete task sequence for each benchmark. All runs use a single NVIDIA GeForce RTX 5090 GPU, and the reported time includes spectral probing for SPAR and classifier alignment where applicable. SPAR is the fastest of the three methods on all four benchmarks while maintaining competitive or superior accuracy (Table 2). LoDA and E^2-LoRA rely on more elaborate components, including update-direction search and constraints, meta-learning, or self-distillation, which add computational overhead. In contrast, SPAR uses a simple probe-based rank scheduler without explicitly searching for or constraining update directions. Although probing adds one epoch per task, its modest cost does not offset the overall time savings, highlighting the efficiency of demand-aware spectral-capacity control.

B Spectral-Capacity Control: Rationale and Analysis

This appendix complements our empirical findings with sufficient conditions for capacity decay and an interpretation of demand-aware scheduling, without claiming an optimal algorithm.

From spectrum replay to practical controls. Spectrum replay shows that suitable spectra support effective continual learning while leaving directions to optimization. The observed decline in spectral capacity motivates decaying learning rate, optimization steps, or LoRA rank. Tested independently, these schedules reshape update spectra and improve performance (Figure 3), providing practical alternatives to full-spectrum matching.

From fixed decay to demand-aware control. Balanced class-incremental splits motivate comparable task demands as a reference case, but fixed decay cannot adapt to heterogeneous demands. SPAR uses probe spectra to adjust rank by relative demand, an extension tested in the imbalanced-task experiments. Rank scheduling need not reproduce the reference spectrum or probe update.

Proposition 1 establishes conditional capacity decay under homogeneous demand; the subsequent analyses explain the probe score and its demand-aware normalization.

B.1 Sufficient Conditions Motivating Capacity Decay

Average performance with heterogeneous interference. We formalize Equation 7 as a conditional surrogate. Let $\mathbf{c} = (c_1, \ldots, c_T) \in [0, C]^T$, where c_t is an abstract adaptation capacity . Approximate final expected performance on task k by

$$\widetilde{a}_{T,k}(\mathbf{c}) = b_k + g_k(c_k) - \sum_{t=k+1}^{T} h_{k,t}(c_t), \tag{17}$$

where b_k is allocation-independent, g_k is the adaptation benefit, and $h_{k,t}$ is the degradation caused by task t. Each g_k and $h_{k,t}$ is continuous and non-decreasing, with $h_{k,t}(0) = 0$. Pairwise interference is heterogeneous but additive and independent of earlier capacities and the optimization trajectory; balanced task sizes alone do not justify this assumption.

Reordering the interference terms gives the equally weighted final average

$$\widetilde{A}_T(\mathbf{c}) = \frac{1}{T}\sum_{k=1}^{T} b_k + \frac{1}{T}\sum_{t=1}^{T} \left[g_t(c_t) - H_t(c_t) \right], \qquad H_t(c) = \sum_{k<t} h_{k,t}(c). \tag{18}$$

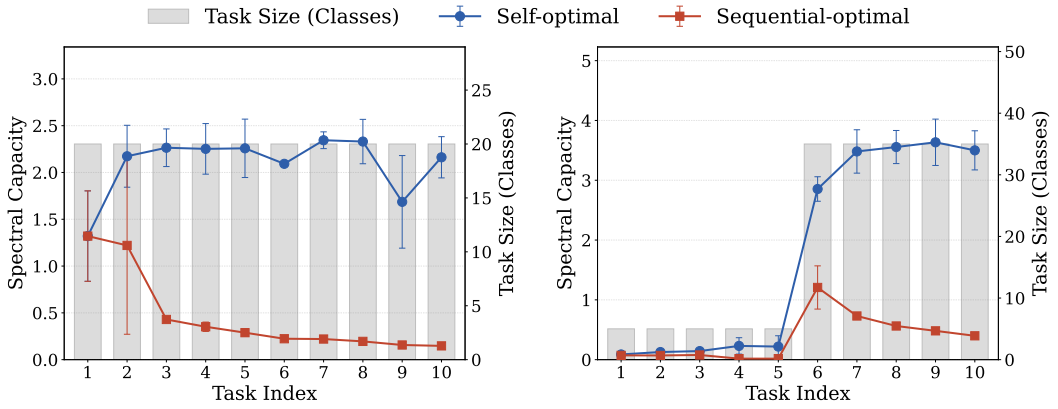


Figure 9: Empirical optimal spectral capacity under approximately homogeneous and heterogeneous task demands. **Self-optimal** denotes the capacity that maximizes current-task performance, while **Sequential-optimal** maximizes mean performance over all tasks observed so far; gray bars show classes per task. In the balanced sequence, all tasks contain 20 classes. In the imbalanced sequence, the first five tasks contain 5 classes each and the last five contain 35 classes each.

With independent bounds and no shared budget, maximization separates into $\max_{c\in[0,C]} J_t(c)$, where $J_t = g_t - H_t$. These unknown functions serve the analysis, not online optimization.

Conditions for sequence-wise decay. Assume additionally that (i) $g_t = g$ (equal adaptation demand) and (ii) $H_{t+1} - H_t$ is non-decreasing on $[0, C]$ (non-decreasing cumulative marginal interference). When differentiable, (ii) is equivalent to $H'_{t+1}(c) \geq H'_t(c)$ for interior c. Equal class counts or more preceding tasks alone imply neither condition. A sufficient case of (ii) is $h_{k,t}(c) = h_k(c)$ for all $t > k$, giving $H_{t+1} - H_t = h_t$ while allowing task-specific sensitivities. Linear interference $h_{k,t}(c) = \lambda_{k,t}c$ also satisfies (ii) whenever $\sum_{k\leq t} \lambda_{k,t+1} \geq \sum_{k<t} \lambda_{k,t}$.

Proposition 1 (Conditional monotonicity)**.** Under the preceding assumptions, select the smallest maximizer $c^\star_t = \min \arg\max_{c\in[0,C]} J_t(c)$. Then $c^\star_{t+1} \leq c^\star_t$. Consequently, these choices form a non-increasing maximizer of Equation 18. If every J_t is strictly concave, its maximizer is unique and no tie-breaking convention is needed.

Proof. Continuity on the compact interval ensures that the smallest maximizers exist. Write $x = c^\star_t$, $y = c^\star_{t+1}$, and $D_t = H_{t+1} - H_t$, so that $J_{t+1} = J_t - D_t$. Suppose $y > x$. Optimality and the monotonicity of D_t imply

$$0 \leq J_{t+1}(y) - J_{t+1}(x) = J_t(y) - J_t(x) - [D_t(y) - D_t(x)] \leq 0. \tag{19}$$

Equality would make x a maximizer of J_{t+1}, contradicting the minimality of $y > x$; hence $y \leq x$. Separability gives the full-sequence result. The argument covers boundary solutions and saturated benefits without requiring differentiability, benefit concavity, or interference convexity. □

Balanced splits as a reference case. Equal-class splits with comparable task sizes motivate approximately homogeneous benefits. If cumulative marginal interference also grows, Proposition 1 balances similar benefits against greater retention costs. This motivates testing non-increasing rank schedules, not a claim that required ranks must decrease.

Empirical effect of heterogeneous task demand. Figure 9 complements the conditional analysis with a direct capacity sweep. In the balanced sequence, the self-optimal capacity is broadly stable after the initial task, whereas the sequential-optimal capacity decreases as more previous tasks must be retained, consistent with the reference case in Proposition 1. In the imbalanced sequence, the self-optimal capacity rises sharply when the sequence switches from 5-class to 35-class tasks.

The sequential-optimal capacity also increases at this transition instead of continuing to decay, and then decreases within the high-demand block as historical constraints accumulate. Thus, sequence position provides a useful decay prior when task demands are similar, but heterogeneous demand can locally override that trend. This observation motivates correcting a fixed decay schedule using a task-dependent demand estimate, as done by SPAR.

Unequal task demands can reverse decay. With heterogeneous demands, greater current-task benefit can outweigh increased interference. At a differentiable interior optimum, $g_t'(c_t^\star) = H_t'(c_t^\star)$. For a two-task example on $[0, 1]$, let

$$g_1(c) = \min\{c, \tfrac{1}{4}\}, \qquad g_2(c) = c - \tfrac{1}{2}c^2, \qquad h_{1,2}(c) = \tfrac{1}{4}c. \tag{20}$$

The first benefit saturates at $c_1^\star = 1/4$, its smallest maximizer; $J_2(c) = 3c/4 - c^2/2$ is uniquely maximized at $c_2^\star = 3/4$. Thus the optimal allocation increases despite added interference. Saturation is essential here: a strictly increasing g_1 with no capacity cost would instead give $c_1^\star = C$. Selecting the smallest maximizer avoids unnecessary capacity after saturation.

More previous tasks need not imply greater interference. Equal demand alone is insufficient without (ii). For three tasks with $g(c) = c - c^2/2$ on $[0, 1]$, choose $h_{1,2}(c) = 0.6c$ and $h_{1,3}(c) = h_{2,3}(c) = 0.1c$. Then $H_2(c) = 0.6c$ but $H_3(c) = 0.2c$, yielding unique optima $c_2^\star = 0.4 < c_3^\star = 0.8$. The later task affects more prior tasks yet has lower total marginal interference.

Connection to rank control and scope. LoRA rank bounds update dimensionality but need not control magnitude or effective rank monotonically; its link to the surrogate is therefore empirical (Figure 3). Proposition 1 guarantees only conditional non-increase, not strict decay or a $1/t$ law, and excludes trajectory-dependent interference, backward transfer, and shared budgets. The counterexamples motivate demand-aware control but do not establish any scheduler's optimality.

B.2 Probe Spectra as a Simple Demand Proxy

A low-rank approximation perspective. Fix a task and layer, suppress their indices, and let $P = \Delta W_{\text{probe}} \neq 0$. Write its singular values in descending order and let $\mathcal{F} = \|P\|_F$ and $p_i = \sigma_i^2/\mathcal{F}^2$. The best rank-$r$ approximation has squared error (Eckart & Young, 1936)

$$E_r(P) = \min_{\text{rank}(X)\le r} \|P - X\|_F^2 = \sum_{i>r} \sigma_i^2 = \mathcal{F}^2 \left(1 - \sum_{i=1}^{r} p_i\right). \tag{21}$$

Absolute error depends on magnitude and spectral concentration, whereas relative error $E_r(P)/\mathcal{F}^2$ removes magnitude. This motivates using both statistics but does not specify their combination.

If P approximates a useful local target and, with other layers fixed, the loss admits the isotropic quadratic surrogate

$$\widehat{\mathcal{L}}(X) = \widehat{\mathcal{L}}(P) + \frac{\lambda}{2}\|X - P\|_F^2, \qquad \lambda > 0. \tag{22}$$

then the minimum rank-constrained excess loss is $\lambda E_r(P)/2$. This view is limited: the probe need not be locally optimal, and the surrogate ignores anisotropic curvature and layer interactions.

Limits of the approximation perspective. Effective rank in Equation 5, with $0 \log 0 = 0$, equals m for m equal nonzero singular values. It summarizes concentration but determines neither spectral tails nor optimal LoRA rank; tail-energy criteria also omit historical-task losses. Thus approximation theory motivates these statistics, not a guaranteed mapping from probe spectra to rank allocations.

Why a multiplicative proxy? Let $R = r_{\text{eff}}$ and consider scores $q(\mathcal{F}, R) > 0$ for $\mathcal{F} > 0$ and $1 \le R \le \min\{d_{\text{out}}, d_{\text{in}}\}$. We choose unit elasticity in each statistic: for positive scaling factors that keep the arguments in this domain,

$$q(a\mathcal{F}, R) = a\, q(\mathcal{F}, R), \qquad q(\mathcal{F}, bR) = b\, q(\mathcal{F}, R). \tag{23}$$

Applying these identities from the reference point $(1, 1)$ gives

$$q(\mathcal{F}, R) = \mathcal{F} q(1, R) = \mathcal{F} R q(1, 1) \propto \mathcal{F} R. \tag{24}$$

The product follows from the chosen unit-elasticity rule, not a derivation of true task demand. Within the broader family $\mathcal{F}^{\alpha}R^{\beta}$, we choose $(\alpha, \beta) = (1, 1)$ for simplicity and to avoid tuning additional exponents, without claiming optimality.

Comparisons with simpler scores (Table 5) and accuracy-guided rank search (Appendix A.6) support the heuristic's utility, not universal calibration. Scores also depend on the starting model and probing effort, especially when one epoch entails different update counts; equal rank caps need not imply equal cumulative cost.

B.3 Online Relative-Demand Normalization

We analyze SPAR's normalization independently of Proposition 1. For a fixed layer with positive scores s_t, define $S_0 = 0$, $S_t = S_{t-1} + s_t$, and $\bar{s}_t = S_t/t$.

Inverse-task decay with a demand correction. Direct substitution yields the exact identity

$$\rho_t = \frac{s_t}{S_t} = \frac{1}{t}\frac{s_t}{\bar{s}_t}. \tag{25}$$

Thus $1/t$ decay is corrected by current demand relative to its running mean; equal scores recover $\rho_t = 1/t$. If scores through task t lie in $[s_\circ(1-\varepsilon), s_\circ(1+\varepsilon)]$, with $s_\circ > 0$ and $0 \le \varepsilon < 1$, then

$$\frac{1-\varepsilon}{1+\varepsilon}\frac{1}{t} \le \rho_t \le \frac{1+\varepsilon}{1-\varepsilon}\frac{1}{t}. \tag{26}$$

Similar scores therefore retain approximate inverse-task decay, an algebraic property of the normalization rather than a consequence of balanced splits.

Connection to a streaming weighted mean. For arbitrary scalar observations x_i weighted by these scores, their weighted mean satisfies

$$m_t = \frac{\sum_{i=1}^{t} s_i x_i}{S_t} = (1-\rho_t)m_{t-1} + \rho_t x_t, \qquad t \ge 2, \tag{27}$$

with $m_1 = x_1$. SPAR borrows the new-observation coefficient ρ_t, not parameter averaging or its statistical guarantees.

An online quadratic interpretation. Let $a \in [0, 1]$ be the fraction of the rank interval above its floor, and consider

$$\Phi_t(a) = \frac{1}{2}S_{t-1}a^2 + \frac{1}{2}s_t(1-a)^2. \tag{28}$$

The terms model historical conservatism and unmet current demand, not measured interference costs. Since $\Phi_t'(a) = S_t a - s_t$ and $\Phi_t''(a) = S_t > 0$, the unique minimizer is

$$a_t^\star = \frac{s_t}{S_{t-1} + s_t} = \rho_t. \tag{29}$$

This includes $a_1^\star = 1$; mapping to the rank interval and rounding gives Equation 10. The construction balances current and accumulated demand, but is optimal only for Φ_t, not actual training loss or continual performance.

Exact properties and limits of the scheduler. For positive scores, $0 < \rho_t \le 1$ and uniform score rescaling leaves it unchanged. Holding the other quantity fixed, ρ_t increases with current demand when $S_{t-1} > 0$ and decreases with accumulated demand. It need not decay along the sequence: scores $(1, 1, 6)$ yield shares $(1, 1/2, 3/4)$, although rounding can suppress rank changes. The first task receives the cap; the floor limits later shrinkage.

Writing the unrounded rank as $\widetilde{r}_t = r_{\min} + (r_{\max} - r_{\min})\rho_t$, equal scores give

$$\sum_{t=1}^{T}(\widetilde{r}_t - r_{\min}) = (r_{\max} - r_{\min})\sum_{t=1}^{T}\frac{1}{t}. \tag{30}$$

The cap is therefore per task and layer, not a shared budget; cumulative rank is not stored adapter size after merging. Scheduling requires one cumulative scalar per layer, or $O(L)$ persistent state, excluding model parameters and temporary probing.