SPECTRAL CORE-PRESERVING OPTIMIZATION FOR CONTINUAL LEARNING WITH MODEL MERGING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual learning aims to enable large pretrained models to acquire new knowledge while retaining previously learned capabilities, yet remains challenged by catastrophic forgetting during sequential adaptation. Existing model-merging-based approaches mitigate forgetting by integrating model updates across tasks, but most still rely on standard fine-tuning and largely overlook the resulting perturbations to the pretrained core subspace. We find that old-task loss is more sensitive to parameter perturbations along pretrained core directions than along residual directions, making new-task updates along these directions more likely to interfere with previously learned knowledge. To address this issue, we propose Spectral Core-Preserving Optimization (SCPO), a fine-tuning method that protects the pretrained core subspace when learning new tasks and produces task updates better suited to model merging-based continual learning. SCPO defines the protection scope using the dominant singular subspaces of the pretrained weights and adaptively modulates the protection strength according to their overlap with the corresponding subspaces of the previous merged model, thereby selectively attenuating gradient components along protected directions. The filtered gradients are further optimized within an online-updated low-dimensional subspace, and the resulting task vectors are subsequently integrated through model merging. Extensive experiments on vision models and large language models demonstrate the effectiveness of SCPO. Across five representative model merging methods, SCPO consistently improves over standard fine-tuning, yielding an average relative accuracy gain of 5.44 % and a maximum gain of 8.63 %.

1 INTRODUCTION

Continual learning aims to enable models to acquire knowledge from sequentially arriving tasks while preserving previously learned capabilities. For large pre-trained models deployed in dynamic environments, this capability enables them to continuously adapt to new data and evolving application requirements without requiring access to all historical data or retraining the model from scratch each time. However, adapting to new tasks can degrade performance on earlier ones, leading to catastrophic forgetting. Existing approaches mainly mitigate this problem through parameter regularization, knowledge distillation, and experience replay. More specifically, regularization constrains changes to parameters important for previous tasks but requires maintaining additional importance statistics; knowledge distillation preserves previous capabilities through a teacher model but introduces additional computation; and experience replay revisits historical samples but requires data storage and may be restricted by privacy or data-retention requirements (Kirkpatrick et al., 2017; Li & Hoiem, 2016; Buzzega et al., 2020). More recently, model merging has provided another route to knowledge retention by integrating models or parameter updates obtained at different stages directly in weight space (Marczak et al., 2025b; Yang et al., 2026b). We study a continual fine-tuning and merging pipeline without replaying training samples from previous tasks, as illustrated in Figure 1. Each new task starts from the previous merged model, and its local update is integrated with accumulated updates. The resulting model serves both the seen tasks and as the initialization for the next task, making each task update important for both knowledge retention and subsequent adaptation.

In merging-based continual learning, knowledge retention depends on both how new task updates are learned and how they are integrated with accumulated updates. Early approaches such as Task Arithmetic combine task vectors through simple linear operations. As the number of tasks grows, in-

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.

terference among task updates becomes increasingly pronounced, motivating methods that resolve sign conflicts, retain larger-magnitude updates, or project new updates onto less interfering subspaces (Ilharco et al., 2023; Yadav et al., 2023; Marczak et al., 2025b; Tang et al., 2025). These methods mainly operate after fine-tuning, whereas recent work modifies the fine-tuning process itself to produce models that are more amenable to subsequent merging. Tangent-space fine-tuning improves weight disentanglement through model linearization but incurs additional inference overhead (Ortiz-Jimenez et al., 2023). SAFT-Merge uses sharpness-aware optimization to obtain flatter solutions that are more robust to subsequent parameter changes, at the cost of additional forward and backward passes (Lee et al., 2025). MergOPT instead models merge-induced parameter shifts and simulates such perturbations during fine-tuning to improve robustness to subsequent merging (Yang et al., 2026a). Despite improving mergeability or robustness to merging perturbations, these methods do not explicitly characterize how parameter updates along different directions affect previously learned tasks, leaving unclear along which directions updates during new-task fine-tuning are most likely to disrupt existing capabilities.

Recent work suggests that different spectral directions in pretrained weights are not equally important for preserving existing capabilities. Core Subspace Preservation (CSP) links the core subspace spanned by the dominant left and right singular vectors of pretrained weights to multi-task generalization and constrains perturbations to this subspace during model merging (Sun et al., 2026). However, in continual learning, each new task is initialized from the previous merged model rather than the original pretrained model, so it remains unclear whether perturbations along the pretrained core directions still induce larger changes in old-task loss after successive task updates. To examine this, we apply equal-norm symmetric perturbations sampled from the pretrained core subspace and its orthogonal complement around merged models obtained at different stages. We find that perturbations along core directions induce larger average increases in old-task loss than those along the orthogonal complement. Thus, even at the same update norm, parameter changes along different spectral directions can affect previously learned tasks differently.

Building on this observation, we propose Spectral Core-Preserving Optimization (SCPO), which selectively attenuates core-related gradient components during new-task fine-tuning. SCPO uses the dominant singular subspace of the pretrained weights as a fixed protection reference and modulates the protection strength according to its overlap with the corresponding subspace of the previous merged model. Directions with stronger spectral overlap receive greater attenuation, while components in the orthogonal complement remain unchanged. The filtered gradients are then optimized with Online Subspace Descent (OSD; Liang et al., 2024), allowing core protection to directly shape each task update. SCPO modifies only the fine-tuning stage and leaves the subsequent merging rule unchanged, making it compatible with different merging methods. We evaluate SCPO across three CLIP-ViT backbones, task sequences of varying lengths, and a Llama model. In the eight-task vision setting, SCPO improves over standard AdamW fine-tuning with all five representative merging methods, yielding an average relative accuracy gain of 5.44% and a maximum gain of 8.63%. Ablation and control experiments further show that these gains cannot be attributed solely to OSD or reduced update magnitude.

The main contributions of this paper are as follows:

- We investigate how parameter changes along different spectral directions affect the retention of previously learned tasks during continual fine-tuning. Using equal-norm symmetric perturbations around historical merged checkpoints, we observe that perturbations along sampled pretrained core directions induce larger average increases in old-task loss than those along residual directions. This provides empirical evidence for selectively protecting spectral directions during fine-tuning.
- We propose SCPO, which attenuates core-related gradients within a fixed pretrained core using weights derived from its overlap with the corresponding singular subspaces of the previous merged model. The filtered gradients are optimized with OSD, without modifying subsequent merging rules.
- We evaluate SCPO on three vision backbones, task sequences of varying lengths, and a Llama model. SCPO improves average accuracy over standard AdamW fine-tuning with each of five representative merging methods in the eight-task vision setting. Ablations and interventions show that these gains cannot be explained solely by low-rank optimization or smaller overall updates.

2 Related Work

2.1 Model Merging

Model merging consolidates knowledge across tasks by integrating the weights or parameter updates of task-specific models in parameter space. Early approaches mainly relied on weight averaging or linear combinations of task vectors (Wortsman et al., 2022; Ilharco et al., 2023). Subsequent work focused on mitigating interference among task updates, for example through sign conflict resolution, sparsification, and low-dimensional or spectral constraints (Yadav et al., 2023; Yu et al., 2024; Gargiulo et al., 2025; Marczak et al., 2025a; Sun et al., 2026). In continual learning, MagMax and OPCM recursively integrate sequential task updates through magnitude-based selection and orthogonal projection, respectively (Marczak et al., 2025b; Tang et al., 2025), while SAIM further coordinates the fine-tuning and merging stages to improve knowledge retention throughout the task sequence (Yang et al., 2026b). These methods primarily focus on how task updates from different stages are integrated, whereas SCPO focuses on how such updates are formed during fine-tuning.

2.2 Merge-Aware Fine-Tuning

Recent work modifies standard fine-tuning to improve model mergeability. Tangent-space fine-tuning enhances weight disentanglement through model linearization but incurs additional inference overhead (Ortiz-Jimenez et al., 2023). SAFT-Merge uses sharpness-aware optimization to obtain flatter solutions at higher training cost (Lee et al., 2025). MergOPT models merging as weight-space parameter shifts and accounts for these perturbations during fine-tuning to improve merging robustness (Yang et al., 2026a). SCPO instead examines how updates along different spectral directions affect previously learned tasks, applying direction-dependent protection to core-related gradient components.

2.3 Traditional Continual Learning

Traditional continual learning approaches mainly include regularization, knowledge distillation, experience replay, and subspace-based constraints. Regularization methods mitigate forgetting by restricting changes to parameters important for previously learned tasks; knowledge distillation uses previous models to preserve existing output behavior; and experience replay revisits a subset of previous training samples when learning new tasks (Kirkpatrick et al., 2017; Li & Hoiem, 2016; Buzzega et al., 2020). A related line of work constrains parameter updates using gradient- or representation-derived subspaces. OGD constructs protected directions from gradients of previous tasks, GPM identifies principal subspaces from network representations, and SGP further applies importance-weighted scaling to gradient components (Farajtabar et al., 2020; Saha et al., 2021; Saha & Roy, 2023). These methods derive their protection signals primarily from gradients, representations, or samples associated with previous tasks. In contrast, SCPO defines the protection scope from the spectral structure of the pretrained weights and uses the spectral information of the previous merged model to modulate the protection strength across directions within this scope.

3 Spectral Core-Preserving Optimization

Spectral Core-Preserving Optimization (SCPO) attenuates core-related gradients within a fixed pretrained spectral subspace, with protection weights determined by the previous merged model. Online Subspace Descent (OSD) then optimizes the filtered gradients. SCPO modifies only fine-tuning and leaves the subsequent merging rule unchanged.

3.1 Problem Setup

Given pretrained encoder parameters Θ_0 and a sequence of tasks $\{\mathcal{T}_t\}_{t=1}^{T}$, we consider continual learning without replaying past training examples. Each task $\mathcal{T}_t$ has a training set $\mathcal{D}_t^{\mathrm{tr}}$ and a fixed classification head h_t. At stage t, the shared encoder is initialized from the previous merged parameters Θ_{t-1} and optimized using only the current task's training data. The current-task loss is

$$\mathcal{L}_t(\Theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}_t^{\mathrm{tr}}}\left[\ell\big(h_t(f_\Theta(x)), y\big)\right]. \tag{1}$$

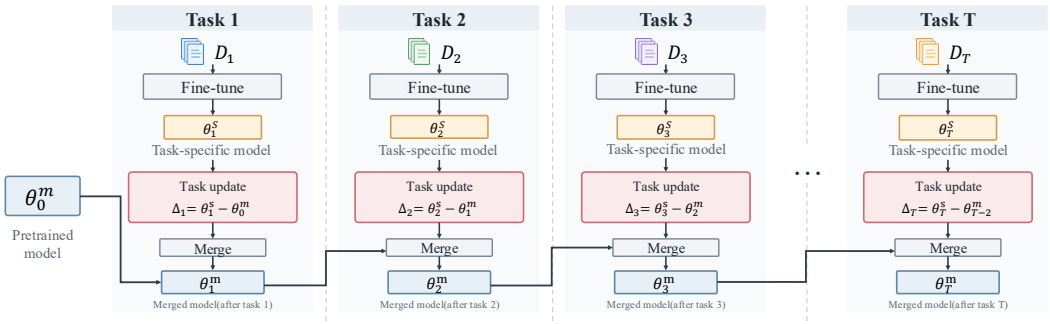

Figure 1: Continual fine-tuning and model merging. Each task contributes a local update, and the merged model initializes the next task. The task-specific and merged models correspond to Θ_t^{ft} and Θ_t, respectively.

Here, f_Θ is the shared encoder and ℓ is cross-entropy. When merging coefficients are selected using validation sets of seen tasks, we follow the merger's original protocol, as detailed in the experiments.

Let Θ_t^{ft} denote the fine-tuned parameters. We define the local task vector relative to the current initialization as

$$\tau_t = \Theta_t^{\text{ft}} - \Theta_{t-1}. \tag{2}$$

As illustrated in Figure 1, the selected merging rule combines τ_t with previous updates to obtain Θ_t, which is deployed on the seen tasks and initializes task $\mathcal{T}_{t+1}$. Below, W denotes a single matrix parameter of the encoder; we omit the layer index for clarity.

Since $\mathcal{L}_t$ does not distinguish update directions by their effects on previous tasks, SCPO applies history-weighted protection within the pretrained core before passing gradients to the optimizer.

3.2 Pretrained Core Subspace

In multitask model merging, Sun et al. (2026) link the leading singular structure of pretrained weights to cross-task generalization. We therefore use this spectral structure as a fixed protection reference throughout the task sequence.

For a pretrained weight matrix $W_0 \in \mathbb{R}^{m\times n}$ with SVD $W_0 = U\Sigma V^\top$, let $U_0 \in \mathbb{R}^{m\times k}$ and $V_0 \in \mathbb{R}^{n\times k}$ contain its top-k left and right singular vectors. The corresponding orthogonal projectors are

$$L_0 = U_0U_0^\top, \qquad R_0 = V_0V_0^\top. \tag{3}$$

The rank k, with $1 \le k \le \min(m,n)$, determines the extent of the pretrained core subspace. For any matrix $X \in \mathbb{R}^{m\times n}$, define the residual component as

$$\mathcal{Q}_0(X) = (I_m - L_0)X(I_n - R_0),$$

where I_m and I_n are identity matrices. Its complementary core component is

$$\mathcal{P}_0(X) = X - \mathcal{Q}_0(X) = L_0X + XR_0 - L_0XR_0. \tag{4}$$

Thus, $\mathcal{P}_0(X)$ contains all components associated with either the leading left or the leading right singular subspace, whereas $\mathcal{Q}_0(X)$ lies in the respective orthogonal complements on both sides. The two components are orthogonal under the Frobenius inner product. The bases U_0, V_0 and the protected subspace they define remain fixed throughout the task sequence.

3.3 History-Weighted Core Protection

Historical spectral weighting. Adaptation and merging change the alignment between pretrained and historical spectral directions. SCPO retains the pretrained core as a fixed reference and uses its overlap with the previous merged model to weight protection within this subspace.

Let W_{t-1} be the weight matrix in Θ_{t-1} corresponding to W_0, with top-k left and right singular vector matrices U_{t-1} and V_{t-1}. In the pretrained singular-vector bases, define

$$M_{u,t} = (U_0^\top U_{t-1})(U_0^\top U_{t-1})^\top, \qquad M_{v,t} = (V_0^\top V_{t-1})(V_0^\top V_{t-1})^\top. \tag{5}$$

The overlap matrices $M_{u,t}, M_{v,t} \in \mathbb{R}^{k\times k}$ are positive semidefinite, with eigenvalues equal to squared cosines of the corresponding principal angles and thus in $[0, 1]$ (Björck & Golub, 1973). Mapping these weights to the original weight space gives

$$A_t = U_0 M_{u,t} U_0^\top, \qquad B_t = V_0 M_{v,t} V_0^\top. \tag{6}$$

These operators satisfy $0 \preceq A_t \preceq L_0$ and $0 \preceq B_t \preceq R_0$. Greater spectral overlap yields a stronger core response, while weaker overlap allows more of the current-task gradient to be retained. The overlap serves as a geometric weight on spectral directions, not a direct estimate of the importance of a historical task or an individual parameter. Both operators are computed at the start of task $\mathcal{T}_t$ and held fixed during its fine-tuning.

Soft core attenuation. To apply direction-dependent protection to gradients within the pretrained core, we define the history-weighted core response operator

$$\mathcal{K}_t(X) = A_t X + X B_t - A_t X B_t. \tag{7}$$

Combining left and right subspace overlap, this operator responds more strongly to core directions aligned with the previous model and has zero response to residual directions. Its response coefficients lie in $[0, 1]$, enabling soft attenuation of core components (Appendix B.1).

Let $G_{t,s}$ denote the matrix gradient at optimization step s of task $\mathcal{T}_t$. Before this gradient enters the base optimizer, SCPO applies

$$\widetilde{G}_{t,s} = \mathcal{F}_t(G_{t,s}) = G_{t,s} - (1-\beta)\mathcal{K}_t(G_{t,s}), \qquad 0 \le \beta \le 1. \tag{8}$$

The coefficient β controls how much of the core-related gradient is retained. Setting $\beta = 1$ disables core protection; decreasing β attenuates directions with stronger core responses more heavily, while leaving gradient components in the pretrained core's orthogonal complement unchanged. SCPO thus applies continuous attenuation to core-related directions without strictly confining optimization to the core's orthogonal complement.

Under the definitions above, the gradient filter satisfies

$$\|\mathcal{F}_t(G)\|_F \le \|G\|_F, \qquad \mathcal{Q}_0(\mathcal{F}_t(G)) = \mathcal{Q}_0(G).$$

Filtering preserves the pretrained residual and cannot increase the gradient's Frobenius norm (Appendix B.1). These guarantees concern the gradient, not the final update, which also involves adaptive optimization, low-dimensional reconstruction, and weight decay.

3.4 Optimization and Continual Merging

Optimization with OSD. We supply the filtered gradients to Online Subspace Descent (OSD; Liang et al., 2024) without changing its online projection-learning or low-dimensional optimization mechanisms. The protection rank k and optimization rank r_{opt} are chosen independently, controlling the protected subspace and optimizer-state dimensions, respectively.

For a left-sided projection, let $P_{t,s} \in \mathbb{R}^{m\times r_{\text{opt}}}$ denote the OSD projection matrix at step s. The filtered gradient is compressed as

$$\widehat{G}_{t,s} = P_{t,s}^\top \widetilde{G}_{t,s}. \tag{9}$$

OSD maintains optimizer states using $\widehat{G}_{t,s}$ and maps the resulting adaptive update back through $P_{t,s}$. Online projection learning also uses $\widetilde{G}_{t,s}$, so filtering affects both parameter updates and subsequent optimization subspaces. The right-sided projection, projection objective, and optimization settings follow OSD (Appendix C.1).

We apply A_t and B_t through their low-rank factors, without explicitly constructing the full $m \times m$ or $n \times n$ response matrices. Parameters outside matrix-wise spectral protection follow the base optimizer's original update rule.

Continual merging. After fine-tuning task $\mathcal{T}_t$, we compute τ_t using Eq. (2) and obtain Θ_t with the selected merging rule, without additional core filtering.

Table 1: Continual-learning test ACC (%) and BWT (pp) across backbones and task streams. Bold and underline denote the best and second-best continual-learning results. Uncertainty denotes sample standard deviation across available runs. TA denotes Task Arithmetic; combinations are written as optimizer + merger.

Method	ViT-B/32			ViT-B/16			ViT-L/14		
	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks	8 tasks	14 tasks	20 tasks
Pre-Trained	47.5	57.4	55.6	54.2	60.9	59.2	64.5	69.3	65.6
Fine-Tuned	93.2	89.9	90.9	94.9	91.5	92.4	96.0	92.6	93.5
C. Fine-Tuned	70.9	67.9	48.7	74.7	70.4	60.2	81.4	81.6	63.1
ACC (%) ↑									
C. TA	$79.5_{\pm 0.4}$	$75.5_{\pm 1.3}$	$72.3_{\pm 0.4}$	$83.2_{\pm 0.3}$	$79.9_{\pm 0.8}$	$76.7_{\pm 0.2}$	$88.6_{\pm 0.0}$	$87.4_{\pm 0.0}$	$84.9_{\pm 0.0}$
EWC	$78.1_{\pm 0.7}$	$72.7_{\pm 0.2}$	$67.7_{\pm 0.1}$	$80.2_{\pm 1.3}$	$77.2_{\pm 0.8}$	$71.9_{\pm 0.3}$	$86.9_{\pm 0.4}$	$84.0_{\pm 0.7}$	$78.8_{\pm 1.1}$
LwF	$80.9_{\pm 0.5}$	$68.1_{\pm 1.2}$	$63.7_{\pm 1.2}$	$87.2_{\pm 1.4}$	$78.7_{\pm 1.1}$	$71.7_{\pm 0.7}$	$90.6_{\pm 0.7}$	$86.5_{\pm 0.4}$	$84.6_{\pm 0.5}$
MergOPT + TA	$80.5_{\pm 0.1}$	$76.2_{\pm 0.9}$	$72.9_{\pm 0.5}$	$83.4_{\pm 0.4}$	$80.2_{\pm 0.5}$	$76.2_{\pm 0.3}$	$88.2_{\pm 0.2}$	$86.8_{\pm 0.3}$	$84.1_{\pm 0.1}$
SAIM	$81.7_{\pm 0.7}$	$80.0_{\pm 0.3}$	$77.3_{\pm 1.1}$	$\underline{87.6}_{\pm 0.2}$	$83.1_{\pm 0.3}$	$\underline{83.2}_{\pm 0.7}$	$\underline{92.0}_{\pm 0.1}$	$89.8_{\pm 0.4}$	$\underline{90.3}_{\pm 0.5}$
SCPO + TA (Ours)	$\underline{84.3}_{\pm 0.3}$	$\underline{80.5}_{\pm 0.0}$	$\underline{78.2}_{\pm 0.1}$	$86.1_{\pm 0.4}$	$\underline{83.2}_{\pm 0.1}$	$80.7_{\pm 0.2}$	$90.7_{\pm 0.3}$	$\underline{90.1}_{\pm 0.1}$	$88.7_{\pm 0.1}$
SCPO + Iso-C (Ours)	$\mathbf{86.2}_{\pm 0.4}$	$\mathbf{82.4}_{\pm 0.5}$	$\mathbf{81.1}_{\pm 0.1}$	$\mathbf{88.6}_{\pm 0.3}$	$\mathbf{85.1}_{\pm 0.4}$	$\mathbf{84.1}_{\pm 0.3}$	$\mathbf{93.0}_{\pm 0.1}$	$\mathbf{91.8}_{\pm 0.1}$	$\mathbf{91.2}_{\pm 0.1}$
BWT (pp) ↑									
C. TA	$-8.7_{\pm 0.8}$	$-11.4_{\pm 1.7}$	$-13.7_{\pm 0.4}$	$-8.7_{\pm 0.4}$	$-9.8_{\pm 0.8}$	$-11.5_{\pm 0.2}$	$-4.7_{\pm 0.7}$	$-5.3_{\pm 0.4}$	$-7.5_{\pm 0.3}$
EWC	$-12.2_{\pm 0.9}$	$-17.5_{\pm 0.1}$	$-23.1_{\pm 0.2}$	$-12.5_{\pm 1.3}$	$-15.2_{\pm 0.2}$	$-21.0_{\pm 0.1}$	$-7.2_{\pm 0.3}$	$-9.8_{\pm 0.6}$	$-15.7_{\pm 1.1}$
LwF	$-8.7_{\pm 0.3}$	$-21.8_{\pm 1.1}$	$-26.7_{\pm 1.4}$	$-4.8_{\pm 1.2}$	$-13.2_{\pm 0.8}$	$-20.7_{\pm 1.3}$	$-3.0_{\pm 0.7}$	$-7.2_{\pm 1.1}$	$-9.4_{\pm 0.9}$
MergOPT + TA	$-8.7_{\pm 0.2}$	$-12.0_{\pm 1.2}$	$-14.3_{\pm 0.3}$	$-8.8_{\pm 0.7}$	$-10.7_{\pm 0.8}$	$-13.5_{\pm 0.2}$	$-4.6_{\pm 0.5}$	$-5.1_{\pm 0.9}$	$-7.5_{\pm 0.2}$
SAIM	$-5.3_{\pm 0.9}$	$\underline{-6.6}_{\pm 0.3}$	$\underline{-9.8}_{\pm 1.4}$	$\underline{-2.8}_{\pm 0.2}$	$-7.3_{\pm 0.5}$	$\underline{-7.7}_{\pm 0.9}$	$\underline{-0.5}$	$\underline{-2.8}$	$\underline{-2.7}$
SCPO + TA (Ours)	$\underline{-4.6}_{\pm 0.4}$	$-8.3_{\pm 0.2}$	$-11.0_{\pm 0.1}$	$-5.2_{\pm 0.4}$	$\underline{-7.1}_{\pm 0.3}$	$-10.1_{\pm 0.1}$	$-2.4_{\pm 0.2}$	$\underline{-2.8}_{\pm 0.5}$	$-4.7_{\pm 0.1}$
SCPO + Iso-C (Ours)	$\mathbf{-1.6}_{\pm 0.5}$	$\mathbf{-6.2}_{\pm 0.5}$	$\mathbf{-7.9}_{\pm 0.1}$	$\mathbf{-1.7}_{\pm 0.7}$	$\mathbf{-5.7}_{\pm 0.5}$	$\mathbf{-7.2}_{\pm 0.4}$	$\mathbf{-0.2}_{\pm 0.3}$	$\mathbf{-1.2}_{\pm 0.1}$	$\mathbf{-2.4}_{\pm 0.1}$

For example, cumulative Task Arithmetic (TA; Ilharco et al., 2023) accumulates $C_t = C_{t-1} + \tau_t$, starting from $C_0 = 0$, and constructs the stage-t merged model as

$$\Theta_t = \Theta_0 + \alpha_t C_t. \tag{10}$$

Here, α_t is the merging coefficient at the current stage. If validation sets from tasks seen so far are used to select α_t, we follow the corresponding experimental protocol. The resulting Θ_t initializes the next task and is used to construct the new history-weighted matrices at the start of task $\mathcal{T}_{t+1}$.

Because SCPO modifies only fine-tuning, it also supports TIES-Merging, Iso-C, and DARE-based merging. Appendix C.2 details their recursive updates and the complete training procedure.

4 Experiments

4.1 Continual Learning on Vision Models

We follow the FusionBench protocol (Tang et al., 2025a; Qiu et al., 2025) and evaluate 8-, 14-, and 20-task image classification streams on ViT-B/32, ViT-B/16, and ViT-L/14. Task orders and implementation details are provided in Appendix D.1.

Evaluation metrics. We use average accuracy (ACC) and backward transfer (BWT) as our primary metrics. Let $\mathcal{A}_{t,i}$ denote the deployed model's accuracy on task i after learning task t. The final average accuracy is $\mathrm{ACC}_T = \frac{1}{T}\sum_{i=1}^{T} \mathcal{A}_{T,i}$. BWT measures forgetting by comparing each earlier task's final accuracy with its accuracy immediately after learning: $\mathrm{BWT}_T = \frac{1}{T-1}\sum_{i=1}^{T-1}(\mathcal{A}_{T,i} - \mathcal{A}_{i,i})$. ACC is reported in percent and BWT in percentage points. Higher values are better for both; negative BWT indicates forgetting on average.

Main results. SCPO + TA improves ACC and BWT over standard TA in all nine settings and surpasses the existing continual-learning baselines in ACC in five (Table 1). SAIM remains stronger in some settings, including 20-task ViT-B/16. SCPO + Iso-C leads in both ACC and BWT across all nine settings, reaching 81.1% ACC on 20-task ViT-B/32 (3.8 points above SAIM). At the same endpoint, BWT improves from -9.8 to -7.9 points, showing that higher accuracy is accompanied by better historical retention. Stage-wise seen-task accuracy curves are provided in Appendix E.1.2 (Figure E.1), complementing the final ACC and BWT results.

Table 2: Final TRACEBench scores after seven tasks with Llama. Bold marks the best continual result.

Method	Task Performance (↑)							Avg. (↑)
	C-STANCE	FOMC	MeetingBank	ScienceQA	NumGLUE-cm	NumGLUE-ds	20Minuten	
Pre-Trained	0.025	0.002	0.190	0.605	0.037	0.134	0.379	0.196
Fine-Tuned	0.411	0.660	0.573	0.900	0.370	0.567	0.392	0.553
C. TA	0.364	0.494	0.452	0.445	0.173	0.514	0.392	0.405
C. TIES	0.411	0.526	0.354	0.672	0.123	0.430	0.375	0.413
C. DARE	0.340	0.269	0.308	0.132	0.160	0.502	0.383	0.299
C. MagMAX	0.305	0.460	**0.495**	0.785	0.123	0.364	0.387	0.417
EWC	0.351	0.267	0.403	0.198	0.099	**0.564**	0.398	0.326
SAIM	0.417	0.320	0.305	0.798	0.173	0.449	0.387	0.407
SCPO + TA (Ours)	**0.478**	**0.652**	0.435	**0.845**	**0.370**	0.533	**0.399**	**0.530**

4.2 Continual Learning on Large Language Models

We evaluate Llama-3.2-1B-Instruct (Meta, 2024) on TRACEBench (Wang et al., 2024) to assess SCPO in continual language-model adaptation.

Evaluation metrics. We report the final deployed model's scores on all seven tasks and their arithmetic mean. Classification tasks use accuracy; MeetingBank and 20Minuten use ROUGE-L and SARI/100, respectively. Avg. weights all seven scores equally. Implementation details are provided in Appendix D.1.

Main results. SCPO + TA achieves the highest average score among continual-learning methods (0.530 versus 0.405 for TA) and leads on five of seven tasks (Table 2). ScienceQA shows a particularly large gain, from 0.445 to 0.845. The gains are not uniform: MagMAX and EWC remain stronger on MeetingBank and NumGLUE-ds, respectively.

4.3 Compatibility with Merging and Fine-Tuning Methods

We combine SCPO with different merging methods on eight-task ViT-B/32 to evaluate its compatibility and whether its gains depend on a particular merging rule.

Main results. SCPO outperforms standard fine-tuning and MergOPT with all five merging methods (Table 3). Gains over standard fine-tuning range from 1.0 to 6.4 ACC points, with the largest gain on MagMAX (75.1% to 81.5%). The improvement with Iso-C (1.0 point) further supports compatibility beyond TA.

Table 3: Eight-task ViT-B/32 ACC (%) across merging and fine-tuning methods. Parentheses report relative gains over standard fine-tuning.

Method	Task Performance								**Avg.** (↑)
	Cars	DTD	EuroSAT	GTSRB	MNIST	RESISC45	SUN397	SVHN	
TA	59.5	50.0	89.6	90.3	98.9	84.9	67.1	93.5	79.2
MergOPT + TA	62.8	58.0	88.6	88.4	98.5	85.5	68.7	93.6	80.5 (+1.65%)
SCPO + TA	66.5	64.9	91.9	95.8	98.8	91.3	71.2	95.0	84.4 (+6.58%)
TIES-Merging	63.1	49.5	90.0	89.1	98.4	87.7	69.2	93.3	80.0
MergOPT + TIES-Merging	65.0	55.9	89.3	91.3	98.6	86.4	68.2	94.5	81.1 (+1.38%)
SCPO + TIES-Merging	68.7	66.0	91.9	96.3	98.7	90.7	71.1	94.2	84.7 (+5.82%)
DARE	61.1	55.3	87.7	90.7	98.7	85.9	68.2	93.9	80.2
MergOPT + DARE	63.4	51.1	90.7	91.2	98.8	86.2	67.8	93.5	80.3 (+0.19%)
SCPO + DARE	65.5	63.8	91.3	96.7	98.7	91.4	71.2	95.0	84.2 (+5.02%)
MagMAX	63.6	35.1	74.6	80.6	98.0	85.0	70.1	93.4	75.1
MergOPT + MagMAX	64.6	45.7	70.9	82.7	97.9	83.5	70.6	94.5	76.3 (+1.66%)
SCPO + MagMAX	66.1	52.1	84.1	94.8	98.5	89.6	72.1	94.9	81.5 (+8.63%)
Iso-C	71.5	66.5	94.9	94.0	98.6	92.8	72.8	91.3	85.3
MergOPT + Iso-C	72.9	62.2	92.2	96.6	98.4	92.5	72.5	93.6	85.1 (−0.21%)
SCPO + Iso-C	71.9	70.2	92.6	97.7	98.7	93.7	73.0	92.2	86.3 (+1.13%)

4.4 Empirical and Theoretical Analysis

To understand why SCPO is effective, we analyze two aspects: the role of core updates in historical retention, and cross-task interference as measured by weight disentanglement.

Core updates and historical retention. We directly intervene on core updates in eight-task ViT-B/32 to measure their effect on earlier tasks. We decompose the deployed model's update relative to the historical model Θ_h as $\delta = \Theta - \Theta_h = c + r$, where c is obtained using the same pretrained core projection across methods and r contains the remaining update. Varying only c, we define the intervened model and the reduction in old-task loss as

$$\Theta_\gamma^{\text{int}} = \Theta_h + r + \gamma c, \qquad R_L(\gamma) = \mathcal{L}_{\text{old}}(\Theta_1^{\text{int}}) - \mathcal{L}_{\text{old}}(\Theta_\gamma^{\text{int}}). \tag{11}$$

Here, $\Theta_1^{\text{int}} = \Theta$ is the original model and $\gamma = 0$ removes its core update. $\mathcal{L}_{\text{old}}$ averages loss equally over earlier tasks. Positive R_L indicates reduced old-task loss; negative R_L favors the original model. Figure 2 also shows old-task accuracy gains and new-task accuracy costs.

Core removal improves old-task accuracy for the baselines at a new-task cost (Figure 2). For SCPO + TA, it instead reduces old-task accuracy by 0.85 points while improving new-task accuracy by only 0.07 points. Thus, SCPO's retained core updates support historical retention in this within-model test; effective protection need not eliminate all core components.

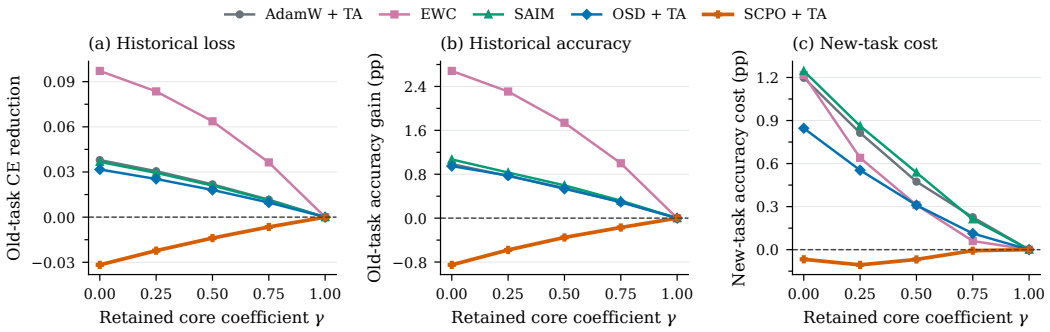

Figure 2: Core-update interventions on deployed models. $\gamma = 1$ is the original model; $\gamma = 0$ removes the core update. Left to right: old-task loss reduction, old-task accuracy gain, and new-task accuracy cost.

Layerwise norm-matched residual and signed random controls distinguish direction from magnitude (Figure 3). Before and after merging, SCPO's core removal increases old-task loss, whereas residual interventions reduce it and random effects are near zero. This core–residual reversal, absent in the baselines, cannot be explained by perturbation norm alone. The reversal is already present in the fine-tuned experts and persists after merging. This connects historical retention to the updates formed during training and supports applying protection before merging. With orthonormal reference bases and $0 \leq \beta \leq 1$, the gradient filter is non-expansive and preserves the residual component (Appendix B.1), formalizing selective protection.

Weight disentanglement and cross-task interference. We apply each local task increment τ_t, defined as the expert model minus its historical model, to the same pretrained model Θ_0 and measure how predictions change when another increment is added. Let p_t denote the predictive probability vector for task t. The pairwise disentanglement error is

$$\xi_{ij} = \sum_{t \in \{i,j\}} \mathbb{E}_{x \sim \mathcal{D}_t} \left\| p_t(x; \Theta_0 + \tau_i + \tau_j) - p_t(x; \Theta_0 + \tau_t) \right\|_1, \tag{12}$$

Lower error indicates less cross-task prediction interference. Across all 28 task pairs with both coefficients fixed at one (Figure 4), SCPO + Iso-C reduces the mean error by 38.8% and 44.4% relative to EWC and SAIM, respectively. Single-increment reference accuracy also reaches 87.68%, compared with 84.88% for EWC and 83.98% for SAIM (Table E.9). Thus, SCPO's increments combine lower cross-task interference with stronger individual-task performance.

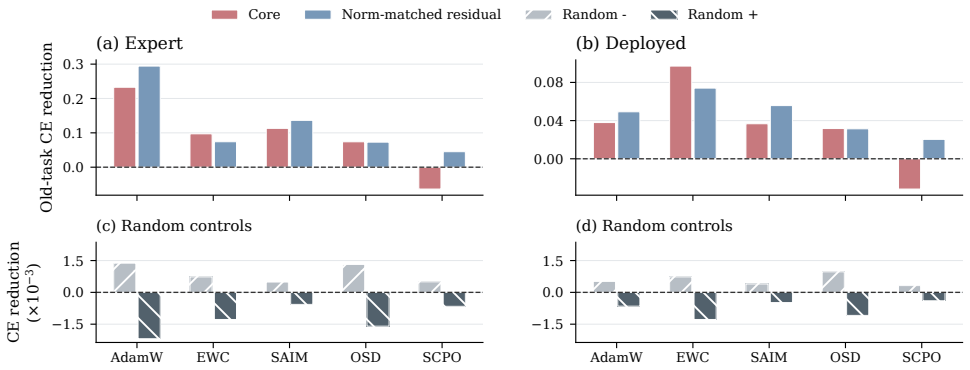

Figure 3: Direction controls before (left) and after merging (right). Top: old-task cross-entropy reduction under core and norm-matched residual interventions. Bottom: signed random controls ($\times 10^{-3}$). Positive values indicate lower loss after intervention.

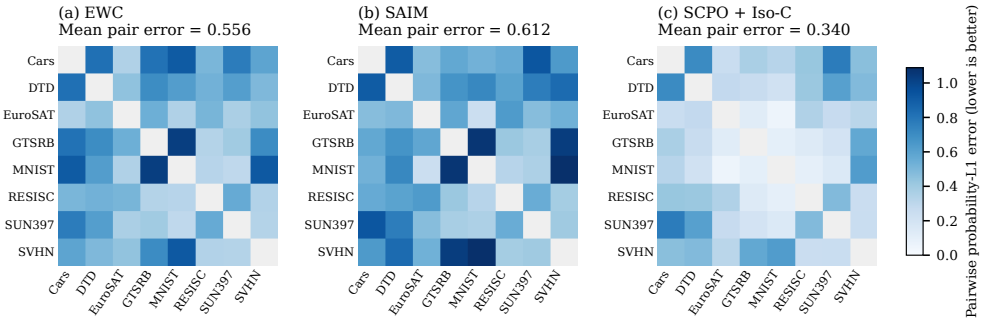

Figure 4: Pairwise weight-disentanglement error on eight-task ViT-B/32. Coefficients are fixed at 1; lighter cells indicate less interference on a shared scale; diagonals are excluded.

4.5 Ablation Studies

We isolate core protection and low-rank optimization on eight-task ViT-B/32 with TA, using the same task order and training conditions. CP-AdamW denotes AdamW with fixed pretrained core protection; SCPO combines history-weighted protection with OSD (Appendix E.2).

Main results. Core protection raises ACC by 1.30 points with AdamW and 1.12 with OSD, while improving BWT by 0.94 and 0.66 points, respectively. SCPO leads both metrics (Table 4). Gains with both optimizers support protection beyond low-rank optimization, consistent with the Iso-C comparison (Appendix E.2).

Table 4: Component ablation on eight-task ViT-B/32 with TA.

Method	Core protection	ACC (↑)	BWT (↑)
AdamW + TA	None	79.22	-9.26
CP-AdamW + TA	Enabled	80.52	-8.32
OSD + TA	None	83.30	-5.31
SCPO + TA	Enabled	**84.42**	**-4.65**

5 Conclusion

We presented SCPO, a fine-tuning method for continual learning with model merging. SCPO protects a fixed pretrained spectral core through history-weighted gradient attenuation and combines this protection with online subspace optimization, without changing the subsequent merging rule. Experiments across three vision backbones, task streams of varying lengths, and a Llama language model demonstrate improved performance and historical retention. In the eight-task vision setting, SCPO improves over standard fine-tuning with all five evaluated merging methods. Ablations and directional interventions support the contribution of core protection beyond low-rank optimization or uniform update shrinkage. Together, these findings show the value of controlling task-update directions during fine-tuning, complementing efforts to reduce interference at merging time.

Ethics Statement

This work studies how to retain learned capabilities when adapting and merging pretrained models. Improved task accuracy does not establish fairness, safety, or suitability for deployment: merged models may retain biases and harmful behaviors from pretraining or subsequent task data. Our experiments evaluate learning and retention, not these broader deployment properties. Although SCPO does not require replaying historical training examples, this does not provide a formal privacy guarantee. We recommend application-specific evaluations of bias, robustness, and harmful outputs before deployment, particularly in high-stakes settings. Use of the method should respect the licenses of source models and datasets, applicable regulations, and appropriate human oversight.

References

Åke Björck and Gene H. Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of Computation*, 27(123):579–594, 1973. doi: 10.1090/S0025-5718-1973-0348991-3.

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*, 2020.

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

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

Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodolà. Task singular vectors: Reducing task interference in model merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18695–18705, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Gargiulo_Task_Singular_Vectors_Reducing_Task_Interference_in_Model_Merging_CVPR_2025_paper.html.

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

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.

Yeoreum Lee, Jinwook Jung, and Sungyong Baik. Mitigating parameter interference in model merging via sharpness-aware fine-tuning. In *International Conference on Learning Representations*, 2025.

Zhizhong Li and Derek Hoiem. Learning without forgetting. In *European Conference on Computer Vision*, 2016.

Kaizhao Liang, Bo Liu, Lizhang Chen, and Qiang Liu. Memory-efficient LLM training with online subspace descent. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/760d09bcc06b949f5ac4b6a918739aa8-Abstract-Conference.html.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81. Association for Computational Linguistics, 2004. URL https://aclanthology.org/W04-1013/.

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. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Daniel Marczak, Simone Magistri, Sebastian Cygert, Bartlomiej Twardowski, Andrew D. Bagdanov, and Joost van de Weijer. No task left behind: Isotropic model merging with common and task-specific subspaces. In *International Conference on Machine Learning*, 2025a.

Daniel Marczak, Bartłomiej Twardowski, Tomasz Trzciński, and Sebastian Cygert. MagMax: Leveraging model merging for seamless continual learning. In *Computer Vision – ECCV 2024*, volume 15143 of *Lecture Notes in Computer Science*, pp. 379–395. Springer Nature Switzerland, 2025b. doi: 10.1007/978-3-031-73013-9_22. URL https://doi.org/10.1007/978-3-031-73013-9_22.

Meta. Llama 3.2 model card. Hugging Face model card, 2024. URL https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct. Llama-3.2-1B-Instruct; accessed 2026-09-25.

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Advances in Neural Information Processing Systems*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Gobinda Saha and Kaushik Roy. Continual learning with scaled gradient projection. In *AAAI Conference on Artificial Intelligence*, 2023.

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

Wenju Sun, Qingyong Li, Tiancheng Li, Yangli-ao Geng, and Boyang Albert Li. Revisiting the role of pretrained weights in model merging: On near-optimality within the core subspace. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/64416.

Anke Tang, Enneng Yang, Li Shen, Yong Luo, Han Hu, Lefei Zhang, Bo Du, and Dacheng Tao. Merging on the fly without retraining: A sequential approach to scalable continual model merging. In *Advances in Neural Information Processing Systems*, 2025.

Xiao Wang, Yuansen Zhang, Tianze Chen, Songyang Gao, Senjie Jin, Xianjun Yang, Zhiheng Xi, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, and Xuanjing Huang. TRACE: A comprehensive benchmark for continual learning in large language models, 2024. URL https://openreview.net/forum?id=xelrLobW0n.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23965–23998. PMLR, 2022. URL https://proceedings.mlr.press/v162/wortsman22a.html.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415, 2016. URL https://aclanthology.org/Q16-1029/.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-Merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, 2023.

Enneng Yang, Qun Yang, Peng Wang, Anke Tang, Guibing Guo, Xiaochun Cao, and Li Shen. MergOPT: A merge-aware optimizer for robust model merging. In *International Conference on Learning Representations*, 2026a.

Qun Yang, Enneng Yang, Wei Chen, Li Shen, and Long Lan. Merge to remember: Sharpness-aware isotropic merging for continual learning. In *International Conference on Machine Learning*, 2026b.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super Mario: Absorbing abilities from homologous models as a free lunch. In *International Conference on Machine Learning*, 2024.

A LLM USAGE STATEMENT

This paper utilizes the large language model (ChatGPT) solely for language polishing, spell-checking and grammatical optimisation. The model has not been used for literature reviews, scientific reasoning, conceptualisation or the formulation of proposed methods. All editing work assisted by the model has been subsequently reviewed and revised by the authors, who assume full responsibility for the content and final version of the manuscript.

B PROOFS

We establish the operator properties of spectral core protection and analyze its connection to historical-task loss.

B.1 PROPERTIES OF SPECTRAL CORE PROTECTION

For a pretrained weight matrix $W_0 \in \mathbb{R}^{m\times n}$, let U_0, V_0 contain its top-k left and right singular vectors. For a fixed task t, let $U_h = U_{t-1}$ and $V_h = V_{t-1}$ denote the corresponding bases of the previous merged model. Assume that all bases have orthonormal columns and $0 \le \beta \le 1$. Write $L_0 = U_0U_0^\top$ and $R_0 = V_0V_0^\top$, and use the Frobenius inner product $\langle X, Y\rangle_F = \mathrm{tr}(X^\top Y)$.

Orthogonal core–residual decomposition. Define

$$\mathcal{P}_0(X) = L_0X + XR_0 - L_0XR_0, \qquad \mathcal{Q}_0(X) = (I - L_0)X(I - R_0). \tag{13}$$

Left and right multiplication commute, and L_0, R_0 are self-adjoint and idempotent. Hence $\mathcal{Q}_0$ is also self-adjoint and idempotent. Its complement $\mathcal{P}_0 = I - \mathcal{Q}_0$ is therefore an orthogonal projector, giving

$$X = \mathcal{P}_0(X) + \mathcal{Q}_0(X), \quad \langle \mathcal{P}_0(X), \mathcal{Q}_0(X)\rangle_F = 0, \quad \|X\|_F^2 = \|\mathcal{P}_0(X)\|_F^2 + \|\mathcal{Q}_0(X)\|_F^2. \tag{14}$$

The core includes components aligned with *either* the leading left or the leading right singular subspace, not only their intersection L_0XR_0. Its dimension is $k(m + n - k)$; the protection rank k is thus not a direct measure of the fraction of protected parameters.

Historical overlap weights. Let

$$\begin{aligned} M_u &= (U_0^\top U_h)(U_0^\top U_h)^\top, \quad A = U_0M_uU_0^\top, \\ M_v &= (V_0^\top V_h)(V_0^\top V_h)^\top, \quad B = V_0M_vV_0^\top. \end{aligned} \tag{15}$$

Orthonormality implies $\|U_0^\top U_h\|_2 \le 1$, so $0 \preceq M_u \preceq I$ and $0 \preceq A \preceq L_0$. Similarly, $0 \preceq B \preceq R_0$. The eigenvalues of M_u, M_v are the squared cosines of the corresponding principal angles. They measure geometric overlap, not historical-task importance directly.

Proposition 1: bounded attenuation and residual preservation. Define $\mathcal{K}(X) = AX + XB - AXB$ and $\mathcal{F}(X) = X - (1 - \beta)\mathcal{K}(X)$. Then $\mathcal{F}$ is self-adjoint and positive semidefinite, and

$$\begin{aligned} \beta\|X\|_F \le \|\mathcal{F}(X)\|_F \le \|X\|_F, \\ \mathcal{Q}_0(\mathcal{F}(X)) = \mathcal{Q}_0(X), \\ \|\mathcal{P}_0(\mathcal{F}(X))\|_F \le \|\mathcal{P}_0(X)\|_F. \end{aligned} \tag{16}$$

Proof. Take complete orthonormal eigenbases $\{u_i\}$ and $\{v_j\}$ of A and B, with eigenvalues $a_i, b_j \in [0, 1]$. The matrices $u_iv_j^\top$ form an orthonormal basis of the matrix space, and

$$\mathcal{F}(u_iv_j^\top) = \big[1 - (1 - \beta)(a_i + b_j - a_ib_j)\big]u_iv_j^\top. \tag{17}$$

Since $a_i + b_j - a_ib_j = 1 - (1 - a_i)(1 - b_j) \in [0, 1]$, every eigenvalue of $\mathcal{F}$ lies in $[\beta, 1]$. Expanding X in this basis gives both norm bounds and establishes self-adjointness and positive semidefiniteness. Moreover, $A(I - L_0) = 0$ and $(I - R_0)B = 0$, which imply $\mathcal{K}\mathcal{Q}_0 = \mathcal{Q}_0\mathcal{K} = 0$. Thus the residual is unchanged, and the core is invariant under the filter. Applying the upper norm bound to $\mathcal{P}_0(X)$ yields the final inequality. □

Boundary cases and implementation. Setting $\beta = 1$ disables protection. When the historical and pretrained subspaces coincide, $\mathcal{F}(X) = \beta\mathcal{P}_0(X) + \mathcal{Q}_0(X)$. Under partial overlap, attenuation depends on direction; even $\beta = 0$ need not remove the entire core component. These properties concern the filtered gradient. Adaptive preconditioning, momentum, projection learning, and weight decay still affect the actual update. For numerical diagnostics, thin QR orthonormalizes the spectral bases while preserving their spans.

B.2 Connection to Historical-Task Interference

Let $\mathcal{L}_{\text{old}}$ denote the task-wise mean loss on previously learned tasks and Θ_h the historical model parameters. Random-direction sensitivity and interventions on actual updates address different questions.

Symmetric perturbations and directional curvature. For a unit direction d and radius $\epsilon > 0$, define

$$\kappa_\epsilon(\Theta_h, d) = \frac{\mathcal{L}_{\text{old}}(\Theta_h + \epsilon d) + \mathcal{L}_{\text{old}}(\Theta_h - \epsilon d) - 2\mathcal{L}_{\text{old}}(\Theta_h)}{\epsilon^2}. \tag{18}$$

If the loss is twice continuously differentiable in this neighborhood, the first-order terms cancel in the Taylor expansions, yielding $\kappa_\epsilon(\Theta_h, d) = d^\top \nabla^2 \mathcal{L}_{\text{old}}(\Theta_h) d + o(1)$. If the Hessian is additionally ρ-Lipschitz along both perturbation segments, each second-order remainder has magnitude at most $\rho\epsilon^3/6$. Consequently,

$$\left|\kappa_\epsilon(\Theta_h, d) - d^\top \nabla^2 \mathcal{L}_{\text{old}}(\Theta_h) d\right| \le \frac{\rho\epsilon}{3}. \tag{19}$$

This follows by bounding the two remainders and dividing by ϵ^2. The symmetric mean loss increase is $\epsilon^2\kappa_\epsilon/2$. Matching global and layer-wise perturbation norms enables a controlled comparison of local sensitivity across the sampled direction families.

Loss contribution of the actual core update. Decompose the update as $\delta = \Theta - \Theta_h = c + r$, where $c = \mathcal{P}_0(\delta)$. With $x = \Theta_h + r$ fixed, define the change in historical-task loss from adding the core component:

$$D_{\text{core}} = \mathcal{L}_{\text{old}}(x + c) - \mathcal{L}_{\text{old}}(x). \tag{20}$$

If the gradient is L-Lipschitz along the segment from x to $x + c$, then

$$|D_{\text{core}} - \langle \nabla \mathcal{L}_{\text{old}}(x), c\rangle| \le \frac{L}{2}\|c\|^2. \tag{21}$$

By the fundamental theorem of calculus, $D_{\text{core}} = \int_0^1 \langle \nabla \mathcal{L}_{\text{old}}(x + sc), c\rangle \, ds$. Subtracting the first-order term and applying Cauchy–Schwarz bounds the integrand by $Ls\|c\|^2$; integration gives the result. □

Core displacement controls the second-order remainder, while the first-order contribution depends on alignment with the historical-loss gradient. We therefore evaluate core energy together with historical-task loss, accuracy, and current-task cost. Random perturbations characterize subspace sensitivity; update interventions measure the contribution of the learned component.

C Algorithmic Details

C.1 SCPO Optimization

Algorithm C.1 gives the single-task update, using left-sided projection for clarity. The protected blocks form $\mathcal{I}$; all bases, response operators, projections, and optimizer states are block-specific. For task t, $\Theta_h = \Theta_{t-1}$; W_h, U_h, and V_h denote the corresponding weight block and its leading singular-vector bases. The protection rank k and optimization rank r_{opt} are independent. Pretrained bases are cached across tasks, whereas historical bases are recomputed from Θ_h at each task boundary. The resulting A, B remain fixed during the task and are applied through their low-rank factors.

Here $\mathcal{B}_s$ is a minibatch from $\mathcal{D}_t^{\text{tr}}$. Ω specifies the training budget N, learning rates η_s, weight decay λ_W, reconstruction scale s_{rec}, and the parameter and projection optimizer settings. $\mathrm{AdamDirection}$ updates the low-dimensional moments ξ_s and returns the bias-corrected adaptive direction, without

Algorithm C.1 SCPO: single-task fine-tuning

Require: Pretrained model Θ_0, historical model Θ_h, training data $\mathcal{D}_t^{\mathrm{tr}}$
Require: Protected blocks $\mathcal{I}$; ranks k, r_{opt}; retention β; optimizer settings Ω
Ensure: Fine-tuned expert Θ^{ft}

$\Theta \leftarrow \Theta_h$; initialize optimizer states ξ_0, ζ_0 for each block
for each block $W \in \mathcal{I}$ **do**
  $(U_0, V_0) \leftarrow \mathrm{TopSVD}_k(W_0)$; $(U_h, V_h) \leftarrow \mathrm{TopSVD}_k(W_h)$
  $M_u \leftarrow (U_0^\top U_h)(U_0^\top U_h)^\top$; $M_v \leftarrow (V_0^\top V_h)(V_0^\top V_h)^\top$
  $A \leftarrow U_0 M_u U_0^\top$; $B \leftarrow V_0 M_v V_0^\top$
end for
for $s = 0, \ldots, N-1$ **do**
  $g_s \leftarrow \mathrm{Clip}\big(\nabla_\Theta \mathcal{L}_t(\Theta; \mathcal{B}_s)\big)$
  for each block $W \in \mathcal{I}$ **do**
    $G_s \leftarrow g_s[W]$; $\widetilde{G}_s \leftarrow G_s - (1-\beta)(AG_s + G_sB - AG_sB)$
    if $s = 0$ **then**
      $P_0 \leftarrow \mathrm{InitProjection}(\widetilde{G}_0, r_{\mathrm{opt}}; \Omega)$
    end if
    $q_s \leftarrow \nabla_P \mathcal{J}_{\mathrm{left}}(P_s; \widetilde{G}_s)$
    $\widehat{G}_s \leftarrow P_s^\top \widetilde{G}_s$
    $(\widehat{D}_s, \xi_{s+1}) \leftarrow \mathrm{AdamDirection}(\widehat{G}_s, \xi_s; \Omega)$
    $D_s \leftarrow s_{\mathrm{rec}} P_s \widehat{D}_s$
    $(P_{s+1}, \zeta_{s+1}) \leftarrow \mathrm{ProjStep}(P_s, q_s, \zeta_s; \Omega)$
    $W \leftarrow (1 - \eta_s \lambda_W) W - \eta_s D_s$
  end for
  Update parameters outside $\mathcal{I}$ with AdamW using g_s
end for
return $\Theta^{\mathrm{ft}} \leftarrow \Theta$

applying a learning rate or weight decay. $\mathrm{ProjStep}$ updates P_s and its optimizer state ζ_s using the configured projection optimizer. Both compression and reconstruction use P_s; P_{s+1} is first used at the next iteration.

Compression and online projection learning. A left-sided projection uses $\widehat{G} = P^\top \widetilde{G}$ and $D = P\widehat{D}$; a right-sided projection uses $\widehat{G} = \widetilde{G}P^\top$ and $D = \widehat{D}P$. The reconstruction scale is s_{rec} in Algorithm C.1. With $Z = \widetilde{G}/(\|\widetilde{G}\|_F + \epsilon_p)$, OSD uses the reconstruction objectives

$$\begin{aligned} \mathcal{J}_{\mathrm{left}}(P) &= \|PP^\top Z - Z\|_F^2 + \lambda_p \|P^\top P - I\|_F^2, \\ \mathcal{J}_{\mathrm{right}}(P) &= \|ZP^\top P - Z\|_F^2 + \lambda_p \|PP^\top - I\|_F^2. \end{aligned} \tag{22}$$

SCPO changes the gradient supplied to this objective, not the objective itself (Liang et al., 2024). Low-dimensional moments persist within each task, and the optimizer is reinitialized for the next task.

Protected parameters and numerical precision. In vision models, protection covers trainable matrix parameters. Higher-order tensors are flattened along their first dimension; one-dimensional biases and normalization parameters are excluded. The ViT-B/32 diagnostics cover 51 visual weight matrices. For Llama, protection covers 112 attention and MLP linear-weight matrices, while the remaining parameters receive full-rank updates. Basis construction and protection operations use FP32 GPU computation. The language implementation constructs static bases with randomized low-rank SVD; the theoretical results assume orthonormal bases.

C.2 Continual Learning with Model Merging

Let Θ_t^{ft} be the expert for task t and Θ_t the merged deployment. Sequential fine-tuning starts from Θ_{t-1} and produces the local increment $\tau_t = \Theta_t^{\mathrm{ft}} - \Theta_{t-1}$. SCPO modifies only fine-tuning. The selected merger combines these local increments and constructs the deployment relative to Θ_0.

The protocol Π either fixes α_t or selects it on the validation sets prescribed by the merger. The merging state S_t follows the recurrence of the selected method, as detailed below.

Algorithm C.2 Continual merging with SCPO

Require: Pretrained model Θ_0, training datasets $\{\mathcal{D}_t^{\mathrm{tr}}\}_{t=1}^T$, merger $\mathcal{M}$
Require: SCPO settings $(\mathcal{I}, k, r_{\mathrm{opt}}, \beta, \Omega)$; merging-coefficient protocol Π
Ensure: Stage-wise deployments $\{\Theta_t\}_{t=1}^T$

$S_0 \leftarrow \mathrm{InitState}_{\mathcal{M}}()$
for $t = 1, \ldots, T$ **do**
  $\Theta_t^{\mathrm{ft}} \leftarrow \mathrm{SCPO}(\Theta_0, \Theta_{t-1}, \mathcal{D}_t^{\mathrm{tr}}; \mathcal{I}, k, r_{\mathrm{opt}}, \beta, \Omega)$ ▷ Algorithm C.1
  $\tau_t \leftarrow \Theta_t^{\mathrm{ft}} - \Theta_{t-1}$
  $S_t \leftarrow \mathrm{Update}_{\mathcal{M}}(S_{t-1}, \tau_t)$
  $\Delta_t \leftarrow \mathrm{Readout}_{\mathcal{M}}(S_t)$
  $\alpha_t \leftarrow \mathrm{MergeCoefficient}(\Pi, t, \Theta_0, \Delta_t)$
  $\Theta_t \leftarrow \Theta_0 + \alpha_t \Delta_t$
end for
return $\{\Theta_t\}_{t=1}^T$

Merging states and recurrence. Task Arithmetic stores the unmodified cumulative increment $C_t = C_{t-1} + \tau_t$ and deploys $\Theta_0 + \alpha_t C_t$ (Ilharco et al., 2023). TIES applies magnitude trimming, sign election, and sign-consistent aggregation to stored increments (Yadav et al., 2023). DARE first drops and rescales individual entries: $\widehat{\tau}_i = M_i \odot \tau_i/(1-p)$ with $M_i \sim \mathrm{Bernoulli}(1-p)$, and then applies the chosen merging rule (Yu et al., 2024). MagMAX retains the signed increment with the largest absolute magnitude at each parameter coordinate (Marczak et al., 2025b).

Accumulation and deployment in Iso-C. Iso-C also maintains $C_t = C_{t-1} + \tau_t$, but equalizes the singular values of each matrix before deployment (Marczak et al., 2025a). For a matrix block $C_t = U\Sigma V^\top$, replacing its singular values with their mean $\overline{\sigma}$ yields $\widetilde{C}_t = \overline{\sigma} U V^\top$. Non-matrix parameters use the arithmetic mean of the increments observed so far, and the model is deployed as $\Theta_t = \Theta_0 + \alpha_t \widetilde{C}_t$. Here C_t stores the accumulated updates, whereas $\widetilde{C}_t$ is used only for the current deployment. The next increment is added to C_t, avoiding repeated spectral equalization of the accumulated state. SCPO leaves this rule unchanged and applies no additional core filter at deployment.

D Experimental Details

D.1 General Setup and Baselines

D.1.1 Datasets and Task Settings

Vision experiments use CLIP ViT-B/32, ViT-B/16, and ViT-L/14 (Radford et al., 2021; Dosovitskiy et al., 2021). Table D.1 lists the fixed task order and default training budgets; the 8- and 14-task settings use the corresponding prefixes. We use the preprocessing of the corresponding CLIP backbone and fine-tune the shared visual encoder with fixed classification heads.

Table D.1: Vision task order and default training epochs. The left and right blocks list tasks 1–10 and 11–20, respectively.

No.	Dataset	Epochs	No.	Dataset	Epochs
1	Cars	35	11	FER2013	10
2	DTD	75	12	OxfordIIITPet	82
3	EuroSAT	12	13	STL10	60
4	GTSRB	11	14	CIFAR100	6
5	MNIST	5	15	CIFAR10	6
6	RESISC45	15	16	Food101	4
7	SUN397	14	17	FashionMNIST	5
8	SVHN	4	18	EMNIST	2
9	Flowers102	147	19	KMNIST	5
10	PCAM	1	20	RenderedSST2	39

Vision datasets use their prescribed validation splits where available. Otherwise, a fixed random split reserves 10% of the training set, capped at 5,000 examples, for validation.

Language experiments use Meta's Llama-3.2-1B-Instruct checkpoint (Meta, 2024) on the seven-task TRACEBench sequence (Wang et al., 2024). Table D.2 gives the task order, data sizes, and training budgets. Methods share data splits, answer-only supervision, and task-specific decoding settings, with a maximum sequence length of 2,048. Classification tasks use accuracy; MeetingBank and 20Minuten use ROUGE-L (Lin, 2004) and SARI (Xu et al., 2016), respectively. Evaluation considers only newly generated answers.

Table D.2: Language task order, sample counts after cleaning, and training budgets.

No.	Task	Train	Validation	Test	Epoch	Metric
1	C-STANCE	5000	2000	2000	5	ACC
2	FOMC	1517	379	494	3	ACC
3	MeetingBank	4999	687	691	7	ROUGE-L
4	ScienceQA	5000	2000	2000	3	ACC
5	NumGLUE-cm	282	41	81	5	ACC
6	NumGLUE-ds	1021	160	321	5	ACC
7	20Minuten	5000	200	200	7	SARI/100

D.1.2 Baseline Methods

Pre-trained evaluates the pretrained model directly. Fine-tuned denotes task-specific experts trained independently from the pretrained model and evaluated only on their respective tasks. C. Fine-tuned sequentially updates one model without additional merging. Table D.3 separates fine-tuning from deployment to distinguish their contributions.

Table D.3: Fine-tuning and deployment procedures of the main baselines.

Method	Fine-tuning	Deployment
AdamW + merging (Loshchilov & Hutter, 2019)	Supervised fine-tuning on the current task	Task Arithmetic, TIES, DARE, MagMAX, or Iso-C
EWC	Fisher-weighted parameter regularization	Direct deployment; no merger
LwF	Current-task supervision and distillation from the previous model	Direct deployment; no merger
MergOPT + TA	Merge-aware weight-robust optimization	Task Arithmetic
SAIM	SA-BCD fine-tuning	Its associated isotropic merger
CP-AdamW + TA	AdamW with dynamic historical core protection	Task Arithmetic
OSD + TA	Online projection learning and low-dimensional adaptive optimization	Task Arithmetic
SCPO	History-weighted core protection with OSD	Combined with the selected merger

EWC, LwF, MergOPT, and SAIM follow their respective implementation conventions (Kirkpatrick et al., 2017; Li & Hoiem, 2016; Yang et al., 2026a;b). Vision EWC uses a regularization coefficient of 10^6. LwF uses temperature 2 and historical-loss weight 1, normalized by the number of old tasks. SAIM uses $\rho = 0.05$ and a selection fraction of 0.3.

D.1.3 Implementation and Evaluation

Table D.4 lists the main SCPO hyperparameters; task-specific budgets appear in the dataset tables. The microbatch and accumulation settings are 128×1 for ViT-B/16 and 32×4 for ViT-L/14. Language runs use FP32 parameters and gradient checkpointing, with projection learning rate 0.005 scheduled alongside the weight learning rate. Smaller β imposes stronger protection on directions with maximal core response.

Merging-coefficient selection. Merging coefficients are selected on the validation sets, while the reported vision ACC and BWT are evaluated on the corresponding test sets. Vision Task Arithmetic and related mergers select coefficients on validation sets of seen tasks, typically using a grid from 0.1 to 3.0 in increments of 0.1. The SAIM and Iso-C grids start at 0.5. The search stops after three

Table D.4: SCPO hyperparameters. The protection and optimization ranks determine the protected subspace and optimizer-state dimension, respectively.

Setting	ViT-B/32	Llama-3.2-1B
Learning rate	10^{-5}	2×10^{-5}
Weight decay	0.1	0.01
Microbatch $\times$ accumulation	128×1	1×8
Learning-rate schedule	Cosine; 500 warmup steps	Cosine; no warmup
Core rank k	16	16
Gradient retention β	0.02	0.5
Optimization rank r_{opt}	512	128
Reconstruction scale s_{rec}	1.0	0.25
Projection optimizer	Adam	AdamW
Projection initialization	SVD	SVD
Projection weight decay	0	0.01
Reconstruction regularizer λ_p	0.1	0
Additional update orthogonalization	Disabled	Disabled
Global gradient-clipping threshold	1.0	1.0

consecutive candidates fail to improve the validation score. The main language comparison fixes the coefficient at 0.8. SCPO computes protection weights directly from pretrained and historical model parameters.

Metrics and statistical reporting. Let $\mathcal{A}_{t,i}$ denote deployment accuracy on task i after learning task t. Following Lopez-Paz & Ranzato (2017), we compute $\text{ACC}_t = t^{-1} \sum_{i \le t} \mathcal{A}_{t,i}$ and $\text{BWT}_T = (T-1)^{-1} \sum_{i<T} (\mathcal{A}_{T,i} - \mathcal{A}_{i,i})$. The diagonal entry $\mathcal{A}_{i,i}$ is the accuracy of the deployment after task i. The language average assigns equal weight to five classification accuracies, MeetingBank ROUGE-L, and 20Minuten SARI/100. We report the mean and sample standard deviation over five random seeds for the main experiments. Additional analyses follow the evaluation protocols specified in their corresponding sections and tables.

D.2 Diagnostic Evaluation Protocols

D.2.1 Spectral Sensitivity

We perturb historical OSD deployments immediately before tasks 2, 4, and 8 of the eight-task ViT-B/32 sequence. Core and residual operators cover 51 visual matrices with $k = 16$. For each probe repetition, random matrices are transformed by the corresponding operator and normalized layer-wise so that layer ℓ has direction norm $\sqrt{n_\ell / \sum_j n_j}$, where n_ℓ is its parameter count. All direction families therefore share a unit global norm and identical layer-wise norms. We compare the pre-trained core, its orthogonal complement, a random core of the same rank, and the history-weighted response. We use three independently sampled probe directions at radii $\epsilon \in \{0.05, 0.1, 0.2\}$, with both perturbation signs evaluated. Each old task uses up to 128 validation examples. Losses are averaged equally across tasks before computing Eq. (18). Error bars denote sample standard deviations across the three probe directions.

D.2.2 Core-Update Interventions and Directional Controls

At both the expert and deployed endpoints, write $\delta = \Theta - \Theta_h = c + r$. Parameters outside the protected matrices remain entirely in r. The core-scaled model is $\Theta_\gamma^{\text{int}} = \Theta_h + r + \gamma c$ for $\gamma \in \{0, 0.25, 0.5, 0.75, 1\}$. Relative to the unmodified endpoint $\Theta_1^{\text{int}} = \Theta$, we report

$$\begin{aligned} R_L(\gamma) &= \mathcal{L}_{\text{old}}(\Theta_1^{\text{int}}) - \mathcal{L}_{\text{old}}(\Theta_\gamma^{\text{int}}), \\ R_A(\gamma) &= \mathcal{A}_o(\Theta_\gamma^{\text{int}}) - \mathcal{A}_o(\Theta_1^{\text{int}}), \\ C_A(\gamma) &= \mathcal{A}_t(\Theta_1^{\text{int}}) - \mathcal{A}_t(\Theta_\gamma^{\text{int}}). \end{aligned} \tag{23}$$

Positive R_L, R_A indicate improved historical-task performance after intervention; $C_A > 0$ indicates a cost on the current task. In particular, $R_L(0) = D_{\text{core}}$. All three quantities measure *within-model* intervention effects.

For each layer, construct $\widehat{r}_\ell = r_\ell \|c_\ell\|_F / \|r_\ell\|_F$ and evaluate $\Theta_1^{\text{int}} - (1-\gamma)\widehat{r}$. Projecting Gaussian matrices onto the residual subspace and matching the same layer-wise norms gives random controls $\Theta_1^{\text{int}} \pm (1-\gamma)z$. A zero core norm gives a zero control displacement; controls requiring normalization of a zero candidate direction are undefined and excluded. The same random directions are used across intervention strengths.

Interventions cover stages 2–8 of each method's continual-learning trajectory. Old tasks are evaluated on their full test sets and weighted equally. We average across stages within each trajectory, then across training repetitions. The repetition counts are $n = 3$ for AdamW and EWC, $n = 2$ for SAIM at stages 5–6 and $n = 3$ elsewhere, and $n = 1$ for OSD and SCPO.

D.2.3 Weight Disentanglement

To fix the comparison reference, we apply each local expert increment $\tau_i = \Theta_i^{\text{ft}} - \Theta_{i-1}$ to the same pretrained model Θ_0. Let $p_i(x; \Theta)$ denote the class-probability vector for task i, and define

$$\begin{aligned} \xi_{i \leftarrow j} &= \mathbb{E}_{x \sim \mathcal{D}_i} \| p_i(x; \Theta_0 + \tau_i + \tau_j) - p_i(x; \Theta_0 + \tau_i) \|_1, \\ \xi_{ij} &= \xi_{i \leftarrow j} + \xi_{j \leftarrow i}, \qquad \overline{\xi} = \binom{T}{2}^{-1} \sum_{i<j} \xi_{ij}. \end{aligned} \tag{24}$$

Both increment coefficients are fixed at 1, giving 28 unordered task pairs and 56 directed comparisons for eight tasks. Each task uses a fixed subset of up to 512 validation examples; heads, preprocessing, and examples remain fixed. Directed errors lie in $[0, 2]$ and pairwise errors in $[0, 4]$. Reference accuracy averages task i accuracy at $\Theta_0 + \tau_i$ over tasks. This fixed-coefficient parameter-addition diagnostic characterizes interference between local task increments.

D.2.4 Shared-History and Update-Norm Matching

Before tasks 2, 4, and 8, OSD and SCPO branch from the same historical model with matched sample order and training budget in three paired repetitions. The respective budgets are 1,050, 2,068, and 2,136 steps, with $k = 16$ and $\beta = 0.02$. Fixed-coefficient evaluations use $\alpha = 0.9, 0.7, 0.6$ at these stages and up to 512 test examples per task. To separate update direction from magnitude, we rescale the actual OSD increment to match SCPO's full-parameter norm and compare seen-task average accuracy after deployment.

E Additional Results

Performance tables bold the best mean within each comparison group, including ties. In diagnostic tables, boldface instead highlights curvature or intervention magnitude as specified in the caption.

E.1 Detailed Performance and Benchmark Evaluations

E.1.1 Per-Task Performance

Table E.1 reports final per-task deployment accuracy after 20 tasks for all three backbones. Avg. is the unweighted mean over all 20 tasks; TA denotes Task Arithmetic.

Table E.1: Per-task test accuracy (%) after learning 20 tasks. SCPO rows are shaded; the best and second-best Avg. within each backbone are bold and underlined, respectively.

Method	Cars	DTD	EuroSAT	GTSRB	MNIST	RESISC45	SUN397	SVHN	Flowers102	PCAM	FER2013	OxfordIIITPet	STL10	CIFAR100	CIFAR10	Food101	FashionMNIST	EMNIST	KMNIST	RenderedSST2	Avg.
ViT-B/32																					
C. TA	53.7	43.6	65.9	57.8	89.1	69.3	64.1	63.3	67.6	85.8	58.0	82.9	95.8	71.1	93.8	70.0	81.5	96.1	74.0	70.7	72.70
EWC	37.6	47.9	45.9	48.6	56.7	63.8	59.8	54.8	64.7	88.7	51.4	75.0	89.8	55.9	85.2	72.0	86.5	99.5	99.4	72.0	67.75
LwF	13.4	28.7	42.4	56.8	77.5	32.3	32.7	80.6	40.2	53.0	56.0	59.2	87.6	69.8	88.4	65.1	93.7	99.6	98.9	71.7	62.38
MergOPT + TA	54.8	46.3	66.7	58.1	89.4	70.9	65.3	59.3	72.5	83.7	58.0	83.4	96.0	71.7	93.9	72.4	82.6	97.1	75.2	68.6	73.30
SAIM	51.1	71.8	78.6	82.4	90.8	84.7	67.5	60.5	63.7	86.7	62.6	81.5	96.8	69.9	91.3	77.0	78.3	88.0	69.9	70.8	76.20
SCPO + TA (Ours)	55.7	59.0	76.5	73.3	86.6	82.9	67.6	69.0	70.6	84.0	57.0	85.3	96.6	74.6	95.5	79.8	85.4	99.1	95.7	69.9	<u>78.21</u>
SCPO + Iso-C (Ours)	53.6	64.9	82.5	83.3	87.3	89.2	71.3	67.7	72.5	88.6	62.9	89.4	96.2	81.1	96.9	81.6	89.7	98.6	94.3	69.9	**81.08**
ViT-B/16																					
C. TA	62.2	48.9	73.6	66.9	94.8	76.0	68.5	72.2	71.6	79.6	56.5	89.7	96.4	76.9	96.0	80.0	85.8	94.5	77.9	65.9	76.69
EWC	48.3	49.5	42.9	57.1	70.7	65.6	62.5	72.0	58.8	87.3	59.2	83.7	91.6	58.8	87.5	81.4	85.6	99.6	99.5	76.9	71.93
LwF	40.2	39.4	42.3	65.6	87.3	55.9	52.2	86.2	49.0	60.0	65.9	77.2	94.6	77.3	95.0	74.9	94.0	99.7	99.3	77.9	71.70
MergOPT + TA	63.1	48.4	64.1	69.2	93.0	73.2	69.1	69.2	70.6	82.6	55.9	90.5	97.0	76.4	95.9	79.5	85.9	94.8	76.2	70.1	76.23
SAIM	70.8	79.3	92.2	90.1	94.2	89.3	72.1	70.4	72.5	83.3	62.0	90.8	97.6	79.4	95.4	86.6	86.8	92.4	92.4	78.3	<u>83.79</u>
SCPO + TA (Ours)	65.1	59.0	72.9	76.8	88.7	85.5	70.3	76.0	73.5	84.5	59.5	90.2	96.8	76.1	96.0	84.3	88.7	98.9	95.7	74.7	80.66
SCPO + Iso-C (Ours)	60.2	65.4	77.3	88.1	88.7	89.7	74.9	76.5	78.4	90.3	65.8	91.6	96.6	82.9	97.0	88.1	93.1	98.7	96.1	78.5	**83.90**
ViT-L/14																					
C. TA	80.7	61.2	89.3	81.9	97.0	88.6	73.5	75.9	79.4	85.1	67.5	94.3	99.0	86.1	98.2	88.0	89.4	96.5	86.0	79.9	84.87
EWC	64.0	60.6	50.6	62.6	82.8	69.4	70.4	83.2	70.6	86.3	64.7	91.3	94.4	69.9	93.1	86.2	92.1	99.6	99.6	83.4	78.75
LwF	68.3	59.6	76.5	87.8	98.9	80.3	67.1	92.0	71.6	77.8	74.0	90.0	98.4	88.0	98.0	85.9	95.1	99.8	99.3	83.5	84.59
MergOPT + TA	77.5	61.2	82.2	79.7	96.3	87.1	73.7	78.1	80.4	84.2	68.2	94.0	98.6	85.3	98.1	87.2	88.6	95.8	87.7	77.6	84.08
SAIM	75.3	85.1	97.7	98.3	97.1	95.5	77.8	84.0	92.2	93.4	73.2	95.4	99.2	88.4	98.5	91.6	90.1	95.8	95.7	81.8	<u>90.30</u>
SCPO + TA (Ours)	85.4	69.7	92.1	91.3	91.7	92.9	75.1	83.7	88.2	90.8	71.6	95.1	99.0	86.6	98.5	91.3	91.2	99.6	98.6	81.5	88.70
SCPO + Iso-C (Ours)	83.9	79.3	95.4	96.8	94.6	95.8	78.5	87.8	91.2	93.9	74.5	96.2	98.8	90.3	98.9	92.3	93.5	99.4	98.2	83.7	**91.15**

SCPO + Iso-C achieves mean ACC of 81.08%, 83.90%, and 91.15% across the three backbones, but its advantage varies by task. For example, SAIM performs better on Cars and DTD with ViT-B/16, whereas SCPO + Iso-C is stronger on PCAM, CIFAR100, and Food101.

Table E.2: Final language scores (higher is better). Best scores per task are bold; Δ is SCPO + TA minus AdamW + TA.

Task	AdamW + TA	SCPO + TA	SAIM	Δ vs. AdamW
C-STANCE	0.3645	**0.4785**	0.4170	+0.1140
FOMC	0.4939	**0.6518**	0.3198	+0.1579
MeetingBank	**0.4518**	0.4345	0.3055	-0.0173
ScienceQA	0.4445	**0.8450**	0.7980	+0.4005
NumGLUE-cm	0.1728	**0.3704**	0.1728	+0.1975
NumGLUE-ds	0.5140	**0.5327**	0.4486	+0.0187
20Minuten	0.3920	**0.3995**	0.3874	+0.0075

Table E.2 provides the corresponding language breakdown. SCPO improves over AdamW + TA on six tasks, with larger gains on ScienceQA and FOMC and a small decrease on MeetingBank.

E.1.2 Stage-Wise Accuracy and Historical Retention

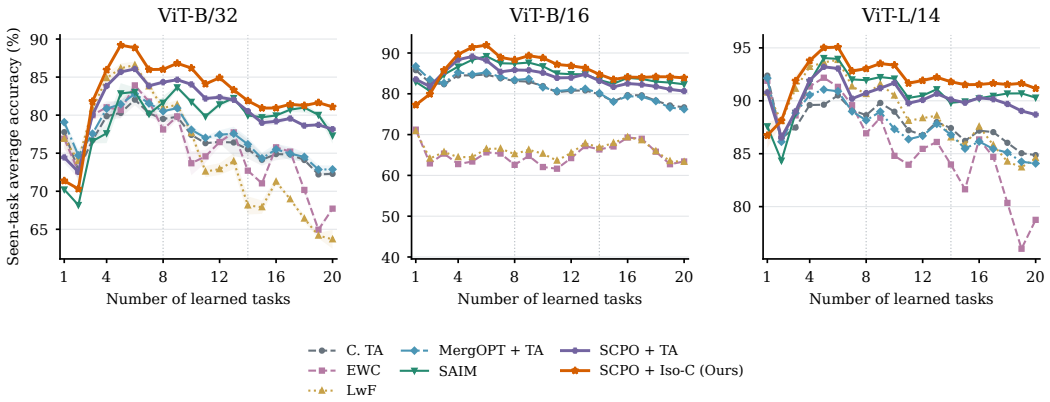

Figure E.1: Seen-task average test ACC over 20 stages. Left to right: ViT-B/32, ViT-B/16, and ViT-L/14.

Figure E.1 complements the final results with seen-task average accuracy over 20 stages. SCPO + Iso-C maintains strong performance through the later stages across all three backbones. These curves reflect both new-task learning and historical retention; BWT and per-task matrices further characterize forgetting.

Figure E.2 compares EWC, SAIM, and SCPO + Iso-C over the 20-task ViT-B/32 sequence. Row t evaluates seen tasks after learning task t. The last-row mean gives final ACC, while its differences from the diagonal determine BWT. Gray cells mask tasks that have not yet been learned.

E.2 Ablation Studies and Training Analysis

E.2.1 Core Protection and Low-Rank Optimization

Table E.3 compares combinations of core protection and low-rank optimization on eight-task ViT-B/32 with TA under a common task order.

Table E.3: Component ablation: test ACC and BWT with TA as the common merger.

Method	Core protection	β	ACC (%)	BWT (pp)
AdamW	None	–	79.22	−9.26
CP-AdamW	Dynamic historical core	0.02	80.52	−8.32
OSD	None	–	83.30	−5.31
SCPO	History-weighted	0.02	**84.42**	**−4.65**

As shown in Table E.3, dynamic historical core protection improves AdamW from 79.22% to 80.52% ACC and from -9.26 to -8.32 pp BWT, corresponding to gains of 1.30 and 0.94 percentage points, respectively. SCPO further improves OSD from 83.30% to 84.42% ACC and from -5.31 to -4.65 pp BWT, yielding gains of 1.12 and 0.66 points. These results show that core protection provides additional benefits under both full-rank AdamW and low-dimensional OSD optimization. CP-AdamW defines its protected subspace directly from the previous merged model, whereas SCPO retains a fixed pretrained core and uses historical spectral overlap to modulate attenuation within it.

Core protection beyond OSD. Table E.4 evaluates core protection beyond OSD on the eight-task ViT-B/32 test sets under TA and Iso-C. We compute test ACC and BWT with fixed checkpoints and merging coefficients.

E.2.2 Comparison with Tangent-Space Fine-Tuning

We compare SCPO with tangent-space fine-tuning under five merging rules on eight-task ViT-B/32. Each method uses its respective training budget.

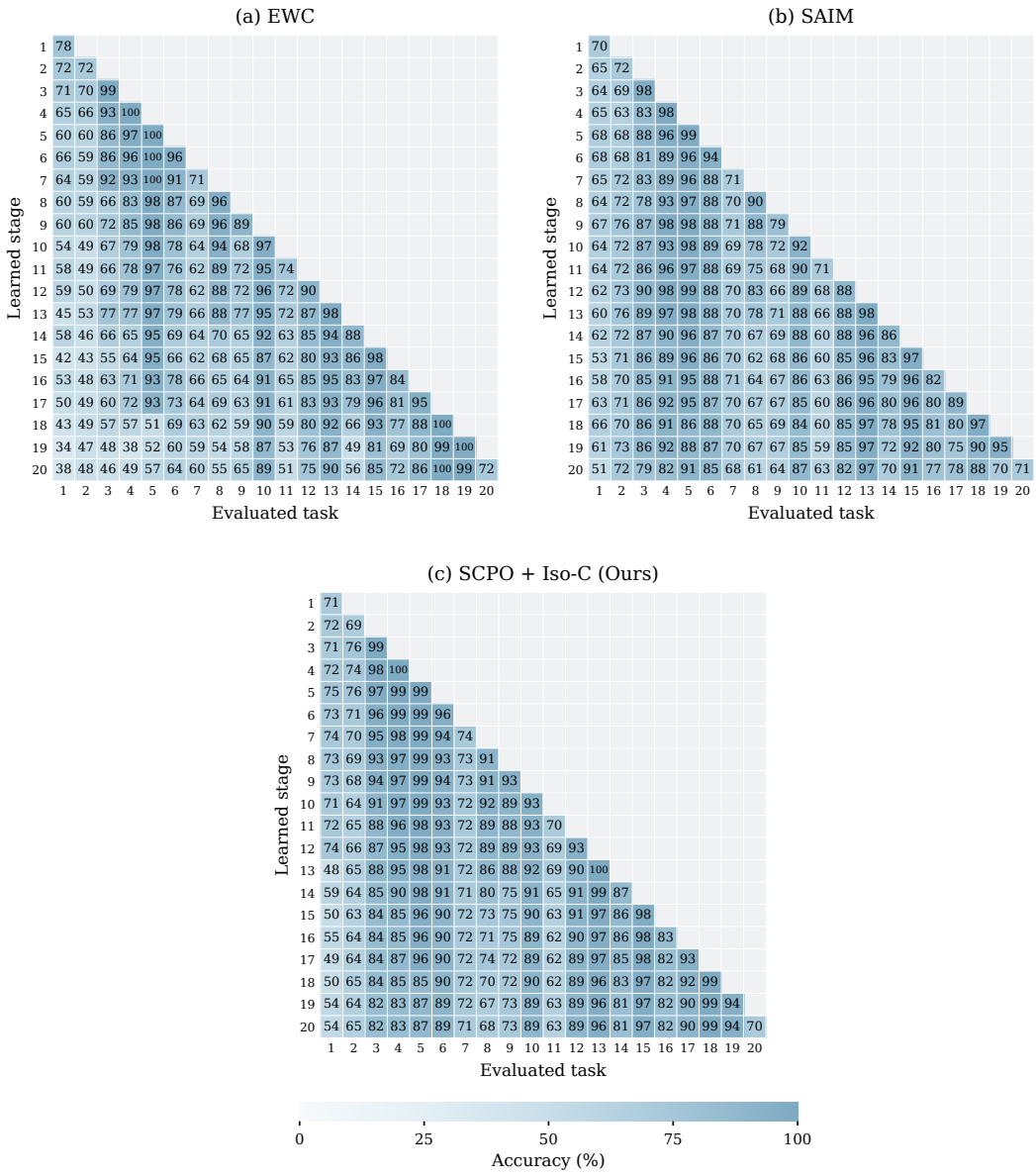

Figure E.2: Per-task test accuracy throughout the 20-task ViT-B/32 sequence. Cells show accuracy (%) rounded to integers; colors use unrounded values on a shared 0–100% scale. Gray cells denote unseen tasks; task indices follow Table D.1.

SCPO + Iso-C retains more early-task performance than EWC in later stages; differences from SAIM vary by task, consistent with Table E.1.

E.3 Mechanistic Analysis and Computational Cost

E.3.1 Spectral Sensitivity Around Historical Models

Figure E.3 and Table E.6 test whether the pretrained spectral core remains functionally sensitive after continual adaptation.

The weaker random-core response indicates that subspace dimension alone does not explain the observed sensitivity. History-weighted and pretrained-core responses are numerically close, indicating similar local sensitivity for these two direction families.

Table E.4: Core protection beyond OSD on eight-task ViT-B/32: test ACC (%) and BWT (pp), both higher is better. ± denotes sample standard deviation. SCPO rows are shaded, and the best mean within each merging rule is bold.

Method	Test ACC (↑)	Test BWT (↑)
AdamW + TA	78.429	−10.340
CP-AdamW + TA	80.124	−8.608
OSD + TA	82.791	−6.150
SCPO + TA	**83.918**	**−5.156**
OSD + Iso-C	85.599 ± 0.202	−2.086 ± 0.211
SCPO + Iso-C	**85.982** ± 0.137	**−1.510** ± 0.444

SCPO improves over OSD under both merging rules. With TA, core protection raises test ACC by 1.13 points and BWT by 0.99 points. With Iso-C, the mean gains are 0.38 and 0.58 points, respectively, with positive ACC and BWT differences in both paired runs. Together, these results support an incremental contribution from training-time core protection beyond low-rank optimization, extending across two merging rules.

Table E.5: Tangent and SCPO accuracy (%) on eight-task ViT-B/32.

Fine-tuning	Task Arithmetic	TIES	DARE	MagMAX	Iso-C
Tangent	82.93	82.13	78.70	78.43	84.44
SCPO	**84.4**	**84.7**	**84.2**	**81.5**	**86.3**

SCPO achieves higher ACC than tangent-space fine-tuning with all five mergers, with gains of approximately 1.5–5.5 percentage points (Table E.5). Training budgets differ between the methods, so these results compare their respective training configurations.

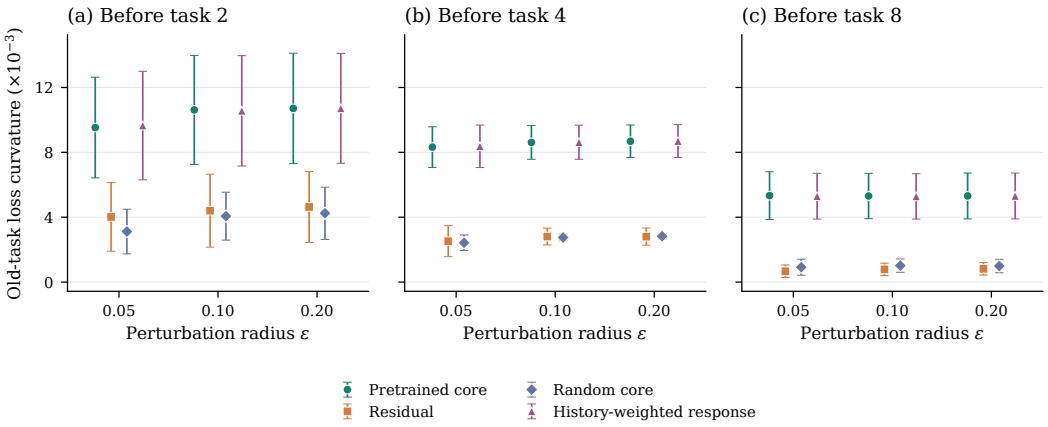

Figure E.3: Equal-norm symmetric perturbations around historical checkpoints. Error bars show sample standard deviation across three probe directions.

Table E.6: Directional curvature at $\epsilon = 0.1$, in units of 10^{-3}. Bold marks the largest mean at each checkpoint, indicating greater sensitivity.

Before task	Pretrained core	Residual	Random core	History-weighted
2	$\mathbf{10.616 \pm 3.362}$	4.403 ± 2.249	4.063 ± 1.478	10.561 ± 3.402
4	8.612 ± 1.040	2.806 ± 0.518	2.750 ± 0.199	$\mathbf{8.625 \pm 1.052}$
8	$\mathbf{5.308 \pm 1.397}$	0.783 ± 0.384	1.012 ± 0.410	5.284 ± 1.400

Sampled core directions have higher mean symmetric curvature than residual directions in all nine stage–radius comparisons. At $\epsilon = 0.1$, the ratios are 2.41, 3.07, and 6.78. These results support the pretrained core as a protection reference after continual adaptation.

E.3.2 Extended Core-Update Interventions

Figure E.4 compares core scaling with layer-wise norm-matched residual interventions.

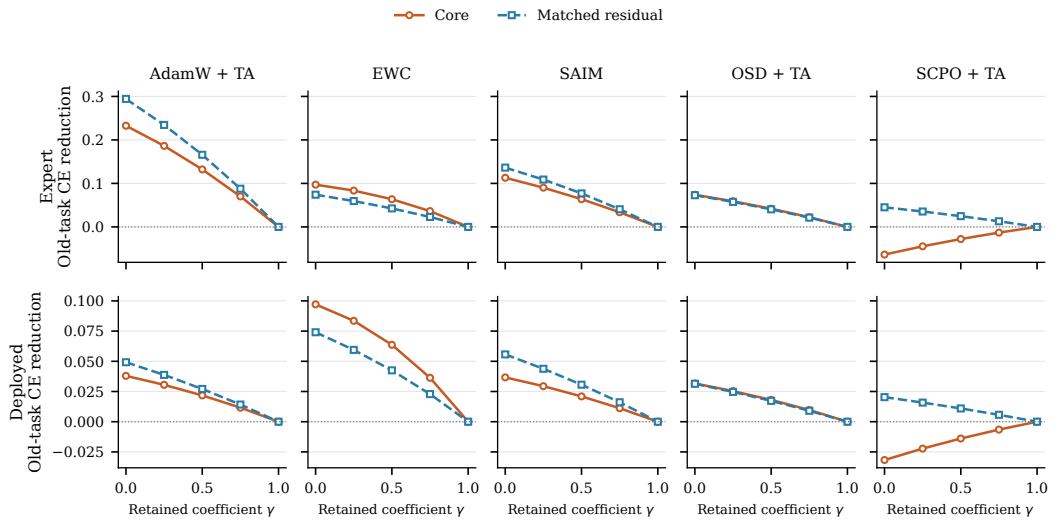

Figure E.4: Core and layer-wise norm-matched residual interventions. Top: experts; bottom: deployments. Axes share a vertical scale within each row; positive values indicate lower old-task loss after intervention.

Table E.7: Within-model effects of core removal; R_A, C_A are in percentage points. Bold marks the largest absolute mean response for each endpoint and metric.

Method	Endpoint	Units	$R_L(0)$	$R_A(0)$ (pp)	$C_A(0)$ (pp)
AdamW + TA	Expert	21	**0.2327**	**+5.47**	+0.05
AdamW + TA	Deployed	21	0.0379	+0.99	+1.20
EWC	Expert	21	0.0971	+2.68	**+1.22**
EWC	Deployed	21	**0.0971**	**+2.68**	+1.22
SAIM	Expert	19	0.1130	+3.04	+0.24
SAIM	Deployed	19	0.0367	+1.07	**+1.25**
OSD + TA	Expert	7	0.0739	+2.02	+0.03
OSD + TA	Deployed	7	0.0317	+0.95	+0.85
SCPO + TA	Expert	7	-0.0636	-1.57	+0.30
SCPO + TA	Deployed	7	-0.0317	-0.85	-0.07

At deployed endpoints, removing the core update improves old-task accuracy by 0.99, 2.68, 1.07, and 0.95 percentage points for AdamW, EWC, SAIM, and OSD, respectively, but reduces SCPO accuracy by 0.85 points (Table E.7). Core removal also increases SCPO's old-task loss at the expert endpoint. SCPO's retained core component therefore supports historical-task performance at both expert and deployed endpoints.

For SCPO, norm-matched residual interventions change old-task loss in the opposite direction to core removal, while random signed perturbations have effects near zero. The observations are thus inconsistent with an explanation based solely on perturbation magnitude.

E.3.3 Weight Disentanglement and Update Structure

Table E.9 and Figure E.5 summarize all 28 task pairs.

Shared-history experiments further quantify the actual parameter displacement. Define the core energy ratio as $\mathrm{CER} = \sum_\ell \|c_\ell\|_F^2 / \sum_\ell \|\delta_\ell\|_F^2$, where the denominator includes only protected matrices; the full-parameter update norm includes all updated parameters.

Table E.8: Stage-wise old-task loss reduction $R_L(0)$ under core removal. Bold marks the largest absolute mean per row; repetition counts and aggregation follow Appendix D.2.2.

Endpoint	Stage	AdamW + TA	EWC	SAIM	OSD + TA	SCPO + TA
Expert	2	0.0687 ± 0.0099	$\mathbf{0.1543 \pm 0.0309}$	0.1277 ± 0.0211	0.0795	-0.0801
Expert	3	$\mathbf{0.1855 \pm 0.0238}$	0.0330 ± 0.0020	0.0280 ± 0.0013	0.0425	-0.0178
Expert	4	$\mathbf{0.3944 \pm 0.0415}$	0.1714 ± 0.0161	0.2923 ± 0.0316	0.0955	-0.1256
Expert	5	$\mathbf{0.4749 \pm 0.0758}$	0.1128 ± 0.0059	0.0672 ± 0.0089	0.0780	-0.0485
Expert	6	$\mathbf{0.1845 \pm 0.0405}$	0.0344 ± 0.0138	0.0667 ± 0.0024	0.0944	-0.0760
Expert	7	$\mathbf{0.0673 \pm 0.0221}$	0.0327 ± 0.0063	0.0603 ± 0.0039	0.0522	-0.0450
Expert	8	$\mathbf{0.2536 \pm 0.0410}$	0.1413 ± 0.0127	0.1091 ± 0.0063	0.0753	-0.0520
Deployed	2	0.0352 ± 0.0093	$\mathbf{0.1543 \pm 0.0309}$	0.0647 ± 0.0054	0.1013	-0.1066
Deployed	3	$\mathbf{0.0596 \pm 0.0146}$	0.0330 ± 0.0020	0.0085 ± 0.0081	0.0185	-0.0077
Deployed	4	0.0532 ± 0.0184	$\mathbf{0.1714 \pm 0.0161}$	0.0541 ± 0.0155	0.0234	-0.0346
Deployed	5	0.0511 ± 0.0061	$\mathbf{0.1128 \pm 0.0059}$	0.0219 ± 0.0055	0.0308	-0.0196
Deployed	6	0.0268 ± 0.0188	0.0344 ± 0.0138	$\mathbf{0.0402 \pm 0.0077}$	0.0159	-0.0248
Deployed	7	0.0069 ± 0.0150	0.0327 ± 0.0063	$\mathbf{0.0401 \pm 0.0033}$	0.0116	-0.0117
Deployed	8	0.0328 ± 0.0179	$\mathbf{0.1413 \pm 0.0127}$	0.0208 ± 0.0033	0.0203	-0.0170

Table E.8 reports stage-wise results to expose variation behind the aggregate.

Table E.9: Weight disentanglement: lower error and higher reference ACC are better. Reference ACC evaluates $\Theta_0 + \tau_i$ on task i, averaged over tasks.

Method	Pairwise mean ↓	Directed mean ↓	Median ↓	Maximum ↓	Reference ACC (%) ↑
EWC	0.5557	0.2778	0.5161	1.0285	84.88
SAIM	0.6118	0.3059	0.5656	1.0884	83.98
SCPO + Iso-C	**0.3403**	**0.1702**	**0.2968**	**0.7671**	**87.68**

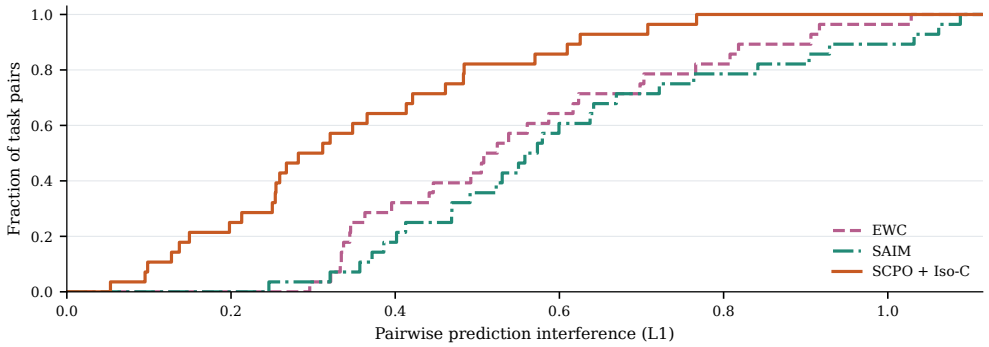

Figure E.5: Empirical cumulative distributions of interference over 28 task pairs. At a fixed threshold, a higher curve indicates more pairs with error below that threshold.

SCPO + Iso-C has mean pairwise error 0.3403, below EWC's 0.5557 and SAIM's 0.6118, together with higher single-increment reference accuracy. The empirical cumulative distribution shows the fraction of task pairs below each interference threshold, complementing the mean error. Together with ACC and BWT, these measurements characterize predictive interference and sequential retention.

Table E.10: Update structure from shared histories: mean ± sample standard deviation over three branch seeds. Bold marks higher ACC or lower core energy and fraction within each stage; total energy is not ranked.

Stage	Method	New-task ACC (%) ↑	CER (%)	$\|c\|^2$	$\|\delta\|^2$
2	OSD	73.83 ± 0.20	7.27 ± 0.09	0.081 ± 0.006	1.144 ± 0.091
2	SCPO	$\mathbf{74.61 \pm 0.20}$	$\mathbf{4.73 \pm 0.24}$	$\mathbf{0.057 \pm 0.002}$	1.234 ± 0.064
4	OSD	$\mathbf{97.27 \pm 0.20}$	5.67 ± 0.03	0.123 ± 0.001	2.279 ± 0.015
4	SCPO	97.07 ± 0.20	$\mathbf{3.26 \pm 0.10}$	$\mathbf{0.074 \pm 0.002}$	2.392 ± 0.021
8	OSD	96.94 ± 0.30	6.68 ± 0.06	0.122 ± 0.002	1.938 ± 0.013
8	SCPO	$\mathbf{97.20 \pm 0.11}$	$\mathbf{2.03 \pm 0.04}$	$\mathbf{0.038 \pm 0.001}$	1.995 ± 0.012

Table E.10 shows that SCPO reduces both absolute core energy and its fraction without reducing total update energy. New-task expert accuracy is higher at stages 2 and 8, with a 0.20-point decrease at stage 4, consistent with direction-selective rather than uniform suppression of adaptation.

Table E.11: Deployment ACC (%) from shared histories with identical merging coefficients. Columns correspond to adaptation at stages 2, 4, and 8 from the shared historical checkpoints defined in Appendix D.2.4.

Method	Stage 2	Stage 4	Stage 8
SCPO	73.535 ± 0.763	$\mathbf{82.389 \pm 0.294}$	$\mathbf{83.040 \pm 0.190}$
OSD	72.005 ± 0.628	82.243 ± 0.278	82.593 ± 0.152
OSD + post-hoc filtering	72.591 ± 0.739	82.259 ± 0.185	83.000 ± 0.147
OSD + global norm matching	72.038 ± 0.564	82.243 ± 0.246	82.528 ± 0.051
OSD + blockwise norm matching	72.396 ± 0.795	82.178 ± 0.224	82.821 ± 0.099
OSD + cumulative CSP	$\mathbf{73.991 \pm 0.203}$	82.194 ± 0.454	82.723 ± 0.249

After matching OSD's total update norm to SCPO's, SCPO retains higher deployment ACC at all three stages (Table E.11), supporting a contribution from update direction beyond magnitude. Among the compared variants, SCPO has the highest ACC at stages 4 and 8, while cumulative CSP leads at stage 2. Post-hoc filtering approaches SCPO at the later stages, highlighting the stage-dependent behavior of the protection strategies.

E.3.4 Computational Cost

For an $m \times n$ matrix, protection bases and overlap weights require $O((m+n)k + k^2)$ storage, while the leading matrix multiplications cost $O(mnk)$. OSD maintains moments in compressed coordinates, but model parameters remain full-sized. Memory savings depend on the optimization rank and parameter shapes.

Table E.12: Resource costs of seven-task language runs; column minima are bold. Peak memory is the maximum allocated memory across tasks.

Method	Fine-tuning time (h) ↓	Peak allocated (GiB) ↓	Tasks
SCPO + TA	8.59	**18.00**	7
AdamW + TA	**6.68**	23.51	7
SAIM	7.94	38.28	7

Table E.12 reports seven-task language runs on RTX A6000 GPUs. SCPO's peak allocated training memory is 18.00 GiB, compared with 23.51 GiB for AdamW + TA, while cumulative fine-tuning time increases from 6.68 to 8.59 hours. Timing includes loading, basis construction, and saving within the fine-tuning routine, but excludes outer-loop merging and evaluation. Timings were measured under shared-GPU execution and summarize end-to-end fine-tuning cost for these configurations.