MED-PRISM: SPECTRALLY ANISOTROPIC MEMORY FOR CONTINUAL MEDICAL MULTIMODAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Medical multimodal models are expected to continually learn skills with different prediction requirements, from diagnostic labels to spatial localization and image-based reasoning. Retaining earlier skills is difficult when past data cannot be revisited. Freezing past adapters is not enough because new updates can still change earlier predictions. A past parameter update can be strong along some directions and weak along others. We propose Med-PRISM, a replay-free continual learning framework that allows shared adjustment while giving stronger protection to high-energy historical directions. The shared adapter learns under a drift constraint. Private adapters are frozen after each skill and reused together. Rather than treating all historical directions equally, we use the spectrum of each past private update to record its directions and their relative strengths. It discourages new private updates from using high-energy historical directions while leaving lower-energy directions more plasticity. No past samples or activations are stored, and inference requires no task-specific routing. We derive a bound on changes in earlier-task losses and identify the interference term. Experiments on MedicalSkill-CL, a benchmark we construct for continual learning across five medical skills, show that Med-PRISM surpasses the strongest evaluated baseline by 1.94 and 2.71 points in terms of Avg and Last, respectively.

1 INTRODUCTION

In clinical practice, the same medical image may need to answer very different questions depending on what a clinician needs to know: what finding is present, which diagnosis it supports, where an abnormality lies, or how evidence across images should be integrated. A generalist medical multimodal model should therefore support these complementary capabilities through a unified interface (Li et al., 2023; Tu et al., 2024). We use *skill* to denote a distinct medical capability characterized by what information the model should extract or reason about. As new clinical skills emerge, such models are expected to continually acquire new skills without forgetting previously learned capabilities (Fig. 1a).

Access to data from previously learned skills is often infeasible in medical settings. Privacy, ethical requirements, and data governance can restrict how medical imaging data are stored, shared, and reused (Kaissis et al., 2020; Rieke et al., 2020). Avoiding raw-image storage does not necessarily eliminate privacy concerns, since gradients and intermediate features may still reveal information about the original inputs (Zhu et al., 2019; Mahendran & Vedaldi, 2015). These constraints motivate continual learning settings in which each new skill is learned using only the currently available data, without retaining or replaying information derived from previous samples.

Parameter-efficient continual learning accumulates task-specific updates while keeping most pre-trained parameters fixed. Freezing earlier adapters preserves their parameters, but new updates can still change earlier predictions. Prior work reduces interference by discouraging new updates from using directions already used by earlier updates (Wang et al., 2023). However, medical skills share visual understanding while requiring different ways to extract and express evidence. Recent studies likewise suggest that low-rank updates can contain both task-shared and task-specific directions (He et al., 2026; Liao et al., 2026). A past update can make large changes along some directions and much smaller changes along others. Applying the same constraint to all of these directions ignores this difference. The spectrum of a learned weight update records its directions and their relative

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

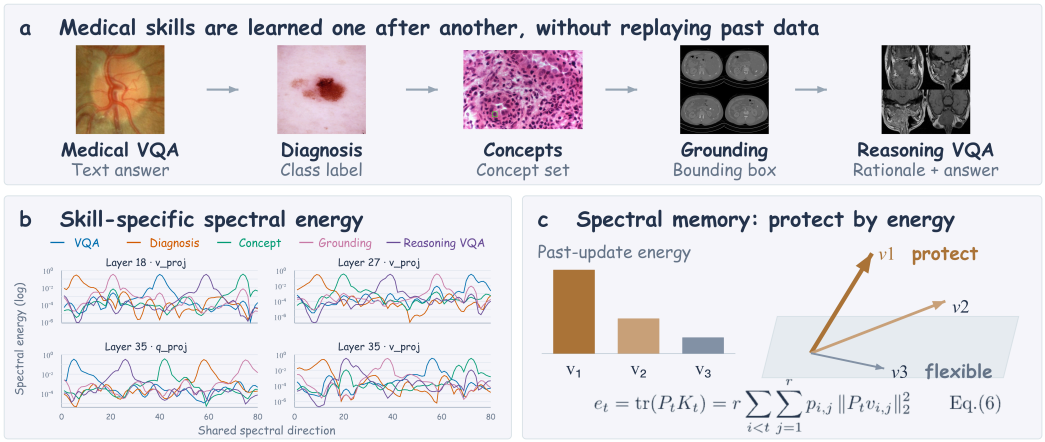


Figure 1: **Motivation for replay-free medical skill accumulation.** **(a)** Medical skills are learned sequentially without replaying past data. **(b)** Private updates learned during Med-PRISM's continual training show low-energy overlap and different high-energy peaks across skills in a common spectral basis within each module. **(c)** SAMC discourages new private updates from using high-energy historical directions while leaving lower-energy directions less constrained.

strengths (Tian et al., 2026). This motivates *using historical spectral energy to apply stronger constraints along high-energy directions while leaving lower-energy directions more flexible*.

We propose **Med-PRISM**, a replay-free framework centered on **Spectrally Anisotropic Memory Consolidation (SAMC)** (Fig. 1c). SAMC computes the spectrum of each historical private update and records its directions and relative energies. During subsequent learning, it discourages new private updates from using high-energy historical directions. Lower-energy directions receive weaker constraints, leaving more room for new learning. Private LoRA adapters are frozen after each skill and reused together. A shared LoRA adapter remains trainable, with a drift constraint limiting its changes across stages. The spectral memory is derived from learned parameters, requires no stored samples or activations, and is used only during training. At inference, the same cumulative model handles all learned skills without task IDs or adapter routing. Consistent with this design, the private updates learned during Med-PRISM's continual training show low-energy overlap and different high-energy peaks across skills (Fig. 1b).

Our contributions are threefold. **First**, we develop SAMC to build spectral memory directly from historical private updates. It uses spectral energies to constrain new private updates more strongly along high-energy historical directions while leaving lower-energy directions more flexible. **Second**, under smoothness assumptions, we derive a bound on earlier-task loss changes and identify the first-order interference term controlled by SAMC. We further establish conditions under which spectral weighting gives a tighter bound on this term than uniform weighting. **Third**, we construct MedicalSkill-CL, a benchmark for continual learning across five medical skills. Experiments with Qwen3-VL-8B show that Med-PRISM surpasses the strongest evaluated baseline by 1.94 and 2.71 points in Avg and Last, respectively. Further analyses examine how historical spectral energy relates to old-task sensitivity and how SAMC guides new private updates.

2 Related Work

2.1 Medical Multimodal Models

Medical multimodal instruction tuning provides a shared interface for image-conditioned responses. LLaVA-Med learns biomedical conversational behavior from image–text supervision (Li et al., 2023), while Med-PaLM M studies a common model across diverse biomedical tasks (Tu et al., 2024). These efforts motivate models that combine medical knowledge with multiple prediction capabilities, rather than maintaining an unrelated specialist for every output format. Learning a predefined collection of capabilities does not by itself address their subsequent expansion. CoIN stud-

ies continual multimodal instruction tuning and identifies instruction-alignment degradation as one source of forgetting (Chen et al., 2024). Our focus is heterogeneous medical tasks under sequential supervision: answer generation, diagnosis, concept sets, localization, and multi-image reasoning.

2.2 Continual Learning under Data-Access Constraints

Continual learning strategies differ in both how they protect earlier behavior and what historical information they retain (van de Ven et al., 2022). Experience replay revisits stored examples (Chaudhry et al., 2019); regularization methods such as EWC penalize changes using previously estimated parameter importance (Kirkpatrick et al., 2017); parameter isolation preserves task-specific components (Wang et al., 2023). These are distinct information dependencies: retaining a statistic or a frozen module is not equivalent to replaying an individual example. Gradient Projection Memory (GPM) extracts historical activation subspaces and projects later gradients away from them (Saha et al., 2021); SGP introduces importance-scaled updates within protected subspaces (Saha & Roy, 2023). These approaches construct memory from historical data measurements. SAMC instead derives its spectral memory directly from frozen effective parameter updates, without requiring additional measurements on historical data.

2.3 Parameter-Efficient Continual Adaptation

LoRA adapts pretrained models through low-rank weight updates while keeping the backbone frozen (Hu et al., 2022). In continual learning, these adapters can be organized and combined in different ways. MoELoRA combines low-rank experts through input-dependent gating (Chen et al., 2024). Another line of work controls interference by constraining update directions. O-LoRA regularizes overlap between task-specific LoRA factor subspaces (Wang et al., 2023), while ASO-LoRA uses attribution-based task similarity to adjust soft orthogonality constraints (Wang et al., 2026). InfLoRA constructs an input reduction subspace using current-task activations and retained historical subspace information (Liang & Li, 2024). Octopus adopts two-stage training, first adapting to the current task and then applying an orthogonality constraint between the new update and gradients computed at historical model parameters using current-task data (Liu et al., 2026).

SAMC derives its spectral memory directly from frozen effective private updates. Rather than allocating adapter capacity as in AdaLoRA (Zhang et al., 2023), it uses normalized historical spectral energies to weight constraints on new private input subspaces. Overlap with high-energy historical directions incurs stronger penalties, while lower-energy directions retain greater flexibility. These spectral summaries guide subsequent training without modifying the historical adapters. Our analysis further connects this spectrally weighted overlap to a local bound on old-task loss drift.

3 Med-PRISM

3.1 Problem Setup and Design Overview

We consider a sequence of medical multimodal tasks $\mathcal{T}_1, \ldots, \mathcal{T}_T$, where each task corresponds to one medical skill. At stage t, the model is trained only on the current-task dataset $\mathcal{D}_t$, whose examples contain one or more medical images, an instruction, and a task-specific target sequence. Task boundaries are available during training to allocate task-specific parameters, whereas inference uses no oracle task identity or task-dependent adapter selection.

Med-PRISM combines a shared adaptation pathway, task-specific private LoRA updates, and **Spectrally Anisotropic Memory Consolidation (SAMC)** (Fig. 2). The shared pathway remains adaptable across tasks. SAMC summarizes each learned private update through its input-side spectral directions and relative energies, which are then used to constrain subsequent private adaptation.

3.2 Cumulative Shared–Private Adaptation

All formulations below are applied layer-wise unless an explicit sum over adapted layers is shown; layer indices are omitted throughout. Let $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ denote the frozen base weight, S_t the shared update after learning task t, and D_i the private update associated with task i. We use a

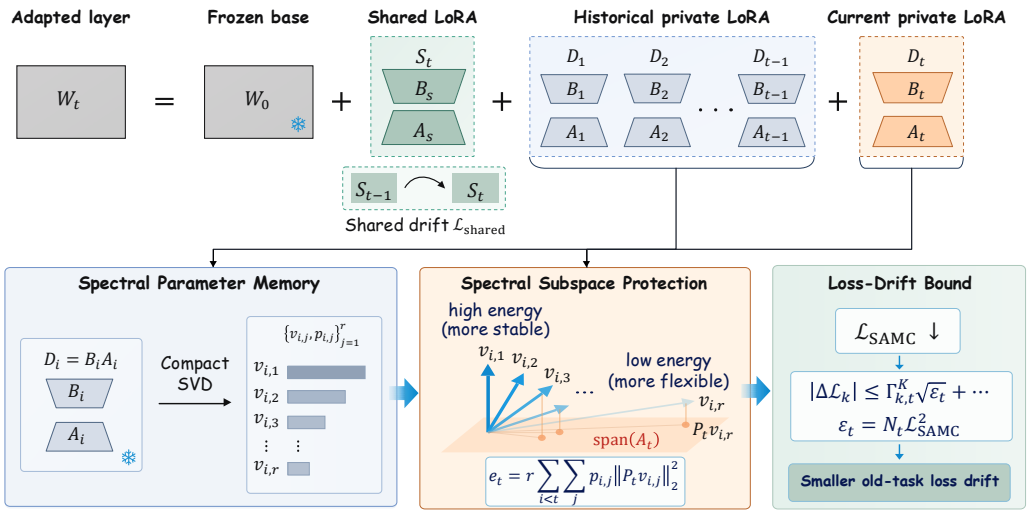


Figure 2: **Med-PRISM and its spectral consolidation mechanism.** Shared and current private adapters learn while historical private banks remain frozen and cumulatively active. Historical effective updates induce a nonuniform input-side spectral metric that penalizes current key-subspace occupancy. This energy bounds the history-weighted magnitude of the realized private update and enters a local old-task loss-drift certificate; the remaining terms account for uncovered sensitivity, shared drift, and nonlinearity (Eq. (17)).

fixed private rank r and shared rank r_{sh}, with $A_i \in \mathbb{R}^{r \times d_{\text{in}}}$, $B_i \in \mathbb{R}^{d_{\text{out}} \times r}$, $A_t^{\text{sh}} \in \mathbb{R}^{r_{\text{sh}} \times d_{\text{in}}}$, and $B_t^{\text{sh}} \in \mathbb{R}^{d_{\text{out}} \times r_{\text{sh}}}$. The cumulative forward map is

$$\mathbf{h}' = \left(W_0 + S_t + \sum_{i=1}^{t} D_i \right) \mathbf{h}, \qquad S_t = B_t^{\text{sh}} A_t^{\text{sh}}, \qquad D_i = B_i A_i. \tag{1}$$

Branch scaling is absorbed into the corresponding effective update.

At stage t, only the shared update S_t and the current private update D_t are optimized, while all previous private updates $\{D_i\}_{i<t}$ remain frozen and active in the cumulative model. At each adapted layer, define the total adaptation and its endpoint change as

$$\Theta_t = S_t + \sum_{i=1}^{t} D_i, \qquad \Theta_t - \Theta_{t-1} = D_t + \Delta S_t, \qquad \Delta S_t = S_t - S_{t-1}. \tag{2}$$

Stacking the effective weights $W_0 + \Theta_t$ across adapted layers gives the full-model endpoint $\boldsymbol{\theta}^t$ used in the theoretical analysis. Continual adaptation therefore changes the model through both the newly introduced private update and the evolving shared pathway.

3.3 Spectrally Anisotropic Consolidation and Training Objective

After learning task i, SAMC constructs a spectral memory directly from its deterministic effective private update D_i. For a nonzero update, let

$$D_i = U_i \Sigma_i V_i^\top, \qquad \Sigma_i = \operatorname{diag}(\sigma_{i,1}, \ldots, \sigma_{i,r}), \qquad p_{i,j} = \frac{\sigma_{i,j}^2}{\sum_{j=1}^{r} \sigma_{i,j}^2}, \tag{3}$$

where $\sigma_{i,j} \ge 0$ is the j-th singular value and $\mathbf{p}_i = (p_{i,1}, \ldots, p_{i,r})^\top$ contains the relative within-bank spectral energies. Zero singular values are retained when D_i is rank deficient. We define

$$H_i = \frac{D_i^\top D_i}{\|D_i\|_F^2} = V_i \operatorname{diag}(\mathbf{p}_i) V_i^\top, \qquad K_t = r \sum_{i<t} H_i, \tag{4}$$

with the convention $H_i = 0$ when $D_i = 0$. Thus, each nonzero H_i is a unit-trace positive-semidefinite summary of one historical private update, while K_t accumulates these summaries in the same fixed-rank scaling used by SAMC. The rank r restores the scale of the SAMC overlap block. Directions with larger relative energy within a historical bank receive greater weight.

For the current private factor $A_t \in \mathbb{R}^{r\times d_{\text{in}}}$, assumed to have full row rank, SAMC represents its input subspace by the orthogonal row-space projector

$$P_t = A_t^\top \left(A_t A_t^\top\right)^{-1} A_t. \tag{5}$$

Because $D_t = B_t A_t$, we have $\mathrm{row}(D_t) \subseteq \mathrm{row}(A_t)$ and hence $D_t P_t = D_t$. The SAMC spectral energy is

$$e_t = \mathrm{tr}(P_t K_t) = r \sum_{i<t} \sum_{j=1}^{r} p_{i,j} \, \|P_t \mathbf{v}_{i,j}\|_2^2, \tag{6}$$

where $\mathbf{v}_{i,j}$ is the j-th right singular vector of D_i. The second form makes the anisotropic weighting explicit: overlap with a historical direction is weighted by its relative within-bank spectral energy rather than treated uniformly.

Let $\mathcal{W}$ denote the set of M adapted layers and define the global spectral energy

$$\mathcal{E}_t := \sum_{\mathcal{W}} e_t, \qquad N_t := Mr^2(t-1), \qquad t > 1. \tag{7}$$

Equivalently, if $\bar{e}_t = M^{-1} \sum_{\mathcal{W}} e_t$ denotes the mean layer energy, then the SAMC regularizer is

$$\mathcal{L}_{\text{SAMC}} = \sqrt{\frac{\mathcal{E}_t}{N_t}} = \sqrt{\frac{\bar{e}_t}{r^2(t-1)}}. \tag{8}$$

We set $\mathcal{L}_{\text{SAMC}} = 0$ for the first task, where no historical memory exists. Thus, for $t > 1$, $\mathcal{E}_t = N_t \mathcal{L}_{\text{SAMC}}^2$ exactly under the definitions above.

The overall training objective is

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_{\text{SAMC}} \mathcal{L}_{\text{SAMC}} + \lambda_{\text{sh}} \mathcal{L}_{\text{shared}}, \tag{9}$$

where $\mathcal{L}_{\text{task}}$ is the autoregressive cross-entropy loss. To limit changes in the shared pathway, we regularize its effective update against the task-start value S_t^0:

$$\mathcal{L}_{\text{shared}} = \frac{\sum_{\mathcal{W}} \|S_t - S_t^0\|_F^2}{\max\{\sum_{\mathcal{W}} \|S_t^0\|_F^2, \epsilon_{\text{sh}}\}}, \tag{10}$$

where $\epsilon_{\text{sh}} = 10^{-8}$ is a numerical stabilizer used only for the shared-path normalization. The coefficients λ_{SAMC} and λ_{sh} are specified in the experimental setup.

3.4 Spectral Geometry and Old-Task Loss Drift

We next connect the SAMC overlap energy in Eq. (6) to the magnitude of the realized private update and, in turn, to the change in an old-task loss. Complete proofs are provided in Appendix A.

Proposition 1 (History-induced spectral geometry)**.** *Define the spectral seminorm*

$$\|X\|_{K_t} := \|X K_t^{1/2}\|_F. \tag{11}$$

For a full-row-rank current factor A_t, the private update satisfies $D_t = D_t P_t$ and

$$D_t^\top D_t \preceq \|D_t\|_2^2 P_t, \qquad \|D_t\|_{K_t} \le \|D_t\|_2 \sqrt{e_t}. \tag{12}$$

Proposition 1 bounds the history-weighted magnitude of the realized update by the SAMC overlap energy. To relate this geometry to forgetting, let G_k denote the old-task gradient block with respect to the adapted weight at a given layer, evaluated at the pre-update endpoint $\boldsymbol{\theta}^{t-1}$, and define the projector onto the positive support of K_t as

$$R_t = K_t K_t^\dagger, \qquad G_k^K = G_k R_t. \tag{13}$$

Without introducing explicit layer indices, aggregate the corresponding layer-wise quantities over the adapted-layer set $\mathcal{W}$ as

$$\rho_t^2 = \sum_{\mathcal{W}} \|D_t\|_F^2, \qquad d_t^2 = \sum_{\mathcal{W}} \|\Delta S_t\|_F^2, \qquad g_k^2 = \sum_{\mathcal{W}} \|G_k\|_F^2, \tag{14}$$

and

$$\left(\Gamma_{k,t}^K\right)^2 = \sum_{\mathcal{W}} \left\| G_k^K K_t^{\dagger/2} \right\|_F^2 \|D_t\|_2^2, \qquad \delta_{k,K}^2 = \sum_{\mathcal{W}} \|G_k(I - R_t)\|_F^2 . \tag{15}$$

These old-task gradient quantities are used only for the analysis and are not required by SAMC during training. Recall from Eq. (7) that, for $t > 1$,

$$N_t = Mr^2(t-1), \qquad \mathcal{E}_t = N_t \mathcal{L}_{\mathrm{SAMC}}^2. \tag{16}$$

Theorem 1 (Local spectral loss-drift bound)**.** *Assume frozen historical private updates, full-row-rank current factors, and a β_k-Lipschitz gradient of the old-task loss along the parameter segment from $\boldsymbol{\theta}^{t-1}$ to $\boldsymbol{\theta}^t$. Then, for $t > 1$,*

$$\begin{aligned} \left|\mathcal{L}_k(\boldsymbol{\theta}^t) - \mathcal{L}_k(\boldsymbol{\theta}^{t-1})\right| \le \Gamma_{k,t}^K \sqrt{N_t \mathcal{L}_{\mathrm{SAMC}}^2} \\ + \delta_{k,K}\rho_t + g_k d_t + \frac{\beta_k}{2}(\rho_t + d_t)^2. \end{aligned} \tag{17}$$

The bound separates four sources of old-task loss drift. The first term captures interference within the historical spectral support and is controlled by the SAMC overlap energy. The second accounts for old-task sensitivity outside that support, the third captures drift in the shared pathway, and the final term is the local smoothness remainder. The result therefore links the SAMC objective to a protected first-order component of old-task loss drift without requiring old-task gradients during training.

Role of spectral weighting. We next answer why SAMC retains the non-uniform spectrum of K_t. For a single layer, we apply an eigendecomposition to K_t and write its positive spectrum as

$$K_t = \sum_{j=1}^{q_t} \lambda_j \mathbf{u}_j \mathbf{u}_j^\top, \qquad \lambda_j > 0, \tag{18}$$

where $q_t = \mathrm{rank}(K_t)$ and $\{\mathbf{u}_j\}_{j=1}^{q_t}$ form an orthonormal basis of its support. Then we define

$$s_j = \|G_k \mathbf{u}_j\|_2^2, \qquad a_j = \|P_t \mathbf{u}_j\|_2^2. \tag{19}$$

Here, s_j measures the old-task first-order sensitivity along historical direction $\mathbf{u}_j$, whereas a_j measures the occupancy of that direction by the current key subspace. Under this eigenbasis, the protected first-order certificate becomes

$$B_{\mathrm{spec}} = \|D_t\|_2 \sqrt{\left(\sum_j \frac{s_j}{\lambda_j}\right)\left(\sum_j \lambda_j a_j\right)}. \tag{20}$$

By Cauchy–Schwarz, the certificate is minimized for $\lambda_j^\star \propto \sqrt{s_j/a_j}$. Hence, the favorable spectral geometry places large spectral weights on historical directions that are important to old tasks (high s_j) but little occupied by the current task (low a_j). So spectral weighting yields a tighter certificate than uniform support weighting only when the historical spectrum is suitably aligned with old-task sensitivity. We examine the resulting sensitivity–occupancy geometry in Sec. 4.4.

4 Experiments

4.1 Experimental Setup

Medical skill-incremental benchmark. We evaluate Med-PRISM on **MedicalSkill-CL**, assembled from existing medical datasets through format harmonization and stratified random sampling. The

Table 1: Results on MedicalSkill-CL (0–100; tasks in sequence order). Bold/underline mark best/second-best scores within each evaluation block. Med-PRISM uses $\lambda_{\mathrm{SAMC}} = 0.10$. Metric definitions are in Sec. 4.1.

	Method	Medical VQA	Diagnosis	Concept	Grounding	Reasoning VQA	Average
	Zero-shot	63.60	46.20	7.96	0.47	10.98	25.84
Avg	Sequential	84.26	**49.30**	17.72	23.99	52.40	45.53
	O-LoRA (Wang et al., 2023)	**88.48**	44.65	25.56	21.24	48.38	45.66
	MoE-LoRA (Chen et al., 2024)	82.36	<u>49.00</u>	14.95	23.34	<u>52.81</u>	44.49
	Octopus (Liu et al., 2026)	<u>87.86</u>	46.40	<u>26.11</u>	<u>40.00</u>	51.78	<u>50.43</u>
	Med-PRISM (ours)	87.42	48.75	**29.24**	**42.99**	**53.45**	**52.37**
Last	Sequential	75.20	44.70	13.29	0.89	52.40	37.30
	O-LoRA (Wang et al., 2023)	**88.50**	39.70	<u>25.06</u>	19.74	48.38	44.28
	MoE-LoRA (Chen et al., 2024)	71.50	**46.80**	9.22	0.40	<u>52.81</u>	36.15
	Octopus (Liu et al., 2026)	85.40	43.80	20.46	<u>38.90</u>	51.78	<u>48.07</u>
	Med-PRISM (ours)	<u>86.70</u>	<u>46.10</u>	**25.20**	**42.43**	**53.45**	**50.78**

tasks follow a fixed sequence: medical VQA, diagnosis, concept recognition, visual grounding, and reasoning VQA. Medical VQA, diagnosis, concept recognition, and visual grounding each contain 10,000 training and 1,000 test records, while reasoning VQA contains 7,347 training and 720 test records, yielding 47,347 training and 4,720 test records in total. These counts are reported before sequence-length filtering. The tasks use different forms of supervision and single- or multi-image inputs, with task-specific targets preserved in a shared conversational format. Exact-image and source-aware lineage checks control training overlap. Dataset construction and task-level counts are detailed in Appendix B (Table 4).

Implementation and baselines. All trained methods adapt query/value projections of frozen Qwen3-VL-8B-Instruct without replay. Baselines are Zero-shot, rank-48 Sequential LoRA, O-LoRA (Wang et al., 2023), MoE-LoRA (Chen et al., 2024), and Octopus (Liu et al., 2026). Med-PRISM uses shared/private ranks $32/16$, learning rates $10^{-5}/10^{-4}$, one epoch per task, and $(\lambda_{\mathrm{SAMC}}, \lambda_{\mathrm{sh}}) = (0.10, 0.01)$. Runs use seed 42, BF16, AdamW, and effective batch size 16; Implementation and unequal training/storage budgets are detailed in Appendices C, D, and F.

Evaluation metrics. We test all seen tasks after each stage without oracle adapter selection. For $T = 5$, let $R_{t,i} \in [0, 1]$ be task i's primary score after stage t: accuracy (Medical VQA/diagnosis), all-concept micro-F1 (Concept), $(\mathrm{Acc@IoU}_{\geq 0.5} + \mathrm{mIoU})/2$ (Grounding), or $(\mathrm{AnswerAcc} + \mathrm{ROUGE\text{-}L}_{\mathrm{reasoning}})/2$ (Reasoning VQA). We report $\mathrm{Avg}_i = \frac{1}{T-i+1}\sum_{t=i}^{T} R_{t,i}$, $\mathrm{Last}_i = R_{T,i}$, their unweighted task means, and $\mathrm{BWT}_5 = \frac{1}{4}\sum_{i=1}^{4}(R_{5,i} - R_{i,i})$ when acquisition scores are available. Higher is better; all scores use a 0–100 display scale. Decoding and aggregation details are in Appendices C.3–C.4.

4.2 Main Results

Table 1 shows that Med-PRISM attains the highest aggregate *Avg*/*Last* (52.37/50.78), exceeding Octopus by 1.94/2.71 points and Sequential by 6.84/13.48 points. It leads Concept, Grounding, and Reasoning VQA under both metrics; final Grounding reaches 42.43 versus Sequential's 0.89, and final Reasoning VQA reaches 53.45. Gains are skill-dependent: O-LoRA performs better on final Medical VQA, while Med-PRISM's final Diagnosis score (46.10) remains below MoE-LoRA (46.80) and Zero-shot (46.20).

4.3 Ablation Study

Task-order robustness. Fig. 3(a) compares the original sequence with four alternative task orders. Medical VQA and Reasoning VQA change little across orders, while Diagnosis shows the largest variation. Concept and Grounding also vary with the task order. Despite these task-level differences, none of the tested alternatives produces a large drop in overall *Avg* or *Last* relative to the original

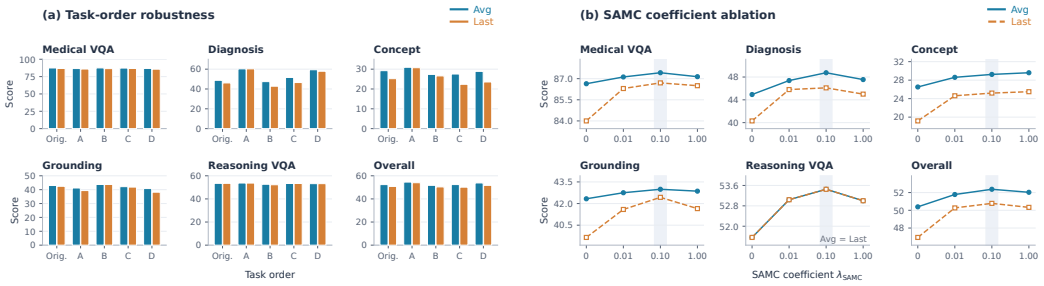


Figure 3: **Task-order robustness and SAMC coefficient ablation.** (a) Results under five task orders: Original (1→2→3→4→5), A (1→4→5→3→2), B (2→3→1→5→4), C (3→2→4→1→5), and D (4→1→3→2→5), where task IDs 1–5 denote Medical VQA, Diagnosis, Concept, Grounding, and Reasoning VQA, respectively. (b) Results with SAMC disabled and with three nonzero coefficients. Both panels report task-wise *Avg* and *Last* scores and their overall means. The shaded band in (b) marks $\lambda_{\text{SAMC}} = 0.10$, used in the main results.

sequence. The overall result therefore does not rely on a single task order, although individual skills remain sensitive to when they are learned.

SAMC and component ablations. Fig. 3(b) and Table 2 evaluate the SAMC coefficient and component variants. Compared with disabling SAMC ($\lambda_{\text{SAMC}} = 0$), all three tested nonzero coefficients (0.01, 0.10, and 1.00) improve both *Avg* and *Last* on every skill. Overall *Avg* rises from 50.39 to 51.78–52.37, *Last* from 46.94 to 50.27–50.78, and BWT from -10.71 to between -7.06 and -6.72. Variation among the nonzero settings is smaller than their gains over disabling SAMC. At $\lambda_{\text{SAMC}} = 0.10$, Med-PRISM further outperforms Uniform and No-shared by 0.75/0.40 points in *Last* and 0.82/0.27 points in BWT, respectively (computed from unrounded scores), while achieving higher overall *Avg* (52.37 versus 51.88/52.11). These results support SAMC's effectiveness across the tested coefficients and the contributions of spectrum-aware weighting and shared adaptation.

Table 2: **Component ablation.** Med-PRISM uses $\lambda_{\text{SAMC}} = 0.10$. *No-SAMC* disables SAMC by setting $\lambda_{\text{SAMC}} = 0$; *Uniform* replaces SAMC's spectrum-aware weighting with equal within-bank weights; *No-shared* removes the shared adapter branch and retains only task-private adaptation.

Setting	Eval.	Medical VQA	Diagnosis	Concept	Grounding	Reasoning VQA	Average	BWT
Med-PRISM	Avg	**87.42**	**48.75**	**29.24**	42.99	**53.45**	**52.37**	**-6.72**
	Last	**86.70**	**46.10**	**25.20**	**42.43**	**53.45**	**50.78**	
No-SAMC	Avg	86.64	44.93	26.52	42.33	51.56	50.39	-10.71
	Last	84.00	40.30	19.18	39.65	51.56	46.94	
Uniform	Avg	87.16	47.28	29.10	**43.05**	52.80	51.88	-7.54
	Last	86.50	44.40	24.74	41.67	52.80	50.02	
No-shared	Avg	86.98	48.15	28.91	43.04	**53.45**	52.11	-6.99
	Last	86.20	45.50	25.01	41.70	**53.45**	50.37	

4.4 Spectral Mechanism Analysis

Which historical directions need stronger protection? We test whether historical spectral energy reflects old-task sensitivity. For direction $\mathbf{u}_j$, λ_j is its memory weight and $s_j = \|G_k \mathbf{u}_j\|_2^2$ measures how sensitive the old-task loss is to local weight changes along it. Old-task gradients are used only for post-hoc analysis. Figure 4(a) shows how old-task sensitivity accumulates as historical directions are included from highest to lowest spectral energy. The steep initial rise indicates that high-energy directions account for a large share of sensitivity within the stored subspace. Across 720 layer–

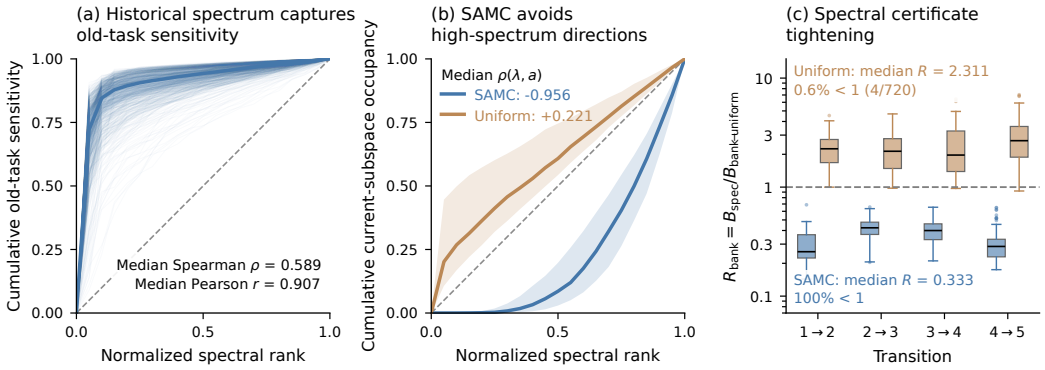


Figure 4: **SAMC spectral diagnostics. (a)** Cumulative old-task sensitivity within positive historical support, ordered by decreasing eigenvalue (thin curves; median and interquartile range). **(b)** Cumulative current-subspace occupancy, with old-task duplicates removed. **(c)** Spectral/bank-uniform first-order certificate ratio at each trained state (Appendix G).

task observations, median Spearman/Pearson correlations are $0.589/0.907$. These results support applying stronger protection to historical directions with higher spectral energy.

Does SAMC steer new adaptation away from these directions? We quantify the overlap between the new private adapter's input subspace and each historical direction using $a_j = ||P_t \mathbf{u}_j||_2^2$. Here, $a_j = 0$ indicates that the subspace is orthogonal to $\mathbf{u}_j$, whereas $a_j = 1$ indicates that $\mathbf{u}_j$ lies in the subspace. In Fig. 4(b), SAMC accumulates little overlap at the high-energy end; most of its overlap within the stored subspace lies toward the lower-energy end. The median energy–overlap Spearman correlation is -0.956 for SAMC versus $+0.221$ for Uniform, across 288 distinct observations per method. This supports *selective protection*: SAMC discourages overlap more strongly along high-energy historical directions while leaving lower-energy directions less constrained, rather than assigning equal weights within each historical bank.

Does the learned geometry tighten the interference bound? At each training stage, we compare two upper bounds on the magnitude of the first-order change in the old-task loss induced by the new private update within the historical subspace. Both bounds are evaluated using the same model parameters, old-task gradient, and current update. The spectral bound incorporates the learned historical spectral weights, whereas the bank-uniform bound assigns equal weight to all directions within each historical bank. Thus, $R_{\text{bank}} = B_{\text{spec}}/B_{\text{bank-uniform}} < 1$ means spectral weighting gives a smaller bound on the same interference term (Appendix A.4). In Fig. 4(c), SAMC-trained states have a median ratio of 0.333, with all $720/720$ ratios below one, versus 2.311 and $4/720$ for Uniform-trained states. This supports the intended link between SAMC's learned geometry and a tighter first-order bound in Sec. 3.4. The contrast shows that spectral anisotropy is not intrinsically beneficial; its certificate advantage emerges only when coupled with a favorable sensitivity–occupancy geometry, as observed at SAMC-trained endpoints.

5 Conclusion

We introduced Med-PRISM, a replay-free continual learning framework for acquiring heterogeneous medical vision–language skills. Med-PRISM combines a shared adapter for transferable knowledge with task-private parameter memory for cumulative reuse, and uses SAMC to selectively constrain new adaptation according to the spectral structure of previously learned updates. Our analysis connects this geometry to a first-order bound on old-task interference, while experiments on MedicalSkill-CL show improved continual learning performance across diverse medical skills. Together, These findings highlight the importance of using the spectral structure of historical parameter updates to guide subsequent adaptation, helping models retain existing capabilities while acquiring new ones. Our current evaluation is limited to a single backbone and random seed. Future work will examine broader model families and multi-seed settings.

ETHICS STATEMENT

This work studies replay-free continual learning for medical multimodal models using publically available medical resources and concept labels derived from image captions. The reported results assess performance on selected benchmark tasks and do not establish clinical safety or effectiveness. All authors of this work have read and commit to adhering to the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

We have taken several steps to ensure the reproducibility of our results. The mathematical assumptions and proofs are in Appendix A. The hyperparameters required for training are explicitly reported. Appendix B describes the five-task construction, split sizes, isolation procedure, and scoring scope. Appendices C–F specify the model version, training hyperparameters, generation settings, adapter configurations, numerical computations, and parameter and memory costs. Appendix G defines the component comparisons and spectral diagnostics reported in the paper.

AI USE STATEMENT

Generative AI tools were used to improve the clarity and readability of the manuscript, including correcting grammatical errors and typographical mistakes. They were also used to examine the correctness and consistency of the theoretical analysis and proofs. All AI-assisted content was reviewed and verified by the authors, who remain fully responsible for the final manuscript and its claims. The concept-recognition supervision was derived using Kimi K2.6, as described in Appendix B.1.

REFERENCES

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

Cheng Chen, Junchen Zhu, Xu Luo, Heng Tao Shen, Jingkuan Song, and Lianli Gao. CoIN: A benchmark of continual instruction tuning for multimodel large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1844. URL https://arxiv.org/abs/2403.08350.

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 *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2603.00191.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Yutao Hu, Tianbin Li, Quanfeng Lu, Wenqi Shao, Junjun He, Yu Qiao, and Ping Luo. OmniMedVQA: A new large-scale comprehensive evaluation benchmark for medical LVLM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22170–22183, 2024. URL https://arxiv.org/abs/2402.09181.

Hugging Face. PEFT: Parameter-efficient fine-tuning documentation. Software documentation, 2026. URL https://huggingface.co/docs/peft/index. Documentation consulted on 26 September 2026.

Georgios A. Kaissis, Marcus R. Makowski, Daniel Rückert, and Rickmer F. Braren. Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence*, 2:305–311, 2020. doi: 10.1038/s42256-020-0186-1. URL https://www.nature.com/articles/s42256-020-0186-1.

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

Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, 1955. doi: 10.1002/nav.3800020109. URL https://onlinelibrary.wiley.com/doi/10.1002/nav.3800020109.

Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. LLaVA-Med: Training a large language-and-vision assistant for biomedicine in one day. In *Advances in Neural Information Processing Systems*, pp. 28541–28564, 2023.

Yan-Shuo Liang and Wu-Jun Li. InfLoRA: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23638–23647, 2024. URL https://arxiv.org/abs/2404.00228.

Huanxuan Liao, Shizhu He, Yupu Hao, Yequan Wang, Wenhao Teng, Xiangwen Liao, Jun Zhao, and Kang Liu. Spectral disentanglement: Rank-aware task adaptation for rehearsal-free continual learning in LLMs. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7360–7376, 2026. doi: 10.18653/v1/2026.acl-long.334. URL https://aclanthology.org/2026.acl-long.334/.

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/.

Yuehao Liu, Shanyan Guan, Weijia Zhang, Xuanming Shang, Yanhao Ge, Wei Li, and Chao Ma. Octopus: History-free gradient orthogonalization for continual learning in multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2605.14938.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. URL https://arxiv.org/abs/1412.0035.

Moonshot AI. Kimi-K2.6: Model card. Hugging Face, 2026. URL https://huggingface.co/moonshotai/Kimi-K2.6. Accessed 26 September 2026.

Qwen Team. Qwen3-VL-8B-Instruct: Model repository. Hugging Face, 2025. URL https://huggingface.co/Qwen/Qwen3-VL-8B-Instruct. Model revision specified in Appendix C; accessed 26 September 2026.

Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R. Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N. Galtier, Bennett A. Landman, Klaus Maier-Hein, Sébastien Ourselin, Micah Sheller, Ronald M. Summers, Andrew Trask, Daguang Xu, Maximilian Baust, and M. Jorge Cardoso. The future of digital health with federated learning. *npj Digital Medicine*, 3:119, 2020. doi: 10.1038/s41746-020-00323-1. URL https://www.nature.com/articles/s41746-020-00323-1.

Gobinda Saha and Kaushik Roy. Continual learning with scaled gradient projection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 9677–9685, 2023. URL https://arxiv.org/abs/2302.01386.

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

Zailong Tian, Yanzhe Chen, Zhuoheng Han, and Lizi Liao. Spectral surgery: Training-free refinement of lora via gradient-guided singular value reweighting, 2026. URL https://arxiv.org/abs/2603.03995.

Tao Tu, Shekoofeh Azizi, Danny Driess, Mike Schaekermann, Mohamed Amin, Pi-Chuan Chang, Andrew Carroll, Chuck Lau, Ryutaro Tanno, Ira Ktena, Anil Palepu, Basil Mustafa, Aakanksha Chowdhery, Yun Liu, Simon Kornblith, David Fleet, Philip Mansfield, Sushant Prakash, Renee Wong, Sunny Virmani, Christopher Semturs, Sara Mahdavi, Bradley Green, Ewa Dominowska, Blaise Aguera y Arcas, Joelle Barral, Dale Webster, Greg Corrado, Yossi Matias, Karan Singhal, Pete Florence, Alan Karthikesalingam, and Vivek Natarajan. Towards generalist biomedical AI. *NEJM AI*, 1(3):AIoa2300138, 2024.

Gido M. van de Ven, Tinne Tuytelaars, and Andreas S. Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 4:1185–1197, 2022. URL https://www.nature.com/articles/s42256-022-00568-3.

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*, pp. 10658–10671, 2023. doi: 10.18653/v1/2023.findings-emnlp.715. URL https://aclanthology.org/2023.findings-emnlp.715/.

Yitong Wang, Xue Han, WenChun Gao, Qian Hu, Jiahui Wang, Ziqing Wang, Lijun Mei, and Junlan Feng. Soft orthogonal low-rank adaptation for knowledge sharing in large language model continual learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 18472–18487, 2026. doi: 10.18653/v1/2026.acl-long.842. URL https://aclanthology.org/2026.acl-long.842/.

Stefano Woerner, Arthur Jaques, and Christian F. Baumgartner. MedIMeta: An easy-to-use metadataset for medical imaging applications. In *Medical Imaging with Deep Learning Short Papers*, 2024. URL https://openreview.net/forum?id=AIPQBCXesT.

Yunfei Xie, Ce Zhou, Lang Gao, Juncheng Wu, Xianhang Li, Hong-Yu Zhou, Sheng Liu, Lei Xing, James Zou, Cihang Xie, and Yuyin Zhou. MedTrinity-25M: A large-scale multimodal dataset with multigranular annotations for medicine. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2408.02900.

Zonghai Yao, Benlu Wang, Yifan Zhang, Junda Wang, Iris Xia, Zhipeng Tang, Shuo Han, Feiyun Ouyang, Zhichao Yang, Arman Cohan, and Hong Yu. Medical thinking with multiple images. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=h2p5eOFpcF.

Jingkun Yue, Siqi Zhang, Zinan Jia, Huihuan Xu, Zongbo Han, Xiaohong Liu, and Guangyu Wang. MedSG-Bench: A benchmark for medical image sequences grounding. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2505.11852.

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*, 2023. URL https://openreview.net/forum?id=lq62uWRJjiY.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://arxiv.org/abs/1906.08935.

A Proofs and Spectral Analysis

We express the SAMC loss in terms of the spectral metric constructed from earlier private adapters, prove the two results in Sec. 3.4, and derive the bound comparisons used in Sec. 4.4.

A.1 Setup and Correspondence to the SAMC Loss

All quantities are layer-wise unless explicitly summed over the set $\mathcal{W}$ of M adapted layers; layer indices are omitted. Lowercase bold letters denote vectors. We use a fixed private rank r, with $A_t \in \mathbb{R}^{r\times d_{\text{in}}}$, $B_t \in \mathbb{R}^{d_{\text{out}}\times r}$, and $D_t = B_t A_t$. Branch scaling is absorbed into B_t. The current A_t has full row rank, while B_t and historical updates may be rank deficient. The analysis evaluates losses and gradients with dropout disabled. Because earlier private adapters remain frozen, the change in each adapted weight from the end of task $t-1$ to the end of task t is $D_t + \Delta S_t$, where $\Delta S_t = S_t - S_{t-1}$. Stacking these changes over $\mathcal{W}$ gives $\Delta\boldsymbol{\theta}_t$.

For a nonzero historical update, use an r-dimensional thin SVD $D_i = U_i\Sigma_i V_i^\top$, retaining zero singular values if the update is rank deficient. Let $\Sigma_i = \operatorname{diag}(\sigma_{i,1},\dots,\sigma_{i,r})$ and $p_{i,j} = \sigma_{i,j}^2/\|D_i\|_F^2$. Then

$$H_i = \frac{D_i^\top D_i}{\|D_i\|_F^2} = V_i \operatorname{diag}(\mathbf{p}_i) V_i^\top, \qquad K_t = r\sum_{i<t} H_i, \qquad e_t = \operatorname{tr}(P_t K_t). \tag{21}$$

We set $H_i = 0$ for a zero update. Each nonzero H_i is positive semidefinite with unit trace. Since

$$\mathbf{x}^\top K_t \mathbf{x} = r \sum_{i<t:\, D_i\neq 0} \frac{\|D_i\mathbf{x}\|_2^2}{\|D_i\|_F^2}, \tag{22}$$

the subspace spanned by the eigenvectors of K_t with positive eigenvalues, referred to below as the historical support, is

$$\operatorname{range}(K_t) = \operatorname{span}_{i<t} \operatorname{row}(D_i). \tag{23}$$

Let $K_t = U\Lambda U^\top$ contain only positive eigenvalues. We write $R_t = UU^\top = K_t K_t^\dagger$ and $K_t^{\dagger/2} := (K_t^\dagger)^{1/2} = U\Lambda^{-1/2}U^\top$. When all historical updates are zero, both matrices are zero. The pseudoinverse square root therefore acts only on directions with positive eigenvalues.

Projector and loss normalization. Because A_t has full row rank, its row space admits an orthonormal basis represented by a matrix $Q \in \mathbb{R}^{r\times d_{\text{in}}}$ with

$$QQ^\top = I, \qquad Q^\top Q = P_t = A_t^\top (A_t A_t^\top)^{-1} A_t, \qquad A_t P_t = A_t. \tag{24}$$

For the private adapter of an earlier task i, define $\mathbf{w}_i = r\mathbf{p}_i$ and $T_i = QV_i \operatorname{diag}(\sqrt{\mathbf{w}_i})$. Then

$$\|T_i\|_F^2 = \operatorname{tr}(P_t\, r H_i). \tag{25}$$

Each T_i has r^2 entries. Directions with zero spectral weight contribute zero to its squared Frobenius norm but remain included in the normalization count. For $t > 1$, define

$$N_t := Mr^2(t-1), \qquad \mathcal{E}_t := \sum_{\mathcal{W}} e_t. \tag{26}$$

Summing the historical blocks over all adapted layers gives

$$\mathcal{L}_{\text{SAMC}}^2 = \frac{\sum_{\mathcal{W}}\sum_{i<t}\|T_i\|_F^2}{N_t} = \frac{\mathcal{E}_t}{N_t}, \qquad \mathcal{E}_t = N_t \mathcal{L}_{\text{SAMC}}^2. \tag{27}$$

This recovers Eq. (8) with $K_t = r\sum_{i<t} H_i$ in exact arithmetic. The implementation adds a small constant before taking the square root, as specified in Appendix E.3. For task 1, the loss is zero because no earlier private adapter exists.

A.2 Proof of Proposition 1

Proof. Since $A_t P_t = A_t$, we have $D_t P_t = D_t$. For every vector $\mathbf{x}$,

$$\mathbf{x}^\top D_t^\top D_t \mathbf{x} = \|D_t P_t \mathbf{x}\|_2^2 \le \|D_t\|_2^2 \|P_t\mathbf{x}\|_2^2. \tag{28}$$

Hence $D_t^\top D_t \preceq \|D_t\|_2^2 P_t$. Taking the trace against $K_t \succeq 0$ gives

$$\|D_t K_t^{1/2}\|_F^2 = \operatorname{tr}(D_t^\top D_t K_t) \le \|D_t\|_2^2 \operatorname{tr}(P_t K_t) = \|D_t\|_2^2 e_t. \tag{29}$$

Taking square roots proves the claim. □

A.3 Proof of Theorem 1

Let G_k be the gradient of old-task loss $\mathcal{L}_k$ with respect to one adapted weight, evaluated before learning task t at $\boldsymbol{\theta}^{t-1}$. Its component within the historical support is $X_k = G_k R_t$. These gradients are used in the proof, not in SAMC training. Define $b = \|X_k K_t^{\dagger/2}\|_F$ for this layer and use the following sums over adapted layers:

$$\begin{aligned} (\Gamma_{k,t}^K)^2 &= \sum_{\mathcal{W}} b^2 \|D_t\|_2^2, \qquad \delta_{k,K}^2 = \sum_{\mathcal{W}} \|G_k(I - R_t)\|_F^2, \\ \rho_t^2 &= \sum_{\mathcal{W}} \|D_t\|_F^2, \qquad d_t^2 = \sum_{\mathcal{W}} \|\Delta S_t\|_F^2, \qquad g_k^2 = \sum_{\mathcal{W}} \|G_k\|_F^2. \end{aligned} \tag{30}$$

We assume that the gradient of $\mathcal{L}_k$ is β_k-Lipschitz on the line segment from $\boldsymbol{\theta}^{t-1}$ to $\boldsymbol{\theta}^t$, with distance measured by the Frobenius norm of the stacked weight changes.

Proof. Historical-support term. Because $X_k = X_k R_t$ and $K_t^{\dagger/2} K_t^{1/2} = R_t$,

$$\langle X_k, D_t \rangle_F = \langle X_k K_t^{\dagger/2}, D_t K_t^{1/2} \rangle_F. \tag{31}$$

Frobenius Cauchy–Schwarz and Proposition 1 therefore yield $|\langle X_k, D_t \rangle_F| \le b \|D_t\|_2 \sqrt{e_t}$. Summing over layers and applying Cauchy–Schwarz once more gives

$$\left| \sum_{\mathcal{W}} \langle X_k, D_t \rangle_F \right| \le \underbrace{\sum_{\mathcal{W}} b \|D_t\|_2 \sqrt{e_t}}_{B_{\mathrm{sum}}} \le \Gamma_{k,t}^K \sqrt{\mathcal{E}_t}. \tag{32}$$

This bound covers only $G_k R_t$. The remaining gradient component, $G_k(I - R_t)$, contributes the separate term $\delta_{k,K}\rho_t$, giving

$$\left| \sum_{\mathcal{W}} \langle G_k, D_t \rangle_F \right| \le B_{\mathrm{sum}} + \delta_{k,K}\rho_t. \tag{33}$$

Endpoint loss change. Stack the layer-wise weight changes $D_t + \Delta S_t$ into $\Delta\boldsymbol{\theta}_t$ and set $\Delta\mathcal{L}_k = \mathcal{L}_k(\boldsymbol{\theta}^t) - \mathcal{L}_k(\boldsymbol{\theta}^{t-1})$. Integrating the gradient along the line segment between the two model states gives

$$\begin{aligned} \left| \Delta\mathcal{L}_k - \sum_{\mathcal{W}} \langle G_k, D_t + \Delta S_t \rangle_F \right| &\le \int_0^1 \beta_k z \|\Delta\boldsymbol{\theta}_t\|_F^2 \, dz \\ &\le \frac{\beta_k}{2} (\rho_t + d_t)^2. \end{aligned} \tag{34}$$

Combining this with Eq. (33) and $|\sum_{\mathcal{W}} \langle G_k, \Delta S_t \rangle_F| \le g_k d_t$ yields

$$|\Delta\mathcal{L}_k| \le B_{\mathrm{sum}} + \delta_{k,K}\rho_t + g_k d_t + \frac{\beta_k}{2}(\rho_t + d_t)^2. \tag{35}$$

Finally substitute $B_{\mathrm{sum}} \le \Gamma_{k,t}^K \sqrt{\mathcal{E}_t}$ and $\mathcal{E}_t = N_t \mathcal{L}_{\mathrm{SAMC}}^2$ from Eq. (27) to obtain Eq. (17). □

With the task-start anchor $S_t^0 = S_{t-1}$ for $t > 1$, Eq. (10) gives $d_t^2 = c_t^{\mathrm{sh}} \mathcal{L}_{\mathrm{shared}}$, where $c_t^{\mathrm{sh}} = \max\{\sum_{\mathcal{W}} \|S_{t-1}\|_F^2, \epsilon_{\mathrm{sh}}\}$. SAMC penalizes overlap between the current private input subspace and historical directions. The shared penalty instead penalizes the magnitude d_t of the shared-weight change. The bound still includes $\delta_{k,K}\rho_t$, which accounts for old-task gradients outside the stored subspace.

A.4 Spectral Weighting and Uniform Comparisons

Fix one model state, its old-task gradient G_k, current update D_t, and current projector P_t. For this subsection, write $K_t = \sum_j \lambda_j \mathbf{u}_j \mathbf{u}_j^\top$ on its positive support, $m = \|D_t\|_2$, and

$$s_j = \|G_k \mathbf{u}_j\|_2^2, \qquad a_j = \|P_t \mathbf{u}_j\|_2^2 \in [0, 1]. \tag{36}$$

Here s_j is the squared old-task gradient magnitude along historical direction $\mathbf{u}_j$, and a_j is its overlap with the current private input subspace. Specifically, $a_j = 0$ means orthogonality and $a_j = 1$ means that the current subspace contains $\mathbf{u}_j$. The weight λ_j is derived from earlier private parameters, not from the old-task gradient. Expanding the two factors in Eq. (32) gives the single-layer bound

$$B_{\text{spec}} = m\sqrt{F(\boldsymbol{\lambda})}, \qquad F(\boldsymbol{\lambda}) = \left(\sum_j \frac{s_j}{\lambda_j}\right)\left(\sum_j \lambda_j a_j\right). \tag{37}$$

Increasing λ_j decreases s_j/λ_j but increases $\lambda_j a_j$. A larger historical weight therefore does not always reduce the product in Eq. (37).

Why nonuniform weights can help. For fixed positive s_j, a_j, Cauchy–Schwarz gives

$$F(\boldsymbol{\lambda}) \geq \left(\sum_j \sqrt{s_j a_j}\right)^2, \qquad \lambda_j^\star = c\sqrt{s_j/a_j}, \quad c > 0. \tag{38}$$

Indeed, apply the inequality to $\sqrt{s_j/\lambda_j}$ and $\sqrt{\lambda_j a_j}$; equality requires these vectors to be proportional, yielding the stated weights. For the flat metric $K_{\text{flat}} = R_t$, the bound is $B_{\text{flat}} = m\sqrt{(\sum_j s_j)(\sum_j a_j)}$. The optimal value in Eq. (38) is no larger, and is strictly smaller when the positive vectors $\mathbf{s}$ and $\mathbf{a}$ are not proportional. These weights minimize the bound when the eigenvectors, gradients, and current subspace are fixed. They are a theoretical comparison: SAMC does not compute or use these gradient-dependent weights. Because the bound is invariant to multiplying a metric by a common positive scalar, the fixed factor r in the definition of K_t does not affect this comparison.

For $m > 0$, $\sum_j s_j > 0$, and $\sum_j a_j > 0$, SAMC's spectral weights yield a smaller bound than K_{flat} exactly when

$$\left(\frac{B_{\text{spec}}}{B_{\text{flat}}}\right)^2 = \frac{\sum_j s_j/\lambda_j}{\sum_j s_j}\,\frac{\sum_j \lambda_j a_j}{\sum_j a_j} < 1. \tag{39}$$

When a denominator is zero, both bounds vanish and the ratio is undefined. Multiplying all weights by a common positive scalar leaves the ratio and the bound unchanged.

Uniform weighting within each historical adapter. Figure 4 uses equal weights within each earlier private adapter, rather than equal weights over the combined historical subspace. Let $q_i = \operatorname{rank}(D_i)$ for a nonzero historical update, and let R_i project onto $\operatorname{row}(D_i)$. With the same factor r as in K_t,

$$K_{\text{bank-uniform}} = r \sum_{i<t:\, D_i \neq 0} \frac{R_i}{q_i}, \qquad B(K') = m\|G_k R_t (K')^{\dagger/2}\|_F \sqrt{\operatorname{tr}(P_t K')}. \tag{40}$$

The formula for $B(K')$ applies when K' has the same support as K_t. Overlapping banks can make $K_{\text{bank-uniform}}$ anisotropic; for example, three rank-one banks along $\mathbf{e}_1, \mathbf{e}_1, \mathbf{e}_2$ give a metric proportional to $\operatorname{diag}(2, 1)$ rather than $R_t = I$. The diagnostic therefore evaluates $R_{\text{bank}} = B(K_t)/B(K_{\text{bank-uniform}})$ directly, rather than substituting the flat expression above. Within each ratio, the model parameters, gradient, and current update are fixed; only the comparison metric changes. Ratios for SAMC and Uniform are evaluated at their respective trained model states, which can differ.

B MedicalSkill-CL Construction and Evaluation Details

We convert five medical data resources into examples containing images, an instruction, and a target response. We then check for shared images and source records and sample examples for the fixed five-task sequence below. The task order is fixed; random sampling selects examples, not the order of tasks.

B.1 Task Definitions and Scoring

Table 3: The five-skill evaluation sequence. All tasks use text generation, with task-specific targets and scores. Grounding and reasoning score definitions appear in Sec. 4.1.

Skill	Source	Input and target	Score
1. Medical VQA	OmniMedVQA (Hu et al., 2024)	Single image; canonical answer text	Accuracy
2. Diagnosis	MedIMeta (Woerner et al., 2024)	Single image; source-specific diagnosis label	Accuracy
3. Concepts	MedTrinity-25M (Xie et al., 2025)	Single image; caption-derived concept set	Micro-F1
4. Grounding	MedSG (Yue et al., 2025)	2–6 images; one normalized box	IoU composite
5. Reasoning VQA	MedThinkVQA (Yao et al., 2026)	1–8 images and context; reasoning and answer	Answer/reasoning composite

Medical VQA and diagnosis. Medical VQA uses single-image OmniMedVQA examples with canonical answer text rather than option letters; the number of answer candidates varies. Diagnosis uses eight MedIMeta subsets, each retaining its source label space. For both tasks, the evaluator matches the generated answer text to the reference answer or label. Diagnosis labels remain specific to each source; they are not mapped to one common label set.

Concept recognition. We use Kimi K2.6 (Moonshot AI, 2026) to extract 1–8 English medical concepts from each MedTrinity-25M caption. The extracted sets undergo automatic checks and manual spot-checks. The model outputs concepts separated by semicolons, and micro-F1 treats them as an unordered set. These labels are generated from captions, not independently annotated by clinicians.

Visual grounding. MedSG examples contain 2–6 images and one target box, serialized as `<box>x1,y1,x2,y2</box>` with integer coordinates in $[0, 1000]$. The evaluator parses the predicted boxes and matches them to reference boxes using the Hungarian algorithm (Kuhn, 1955), with intersection over union (IoU) as the matching score. The task score is the mean of accuracy at IoU ≥ 0.5 and mean IoU.

Reasoning VQA. MedThinkVQA provides 1–8 images, clinical context, five answer candidates, a reasoning sequence, and a canonical final answer. The primary score is the mean of final-answer accuracy and ROUGE-L (Lin, 2004) between the generated and reference reasoning text. ROUGE-L measures text overlap, not the clinical correctness of the reasoning. Cross-stage aggregation is specified in Appendix C.4; decoding is specified in Appendix C.3.

B.2 Construction and Random Sampling

Figure 5 shows the construction pipeline. We sample without replacement using seed 42. Sampling is stratified by task and source, with modality balancing within each source. Each task contributes at most 10,000 training and 1,000 test examples. All 7,347 training and 720 test examples for Reasoning VQA are retained because this task is below both limits. Sampling does not enforce equal class frequencies. Table 4 gives the counts before and after sampling; all performance tables use the sampled data.

These counts are measured before model-specific encoding. The training pipeline removes examples that exceed the configured sequence-length limit, so the number of examples used for training can be smaller than the counts in Table 4.

B.3 Image and Source-Identifier Overlap Checks

We check exact image hashes and normalized identifiers derived from available patient, case, group, source, and file-path information. These identifiers, called lineage tokens, link records that may share a common origin. We first remove training records that share a lineage token with any test

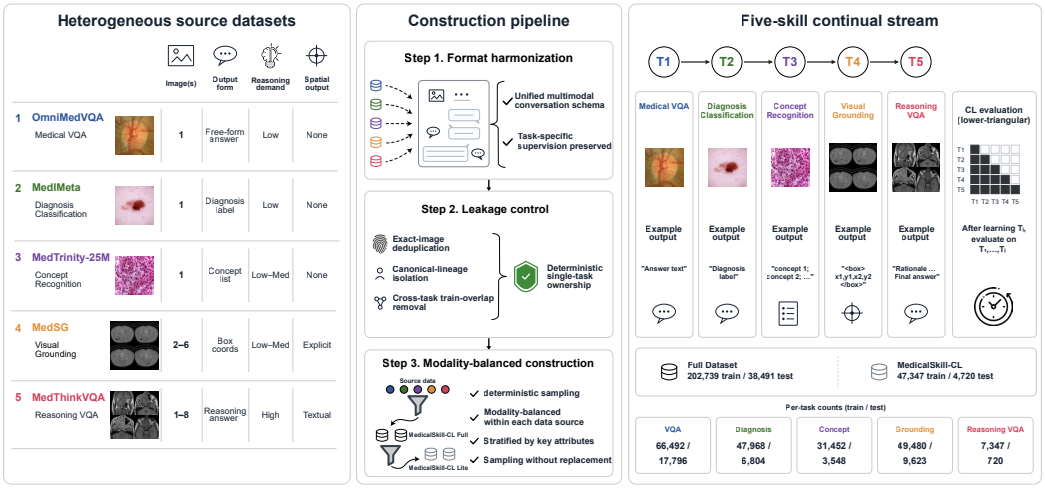


Figure 5: **Construction of MedicalSkill-CL.** Examples share an image–instruction–response format but retain their task-specific targets. Image and source-identifier checks remove detected overlaps before stratified sampling. Counts on the right describe the full data pool before benchmark sampling. The reasoning labels illustrate task characteristics; they are not measured difficulty scores.

Table 4: Record counts for the full data pool and the sampled MedicalSkill-CL benchmark before filtering examples that exceed the encoded sequence-length limit. All experiments use the sampled benchmark. The totals summarize the fixed five-task sequence.

	Full data pool		MedicalSkill-CL	
Task	Train	Test	Train	Test
Medical VQA	66,492	17,796	10,000	1,000
Diagnosis	47,968	6,804	10,000	1,000
Concepts	31,452	3,548	10,000	1,000
Grounding	49,480	9,623	10,000	1,000
Reasoning VQA	7,347	720	7,347	720
Total	202,739	38,491	47,347	4,720

record. We then connect the remaining training records whenever they share a token. Records connected directly or through other records form one group, which is assigned to a single task using a deterministic rule. Exact-image checks are applied in addition to these identifier-based checks.

The overlap checks found no shared image hashes or lineage tokens between different training tasks, or between any training and test sets. Different test sets may still overlap. Because some sources lack reliable patient identifiers, these checks do not establish patient-level separation across every source.

C Common Implementation and Evaluation Settings

The following settings apply to the Sequential, O-LoRA, MoE-LoRA, Octopus, and Med-PRISM runs reported in this paper.

C.1 Backbone and Supervision

All methods use `Qwen/Qwen3-VL-8B-Instruct` (Qwen Team, 2025). The model files are fixed to repository commit `0c351dd01ed87e9c1b53cbc748cba10e6187ff3b`. The language model has 36 layers, hidden size 4,096, 32 query heads, and 8 key/value heads. Adapters target the 72 query/value projections, with shapes $4{,}096 \to 4{,}096$ and $4{,}096 \to 1{,}024$, respectively.

All pretrained weights, including the vision encoder, merger, embeddings, and output head, remain frozen. Baseline adapters are implemented with extensions to the PEFT library (Hugging Face, 2026). Med-PRISM uses a custom implementation of the shared and private adapters.

Tasks are trained sequentially on the sampled datasets described in Appendix B.2. Training and data seeds are 42, and training examples are shuffled. Encoded examples exceeding the method-specific length limit are removed. No validation split is created automatically, and training-time evaluation is disabled.

Multimodal processing. The image/text processor from the same repository commit and the `qwen3_vl` conversation template read image paths and messages from JSONL files. During training, the maximum pixel count is 200,704 and no minimum is explicitly set. During evaluation, both pixel-count bounds are 200,704. The processor determines resizing and patch alignment; images are not forcibly resized to 448×448. Evaluation removes textual image placeholders, supplies the images and user message, and adds the assistant-generation prompt. Reference answers are not included in the model input.

Training targets. Every task uses causal cross-entropy on the assistant response and the template's assistant-response suffix. User and system messages, visual tokens, and padding are excluded from the loss using label -100. Label smoothing is disabled. Loss normalization counts supervised tokens across gradient-accumulation steps and distributed processes, rather than averaging examples with different target lengths equally. Diagnosis and grounding use the same generative objective, without auxiliary classification or box-regression heads.

C.2 Training Hyperparameters and Checkpoints

Table 5: Shared training settings for the five continual methods. Method-specific ranks, learning rates, sequence lengths, and additional training stages appear in Table 6.

Setting	Value
Optimizer	AdamW (Loshchilov & Hutter, 2019) (`adamw_torch`)
Weight decay; betas; epsilon	0.1; $(0.9, 0.95)$; 10^{-8}
Schedule; warmup ratio	Cosine; 0.03
Maximum gradient norm	1.0
Per-device batch; accumulation	1; 8 steps
Distributed training	Two processes / two GPUs; effective batch 16
Precision; attention	BF16; scaled dot-product attention (SDPA)
Gradient checkpointing	Enabled for training; vision checkpointing disabled
LoRA dropout; bias	0.05; no additional bias parameters
Workers; dataset processes	0; 1
Sequence packing	Disabled
Unused-parameter search in DDP	Disabled
Training-time evaluation	Disabled; no automatically created validation split
Checkpoint policy	Save at epoch ends; retain one checkpoint per training directory

The effective batch is $1 \times 8 \times 2 = 16$, except for incomplete final batches. Med-PRISM used two A100-PCIE-40GB GPUs; baseline runs also used two processes. FP16 and DeepSpeed were disabled. SAMC's local precision controls are given in Appendix E.3. At each new task or training substage, the optimizer and learning-rate schedule are reinitialized while the required learned weights are loaded. Checkpoints are saved at epoch ends in separate stage directories. Resuming an interrupted stage restores the model weights, optimizer, learning-rate schedule, and random-number-generator state.

C.3 Decoding and Task Identity

Generation is greedy, with one beam and no sampling. Maximum new-token limits for tasks 1–5 are 64, 64, 128, 96, and 256. Evaluation uses all sampled test examples in their saved order. Baseline evaluation generally uses one example at a time, whereas Med-PRISM uses batches of four with

left padding. For Med-PRISM, each GPU independently evaluates a trained stage on a test task. Different batch sizes can produce small floating-point differences in BF16.

No method uses the true test-task ID to select adapters. Task type still determines the output limit and scorer, and instructions may contain task cues. Med-PRISM uses the shared adapter and all private adapters learned up to that stage. Spectral-memory files are not used during generation.

C.4 Cross-Stage Aggregation

Let $R_{t,i} \in [0,1]$ denote task i's score after stage t, with $T = 5$. For task i, $\mathrm{Avg}_i = (T - i + 1)^{-1} \sum_{t=i}^{T} R_{t,i}$ averages its score from the stage when it is first learned through stage T, whereas $\mathrm{Last}_i = R_{T,i}$ uses only the final stage. The overall Avg and Last scores are the unweighted means of these per-task values. Consequently, overall Avg is not the mean of all 15 evaluated stage–task pairs. Zero-shot reports the mean of $R_{0,i}$ over the five tasks. For the final task, $\mathrm{Avg}_T = \mathrm{Last}_T$.

Where acquisition scores are available, $\mathrm{BWT}_5 = \frac{1}{4} \sum_{i=1}^{4} (R_{5,i} - R_{i,i})$. Negative BWT means that earlier tasks perform worse at the final stage than immediately after they were learned. It is not the difference between overall Last and overall Avg. Aggregation uses unrounded scores before multiplying by 100; BWT is reported in score points.

D Baseline Implementations and Training Budgets

Table 6 lists the settings used in our implementations. New baseline adapters use Kaiming-uniform initialization for A, zero initialization for B, dropout 0.05, and LoRA scale $\alpha/r = 2$. Med-PRISM uses scale 1 and a different initialization for the private input factors A (Appendix E.1). The methods do not have matched training budgets: adapter ranks, the number of training stages, and additional gradient computations differ.

Table 6: Method-specific training settings. Lengths $1{,}024/4{,}096$ refer to tasks 1–4 / task 5. For MoE-LoRA, rank and alpha are for the entire four-expert bank. Epoch counts refer to each listed training stage, not to a shared total budget.

Method / component	r	α	Learning rate	Epochs	Max. length
Sequential	48	96	10^{-4}	1	1,024/4,096
O-LoRA	16	32	10^{-4}	1	1,024/4,096
MoE-LoRA (4×12)	48	96	10^{-4}	1	1,024/4,096
Octopus, Stage 1	16	32	10^{-4}	1	1,024/4,096
Octopus, Stage 2	16	32	10^{-4}	1	1,024/4,096
Med-PRISM, shared	32	32	10^{-5}	1	1,024/4,096
Med-PRISM, private	16	16	10^{-4}	1	1,024/4,096

D.1 Sequential

A single rank-48 adapter is continued across tasks, with both factors trained by task loss. Training and inference use $W_0 + 2B_tA_t$ from the current checkpoint. Earlier checkpoints are saved copies and are not loaded as additional adapters at inference. No additional regularizer or router is used.

D.2 O-LoRA

O-LoRA (Wang et al., 2023) adds one rank-16 adapter per task and keeps all earlier adapters frozen but included in the model. With dropout disabled, the effective layer weight is $W_0 + \sum_{i=1}^{t} 2B_iA_i$. With layer indices suppressed, the loss is

$$\mathcal{L}_{\text{O-LoRA}} = \mathcal{L}_{\text{task}} + 0.5 \sum_{\mathcal{W}} \sum_{i<t} \|A_i A_t^\top\|_{1,\text{entrywise}}. \tag{41}$$

The penalty sums the absolute values of all entries of $A_iA_t^\top$, without row normalization or averaging. Its gradients update only the current A_t. No factor-norm penalty is used. Inference sequentially

merges the learned adapters into the base, leaving no separate LoRA modules. Merging in BF16 can produce small rounding differences compared with summing the adapter outputs during training.

D.3 MoE-LoRA

Our MoE-LoRA implementation follows the mixture-of-experts design in CoIN (Chen et al., 2024). It uses one rank-48 adapter, split into four rank-12 experts, throughout the task sequence. Each target layer has a bias-free $4{,}096 \to 4$ token router. For input $\mathbf{h}$,

$$\mathbf{p}(\mathbf{h}) = \mathrm{softmax}(R\mathbf{h}), \qquad \Delta\mathbf{h} = 2\sum_{e=1}^{4} p_e(\mathbf{h}) B_e A_e \,\mathrm{dropout}(\mathbf{h}). \tag{42}$$

The router receives the input before dropout, while all experts receive the same input after dropout. All four experts are active, with no top-k selection or task-ID input. The scale is $96/48 = 2$. Experts and routers continue across tasks under task loss alone, without load balancing. The routing weights depend on the input at inference, so the experts cannot be replaced by one fixed merged weight matrix. The 72 routers add 1,179,648 parameters.

D.4 Octopus

Octopus (Liu et al., 2026) uses two rank-16 stages per task, each trained for one epoch, including task 1. Stage 1 continues the preceding Stage-1 adapter. Stage 2 initializes from the current task's Stage 1 and adds all frozen historical Stage-2 adapters, giving $W_0 + \sum_{i=1}^{t} 2B_i^{(2)} A_i^{(2)}$. Stage 1 uses task loss. For Stage 2, let G be the sum of gradient estimates obtained by evaluating earlier cumulative Stage-2 models on current-task data, as described below. Set $\varepsilon = 10^{-8}$ and define

$$\begin{aligned} Q &= \mathrm{stopgrad}(A_t G^\top), \\ \mathcal{L}_O &= \sum_{\mathcal{W}} \left| \left\langle \frac{B_t}{\|B_t\|_F + \varepsilon}, \frac{Q^\top}{\|Q\|_F + \varepsilon} \right\rangle_F \right|, \qquad \mathcal{L}_B = \sum_{\mathcal{W}} \|B_t\|_F. \end{aligned} \tag{43}$$

For $t \geq 2$, the loss is $\mathcal{L}_{\mathrm{task}} + 0.01\mathcal{L}_O + 0.01\mathcal{L}_B$; both auxiliary coefficients are zero on task 1. The stop-gradient operation on Q prevents the auxiliary losses from updating A directly; they update B instead.

Gradient estimation and inference. For each earlier stage $j < t$, the Stage-2 adapters from tasks 1 through j are merged into a copy of the pretrained base model. Gradients are computed on the first 256 training examples of the current task, in training mode with batch size one on one GPU. Only the target query/value weights require gradients, and no optimizer steps are taken. For each cumulative model, gradients are averaged in FP32 and saved in BF16. Stage 2 sums these estimates in FP32 without dividing by the number of earlier stages, then converts the sum to the dtype of B_t. This procedure uses current-task data rather than replaying old examples, but requires additional computation and storage for dense gradient matrices. Inference merges only the Stage-2 adapters; Stage-1 checkpoints are saved but not used for generation.

E Med-PRISM Implementation and Numerical Details

The full model minimizes Eq. (9) with $\lambda_{\mathrm{SAMC}} = 0.1$ and $\lambda_{\mathrm{sh}} = 0.01$. The objective contains only the task loss, SAMC loss, and shared-drift penalty. Figure 3 compares SAMC coefficients, and Appendix G.1 describes the component comparisons.

E.1 Parameterization and Initialization

Each of the 72 target projections has one rank-32 shared adapter and one private adapter per task with total rank 16. The private adapter is implemented as 16 rank-one branches. The shared adapter uses $\alpha_{\mathrm{sh}} = 32$, and the complete private adapter uses $\alpha_{\mathrm{pr}} = 16$, so both effective scales are one. Private scaling is based on the total rank 16, not the rank of an individual branch. The shared input factor A uses Kaiming-uniform initialization, private input factors use $\mathcal{N}(0, 0.02^2)$, and all output factors B start at zero. Dropout is 0.05.

The 16 current private branches are trained jointly without routing. With dropout disabled, concatenating their factors gives one effective update of rank at most 16. Because dropout is applied independently to each branch during training, changing the factorization can change the dropout-perturbed outputs even when the deterministic update is unchanged. Only shared and current private factors enter the optimizer, at learning rates 10^{-5} and 10^{-4}, respectively. Each task uses one epoch and the common settings in Table 5. Historical private factors and all pretrained weights remain frozen.

E.2 Compact Spectral Memory

At a task boundary, concatenate the private factors into $A \in \mathbb{R}^{16\times 4{,}096}$ and $B \in \mathbb{R}^{d_{\text{out}}\times 16}$. We compute reduced QR factorizations on the CPU in FP64 and apply SVD to the resulting 16×16 matrix:

$$B = Q_B R_B, \qquad A^\top = Q_A R_A, \qquad R_B R_A^\top = \widetilde{U}\Sigma\widetilde{V}^\top, \qquad V = Q_A\widetilde{V}. \tag{44}$$

All 16 directions are retained, without energy-threshold truncation. For each nonzero task–layer update, $p_j = \sigma_j^2/\sum_f \sigma_f^2$ and $w_j = 16p_j$; zero-energy updates receive zero weights. Thus the weights sum to 16 for each nonzero task–layer update.

A separate spectral-memory file stores $V^\top$ and $\mathbf{w}$ in FP32, and $\boldsymbol{\sigma}$ and $\mathbf{p}$ in FP64. Its metadata record the source adapter, target projections, model commit, and configuration. Left singular vectors are used only to check that the factorization reconstructs the private update. The file stores neither the dense update nor an input-space projector, and the original private factors remain unchanged. Constructing this memory requires only the learned parameters, not training images, features, or gradients.

E.3 Runtime Overlap and Numerical Checks

Let $V_h^\top \in \mathbb{R}^{16(t-1)\times 4{,}096}$ concatenate historical directions and let $\mathbf{w}_h$ contain their weights. At each layer,

$$\begin{aligned} &Z = \text{rowNormalize}(A_t; 10^{-8}), \qquad C = ZV_h, \\ &G = \tfrac{1}{2}\big(ZZ^\top + (ZZ^\top)^\top\big) = LL^\top, \qquad T = L^{-1}C\,\text{diag}(\sqrt{\mathbf{w}_h}). \end{aligned} \tag{45}$$

Here G is the 16×16 Gram matrix of the row-normalized current input factors, not the old-task gradient G_k in the theory. Triangular solves compute $L^{-1}C$ without a dense inverse or input-space projector. For $t > 1$, the implemented loss is

$$\mathcal{L}_{\text{SAMC}}^{\text{impl}} = \sqrt{\frac{\sum_{\mathcal{W}} \|T\|_F^2}{72 \cdot 16^2(t-1)} + \epsilon_{\text{rms}}}, \qquad \epsilon_{\text{rms}} = 10^{-8}. \tag{46}$$

The constant is added once, after averaging over layers and historical directions. In exact arithmetic, Eq. (27) gives $(\mathcal{L}_{\text{SAMC}}^{\text{impl}})^2 = \mathcal{L}_{\text{SAMC}}^2 + \epsilon_{\text{rms}}$. Thus the theoretical and implemented losses are not exactly equal. For task 1, a separate branch returns zero with zero gradient.

Normalization, overlap products, and Gram products use FP32 with local autocast and TF32 disabled. The symmetrized Gram is converted to FP64 for eigenvalue checks, Cholesky factorization, and triangular solves; the solved overlap matrix is converted back to FP32 for summation. The previous TF32 settings are restored afterward. The implementation requires finite values, a positive-definite Gram, and condition number at most 10^4. If a check fails, the implementation raises `NumericalBlocked` and stops every distributed worker. It does not add a diagonal perturbation to G, replace the solve with a pseudoinverse, or switch to another overlap formula. The constants used in row normalization and Eq. (46) are not added to the Gram matrix. Gradients pass through these operations to current private A; historical tensors are detached.

Relation to the exact-arithmetic proof. The stored tensors define the positive semidefinite metric $\widehat{K}_t = \sum_{i<t} \widehat{V}_i\,\text{diag}(\widehat{\mathbf{w}}_i)\widehat{V}_i^\top$. Because $\mathbf{w}_i = r\mathbf{p}_i$ with $r = 16$, this uses the same scaling as $K_t = r\sum_{i<t} H_i$. For any symmetric computed projector $\widehat{P}_t$,

$$\left|\text{tr}(P_tK_t) - \text{tr}(\widehat{P}_t\widehat{K}_t)\right| \le r\|K_t - \widehat{K}_t\|_2 + \text{tr}(\widehat{K}_t)\|P_t - \widehat{P}_t\|_2. \tag{47}$$

The inequality follows by separating the metric error from the projector error and using $\mathrm{tr}(P_t) = r$. It does not include rounding in the final scalar summation or the known ϵ_{rms} offset. We do not report measured bounds on these floating-point errors.

E.4 Shared Drift and Task Transitions

At the start of task $t > 1$, we save a fixed copy of the previous shared update, $S_t^0 = S_{t-1}$, as the reference for the shared penalty. Equation (10) divides the sum of squared effective-update differences by the summed squared anchor norm, floored at 10^{-8}. Trace products of the low-rank factors are computed in FP32, avoiding construction of dense weight matrices. A negative squared-difference value caused by rounding is replaced by zero at each layer before summation. Only shared factors receive this gradient. Task 1 returns exact zero for the shared penalty.

Training loads all earlier private adapters, keeps them frozen in the forward pass, creates the current private adapter, and reads the earlier spectral-memory files for SAMC. At the end of the task, the shared and current private factors are saved separately, and the current private adapter is used to construct a new spectral-memory file. Resuming a stage also restores the optimizer, learning-rate schedule, and random-number-generator state. Some earlier adapter or memory files are loaded from paths outside the stage checkpoint, so that checkpoint alone is not necessarily sufficient to resume training.

At stage-t inference, the shared adapter and all private adapters from tasks 1 through t are frozen and active. Generation does not load spectral-memory files or compute Gram factorizations and overlap penalties. SAMC therefore adds no computation beyond evaluating these adapters. The number of retained private parameters still grows with the number of tasks.

F Parameter and Auxiliary-Memory Accounting

F.1 Trainable Parameters and Retained Updates

For 36 query/value pairs, a rank-r adapter contains $C(r) = 36[(4{,}096{+}4{,}096){+}(4{,}096{+}1{,}024)]r = 479{,}232r$ scalar parameters in A and B. Thus $C(16) = 7{,}667{,}712$, $C(32) = 15{,}335{,}424$, and $C(48) = 23{,}003{,}136$. Table 7 excludes the pretrained backbone, optimizer state, archives, and auxiliary tensors. The retained parameter count is measured before any inference-time merging. O-LoRA and Octopus merge their adapter updates into the base weights; MoE-LoRA keeps its experts and token routers active.

Table 7: Adapter and router parameters, in millions rounded to three decimals. Trainable counts are per training stage; retained counts are after task 5, before merging. Retained counts include routers but exclude earlier checkpoint copies and auxiliary memory. S2 denotes Octopus Stage 2.

Method	Trainable	Retained	Stage-5 inference use
Sequential	23.003	23.003	One continuing adapter
O-LoRA	7.668	38.339	Five updates merged into base
MoE-LoRA	24.183	24.183	Four experts and token routers
Octopus	7.668	38.339	Five S2 updates merged into base
Med-PRISM	23.003	53.674	Shared plus all five private adapters

MoE-LoRA has 24,182,784 trainable scalars, including 1,179,648 router parameters. Each Octopus substage trains 7,667,712 scalars; the two substages are not optimized simultaneously. Med-PRISM trains 23,003,136 scalars per task and retains 53,673,984 shared–private scalars at task 5.

F.2 Archives and Auxiliary State

The five O-LoRA updates and five Octopus Stage-2 updates each contain 38,338,560 factor scalars before merging. Octopus's combined Stage-1/2 archive contains 76,677,120 scalars. Earlier Sequential checkpoints and copies of the Med-PRISM shared adapter are saved for record keeping, but are not all loaded during inference.

Each task-specific SAMC spectral-memory file contains $72(16 \cdot 4{,}096 + 16 + 16 + 16) = 4{,}722{,}048$ stored numeric values, with the precisions listed in Appendix E.2. The task-5 file is reported as 18,932,856 bytes. Five files of this size would total approximately 94.66 MB, using $1\,\mathrm{MB} = 10^6$ bytes and excluding JSON metadata. Training task 5 reads the four earlier spectral-memory files. The fifth file is saved after task 5 for any later task and is not used in final inference.

G Component Comparisons and Spectral Diagnostics

G.1 Reported Comparisons

Figure 3 reports one run for each $\lambda_{\mathrm{SAMC}} \in \{0, 0.01, 0.10, 1.00\}$. The main results and component comparisons use the same $\lambda_{\mathrm{SAMC}} = 0.10$ run. Each trained stage is evaluated on the tasks learned so far, using the aggregation in Appendix C.4.

Uniform assigns equal weights to the nonzero singular directions of each earlier private update: $H_i^{\mathrm{uni}} = R_i / \operatorname{rank}(D_i)$. It uses the same row-space projector formula for the current input factors and the same normalization count $N_t = Mr^2(t-1)$ as SAMC. The two methods are trained separately; their learned projectors need not be equal. Appendix A.4 distinguishes this comparison from uniform weighting over the combined historical subspace.

G.2 Spectral Diagnostics: Definitions and Interpretation

Figure 4 analyzes the model states after learning tasks 2–5 from a single-seed experiment. These are post-training analyses of the parameters and old-task gradients, not additional training objectives or inputs to SAMC. At each layer, order the positive historical eigenvalues as $\lambda_1 \geq \cdots \geq \lambda_q > 0$ and use s_j and a_j from Eq. (36). For nonzero denominators, the cumulative curves are

$$C_s(m) = \frac{\sum_{j=1}^{m} s_j}{\sum_{j=1}^{q} s_j}, \qquad C_a(m) = \frac{\sum_{j=1}^{m} a_j}{\sum_{j=1}^{q} a_j}, \qquad x = m/q. \tag{48}$$

Sensitivity, panel (a). A rapid rise in C_s means that leading historical directions capture much of the sensitivity *within the stored subspace*. At stages 2–5, there are respectively 1, 2, 3, and 4 old tasks, yielding $72(1+2+3+4) = 720$ layer–stage–old-task observations. The median Spearman and Pearson correlations between λ_j and s_j are 0.589 and 0.907, respectively. The association concerns only directions within the stored subspace. It does not measure how much of the full old-task gradient lies outside that subspace, which is represented by $\delta_{k,K}$.

Subspace overlap, panel (b). A slow initial rise in C_a means that little of the measured overlap lies in high-energy historical directions. The median energy–overlap Spearman correlation is -0.956 for SAMC and $+0.221$ for Uniform. For a fixed stage and layer, overlap does not depend on which old task is being evaluated. We therefore count each stage–layer pair once, giving $4 \times 72 = 288$ observations per method.

Bound comparison, panel (c). Each R_{bank} is computed with the model, old-task gradient, and current update held fixed, changing only the comparison metric. SAMC-trained states have median ratio 0.333, with $720/720$ below one; Uniform-trained states have median 2.311, with $4/720$ below one. These ratios compare bounds on the same first-order loss term within each model state. They do not directly measure accuracy retention or bound the full loss change, which also includes the residual, shared drift, and smoothness terms in Theorem 1.