ALME: ADAPTIVE LOW-RANK MODEL EDITING FOR MULTI-DOMAIN FEW-SHOT SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In real-world scenarios, segmentation models often need to adapt continually to emerging specialized domains with only a few annotated examples. However, segmentation foundation models often perform poorly in such domains because they lack the required domain-specific knowledge. Continual adaptation can acquire this knowledge from limited data, but existing methods typically update the model sequentially across domains, which can interfere with previously acquired knowledge and lead to *catastrophic forgetting*. To address this problem, we propose **A**daptive **L**ow-Rank **M**odel **E**diting (**ALME**), which decouples domain-specific knowledge acquisition from cross-domain knowledge accumulation. ALME acquires knowledge for each domain independently from the same pretrained model through adaptive low-rank updates, thereby preventing adaptation to a new domain from directly overwriting knowledge acquired from earlier domains. It then calibrates and accumulates the acquired domain knowledge and integrates it with regularization anchored to the pretrained model, preserving previously acquired knowledge and pretrained capabilities while yielding a single deployable model. Experiments with SAM3 across 13 sequential segmentation domains in the 5-shot setting show that ALME achieves 76.81% mIoU, outperforming the strongest non-rehearsal continual learning baseline by 4.92 points.

1 INTRODUCTION

Segmentation foundation models (Kirillov et al., 2023; Ravi et al., 2024; Carion et al., 2026) provide strong general-purpose segmentation capabilities across diverse visual applications. However, real-world deployment often requires these models to operate in specialized domains whose visual distributions or segmentation targets differ from those encountered during pretraining, such as medical imaging and remote sensing (Chen et al., 2023; Kaidong & Dong, 2023; Wu et al., 2025; Zhang et al., 2025). Adapting to such domains typically requires domain-specific segmentation annotations, which are costly to obtain and therefore often available only in limited quantities (Ma et al., 2024). This challenge becomes more difficult when specialized domains arrive sequentially rather than being known jointly in advance (Yang et al., 2024; Wang et al., 2026). A practical segmentation model must therefore continually acquire knowledge from each new domain under limited supervision while retaining both previously acquired knowledge and the general segmentation capabilities inherited from pretraining. We formulate this problem as *few-shot continual cross-domain adaptation* for segmentation foundation models.

A natural approach is to adapt the model whenever a new domain arrives. Existing continual adaptation methods typically perform this process sequentially, learning each incoming domain by further updating the model obtained after adapting to preceding domains (Douillard et al., 2021; Zhang et al., 2022). This sequential design forces the same evolving model to serve two roles: preserving knowledge acquired from previous domains and providing the initialization for learning the current domain. Consequently, parameter updates for acquiring new knowledge can directly interfere with parameters that support previously acquired knowledge or pretrained capabilities, potentially causing *catastrophic forgetting* (Kirkpatrick et al., 2017). Conventional sequential adaptation therefore couples two conceptually distinct objectives, *task-wise knowledge acquisition* and *cross-domain knowledge accumulation*, within a single evolving parameter trajectory.

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.

This observation suggests a different perspective: *continual knowledge accumulation does not require repeatedly adapting a continually updated model*. Based on this insight, we propose **A**daptive **L**ow-Rank **M**odel **E**diting (**ALME**), which decouples domain-specific knowledge acquisition from cross-domain knowledge accumulation. ALME follows a three-stage pipeline. It first acquires each incoming domain independently from the same pretrained model W_0, then converts the acquired knowledge into an accumulable representation, and finally integrates the accumulated knowledge into a single deployable model. This design avoids sequential interference during domain acquisition, supports incremental knowledge accumulation without revisiting previous raw data, and removes the need to retain temporary acquisition modules at inference.

For effective knowledge acquisition from limited examples, ALME performs *adaptive low-rank knowledge acquisition*, allocating the adaptation capacity according to the requirements of each domain. Because each domain is learned independently from the same pretrained model W_0, acquiring knowledge for a new domain does not directly modify the knowledge acquired for previous domains.

To accumulate independently acquired knowledge across domains, ALME summarizes the knowledge acquired from each domain using compact regression statistics and calibrates the contribution of each domain before accumulation. These statistics provide an incrementally updatable representation of domain knowledge, enabling cross-domain accumulation without sequentially modifying model parameters or revisiting previous raw data.

Finally, ALME performs *pretrained-anchored knowledge integration*, incorporating the accumulated knowledge into the original model while retaining W_0 as an explicit reference. This stage produces a single model that incorporates knowledge from all observed domains while preserving the general segmentation capabilities inherited from pretraining.

We evaluate ALME with SAM3 across 13 sequential segmentation domains under the 5-shot setting. ALME achieves 76.81% mIoU, outperforming the strongest non-rehearsal baseline by 4.92 points while retaining strong performance on the pretrained-domain evaluation. We further evaluate retention of previously acquired and pretrained capabilities, sensitivity to domain order, scalability as the task sequence grows, and the effects of adaptive rank allocation and cross-domain statistic calibration. These results show that separating domain-specific acquisition from cross-domain integration provides an effective alternative to continual adaptation through sequential parameter updates.

Our contributions are summarized as follows:

- We formulate few-shot continual cross-domain adaptation for segmentation models and identify the coupling between domain-specific knowledge acquisition and cross-domain knowledge accumulation as a key source of interference in sequential adaptation.
- We propose ALME, which decouples these two processes by independently acquiring each domain from a shared pretrained model and integrating the acquired knowledge into a single deployable model without retaining previous raw data.
- Our ALME achieves strong segmentation performance while effectively preserving previously acquired knowledge and pretrained capabilities. Experimental results across 13 cross-domain datasets demonstrate the effectiveness of the proposed method.

2 Related Work

Continual Adaptation for Segmentation Foundation Models. Continual semantic segmentation aims to continually incorporate new knowledge while retaining learned segmentation capabilities. Early studies mainly consider class-incremental settings. MiB (Cermelli et al., 2020) explicitly models the background class's semantic shift, while PLOP (Douillard et al., 2021) combines multi-scale feature distillation with pseudo-labeling to preserve old classes. RCIL (Zhang et al., 2022) further introduces representation compensation and evaluates both continual class and domain segmentation. As segmentation foundation models emerge, recent works begin studying continual adaptation of SAM. CoSAM (Yang et al., 2024) introduces a continual SAM benchmark and uses a mixture of domain adapters, while SAMCL (Wang et al., 2026) learns task-specific modules with a module selector for dynamic domains. These methods mainly adapt continually by preserving or selecting task/domain-specific knowledge during sequential learning. We instead investigate a different

knowledge accumulation scheme: we independently adapt each domain around the same pretrained model and subsequently integrate extracted task knowledge into one model.

Parameter-Efficient Adaptation. Parameter-efficient fine-tuning adapts large pretrained models while updating only a small fraction of parameters. Approaches include adapters (Houlsby et al., 2019), visual prompt tuning (Jia et al., 2022), AdaptFormer (Chen et al., 2022), and Low-Rank Adaptation (LoRA) (Hu et al., 2022). Beyond fixed low-rank budgets, AdaLoRA (Zhang et al., 2023) adaptively allocates parameter budgets across weight matrices by importance. PEFT is also widely explored for SAM adaptation. SAM-Adapter (Chen et al., 2023) injects domain-specific information through lightweight adapters, SAMed (Kaidong & Dong, 2023) applies LoRA to medical image segmentation, and Med-SA (Wu et al., 2025) introduces adapter-based adaptation for 2D and 3D medical images. These works primarily target efficient pretrained model adaptation to a target domain. In ALME, low-rank adaptation is instead the task-wise knowledge acquisition stage: brief gradient probing determines adaptation capacity distribution across layers before task statistics are extracted for later integration.

Model Editing and Knowledge Integration. Model editing studies modifying specific knowledge or model behaviors while limiting unintended changes to unrelated predictions (Mitchell et al., 2022a;b). Related work investigates the internal representations supporting such edits. Geva et al. (2021) show that feed-forward layers in transformer language models can be interpreted as key-value memories. Based on localized factual associations, ROME (Meng et al., 2022) performs rank-one weight updates, and MEMIT (Meng et al., 2023) extends direct editing to many memories. Newer methods such as AlphaEdit (Fang et al., 2025) further study knowledge preservation under sequential editing. Knowledge integration is also closely related to model merging, which combines independently adapted models into one parameterization. Task Arithmetic (Ilharco et al., 2023) represents task-specific changes as parameter-space vectors, while TIES-Merging (Yadav et al., 2023) and DARE (Yu et al., 2024) reduce interference or redundancy among such updates. Particularly relevant here, RegMean (Jin et al., 2023) formulates merging independently fine-tuned models as layer-wise regression and derives closed-form weights using activation inner-product statistics. Building on regression-based model merging, ALME calibrates domain statistics and integrates the acquired layer responses through pretrained-regularized affine regression. Its distinction lies in the continual acquisition–integration procedure: each incoming domain is acquired from the same pretrained anchor, summarized as sufficient relation statistics, and integrated into an updated model after every stage.

3 METHOD

3.1 PROBLEM FORMULATION AND OVERVIEW

Let $f(\cdot; W_0)$ denote a pretrained segmentation model and $\{\mathcal{D}_t\}_{t=1}^{T}$ a sequence of sequentially arriving few-shot domains. At stage t, only labeled samples from the current domain $\mathcal{D}_t$ are available, while raw samples from previous domains are not retained. The goal is a single model after each stage that performs well on all observed domains while preserving the pretrained segmentation capabilities.

As Figure 1 shows, ALME separates domain-specific knowledge acquisition from cross-domain integration. Each domain is acquired independently from W_0 rather than the previously integrated model, isolating domain acquisition from earlier parameter updates. The method has three stages. First, a short task-specific probe allocates a fixed low-rank budget across editable layers. We then discard the probe and restore W_0; we train temporary low-rank adapters on the current domain and summarize the resulting layer-wise input–output relations as compact regression statistics. Second, each domain's statistics are normalized by their total input-feature energy before entering running aggregates, reducing sensitivity to statistic-scale differences. Finally, we fit shared affine projections to accumulated relations with regularization toward W_0. The resulting closed-form solution produces one deployable model W_t^* without retaining domain-specific adapters, so cross-domain knowledge interacts only during integration.

3.2 TASK-WISE KNOWLEDGE ACQUISITION

Let $\mathcal{L}$ denote the set of editable feed-forward output projections. Both the probe and low-rank adaptation optimize the SAM3 mask objective (Carion et al., 2026).

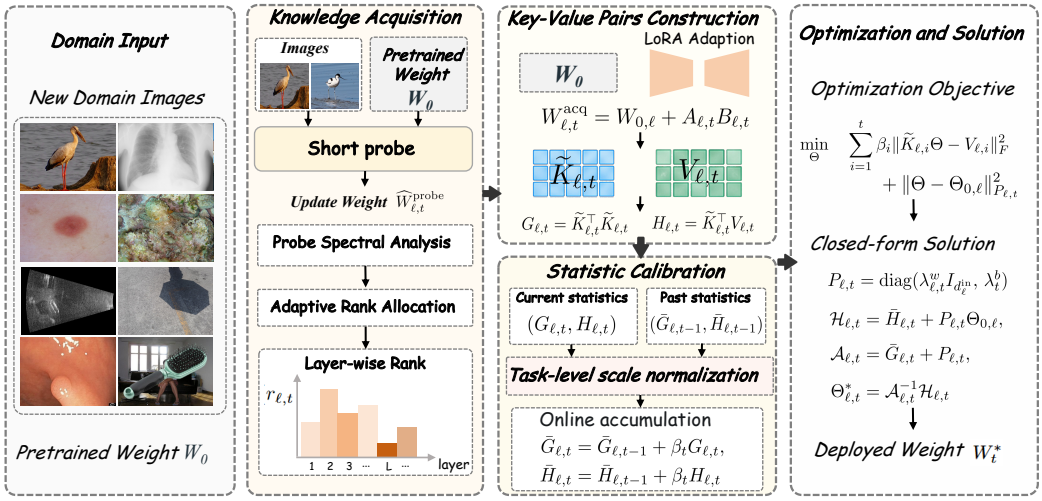

Figure 1: ALME overview. Each incoming domain is temporarily acquired from the fixed pretrained anchor W_0. We calibrate and accumulate relation statistics $S_t = \{(G_{\ell,t}, H_{\ell,t})\}_{\ell\in\mathcal{L}}$ online; their aggregates and W_0 produce one deployed model W^*_t after every stage.

Probe-guided rank allocation. Under a fixed adaptation budget, rank allocation determines each layer's capacity for learning the current domain. A short probe estimates task-specific parameter changes to allocate ranks by retainable spectral energy. Starting from W_0, the probe updates only editable projection weights on $\mathcal{D}_t$ for a few gradient steps, with their biases and all other parameters fixed. For the probed weight $\widehat{W}^{\text{probe}}_{\ell,t}$, define

$$\Delta W^{\text{probe}}_{\ell,t} = \widehat{W}^{\text{probe}}_{\ell,t} - W_{0,\ell}, \qquad \mathcal{R}_{\ell,t}(r) = \sum_{k=1}^{r} \sigma^2_{\ell,t,k}, \tag{1}$$

where $\sigma_{\ell,t,1} \geq \sigma_{\ell,t,2} \geq \cdots$ are the singular values of the probe update. By the Eckart–Young–Mirsky theorem (Eckart & Young, 1936; Mirsky, 1960), $\mathcal{R}_{\ell,t}(r)$ is the energy retained by the best rank-r approximation in the Frobenius norm. Increasing the rank from r to $r+1$ therefore yields a marginal energy gain of $\sigma^2_{\ell,t,r+1}$. We compute only the leading singular values required by the admissible rank range.

Each additional rank introduces $d^{\text{out}}_\ell + d^{\text{in}}_\ell$ trainable parameters for a $d^{\text{out}}_\ell \times d^{\text{in}}_\ell$ projection. We allocate ranks within groups $\mathcal{G}_g \subseteq \mathcal{L}$ of same-shaped projections with equal parameter cost per rank increment. A mean rank budget B determines each group's total low-rank capacity. We allocate this budget to maximize retained probe energy:

$$\max_{\{r_{\ell,t}\}_{\ell\in\mathcal{G}_g}} \sum_{\ell\in\mathcal{G}_g} \mathcal{R}_{\ell,t}(r_{\ell,t}) \qquad \text{s.t.} \quad \sum_{\ell\in\mathcal{G}_g} r_{\ell,t} = |\mathcal{G}_g|B, \quad r_{\min} \leq r_{\ell,t} \leq r_{\max}, \quad r_{\ell,t} \in \mathbb{Z}. \tag{2}$$

$r_{\min}$ and $r_{\max}$ bound each layer's rank. Starting from $r_{\ell,t} = r_{\min}$ for all layers, we assign each remaining rank to the eligible layer with the largest marginal energy gain, subject to $r_{\max}$. For a feasible budget, this greedy procedure is optimal because marginal gains are non-increasing per layer and have equal cost per group.

Low-rank acquisition. The probe is used only for rank selection. After fixing the ranks, we discard the probe weights, restore W_0, and adapt to $\mathcal{D}_t$ through

$$W^{\text{acq}}_{\ell,t} = W_{0,\ell} + A_{\ell,t}B_{\ell,t}, \qquad A_{\ell,t} \in \mathbb{R}^{d^{\text{out}}_\ell \times r_{\ell,t}}, \quad B_{\ell,t} \in \mathbb{R}^{r_{\ell,t}\times d^{\text{in}}_\ell}. \tag{3}$$

We initialize $B_{\ell,t} = 0$, so acquisition starts exactly from W_0. Only the low-rank factors are optimized, while the base weights and biases remain frozen.

Regression statistic extraction. To integrate the independently acquired models through a shared affine fit, we retain their layer-wise input–output relations rather than their adapters. For layer ℓ, let $K_{\ell,t} \in \mathbb{R}^{M_{\ell,t}\times d^{\text{in}}_\ell}$ and $V_{\ell,t} \in \mathbb{R}^{M_{\ell,t}\times d^{\text{out}}_\ell}$ contain the acquired projection's observed inputs and

corresponding outputs, respectively, where $M_{\ell,t}$ denotes the feature token count. Augmenting the inputs as $\widetilde{K}_{\ell,t} = [K_{\ell,t}, \mathbf{1}]$ accounts for affine bias and yields

$$G_{\ell,t} = \widetilde{K}_{\ell,t}^\top \widetilde{K}_{\ell,t}, \qquad H_{\ell,t} = \widetilde{K}_{\ell,t}^\top V_{\ell,t}, \qquad S_t = \{(G_{\ell,t}, H_{\ell,t})\}_{\ell \in \mathcal{L}}. \tag{4}$$

The Gram matrix $G_{\ell,t}$ captures the second-order structure of the augmented inputs, while $H_{\ell,t}$ captures the input–output cross-products. Together, these statistics determine each domain's contribution to the affine least-squares normal equations and suffice to compute the fitted parameters. We accumulate the products over mini-batches without retaining the full feature matrices. After extraction, we discard the temporary acquisition model and pass S_t to calibration.

3.3 Cross-domain Statistic Calibration

The extracted statistic magnitudes depend on observed feature counts and norms, which can distort domain contributions under direct aggregation. For example, duplicating a domain's input–output pairs doubles both $G_{\ell,t}$ and $H_{\ell,t}$, increasing its regression weight without new information. We therefore normalize each domain's statistics by their total input-feature energy to reduce sensitivity to statistic scale.

Let $G^{xx}_{\ell,t} = K_{\ell,t}^\top K_{\ell,t}$ denote the non-bias block of $G_{\ell,t}$. Its trace satisfies $\mathrm{tr}(G^{xx}_{\ell,t}) = \|K_{\ell,t}\|_F^2$. Summing this quantity over editable layers measures the domain's total input energy while excluding the constant bias coordinate. We use this energy to rescale incoming statistics and update the running aggregates:

$$E_t = \sum_{\ell \in \mathcal{L}} \mathrm{tr}(G^{xx}_{\ell,t}), \quad \beta_t = \frac{1}{E_t + \epsilon}, \quad \overline{G}_{\ell,t} = \overline{G}_{\ell,t-1} + \beta_t G_{\ell,t}, \quad \overline{H}_{\ell,t} = \overline{H}_{\ell,t-1} + \beta_t H_{\ell,t}, \tag{5}$$

with zero-initialized aggregates and a small constant ϵ for numerical stability. Applying the same β_t to both $G_{\ell,t}$ and $H_{\ell,t}$ is equivalent to weighting the domain's least-squares contribution by β_t. Using a single coefficient across all layers adjusts the domain's overall contribution while preserving relative statistic scales across layers. The running aggregates retain the calibrated relations from all observed domains, after which individual domain statistics can be discarded.

3.4 Regularized Knowledge Integration

For each editable layer, we fit a single shared affine map to the acquired relations while penalizing deviations from the pretrained parameters. Whereas acquisition freezes the biases and learns low-rank weight updates, integration jointly fits the weight and bias of each projection. Let $\Theta_\ell = [W_\ell^\top;\, b_\ell^\top] \in \mathbb{R}^{(d_\ell^{\mathrm{in}}+1)\times d_\ell^{\mathrm{out}}}$ and $\Theta_{0,\ell} = [W_{0,\ell}^\top;\, b_{0,\ell}^\top]$ denote the affine parameters and their pretrained values, respectively. We solve

$$\Theta^*_{\ell,t} = \arg\min_{\Theta} \sum_{i=1}^{t} \beta_i \left\| \widetilde{K}_{\ell,i}\Theta - V_{\ell,i} \right\|_F^2 + \|\Theta - \Theta_{0,\ell}\|^2_{P_{\ell,t}}, \tag{6}$$

where $\|A\|_P^2 = \mathrm{tr}(A^\top P A)$. The data term fits the acquired projection outputs using calibrated domain weights, while the regularizer anchors the shared parameters to their pretrained values. The matrix $P_{\ell,t}$ controls separate penalties on weight and bias changes. We optimize each editable projection independently.

We scale the weight penalty by the mean diagonal value of the accumulated non-bias Gram matrix, $\lambda^w_{\ell,t} = \lambda\ \mathrm{tr}(\overline{G}^{xx}_{\ell,t})/d_\ell^{\mathrm{in}}$. The bias penalty is shared across layers, $\lambda^b_t = \lambda_b/\mu_t$, where

$$\mu_t = \exp\left(\frac{1}{t}\sum_{i=1}^{t} \log(E_i + \epsilon)\right), \qquad P_{\ell,t} = \mathrm{diag}\left(\lambda^w_{\ell,t} I_{d_\ell^{\mathrm{in}}}, \lambda^b_t\right). \tag{7}$$

λ and λ_b control regularization strengths, and μ_t is the geometric mean of the stabilized domain energies. Computing μ_t requires only a running sum of log energies and the domain count t, without retaining individual domain energies.

Although the objective uses input–output pairs from all observed domains, all parameter-dependent data terms are fully determined by $\overline{G}_{\ell,t}$ and $\overline{H}_{\ell,t}$. Historical samples and feature pairs are therefore unnecessary for computing the solution. The normal equations yield

$$\Theta^*_{\ell,t} = \left(\overline{G}_{\ell,t} + P_{\ell,t}\right)^{-1} \left(\overline{H}_{\ell,t} + P_{\ell,t}\Theta_{0,\ell}\right). \tag{8}$$

In practice, we solve this linear system without explicit inversion. Writing fitted weights and biases back to their original projections produces a single deployed model W_t^* without retaining domain-specific acquisition modules. Subsequent integration requires only the running matrix aggregates, the log-energy sum, the domain count, and the fixed pretrained reference. For fixed acquired statistics, both the calibrated aggregates and the log-energy sum are independent of domain accumulation order; the integrated solution is therefore also order-independent.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets and Protocol. We evaluate ALME on 13 segmentation datasets from several specialized domains: *medical imaging*, including ISIC (Codella et al., 2018), Kvasir (Jha et al., 2020), and JSRT (Shiraishi et al., 2000); *remote sensing*, including HRSID (Wei et al., 2020); *camouflaged and salient object segmentation*, including CAMO (Le et al., 2019), COD (Fan et al., 2020), DIS (Qin et al., 2022), HRSOD (Zeng et al., 2019), and COIFT (Mansilla & Miranda, 2019); and *shadow and structural segmentation*, including ISTD (Wang et al., 2018), SBU (Vicente et al., 2016), FLS (Singh & Valdenegro-Toro, 2021), and ThinObject (Liew et al., 2021). We follow the domain-incremental continual adaptation protocol of SAMCL (Wang et al., 2026), where target domains arrive sequentially.

We extend this protocol to the few-shot setting by restricting each incoming domain to a K-shot support set, where each shot is one labeled image–mask pair. We use three positive point prompts sampled within the ground-truth mask during both adaptation and evaluation, with identical prompt protocols for all compared methods. Previous-domain data are unavailable to non-rehearsal methods, while rehearsal baselines use their prescribed replay buffers. We evaluate the 1-shot, 5-shot, and 10-shot settings using three support splits and report the mean and standard deviation. Unless otherwise stated, detailed continual and diagnostic analyses use the 5-shot setting. We additionally evaluate COCO (Lin et al., 2014) as a general-domain reference to track how continual adaptation affects general segmentation performance. Target performance is the macro average over the 13 target datasets. Dataset statistics, support construction, and further protocol details are provided in the supplementary material.

Baselines and Evaluation. We compare ALME with four continual adaptation method groups: sequential fine-tuning (SeqLoRA FT, fine-tuning LoRA modules sequentially across domains), rehearsal (ER (Rolnick et al., 2019), DER++ (Buzzega et al., 2020)), regularization (EWC (Kirkpatrick et al., 2017)), and other continual methods (O-LoRA (Wang et al., 2023), SPPA (Lin et al., 2022), LAG (Yuan et al., 2024), MoDA (Yang et al., 2024), SAMCL (Wang et al., 2026), RegCL (chen Shu et al., 2026)). Pristine SAM3 and independent fine-tuning (Indep. FT, keeping a separate model per domain) serve as references.

Table 1 reports target-domain mIoU, F1, and MAE at 1, 5, and 10 shots. In the 5-shot continual analysis (Table 2), we also report average incremental accuracy (AIA) (Rebuffi et al., 2017), backward and forward transfer (BWT, FWT) (Lopez-Paz & Ranzato, 2017), and COCO mIoU as a general-domain reference.

To isolate the integration step, we also apply RegMean (Jin et al., 2023), Weight Averaging, Task Arithmetic (Ilharco et al., 2023), TIES-Merging (Yadav et al., 2023), and DARE (Yu et al., 2024) to the same task acquisitions as ALME (Sec. 4.3).

Implementation Details. All main experiments use SAM3 as the backbone. ALME uses the editable feed-forward output projections defined in Sec. 3.2, with a default mean LoRA rank budget of $B = 8$. Other implementation details can be found in the supplementary material.

4.2 MAIN RESULTS

Matched few-shot comparison. Table 1 compares ALME with continual adaptation methods under matched 1-, 5-, and 10-shot protocols. ALME leads continual methods in mIoU and F1 at all three shot counts, reaching 71.75, 76.81, and 78.38 mIoU. It outperforms the strongest non-rehearsal baselines by 4.61, 4.92, and 4.63 points, and surpasses the strongest rehearsal-based baseline, DER++, by 1.44, 1.91, and 1.25 points, respectively. Its MAE is best at 10-shot and competitive at 1- and 5-shot. ALME remains close to independent few-shot fine-tuning, in the stricter continual setting.

Table 1: Continual few-shot adaptation performance across **13** diverse target domains based on SAM3. Entries report means over three support splits. The best results among reported continual methods per setting are highlighted in **bold**. Superscripts denote method types: ^{a} Sequential FT, ^{b} Rehearsal-based, ^{c} Regularization, ^{d} Others. Pristine SAM3 and independent fine-tuning serve as references.

Task Sequence: ISIC → ISTD → Kvasir → CAMO → COD → COIFT → DIS → FLS → HRSID → HRSOD → JSRT → SBU → ThinObject

Method	1-shot			5-shot			10-shot		
	mIoU (%) ↑	F1 (%) ↑	MAE ↓	mIoU (%) ↑	F1 (%) ↑	MAE ↓	mIoU (%) ↑	F1 (%) ↑	MAE ↓
Base Model: SAM3									
Pristine SAM3 (0-shot)	62.37	71.69	0.1143	62.37	71.69	0.1143	62.37	71.69	0.1143
Indep. FT	70.94	79.91	0.0588	77.93	85.35	0.0440	80.71	87.95	0.0282
SeqLoRA FT^{a}	53.49	64.05	0.1846	62.60	72.31	0.1202	61.92	71.74	0.1209
ER^{b}	69.63	79.20	0.0527	74.55	83.25	0.0392	75.35	84.00	0.0383
DER++^{b}	70.31	79.58	**0.0480**	74.90	83.41	**0.0368**	77.13	85.32	0.0339
EWC^{c}	59.49	70.03	0.1170	65.43	75.49	0.0801	67.19	77.28	0.0506
O-LoRA^{d}	53.26	64.49	0.1478	64.18	74.02	0.1000	64.70	74.56	0.0956
SPPA^{d}	41.12	53.69	0.1670	63.47	73.77	0.0841	63.55	73.79	0.0841
LAG^{d}	56.10	67.14	0.1382	62.86	73.51	0.0787	63.76	74.16	0.0787
MoDA^{d}	64.63	74.00	0.0914	67.95	77.23	0.0679	71.18	81.54	0.0508
SAMCL^{d}	63.85	73.85	0.0782	71.89	81.11	0.0517	73.75	82.67	0.0462
RegCL^{d}	67.14	76.68	0.0628	68.36	78.04	0.0523	67.80	77.68	0.0579
ALME (Ours)	**71.75**	**80.15**	0.0569	**76.81**	**84.66**	0.0386	**78.38**	**86.08**	**0.0336**

Table 2: Continual performance under matched **5-shot** evaluation across 13 sequential target domains with SAM3. Metrics show mean ± standard deviation over three support splits; COCO denotes COCO mIoU after the last stage.

Method	AIA (%) ↑	BWT (%) ↑	FWT (%) ↑	Final Target (%) ↑	COCO (%) ↑
Pristine SAM3 (0-shot)	–	–	–	62.37	51.47
SeqLoRA FT^{a}	69.23 ± 1.29	-10.44 ± 3.23	-4.93 ± 1.01	62.60 ± 2.66	55.04 ± 3.60
ER^{b}	78.23 ± 0.40	-1.42 ± 0.64	1.63 ± 0.43	74.55 ± 0.67	56.24 ± 1.35
DER++^{b}	**79.43** ± 1.23	-1.15 ± 0.85	1.47 ± 0.59	<u>74.90</u> ± 1.11	55.70 ± 2.05
EWC^{c}	65.22 ± 2.28	-12.71 ± 2.66	-8.57 ± 1.80	65.43 ± 1.58	57.13 ± 1.52
O-LoRA^{d}	72.66 ± 4.36	-8.75 ± 3.57	-1.47 ± 4.79	64.18 ± 2.03	57.03 ± 3.05
SPPA^{d}	65.81 ± 1.90	-15.06 ± 6.23	-6.61 ± 1.85	63.47 ± 5.19	<u>57.24</u> ± 1.19
LAG^{d}	63.67 ± 1.26	-13.30 ± 3.55	-10.28 ± 1.71	62.86 ± 2.75	56.53 ± 3.52
MoDA^{d}	73.58 ± 0.45	<u>0.51</u> ± 0.58	1.59 ± 0.17	67.95 ± 0.50	53.04 ± 0.60
SAMCL^{d}	76.74 ± 0.78	-0.74 ± 0.29	**3.22** ± 0.89	71.89 ± 0.66	56.81 ± 0.73
RegCL^{d}	76.42 ± 0.65	-1.95 ± 0.58	<u>1.70</u> ± 1.42	68.36 ± 1.56	52.15 ± 1.75
ALME (Ours)	<u>78.56</u> ± 2.90	**2.71** ± 1.31	-0.73 ± 2.57	**76.81** ± 0.74	**57.44** ± 0.71

Continual evaluation. Table 2 reports sequence-level metrics for 5-shot evaluation. While most baselines show negative BWT, e.g., SeqLoRA FT (-10.44) and SPPA (-15.06), ALME achieves the highest BWT (2.71 ± 1.31 percentage points) and final target mIoU (76.81 ± 0.74). This suggests later domains can refine shared representations without overwriting earlier knowledge, likely because calibrated cross-domain aggregation under a pretrained anchor regularizes noisy few-shot updates. ALME's AIA (78.56 ± 2.90) is competitive with DER++ (79.43 ± 1.23), its FWT remains near zero (-0.73 ± 2.57 percentage points), and it achieves the highest COCO mIoU (57.44 ± 0.71), surpassing pristine SAM3 (51.47). These results indicate ALME improves final performance, maintaining stable continual behavior and COCO reference performance.

4.3 Analysis of ALME

We first examine the two choices underlying ALME's acquisition–integration formulation: where task knowledge is acquired and how independently acquired relations are integrated. We then analyze spectral budget allocation and cross-domain scale calibration. Unless otherwise stated, all comparisons use the matched 5-shot setting.

Table 3: Acquisition reference effect (5-shot). Current-domain results are averaged over stages 2–13; final results are averaged over all 13 target domains. All values are mIoU (%).

Evaluation ↑	**Evolving** W^*_{t-1}	**Pretrained** W_0
Current domain		
Before integration	61.48 ± 9.07	$\mathbf{77.43 \pm 1.24}$
After integration	51.89 ± 13.14	$\mathbf{74.41 \pm 2.24}$
All target domains		
Final stage	57.99 ± 11.59	$\mathbf{76.81 \pm 0.74}$

Table 4: Comparison of integration strategies using the same independently acquired task adaptations under the 5-shot setting. Both metrics are mIoU (%).

Integration	**Target** ↑	**COCO** ↑
Weight Averaging	68.79	55.47
Task Arithmetic	69.07	56.52
TIES-Merging	63.72	53.57
DARE	68.32	55.41
RegMean	71.51	54.97
ALME	**76.81**	**57.44**

Table 5: Probe-guided spectral allocation under matched low-rank budgets. Δ denotes target-mIoU gain over uniform.

Rank B	**Uniform**	**Spectral**	**Δ**
2	71.18	72.00	+0.82
4	73.19	75.24	+2.05
8	74.86	76.81	+1.95
16	76.49	76.96	+0.47

Table 6: Ablation of task-level statistic calibration under the 5-shot setting, using identical task acquisitions.

Calibration	**Target mIoU (%)** ↑
None ($\beta_i = 1$)	72.54 ± 0.71
Inverse-trace (ALME)	$\mathbf{76.81 \pm 0.74}$

Acquisition reference. ALME acquires each incoming domain from the shared pretrained model W_0, rather than the evolving deployed model W^*_{t-1}. Table 3 compares these acquisition references within the same acquisition–integration pipeline. Excluding the first stage, acquisition from W_0 averages 77.43 acquired-task mIoU, versus 61.48 from the evolving model. The gap persists after integration: the corresponding current-task mIoUs are 74.41 and 51.89, and the final target mIoUs are 76.81 and 57.99, respectively. This indicates degradation already appears during task acquisition, supporting a common pretrained reference in ALME.

Integration strategy. We next isolate the integration step by applying different strategies to the same independently acquired task adaptations. As Table 4 shows, direct parameter-space merging yields target mIoUs between 63.72 and 69.07, while RegMean improves to 71.51. ALME reaches 76.81 target mIoU and 57.44 COCO mIoU, the highest among the compared methods. This shows that the integration step contributes substantially beyond individual task acquisition quality.

Spectral rank allocation. We examine whether the spectral probe provides useful task-conditioned guidance for distributing a fixed low-rank acquisition budget across layers. For each mean rank budget $B \in \{2, 4, 8, 16\}$, spectral and uniform allocation use equal trainable parameter counts and differ only in rank distribution across layers. In Table 5, spectral allocation consistently outperforms uniform allocation across all tested budgets, improving mIoU by 0.47 to 2.05 points, with a 1.95-point gain at the default budget $B = 8$. The smaller gain at $B = 16$ suggests the probe helps most under tight rank budgets.

Scale calibration. We ablate task-level scale calibration in Eq. (5) under the matched 5-shot setting. We compare direct aggregation ($\beta_i = 1$) with inverse-trace normalization ($\beta_i = (E_i + \epsilon)^{-1}$), keeping task acquisition, extracted statistics, and integration solver identical. As Table 6 reports, direct aggregation yields 72.54 ± 0.71 target mIoU, whereas inverse-trace normalization achieves 76.81 ± 0.74, a gain of 4.27 points. This suggests that different domains produce differently scaled statistics, and normalizing these scales improves cross-domain integration.

Regularization strength. Table 7 evaluates sensitivity to the integration regularization coefficient λ, with task acquisition and scale calibration fixed. Removing the weight regularizer ($\lambda = 0$) reduces target mIoU to 6.54, and at $\lambda = 0.01$ it is only 55.84, indicating that the pretrained weight anchor is necessary. For $0.05 \leq \lambda \leq 1.0$, performance remains stable, ranging from 76.34 to 76.81 target mIoU. The default $\lambda = 0.1$ lies in this stable range, indicating insensitivity to the exact λ value

Table 7: Effect of integration regularization coefficient λ under the 5-shot setting. Target mIoU is averaged over three support splits. The default setting is shaded.

λ	0	0.01	0.05	**0.1**	0.2	1.0
Target mIoU (%)	6.54	55.84	76.34	**76.81**	76.73	76.45

within this interval. A likely reason is that few-shot feature statistics are often low-rank, so the pretrained anchor helps regularize underconstrained directions during integration.

4.4 Continual Integration Properties

We next examine three properties of ALME's continual integration: final-state accumulation order, running state updates, and deployment.

Final-state accumulation order. Figure 2 compares final target performance across five domain orders. While the evaluated baselines exhibit order-dependent variation, ALME achieves 76.81% mIoU under all five orders. This consistency follows from its acquisition–integration formulation: each domain is acquired from the same pretrained anchor, and integration depends on order-independent aggregates of acquired statistics. For fixed task statistics, changing their accumulation order leaves the final solution unchanged. Intermediate performance remains order-dependent because different orders expose the model to different domain prefixes.

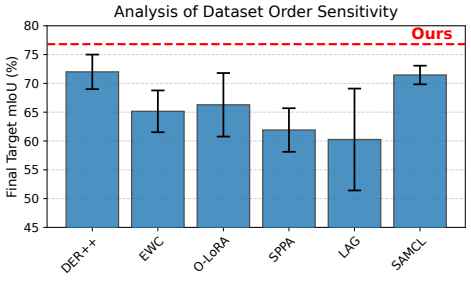

Figure 2: Sensitivity of final target performance to domain order. Bars show mean final target mIoU over five domain orders; error bars indicate one standard deviation. The red dashed line denotes ALME (76.81%).

Running integration. The additive form of Eq. (5) lets ALME update a running relation state without retaining and reaggregating all previous task statistics. Under the matched 5-shot protocol, running aggregation and cached-prefix recomputation both produce 76.81 target mIoU and 57.44 COCO mIoU. Across three support splits, the running solution matches the formal reaggregation reference within a maximum relative Frobenius difference of 1.39×10^{-10}. Persistent integration state size is independent of observed domain count; the supplement details storage and implementation costs.

Deployment. After each running update, we can discard temporary acquisition artifacts and individual task statistics. We write the integrated affine parameters directly into the original SAM3 projections, yielding one standard model without task-specific acquisition modules or domain-dependent routing at inference. Deployment inference over all three support splits reproduces ALME's final 76.81 target mIoU and 57.44 COCO mIoU.

5 Conclusion

We formulated few-shot continual segmentation adaptation as two decoupled operations: task-wise knowledge acquisition from a shared pretrained anchor, and cross-domain knowledge integration into one deployed model. Our instantiation, ALME, converts temporary low-rank domain adaptation into calibrated layer-wise regression statistics and integrates them through closed-form, pretrained-anchored regression, requiring neither retained raw samples nor domain-specific inference modules at deployment. Across 13 sequential segmentation domains with SAM3, ALME outperforms the strongest non-rehearsal continual baseline by 4.92 mIoU points under 5-shot adaptation. Supplementary evaluations cover additional segmentation foundation architectures, and we expect the decoupled formulation to offer a practical, general foundation for cross-domain continual learning.

AI USE STATEMENT

Generative AI tools were used primarily to refine the writing and presentation of the manuscript, with limited assistance in code editing and testing. All AI-assisted revisions and code changes were reviewed and verified by the authors. The authors take responsibility for the content, results, and conclusions of this work.

REFERENCES

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *NeurIPS*, 2020.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International Conference on Learning Representations*, volume 2026, pp. 138846–138923, 2026.

Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9230–9239. IEEE, 2020.

Shoufa Chen, Chongjian GE, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 16664–16678. Curran Associates, Inc., 2022. doi: 10.52202/068431-1212. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/69e2f49ab0837b71b0e0cb7c555990f8-Paper-Conference.pdf.

Tianrun Chen, Lanyun Zhu, Chaotao Deng, Runlong Cao, Yan Wang, Shangzhan Zhang, Zejian Li, Lingyun Sun, Ying Zang, and Papa Mao. Sam-adapter: Adapting segment anything in underperformed scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 3367–3375, October 2023.

Yuan chen Shu, Zhiwei Lin, Xiaoyu Zhou, and Yongtao Wang. RegCL: Compact continual SAM adaptation for visual grounding in multi-sensorial media. In *2nd International Workshop on Multi-Sensorial Media and Applications*, 2026. URL https://openreview.net/forum?id=T4BfyftRTf.

Noel Codella, Veronica Rotemberg, Philipp Tschandl, M Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*, 2018.

Arthur Douillard, Yifu Chen, Arnaud Dapogny, and Matthieu Cord. Plop: Learning without forgetting for continual semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4040–4050, June 2021.

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

Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In *CVPR*, 2020.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. Alphaedit: Null-space constrained knowledge editing for language models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 16366–16396, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/29c8c615b3187ee995029284702d3f43-Paper-Conference.pdf.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.446. URL https://aclanthology.org/2021.emnlp-main.446/.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799. PMLR, 09–15 Jun 2019.

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

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Debesh Jha, Pia H Smedsrud, Michael A Riegler, Pål Halvorsen, Thomas De Lange, Dag Johansen, and Håvard D Johansen. Kvasir-seg: A segmented polyp dataset. In *MMM*, 2020.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European conference on computer vision*, pp. 709–727. Springer, 2022.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=FCnohuR6AnM.

Zhang Kaidong and Liu Dong. Customized segment anything model for medical image segmentation. *arXiv preprint arXiv:2304.13785*, 2023.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4015–4026, October 2023.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *PNAS*, 2017.

Trung-Nghia Le, Tam V. Nguyen, Zhongliang Nie, Minh-Triet Tran, and Akihiro Sugimoto. Anabranch network for camouflaged object segmentation. *CVIU*, 2019.

Jun Hao Liew, Scott Cohen, Brian Price, Long Mai, and Jiashi Feng. Deep interactive thin object selection. In *WACV*, 2021.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision – ECCV 2014*, pp. 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

Zihan Lin, Zilei Wang, and Yixin Zhang. Continual semantic segmentation via structure preserving and projected feature alignment. In *ECCV*, 2022.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images. *Nat Commun*, 15(1):654, January 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-44824-z. URL https://www.nature.com/articles/s41467-024-44824-z.

Lucy AC Mansilla and Paulo AV Miranda. Object segmentation by oriented image foresting transform with connectivity constraints. In *SIBGRAPI*, 2019.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 17359–17372. Curran Associates, Inc., 2022. doi: 10.52202/068431-1262. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/6f1d43d5a82a37e89b0665b33bf3a182-Paper-Conference.pdf.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=MkbcAHIYgyS.

Leonid Mirsky. Symmetric gauge functions and unitarily invariant norms. *Quarterly Journal of Mathematics*, 11(1):50–59, 1960.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/pdf?id=0DcZxeWfOPt.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. Memory-based model editing at scale. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15817–15831. PMLR, 17–23 Jul 2022b. URL https://proceedings.mlr.press/v162/mitchell22a.html.

Xuebin Qin, Hang Dai, Xiaobin Hu, Deng-Ping Fan, Ling Shao, and Luc Van Gool. Highly accurate dichotomous image segmentation. In *ECCV*, 2022.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714.

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

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. In *NeurIPS*, 2019.

Junji Shiraishi, Shigehiko Katsuragawa, Junpei Ikezoe, Tsuneo Matsumoto, Takeshi Kobayashi, Ken-ichi Komatsu, Mitate Matsui, Hiroshi Fujita, Yoshie Kodera, and Kunio Doi. Development of a digital image database for chest radiographs with and without a lung nodule: receiver operating characteristic analysis of radiologists' detection of pulmonary nodules. *American journal of roentgenology*, 2000.

Deepak Singh and Matias Valdenegro-Toro. The marine debris dataset for forward-looking sonar semantic segmentation. In *ICCV*, 2021.

Tomás F Yago Vicente, Le Hou, Chen-Ping Yu, Minh Hoai, and Dimitris Samaras. Large-scale training of shadow detectors with noisily-annotated shadow examples. In *ECCV*, 2016.

Jifeng Wang, Xiang Li, and Jian Yang. Stacked conditional generative adversarial networks for jointly learning shadow detection and shadow removal. In *CVPR*, 2018.

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

Zeqing Wang, Kangye Ji, Di Wang, Haibin Zhang, and Fei Cheng. Samcl: Empowering sam to continually learn from dynamic domains with extreme storage efficiency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 26741–26750, 2026.

Shunjun Wei, Xiangfeng Zeng, Qizhe Qu, Mou Wang, Hao Su, and Jun Shi. Hrsid: A high-resolution sar images dataset for ship detection and instance segmentation. *IEEE Access*, 2020.

Junde Wu, Ziyue Wang, Mingxuan Hong, Wei Ji, Huazhu Fu, Yanwu Xu, Min Xu, and Yueming Jin. Medical SAM adapter: Adapting segment anything model for medical image segmentation. *Medical Image Analysis*, 102:103547, May 2025. ISSN 13618415. doi: 10.1016/j.media.2025.103547. URL https://linkinghub.elsevier.com/retrieve/pii/S1361841525000945.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 7093–7115. Curran Associates, Inc., 2023. doi: 10.52202/075280-0310. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1644c9af28ab7916874f6fd6228a9bcf-Paper-Conference.pdf.

Jinglong Yang, Yichen Wu, Jun Cen, Wenjian Huang, Hong Wang, and Jianguo Zhang. Continual learning for segment anything model adaptation, 2024. URL https://arxiv.org/abs/2412.06418.

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 Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57755–57775. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/yu24p.html.

Bo Yuan, Danpei Zhao, and Zhenwei Shi. Learning at a glance: Towards interpretable data-limited continual semantic segmentation via semantic-invariance modelling. *IEEE TPAMI*, 2024.

Yi Zeng, Pingping Zhang, Jianming Zhang, Zhe Lin, and Huchuan Lu. Towards high-resolution salient object detection. In *ICCV*, 2019.

Chang-Bin Zhang, Jia-Wen Xiao, Xialei Liu, Ying-Cong Chen, and Ming-Ming Cheng. Representation compensation networks for continual semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7053–7064, June 2022.

Enkai Zhang, Jingjing Liu, Anda Cao, Zhen Sun, Haofei Zhang, Huiqiong Wang, Li Sun, and Mingli Song. Rs-sam: Integrating multi-scale information for enhanced remote sensing image segmentation. In Minsu Cho, Ivan Laptev, Du Tran, Angela Yao, and Hongbin Zha (eds.), *Computer Vision – ACCV 2024*, pp. 280–296, Singapore, 2025. Springer Nature Singapore. ISBN 978-981-96-0966-6.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=lq62uWRJjiY.

A Algorithm and Mathematical Derivations

A.1 ALME Detailed Workflow

Algorithm 1 outlines the stage-wise execution of ALME. Each domain is acquired independently from the shared anchor W_0, summarized into sufficient statistics, and seamlessly integrated into the deployed parameters without storing historical raw data.

Algorithm 1 Adaptive Low-Rank Model Editing (ALME)

Require: Pretrained anchor W_0; Sequential stream $\{\mathcal{D}_t\}_{t=1}^{T}$; Mean rank budget B; Regularization coefficients λ, λ_b.
Ensure: Integrated deployed model W_T^*.

Initialize running statistics: $\overline{G}_{\ell,0} \leftarrow \mathbf{0}$, $\overline{H}_{\ell,0} \leftarrow \mathbf{0}$ for all $\ell \in \mathcal{L}$; $\mathcal{S}_0 \leftarrow 0$
for stage $t = 1, \dots, T$ **do**
  // Phase 1: Probe-Guided Spectral Rank Allocation
  $\Delta W_{\ell,t}^{\text{probe}} \leftarrow \text{Probe}(\mathcal{D}_t; W_{0,\ell})$ ▷ Short gradient probe from W_0
  $\{r_{\ell,t}\} \leftarrow \text{GreedyAlloc}(\{\text{SVD}(\Delta W_{\ell,t}^{\text{probe}})\}, B)$ ▷ Spectral allocation via Eq. (2)
  // Phase 2: Independent Acquisition & Relation Extraction
  $A_{\ell,t}, B_{\ell,t} \leftarrow \text{LoRA}(\mathcal{D}_t; W_0, \{r_{\ell,t}\})$ ▷ Adapt allocated low-rank factors from W_0
  $G_{\ell,t} \leftarrow \widetilde{K}_{\ell,t}^{\top}\widetilde{K}_{\ell,t}, \quad H_{\ell,t} \leftarrow \widetilde{K}_{\ell,t}^{\top}V_{\ell,t}$ ▷ Extract layer-wise sufficient statistics
  $E_t \leftarrow \sum_{\ell\in\mathcal{L}} \text{tr}(G_{\ell,t}^{xx}), \quad \beta_t \leftarrow (E_t + \epsilon)^{-1}$ ▷ Compute domain input energy & weight
  // Phase 3: Online Calibration & Regularized Integration
  $\overline{G}_{\ell,t} \leftarrow \overline{G}_{\ell,t-1} + \beta_t G_{\ell,t}, \quad \overline{H}_{\ell,t} \leftarrow \overline{H}_{\ell,t-1} + \beta_t H_{\ell,t}$ ▷ Update running aggregates
  $\mathcal{S}_t \leftarrow \mathcal{S}_{t-1} + \log(E_t + \epsilon), \quad \mu_t \leftarrow \exp(\mathcal{S}_t/t)$ ▷ Update geometric reference scale
  Construct diagonal anchor matrix $P_{\ell,t} \leftarrow \text{diag}(\lambda_{\ell,t}^{w} I_{d_\ell^{\text{in}}}, \lambda_b/\mu_t)$
  $\Theta_{\ell,t}^{*} \leftarrow (\overline{G}_{\ell,t} + P_{\ell,t})^{-1}(\overline{H}_{\ell,t} + P_{\ell,t}\Theta_{0,\ell})$ ▷ Closed-form affine solve (Eq. 8)
  $W_t^{*} \leftarrow \text{Deploy}(W_0, \{\Theta_{\ell,t}^{*}\}_{\ell\in\mathcal{L}})$ ▷ Update editable projections for inference
end for

A.2 Derivation of the Closed-Form Integration (Eq. 8)

The regularized affine integration objective at stage t for editable layer ℓ is formulated in Eq. (6) as:

$$\Theta_{\ell,t}^{*} = \arg\min_{\Theta} \sum_{i=1}^{t} \beta_i \left\| \widetilde{K}_{\ell,i}\Theta - V_{\ell,i} \right\|_F^2 + \|\Theta - \Theta_{0,\ell}\|_{P_{\ell,t}}^2 . \tag{9}$$

Expanding the Frobenius norms via the trace operator:

$$\begin{aligned}
\mathcal{J}(\Theta) &= \sum_{i=1}^{t} \beta_i \,\text{tr}\left((\widetilde{K}_{\ell,i}\Theta - V_{\ell,i})^{\top}(\widetilde{K}_{\ell,i}\Theta - V_{\ell,i}) \right) + \text{tr}\left((\Theta - \Theta_{0,\ell})^{\top} P_{\ell,t} (\Theta - \Theta_{0,\ell}) \right) \\
&= \text{tr}\left(\Theta^{\top} \left(\sum_{i=1}^{t} \beta_i \widetilde{K}_{\ell,i}^{\top}\widetilde{K}_{\ell,i} \right) \Theta - 2\Theta^{\top} \left(\sum_{i=1}^{t} \beta_i \widetilde{K}_{\ell,i}^{\top} V_{\ell,i} \right) \right) \\
&\quad + \text{tr}\left(\Theta^{\top} P_{\ell,t} \Theta - 2\Theta^{\top} P_{\ell,t} \Theta_{0,\ell} \right) + \mathcal{C},
\end{aligned} \tag{10}$$

where $\mathcal{C}$ collects parameter-independent constants. Substituting the definitions of running statistics $\overline{G}_{\ell,t} = \sum_{i=1}^{t} \beta_i G_{\ell,i}$ and $\overline{H}_{\ell,t} = \sum_{i=1}^{t} \beta_i H_{\ell,i}$:

$$\mathcal{J}(\Theta) = \text{tr}\left(\Theta^{\top}(\overline{G}_{\ell,t} + P_{\ell,t})\Theta - 2\Theta^{\top}(\overline{H}_{\ell,t} + P_{\ell,t}\Theta_{0,\ell}) \right) + \mathcal{C}. \tag{11}$$

Setting the matrix gradient with respect to Θ to zero yields the normal equations:

$$2(\overline{G}_{\ell,t} + P_{\ell,t})\Theta - 2(\overline{H}_{\ell,t} + P_{\ell,t}\Theta_{0,\ell}) = \mathbf{0} \implies (\overline{G}_{\ell,t} + P_{\ell,t})\Theta = \overline{H}_{\ell,t} + P_{\ell,t}\Theta_{0,\ell}. \tag{12}$$

Because $\overline{G}_{\ell,t}$ is positive semi-definite and $P_{\ell,t}$ is strictly positive definite by construction ($\lambda > 0, \lambda_b > 0$), the coefficient matrix $(\overline{G}_{\ell,t} + P_{\ell,t})$ is strictly positive definite and invertible. We solve this system using Cholesky factorization in double precision (float64), ensuring stable parameters without explicit matrix inversion.

Table 8: Per-domain adaptation time in the 5-shot setting (seconds, mean over three shared splits). Sum adds the reported GPU and CPU components without assuming overlap; validation and evaluation are excluded.

Method	Spectral Probe (GPU)	Adaptation (GPU)	Affine Solve (CPU)	Sum
SeqLoRA FT	–	84.8	–	84.8
EWC	–	86.7	–	86.7
O-LoRA	–	94.1	–	94.1
ER	–	114.8	–	114.8
DER++	–	115.5	–	115.5
SAMCL	–	130.7	–	130.7
ALME (Ours)	14.2	82.6	20.0	116.8

B Adaptation Cost and Deployment Efficiency

We examine the cost of domain acquisition, statistical accumulation, and model deployment. All benchmarks follow the same protocol, using 13 sequential target domains and three fixed splits (00/01/02) shared across methods. Adaptation runs for 30 epochs per domain with batch size 2, targeting all 34 editable FFN output projections in SAM3. Measurements are collected on a single NVIDIA GeForce RTX 4090 GPU and a dual-socket Intel Xeon Gold 6530 system.

B.1 Adaptation Runtime

Table 8 separates GPU adaptation from CPU integration, excluding validation and evaluation. ALME uses $96.8\,\mathrm{s}$ of GPU time per domain, comprising $82.6\,\mathrm{s}$ for domain-specific LoRA adaptation and $14.2\,\mathrm{s}$ for the spectral probe. The closed-form solve takes $20.0\,\mathrm{s}$ on the CPU, bringing the sum of these components to $116.8\,\mathrm{s}$. Once the relation statistics have been extracted, integration requires no further GPU optimization.

Independent domain acquisition. Each domain is acquired from the fixed pretrained anchor W_0, so acquisition does not depend on the checkpoint produced by the preceding stage. This allows different domains to be acquired on separate workers without waiting for earlier model integration. Their relation statistics are subsequently accumulated using Eq. 5. Each update operates on the same set of matrices, and each integration stage produces a single deployed model.

B.2 Scaling with the Domain Stream

Let T denote the number of domains, M_t the number of feature tokens observed in domain $\mathcal{D}_t$, and $p_\ell = d_{\mathrm{in},\ell} + 1$ the augmented input dimension of editable layer ℓ. We consider $L = 34$ editable layers with output dimensions $d_{\mathrm{out},\ell}$.

Spectral rank allocation. We compute the leading singular values of each probe update over the admissible rank range. Within each group of same-shaped projections, the allocator distributes a fixed total rank budget according to the marginal spectral energy gains. The decomposition takes less than $0.8\,\mathrm{s}$ in our measurements.

Streaming statistics. ALME processes features in mini-batches and accumulates their relation statistics, avoiding storage of all M_t feature tokens. The running matrices require

$$\mathcal{O}\left(\sum_{\ell=1}^{L}\left[p_\ell^2 + p_\ell d_{\mathrm{out},\ell}\right]\right)$$

storage. Their size is determined by the editable layer dimensions and does not grow with the number of observed domains or tokens.

Table 9: Auxiliary state and deployed model structure for the evaluated configurations. Growth is measured with respect to the number of domains T, with fixed backbone dimensions, per-domain rank limits, and replay capacity. Auxiliary state excludes backbone checkpoints and transient optimization buffers.

Method	Auxiliary State	Growth	Deployed Structure
SeqLoRA / EWC	LoRA factors / regularization state	$\mathcal{O}(1)$	Merged base weights
ER / DER++	Adaptation state and a fixed replay buffer	$\mathcal{O}(1)$	Merged base weights
O-LoRA	Accumulated LoRA bases	$\mathcal{O}(T)$	Retained LoRA branches
SAMCL	Task selector and domain experts	$\mathcal{O}(T)$	Routing and expert selection
ALME (Ours)	FP64 relation statistics in CPU RAM	$\mathcal{O}(1)$	**Original SAM3 graph**

Per-stage integration. Integration solves one regularized linear system per layer. Cholesky factorization and the subsequent solves require

$$\mathcal{O}\left(\sum_{\ell=1}^{L}\left[p_\ell^3 + p_\ell^2 d_{\mathrm{out},\ell}\right]\right)$$

computation per stage. For a fixed backbone, the integration cost therefore remains independent of the number of previously observed domains T.

B.3 Persistent State and Model Deployment

CPU-resident accumulation. ALME maintains the running statistics $\overline{G}_{\ell,t}$ and $\overline{H}_{\ell,t}$ in CPU memory using FP64 to improve numerical stability. Each new domain updates these existing aggregates without adding domain-specific copies. Past images, feature tokens, and acquisition adapters need not be retained for subsequent integration. The persistent statistical state thus remains fixed throughout the domain stream.

Single-model deployment. ALME writes the integrated parameters directly into the original projection layers. The resulting model retains the standard SAM3 architecture, with no routing modules, additional adapter branches, or extra inference FLOPs. The pretrained anchor and accumulated statistics support future adaptation but are not required to serve the integrated model. Deployment therefore uses a single model whose parameter count and inference structure remain unchanged as new domains are incorporated.

Table 9 summarizes the state retained for adaptation and the deployed structure of the evaluated methods. ALME combines fixed-size statistical accumulation with an unchanged inference graph.

C Dataset Profiles and Experimental Protocol

C.1 Target Datasets

Table 10 summarizes the 13 specialized segmentation domains and the COCO general-domain reference. All images are resized to 1024×1024 before the SAM3 interactive point-prompt path. Adaptation samples a K-shot support set ($K \in \{1, 5, 10\}$; default $K = 5$) from the training pool of each dataset using three frozen splits; evaluation uses the full test split.

C.2 Few-Shot Split Construction & Prompt Sampling

For K-shot support sets ($K \in \{1, 5, 10\}$), each shot corresponds to an independent image–mask pair. We instantiate three evaluation splits using distinct seeds (42, 123, 2024). All baselines and

Table 10: Evaluation datasets spanning 13 specialized domains and the COCO general-domain reference. All inputs are resized to 1024×1024.

Domain Category	Dataset	Modality / Visual Target
Medical Imaging	ISIC (Codella et al., 2018)	Dermoscopic Lesions
	Kvasir (Jha et al., 2020)	Gastrointestinal Polyps
	JSRT (Shiraishi et al., 2000)	Chest Radiographs
Remote Sensing	HRSID (Wei et al., 2020)	SAR Ship Segmentation
Camouflaged & Salient Objects	CAMO (Le et al., 2019)	Camouflaged Animals/Objects
	COD (Fan et al., 2020)	Concealed Targets in Clutter
	DIS (Qin et al., 2022)	Highly Accurate Boundary Masks
	HRSOD (Zeng et al., 2019)	High-Resolution Salient Objects
	COIFT (Mansilla & Miranda, 2019)	Complex Topology Foreground
Shadow & Structures	ISTD (Wang et al., 2018)	Ground Surface Shadow Regions
	SBU (Vicente et al., 2016)	Scene Shadow Contours
	FLS (Singh & Valdenegro-Toro, 2021)	Forward-Looking Sonar Imagery
	ThinObject (Liew et al., 2021)	Extreme Aspect-Ratio Structures
General Reference	COCO (Lin et al., 2014)	In-the-Wild Natural Objects

ALME share identical splits. Point prompt sampling follows the protocol in Sec. 4.1: three positive points are uniformly sampled within the ground-truth mask interior for every object.

C.3 Formal Metric Definitions

Let $R_{t,j}$ denote the mIoU on task j after completing stage t. The metrics in Table 2 are computed as:

- **Average Incremental Accuracy (AIA):** $\mathrm{AIA} = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{t}\sum_{j=1}^{t} R_{t,j}\right)$.
- **Backward Transfer (BWT):** $\mathrm{BWT} = \frac{1}{T-1}\sum_{j=1}^{T-1}(R_{T,j} - R_{j,j})$.
- **Forward Transfer (FWT):** $\mathrm{FWT} = \frac{1}{T-1}\sum_{j=2}^{T}(R_{j-1,j} - R_{0,j}^{\mathrm{pristine}})$.

D Extended Empirical Analyses

D.1 Robustness to Statistic Rescaling

We test whether cross-domain integration remains stable when the scale of one domain's statistics changes. For each selected domain, we multiply both G and H by $\rho \in \{0.1, 10, 100\}$ before aggregation. Inverse-trace calibration normalizes each domain's contribution by its total input-feature energy. As shown in Table 11, ALME maintains 76.81–76.82% target mIoU across the tested rescalings, while uncalibrated aggregation drops as low as 41.78%. These results demonstrate the robustness provided by task-level statistic calibration.

D.2 Extension to Alternative Foundation Backbones

Table 12 extends ALME to SAM and SAM2 under matched 1-, 5-, and 10-shot settings. ALME outperforms sequential fine-tuning across architectures, demonstrating that decoupled model editing is applicable across the evaluated SAM backbones.

E Qualitative Visualizations

Figure 3 compares qualitative segmentation masks.

Table 11: Controlled scale stress under the 5-shot setting. Sufficient statistics (G, H) for a given domain are rescaled by ρ before aggregation.

Stressed Domain	ρ	**No Calibration** ($\beta_i = 1$)	**Inverse-Trace (ALME)**
ISIC (Medical)	0.1	72.47	76.81
	10	72.19	76.81
	100	50.98	76.82
HRSID (Remote Sensing)	0.1	72.13	76.81
	10	71.90	76.81
	100	41.78	76.82
CAMO (Camouflage)	0.1	72.15	76.81
	10	72.24	76.81
	100	71.77	76.82

Table 12: Performance comparison on continuous few-shot adaptation across **13** diverse target tasks based on SAM and SAM-2. The best results for each setting among continual methods are highlighted in **bold**. Superscripts denote method types: ^{a} Sequential FT, ^{b} Rehearsal-based, ^{c} Regularization, ^{d} Others.

Task Sequence: ISIC → ISTD → Kvasir → CAMO → COD → COIFT → DIS → FLS → HRSID → HRSOD → JSRT → SBU → ThinObject									
Method	**1-shot**			**5-shot**			**10-shot**		
	mIoU (%) ↑	F1 (%) ↑	MAE ↓	mIoU (%) ↑	F1 (%) ↑	MAE ↓	mIoU (%) ↑	F1 (%) ↑	MAE ↓
Base Model: SAM									
Pristine SAM (0-shot)	57.80	68.62	0.1258	57.80	68.62	0.1258	57.80	68.62	0.1258
Indep. FT	60.25	71.21	0.1083	71.86	81.24	0.0482	72.35	80.93	0.0502
SeqLoRA FT^a	53.83	64.30	0.1792	57.94	68.68	0.1169	58.63	68.33	0.1619
ER^b	60.17	71.77	0.0819	68.46	78.71	0.0611	68.99	78.79	0.0532
DER^b	61.08	72.71	0.0756	67.02	77.31	0.0681	70.25	79.73	**0.0514**
EWC^c	51.71	63.05	0.1767	58.38	69.39	0.1100	60.80	70.59	0.0820
O-LoRA^d	49.97	60.82	0.1983	55.12	66.20	0.1517	59.38	69.14	0.1231
SPPA^d	50.37	61.59	0.1796	52.63	63.54	0.1583	56.44	65.95	0.1354
LAG^d	53.11	64.18	0.1559	55.13	65.77	0.1480	58.53	68.30	0.1374
SAMCL^d	44.91	58.08	0.1246	62.90	74.14	0.0734	68.98	79.06	0.0552
ALME (Ours)	**64.25**	**74.96**	**0.0755**	**71.58**	**82.45**	**0.0587**	**73.26**	**82.24**	0.0521
Base Model: SAM-2									
Pristine SAM2 (0-shot)	61.10	71.26	0.1249	61.10	71.26	0.1249	61.10	71.26	0.1249
Indep. FT	68.23	77.85	0.0805	75.12	83.35	0.0544	77.21	85.18	0.0453
SeqLoRA FT^a	59.09	70.35	0.1136	60.54	70.53	0.1577	62.79	72.37	0.1296
ER^b	62.14	72.48	0.0983	67.11	76.72	0.0925	69.06	78.42	0.0768
DER^b	62.08	72.35	0.1036	65.21	74.96	0.1109	68.81	80.38	0.0994
EWC^c	58.07	68.34	0.1236	60.72	70.74	0.1551	62.65	72.25	0.1306
O-LoRA^d	57.08	67.34	0.1336	60.55	70.55	0.1570	62.80	72.38	0.1297
SPPA^d	62.07	72.34	0.1036	65.17	75.01	0.1101	67.67	77.30	0.0974
LAG^d	62.08	72.34	0.1036	65.21	74.97	0.1109	67.81	77.38	0.1294
SAMCL^d	61.00	71.47	0.1157	70.07	79.17	0.0791	71.94	80.61	0.0756
ALME (Ours)	**65.38**	**75.41**	**0.0924**	**72.73**	**81.52**	**0.0622**	**74.11**	**82.65**	**0.0541**

F Limitations and Broader Impacts

F.1 Limitations

While ALME effectively reduces forgetting without storing raw samples, it operates on the premise that domain adaptation can be captured through affine output projections in feed-forward layers. For extreme distribution shifts requiring major non-linear topology changes in the attention mechanism, purely affine least-squares editing may yield diminishing returns. Second, computing sufficient statistics requires extracting activation matrices via forward passes, which scales linearly with the number of tokens.

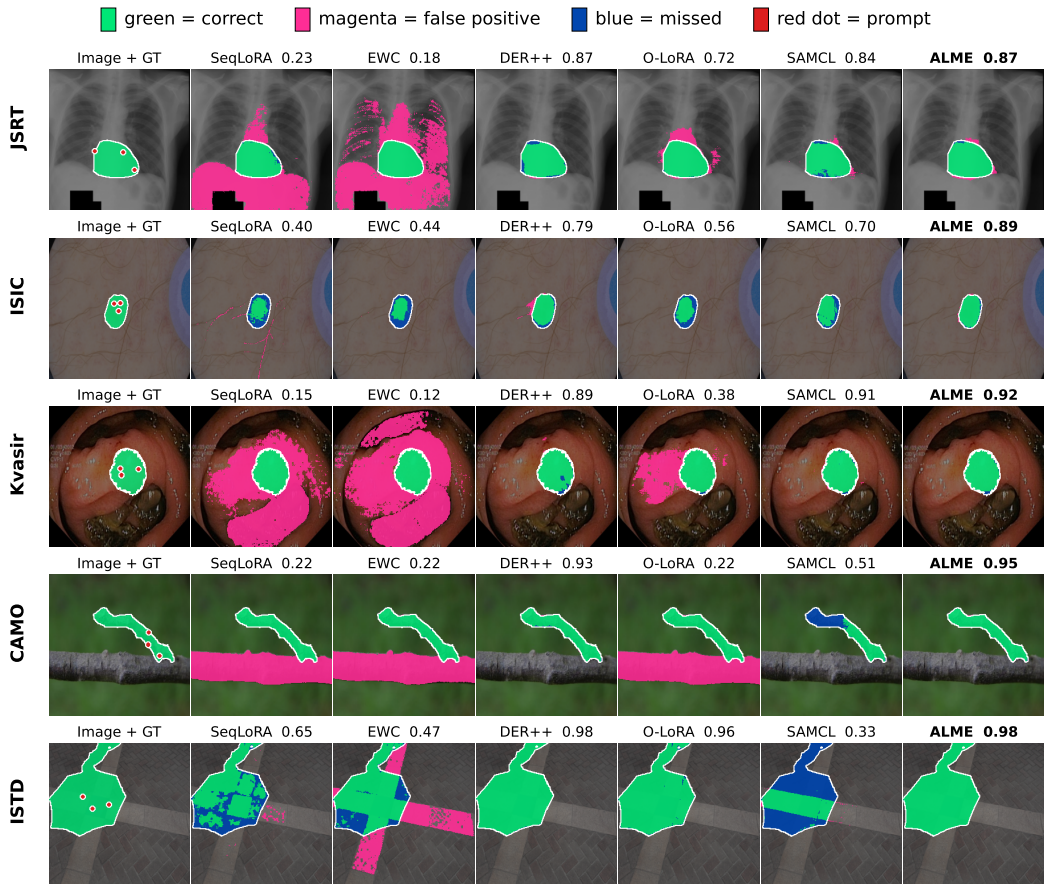

Figure 3: Visual comparisons across specialized domains. Positive point prompts are marked in red.

F.2 Broader Impacts

ALME facilitates the continuous deployment of large visual models in sensitive fields—such as healthcare (dermatology, endoscopy) and earth observation—where privacy regulations prohibit retaining raw client data. By eliminating the replay buffer, ALME ensures strict data governance while preventing model obsolescence.