SLARC: SHRINKAGE-CONTROLLED LOW-RANK ALIGNMENT AND REGULARIZED CLASSIFICATION FOR EXEMPLAR-FREE CLASS-INCREMENTAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continued adaptation of pre-trained representations can make stored class statistics stale. In exemplar-free class-incremental learning, old images are unavailable for recomputing these statistics. While existing methods compensate for representation drift and exploit class covariances for classification, two questions remain challenging without old images: how much of an estimated correction should be applied to old-class means, and how much class-specific covariance structure should the classifier retain? Shrinkage-Controlled Low-Rank Alignment and Regularized Classification (SLARC) addresses these questions through constrained mean alignment and a regularized Gaussian head. The alignment fits a low-rank affine residual and reduces its strength when disagreement between image-split fits is large relative to the proposed movement at old-class means. The head uses maintained means and combines retained class covariances with pooled and spherical estimates. Scores recorded at class introduction select the regularization parameters, allowing earlier classes to inform the choice without retaining their images or individual features. Across four ten-task benchmarks, SLARC improves E^2-LoRA's final accuracy by 0.70–4.08 percentage points.

1 INTRODUCTION

In exemplar-free class-incremental learning, a model learns new classes without retaining old training images and must recognize all classes seen so far (Zhu et al., 2022; Honda, 2026). Class means and covariances retain information for classification or for generating training features (Zhang et al., 2023; Goswami et al., 2023; Rypeść et al., 2024). A mean locates a class in feature space, and its covariance describes the variation around that location.

These summaries depend on the feature representation used to compute them. RanPAC and FeCAM keep the feature mapping fixed during later tasks (McDonnell et al., 2023; Goswami et al., 2023). We focus on continued adaptation of pre-trained vision transformers, primarily using LoRA (Li et al., 2026). Updating an adapter can change features even when the backbone weights remain fixed. Stored statistics may then describe an earlier representation, while test images are encoded by the updated model. Recomputing old-class statistics would require the discarded images.

Drift compensation addresses this mismatch by inferring changes in old-class statistics from current-task data. Existing methods estimate prototype shifts (Yu et al., 2020; Magistri et al., 2024; Goswami et al., 2024), learn feature-space maps (Gomez-Villa et al., 2024), or adapt both means and covariances (Rypeść et al., 2024; He et al., 2025). In pre-trained transformers adapted with LoRA, SLDC fits a ridge map and blends it with the identity (Rao et al., 2026). For classification, FeCAM and AdaGauss also show how class covariances can inform prediction (Goswami et al., 2023; Rypeść et al., 2024) (see Appendix §S1 for more details).

For learners with evolving features, our goal is to keep stored class summaries useful for prediction after each task. This requires choosing the directions and strength of corrections fitted on new classes, and how much historical class-specific covariance structure to use when classifying with those means.

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.

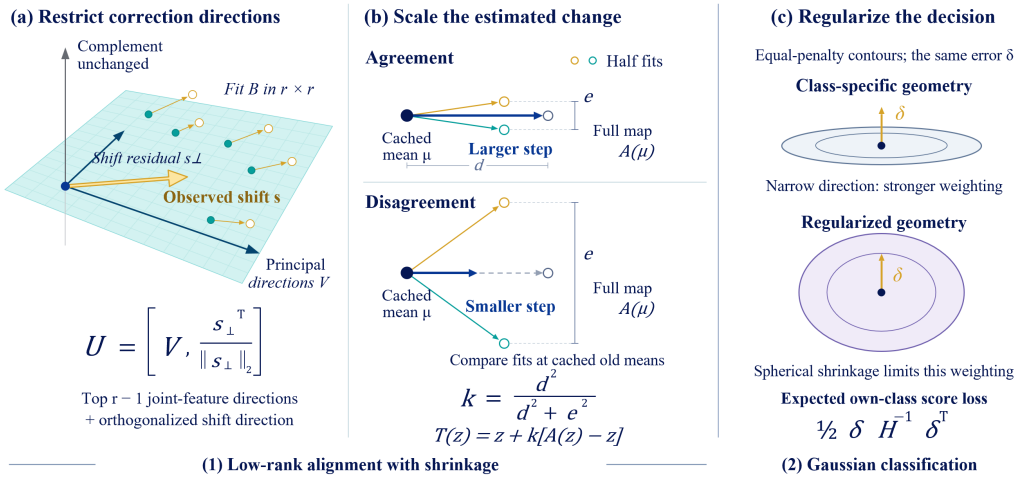

Figure 1: **SLARC couples constrained mean alignment (a,b) with regularized Gaussian classification (c). (a)** The correction subspace retains principal directions and observed mean displacement. **(b)** At cached old means, d is median full-map displacement and e median image-split disagreement; k scales the entire correction. **(c)** Covariance regularization controls mean-error weighting ($\boldsymbol{\delta}$: estimated minus true mean). The expected own-class score loss assumes an independent query at fixed trained state and scoring covariance H; it is not an accuracy guarantee.

We propose **SLARC**, *Shrinkage-Controlled Low-Rank Alignment and Regularized Classification*, with two components (Figure 1). *Constrained mean alignment* restricts the correction subspace and adapts the update strength to fit sensitivity. *Regularized Gaussian classification* combines the maintained means with historical covariances whose contribution is controlled by pooling and shrinkage.

The correction predicted at an old-class mean can vary with the current-task images used to fit the map. We fit an affine residual in a low-dimensional subspace that retains principal directions and the observed mean displacement (Figure 1a). This limits the fit's degrees of freedom while preserving the average shift observed on current images. We then compare fits from disjoint image halves at stored old-class means. When their disagreement is large relative to the movement predicted by the full-data fit, we reduce the entire correction (Figure 1b). This assesses fit sensitivity where the update is applied.

For classification, SLARC regularizes the use of historical class covariances. Each scoring covariance combines a class-specific term with the average across classes, with image counts controlling the class-specific contribution. Spherical shrinkage moderates directional weighting (Friedman, 1989). Together, these terms determine the distance to a maintained mean and the log-determinant contribution to its Gaussian score. This covariance also determines how residual mean error affects the score, linking classification to the preceding mean update (Figure 1c).

SLARC scores each candidate head when classes are introduced and selects regularization from a class-count-weighted aggregate of these scores. The archive lets earlier classes influence later choices after their images and individual features have been discarded.

Our contributions are as follows:

- **Mean alignment with split-fit shrinkage.** An identity-centered low-rank ridge fit retains the observed mean displacement, and an image-split rule scales the entire update. The analysis identifies conditions under which a shared correction reduces average old-class mean error.
- **Regularized classification without old images.** A Gaussian head combines retained class covariances with pooled and spherical estimates. Archived class-introduction scores select its regularization without retaining old images or individual features.
- **Plug-and-play integration.** SLARC integrates mean maintenance and Gaussian classification into learners with evolving features. Experiments with SLCA, TUNA (Wang et al., 2025), and E^2-LoRA (Li et al., 2026) show gains across different hosts. These gains persist in a minimal SeqLoRA control without E^2-LoRA-specific mechanisms.

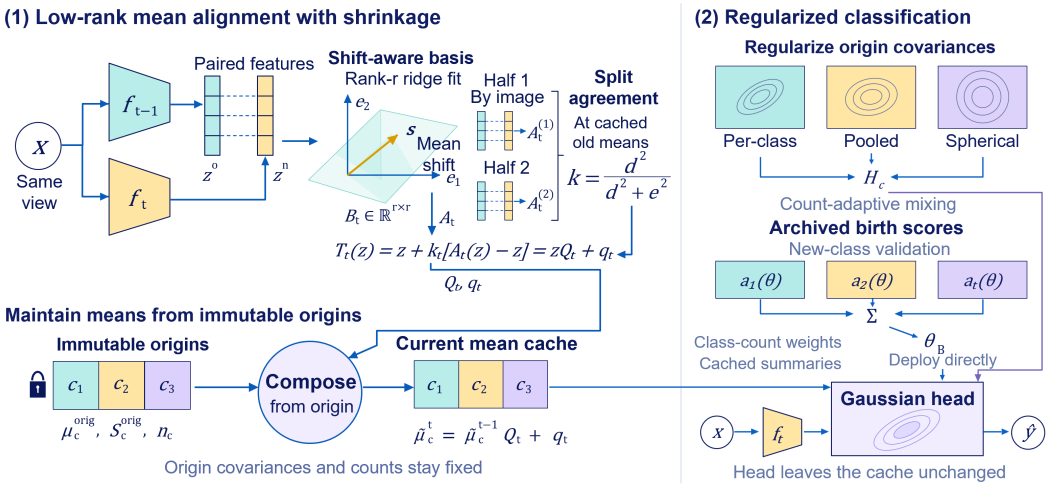

Figure 2: **SLARC at a task boundary.** (1) Paired current-task features fit an affine residual in a low-rank subspace containing the observed mean displacement. Disagreement between image-split fits controls the update strength; k, d, e denote k_t, d_t, e_t. Affine composition maintains means from immutable origins, while origin covariances and image counts remain fixed. (2) The Gaussian head combines class-specific, pooled, and spherical covariance estimates. Scores obtained at class introduction, using new-class validation folds and cached old-class competitors, are archived. Their class-count-weighted aggregate selects θ_B for direct deployment. Optional host replay, omitted here, uses maintained means and origin covariances rather than the head's regularized covariances.

2 Method

SLARC fits a correction from current images encoded before and after host training, then applies it to stored old-class means (§2.2). A Gaussian head uses these maintained means and regularized origin covariances for prediction (§2.3). Figure 2 summarizes the task-boundary procedure.

2.1 Problem formulation

We consider exemplar-free class-incremental learning over T tasks. Task t provides a labeled dataset $\mathcal{D}_t = \{(\boldsymbol{x}_{t,i}, y_{t,i})\}_{i=1}^{m_t}$, with labels in $\mathcal{N}_t$. Classes are disjoint across tasks: $\mathcal{N}_s \cap \mathcal{N}_t = \varnothing$ for $s \neq t$. The set of seen classes is $\mathcal{C}_t = \bigcup_{s=1}^{t} \mathcal{N}_s$. Task boundaries are known during training. At stage t, real training images are available only from $\mathcal{D}_t$; after training, the model must recognize all classes in $\mathcal{C}_t$ without the query's task identity.

Let $f_t : \mathcal{X} \to \mathbb{R}^{1\times D}$ be the feature mapping after task t, including the backbone and active adaptation modules. The mapping can change between tasks, but its output dimension D is fixed. Without old images, the current mean $\mathbb{E}[f_t(\boldsymbol{x}) \mid y = c]$ of an old class c cannot be directly re-estimated. Feature vectors are row vectors.

2.2 Low-rank mean alignment with split-fit shrinkage

Stored state and feature coordinates. When class c first appears at task s_c, we record its empirical mean $\boldsymbol{\mu}_c^{\text{orig}}$, unbiased sample covariance S_c^{orig}, and distinct-image count $n_c \geq 2$ under f_{s_c}. We keep these origin summaries fixed. A separate maintained mean, initialized as $\tilde{\boldsymbol{\mu}}_c^{s_c} = \boldsymbol{\mu}_c^{\text{orig}}$, is updated after each later task. Augmented copies and synthetic replay samples do not increase n_c. SLARC stores these summaries, affine-map state, and scalar validation scores alongside the host model, without referring to any old images or individual old features.

Current-task pairs. For each task $t > 1$, we collect N current-task feature pairs $(\boldsymbol{z}_i^o, \boldsymbol{z}_i^n)_{i=1}^N$, using the same image and augmented view under the before/after mappings. Let $\bar{\boldsymbol{z}}^o$ and $\bar{\boldsymbol{z}}^n$ be their means, and $\boldsymbol{s} = \bar{\boldsymbol{z}}^n - \bar{\boldsymbol{z}}^o$ the observed mean displacement. Following ridge-based drift compensation (Rao et al., 2026), we fit an affine map to these pairs. Because the map is applied at cached old-class means, we constrain its residual and assess fit sensitivity at those application points using image-split fits.

Displacement-aware subspace. We use the same subspace for the residual's input dependence and output, reducing its linear block from $D \times D$ to $r \times r$ coefficients. We choose principal directions

from both feature sets because the same basis must also represent the inputs to the fitted map. PCA of the feature differences would instead describe variation in the updates. To retain the observed mean displacement even when leading variance directions miss it, we reserve one direction for $\boldsymbol{s}$. For rank $1 \le r \le D$, let $V \in \mathbb{R}^{D\times(r-1)}$ contain the leading principal directions of the stacked before/after features, centered at their joint mean $(\bar{\boldsymbol{z}}^o + \bar{\boldsymbol{z}}^n)/2$. Define

$$\boldsymbol{s}_\perp = \boldsymbol{s}(I_D - VV^\top), \qquad U_t = \left[V, \frac{\boldsymbol{s}_\perp^\top}{\|\boldsymbol{s}_\perp\|_2}\right]. \tag{1}$$

When $\boldsymbol{s}_\perp = 0$, a deterministic orthogonal completion supplies the final column. Thus $U_t^\top U_t = I_r$ and $\boldsymbol{s}U_tU_t^\top = \boldsymbol{s}$.

Identity-centered ridge fit. We project the separately centered features as $\boldsymbol{x}_i = (\boldsymbol{z}_i^o - \bar{\boldsymbol{z}}^o)U_t$ and $\boldsymbol{y}_i = (\boldsymbol{z}_i^n - \bar{\boldsymbol{z}}^n)U_t$. We fit the $r \times r$ linear block by

$$B_t = \arg\min_B \left\{ \frac{1}{N}\sum_i \|\boldsymbol{x}_i B - \boldsymbol{y}_i\|_2^2 + \lambda\|B - I_r\|_F^2 \right\} = (K_{oo} + \lambda I_r)^{-1}(K_{on} + \lambda I_r), \tag{2}$$

where $K_{oo} = N^{-1}\sum_i \boldsymbol{x}_i^\top \boldsymbol{x}_i$, $K_{on} = N^{-1}\sum_i \boldsymbol{x}_i^\top \boldsymbol{y}_i$, and $\lambda > 0$. The penalty shrinks the linear block toward I_r; adding the mean displacement gives the full-space map

$$A_t(\boldsymbol{z}) = \boldsymbol{z} + \left[(\boldsymbol{z} - \bar{\boldsymbol{z}}^o)U_t(B_t - I_r) + \boldsymbol{s}U_t\right]U_t^\top. \tag{3}$$

Its residual lies in $\mathrm{span}(U_t)$, leaving the orthogonal complement unchanged, and $A_t(\bar{\boldsymbol{z}}^o) = \bar{\boldsymbol{z}}^n$. The low-rank constraint therefore applies to the residual, not to the full representation.

Split-fit shrinkage. The subspace limits correction directions, while ridge regularizes the projected linear block and leaves the translation $\boldsymbol{s}$ unpenalized. We assess the entire correction's sensitivity to image selection at the cached old means. We split current-task images into disjoint halves, keep all views of an image together, and refit each half's feature means, subspace, and ridge solution. At the maintained means $\{\tilde{\boldsymbol{\mu}}_c^{t-1} : c \in \mathcal{C}_{t-1}\}$, let e_t be the median disagreement between half fits and d_t the median displacement under the full-data map A_t. We use

$$k_t = \frac{d_t^2}{d_t^2 + e_t^2}, \qquad T_t(\boldsymbol{z}) = \boldsymbol{z} + k_t[A_t(\boldsymbol{z}) - \boldsymbol{z}] = \boldsymbol{z}Q_t + \boldsymbol{q}_t, \tag{4}$$

with $k_t = 0$ when both terms vanish. The coefficient scales the complete residual, including translation, and decreases as disagreement grows relative to the predicted displacement. Disagreement measures fit sensitivity, not correction error: both fits can share transfer bias. Thus k_t is a stability heuristic, not an optimal shrinkage coefficient or calibrated confidence estimate.

The task-level coefficient is shared across old classes. We compare its effect on final accuracy with fixed-strength and sample-count controls (§3.3.3). Figure 3 summarizes the saved coefficients and examines their local effect on reference mean error.

Maintaining class means. We apply the shrunk affine map only to old-class means

$$\tilde{\boldsymbol{\mu}}_c^t = \tilde{\boldsymbol{\mu}}_c^{t-1}Q_t + \boldsymbol{q}_t, \qquad c \in \mathcal{C}_{t-1}. \tag{5}$$

New classes start from their origin means, and there are no old means to align at the first task. Origin covariances S_c^{orig} and counts n_c remain fixed. Composing the fitted affine maps reconstructs the maintained means from their immutable origins, with the same result as sequential updates. Appendix §S2 gives the affine coefficients and composition rule. Appendix §S2.5 relates the deployed update to Proposition 1, separating shrinkage, map-estimation error, and the remaining old-class discrepancy.

Historical covariances and prediction. Transporting S as $Q_t^\top SQ_t$ applies the fitted map to within-class variation. Our split-fit check evaluates maps at old means and provides no corresponding check on that variation. We retain S_c^{orig} as a birth-time summary and construct a regularized scoring covariance H_c for prediction (§2.3). Its inverse and log determinant define the Gaussian decision geometry. Pooling and shrinkage control the use of historical statistics; they do not reconstruct current covariances. We compare covariance families at fixed maintained means to assess their predictive use.

Transfer to old classes. The following idealized result distinguishes identification from current-task pairs from improvement on old classes.

Proposition 1 (Shared affine mean correction). *Fix $t > 1$ and an orthonormal $U_t \in \mathbb{R}^{D\times r}$. Suppose current-task pairs have finite second moments and satisfy*

$$\boldsymbol{z}^n = \boldsymbol{z}^o + g_t(\boldsymbol{z}^o) + \boldsymbol{\xi}, \qquad \mathbb{E}[\boldsymbol{\xi} \mid \boldsymbol{z}^o] = 0, \tag{6}$$

where $g_t(\boldsymbol{z}) = (\boldsymbol{z}U_tL + \boldsymbol{b})U_t^\top$, $L \in \mathbb{R}^{r\times r}$, and $\boldsymbol{b} \in \mathbb{R}^{1\times r}$. If $\mathrm{Cov}(\boldsymbol{z}^oU_t) \succ 0$, unregularized population least squares over this affine residual family uniquely recovers g_t. For each old class, let $\boldsymbol{\mu}_c = \tilde{\boldsymbol{\mu}}_c^{t-1}$ and $\boldsymbol{\mu}_{c,t}^\star$ be its true current mean, and define

$$\boldsymbol{\mu}_{c,t}^\star = \boldsymbol{\mu}_c + g_t(\boldsymbol{\mu}_c) + \boldsymbol{r}_{c,t}. \tag{7}$$

Here $\boldsymbol{r}_{c,t}$ includes class-specific drift and inherited cache error. Let $\mathbb{E}_c$ average uniformly over $\mathcal{C}_{t-1}$. If $\mathbb{E}_c\langle g_t(\boldsymbol{\mu}_c), \boldsymbol{r}_{c,t}\rangle = 0$, then

$$\mathbb{E}_c\|\boldsymbol{\mu}_c - \boldsymbol{\mu}_{c,t}^\star\|_2^2 - \mathbb{E}_c\|\boldsymbol{\mu}_c + g_t(\boldsymbol{\mu}_c) - \boldsymbol{\mu}_{c,t}^\star\|_2^2 = \mathbb{E}_c\|g_t(\boldsymbol{\mu}_c)\|_2^2 \geq 0. \tag{8}$$

For the fitted map, define $\boldsymbol{\eta}_{c,t} = [A_t(\boldsymbol{\mu}_c) - \boldsymbol{\mu}_c] - g_t(\boldsymbol{\mu}_c)$. The deployed update obeys the exact identity

$$\tilde{\boldsymbol{\mu}}_c^{\,t} - \boldsymbol{\mu}_{c,t}^\star = -(1-k_t)g_t(\boldsymbol{\mu}_c) + k_t\boldsymbol{\eta}_{c,t} - \boldsymbol{r}_{c,t}. \tag{9}$$

Reducing k_t attenuates map-estimation error but leaves more shared displacement uncorrected; the old-class discrepancy remains. Appendix §S2.5 gives the proof and ridge bias. Orthogonality is an additional sufficient condition for the ideal gain, not a necessary condition for improvement. It does not follow from the pair model and is not established by the ablations. The fitted subspace need not contain actual drift, and $\boldsymbol{\eta}_{c,t}$ includes ridge bias, finite-sample error, and subspace error. Thus the proposition does not guarantee lower deployed mean error or improved classification accuracy.

2.3 Regularized Gaussian classification

We use the maintained means in a Gaussian classifier, following covariance-aware prediction in exemplar-free learning (Goswami et al., 2023; Rypeść et al., 2024). For a fixed positive-definite covariance H, the score is

$$g(\boldsymbol{z}; \boldsymbol{\mu}, H) = -\tfrac{1}{2}(\boldsymbol{z} - \boldsymbol{\mu})H^{-1}(\boldsymbol{z} - \boldsymbol{\mu})^\top - \tfrac{1}{2}\log\det H. \tag{10}$$

Suppressing class/task indices, let $\boldsymbol{\mu}^\star$ be the true current mean and $\boldsymbol{\delta} = \hat{\boldsymbol{\mu}} - \boldsymbol{\mu}^\star$ the maintained-mean error. Then

$$g(\boldsymbol{z}; \hat{\boldsymbol{\mu}}, H) - g(\boldsymbol{z}; \boldsymbol{\mu}^\star, H) = \boldsymbol{\delta}H^{-1}(\boldsymbol{z} - \boldsymbol{\mu}^\star)^\top - \tfrac{1}{2}\boldsymbol{\delta}H^{-1}\boldsymbol{\delta}^\top. \tag{11}$$

Small covariance eigenvalues can amplify the effect of mean error. Mean alignment targets this error; covariance regularization controls its weighting. Appendix §S3 gives a bound and expected-score identity. These concern score sensitivity at fixed H, not an accuracy guarantee from changing the covariance.

Count-dependent covariance regularization. A shared covariance pools evidence but removes class-specific geometry; separate covariances retain that geometry but can be poorly estimated. We interpolate between these choices and shrink toward a spherical model to control small eigenvalues. Distinct-image counts provide a sample-support proxy for the class-specific contribution. Writing $\boldsymbol{\mu}_c = \tilde{\boldsymbol{\mu}}_c^{\,t}$, we regularize the unchanged origin covariances with an RDA-type head (Friedman, 1989)

$$\begin{aligned} P &= \frac{1}{|\mathcal{C}_t|}\sum_{c\in\mathcal{C}_t} S_c^{\mathrm{orig}}, \qquad \beta_c = \beta_0\frac{n_c}{n_c+\tau}, \qquad M_c = (1-\beta_c)P + \beta_c S_c^{\mathrm{orig}}, \\ H_c &= (1-\alpha)M_c + \alpha\frac{\mathrm{tr}(M_c)}{D}I_D + \varepsilon I_D, \quad 0 \leq \alpha, \beta_0 \leq 1, \quad \tau \geq 0. \end{aligned} \tag{12}$$

The pool P estimates within-class variation without between-class scatter. We weight classes equally so that unequal training-set sizes do not determine the shared regularization target. Image counts control reliance on each class's own covariance through β_c: for $\tau > 0$, classes with fewer images rely more on the pool. Counts measure birth-time sample support, not covariance drift. The stored covariances may be stale, so their average need not match the current pooled covariance. Equal n_c gives a common mixing weight for all classes. Spherical shrinkage and loading $\varepsilon > 0$ stabilize inversion, with $\lambda_{\min}(H_c) \geq \alpha\,\mathrm{tr}(M_c)/D + \varepsilon$.

For a query feature $\boldsymbol{z} = f_t(\boldsymbol{x})$, prediction uses equal class priors

$$\hat{y}(\boldsymbol{z}) = \arg\max_{c\in\mathcal{C}_t} g(\boldsymbol{z}; \boldsymbol{\mu}_c, H_c). \tag{13}$$

At $\beta_0 = 0$, all classes share the same covariance, giving a linear discriminant.

Selection from archived birth scores. Selection from the latest task uses only new-class queries, while revalidating all seen classes would require old examples. We preserve validation evidence at each class introduction by evaluating 66 fixed candidates $\theta = (\alpha, \beta_0, \tau) \in \Theta$ and storing one score per candidate and task. Each candidate denotes the same regularization rule as the means, covariance pool, and competitors change.

At task s, fix f_s and use one deterministic evaluation view per image. Split new-class images into two halves, fit summaries on one half and score the other, then swap. This holdout applies only to those summaries. Old classes enter through maintained means and origin covariances: queries belong to $\mathcal{N}_s$, while prediction covers $\mathcal{C}_s$. The representation and mean maps were fitted using current-task training images before these folds, so birth scores are in-sample for those stages. They compare head candidates under the fixed trained state; they are not held-out test-accuracy estimates. No test images are used.

Average image accuracy within each class and nonempty validation half, then over these halves and new classes, to obtain $a_s(\theta) \in [0, 1]$. At task t, select

$$\theta_B = \arg\max_{\theta \in \Theta} \sum_{s \le t} |\mathcal{N}_s| \, a_s(\theta). \tag{14}$$

Ties select the first maximizer in the fixed candidate order. We deploy this setting directly on the current cache. Archived scores are not recomputed under later representations or against later classes. Their aggregate measures performance across past introduction events, not current all-class validation accuracy; historical rankings need not persist. Appendix S4 describes the candidate family and task-boundary procedure.

Host integration. When the host uses statistical replay for classifier alignment, it samples features from $\mathcal{N}(\tilde{\boldsymbol{\mu}}_c^t, S_c^{\text{orig}} + \varepsilon I_D)$. We retain the host classifier for later distillation and warm starts. We use the Gaussian head only for prediction: its regularized covariances neither overwrite the cache nor enter replay. Mean maintenance can therefore affect later training through host alignment, whereas head selection does not feed back through this interface.

3 Experiments

3.1 Experimental setup

Implementation details. We consider L2P (Wang et al., 2022b), DualPrompt (Wang et al., 2022a), RanPAC (McDonnell et al., 2023), SSIAT (Tan et al., 2024), EASE (Zhou et al., 2024), InfLoRA (Liang & Li, 2024), SD-LoRA (Wu et al., 2025), BiLoRA (Zhu et al., 2025), MOS (Sun et al., 2025b), GOD (Jia et al., 2026), SLCA (Zhang et al., 2023), TUNA (Wang et al., 2025), and E^2-LoRA (Li et al., 2026) as baselines. E^2-LoRA is the primary host, with SLCA and TUNA as additional hosts. All use ViT-B/16 pre-trained on ImageNet-21K (Dosovitskiy et al., 2021). E^2-LoRA is implemented in PILOT (Sun et al., 2025a). Within each host, native and SLARC runs share class orders, seeds, and representation-learning settings, following the host's training recipe. Mean alignment uses rank 64 and ridge penalty 10^{-2}; origin covariances remain fixed. Archived birth scores select the Gaussian-head parameters (§2.3) without old images or test data.

Evaluation. We evaluate on ten-task class-incremental splits of CIFAR-100 (Krizhevsky, 2009), ImageNet-R (Hendrycks et al., 2021a), ImageNet-A (Hendrycks et al., 2021b), and Cars-196 (Krause et al., 2013). After task t, A_t denotes test accuracy over all classes seen so far, without task identity. We report final accuracy $A_{\text{last}} = A_T$ and average incremental accuracy $A_{\text{inc}} = T^{-1} \sum_{t=1}^{T} A_t$. For our runs, both metrics are computed per seed and averaged over five seeds. Table 1 separates quoted references from our matched host comparisons; claims about adding SLARC use the latter. Appendix §S6.1 gives the partitions, protocols, and reference sources.

3.2 Main results

SLARC improves E^2-LoRA on all four benchmarks by 0.70–4.08 percentage points (pp) in final accuracy and 0.61–3.21 pp in average incremental accuracy (Table 1). Both metrics improve in eleven of twelve host–dataset settings. SLCA gains 0.55–9.62 pp in final accuracy; TUNA improves on three datasets, with ImageNet-R decreasing by 0.18 pp in final and 0.16 pp in average incremental accuracy. Cars-196 gives the largest final gain for each host: 4.08, 9.62, and 6.53 pp for E^2-LoRA, SLCA, and TUNA, respectively.

Table 1: **Final and average incremental accuracy (A_{last}, A_{inc}; %) on ten-task benchmarks.** The upper block contains quoted reference results; the lower block contains matched host comparisons. Shaded rows add SLARC, and **bold** marks the higher mean within each host pair.

	CIFAR-100		ImageNet-R		ImageNet-A		Cars-196	
Method	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}
L2P [CVPR22]	80.94	86.38	65.34	69.51	42.94	51.40	38.18	51.79
DualPrompt [ECCV22]	85.27	88.83	67.21	73.14	45.49	54.68	40.14	56.74
RanPAC [NeurIPS23]	91.09	94.03	75.28	80.66	54.76	64.22	61.43	72.36
SSIAT [CVPR24]	91.35	94.35	79.38	83.63	62.43	70.83	71.02	76.93
EASE [CVPR24]	88.57	91.94	75.51	78.34	59.25	68.92	35.46	48.96
InfLoRA [CVPR24]	82.43	86.64	72.19	73.84	49.20	60.92	57.66	70.03
SD-LoRA [ICLR25]	86.88	90.18	77.73	79.48	55.96	64.95	60.30	73.05
BiLoRA [CVPR25]	87.46	92.50	77.95	81.52	51.05	62.82	—	—
MOS [AAAI25]	90.53	94.09	78.10	83.61	54.60	63.57	67.76	74.38
GOD [CVPR26]	88.11	92.87	79.20	85.41	59.38	70.50	66.80	77.10
SLCA [ICCV23]	90.10	93.70	80.46	84.52	58.08	66.77	69.01	75.06
+ SLARC	**90.65**	**94.02**	**82.47**	**86.05**	**60.61**	**68.74**	**78.63**	**82.84**
TUNA [ICCV25]	91.61	94.51	**79.75**	**84.29**	61.42	68.59	69.64	75.64
+ SLARC	**91.77**	**94.84**	79.57	84.13	**62.60**	**69.11**	**76.17**	**80.87**
E^2-LoRA [ICML26]	92.13	94.61	82.53	86.28	62.54	71.03	75.48	80.29
+ SLARC	**92.87**	**95.46**	**83.23**	**86.89**	**63.40**	**71.76**	**79.56**	**83.50**

Scope of the comparison. Mean maintenance can affect subsequent training through E^2-LoRA's and SLCA's classifier alignment. TUNA also changes its inference path: SLARC replaces entropy routing and universal-adapter fusion with Gaussian classification on the current task-specific adapter (Appendix §S6.1). These comparisons evaluate the complete integration for each host. The E^2-LoRA ablations below separate mean maintenance from head replacement; SeqLoRA tests their combined prediction-time contribution without statistical feedback to training.

3.3 Ablation study

We first ask whether maintaining means improves native inference, and whether the Gaussian head adds to that gain. Component and mean-update ablations cover Cars-196 and CIFAR-100; the main tables show Cars-196 and the appendix gives both datasets (§S6.5). Update-strength and archive-selection controls cover all four benchmarks. These final accuracy comparisons are complemented by the single-seed local mean-error diagnostic in Figure 3(b). SeqLoRA tests whether the combined method remains useful without E^2-LoRA-specific mechanisms. Unless stated otherwise, ablations use ten-task E^2-LoRA and report five-seed mean final accuracy, with differences in pp.

3.3.1 Contribution of the two components

Table 2(a) separates the contributions of mean maintenance and classifier replacement. Because E^2-LoRA's classifier alignment uses cached statistics, maintenance can affect subsequent training. The first comparison therefore enables maintenance while keeping native inference; the second replaces that maintained run's classifier with the Gaussian head. These interventions add 2.47 and 1.61 pp, respectively, accounting for the 4.08 pp gain in this order. The Gaussian head does not affect training. The last two rows test whether maintained means still matter with a regularized Gaussian head. Fixing the final representation, origin covariances, counts, and head parameters, replacing maintained means with origin means lowers accuracy by 5.74 pp. Covariance regularization therefore does not remove sensitivity to stale means. This fixed-head contrast is not an additional contribution to the overall gain.

3.3.2 Mean-update design

Table 2(b) compares residual subspaces and update strengths. Each rule accumulates its own mean cache on fixed feature trajectories, without training feedback. Paired features, image partitions, origin summaries, ridge penalty, and Gaussian-head parameters are shared within each seed; the head is not reselected. On Cars-196, retaining displacement at rank 64 adds 2.16 pp over principal

(a) Mean maintenance and classifier replacement

Maintenance	Readout	Head means	A_{last}
Off	Native	—	75.48
On	Native	—	77.95
On	Gaussian	Origin	73.82
On	Gaussian	Maintained	**79.56**

(b) Mean-update design

Mean update	Residual rank	k_t	A_{last}
Origin means	—	—	73.82
Full-dimensional	D	1	77.36
Full-dimensional	D	split-fit	78.12
Principal directions	64	1	76.39
Displacement-aware	64	1	78.55
Displacement-aware	64	split-fit	**79.56**

Table 2: Ablations on Cars-196. **(a)** Mean maintenance and classifier replacement; Native denotes the host classifier. **(b)** Mean-update design. Rank refers to the alignment residual; $k_t = 1$ disables shrinkage, while split-fit uses Eq. (4).

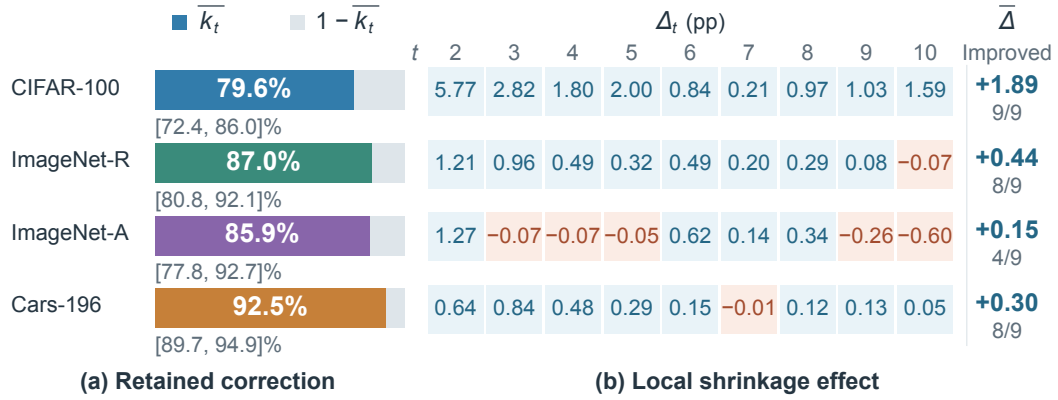

Figure 3: **Behavior of split-fit shrinkage.** (a) Mean and min–max range of k_t across five seeds and tasks 2–10. (b) $\Delta_t = 100(E_{H,t}^{\text{unshrunk}} - E_{H,t}^{\text{shrink}})/E_{H,t}^{\text{no}}$, where E_H denotes inverse-head-covariance-weighted mean error. $\bar{\Delta}$ averages the nine transitions; fractions count $\Delta_t > 0$. Positive values favor shrinkage; units are error-reduction pp (Appendix §S6.8).

directions alone. This displacement-aware fit exceeds the full-dimensional fit by 1.19 pp at $k_t = 1$ and 1.44 pp with split-fit shrinkage. The corresponding CIFAR-100 gains are 0.75 and 0.38 pp. Shrinkage improves both fits on both datasets (Appendix §S6.5).

3.3.3 Behavior of split-fit shrinkage

Table S5 tests update strength on fixed trajectories and heads, with five-seed paired differences. A constant set to the dataset median of 45 deployed coefficients is 0.36–1.27 pp lower than split-fit across all four datasets; the sample-count rule is lower by 0.41–2.46 pp. Table S6 adds SDC translation on fixed trajectories and an end-to-end E^2-LoRA control using our rank-64 map with SLDC's sample-count shrinkage. SLARC exceeds both on all four datasets, by 0.31–1.93 pp for the end-to-end control.

Figure 3 examines how much of the fitted correction is retained and whether shrinkage reduces reference mean error at each transition. Panel (a) shows that most of the correction is retained: mean k_t ranges from 79.6% on CIFAR-100 to 92.5% on Cars-196. Panel (b) compares shrunk and unshrunk updates from the same cache with the same Gaussian head. Reference means use old-class training images only for this offline diagnostic. Δ_t is positive at 29 of 36 transitions and on average for every dataset, although only four of nine ImageNet-A transitions benefit. These single-step comparisons differ from the independently accumulated caches in Table 2(b); squared-L_2 diagnostics appear in Appendix §S6.8.

3.3.4 Covariance regularization and archived selection

Table 3 separates the effects of covariance design and parameter-selection evidence. Both blocks fix the representation, maintained means, origin covariances, counts, and diagonal loading. The upper block progressively adds spherical shrinkage (α), class-specific mixing (β_0), and modulation by distinct-image counts (τ), using birth-archive selection, to test whether class-specific geometry improves on a shared covariance.

Spherical shrinkage alone changes accuracy little. Class mixing adds 1.41 pp on Cars-196 and 1.04 pp on CIFAR-100; count modulation adds 0.85 and 0.35 pp. These results favor regularized class-specific geometry, but do not isolate a benefit from class-varying weights: CIFAR-100 has equal class counts, so τ expands the available common mixing weights.

The lower block fixes the 66 candidates and varies the selection evidence. Full archive adds 2.75 pp on Cars-196 and 1.33 pp on CIFAR-100 over latest-task scores. Table S7 extends this comparison to all four datasets, with five-seed mean $\pm$ SD for first-only, latest-only, past-only, and full-archive policies, plus paired contrasts with a fixed default. Full archive has the highest mean among the four policies on every dataset and exceeds the default on three (Appendix §S6.7).

Table 3: Gaussian-head design (A_{last}). The upper block varies the covariance family; the lower block fixes the candidate set and varies the selection evidence.

Covariance / selection	Cars-196	CIFAR-100
Covariance family; archived selection		
Pooled ($\alpha = \beta_0 = 0$)	77.17	91.48
+ spherical (α)	77.30	91.48
+ class mixing (β_0; $\tau = 0$)	78.71	92.52
+ count weights (τ): full	**79.56**	**92.87**
Selection evidence; same 66 candidates		
Latest-task birth scores	76.81	91.54
All archived birth scores	**79.56**	**92.87**

3.3.5 A MINIMAL INCREMENTAL-LORA CONTROL

SeqLoRA tests whether SLARC improves prediction without statistical feedback to training. It retains task-wise LoRA accumulation: each task adds fresh fixed-rank adapters to a frozen backbone, while earlier adapters remain active but frozen. Other E^2-LoRA mechanisms are disabled, with data and optimization settings matched (Appendix §S6.2). Cached statistics affect neither training nor native inference, so both readouts use the same trained representation.

Table 4: SeqLoRA readout comparison (A_{last}). Both readouts use the same trained representation.

Readout	CIFAR 100	ImageNet R	ImageNet A	Cars 196
SeqLoRA	89.60	78.36	51.68	62.71
+ SLARC	**92.55**	**81.92**	**62.70**	**76.08**

Combined mean maintenance and Gaussian classification improve final accuracy by 2.95–13.37 pp across all four datasets (Table 4), showing that gains persist without training feedback or E^2-LoRA-specific mechanisms.

3.4 PERFORMANCE OVER LONGER TASK SEQUENCES

Longer sequences test whether SLARC's gains persist through more rounds of adaptation. SLARC improves E^2-LoRA under every evaluated partition (Figure 4). On CIFAR-100 and ImageNet-R, gains increase with task count while both methods' accuracies fall. Class partitions and training histories also change. Appendix §S7 gives complete accuracies and the fifty-task coverage following Li et al. (2026).

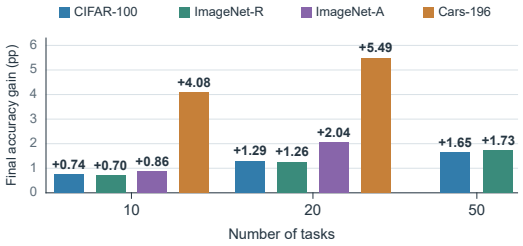

Figure 4: Gains in final accuracy over matched E^2-LoRA runs (pp). Fifty-task results are reported only for CIFAR-100 and ImageNet-R.

4 CONCLUSION

SLARC updates old-class means as the representation changes and uses them in a regularized Gaussian classifier. The mean update is a low-rank affine correction whose subspace contains the observed mean shift. Its strength depends on how well fits from disjoint image splits agree at old-class means. The classifier combines class-specific and shared covariances, with regularization chosen from scores recorded at class introduction. Neither step requires stored old images or individual features. SLARC improves E^2-LoRA and SLCA on the four main benchmarks; TUNA gains vary when its routed inference is replaced by a single-path Gaussian readout.

Limitations and future work. Our experiments cover image classification with pre-trained backbones and a feature mapping that changes but remains fixed in dimension. Mean maintenance requires paired current-task features, and we store a full covariance for each class at introduction. Structured covariance models could reduce this storage cost for larger class sets and higher feature dimensions. We also leave data-dependent selection of the alignment rank to future work.

AI USE STATEMENT

AI assistance was used to polish the manuscript text and assist with writing code for this work. All AI-assisted text and code were reviewed and verified by the authors.

ETHICS STATEMENT

This work studies exemplar-free class-incremental learning. It does not collect new human-subject data. The methods inherit the capabilities and limitations of the underlying models and datasets; the reported benchmark improvements concern image-classification accuracy and do not establish suitability for safety-critical decisions.

REPRODUCIBILITY STATEMENT

Our implementation uses PyTorch (Paszke et al., 2019) and NVIDIA RTX 6000 Ada 48 GB GPUs. The full implementation will be publicly released. §2 and Appendix §S4 provides additional details of our method.

REFERENCES

Yilun Chen, Ami Wiesel, Yonina C. Eldar, and Alfred O. Hero, III. Shrinkage algorithms for MMSE covariance estimation. *IEEE Transactions on Signal Processing*, 58(10):5016–5029, 2010. doi: 10.1109/TSP.2010.2053029.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Jerome H. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175, 1989. doi: 10.1080/01621459.1989.10478752.

Alex Gomez-Villa, Dipam Goswami, Kai Wang, Andrew D. Bagdanov, Bartłomiej Twardowski, and Joost van de Weijer. Exemplar-free continual representation learning via learnable drift compensation. In *ECCV*, 2024.

Dipam Goswami, Yuyang Liu, Bartłomiej Twardowski, and Joost van de Weijer. FeCAM: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. In *NeurIPS*, 2023.

Dipam Goswami, Albin Soutif-Cormerais, Yuyang Liu, Sandesh Kamath, Bartłomiej Twardowski, and Joost van de Weijer. Resurrecting old classes with new data for exemplar-free continual learning. In *CVPR*, 2024.

Run He, Di Fang, Yicheng Xu, Yawen Cui, Ming Li, Cen Chen, Ziqian Zeng, and Huiping Zhuang. Semantic shift estimation via dual-projection and classifier reconstruction for exemplar-free class-incremental learning. In *ICML*, 2025.

Run He, Di Fang, Yizhu Chen, Kai Tong, Cen Chen, Yi Wang, Lap-pui Chau, and Huiping Zhuang. REAL: Representation enhanced analytic learning for exemplar-free class-incremental learning. *Knowledge-Based Systems*, pp. 114901, 2026. doi: 10.1016/j.knosys.2025.114901.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, 2021a.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, 2021b.

Hiroto Honda. Adversarial pseudo-replay for exemplar-free class-incremental learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 7493–7502, March 2026.

Wangwang Jia, Zijian Gao, Tianjiao Wan, Yuan Cao, Yong Dou, and Kele Xu. Geometry-driven OOD detectors are class-incremental learners. In *CVPR*, 2026.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, pp. 554–561, 2013. doi: 10.1109/ICCVW.2013.77.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Longhua Li, Lei Qi, Qi Tian, and Xin Geng. Energy-structured low-rank adaptation for continual learning. In *ICML*, 2026.

Yan-Shuo Liang and Wu-Jun Li. InfLoRA: Interference-free low-rank adaptation for continual learning. In *CVPR*, 2024.

Simone Magistri, Tomaso Trinci, Albin Soutif-Cormerais, Joost van de Weijer, and Andrew D. Bagdanov. Elastic feature consolidation for cold start exemplar-free incremental learning. In *ICLR*, 2024. URL https://openreview.net/forum?id=7D9X2cFnt1.

Mark D. McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton van den Hengel. RanPAC: Random projections and pre-trained models for continual learning. In *NeurIPS*, 2023.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

Elias Raninen and Esa Ollila. Coupled regularized sample covariance matrix estimator for multiple classes. *IEEE Transactions on Signal Processing*, 69:5681–5692, 2021. doi: 10.1109/TSP.2021.3118546.

Xuan Rao, Simian Xu, Zheng Li, Bo Zhao, Derong Liu, Mingming Ha, and Cesare Alippi. Compensating distribution drifts in continual learning with pre-trained vision transformers. In *AAAI*, 2026.

Grzegorz Rypeść, Sebastian Cygert, Tomasz Trzciński, and Bartłomiej Twardowski. Task-recency bias strikes back: Adapting covariances in exemplar-free class incremental learning. In *NeurIPS*, 2024.

James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. CODA-Prompt: COntinual decomposed attention-based prompting for rehearsal-free continual learning. In *CVPR*, 2023.

Hai-Long Sun, Da-Wei Zhou, De-Chuan Zhan, and Han-Jia Ye. PILOT: A pre-trained model-based continual learning toolbox. *Science China Information Sciences*, 68(4):147101, 2025a. doi: 10.1007/s11432-024-4276-4. URL https://link.springer.com/article/10.1007/s11432-024-4276-4.

Hai-Long Sun, Da-Wei Zhou, Hanbin Zhao, Le Gan, De-Chuan Zhan, and Han-Jia Ye. MOS: Model surgery for pre-trained model-based class-incremental learning. In *AAAI*, 2025b.

Yuwen Tan, Qinhao Zhou, Xiang Xiang, Ke Wang, Yuchuan Wu, and Yongbin Li. Semantically-shifted incremental adapter-tuning is a continual ViTransformer. In *CVPR*, 2024.

Yan Wang, Da-Wei Zhou, and Han-Jia Ye. Integrating task-specific and universal adapters for pre-trained model-based class-incremental learning. In *ICCV*, 2025.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. DualPrompt: Complementary prompting for rehearsal-free continual learning. In *Computer Vision – ECCV 2022*, volume 13686 of *Lecture Notes in Computer Science*, pp. 631–648, 2022a. doi: 10.1007/978-3-031-19809-0_36. URL https://www.ecva.net/papers/eccv_2022/papers_ECCV/html/5428_ECCV_2022_paper.php.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *CVPR*, 2022b.

Yichen Wu, Hongming Piao, Long-Kai Huang, Renzhen Wang, Wanhua Li, Hanspeter Pfister, Deyu Meng, Kede Ma, and Ying Wei. SD-LoRA: Scalable decoupled low-rank adaptation for class incremental learning. In *ICLR*, 2025. URL https://openreview.net/forum?id=5U1rlpX68A.

Murat Onur Yildirim, Elif Ceren Gok Yildirim, and Joaquin Vanschoren. Unlocking [CLS] features for continual post-training. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=OWfWyj6krc.

Lu Yu, Bartłomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *CVPR*, 2020.

Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. SLCA: Slow learner with classifier alignment for continual learning on a pre-trained model. In *ICCV*, 2023.

Yaoyue Zheng, Yin Zhang, Joost van de Weijer, Gido M. van de Ven, Shaoyi Du, Xuetao Zhang, and Zhiqiang Tian. Revisiting weight regularization for low-rank continual learning. In *ICLR*, 2026. URL https://openreview.net/forum?id=pZj2DhfaVD.

Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *CVPR*, 2024.

Hao Zhu, Yifei Zhang, Junhao Dong, and Piotr Koniusz. BiLoRA: Almost-orthogonal parameter spaces for continual learning. In *CVPR*, 2025.

Kai Zhu, Wei Zhai, Yang Cao, Jiebo Luo, and Zheng-Jun Zha. Self-sustaining representation expansion for non-exemplar class-incremental learning. In *CVPR*, 2022.

Huiping Zhuang, Run He, Kai Tong, Ziqian Zeng, Cen Chen, and Zhiping Lin. DS-AL: A dual-stream analytic learning for exemplar-free class-incremental learning. In *AAAI*, 2024.

SUMMARY OF THE APPENDIX

This appendix provides extended related work, supporting derivations, implementation details, and additional experiments. Its contents are organized as follows:

- §S1. Extended related work.
- §S2. Mean alignment: fitting, shrinkage, and composition.
- §S2.5. Proof and scope of Proposition 1.
- §S3. Mean error and covariance regularization in Gaussian scores.
- §S4. Birth-score selection and algorithm pseudocode.
- §S5. Computational costs and numerical settings.
- §S6.1. Host settings, class partitions, and sources of reference results.
- §S6.2. SeqLoRA implementation.
- §S6.3. Run aggregation.
- §S6.4. Supplementary CUB-200 evaluation.
- §S6.5. Full component and mean-update ablations.
- §S6.6. Controls for shrinkage and mean-update rules.
- §S6.7. Head-selection controls.
- §S6.8. Single-step shrinkage diagnostics.
- §S6.9. Conditional diagnosis of mean maintenance.
- §S7. Long-sequence results.

S1 EXTENDED RELATED WORK

Representation updates in continual learning. Whether stored statistics become stale depends on the feature path used for prediction. RanPAC and FeCAM fix the effective encoder during later tasks, avoiding encoder-induced changes to those statistics (McDonnell et al., 2023; Goswami et al., 2023). Analytic learners also update classifiers on fixed features: DS-AL uses recursive least squares with a compensation stream, while REAL strengthens the initial representation and combines features from multiple layers (Zhuang et al., 2024; He et al., 2026). EASE preserves old adapter subspaces while adding new ones, so old-class information must be completed in the added coordinates (Zhou et al., 2024). Prompt and LoRA methods can change effective features even with frozen backbone weights (Wang et al., 2022b;a; Smith et al., 2023; Liang & Li, 2024; Li et al., 2026). E^2-LoRA organizes and recycles low-rank adaptation; TUNA combines task-specific and universal adapters (Li et al., 2026; Wang et al., 2025). GOD instead preserves task-specific experts and their feature mappings (Jia et al., 2026). SLARC addresses a fixed-dimensional feature path that evolves between tasks. It maintains means stored at class introduction for use with the current query path; this setting does not describe every exemplar-free learner.

Compensating for representation drift. Current-task data have long been used to estimate changes in old-class statistics. SDC estimates prototype shifts (Yu et al., 2020); later methods use perturbed samples (Goswami et al., 2024), adapter semantic-shift estimates (Tan et al., 2024), or shift estimation with classifier reconstruction (He et al., 2025). Learned feature-space maps provide another route (Gomez-Villa et al., 2024), and AdaGauss updates both means and covariances (Rypeść et al., 2024). SLDC already fits a ridge-based map, blends it with the identity, and transports both moments (Rao et al., 2026). SLARC builds on ridge transport and identity shrinkage. Its residual is restricted to a low-rank subspace containing the observed mean displacement, and its strength is set by comparing image-half fits at stored old-class means. It moves only the means and regularizes the unchanged origin covariances when constructing the classifier.

Classification from stored distributions. Stored distributions support both statistical replay and direct prediction. SLCA samples Gaussian features to align a linear classifier (Zhang et al., 2023). FeCAM uses class-specific Mahalanobis distances and compares them with sampled linear-head training (Goswami et al., 2023); AdaGauss predicts from retained Gaussian models while updating the representation (Rypeść et al., 2024). Regularized discriminant analysis mixes class-specific and pooled covariances and shrinks them toward spherical estimates (Friedman, 1989). Shrinkage and

coupled estimators also address limited-sample covariance estimation (Chen et al., 2010; Raninen & Ollila, 2021). FeCAM uses protocol-specific shrinkage settings, whereas AdaGauss couples distribution adaptation with anti-collapse representation learning (Goswami et al., 2023; Rypeść et al., 2024). SLARC uses a Gaussian score, including its log determinant, with a count-dependent covariance mixture. It selects the head from scores archived at class introduction, allowing earlier tasks to contribute after their images have been discarded. The head affects prediction; optional statistical replay uses the unchanged origin covariances.

S2 Mean alignment: definitions, composition, and scope

S2.1 Origin statistics and feature coordinates

The cache separates immutable birth statistics from the means updated by later tasks. For class c introduced at task s_c, let $\boldsymbol{z}_{ci} = f_{s_c}(\boldsymbol{x}_{ci}) \in \mathbb{R}^{1\times D}$ be the feature associated with original image i. The origin summaries are

$$\begin{aligned}\boldsymbol{\mu}_c^{\mathrm{orig}} &= \frac{1}{n_c}\sum_{i=1}^{n_c}\boldsymbol{z}_{ci},\\ S_c^{\mathrm{orig}} &= \frac{1}{n_c-1}\sum_{i=1}^{n_c}(\boldsymbol{z}_{ci}-\boldsymbol{\mu}_c^{\mathrm{orig}})^\top(\boldsymbol{z}_{ci}-\boldsymbol{\mu}_c^{\mathrm{orig}}).\end{aligned} \tag{15}$$

Both summaries remain fixed, and $n_c \geq 2$ counts distinct original images. Augmented views and synthetic samples do not add to this count. If a stored covariance uses denominator n_c, multiply it by $n_c/(n_c-1)$ to obtain the unbiased sample covariance above. The cache, pair fitting, prediction, and optional replay must use compatible feature coordinates and normalization conventions. In particular, the affine identities below assume no sample-wise normalization after applying the map.

S2.2 Displacement-aware fitting

The subspace constrains the fitted correction while retaining the observed mean displacement. Its rank satisfies $1 \leq r \leq D$; V has no columns when $r = 1$. If $\boldsymbol{s}_\perp = 0$ in Eq. (1), a deterministic orthogonal completion supplies the final direction. Rank-deficient pair data can similarly be completed with orthonormal null directions. Each run fixes the completion rule and numerical tolerance.

In exact arithmetic, both $\boldsymbol{s}VV^\top$ and $\boldsymbol{s}_\perp$ lie in $\mathrm{span}(U_t)$, so $\boldsymbol{s}U_tU_t^\top = \boldsymbol{s}$. Hence

$$\begin{aligned}A_t(\bar{\boldsymbol{z}}^o) &= \bar{\boldsymbol{z}}^o + \boldsymbol{s} = \bar{\boldsymbol{z}}^n,\\ [A_t(\boldsymbol{z}) - \boldsymbol{z}](I_D - U_tU_t^\top) &= \mathbf{0}.\end{aligned} \tag{16}$$

Thus the fitted map matches the current-task mean and leaves the orthogonal complement unchanged. Its residual linear block has rank at most r; the identity path retains all D coordinates. These properties impose no rank assumption on true drift. If the completion tolerance discards a small nonzero displacement component, mean matching is accurate only up to that component and roundoff.

Differentiating the objective in Eq. (2) gives

$$(K_{oo} + \lambda I_r)B_t = K_{on} + \lambda I_r. \tag{17}$$

The positive ridge penalty makes $K_{oo} + \lambda I_r$ positive definite, so the solution is unique even for rank-deficient projected features. For unchanged pairs, $K_{on} = K_{oo}$ and $\boldsymbol{s} = \mathbf{0}$; the solution then reduces to $B_t = I_r$ and $A_t(\boldsymbol{z}) = \boldsymbol{z}$.

S2.3 Split-fit shrinkage

We assess sensitivity to the fitting images at the old means where the map will be applied. Each image-half fit recomputes its feature means, subspace, and ridge solution, keeping all views of an image together. Before updating the cache, we compute the disagreement and full-map displacement used in Eq. (4):

$$\begin{aligned}e_t &= \mathrm{median}_{c\in\mathcal{C}_{t-1}}\,\|A_t^{(1)}(\tilde{\boldsymbol{\mu}}_c^{\,t-1}) - A_t^{(2)}(\tilde{\boldsymbol{\mu}}_c^{\,t-1})\|_2,\\ d_t &= \mathrm{median}_{c\in\mathcal{C}_{t-1}}\,\|A_t(\tilde{\boldsymbol{\mu}}_c^{\,t-1}) - \tilde{\boldsymbol{\mu}}_c^{\,t-1}\|_2.\end{aligned} \tag{18}$$

The rule sets $k_t = 0$ when $e_t = d_t = 0$. It performs no old-mean update at the first task and returns the identity when fewer than two distinct current-task images are available.

The ratio in k_t follows the form of an idealized signal–noise tradeoff. Suppose a displacement estimate is $\widehat{\boldsymbol{\Delta}} = \boldsymbol{\Delta} + \boldsymbol{\eta}$, with finite second moments, $\mathbb{E}\boldsymbol{\eta} = \mathbf{0}$, and $\mathbb{E}\langle \boldsymbol{\Delta}, \boldsymbol{\eta} \rangle = 0$. Writing $S = \mathbb{E}\|\boldsymbol{\Delta}\|_2^2$ and $V = \mathbb{E}\|\boldsymbol{\eta}\|_2^2$, we obtain

$$\mathbb{E}\|k\widehat{\boldsymbol{\Delta}} - \boldsymbol{\Delta}\|_2^2 = (1-k)^2 S + k^2 V, \qquad k^\star = \frac{S}{S+V} \tag{19}$$

when $S + V > 0$; every k has zero loss when $S = V = 0$. This calculation motivates the empirical rule without proving it optimal. In particular, d_t^2 and e_t^2 do not unbiasedly estimate S and V: the full-map displacement includes fitting error, and half-map disagreement does not measure full-map error. Both half fits can also share a bias that their disagreement cannot reveal.

S2.4 Affine coefficients and immutable origins

Affine composition lets us reconstruct each maintained mean from its immutable origin. Expanding T_t in Eq. (4) gives

$$\begin{aligned} Q_t &= I_D + k_t U_t (B_t - I_r) U_t^\top, \\ \boldsymbol{q}_t &= k_t \big[\boldsymbol{s} U_t - \bar{\boldsymbol{z}}^o U_t (B_t - I_r) \big] U_t^\top. \end{aligned} \tag{20}$$

For fixed $Q, \boldsymbol{q}$ and an integrable random row vector Z, linearity gives $\mathbb{E}[ZQ + \boldsymbol{q}] = \mathbb{E}[Z]Q + \boldsymbol{q}$. The corresponding empirical identity holds for any realized map, including one fitted to the same finite sample. A population identity for a learned random map requires conditioning on that map or using an independent reference distribution.

Initialize $G_{s\to s} = I_D$ and $\boldsymbol{a}_{s\to s} = \mathbf{0}$. For each introduction task s, compose subsequent updates as

$$\begin{aligned} G_{s\to t} &= G_{s\to t-1} Q_t, \\ \boldsymbol{a}_{s\to t} &= \boldsymbol{a}_{s\to t-1} Q_t + \boldsymbol{q}_t, \\ \tilde{\boldsymbol{\mu}}_c^{\,t} &= \boldsymbol{\mu}_c^{\mathrm{orig}} G_{s_c\to t} + \boldsymbol{a}_{s_c\to t}. \end{aligned} \tag{21}$$

The last identity holds at initialization, $t = s_c$; substituting it into Eq. (5) proves the induction step. Composition therefore reproduces sequential mean updates up to roundoff, with one shared composite map per introduction task. Although each $Q_t - I_D$ has rank at most r, the composed residual $G_{s\to t} - I_D$ need not. Covariances and image counts remain at their origins. Reconstructing the fitted updates preserves their accumulated estimation error; it does not recover the unknown current class means.

S2.5 Proof and scope of Proposition 1

Proposition 1 separates two questions: whether current-task pairs identify a shared affine correction, and when that correction reduces average old-class mean error. We fix U_t and both representation mappings throughout the population argument. Unsubscripted expectations are over current-task pairs; $\mathbb{E}_c$ averages uniformly over $\mathcal{C}_{t-1}$ at the fixed trained state.

Proof of Proposition 1. The assumed pair model is

$$\boldsymbol{z}^n - \boldsymbol{z}^o = g_t(\boldsymbol{z}^o) + \boldsymbol{\xi}, \qquad \mathbb{E}[\boldsymbol{\xi} \mid \boldsymbol{z}^o] = \mathbf{0}, \tag{22}$$

where $g_t(\boldsymbol{z}) = (\boldsymbol{z} U_t L + \boldsymbol{b}) U_t^\top$. For another member $h(\boldsymbol{z}) = (\boldsymbol{z} U_t \bar{L} + \bar{\boldsymbol{b}}) U_t^\top$ of the same affine residual family, define $\mathcal{L}(h) = \mathbb{E}\|\boldsymbol{z}^n - \boldsymbol{z}^o - h(\boldsymbol{z}^o)\|_2^2$. The conditional-mean assumption removes the cross term with $\boldsymbol{\xi}$, giving

$$\mathcal{L}(h) - \mathcal{L}(g_t) = \mathbb{E}\|h(\boldsymbol{z}^o) - g_t(\boldsymbol{z}^o)\|_2^2. \tag{23}$$

Let $\Delta L = \bar{L} - L$, $\Delta \boldsymbol{b} = \bar{\boldsymbol{b}} - \boldsymbol{b}$, $\boldsymbol{m}^o = \mathbb{E}\boldsymbol{z}^o$, and $K = \mathrm{Cov}(\boldsymbol{z}^o U_t)$. Since $U_t^\top U_t = I_r$, centering the projected source features yields

$$\begin{aligned} \mathbb{E}\|h(\boldsymbol{z}^o) - g_t(\boldsymbol{z}^o)\|_2^2 = \operatorname{tr}(\Delta L^\top K \Delta L) \\ + \|\boldsymbol{m}^o U_t \Delta L + \Delta \boldsymbol{b}\|_2^2. \end{aligned} \tag{24}$$

Both terms are nonnegative. Since $K \succ 0$, zero loss difference requires $\Delta L = 0$; the second term then requires $\Delta \boldsymbol{b} = \mathbf{0}$. Unregularized population least squares therefore uniquely identifies both coefficients and the affine function g_t.

For an old class, write $\boldsymbol{\mu}_c = \tilde{\boldsymbol{\mu}}_c^{t-1}$. Its current target mean satisfies the decomposition

$$\boldsymbol{\mu}_{c,t}^{\star} = \boldsymbol{\mu}_c + g_t(\boldsymbol{\mu}_c) + \boldsymbol{r}_{c,t}, \tag{25}$$

which defines the discrepancy $\boldsymbol{r}_{c,t}$.

Subtracting the target mean gives

$$\begin{aligned} \boldsymbol{\mu}_c + g_t(\boldsymbol{\mu}_c) - \boldsymbol{\mu}_{c,t}^{\star} &= -\boldsymbol{r}_{c,t}, \\ \boldsymbol{\mu}_c - \boldsymbol{\mu}_{c,t}^{\star} &= -g_t(\boldsymbol{\mu}_c) - \boldsymbol{r}_{c,t}. \end{aligned} \tag{26}$$

Their squared-norm difference is

$$\begin{aligned} &\|\boldsymbol{\mu}_c - \boldsymbol{\mu}_{c,t}^{\star}\|_2^2 - \|\boldsymbol{\mu}_c + g_t(\boldsymbol{\mu}_c) - \boldsymbol{\mu}_{c,t}^{\star}\|_2^2 \\ &\quad = \|g_t(\boldsymbol{\mu}_c)\|_2^2 + 2\langle g_t(\boldsymbol{\mu}_c), \boldsymbol{r}_{c,t}\rangle. \end{aligned} \tag{27}$$

Under the additional old-class condition

$$\mathbb{E}_c\langle g_t(\boldsymbol{\mu}_c), \boldsymbol{r}_{c,t}\rangle = 0, \tag{28}$$

averaging over $\mathcal{C}_{t-1}$ gives

$$\begin{aligned} &\mathbb{E}_c\|\boldsymbol{\mu}_c - \boldsymbol{\mu}_{c,t}^{\star}\|_2^2 - \mathbb{E}_c\|\boldsymbol{\mu}_c + g_t(\boldsymbol{\mu}_c) - \boldsymbol{\mu}_{c,t}^{\star}\|_2^2 \\ &\quad = \mathbb{E}_c\|g_t(\boldsymbol{\mu}_c)\|_2^2 \geq 0, \end{aligned} \tag{29}$$

which is Eq. (8). □

The old-class condition in Eq. (28) is additional to the current-task pair model and cannot be verified from current-task pairs. It is sufficient for the stated average gain, but is not necessary for improvement: Eq. (27) allows positive gain with a nonzero average cross term, and a sufficiently negative cross term reverses it. Even under orthogonality, the result concerns the old-class average, with no per-class guarantee.

Error inherited from the cache. To make the role of past estimation error explicit, define the true mean under f_{t-1} and its cache error by $\boldsymbol{\mu}_{c,t-1}^{\star} = \mathbb{E}[f_{t-1}(\boldsymbol{x}) \mid y = c]$ and $\boldsymbol{e}_c = \boldsymbol{\mu}_c - \boldsymbol{\mu}_{c,t-1}^{\star}$. If the transition between true means is written as

$$\boldsymbol{\mu}_{c,t}^{\star} = \boldsymbol{\mu}_{c,t-1}^{\star} + g_t(\boldsymbol{\mu}_{c,t-1}^{\star}) + \boldsymbol{\rho}_{c,t},$$

then the affine form of g_t gives

$$\boldsymbol{r}_{c,t} = \boldsymbol{\rho}_{c,t} - \boldsymbol{e}_c(I_D + U_t L U_t^{\top}). \tag{30}$$

Thus $\boldsymbol{r}_{c,t}$ combines the residual change in the true mean with propagated cache error. The proposition requires orthogonality with this full discrepancy, without assuming that the previous cache was exact.

Population bias of identity-centered ridge. The deployed ridge fit differs from the unregularized estimator in the proposition. To isolate this difference, let $\boldsymbol{m}^n = \mathbb{E}\boldsymbol{z}^n$ and define population-centered coordinates $\boldsymbol{x} = (\boldsymbol{z}^o - \boldsymbol{m}^o)U_t$ and $\boldsymbol{y} = (\boldsymbol{z}^n - \boldsymbol{m}^n)U_t$. Under the proposition's model, $\boldsymbol{y} = \boldsymbol{x}B^{\star} + \boldsymbol{\xi}U_t$, where $B^{\star} = I_r + L$. Thus $\mathbb{E}[\boldsymbol{x}^{\top}\boldsymbol{y}] = KB^{\star}$, and the population identity-centered ridge solution is

$$\begin{aligned} B_\lambda &= (K + \lambda I_r)^{-1}(KB^{\star} + \lambda I_r), \\ B_\lambda - B^{\star} &= -\lambda(K + \lambda I_r)^{-1}(B^{\star} - I_r). \end{aligned} \tag{31}$$

A fixed $\lambda > 0$ generally biases the linear block even when the population moments are known. Exact recovery holds at $\lambda = 0$ when $K \succ 0$. A pure translation has $B^{\star} = I_r$ and therefore no linear-block bias.

Let $g_{t,\lambda}^{\mathrm{pop}}$ be the residual in Eq. (3) formed with population means and B_λ. Since $\boldsymbol{m}^n - \boldsymbol{m}^o = g_t(\boldsymbol{m}^o)$, both the true and ridge residuals reproduce the source mean displacement. Away from that mean, their difference is

$$\begin{aligned} g_{t,\lambda}^{\mathrm{pop}}(\boldsymbol{z}) - g_t(\boldsymbol{z}) &= (\boldsymbol{z} - \boldsymbol{m}^o)U_t(B_\lambda - B^\star)U_t^\top \\ &= -\lambda(\boldsymbol{z} - \boldsymbol{m}^o)U_t(K + \lambda I_r)^{-1}(B^\star - I_r)U_t^\top. \end{aligned} \tag{32}$$

Matching the current-task mean therefore does not make the correction unbiased at old-class means. This calculation also holds U_t fixed; it gives no assurance that the subspace estimated from a finite sample contains the actual shared change.

Relation to the deployed update. To separate retained drift from estimation error, define the fitted residual $\hat{g}_t(\boldsymbol{z}) = A_t(\boldsymbol{z}) - \boldsymbol{z}$ and its error at an old mean as $\boldsymbol{\eta}_{c,t} = \hat{g}_t(\boldsymbol{\mu}_c) - g_t(\boldsymbol{\mu}_c)$. The deployed update is $\tilde{\boldsymbol{\mu}}_c^t = \boldsymbol{\mu}_c + k_t\hat{g}_t(\boldsymbol{\mu}_c)$. Substituting Eq. (25) gives the exact identity

$$\tilde{\boldsymbol{\mu}}_c^t - \boldsymbol{\mu}_{c,t}^\star = -(1 - k_t)g_t(\boldsymbol{\mu}_c) + k_t\boldsymbol{\eta}_{c,t} - \boldsymbol{r}_{c,t}. \tag{33}$$

For any fitted map and $k_t \in [0, 1]$, this identity separates the uncorrected shared displacement, the scaled estimation error, and the remaining old-class discrepancy. The error $\boldsymbol{\eta}_{c,t}$ includes regularization, finite-sample, and subspace errors relative to g_t. A smaller k_t reduces their contribution while retaining more of the shared displacement.

In the ideal case $\boldsymbol{\eta}_{c,t} = \mathbf{0}$ for all old classes, Eq. (28) yields

$$\begin{aligned} &\mathbb{E}_c\|\boldsymbol{\mu}_c - \boldsymbol{\mu}_{c,t}^\star\|_2^2 - \mathbb{E}_c\|\boldsymbol{\mu}_c + k_t g_t(\boldsymbol{\mu}_c) - \boldsymbol{\mu}_{c,t}^\star\|_2^2 \\ &\qquad = k_t(2 - k_t)\mathbb{E}_c\|g_t(\boldsymbol{\mu}_c)\|_2^2. \end{aligned} \tag{34}$$

Here k_t is shared across classes and fixed within the average. Estimated maps retain terms involving $\boldsymbol{\eta}_{c,t}$, and split-fit disagreement identifies neither these errors nor $\boldsymbol{r}_{c,t}$. The ideal gain therefore does not establish optimality of the empirical rule. The proposition addresses mean error only; it provides no guarantee for covariance recovery, head selection, or classification accuracy.

S3 How mean error enters the Gaussian decision score

This calculation explains why covariance regularization matters when a maintained mean is imperfect. Condition on the trained representation and retained summaries, and compare scores at the same fixed $H \succ 0$. Let $\hat{\boldsymbol{\mu}}$ be the estimated mean and $\boldsymbol{\mu}^\star$ the true current class mean. Write $\boldsymbol{\delta} = \hat{\boldsymbol{\mu}} - \boldsymbol{\mu}^\star$ and $\boldsymbol{r} = \boldsymbol{z} - \boldsymbol{\mu}^\star$. Expanding Eq. (10) yields Eq. (11); the common log determinant cancels. Cauchy–Schwarz then bounds the pointwise score change:

$$\begin{aligned} &|g(\boldsymbol{z}; \hat{\boldsymbol{\mu}}, H) - g(\boldsymbol{z}; \boldsymbol{\mu}^\star, H)| \\ &\quad \le \|\boldsymbol{\delta}H^{-1/2}\|_2\|\boldsymbol{r}H^{-1/2}\|_2 + \tfrac{1}{2}\|\boldsymbol{\delta}H^{-1/2}\|_2^2 \\ &\quad \le \frac{\|\boldsymbol{\delta}\|_2\|\boldsymbol{r}\|_2 + \frac{1}{2}\|\boldsymbol{\delta}\|_2^2}{\lambda_{\min}(H)}. \end{aligned} \tag{35}$$

For an independent query from class c under the fixed trained representation, assume $\mathbb{E}[\boldsymbol{z} \mid \text{trained state}, c] = \boldsymbol{\mu}^\star$ and $\mathbb{E}[\|\boldsymbol{z}\|_2 \mid \text{trained state}, c] < \infty$. Taking the expectation of the score difference gives

$$\mathbb{E}[g(\boldsymbol{z}; \boldsymbol{\mu}^\star, H) - g(\boldsymbol{z}; \hat{\boldsymbol{\mu}}, H) \mid \text{trained state}, c] = \tfrac{1}{2}\boldsymbol{\delta}H^{-1}\boldsymbol{\delta}^\top. \tag{36}$$

This is an expected own-class score loss at fixed H, with no Gaussian population assumption. A finite first moment suffices for the expected pointwise difference; separate expectations of the two quadratic scores would require finite second moments.

For the positive-semidefinite M_c in Eq. (12), $0 \le \alpha \le 1$ and $\varepsilon > 0$ give

$$\begin{aligned} \lambda_{\min}(H_c) &= (1 - \alpha)\lambda_{\min}(M_c) + \alpha\frac{\mathrm{tr}(M_c)}{D} + \varepsilon \\ &\ge \alpha\frac{\mathrm{tr}(M_c)}{D} + \varepsilon. \end{aligned} \tag{37}$$

The floor controls the sensitivity bound for a given mean error. It does not make larger α uniformly better: changing H_c changes both score geometry and the log determinant, while prediction depends on all competing classes. The calculation also leaves the origin covariances unchanged, so it establishes no correction of their possible drift.

S4 BIRTH-SCORE SELECTION AND THE TASK-BOUNDARY PROCEDURE

S4.1 CANDIDATE FAMILY AND ARCHIVE

The archive preserves evidence for head selection after earlier images are discarded. All hosts use the same birth-score procedure and the same 66-candidate deployed family, fixed before test evaluation:

$$\begin{aligned}\Theta_{\text{pool}} &= \{(\alpha, 0, 0) : \alpha \in \{0, 0.1, 0.3, 0.5, 0.7, 0.9\}\},\\ \Theta_{\text{class}} &= \{(\alpha, \beta_0, \tau) : \alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\},\\ &\qquad \beta_0 \in \{0.1, 0.25, 0.5\},\ \tau \in \{0, 512, 2048, 8192\}\},\\ \Theta &= \Theta_{\text{pool}} \cup \Theta_{\text{class}}, \qquad |\Theta| = 6 + 60 = 66.\end{aligned} \tag{38}$$

Enumeration starts with pooled members in increasing α. Class-specific members follow with increasing β_0 in the outer loop, α in the middle loop, and τ in the inner loop. Only pooled members allow $\alpha = 0$; all pooled members have $\beta_0 = \tau = 0$.

At task s, encode each birth-scoring image once with an unaugmented, deterministic evaluation view. Divide current new-class images into two halves using seed $20260819+$ the implementation's task index. Fit temporary new-class means and covariances on one half and score the other, then swap. Each head combines the fit-half summaries with the cached old-class summaries, computes its covariance pool from that combined set, and scores new-class queries against all of $\mathcal{C}_s$. There is no view averaging.

The holdout applies only to new-class summaries in the scored head. The representation and, after task one, the old-mean maps were already fitted using current-task training images. Birth scores therefore remain in-sample for those stages. They compare head candidates sharing a trained backbone, maps, and old-class competitors; they do not estimate held-out test accuracy. The procedure uses no test images or retained old images.

To weight new classes equally, first average image accuracy separately for each class and validation half. Skip a half with no validation images for that class. Let V_{scj} contain class c's validation images when half j is used for fitting, and let $h_{s,j,\theta}$ be the resulting classifier. The archived task score is

$$\begin{aligned}J_{sc} &= \{j \in \{1, 2\} : |V_{scj}| > 0\},\\ a_s(\theta) &= \frac{1}{|\mathcal{N}_s|} \sum_{c \in \mathcal{N}_s} \frac{1}{|J_{sc}|} \sum_{j \in J_{sc}} \frac{1}{|V_{scj}|} \sum_{\boldsymbol{x} \in V_{scj}} \mathbf{1}\{h_{s,j,\theta}(f_s(\boldsymbol{x})) = c\}.\end{aligned} \tag{39}$$

Since every new class has at least two full-data images, it contributes to at least one validation half. Classes receive equal weight within the task; their images are never pooled before averaging. Only one scalar per candidate and task is retained, with no individual validation features or predictions. At later tasks, aggregate these scores with weights $|\mathcal{N}_s|$ as in Eq. (14) and choose the first maximizer in the fixed enumeration. Deploy θ_B directly on the full class cache, with no augmented-view to evaluation-view scaling of α. Scores remain tied to their introduction events: they are not recomputed with later means, covariance pools, or competitors, and do not measure current all-class validation accuracy.

Small fit-half classes. A split can leave a class with too few fit images for an unbiased covariance. The following fallbacks define its temporary birth-scoring head without changing the full-data origins in Eq. (15). For class c and fit half j, let m_{cj} be the fit-image count and $\boldsymbol{z}_{ci}$ the features. Use

$$\begin{aligned}\boldsymbol{\mu}_{cj}^{\text{fit}} &= \begin{cases} m_{cj}^{-1} \sum_{i=1}^{m_{cj}} \boldsymbol{z}_{ci}, & m_{cj} \geq 1,\\ \mathbf{0}, & m_{cj} = 0,\end{cases}\\ S_{cj}^{\text{fit}} &= \begin{cases} \dfrac{\sum_{i=1}^{m_{cj}} (\boldsymbol{z}_{ci} - \boldsymbol{\mu}_{cj}^{\text{fit}})^\top (\boldsymbol{z}_{ci} - \boldsymbol{\mu}_{cj}^{\text{fit}})}{m_{cj} - 1}, & m_{cj} \geq 2,\\ 0_{D \times D}, & m_{cj} \leq 1.\end{cases}\end{aligned} \tag{40}$$

A singleton therefore uses its feature as the mean and a zero covariance; an empty fit half uses both a zero mean and covariance. Set the temporary head count to $\nu_{cj} = \max(m_{cj}, 1)$ so it remains positive, including for empty classes. Cached old classes keep their full-data counts n_c, and these fallbacks never change stored counts.

Zero covariances remain in the head's equal-class pool. Before spherical shrinkage and loading, $S_{cj}^{\text{fit}} = 0$ gives $M_{cj} = (1-\beta_{cj})P_j$, where β_{cj} is the temporary head's effective class-specific weight. A zero mean does not make its score candidate-independent: the quadratic term, log determinant, and competitor scores still depend on the head. The author-reported ImageNet-A implementation audit found 62 affected class–folds among 2,000 across five seeds (3.1%): 50 singletons and 12 empty fit halves, spread across five to nine tasks per seed. For seed 1993, all four empty class–folds had zero own-class accuracy under the four checked parameter settings, including the reported selected head; each had two to five validation images. That check covers neither all 66 candidates nor the effect of zero covariances on the pool, competing predictions, or selected parameters.

S4.2 Task-boundary pseudocode

Algorithm 1 specifies the retained state and task order; Algorithms 2 and 3 define mean alignment and Gaussian prediction. Features are D-dimensional row vectors. Birth scoring uses the two-half, single-view protocol above, Eq. (40) for temporary summaries, and the candidate order following Eq. (38). Numerical tolerances are run inputs. Temporary new-class counts are $\max(m_{cj}, 1)$; stored distinct-image counts never include augmented or synthetic samples.

The host classifier is retained for later distillation and warm starts. If the host performs statistical alignment, maintained means can affect later training through replay. The Gaussian head is used only for prediction: its H_c neither replaces the origin covariance nor enters replay. At task one, the procedure skips mean alignment but still computes origins and birth scores. Fold summaries remain temporary, and archived scores are never recomputed.

The interface requires fixed-dimensional features and access to current-task images before and after training. For TUNA, birth scoring uses the birth task's adapter f_s, while later queries use the current adapter f_t. Its Gaussian predictor omits native routing and universal adapter fusion. Appendix S6.1 explains this inference replacement and distinguishes it from transport with TUNA's native classifier.

The same head constructor serves validation and deployment. Deployment supplies $(\boldsymbol{m}_c, \Sigma_c, \nu_c) = (\tilde{\boldsymbol{\mu}}_c^t, S_c^{\text{orig}}, n_c)$; a validation fold substitutes only the temporary new-class summaries and rebuilds the pool from that combined set. The trained f_t and maintained old-class cache stay fixed across folds and candidates. Neither head construction nor archived selection changes the origin cache or replay covariance.

S5 Computation and numerical specification

The low-rank map reduces the alignment calculation, but does not make the entire method low-rank in memory or computation. Once U_t is available, projection costs $O(NDr)$, the two $r \times r$ products cost $O(Nr^2)$, and the ridge solve costs $O(r^3)$. Feature extraction and subspace estimation add to these costs. Both half fits use the same operations as the full fit. Applying a mean update through its low-rank factors costs $O(Dr + r^2)$ per class, with no covariance transport.

Algorithm 1 also composes one dense affine map per previous introduction task and reconstructs the old means from their origins. With generic dense multiplication, composition costs $O((t-1)D^3)$ per task; retaining the factors of $Q_t - I_D$ permits $O((t-1)D^2 r)$ computation instead. Direct dense reconstruction in Eq. (21) costs $O(|\mathcal{C}_{t-1}|D^2)$.

Dense origin covariances require $O(|\mathcal{C}_t|D^2)$ memory, beyond means, counts, and introduction indices. Dense composite maps add $O(tD^2)$, and the birth archive adds $t|\Theta|$ scalar scores; these counts exclude the host model. Sequential mean updates are possible under the affine identities, but storing origins and composite maps does not give subquadratic total memory.

Head selection repeats construction and scoring for each candidate and fold. Dense class-specific heads can require $O(|\mathcal{C}_t|D^3)$ for factorization and $O(m|\mathcal{C}_t|D^2)$ for m validation features per candidate and fold. Deployment requires $O(|\mathcal{C}_t|D^3)$ to build the factors and $O(|\mathcal{C}_t|D^2)$ per query. Shared-covariance members can reuse one factorization.

The supplied implementation specification uses

$$r = \min(64, D), \qquad \lambda = 10^{-2}, \qquad \varepsilon = 10^{-6}, \tag{41}$$

Algorithm 1 SLARC task boundaries and retained state

Require: Task stream $\{\mathcal{D}_t, \mathcal{N}_t\}_{t=1}^{T}$; host learner; rank r, ridge λ, loading ε; fixed candidate family Θ and run conventions stated above.
Ensure: Current means; immutable origin summaries and counts; composite maps; birth-score archive; selected Gaussian head.
Initialize the host from pre-trained f_0, with an empty class cache and birth-score archive.
for $t = 1, \ldots, T$ **do**
  Preserve f_{t-1} before current-task changes; train the host on $\mathcal{D}_t$ to obtain f_t.
  if $t > 1$ **then**
    Encode paired features $\mathcal{P}_t$ using the same images and augmented views under f_{t-1} and f_t.
    $(Q_t, \boldsymbol{q}_t, k_t) \leftarrow$ Algorithm 2$(\mathcal{P}_t, \{\tilde{\boldsymbol{\mu}}_c^{t-1}\}_{c \in \mathcal{C}_{t-1}}; r, \lambda)$.
    for each previous introduction task $s < t$ **do**
      $G_{s \to t} \leftarrow G_{s \to t-1} Q_t$; $\boldsymbol{a}_{s \to t} \leftarrow \boldsymbol{a}_{s \to t-1} Q_t + \boldsymbol{q}_t$.
    end for
    For every old class, set $\tilde{\boldsymbol{\mu}}_c^{\,t} \leftarrow \boldsymbol{\mu}_c^{\mathrm{orig}} G_{s_c \to t} + \boldsymbol{a}_{s_c \to t}$.
  end if
  Initialize $G_{t \to t} \leftarrow I_D$ and $\boldsymbol{a}_{t \to t} \leftarrow \mathbf{0}$.
  for each new class $c \in \mathcal{N}_t$ **do**
    Store $(\boldsymbol{\mu}_c^{\mathrm{orig}}, S_c^{\mathrm{orig}}, n_c, s_c = t)$ from f_t; set $\tilde{\boldsymbol{\mu}}_c^{\,t} \leftarrow \boldsymbol{\mu}_c^{\mathrm{orig}}$.
  end for
  Encode each birth-scoring image with one deterministic evaluation view; split into halves $\mathcal{D}_t^{(1)}$ and $\mathcal{D}_t^{(2)}$ using seed 20260819+ the implementation's task index.
  for $j \in \{1, 2\}$ **do**
    Form temporary new-class means and covariances from the fit half $\mathcal{D}_t^{(j)}$ using Eq. (40); set each new-class head count $\nu_{cj} \leftarrow \max(m_{cj}, 1)$.
    Combine these with the fixed old-class $(\tilde{\boldsymbol{\mu}}_c^{\,t}, S_c^{\mathrm{orig}}, n_c)$ to obtain temporary summaries $\mathcal{S}_j$ for all $\mathcal{C}_t$.
    for each $\theta \in \Theta$ **do**
      Build Algorithm 3$(\mathcal{S}_j, \theta, \varepsilon)$; score new-class features from $\mathcal{D}_t^{(3-j)}$ against all classes in $\mathcal{C}_t$.
    end for
  end for
  For each candidate, average image accuracy within each class and nonempty validation half, then over those halves and new classes; archive $a_t(\theta)$ as in Eq. (39).
  if the host uses statistical classifier alignment **then**
    Train only its classifier on samples from $\mathcal{N}(\tilde{\boldsymbol{\mu}}_c^{\,t}, S_c^{\mathrm{orig}} + \varepsilon I_D)$; keep f_t fixed.
  end if
  $\theta_B \leftarrow \arg\max_{\theta \in \Theta} \sum_{s \le t} |\mathcal{N}_s| a_s(\theta)$; select the first maximizer in the fixed candidate order.
  Deploy Algorithm 3 using θ_B and all $(\tilde{\boldsymbol{\mu}}_c^{\,t}, S_c^{\mathrm{orig}}, n_c)$.
  Discard current images and individual features from retained memory; keep the host state, summaries, maps, and scalar archive.
end for

with approximately 12,800 augmented current-task pair views as the target and a seeded source-image split. Augmentation leaves distinct-image counts unchanged. Symmetrize covariances for roundoff and evaluate the head with Cholesky solves and log determinants. Each run must record its basis-completion tolerance, feature conventions, folds, candidate list, and tie rule, together with the selected α, β_0, τ and effective class weights.

For equal class counts, (β_0, τ) determines one common effective class-specific weight, so distinct parameter pairs can yield the same head. Varying τ can still add effective weights to the finite candidate set; equal counts remove variation across classes, not variation across candidates. The fixed enumeration and tie rule choose a reproducible representative. Rank, ridge penalty, pair budget, loading, and the candidate domain remain prescribed settings, so archived selection is not parameter-free.

S6 Additional experimental details and results

S6.1 Host settings and class partitions

Our within-host comparisons test the effect of adding SLARC under matched experimental conditions. We implement E^2-LoRA, TUNA, SLCA, and the SeqLoRA control in PILOT. Native and

Algorithm 2 Displacement-aware mean alignment with split-fit shrinkage

Require: Same-view feature pairs $\mathcal{P}$ with source-image IDs; old means $\mathcal{M}$; $1 \le r \le D$ and $\lambda > 0$.
Ensure: $(Q, \boldsymbol{q}, k)$ with $T(\boldsymbol{z}) = \boldsymbol{z}Q + \boldsymbol{q}$.
if $\mathcal{M}$ is empty or $\mathcal{P}$ has fewer than two distinct images **then**
  return $(I_D, \mathbf{0}, 0)$.
end if
Split source images into two disjoint halves, retaining all views together; let $\mathcal{P}_0 = \mathcal{P}$ and $\mathcal{P}_1, \mathcal{P}_2$ be their pair sets.
for $j \in \{0, 1, 2\}$ **do**
  $N_j \leftarrow |\mathcal{P}_j|$; compute its own before/after means $\bar{\boldsymbol{z}}_j^o, \bar{\boldsymbol{z}}_j^n$; $\boldsymbol{s}_j \leftarrow \bar{\boldsymbol{z}}_j^n - \bar{\boldsymbol{z}}_j^o$.
  $V_j \leftarrow$ leading $r-1$ directions of the stacked feature sets, centered at $(\bar{\boldsymbol{z}}_j^o + \bar{\boldsymbol{z}}_j^n)/2$.
  $\boldsymbol{s}_{\perp,j} \leftarrow \boldsymbol{s}_j(I_D - V_j V_j^\top)$.
  if $\boldsymbol{s}_{\perp,j} = \mathbf{0}$ **then**
    Set $U_j = [V_j, \boldsymbol{u}_j]$ using deterministic orthogonal completion.
  else
    $U_j \leftarrow [V_j, \boldsymbol{s}_{\perp,j}^\top / \|\boldsymbol{s}_{\perp,j}\|_2]$.
  end if
  $\boldsymbol{x}_i \leftarrow (\boldsymbol{z}_i^o - \bar{\boldsymbol{z}}_j^o)U_j$; $\boldsymbol{y}_i \leftarrow (\boldsymbol{z}_i^n - \bar{\boldsymbol{z}}_j^n)U_j$ for pairs in $\mathcal{P}_j$.
  $K_{oo,j} \leftarrow N_j^{-1} \sum_i \boldsymbol{x}_i^\top \boldsymbol{x}_i$; $K_{on,j} \leftarrow N_j^{-1} \sum_i \boldsymbol{x}_i^\top \boldsymbol{y}_i$.
  Solve $(K_{oo,j} + \lambda I_r)B_j = K_{on,j} + \lambda I_r$.
  Define $A_j(\boldsymbol{z}) = \boldsymbol{z} + [(\boldsymbol{z} - \bar{\boldsymbol{z}}_j^o)U_j(B_j - I_r) + \boldsymbol{s}_j U_j]U_j^\top$.
end for
$d \leftarrow \mathrm{median}_{\boldsymbol{\mu} \in \mathcal{M}} \|A_0(\boldsymbol{\mu}) - \boldsymbol{\mu}\|_2$; $e \leftarrow \mathrm{median}_{\boldsymbol{\mu} \in \mathcal{M}} \|A_1(\boldsymbol{\mu}) - A_2(\boldsymbol{\mu})\|_2$.
Set $k \leftarrow d^2/(d^2 + e^2)$, or $k \leftarrow 0$ when $d = e = 0$.
$Q \leftarrow I_D + kU_0(B_0 - I_r)U_0^\top$.
$\boldsymbol{q} \leftarrow k[\boldsymbol{s}_0 U_0 - \bar{\boldsymbol{z}}_0^o U_0(B_0 - I_r)]U_0^\top$.
return $(Q, \boldsymbol{q}, k)$.

Algorithm 3 Gaussian head construction and prediction

Require: Supplied class summaries $\mathcal{S} = \{(\boldsymbol{m}_c, \Sigma_c, \nu_c) : c \in \mathcal{C}\}$ with $\nu_c \ge 1$; $\theta = (\alpha, \beta_0, \tau)$; loading $\varepsilon > 0$.
Ensure: Predictor h_θ with equal class priors.
$P \leftarrow |\mathcal{C}|^{-1} \sum_{c \in \mathcal{C}} \Sigma_c$.
for each $c \in \mathcal{C}$ **do**
  $\beta_c \leftarrow \beta_0 \nu_c/(\nu_c + \tau)$; $M_c \leftarrow (1 - \beta_c)P + \beta_c \Sigma_c$.
  $H_c \leftarrow (1 - \alpha)M_c + \alpha \,\mathrm{tr}(M_c) I_D/D + \varepsilon I_D$.
  Symmetrize for roundoff; factor $H_c = L_c L_c^\top$; $\ell_c \leftarrow 2 \sum_i \log(L_c)_{ii}$.
end for
For any query feature $\boldsymbol{z}$, define $\boldsymbol{v}_c(\boldsymbol{z})$ by $L_c \boldsymbol{v}_c(\boldsymbol{z}) = (\boldsymbol{z} - \boldsymbol{m}_c)^\top$ for each class.
return the predictor $h_\theta(\boldsymbol{z}) = \arg\max_{c \in \mathcal{C}} \{-\frac{1}{2}\|\boldsymbol{v}_c(\boldsymbol{z})\|_2^2 - \frac{1}{2}\ell_c\}$.

augmented variants share hardware, software, representation-learning settings, class order, and run seeds within each host. Each published host retains its own training recipe; results quoted from other sources are outside this matching.

For ten tasks, CIFAR-100 introduces 10 classes per task and the 200-class benchmarks introduce 20; Cars-196 uses 20-class increments and a smaller final task. Twenty-task settings halve these increments. Fifty-task settings introduce two classes per task on CIFAR-100 and four on ImageNet-R. Our ImageNet-R split samples 24,000 training and 6,000 test images in an 80/20 ratio. Its size matches the L2P, DualPrompt, and CODA-P split, but the files differ. Cars-196 contains 8,144 training and 8,041 test images; our ImageNet-A split has 2–86 training images per class.

E^2-LoRA uses batch size 64, classifier learning rate 0.01, adapter learning rate 5×10^{-4}, and three classifier-alignment epochs. Representation learning runs for 5 epochs per task on CIFAR-100 and ImageNet-R, 10 on ImageNet-A, and 50 on Cars-196. Both arms use the same age-dependent mean-scaling heuristic. SLCA fine-tunes the full backbone and is the only host that changes pretrained weights. The SeqLoRA implementation below specifies which E^2-LoRA mechanisms are removed.

TUNA feature paths and inference. Adding SLARC to TUNA changes both mean maintenance and inference. Stock TUNA and +Transport retain the native classifier, entropy-based selection of

a task-specific branch, and fusion with the universal adapter. +Ours uses Gaussian classification on task-specific adapter features: a class's birth score uses the adapter for its introduction task s, whereas every query after task t uses the current adapter f_t. Per-query entropy routing and universal-adapter fusion are removed. Only classes introduced at task t therefore have birth scores recorded under the current query adapter. Earlier birth scores provide historical selection evidence rather than evaluations under f_t.

The weaker +Ours results relative to +Transport on the four primary benchmarks may reflect the change from birth-time to current adapter features or the removal of fusion, but these effects have not been isolated. The comparison evaluates the combined inference replacement; it cannot identify a Gaussian-head benefit within TUNA's original routing-and-fusion system.

Sources of the reference rows. The reference rows of Table 1 place our matched comparisons in the context of published and supplied results. The supplied anonymous manuscript *FARA: Failure-Aware Residual Adaptation for Class-Incremental Learning*, Tables 1–2, supplies L2P, DualPrompt, EASE, InfLoRA, and SD-LoRA on CIFAR-100, ImageNet-R, and ImageNet-A. For Cars-196, L2P, DualPrompt, and EASE use Li et al. (2026, Table 1); InfLoRA uses their Table 4 variant without KD or CA; SD-LoRA uses Jia et al. (2026, Table 1). RanPAC, SSIAT, BiLoRA (Zhu et al., 2025), and MOS (Sun et al., 2025b) use Li et al. (2026, Table 1) on CIFAR-100 and ImageNet-R and, where reported, Cars-196. Their ImageNet-A sources are Yildirim et al. (2026, Table 1) for RanPAC and MOS, Tan et al. (2024, Table 1) for SSIAT, and Zheng et al. (2026, Table 2) for BiLoRA. GOD denotes the Refined variant with Top-$k = 3$ from Jia et al. (2026, Table 1) on all four datasets.

Each final/incremental pair comes from one source, and none of these reference rows is rerun here. The GOD source uses fixed seed 1993; the quoted rows have no established common run cohort. BiLoRA's ImageNet-A source defines stage accuracy by averaging task accuracies, whose equivalence to pooled all-seen accuracy is unverified for that implementation. Shared backbone and task-count labels alone do not ensure matched splits, class orders, implementations, or seeds. Claims about adding SLARC therefore use the lower block's within-host comparisons, subject to TUNA's inference change above.

S6.2 SeqLoRA: a minimal incremental-LoRA control

SeqLoRA tests whether SLARC's gains persist after jointly removing E^2-LoRA's specialized mechanisms. It retains task-wise accumulation of fresh adapters and removes the mechanisms in Table S1. The backbone, data pipeline, class orders, task splits, optimizer, and training schedule match our E^2-LoRA implementation. Section 3.3.5 reports the readout comparison; here we specify the implementation and its scope.

Architecture and capacity growth. SeqLoRA freezes the ImageNet-21K pre-trained ViT-B/16 backbone. Each task, including the first, adds freshly initialized LoRA adapters at the same projections as our E^2-LoRA implementation. Their rank is a fixed ratio of 0.1 of each adapted layer's output width. Earlier adapters remain active and are not recycled, so capacity grows with task count. The shared linear classifier adds one block of output units per task.

Training without E^2-LoRA-specific mechanisms. Training updates only the current task's adapters and classifier block; all earlier adapters and blocks are frozen and excluded from the optimizer. Cross-entropy uses current-task images and logits from the current block. Energy-based rank allocation, rank recycling, energy-ordered rotation, distillation, and classifier alignment are disabled together. SGD uses batch size 64, classifier learning rate 0.01, and adapter learning rate 5×10^{-4}. The matched schedule runs for 5 epochs per task on CIFAR-100 and ImageNet-R, 10 on ImageNet-A, and 50 on Cars-196.

Class-incremental inference. After task t, each adapted projection is $W_0 + \sum_{k \le t} A_k B_k$, so inference uses one forward pass with all accumulated adapters. Adding an adapter changes effective features for old and new classes even though the backbone and earlier adapters are frozen. The native classifier takes the argmax over all seen-class logits without task identity. Its task-local training loss therefore does not restrict the inference candidate classes.

Statistics and readout comparison. SeqLoRA records each class's mean, unbiased covariance, and distinct-image count at introduction under the current effective feature mapping. These summaries support maintenance and prediction only; they enter neither adapter nor native classifier training. Under matched randomness, enabling mean maintenance therefore leaves the training trajectory and native predictions unchanged, $N_T = N_0$. Table 4 holds this trained representation fixed and replaces native inference with SLARC's maintained means and regularized Gaussian head. The measured gain combines both prediction-time components; it does not isolate mean substitution within a fixed head.

Because the host mechanisms are removed jointly and adapter capacity grows with task count, this control does not assign an effect to any individual E^2-LoRA mechanism. It is neither capacity-matched to such an ablation nor a reimplementation of InfLoRA or SD-LoRA.

Table S1: SeqLoRA implementation relative to E^2-LoRA. The listed changes are applied jointly, with the backbone, data pipeline, and optimization settings matched.

Component	E^2-LoRA	SeqLoRA
Distillation from previous model	on	off
Classifier alignment	3 epochs per task	off
Adapter rank	energy-based allocation	fixed ratio, 0.1
First-task adapter	full rank	same fixed ratio
New-adapter initialization	recycled directions	fresh
Post-task rotation into energy basis	on	off
Training loss	task-local CE + distillation	task-local CE

S6.3 Run aggregation

Our aggregate results use five seeds, with native and augmented arms sharing the seed cohort within each host–dataset setting. Reported standard deviations use denominator $m - 1$ for $m = 5$ runs.

Table 1 reports ten-task final and incremental accuracies. Component and long-sequence contrasts are differences between displayed means. Paired confidence intervals cannot be recovered from these rounded means or marginal standard deviations. Figure 4 compares final-accuracy gains across task partitions; Appendix S7 gives the complete absolute scores and published reference results.

S6.4 Supplementary CUB-200 evaluation

Table S2 tests whether the combined SLARC method also helps on CUB-200, comparing each stock host with both SLARC components under the ten-task setting. Both metrics increase for all three hosts. For E^2-LoRA, TUNA, and SLCA, respectively, final accuracy rises by 0.52, 0.79, and 1.97 percentage points, and average incremental accuracy by 0.26, 0.47, and 1.16 points. These differences use the displayed means and measure mean alignment and Gaussian classification together. They do not isolate either component; TUNA also uses the inference replacement described above.

Table S2: CUB-200, ten tasks: accuracy (%, mean ± SD over five seeds). +Ours combines mean alignment with split-fit shrinkage and regularized Gaussian classification with archived birth-score selection.

Host	Metric	Stock	+ Ours
E^2-LoRA	A_{last}	89.00 ± 0.15	89.52 ± 0.18
	A_{inc}	92.54 ± 0.38	92.80 ± 0.41
TUNA	A_{last}	88.73 ± 0.29	89.52 ± 0.24
	A_{inc}	92.40 ± 0.43	92.87 ± 0.35
SLCA	A_{last}	85.59 ± 0.48	87.56 ± 0.19
	A_{inc}	90.58 ± 0.30	91.74 ± 0.32

S6.5 FULL COMPONENT AND MEAN-UPDATE ABLATIONS

The following comparisons separate the effect of enabling mean maintenance from replacing the classifier, then examine the mean-update design on fixed feature trajectories. They expand the Cars-196 results in Table 2(a) and (b) to CIFAR-100. Both tables report E^2-LoRA final accuracy averaged over five seeds; contrasts are differences between displayed means.

Component readouts. Table S3 asks how the two components contribute when introduced in order. N_0 is the original host's native readout. Enabling mean maintenance gives a training run with native readout N_T; cached means can affect this run through the host's classifier alignment. On that same run, H_O uses a Gaussian head with origin means and H_T uses the same head with deployed maintained means. The Gaussian readouts share the final encoder, origin covariances, counts, and selected parameters. Substituting means does not rescore the birth archive.

Table S3: Component readouts for E^2-LoRA and their contrasts. Readouts are five-seed mean final accuracies (%); contrasts are differences between displayed means (pp). N_T, H_O, and H_T share a maintained training run within each seed.

Quantity	Readout or comparison	Cars-196	CIFAR-100
$A(N_0)$	Native host	75.48	92.13
$A(N_T)$	Native after mean maintenance	77.95	92.43
$A(H_O)$	Gaussian head, origin means	73.82	91.52
$A(H_T)$	Gaussian head, maintained means	79.56	92.87
Δ_N	$A(N_T) - A(N_0)$	$+2.47$	$+0.30$
$\Delta_{H\|O}$	$A(H_O) - A(N_0)$	-1.66	-0.61
$\Delta_{H\|T}$	$A(H_T) - A(N_T)$	$+1.61$	$+0.44$
$\Delta_{M\|H}$	$A(H_T) - A(H_O)$	$+5.74$	$+1.35$
Total change	$A(H_T) - A(N_0)$	$+4.08$	$+0.74$

The ordered change from native to full SLARC is $\Delta_N + \Delta_{H|T}$: first enable maintenance while retaining native inference, then replace the maintained run's classifier. Both steps improve the displayed mean on each dataset. The separate contrast $\Delta_{M|H}$ fixes the Gaussian head and substitutes maintained for origin means. Its positive value shows that covariance regularization does not remove the cost of stale means, but it is not an additional share of the total gain. These interventions do not assign independent contributions to the components. In particular, $\Delta_{H|O}$ compares different training histories and cannot isolate classifier replacement.

Mean-update choices. Table S4 asks whether displacement preservation and split-fit shrinkage help beyond reducing the fit dimension. Within each seed, the variants share the feature trajectory, feature pairs, image partitions, origin covariances, counts, ridge penalty, and selected head. Each rule accumulates its own mean cache offline, with no feedback to training or head reselection. These reconstructed caches are distinct from the deployed-cache readouts in Table S3.

Table S4: Mean-update rules with separately accumulated caches on fixed E^2-LoRA trajectories. Accuracy is the five-seed mean (%); Δ is the difference from origin means (pp). Rank denotes the alignment residual. The Gaussian head is shared and is not reselected.

			Cars-196		CIFAR-100	
Mean update	Rank	k_t	Acc.	Δ	Acc.	Δ
Origin means	—	—	73.82	—	91.52	—
Full-dimensional	D	1	77.36	$+3.54$	92.01	$+0.49$
Full-dimensional	D	split-fit	78.12	$+4.30$	92.49	$+0.97$
Principal directions	64	1	76.39	$+2.57$	92.23	$+0.71$
Displacement-aware	64	1	78.55	$+4.73$	92.76	$+1.24$
Displacement-aware	64	split-fit	79.56	$+5.74$	92.87	$+1.35$

At $k_t = 1$, principal directions alone underperform the full-dimensional fit on Cars-196 but improve on it on CIFAR-100. Preserving displacement at rank 64 adds 2.16 and 0.53 pp, respectively, placing this fit above the full-dimensional alternative by 1.19 and 0.75 pp. Split-fit shrinkage adds 0.76 and 0.48 pp to the full-dimensional fit, and 1.01 and 0.11 pp to the displacement-aware fit. With both using split-fit, the latter remains higher by 1.44 and 0.38 pp. These comparisons support the displacement-aware low-rank design under both strength rules; reducing dimension alone does not improve both datasets. The full-rule reconstruction reaches 92.87%, matching the displayed deployed Gaussian readout in Table S3 and the full-SLARC result in Table 1. Agreement of these endpoints does not make the offline and deployed cache constructions the same experiment.

S6.6 Additional transport-rule controls

Table S5 isolates update strength on fixed ten-task E^2-LoRA trajectories. Its two controls share the representation, fitted affine map, selected Gaussian-head parameters, origin covariances, and image counts. Only the strength of each old-mean update changes, and the alternative caches never feed back into training. Results use seeds 1993, 0, 1, 2, and 3. We compute accuracy differences within each seed and report their mean and sample standard deviation.

Constant and sample-count shrinkage. The constant control tests whether a typical deployed strength can replace fit-dependent shrinkage. It reuses the median of a dataset's 45 deployed coefficients (five seeds, tasks 2–10) at every transition. This is a retrospective match, not an accuracy-tuned optimum or a prospectively fixed universal default. The sample-count control instead sets $k_t = 1 - \exp(-n_t/768)$ from the distinct current-task image count n_t, following SLDC (Rao et al., 2026). It tests an alternative that does not measure fit sensitivity. Both controls retain our fitted map, so the comparison isolates the shrinkage rule and does not reproduce the full SLDC method.

Table S5: Shrinkage-rule controls on fixed E^2-LoRA trajectories. ΔA_{last} is control minus the deployed split-fit rule (pp, paired mean ± SD over five seeds); positive favors the control.

	CIFAR-100	ImageNet-R	ImageNet-A	Cars-196
Dose-matched constant k	0.796	0.870	0.870	0.925
Paired ΔA_{last} (pp)				
Constant k	-1.27 ± 0.08	-0.36 ± 0.09	-0.43 ± 0.03	-0.50 ± 0.12
Sample-count rule	-0.63 ± 0.14	-0.73 ± 0.04	-0.41 ± 0.13	-2.46 ± 0.12

Split-fit exceeds the matched constant by 1.27, 0.36, 0.43, and 0.50 pp on CIFAR-100, ImageNet-R, ImageNet-A, and Cars-196, respectively. The sample-count rule is also lower on all four datasets, by 0.41–2.46 pp. These controls support choosing the update strength from the current fit rather than using a typical deployed strength or image count alone. They do not compare against an accuracy-tuned best constant or a prospectively selected constant.

Distance-weighted translation. To examine an alternative compensation map, we apply SDC's local translation rule (Yu et al., 2020) on the fixed trajectories. Each old mean receives a weighted average of current-image feature displacements, with weights proportional to $\exp(-\|u - \mu\|^2/(2\sigma^2))$ in pre-update feature coordinates. We use $\sigma = 0.2$ on CIFAR-100 and 0.3 elsewhere. Table S6 reports final accuracy for this rule in our host and readout, rather than reproducing the original SDC training system.

End-to-end sample-count shrinkage. The second row retrains E^2-LoRA end-to-end using our rank-64 map, with update strength replaced by Rao et al.'s sample-count rule $k_t = 1 - \exp(-n_t/768)$ (Rao et al., 2026). Unlike the fixed-trajectory comparison in Table S5, maintenance can affect subsequent training. This tests SLDC's shrinkage within our framework, not the complete SLDC pipeline. SLARC's mean final accuracy is higher by 0.48, 0.31, 0.35, and 1.93 pp on CIFAR-100, ImageNet-R, ImageNet-A, and Cars-196, respectively.

Table S6: Alternative maintenance controls with ten-task E^2-LoRA (final accuracy, %, mean ± SD over five seeds). SDC uses fixed trajectories; SLDC shrinkage uses end-to-end training.

Rule	CIFAR-100	ImageNet-R	ImageNet-A	Cars-196
SDC translation	92.16 ± 0.16	82.69 ± 0.76	62.87 ± 0.65	77.15 ± 0.47
SLDC shrinkage	92.39 ± 0.21	82.92 ± 0.47	63.05 ± 0.11	77.63 ± 0.24

S6.7 HEAD-SELECTION CONTROLS

The head-selection controls ask whether retaining earlier birth scores helps choose a Gaussian head. They fix E^2-LoRA's maintained statistics and the deployed grid of 66 candidates, changing only the selection evidence. This differs from the covariance-family ablation in the upper block of Table 3, which changes the available head family. All policies in Table S7 share the candidate set and five seeds (1993, 0–3). The family comparison also depends on class counts: CIFAR-100 has equal counts, so varying τ expands the available common mixing weights without testing class-varying weight allocation.

First-only selects once from task 1 and retains that choice. At final evaluation, latest-only uses task 10, past-only uses tasks 1–9, and full archive uses tasks 1–10, with the weighting in Section 2.3. These are different uses of historical birth scores; none re-evaluates old classes under the final representation.

Table S7: Selection evidence with a fixed 66-candidate Gaussian-head grid and maintained statistics. The four archive policies report final all-class accuracy (%, mean ± SD over five seeds). The fixed default reports paired mean ± SD differences from full archive (pp) over the same five seeds.

Selection	CIFAR-100	ImageNet-R	ImageNet-A	Cars-196
First task	92.43 ± 0.26	82.55 ± 0.27	62.87 ± 0.30	76.86 ± 2.17
Latest task	91.54 ± 0.20	81.87 ± 0.19	62.19 ± 0.29	76.81 ± 1.41
Past tasks	92.31 ± 0.09	83.05 ± 0.24	62.08 ± 0.15	78.98 ± 0.49
Full archive	92.87 ± 0.09	83.23 ± 0.21	63.40 ± 0.13	79.56 ± 0.39
Paired ΔA_{last}: fixed default minus full archive (pp)				
Fixed default	$+0.06 \pm 0.13$	-1.02 ± 0.67	-1.67 ± 0.14	-2.34 ± 0.21

The fixed-default control removes score-based selection and uses $(\alpha, \beta_0, \tau) = (0.5, 0.25, 0)$ throughout: the middle α and β_0 values of the class-specific grid, with no count modulation ($\tau = 0$). Full archive has the highest displayed mean among the four archive policies on every dataset. Relative to latest-task scores, it adds 1.33, 1.36, 1.21, and 2.75 pp on CIFAR-100, ImageNet-R, ImageNet-A, and Cars-196, respectively. It also exceeds the fixed default by 1.02–2.34 pp on ImageNet-R, ImageNet-A, and Cars-196; the default is 0.06 pp higher on CIFAR-100. The results support retaining historical selection evidence within this deployed grid. They do not establish superiority to all fixed settings, in every seed, or on other hosts.

Original-family offline reference. The original candidate-family evaluation also included a separate selector using all-class training features. Its five-seed means are 80.52% on Cars-196 and 92.59% on CIFAR-100. This reference uses old training-image folds and is in-sample for the trained backbone, so it is neither a deployable exemplar-free selector nor an accuracy upper bound. It belongs to the original family evaluation, not the 66-candidate comparison above; it does not isolate selection evidence within that fixed grid.

S6.8 SINGLE-STEP SHRINKAGE DIAGNOSTIC

The single-step diagnostic asks whether shrinkage reduces reference mean error relative to the unshrunk update at a given boundary. Figure 3(a) first summarizes the saved k_t values from five seeds (0, 1, 2, 3, and 1993) and tasks 2–10, weighted equally. Panel (b) examines one E^2-LoRA trajectory per dataset (seed 1993). At each boundary, the unshrunk and deployed shrunk maps start from the

same deployed pre-update cache. They share the representation, empirical reference means, and Gaussian head within that task; these quantities may change at later tasks. Reference means are computed from old-class training images under the current representation, using images unavailable to the deployed update.

We measure discrepancy using inverse-head-covariance weighting (H) and squared Euclidean distance (squared L_2). For metric m, denote the error relative to the reference means by $E^v_{m,t}$ and define

$$G^v_{m,t} = 1 - \frac{E^v_{m,t}}{E^{\mathrm{no}}_{m,t}}, \qquad \Delta_{m,t} = 100\big(G^{\mathrm{shrink}}_{m,t} - G^{\mathrm{unshrunk}}_{m,t}\big).$$

The no-update branch keeps the current cache, including earlier maintenance; it does not reset means to their birth-time values. With a common positive no-update denominator, $\Delta_{m,t} > 0$ therefore indicates that shrinkage lowers reference error relative to the unshrunk update.

Table S8 reports the mean difference and number of strictly positive transitions for each metric. Its H columns match Figure 3(b). Mean differences are positive on every dataset under both metrics, but shrinkage does not help at every boundary: H improves at 29 of 36 boundaries and only four of nine on ImageNet-A. Means and positive counts are computed before display rounding. Squared L_2 uses its own no-update denominator, so magnitudes are not directly comparable across metrics.

These results describe local error changes against empirical reference means. They neither test the population assumptions of Proposition 1 nor measure classification accuracy. Because both local branches start from the deployed cache, they also cannot explain or reproduce the final-accuracy differences between independently accumulated caches in the five-seed comparison of Table S4.

Table S8: Cross-metric single-step shrinkage diagnostics on E^2-LoRA (seed 1993, tasks 2–10). Means are in percentage points of normalized reference-error reduction, not accuracy; positive counts use $\Delta_{m,t} > 0$. The H columns reproduce the summaries in Figure 3(b).

	H		Squared L_2	
Dataset	Mean (pp)	Positive	Mean (pp)	Positive
CIFAR-100	$+1.89$	$9/9$	$+2.79$	$9/9$
ImageNet-R	$+0.44$	$8/9$	$+0.90$	$9/9$
ImageNet-A	$+0.15$	$4/9$	$+0.21$	$5/9$
Cars-196	$+0.30$	$8/9$	$+0.23$	$8/9$

S6.9 Conditional diagnosis of mean maintenance

Figures S1 and S2 ask how changing old-class means affects mean discrepancy and classification on one E^2-LoRA trajectory per dataset (seed 1993). At each task, three readouts share the representation, origin covariances, distinct-image counts, birth-archive-selected head, and new-class means. They differ only in their old-class means: unchanged birth-time means, the run's maintained cache, or reference means recomputed from old training images under the current representation. The reference requires unavailable old images and is neither deployable nor a guaranteed accuracy bound. Head parameters may change between tasks but remain fixed across the three readouts within a task.

Both trajectories use mean maintenance during training. The origin-mean branch therefore substitutes a readout on the maintained trajectory; it does not reproduce training without maintenance. This isolates the conditional effect of old-mean substitution within each task. It complements the five-seed component studies but does not estimate across-seed effects or isolate the ridge penalty or split-fit rule.

Panel (a) averages relative mean error equally over old classes. Each class's error is the squared distance to its offline reference mean divided by the squared norm of that reference mean, rather than by within-class variance. Old- and new-class accuracies likewise weight classes equally within each group. All-class accuracy instead weights the cumulative test images directly. Because these accuracy columns use different weighting conventions, the all-class result cannot be recovered by treating the old and new groups as equally weighted.

On Cars-196, maintenance substantially reduces mean discrepancy and closes much of the old-class accuracy gap to the offline reference. On CIFAR-100, reduced discrepancy accompanies a much smaller accuracy change. At the final task, using either maintained or reference old means lowers new-class accuracy relative to the origin readout, although new-class means are unchanged. Moving old means changes competition for every query, so better old-mean estimates need not improve each class group's accuracy. The trajectories support evaluating mean error and classification jointly; they do not identify drift magnitude alone as the cause of the difference between datasets.

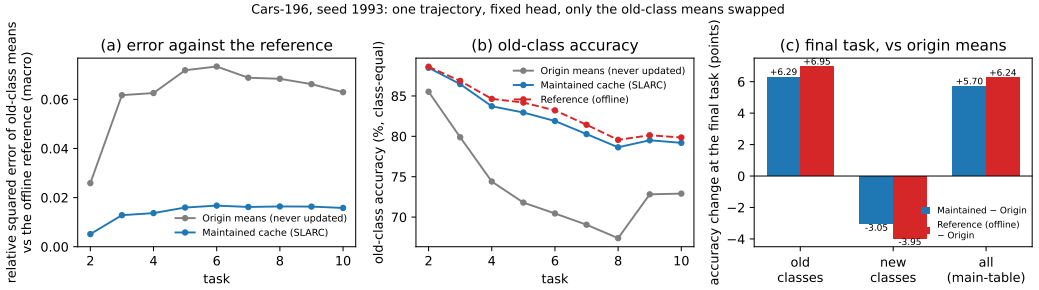

Figure S1: Old-mean substitution on a maintained Cars-196 trajectory (seed 1993). (a) Relative mean error against the offline reference. (b) Class-equal old-class accuracy. (c) Final-task accuracy changes from the origin-mean readout; old/new groups are class-equal, while all-class accuracy is image-weighted. The readouts share head settings within each task. No across-seed uncertainty is estimated.

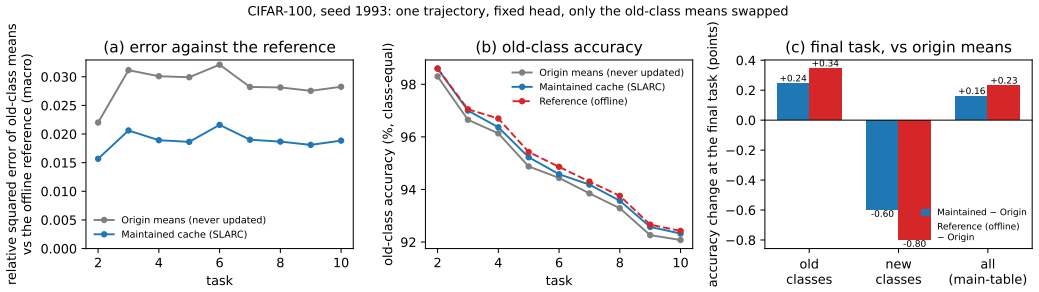

Figure S2: Old-mean substitution on a maintained CIFAR-100 trajectory (seed 1993), with the controls and weighting conventions of Figure S1. Reduced mean discrepancy accompanies small accuracy changes. Axis scales differ across datasets, so curve heights do not directly compare effect magnitudes.

S7 LONG-SEQUENCE COMPARISON

We test whether the gains from SLARC persist when learning is divided into more tasks. The matched comparisons below use native E^2-LoRA and the full SLARC method.

S7.1 COMPLETE MATCHED COMPARISONS

Table S9 reports final accuracy for each matched task partition. The ten-task means repeat Table 1. Within each dataset and partition, subtracting the native mean from the SLARC mean gives the gain discussed in Section 3.4. SLARC improves every reported comparison. Its gains increase with task count, although the class partitions and training histories also change; these comparisons do not isolate the effect of task count alone.

Table S9: Final accuracy (%) of native E^2-LoRA and full SLARC under matched task partitions. Dashes denote unreported settings.

Tasks	Readout	CIFAR 100	ImageNet R	ImageNet A	Cars 196
10	Native	92.13	82.53	62.54	75.48
	SLARC	**92.87**	**83.23**	**63.40**	**79.56**
20	Native	91.07	80.53	60.26	71.57
	SLARC	**92.36**	**81.79**	**62.30**	**77.06**
50	Native	89.88	77.81	—	—
	SLARC	**91.53**	**79.54**	—	—

Why fifty-task results cover two datasets. Following Li et al. (2026), we report fifty-task comparisons on CIFAR-100 and ImageNet-R. A fifty-task partition of our ImageNet-A split (approximately 6,000 training images across 200 classes; 2–86 per class) would average about 120 images per task; Cars-196 (8,144 training images, 196 classes) would average about 160. Smaller distinct-image pools may make both split-fit maps and birth-score evaluation more sensitive to the image partition. We leave these two low-data settings for dedicated evaluation; this sample-size argument is not a measured failure threshold.

S7.2 LONG SEQUENCES REFERENCE RESULTS

Table S10 reports CIFAR-100 and ImageNet-R results for equal-increment 20-/50-task splits with ViT-B/16 pre-trained on ImageNet-21K.

Table S10: Final accuracy (%) on longer task sequences, with SD where available.

	CIFAR-100		ImageNet-R	
Method	20 tasks	50 tasks	20 tasks	50 tasks
EASE	82.21 ± 0.72	82.10 ± 0.66	73.78 ± 0.44	68.54 ± 0.71
LoRA-DRS	88.69 ± 0.15	87.29 ± 0.31	74.80 ± 0.73	72.12 ± 0.87
MOS	88.85 ± 0.43	85.51 ± 0.04	75.58 ± 0.47	77.29 ± 0.77
TUNA	91.31 ± 0.12	89.41 ± 0.23	77.32 ± 0.31	75.35 ± 0.67
E^2-LoRA	91.07 ± 0.35	89.88 ± 0.35	80.53 ± 0.41	77.81 ± 0.16
+ SLARC	**92.36** ± 0.33	**91.53** ± 0.27	**81.79** ± 0.17	**79.54** ± 0.17