WHEN DOES ADAPTIVE CAPACITY HELP? A COMPONENT AUDIT OF CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual-learning systems combine trainable modules with allocation, routing, and classifier correction. We audit these components separately and identify two ways their interactions can obscure module utility. In a frozen-feature adapter system, AURA, bypassing the pool preserves aggregate final accuracy within one percentage point on four feature sets when prototype logit correction remains. Text factorials and inference interventions show that prototype scores largely compensate for pool-associated harm. Conversely, validation-selected DINOv2-B adapters with class-centroid routing gain 1.37 points post-encoder and 3.07 points in four internal blocks over no adapters. At the latter configuration's fixed learning rates, replacing class-centroid with task-mean routing lowers final accuracy by 14.66 points; separately tuned task-mean routing performs substantially better. This follow-up isolates routing at one operating point, not universal router superiority. Completed ViT-B/16 post-encoder controls do not improve accuracy. Native CODA-Prompt provides a positive allocation control, SEMA shows aggregation-sensitive conclusions, and ridge exceeds AURA on all four frozen feature sets. These results distinguish module utility from additional allocation benefit and support testing both under explicit classifier-correction, routing, and optimization controls.

1 INTRODUCTION

Pre-trained encoders make it attractive to continually learn by adding small modules and allocating, sharing, or routing them across tasks (Wang et al., 2022b;a; Smith et al., 2023; Wang et al., 2025). But a successful full system does not establish that its modules are useful, and an unsuccessful adapter configuration does not establish that adaptation is ineffective. Both conclusions depend on what the comparison holds fixed.

We distinguish *subsystem utility*, measured against a no-module predictor, from *allocation benefit*, measured against shared or reduced modules. An allocation-only ablation leaves the subsystem present in both arms. It therefore cannot reveal whether classifier correction compensates for module-associated harm. Conversely, testing modules with one router and optimization setting can conceal a useful configuration. We audit these explanations through component removal, factorial interventions, validation-selected configurations, and a fixed-optimization routing control.

Correction can conceal a weak subsystem. In the frozen-feature adapter system AURA, the pool is removable within one percentage point on four feature sets when prototype logit correction remains. On text, the fixed-setting factorial shows that correction largely offsets the accuracy loss associated with inserting the pool. Prototype-only prediction is close to the complete system. Four BGE task orders retain the small aggregate pool effect. This is an aggregate, tolerance-dependent result, not equivalence on every task. Ridge classification still exceeds AURA on all four feature sets, so subsystem removability does not imply a competitive remaining system.

Routing can conceal useful adaptation. Validation-selected DINOv2-B adapters gain 1.37 points post-encoder and 3.07 points in four internal blocks over no adapters with class-centroid routing. The selected internal task-mean configuration gains 0.93 points. A follow-up fixes both learning rates at the class-centroid operating point and changes only the router: class-centroid reaches 76.72%

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.

final accuracy versus 62.06% for task-mean, a 14.66-point paired difference. This establishes routing sensitivity at that operating point. It does not establish uniform superiority after separate tuning. The completed ViT-B/16 post-encoder controls remain negative.

What the audit contributes. Our contribution is an empirical diagnosis of how classifier correction and routing affect measured module utility, with seed-paired uncertainty and counterexamples to broad capacity redundancy. CODA's positive allocation contrast and SEMA's aggregation-sensitive results bound its scope. The study connects controlled comparisons in the evaluated systems. It introduces neither a new algorithm nor a universal allocation rule.

2 Related Work

Modules, sharing, and expansion. Bottleneck adapters (Houlsby et al., 2019) and AdapterFusion (Pfeiffer et al., 2021) separate task-specific transformations from their composition. L2P (Wang et al., 2022b), DualPrompt (Wang et al., 2022a), and CODA-Prompt (Smith et al., 2023) use different prompt-selection and sharing mechanisms. SEMA (Wang et al., 2025) adds adapters when representation descriptors detect a shift. EASE (Zhou et al., 2024b) ensembles subspaces. InfLoRA (Liang & Li, 2024) constrains updates to reduce interference. HiDe-Prompt (Wang et al., 2023a) separates task inference from within-task prediction. Our frozen-feature variants alter these original architectures and cannot establish their native performance.

The closest simplification studies. OVOR (Huang et al., 2024) shows that OnePrompt can replace a prompt pool, and its virtual-outlier classifier regularization transfers across prompt methods. Thus single-capacity simplification and transferable classifier correction both precede this work. Hash (Li et al., 2026) studies prompt sharing with component ablations and prompt-position experiments. Its evidence is not merely a comparison of complete methods. SEMA's ablation results examine disabling adapter expansion, changing adapter routing, and varying which transformer layers permit expansion on ImageNet-A and VTAB (Wang et al., 2025). We combine a pool-by-logit-correction factorial, reuse-policy controls, native SEMA/CODA interventions, and placement comparisons under matched module counts or expanded learning-rate selection. A fixed-optimization router intervention separates routing sensitivity from separately selected performance. Our contribution concerns the resulting conditional diagnoses, not the introduction of these intervention types or a general simplification result.

Classifiers and strong frozen-feature baselines. Nearest-class-mean methods (Janson et al., 2022; Zhou et al., 2024a) and RanPAC (McDonnell et al., 2023) show that strong representations can support continual classification with little or no gradient-based adaptation. SLCA (Zhang et al., 2023) emphasizes representation learning rates and classifier alignment. Our classifier study quantifies the conditional interaction between module utility and prototype logit correction. It does not introduce classifier correction or establish probability calibration. Implementation corrections are documented separately in Appendices A.10 and A.11.

Measurement scope. Benchmarking critiques in recommendation and metric learning (Ferrari Dacrema et al., 2019; Musgrave et al., 2020) motivate careful baseline and implementation checks. The linear-probe/fine-tuning distinction (Kumar et al., 2022) also motivates separating frozen weights from frozen representations. Original prompt and adapter methods may freeze encoder weights while changing intermediate activations. Our downstream variants freeze the representation itself. A nonlinear transformation of fixed features can still improve classification. The absence of new input information does not force a zero capacity effect.

3 Systems and Interventions

We study class-incremental prediction over all seen classes, with task boundaries available during training but no task identity at inference. We distinguish *subsystem utility*, measured against a no-module predictor, from *allocation benefit*, measured against reduced capacity (Table 1). AURA transforms fixed representations. Internal placement changes activations inside a frozen-weight encoder.

Table 1: Method-specific interventions. No-module versus full tests subsystem utility. Reduced versus full tests allocation benefit.

System	No module	Reduced	Full
AURA	Bypass outputs	One shared adapter	Triggered allocation
Placement	No adapters	One shared set	One set per task
SEMA	Adapters disabled	Twelve initial modules without expansion	Triggered expansion
CODA	No prompts	One component per task	Ten components per task

AURA as a component testbed. AURA (Adaptive Unsupervised Routing of Adapters) is our implementation for this audit, not a proposed algorithmic contribution. Despite the name, training uses class labels. Inference does not use task labels. It operates on a precomputed embedding $\mathbf{x} \in \mathbb{R}^{d_h}$. A residual bottleneck adapter has a learned input-channel gate, an output squeeze-and-excitation gate, and a learned residual scale whose initialization is specified per configuration. At task arrival, the maximum cosine similarity between new and stored class centroids determines allocation: below a threshold, create an adapter. Otherwise reuse the maximizing adapter. At inference, temperature-scaled centroid similarities weight a mixture of the available adapters without task identity. Shared pre-adapter normalization and routing temperature are learned despite the fixed input features (Appendix A.9).

The corrected policy trains the selected adapter on every task assigned to it, including reuse, while freezing old heads and inactive adapters. We separately evaluate the historical policy that freezes an adapter after its first task. Additional arms force one shared adapter, allocate one adapter per task, or bypass adapter outputs during both training and inference. Each arm is fitted under its stated protocol. Bypass retains an unused allocated module. It removes adapter contributions to prediction, not all stored adapter tensors.

Prototype logit correction. The mixture feeds per-task cosine heads whose logits are concatenated over all seen classes. AURA adds a prototype term to each class logit:

$$\hat{y}_c = y_c^{(raw)} + \alpha_{proto} \log(p_c^{(C)} + \epsilon), \qquad p_c^{(C)} = \frac{\exp(\cos(\mathbf{x}_{norm}, \mathbf{c}_c^{(C)})/\tau_{proto})}{\sum_j \exp(\cos(\mathbf{x}_{norm}, \mathbf{c}_j^{(C)})/\tau_{proto})}. \quad (1)$$

Here $\mathbf{x}_{norm}$ is the learned shared LayerNorm output and class prototypes $\mathbf{c}_c^{(C)}$ are updated by exponential moving averages on training batches. For $\alpha_{proto} > 0$, the bias is bounded below by $\alpha_{proto} \log \epsilon$. Both prototype correction and routing read the pre-adapter representation. Consequently, a weak pool effect may reflect their interaction with the classifier rather than a property of fixed features. We call this correction the *prototype gate* and test a 2×2 pool-by-gate factorial. If F_{pg} denotes final accuracy with pool indicator p and gate indicator g, its interaction is

$$I = (F_{11} - F_{10}) - (F_{01} - F_{00}). \quad (2)$$

The term changes classifier scores. We do not measure probability calibration or claim that it produces calibrated posteriors. This is a controlled interaction at fixed optimization settings. When learning rates are selected separately, the corresponding contrast also includes optimization changes.

Placement controls. A separate residual-adapter implementation places the same bottleneck family after the encoder or in its final blocks. Pool arms allocate one set per task. Shared arms retrain one set. Unlike AURA, this router uses fixed cached, unadapted final classification-token (CLS) features and a fixed temperature, with the same soft mixture during training and inference. Post-encoder stacks of one, four, or eight adapters match the parameter count of corresponding internal placements. They are actual transformations. The no-adapter condition contains none. These controls alter where trainable transformations act while keeping the routing reference fixed.

4 Experimental Setup

Data and encoders. Frozen-feature text experiments use 14 source-specific tasks and 70 globally distinct class IDs, with 27,931 training and 11,984 test rows (Wang et al., 2018; Socher et al., 2013;

Maas et al., 2011; Zhang et al., 2015; Williams et al., 2018; Bowman et al., 2015; Warstadt et al., 2019; Dolan & Brockett, 2005; Lehmann et al., 2015; Lang, 1995). Semantically similar labels from different datasets retain different IDs, so evaluation includes source discrimination. A stripped-text audit found no exact train/test overlap. Near-duplicates and pretraining overlap remain possible. The three encoders are T5-Large, BGE-Large, and Qwen3-1.7B (Raffel et al., 2020; Xiao et al., 2023; Yang et al., 2025).

Frozen-feature capacity and analytic-baseline experiments use DINOv2-L on ten-task, 200-class ImageNet-R: 24,002 training and 5,998 test images (Oquab et al., 2023; Hendrycks et al., 2021). Native CODA and placement instead use a capped export of 24,320 images with a 19,455/4,865 train/test split. CODA uses the authors' seeded class ordering, but these image lists differ from the published lists. Placement uses DINOv2-B and supervised ImageNet-21k ViT-B/16 (Dosovitskiy et al., 2021). Native SEMA uses the authors' original five-task VTAB archive and class order: 50 classes, 1,796 training and 8,619 test images (Wang et al., 2025). Task sample counts are unequal. Appendix A.7 records data scope. Encoders and datasets that differ across experiments are not interchangeable controls.

Metrics. Let $a_{i,j}$ be task j accuracy after task i, predicting over all classes seen by i. We report task-macro learning accuracy $\mathrm{AP} = T^{-1}\sum_i a_{i,i}$, final accuracy $\mathrm{FP} = T^{-1}\sum_j a_{T,j}$, and $\mathrm{BWT} = (T-1)^{-1}\sum_{j<T}(a_{T,j} - a_{j,j})$. They satisfy

$$\mathrm{FP} = \mathrm{AP} + \frac{T-1}{T}\mathrm{BWT}. \tag{3}$$

Average incremental accuracy is $\bar{\mathcal{A}} = T^{-1}\sum_i i^{-1}\sum_{j\le i} a_{i,j}$, with no general ordering relative to FP. Native SEMA instead aggregates examples for its logged total at each stage. We report that metric alongside task-macro results. Native CODA's logged global curve is task-macro. These related metrics are not independent confirmations. Tables report percentages, and differences are percentage points (pp).

Frozen-feature selection. Corrected capacity runs retain the earlier backbone-specific cosine-head configuration as an operating point fixed before these corrected comparisons. Each arm is evaluated both at that fixed setting and after equal learning-rate (LR) selection over five multipliers $\{0.1, 0.3, 1, 3, 10\}$, holding its other settings fixed. Selection uses a 15% stratified split of training data and seeds 0–1. Final fitting restores the full training set and evaluates seeds 0–9. Test examples never select a configuration, but selection and evaluation random-seed sets overlap. "Retuned" therefore means equal LR selection, not an exhaustive architecture search. Appendix A.3 gives protocol and classifier details.

Placement and native protocols. The initial placement study selects learning rate from $\{3\times10^{-4}, 10^{-3}, 3\times10^{-3}\}$ on validation seeds 0–1, then evaluates disjoint seeds 10–14. Its 15% validation subset remains excluded from fitting. Each task receives 400 AdamW steps with batch 64 and weight decay 0.01. Native SEMA and CODA use pinned released code, published training configurations, and five seeds each. Interventions and compatibility patches are documented in Appendix A.5. A failed CODA no-prompt run was restarted with zero data-loader workers instead of four. This changes augmentation RNG streams, so no-prompt contrasts match seed labels and class order, not identical stochastic training trajectories.

The follow-up separates adapter LR $\{10^{-5}, 3\cdot10^{-5}, 10^{-4}, 3\cdot10^{-4}, 10^{-3}\}$ and head LR $\{3\cdot10^{-4}, 10^{-3}\}$, with validation seeds 0–1 and disjoint test seeds 20–24. Post-encoder cells use the full grid. Unfinished internal searches switch to a validation-seed-0 screen and seed-1 confirmation of the top two settings. One previously selected cell was completed after the initial compute cutoff under frozen settings: eight cells are now complete, while both ViT-B/16 internal cells remain excluded in full. Appendix A.2 records the selection rule and completion history.

The fixed-optimization routing follow-up holds the class-centroid internal4 rates and all training settings fixed, changes only routing, and reuses paired seed labels 20–24. Its single primary endpoint is the paired final-accuracy difference. AP, BWT, and drift summaries are descriptive. The hypothesis follows observed results, and the benchmark is not newly unseen (Appendix A.1).

Table 2: Corrected gated AURA final accuracy (%, mean ± SD, ten seeds). Full minus adapter bypass is in percentage points with a paired 95% CI. Every displayed bypass contrast passes Holm-adjusted equivalence at ± 1 point within its declared family. Equivalence allows small nonzero differences. Retuning selects among five learning-rate multipliers using validation data.

Encoder	LR regime	Full	Shared single	Bypass	Full − bypass [95% CI]
T5-Large	Fixed	63.83 ± 0.11	63.83 ± 0.11	63.84 ± 0.12	-0.01 $[-0.08, +0.06]$
	Retuned	63.83 ± 0.11	63.83 ± 0.11	63.84 ± 0.12	-0.01 $[-0.08, +0.06]$
BGE-Large	Fixed	57.06 ± 0.13	56.84 ± 0.14	57.35 ± 0.23	-0.29 $[-0.48, -0.09]$
	Retuned	56.98 ± 0.15	57.10 ± 0.20	57.35 ± 0.23	-0.36 $[-0.51, -0.21]$
Qwen3-1.7B	Fixed	43.14 ± 0.20	43.14 ± 0.20	43.17 ± 0.18	-0.03 $[-0.07, +0.01]$
	Retuned	43.14 ± 0.20	43.14 ± 0.20	43.17 ± 0.18	-0.03 $[-0.07, +0.01]$
DINOv2-L	Fixed	85.19 ± 0.09	84.97 ± 0.16	85.20 ± 0.08	-0.01 $[-0.09, +0.07]$
	Retuned	85.19 ± 0.09	84.97 ± 0.16	85.20 ± 0.08	-0.01 $[-0.09, +0.07]$

Uncertainty. We report seed-paired 95% t intervals and 90% intervals for two one-sided tests (TOST) of equivalence. Margins are ± 1 pp for frozen-feature capacity/placement and ± 0.5 pp for native methods. Holm correction is applied separately within each stated dataset/regime/metric family. The initial placement equivalence checks and all intervals are pointwise. Exact structural identities receive no fabricated p-value. Equivalence is margin- and protocol-specific. Nonsignificance alone does not establish it. Native no-prompt recovery and computational accounting are disclosed separately. All experiments ran on one NVIDIA GB10 node, sometimes concurrently. Recorded job duration is not a controlled speed benchmark. Appendices A.2–A.6 specify configurations, seed sets, selection rules, and statistical procedures.

5 Results

5.1 Small pool effects with prototype logit correction

Table 2 reports corrected AURA with cosine heads, ten paired seeds, and explicit retraining of reused adapters. Figure 1 shows the text-system pool effects. Full-versus-bypass FP is equivalent within ± 1 pp on all four feature sets, both at fixed settings and after equal LR selection. Every non-identical full/single and full/bypass comparison passes the family-adjusted equivalence test. Full and shared-single are identical on T5 and Qwen3 because the trigger allocates only one adapter. They are identities rather than statistical tests. DINOv2-L allocates ten adapters for ten tasks, while BGE allocates two.

The largest absolute full-minus-bypass mean effect is -0.36 pp on LR-retuned BGE. Bypass is slightly better. Full-minus-single on BGE changes from $+0.22$ pp at the fixed rate to -0.12 pp after LR selection. Correcting reuse therefore preserves the small-effect conclusion in this gated system, but not every sign or numerical value. Retraining a shared BGE adapter rather than freezing it after the first task costs 0.51 pp at the fixed configuration (95% CI $[-0.60, -0.41]$ for retrained-minus-frozen). This is a real training-policy effect even though it falls within the study's one-point practical margin. Settings selected identically across fixed and LR-selected regimes are not independent replications.

The margin matters: under the same Holm families, T5, Qwen3, and DINOv2-L also pass at ± 0.25 pp, whereas BGE does not pass at ± 0.5 pp in either regime. In the five-seed text follow-up, BGE task effects range from -1.70 to $+0.38$ pp despite its small macro effect (Appendix A.1). Aggregate equivalence is not a per-task guarantee.

5.2 The prototype gate changes what the pool contributes

The fixed-configuration factorial separates pool removal from prototype logit correction. On T5, BGE, and Qwen3, prototype correction improves FP by 37.26, 43.78, and 31.87 pp with the pool present, versus 19.42, 11.19, and 16.54 pp with it bypassed. The interactions in Eq. 2 are correspondingly 17.83, 32.58, and 15.34 pp, with intervals excluding zero (Table 3).

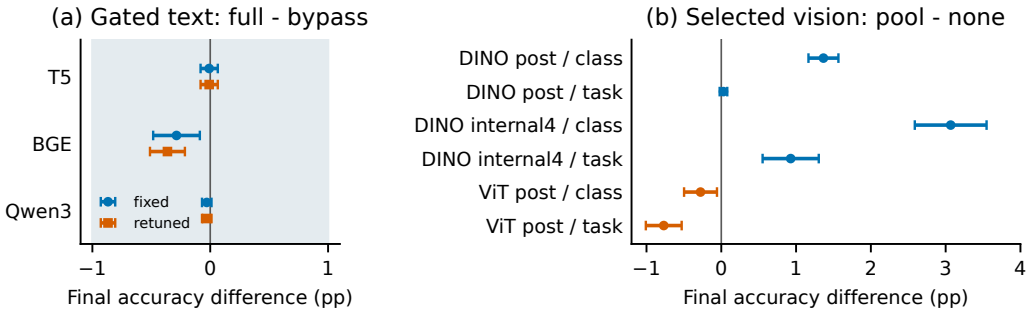


Figure 1: Module effects depend on the evaluated system. (a) Gated text full-minus-bypass FP: ten paired seeds, fixed and selected rates. Shading denotes the ± 1 pp margin. Repeated T5/Qwen3 settings are not independent replications. (b) All six completed selected vision pool-minus-none contrasts: five paired seeds. Bars are pointwise 95% intervals. Panels use different scales, systems, and protocols. Their difference is not a causal modality comparison. Table 5 separately fixes optimization to isolate a router intervention.

Table 3: Fixed-hyperparameter gate $\times$ pool factorial effects on final accuracy (percentage points). Interaction is the gate effect with a pool minus its effect without adapters. Paired 95% CIs are reported. This is a component interaction within this implementation, not a decomposition of all architectural benefit.

Encoder	Gate effect: pool	Gate effect: bypass	Interaction [95% CI]
T5-Large	37.26	19.42	$+17.83$ $[+16.41, +19.26]$
BGE-Large	43.78	11.19	$+32.58$ $[+32.35, +32.81]$
Qwen3-1.7B	31.87	16.54	$+15.34$ $[+15.00, +15.67]$
DINOv2-L	0.73	0.86	-0.13 $[-0.20, -0.06]$

These positive interactions do not mean the combined system earns a large gain over the gate alone. Without the gate, inserting the pool *reduces* FP by 17.84, 32.87, and 15.37 pp. With the gate, its effect is approximately zero. Prototype logit correction largely compensates for the pool-associated accuracy loss in these text configurations. On DINOv2-L, the interaction is only -0.13 pp, while the gate contributes 0.73–0.86 pp with or without the pool. The result is strongly dataset/encoder-dependent. LR-selected factorial contrasts retain separate labels because their optimization settings vary across arms. They are not the same controlled interaction.

Follow-up inference interventions support the prototype-score explanation. In the five-seed T5 full models, removing the gate only at inference changes macro FP from 63.85% to 7.70%, while prototype-only prediction reaches 62.35%. The corresponding BGE values are 57.05%, 8.78%, and 57.14%. This intervention is distinct from training without the gate. Across canonical, two random, and one cluster-blocked BGE order, mean full-minus-bypass effects range from -0.368 to -0.054 pp. All four pass the ± 1 pp equivalence test after correction across orders. These controls reuse seed labels from the capacity study and are not independent replications (Appendix A.2).

5.3 Placement under selected routing/optimization protocols

Table 4 reports eight complete five-seed placement cells under validation-selected head and adapter learning rates. Both routers use fixed unadapted reference features and matched training/inference mixtures without task identity.

On DINOv2-B, no adapters reaches 73.65% FP. A post-encoder pool with class-centroid routing reaches 75.02%, a $+1.37$ pp gain (95% CI $[1.17, 1.57]$). Four-block internal pools reach 76.72% with class-centroid routing, gaining 3.07 pp ($[2.59, 3.55]$), and 74.58% with task-mean routing, gaining 0.93 pp ($[0.55, 1.30]$). All three gains survive Holm correction across the six completed pool-versus-none contrasts. The post-encoder task-mean pool is nearly unchanged at 73.68%. For

Table 4: Completed expanded-search placement cells (five test seeds, FP in %, effects in pp). Task/class denote task-mean/class-centroid routing. All eight completed cells are shown. Both unfinished ViT-B/16 internal cells are excluded in full. Intervals are pointwise paired 95% intervals.

Encoder	Configuration / router	FP	Difference from none
DINOv2-B	None	73.65 ± 0.05	—
DINOv2-B	Post / class	75.02 ± 0.13	$+1.37\ [+1.17, +1.57]$
DINOv2-B	Post / task	73.68 ± 0.04	$+0.03\ [-0.02, +0.08]$
DINOv2-B	Internal4 / class	76.72 ± 0.34	$+3.07\ [+2.59, +3.55]$
DINOv2-B	Internal4 / task	74.58 ± 0.27	$+0.93\ [+0.55, +1.30]$
ViT-B/16	None	48.74 ± 0.17	—
ViT-B/16	Post / class	48.46 ± 0.28	$-0.28\ [-0.50, -0.06]$
ViT-B/16	Post / task	47.97 ± 0.26	$-0.77\ [-1.01, -0.53]$

Table 5: Fixed-optimization DINOv2-B internal4 routing control (five paired seeds, mean $\pm$ SD, %). Adapter/head LR are $3 \times 10^{-5}/3 \times 10^{-4}$ in both arms. Class minus task FP is 14.66 pp (95% CI $[13.73, 15.60]$, $p = 1.66 \times 10^{-6}$).

Router	FP	AP	BWT
Class-centroid	76.72 ± 0.34	90.17 ± 0.21	-14.95 ± 0.21
Task-mean	62.06 ± 0.53	92.77 ± 0.14	-34.12 ± 0.70

ViT-B/16, no adapters reaches 48.74%. Task-mean and class-centroid post-encoder pools reach 47.97% and 48.46%, respectively. Thus useful adapter configurations exist here, but the completed post-encoder controls do not show a benefit on ViT-B/16.

Both unfinished ViT-B/16 internal cells are excluded in full. Several selected rates lie on a grid boundary. Both internal DINOv2 selections use two-stage validation screening (Appendix A.2). These separately selected systems do not isolate a router or placement effect, and internal4 has four times the module count of post-encoder placement.

Changing only routing at fixed optimization. We therefore fit five additional task-mean internal4 runs at the class-centroid rates: adapter 3×10^{-5} and head 3×10^{-4}. Source, data, initialization rules, seed labels, batch order, steps, workers, and cached reference features are matched. Class-centroid routing reaches 76.72% FP versus 62.06% for task-mean, a $+14.66$ pp paired effect (Table 5). Task-mean has higher AP but much worse BWT: its failure is not an inability to fit incoming tasks. Descriptive routing measurements also show more diffuse mixtures and larger old-logit changes, without proving mediation (Appendix A.1).

This intervention was chosen after observing the existing results and is conditional on one operating point. The task-mean arm selected at adapter LR 10^{-5} reaches 74.58%, so the 14.66-point fixed-rate effect is not the separately tuned performance gap. Changing routing during both training and inference isolates that joint intervention, not inference-only routing or a universal router ranking. It nevertheless demonstrates that identical adapter capacity can be useful or harmful relative to the same no-adapter reference depending on routing at fixed optimization.

5.4 A STRONGER CLASSIFIER CHANGES THE PRACTICAL COMPARISON

Corrected RanPAC-style ridge classification reaches 74.82%, 65.84%, 68.50%, and 88.84% FP on T5, BGE, Qwen3, and DINOv2-L, respectively (Table 6). Relative to LR-selected AURA, the gains are 10.99, 8.86, 25.36, and 3.65 pp, all with seed-paired intervals excluding zero. Even ridge regression without random projection outperforms AURA on all four feature sets. NCM is weaker and does not establish this practical comparison by itself.

Our baseline uses accumulated sufficient-statistic generalized cross-validation (GCV), not the upstream RanPAC current-task holdout rule. It is explicitly a RanPAC-style adaptation (McDonnell et al., 2023). These established classifiers limit AURA's practical significance: removing its adapters preserves accuracy but leaves it substantially below an available alternative. At projection width

Table 6: Corrected analytic baselines on the same frozen features. Final accuracy is reported in %. AURA and random-projection ridge report mean ± SD over ten seeds. NCM and unprojected ridge are deterministic repeated checks, so no inferential SD is shown. Ridge regularization uses training-only GCV. The random projection has 10,000 dimensions. AURA uses the retuned regime.

Encoder	Full AURA	NCM	Ridge (no proj.)	RanPAC-style
T5-Large	63.83 ± 0.11	62.44	70.64	74.82 ± 0.21
BGE-Large	56.98 ± 0.15	56.76	64.53	65.84 ± 0.29
Qwen3-1.7B	43.14 ± 0.20	43.09	66.24	68.50 ± 0.17
DINOv2-L	85.19 ± 0.09	83.64	87.15	88.84 ± 0.14

10,000, the implementation stores approximately 810–849 MiB of persistent tensors, versus 9–34 MiB without projection (Table 28). Analytically fitted does not mean parameter-free or memory-free.

5.5 Native positive control and metric-sensitive diagnostic

CODA-Prompt: positive control for allocation benefit. On the custom ImageNet-R split, native full CODA reaches 75.49% FP, versus 73.96% with one component per task and 70.59% without prompts (Table 7). Full-minus-reduced is $+1.54$ pp (95% CI $[0.79, 2.29]$, $p_{\text{Holm}} = 0.0047$). This positive control retains the published training settings and quantifies allocation benefit with paired uncertainty on the stated split. It bounds the small-effect AURA result rather than claiming to discover that capacity can matter. The reduced arm forgets less but learns less: its AP falls by 3.09 pp while BWT improves by 1.72 pp. Better retention alone does not compensate for the learning loss.

Full-minus-no-prompt is $+4.91$ pp (95% CI $[4.51, 5.30]$). This contrast is less tightly controlled because the no-prompt recovery changed data-loader workers and hence augmentation RNG streams (Appendix A.5). It supports a benefit under the documented recovery protocol, not a bitwise-matched intervention. No failed or partial seed was included.

SEMA: diagnostic of metric-sensitive conclusions. Original-split VTAB produces 88.32% task-macro FP for full SEMA, 89.11% with expansion disabled, and 87.47% without adapters. Disabling expansion increases macro FP by 0.78 pp (95% CI $[0.30, 1.27]$, $p_{\text{Holm}} = 0.022$), while lowering AP by 0.38 pp and improving BWT by 1.45 pp. Full SEMA adds 8.6 ± 0.55 modules beyond twelve initial modules, so this is not a near-zero-expansion case. For macro FP, no-adapter minus full is -0.85 pp with interval $[-2.13, 0.42]$: neither superiority nor equivalence is established.

The native example-weighted final metric ranks full above expansion-disabled SEMA, 88.67% versus 88.26%. The disabled-minus-full effect is -0.40 pp (95% CI $[-0.87, 0.07]$), and its 90% interval extends beyond the ± 0.5 pp margin. In contrast, the native example-weighted *average-incremental* metric is equivalent: disabled-minus-full is -0.16 pp, with 90% CI $[-0.40, 0.08]$ and Holm-adjusted TOST $p = 0.039$. Removing adapters lowers that metric by 1.87 pp (95% CI $[1.26, 2.49]$ for full-minus-no-adapter). Thus the supported equivalence concerns a particular aggregation and trajectory metric, not all notions of final performance.

The fourth VTAB task accounts for 62.65% of examples. Its negative expansion-disabled effect outweighs gains on the first two tasks under example weighting (Appendix A.1). This explains the aggregation reversal within our runs, not the discrepancy with published SEMA ablations. The one-seed upstream bridge does not resolve that discrepancy. The native result remains a diagnostic rather than a general recommendation to disable expansion.

6 Limitations

Conditional scope. The audit identifies failure modes in particular systems, not universal adapter or prompt behavior. AURA remains below analytic baselines. CODA uses a custom split, and its no-prompt recovery changes worker-driven augmentation RNG streams. The SEMA bridge covers one

Table 7: Native capacity interventions. Final task-macro and official accuracy are mean ± SD (%, five seeds). Effects are intervention minus full (percentage points, paired 95% CI). Official CODA aggregation is task-macro. SEMA uses example weights, which reverse the sign of its expansion effect.

Method	Intervention	Task-macro FP	Official FP	Δ macro [95% CI]
CODA-Prompt	Full	75.49 ± 0.43	75.49 ± 0.43	—
	1/task	73.96 ± 0.65	73.96 ± 0.65	$-1.54\ [-2.29, -0.79]$
	No prompts	70.59 ± 0.31	70.59 ± 0.31	$-4.91\ [-5.30, -4.51]$
SEMA (VTAB)	Full	88.32 ± 0.97	88.67 ± 0.67	—
	Expansion off	89.11 ± 1.08	88.26 ± 0.83	$+0.78\ [+0.30, +1.27]$
	No adapters	87.47 ± 0.58	87.25 ± 0.60	$-0.85\ [-2.13, +0.42]$

paired seed and leaves the published no-expansion discrepancy unresolved. Aggregation explains a within-run reversal, not that reproduction gap.

Optimization and controls. Several selected rates lie at grid boundaries. Internal DINOv2 uses adaptive validation screening, and expanded post/internal comparisons are not parameter-matched. The fixed-rate control isolates a joint training/inference router intervention at one inherited operating point, chosen after observing results. It does not compare each router's optimum or identify a unique mechanism. Both ViT internal cells and the matched-worker CODA extension remain incomplete and excluded.

Uncertainty and data. Seed intervals condition on the datasets and orders. Reused seeds and test images are not independent replications. Aggregate equivalence depends on the chosen tolerance and need not hold per task. Original text sampling/extraction lineage is incomplete. Near-duplicates and pretraining contamination remain possible. Oracle routes use unavailable task identity. Concurrent timing is not a controlled speed comparison.

7 Conclusion

Allocation-only ablations can miss whether modules are useful. In AURA text systems, prototype correction largely compensates for pool-associated harm, while a strong analytic classifier remains better. In internal DINOv2 adapters, changing only the router at fixed optimization changes final accuracy by 14.66 points, despite identical adapter capacity. Separate tuning substantially improves the weaker router. These diagnoses motivate reporting subsystem removal and reduced allocation alongside classifier-correction, routing, and optimization controls. CODA's allocation benefit and SEMA's metric sensitivity bound the conclusion: useful capacity is conditional on the evaluated system, rather than generally redundant.

AI use statement

Generative AI assistants were used in this work across the categories below. The authors take full responsibility for all content.

Research ideation and execution. AI assisted in designing and giving feedback on the audit methodology, the factorial and inference interventions, the corrective analyses, and the follow-up experiments. AI also assisted in implementing the experiment code, the evaluation harness, the baseline ports, and the figure and table generation scripts.

Mathematical claims and proofs. AI assisted in formulating the allocation properties in Appendix A.14 and in constructing and writing their proofs, including the three-centroid counterexample establishing order dependence. The authors verified each statement, construction, and proof step.

Retrieval and discovery. AI was used to search for and summarize related literature and to identify candidate baselines and comparison points. Bibliographic details were checked against primary sources, and reported literature values were looked up per citation rather than taken from model output.

Drafting and writing. AI drafted sections of the manuscript from author-supplied results, notes, and artifacts, and was used for language editing, readability, and structural revision of author-written text.

Interpretation and review. AI assisted in interpreting experimental results and in critically reviewing the manuscript's claims against the recorded experiment artifacts.

Limits and responsibility. Reported empirical values come from experiment artifacts rather than model output. AI-generated suggestions are not treated as experimental evidence. No synthetic data were generated. The authors are responsible for checking the final claims, citations, proofs, code, and artifacts.

ETHICS STATEMENT

This study uses public classification benchmarks and introduces no new human-subject data collection. Dataset sources are cited and their original licenses apply. Findings are scoped to the evaluated implementations and protocols. Modified frozen-feature ports should not be read as evaluations of the original algorithms. Implementation defects identified during the audit concern our experimental harness unless explicitly stated otherwise.

REPRODUCIBILITY STATEMENT

The method equations, evaluation metrics, hyperparameters, selection rules, and statistical procedures are specified in the paper and appendices. Supplementary code, configurations, and complete seed-level results support independent reproduction. A companion contains exact cached features. Raw datasets, pretrained weights, and large training checkpoints are not bundled. The original NLP sampling and earliest T5 extraction lineage remain incompletely reconstructible. Appendix A.7 separates exact-feature reproduction from raw-data reconstruction. Appendix A.2 records the initial compute cutoff and subsequent completion of a cell under frozen settings. Appendices A.3–A.5 document the earlier complete protocols. Appendix A.6 describes statistical families and resource accounting. The fixed-optimization router intervention, narrower-margin sensitivity, and task decompositions are specified in Appendix A.1. Their portable checker recomputes tables from bundled raw matrices. Historical and corrected experiment versions are identified separately.

REFERENCES

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642. Association for Computational Linguistics, 2015.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017. doi: 10.1109/JPROC.2017.2675998. URL https://arxiv.org/abs/1703.00121.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3606–3613, 2014.

Bill Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*, 2005.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszko-

reit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2010.11929.

Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys)*, 2019.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8340–8349, 2021.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799. PMLR, 2019. URL http://proceedings.mlr.press/v97/houlsby19a.html.

Wei-Cheng Huang, Chun-Fu Chen, and Hsiang Hsu. OVOR: OnePrompt with virtual outlier regularization for rehearsal-free class-incremental learning. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/827e6353044227937b77ef0f166354a3-Abstract-Conference.html.

Paul Janson, Wenxuan Zhang, Rahaf Aljundi, and Mohamed Elhoseiny. A simple baseline that questions the use of pretrained-models in continual learning. In *NeurIPS 2022 Workshop on Distribution Shifts*, 2022.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations (ICLR)*, 2022.

Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning (ICML 1995)*, pp. 331–339. Morgan Kaufmann, 1995.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195, 2015.

Jiangyang Li, Chenhao Ding, Songlin Dong, Qiang Wang, Jianchao Zhao, Yuhang He, and Yihong Gong. Is parameter isolation better for prompt-based continual learning? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/papers/Li_Is_Parameter_Isolation_Better_for_Prompt-Based_Continual_Learning_CVPR_2026_paper.pdf.

Yan-Shuo Liang and Wu-Jun Li. InfLoRA: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23638–23647, June 2024.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150. Association for Computational Linguistics, 2011.

Andrea Madotto, Zhaojiang Lin, Zhenpeng Zhou, Seungwhan Moon, Paul A. Crook, Bing Liu, Zhou Yu, Eunjoon Cho, Pascale Fung, and Chien-Sheng Wu. Continual learning in task-oriented dialogue systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pp. 7452–7467. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EMNLP-MAIN.590. URL https://doi.org/10.18653/v1/2021.emnlp-main.590.

Mark D. McDonnell, Dong Gong, Amin Parveneh, Ehsan Abbasnejad, and Anton van den Hengel. Ranpac: Random projections and pre-trained models for continual learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *European Conference on Computer Vision (ECCV)*, 2020.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. URL https://arxiv.org/abs/2304.07193.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2021*, pp. 487–503. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EACL-MAIN.39. URL https://doi.org/10.18653/v1/2021.eacl-main.39.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. URL http://jmlr.org/papers/v21/20-1307.html.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 11909–11919. IEEE, 2023. doi: 10.1109/CVPR52729.2023.01146. URL https://doi.org/10.1109/CVPR52729.2023.01146.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642. Association for Computational Linguistics, 2013.

Chenyang Song, Xu Han, Zhengyan Zhang, Huadong Wang, Kaihuo Zhang, Yingfa Chen, Chaojun Xiao, Haozhe Ji, Zhiyuan Liu, and Maosong Sun. Conpet: Continual parameter-efficient tuning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024. doi: 10.18653/V1/2024.FINDINGS-ACL.580. URL https://doi.org/10.18653/v1/2024.findings-acl.580.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32:323–332, 2012.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355. Association for Computational Linguistics, 2018.

Huiyi Wang, Haodong Lu, Lina Yao, and Dong Gong. Self-expansion of pre-trained models with mixture of adapters for continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025*, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/papers/Wang_Self-Expansion_of_Pre-trained_Models_with_Mixture_of_Adapters_for_Continual_CVPR_2025_paper.pdf.

Liyuan Wang, Jingyi Xie, Xingxing Zhang, Mingyi Huang, Hang Su, and Jun Zhu. Hierarchical decomposition of prompt-based continual learning: Rethinking obscured sub-optimality. *Advances in Neural Information Processing Systems*, 2023a.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10658–10671, 2023b. doi: 10.18653/V1/2023.FINDINGS-EMNLP.715. URL https://doi.org/10.18653/v1/2023.findings-emnlp.715.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer G. 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. Springer, 2022a. doi: 10.1007/978-3-031-20083-0_37. URL https://doi.org/10.1007/978-3-031-20083-0_37.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer G. Dy, and Tomas Pfister. Learning to prompt for continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022*, pp. 139–149. IEEE, 2022b. doi: 10.1109/CVPR52688.2022.00024. URL https://doi.org/10.1109/CVPR52688.2022.00024.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122. Association for Computational Linguistics, 2018.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. *arXiv preprint arXiv:2309.07597*, 2023. URL https://arxiv.org/abs/2309.07597. Introduces the BGE embedding models, including bge-large-en-v1.5.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388. Qwen Team, Alibaba Group. The backbone used here is the Qwen3-1.7B base model, mean-pooled, not the separate Qwen3-Embedding family.

Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. SLCA: Slow learner with classifier alignment for continual learning on a pre-trained model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19148–19158, 2023. URL https://arxiv.org/abs/2303.05118.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28*, pp. 649–657. Curran Associates, Inc., 2015.

Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *International Journal of Computer Vision*, 2024a.

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*, pp. 23554–23564, 2024b.

A Appendix

A.1 Fixed-optimization Routing Control and Sensitivity Analyses

Protocol and provenance. After observing the completed placement results, we froze a single follow-up: task-mean routing at the existing DINOv2-B internal4/class-centroid operating point. The only scientific intervention is the router used in both training and inference. Five full fits use seeds 20–24, adapter LR 3×10^{-5}, head LR 3×10^{-4}, bottleneck 64, route temperature 0.05, 400 AdamW steps per task, batch 64, weight decay 0.01, and zero data-loader workers. They retain the ten-task split and excluded validation rows, cached unadapted CLS features, initialization and sampler seed rules, frozen encoder revision, and training source of the reference arm. This fixed control is not a new task-mean validation winner. It was planned after the original results and reuses test images and seed labels, so the paired interval describes variation over these training seeds, not a new benchmark replication. All five runs completed and passed source, argument, split, cache, matrix, and diagnostic checks. No additional search or interrupted seed enters the comparison.

Statistical interpretation. The primary contrast is class-centroid minus task-mean FP, with one two-sided paired t test and its pointwise 95% interval. It forms a separate follow-up analysis, not an extra member silently added to the original six pool-versus-none comparisons. Its 14.66 pp effect and interval $[13.73, 15.60]$ agree under independent raw-matrix recomputation. AP and BWT in Table 5 are descriptive and linked by Eq. 3. They are not additional independent confirmations. No test-based tuning follows this result. The 74.58% task-mean score under its separately selected adapter LR 10^{-5} remains visible in Table 4, preventing the fixed-rate gap from being read as an optimized router ranking.

Routing diagnostics. On the existing fixed 128-example-per-task diagnostic subsets, final-stage mean mixture entropy is 0.718 for class-centroid and 2.017 for task-mean routing (natural-log units). Averaging within seed over all 45 old-task transitions and then over seeds gives old-class logit RMS changes of 0.174 and 0.340, and mass assigned to newly allocated sets of 0.044 and 0.111, respectively. These trajectories are consistent with more diffuse routing and greater score drift in the lower-FP arm. They neither identify a causal mediator nor separate training-time from inference-time routing effects. The task subsets and repeated stages are correlated measurements, not additional replicates.

Equivalence-margin sensitivity. The one-point primary tolerance expresses the study's practical resolution, not a universal deployment requirement. Table 8 recomputes TOST at narrower margins without changing means, seeds, or the original eight-contrast Holm families. The BGE conclusion requires the one-point margin. T5, Qwen3, and DINOv2-L pass even at 0.25 points. This sensitivity analysis was added after the primary results and does not replace the original margin.

Task-level heterogeneity. Table 9 shows every canonical text task in the five-seed follow-up. BGE losses on SNLI, CoLA, and MRPC exceed one point even though the macro effect is small. Thus aggregate equivalence cannot justify a task-wise guarantee. These task means are descriptive. No task-level independence or additional significance claim is assumed.

SEMA aggregation and the unresolved reproduction gap. Table 10 decomposes the expansion-disabled minus full effect. Task 4 receives 62.65% weight by examples but 20% by tasks. Its -1.36 pp effect contributes approximately -0.85 pp to the example-weighted aggregate, outweighing gains on tasks 1 and 2. Rounded task logs reconstruct approximately $+0.78$ pp macro and -0.40 pp weighted differences, consistent with the native logs to rounding precision.

This within-run explanation does not reconcile our results with Wang et al. (2025), Table 2. That table reports VTAB full/no-expansion final example-weighted accuracy of 89.64/83.66% and average-incremental accuracy of 91.26/86.21%. Our five-seed counterparts are 88.67/88.26% and 90.85/90.69%. The gap persists with matched metric definitions. The pinned released-code configuration, threshold intervention, training schedule, and data counts are specified in Appendix A.5. The upstream-versus-patched bridge at seed 1993 matches stage classifier hashes and recorded raw accuracy when evaluation cadence matches (Appendix A.2). This single-seed check cannot establish

Table 8: Full-minus-bypass FP equivalence at three margins (pp), ten paired seeds. Each margin retains all eight original contrasts in its dataset/regime Holm family, including undefined identity tests as $p = 1$. No means insufficient evidence of equivalence, not established nonequivalence. The original primary margin remains 1 pp.

Encoder / rates	Mean	± 0.25	± 0.5	± 1
T5 / fixed	-0.01	Yes	Yes	Yes
T5 / selected	-0.01	Yes	Yes	Yes
BGE / fixed	-0.29	No	No	Yes
BGE / selected	-0.36	No	No	Yes
Qwen3 / fixed	-0.03	Yes	Yes	Yes
Qwen3 / selected	-0.03	Yes	Yes	Yes
DINOv2-L / fixed	-0.01	Yes	Yes	Yes
DINOv2-L / selected	-0.01	Yes	Yes	Yes

Table 9: All canonical text-task full-minus-bypass FP effects with prototype correction, averaged over the five follow-up seeds (pp). Tasks are correlated components of the same stream, not independent replications.

Task	T5	BGE
sst2	-0.05	$+0.07$
ag_news	-0.24	$+0.14$
mnli	-0.34	-0.32
imdb	-0.74	-0.12
dbpedia	-0.02	$+0.38$
snli	$+0.48$	-1.70
cola	$+0.16$	-1.43
yelp	-0.18	-0.04
yahoo	$+0.04$	-0.16
qnli	-0.46	-0.34
mrpc	$+0.54$	-1.32
amazon	$+0.22$	$+0.04$
newsgroup20	-0.25	-0.43
rte	$+0.00$	$+0.08$

five-seed reproduction or the cause of the published discrepancy. We retain SEMA as an aggregation diagnostic under the stated protocol, not a refutation of its published expansion benefit.

A.2 Completed Follow-up Controls and Compute Cutoff

At the September 15 compute cutoff, seven placement cells had complete five-seed tests. On September 17, we restarted the previously selected DINOv2-B internal/class-centroid cell under its frozen settings and completed all five seeds that day, without further search. The final follow-up includes 40 gate-factorial runs, 30 additional-order runs, and 40 placement test runs across eight cells. The four historical interrupted runs remain excluded. Only the five complete new runs enter the added cell. Both ViT-B/16 internal cells lack complete five-seed tests and are excluded in full. No partial result is used to rank configurations or substantiate a claim. The matched-worker CODA extension did not finish. Six SEMA bridge runs cover only seed 1993 and are labeled as a pilot below. These incomplete controls limit scope. They are not negative evidence.

Gate and order controls. The follow-up retains each backbone's fixed cosine-head configuration and corrected retraining of reused adapters. The four training cells independently vary pool bypass and prototype gating. Seeds 0–4 and the canonical order overlap earlier capacity evidence. This is a diagnostic extension, not an independent replication. Inference interventions evaluate the same fitted model with raw logits, prototype-corrected logits, and training-updated prototype scores alone. No test label fits a gate or scale. For BGE, the two predefined random orders use seeds 101 and 202 for task permutations. The blocked order groups sentiment, topic, inference, and remaining binary

Table 10: SEMA expansion-disabled minus full: five-seed mean final task effects and their aggregate contributions (pp). Weighted reconstruction uses rounded task logs, so it can differ slightly from native global logs.

Task	Test n	Effect	Macro contribution	Weighted contribution
1	1,000	$+3.26$	$+0.65$	$+0.38$
2	400	$+2.35$	$+0.47$	$+0.11$
3	985	-0.30	-0.06	-0.03
4	5,400	-1.36	-0.27	-0.85
5	834	-0.02	-0.00	-0.00

Table 11: Follow-up fixed-configuration gate factorial, five seeds (FP, %). These runs reuse seed labels and settings from the larger capacity suite and are not independent replications.

Encoder	Pool + gate	Bypass + gate	Pool, no gate	Bypass, no gate
T5	63.85	63.90	25.60	44.62
BGE	57.05	57.41	13.33	46.23

tasks. Full and bypass share each order and seed. The canonical order reuses the matching gate runs. Treating all order–seed observations as independent would exaggerate evidence.

The gated pool differences remain within the one-point margin for both encoders and across all four BGE orders. The large gate-by-pool interactions do not imply a large advantage over bypass with gating. Removing the gate only at inference is also not equivalent to training without it: the raw classifier in a gate-trained model can rely on correction during optimization. Prototype-only accuracy being close to gated accuracy supports a large contribution from the prototype scores, without isolating a unique causal mechanism.

Expanded placement optimization. The no-adapter head uses rates $\{3 \cdot 10^{-4}, 10^{-3}, 3 \cdot 10^{-3}\}$. Adapter cells use all combinations of adapter rates $\{10^{-5}, 3 \cdot 10^{-5}, 10^{-4}, 3 \cdot 10^{-4}, 10^{-3}\}$ and head rates $\{3 \cdot 10^{-4}, 10^{-3}\}$. All retain 400 AdamW steps per task, batch size 64, bottleneck 64, weight decay 0.01, route temperature 0.05, and the original fixed split. The 15% validation subset remains excluded during test fitting. Selection uses validation seeds 0–1. Final tests use disjoint seeds 20–24.

Post-encoder cells complete the full two-seed grid. Before reading their test results, the unfinished internal searches were amended to rank every candidate by seed 0 validation FP, break ties by lower adapter LR then lower head LR, confirm only the top two candidates on seed 1, and select by their two-seed mean. Already-running seed 1 jobs were allowed to finish, but any outside the shortlist were excluded from selection. Both completed DINOv2-B internal cells use this rule. The task-mean shortlist uses adapter LR 10^{-5}. The class-centroid shortlist uses $3 \cdot 10^{-5}$. Each pairs that rate with head LR $3 \cdot 10^{-4}$ or 10^{-3} and selects the former. The September 17 completion retains the earlier selection. Table 15 reports all completed cells' selected rates. This is a reduced validation budget, not an exhaustive two-seed search. Boundary optima do not establish that the search brackets the best rate.

Router and drift measurements. Task-mean routing uses one mean unadapted reference feature per task. The class-centroid alternative stores normalized per-class training-feature means. For an input, the score for a task is the maximum cosine similarity to its class centroids. A softmax across task scores supplies mixture weights. Both use cached unadapted encoder features and the same mixtures at training and inference. Neither receives the inference task ID. Rates are selected separately for each router, so their test contrast incorporates the selection protocol. The post-encoder pool has one module per task. Internal4 has four, so their direct comparison is not parameter-matched. The earlier parameter-matched study is retained separately below.

After every task, the first 128 fixed evaluation examples per seen task record mixture entropy, mean mixture weights, mass on newly allocated sets, change in old mixture weights, and RMS change in old-class logits relative to the previous stage. These are descriptive trajectories on a fixed subset, not additional independent seeds. Ten tasks yield 55 stage/task measurements per completed run. These

Table 12: Follow-up factorial effects (pp). Holm correction covers the three contrasts within each encoder.

Encoder	Contrast	Effect [95% CI]	Adjusted p
T5	Pool, gate on	-0.06 $[-0.16, +0.04]$	0.165
T5	Pool, gate off	-19.02 $[-20.31, -17.72]$	$6.5e-06$
T5	Interaction	$+18.96$ $[+17.64, +20.28]$	$6.5e-06$
BGE	Pool, gate on	-0.37 $[-0.66, -0.08]$	0.024
BGE	Pool, gate off	-32.90 $[-33.16, -32.64]$	$1.16e-09$
BGE	Interaction	$+32.53$ $[+32.19, +32.87]$	$2.38e-09$

Table 13: Inference-only interventions on the same trained models (task-macro %). Changed is the fraction of predictions changed by adding the prototype gate, averaged over tasks and seeds. No retraining is performed.

Trained arm	Raw	Gated	Prototype only	Changed
T5 / full	7.70	63.85	62.35	93.47
T5 / bypass	7.76	63.90	62.35	93.59
T5 / fullnogate	25.60	63.35	63.26	70.20
T5 / bypassnogate	44.62	63.41	63.21	44.80
BGE / full	8.78	57.05	57.14	91.23
BGE / bypass	16.88	57.41	57.20	88.67
BGE / fullnogate	13.33	56.85	56.83	81.88
BGE / bypassnogate	46.23	56.51	56.50	33.24

diagnostics do not enter model selection. Oracle-route accuracy is a diagnostic using unavailable task identity and is not substituted for the reported router.

Native bridge and norm-alignment pilot. At seed 1993, pinned upstream SEMA and the patched source with upstream evaluation cadence have identical initial-parameter hashes and identical classifier-weight hashes after every task, in both full and expansion-disabled arms. Their final raw predictions produce identical recorded accuracies. The sparse-evaluation variant is kept distinct because evaluation cadence can alter random-number consumption during training. One seed does not establish five-seed reproducibility or resolve every discrepancy with published scores.

The norm-alignment diagnostic derives γ from the mean old-class weight norm divided by the mean new-class weight norm, rescales only the new-class weight contribution, retains biases, and never changes fitted parameters. Test labels do not determine γ. This is an inference-only training-norm diagnostic, not a reproduction of a complete alternative training method. Saved logits reproduce the raw and aligned summaries, and old logits remain unchanged by alignment. In Table 16, `single` means expansion disabled with twelve initial modules, `eval` retains upstream evaluation cadence, and `sparse` omits per-epoch evaluation. All completed pilot cells are reported without seed-level significance tests. The original five-seed SEMA results retain their separate protocol label.

Evidence boundary. Tables in this subsection and the main placement comparison summarize complete seed sets only. Their means, paired differences, intervals, and multiplicity adjustments use the definitions in Appendix A.6. Interrupted and uncompleted cells contribute no observations. Appendix A.4 reports the earlier restricted-grid study separately, so readers can compare its scope and outcomes with the expanded search.

A.3 Corrected Frozen-feature Protocols

The capacity suite contains eight arms: retrained-reuse full AURA, retrained shared-single, frozen-reuse full, frozen shared-single, one adapter per task, pool bypass, full without prototype correction, and bypass without prototype correction. All use cosine heads. Fixed comparisons use the same configuration within each feature set. LR-selected comparisons give every arm five learning rates and two validation seeds, retaining the other parameters. Test fitting uses the entire original training split. The validation split is stratified by class with fraction 0.15 and split seed 12345. Tuning seeds

Table 14: BGE full minus bypass across four orders (five paired seeds each, pp). Canonical reuses the gate runs. TOST margin is ± 1 pp. Corrections cover the four orders.

Order	Effect [95% CI]	Adjusted p	Adjusted TOST p
canonical	-0.37 $[-0.66, -0.08]$	0.0721	0.00185
random_101	-0.12 $[-0.25, +0.02]$	0.0721	$5.5e-05$
random_202	-0.24 $[-0.33, -0.15]$	0.00636	$2.58e-05$
cluster_blocked	-0.05 $[-0.10, -0.01]$	0.0721	$1.15e-06$

Table 15: Validation-selected learning rates for the completed placement cells. The no-adapter rate applies only to the head. Boundary optima remain a limitation.

Encoder	Configuration	Adapter LR	Head LR	Selection
DINOv2-B	None	—	0.001	Full grid
DINOv2-B	Post / class	0.0003	0.0003	Full grid
DINOv2-B	Post / task	1e-05	0.001	Full grid
DINOv2-B	Internal4 / class	3e-05	0.0003	Adaptive
DINOv2-B	Internal4 / task	1e-05	0.0003	Adaptive
ViT-B/16	None	—	0.001	Full grid
ViT-B/16	Post / class	1e-05	0.0003	Full grid
ViT-B/16	Post / task	1e-05	0.0003	Full grid

0–1 overlap final seeds 0–9. Held-out test examples do not enter selection. The earlier configuration was selected with 30 validation trials for the full model, so the architecture settings are not independently optimized for each corrected arm.

AdamW and the cosine schedule are shared within each feature set. The full and shared-single BGE arms select 0.3 times the base LR. All other full/shared/bypass capacity arms retain the base rate. With prototype correction disabled, full AURA selects multipliers $(0.3, 0.3, 1, 1)$ and bypass selects $(3, 1, 3, 1)$ for T5-Large, BGE-Large, Qwen3-1.7B, and DINOv2-L, respectively. Frozen-reuse and per-task arms retain the base rate on every feature set. There are 640 fixed/LR-selected test records, 320 validation records, and 120 analytic-baseline records. Fixed and LR-selected evaluations with identical configurations can yield identical matrices and are not independent evidence.

Corrected analytic classifiers. NCM fits class means. Ridge without projection fits a regularized linear readout. The RanPAC-style arm applies a frozen Gaussian projection of width 10,000 and ReLU before ridge fitting. All three classifiers first standardize each input across its d coordinates, $\tilde{x}_k = (x_k - \bar{x})/\sqrt{d^{-1}\sum_l (x_l - \bar{x})^2 + 10^{-5}}$, without learned affine parameters. NCM averages these standardized training features within each class and predicts the largest cosine similarity to a class mean. Unprojected ridge uses $\tilde{x}$ directly. Projected ridge uses $h(x) = \mathrm{ReLU}(\tilde{x}P)$, with independent $P_{kl} \sim \mathcal{N}(0, 1/d)$ drawn once per seed (0–9) and held fixed across tasks. Prediction selects the largest readout score among all seen classes. With feature matrix H and one-hot targets Y, the sufficient statistics are $G = H^\top H$, $C = H^\top Y$, sample count n, and $q = \|Y\|_F^2$. After every task,

$$W_\lambda = (G + \lambda I)^{-1} C, \qquad d_\lambda = \mathrm{tr}[G(G + \lambda I)^{-1}], \tag{4}$$

$$R_\lambda = q - 2\,\mathrm{tr}(W_\lambda^\top C) + \mathrm{tr}(W_\lambda^\top G W_\lambda), \tag{5}$$

$$\mathrm{GCV}(\lambda) \propto R_\lambda / (n - d_\lambda)^2. \tag{6}$$

The grid is $\lambda \in \{10^{-4}, 10^{-3}, \ldots, 10^{6}\}$. Float64 accumulated statistics and a symmetric eigendecomposition evaluate this criterion without replaying prior examples. The omitted proportionality factor n is constant across candidates. This corrects the historical surrogate residual, but remains different from upstream RanPAC's current-task 80/20 holdout selection (McDonnell et al., 2023). NCM and unprojected ridge are deterministic under this fixed feature/order protocol. Their repeated seed records should not be interpreted as ten different fits with independent randomness.

Table 16: SEMA bridge pilot, seed1993 only. Raw and inference-only norm-aligned final accuracies (%). This is a diagnostic, not a five-seed replacement for Table 7.

Variant / arm	Raw macro	Raw weighted	Aligned macro	Aligned weighted
patched_eval_full	89.02	88.11	88.96	88.07
patched_eval_single	89.74	87.55	89.76	87.53
patched_sparse_full	89.50	88.91	89.41	88.84
patched_sparse_single	90.42	88.85	90.31	88.75
upstream_full	89.02	88.11	88.96	88.07
upstream_single	89.74	87.55	89.76	87.53

Table 17: Prespecified capacity operating points. LR selection multiplies the listed LR by $0.1, 0.3, 1, 3$, or 10 and changes no other entry. Routing and prototype temperatures are separately learned parameters initialized to the same listed value.

Parameter	T5-Large	BGE-Large	Qwen3-1.7B	DINOv2-L
Feature dimension	1024	1024	2048	1024
Adapter bottleneck	64	512	64	256
Batch size	64	128	32	128
Steps/task	679	668	878	845
Warmup steps	154	58	34	33
Base LR	$3.230\,10^{-5}$	$4.223\,10^{-5}$	$2.715\,10^{-5}$	$5.070\,10^{-5}$
Weight decay	$6.943\,10^{-5}$	$2.778\,10^{-3}$	$3.875\,10^{-5}$	$5.978\,10^{-4}$
Dropout	0.2846	0.4774	0.1327	0.3593
Expansion threshold	0.6324	0.7589	0.8152	0.9292
Temperature initial value	0.01794	0.002277	0.0002443	0.09035
Prototype scale initial value	29.676	24.110	6.861	21.152
Residual scale initial value	$3.984\,10^{-5}$	$1.760\,10^{-5}$	$4.720\,10^{-3}$	$1.014\,10^{-5}$
Prototype EMA momentum	0.9765	0.9579	0.9788	0.9902

A.4 Corrected Placement Protocol and Complete Results

This subsection records the initial restricted-grid study. Its all-negative adapter outcomes are superseded as a general placement conclusion by the completed expanded-search cells in Table 4. The protocols have different grids and seed sets. Their numerical differences do not isolate one intervention.

The two encoders share the same ten-task ImageNet-R membership and image split. A class-stratified 15% validation subset is removed from the training images using split seed 20260908 and remains excluded during test fitting. Unadapted final CLS features are cached in FP32 and used for every router. Internal adapters receive images. Post-encoder adapters receive cached features. Both use the same fixed reference for routing and the same soft mixture at training and inference. The active adapter set is trainable on its task. Shared-single is trainable on every task, while prior pool sets and heads are frozen. Head initialization and task data-order seeds are matched across arms without allowing different adapter counts to consume those RNG streams.

Placement module and classifier. Both the initial and expanded placement protocols use the residual bottleneck

$$A(\mathbf{h}) = \mathbf{h} + W_{up}\,\mathrm{GELU}(W_{down}\,\mathrm{LN}(\mathbf{h}) + \mathbf{b}_{down}) + \mathbf{b}_{up},$$

with zero-initialized up-projection weights and bias. Internal modules act on every token after each selected complete transformer block. Post-encoder modules act on the CLS representation after the encoder's final LayerNorm. Encoder weights remain frozen and the encoder remains in evaluation mode during training. Per-task cosine heads normalize features and class weights and use a learned scale initialized to 16. These placement bottlenecks do not include AURA's channel/output gates, dropout, or learned residual multiplier.

Every cell uses a bottleneck of 64, routing temperature 0.05, 400 AdamW steps per task, batch 64, and weight decay 0.01. Learning rate is selected from $3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}$ by mean validation

Table 18: Complete corrected AURA cells on T5-Large. Mean ± seed SD (%, ten seeds). Fixed and validation-retuned regimes are separate. Frozen-reuse arms retain the historical behavior explicitly as controls.

Regime	Intervention	FP	AP	BWT
Fixed	Full (retrain reuse)	63.83 ± 0.11	69.78 ± 0.10	-6.41 ± 0.12
Fixed	Shared single (retrain)	63.83 ± 0.11	69.78 ± 0.10	-6.41 ± 0.12
Fixed	Full (frozen reuse)	63.84 ± 0.10	69.61 ± 0.14	-6.21 ± 0.15
Fixed	Single (frozen reuse)	63.84 ± 0.10	69.61 ± 0.14	-6.21 ± 0.15
Fixed	Per-task pool	63.85 ± 0.12	69.57 ± 0.11	-6.16 ± 0.12
Fixed	Bypass adapters	63.84 ± 0.12	69.61 ± 0.13	-6.22 ± 0.17
Fixed	Full, gate off	26.57 ± 1.99	76.12 ± 0.21	-53.37 ± 2.14
Fixed	Bypass, gate off	44.41 ± 0.51	74.04 ± 0.28	-31.91 ± 0.49
Retuned	Full (retrain reuse)	63.83 ± 0.11	69.78 ± 0.10	-6.41 ± 0.12
Retuned	Shared single (retrain)	63.83 ± 0.11	69.78 ± 0.10	-6.41 ± 0.12
Retuned	Full (frozen reuse)	63.84 ± 0.10	69.61 ± 0.14	-6.21 ± 0.15
Retuned	Single (frozen reuse)	63.84 ± 0.10	69.61 ± 0.14	-6.21 ± 0.15
Retuned	Per-task pool	63.85 ± 0.12	69.57 ± 0.11	-6.16 ± 0.12
Retuned	Bypass adapters	63.84 ± 0.12	69.61 ± 0.13	-6.22 ± 0.17
Retuned	Full, gate off	33.28 ± 3.30	60.02 ± 0.83	-28.80 ± 3.66
Retuned	Bypass, gate off	49.99 ± 0.17	75.39 ± 0.08	-27.35 ± 0.23

Table 19: Complete corrected AURA cells on BGE-Large. Mean ± seed SD (%, ten seeds). Fixed and validation-retuned regimes are separate. Frozen-reuse arms retain the historical behavior explicitly as controls.

Regime	Intervention	FP	AP	BWT
Fixed	Full (retrain reuse)	57.06 ± 0.13	63.20 ± 0.16	-6.61 ± 0.17
Fixed	Shared single (retrain)	56.84 ± 0.14	63.34 ± 0.14	-7.00 ± 0.12
Fixed	Full (frozen reuse)	57.29 ± 0.18	63.11 ± 0.12	-6.27 ± 0.13
Fixed	Single (frozen reuse)	57.35 ± 0.22	63.12 ± 0.17	-6.21 ± 0.13
Fixed	Per-task pool	57.29 ± 0.11	63.14 ± 0.12	-6.30 ± 0.10
Fixed	Bypass adapters	57.35 ± 0.23	63.11 ± 0.17	-6.21 ± 0.13
Fixed	Full, gate off	13.29 ± 0.21	70.21 ± 0.13	-61.30 ± 0.26
Fixed	Bypass, gate off	46.15 ± 0.16	67.06 ± 0.22	-22.51 ± 0.18
Retuned	Full (retrain reuse)	56.98 ± 0.15	62.03 ± 0.15	-5.43 ± 0.10
Retuned	Shared single (retrain)	57.10 ± 0.20	62.15 ± 0.17	-5.44 ± 0.08
Retuned	Full (frozen reuse)	57.29 ± 0.18	63.11 ± 0.12	-6.27 ± 0.13
Retuned	Single (frozen reuse)	57.35 ± 0.22	63.12 ± 0.17	-6.21 ± 0.13
Retuned	Per-task pool	57.29 ± 0.11	63.14 ± 0.12	-6.30 ± 0.10
Retuned	Bypass adapters	57.35 ± 0.23	63.11 ± 0.17	-6.21 ± 0.13
Retuned	Full, gate off	39.78 ± 0.74	64.14 ± 0.31	-26.22 ± 0.93
Retuned	Bypass, gate off	46.15 ± 0.16	67.06 ± 0.22	-22.51 ± 0.18

FP over seeds 0–1. Exact ties select the smaller rate. Test seeds are 10–14. The minimal tier has six cells, reused unchanged inside the eighteen-cell extended tier: 108 unique validation runs and 90 test runs in total. A no-adapter model has zero adapter parameters. A one-module-per-task pool has 1,006,720. Four- and eight-module stacks have 4,026,880 and 8,053,760. Shared versions contain one tenth as many. Post-encoder placement stacks one, four, or eight downstream adapters per allocated set, matching the corresponding internal depth without modifying encoder blocks.

Primary evaluation never receives task identity. Oracle routing selects the task's adapter set only as a post hoc diagnostic and is excluded from LR selection. Its improvement over routed inference is a sensitivity result, not an attainable task-agnostic score or a proof that routing alone explains the loss. Collapse is a descriptive FP threshold at twice global-class chance (1% for 200 classes). Nonfinite loss is an execution failure handled separately. Each selected cell has zero observed collapses among five seeds, but the exact 95% binomial interval is $[0, 52.18\%]$. We do not pool correlated cells into an artificially precise stability claim.

Table 20: Complete corrected AURA cells on Qwen3-1.7B. Mean ± seed SD (%, ten seeds). Fixed and validation-retuned regimes are separate. Frozen-reuse arms retain the historical behavior explicitly as controls.

Regime	Intervention	FP	AP	BWT
Fixed	Full (retrain reuse)	43.14 ± 0.20	52.93 ± 0.24	-10.54 ± 0.35
Fixed	Shared single (retrain)	43.14 ± 0.20	52.93 ± 0.24	-10.54 ± 0.35
Fixed	Full (frozen reuse)	43.17 ± 0.17	52.86 ± 0.23	-10.44 ± 0.33
Fixed	Single (frozen reuse)	43.17 ± 0.17	52.86 ± 0.23	-10.44 ± 0.33
Fixed	Per-task pool	43.22 ± 0.32	52.96 ± 0.36	-10.49 ± 0.32
Fixed	Bypass adapters	43.17 ± 0.18	52.88 ± 0.23	-10.46 ± 0.33
Fixed	Full, gate off	11.26 ± 0.37	61.82 ± 0.25	-54.45 ± 0.34
Fixed	Bypass, gate off	26.63 ± 0.23	59.27 ± 0.29	-35.15 ± 0.36
Retuned	Full (retrain reuse)	43.14 ± 0.20	52.93 ± 0.24	-10.54 ± 0.35
Retuned	Shared single (retrain)	43.14 ± 0.20	52.93 ± 0.24	-10.54 ± 0.35
Retuned	Full (frozen reuse)	43.17 ± 0.17	52.86 ± 0.23	-10.44 ± 0.33
Retuned	Single (frozen reuse)	43.17 ± 0.17	52.86 ± 0.23	-10.44 ± 0.33
Retuned	Per-task pool	43.22 ± 0.32	52.96 ± 0.36	-10.49 ± 0.32
Retuned	Bypass adapters	43.17 ± 0.18	52.88 ± 0.23	-10.46 ± 0.33
Retuned	Full, gate off	11.26 ± 0.37	61.82 ± 0.25	-54.45 ± 0.34
Retuned	Bypass, gate off	29.12 ± 0.26	61.63 ± 0.13	-35.02 ± 0.29

Table 21: Complete corrected AURA cells on DINOv2-L. Mean ± seed SD (%, ten seeds). Fixed and validation-retuned regimes are separate. Frozen-reuse arms retain the historical behavior explicitly as controls.

Regime	Intervention	FP	AP	BWT
Fixed	Full (retrain reuse)	85.19 ± 0.09	90.39 ± 0.08	-5.77 ± 0.12
Fixed	Shared single (retrain)	84.97 ± 0.16	90.78 ± 0.06	-6.45 ± 0.19
Fixed	Full (frozen reuse)	85.19 ± 0.09	90.39 ± 0.08	-5.77 ± 0.12
Fixed	Single (frozen reuse)	85.19 ± 0.07	90.37 ± 0.06	-5.75 ± 0.07
Fixed	Per-task pool	85.19 ± 0.09	90.39 ± 0.08	-5.77 ± 0.12
Fixed	Bypass adapters	85.20 ± 0.08	90.34 ± 0.06	-5.71 ± 0.06
Fixed	Full, gate off	84.46 ± 0.05	92.05 ± 0.04	-8.43 ± 0.06
Fixed	Bypass, gate off	84.34 ± 0.04	92.01 ± 0.03	-8.52 ± 0.05
Retuned	Full (retrain reuse)	85.19 ± 0.09	90.39 ± 0.08	-5.77 ± 0.12
Retuned	Shared single (retrain)	84.97 ± 0.16	90.78 ± 0.06	-6.45 ± 0.19
Retuned	Full (frozen reuse)	85.19 ± 0.09	90.39 ± 0.08	-5.77 ± 0.12
Retuned	Single (frozen reuse)	85.19 ± 0.07	90.37 ± 0.06	-5.75 ± 0.07
Retuned	Per-task pool	85.19 ± 0.09	90.39 ± 0.08	-5.77 ± 0.12
Retuned	Bypass adapters	85.20 ± 0.08	90.34 ± 0.06	-5.71 ± 0.06
Retuned	Full, gate off	84.46 ± 0.05	92.05 ± 0.04	-8.43 ± 0.06
Retuned	Bypass, gate off	84.34 ± 0.04	92.01 ± 0.03	-8.52 ± 0.05

A.5 Native Implementations, Metrics, and Recovery

SEMA on original VTAB. The pinned SEMA commit is `5a4551262d659c6d93dd6c50ff814e028554fbf3`. We use the authors' VTAB archive, published five-task configuration, and unshuffled class order. Seeds are 1993–1997. Functional-adapter and representation-descriptor phases each use 20 epochs, with SGD, batch 32, weight decay 5×10^{-4}, functional LR 0.005 and descriptor LR 0.01. The functional bottleneck is 16 and descriptor width 128. Expansion threshold is 1 and expandable block indices are 9–11 (zero-based). Twelve modules are initially created. Expansion-disabled sets the threshold to 10^9. No-adapter disables the adapter flag. Compatibility changes allow the no-adapter arm to bypass absent modules and omit descriptor optimization when no descriptors exist. Optional per-epoch evaluation is disabled. Final stage evaluations remain intact.

The original archive contains test-task sizes 1,000, 400, 985, 5,400, and 834. Its corresponding training-task sizes are 221, 218, 268, 1,000, and 89. Therefore the fourth task contributes 62.65% of final example-weighted accuracy but only 20% of task-macro accuracy. This unequal weighting

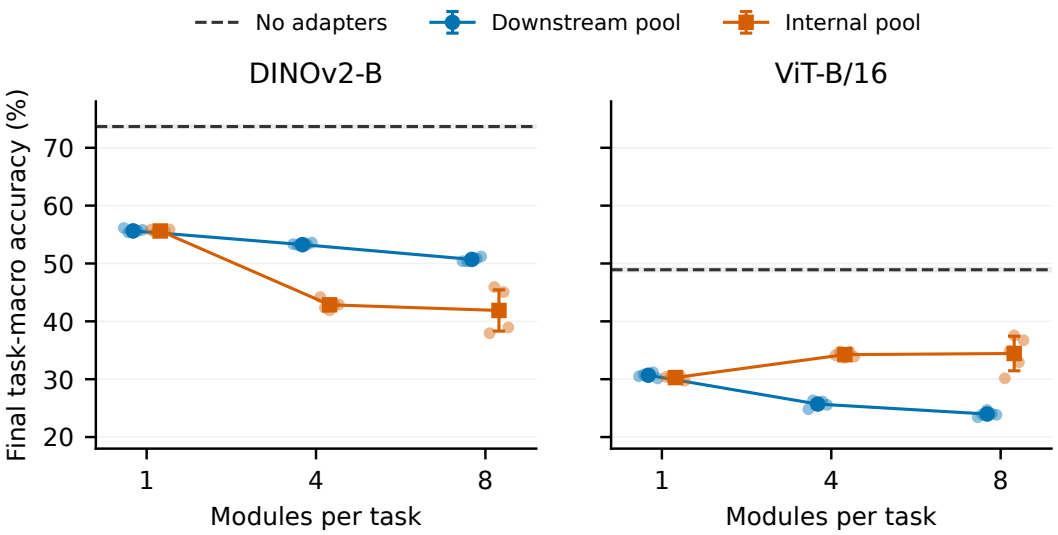


Figure 2: Initial corrected restricted-grid placement study with parameter-matched pools on ImageNet-R. Each point uses an independently validation-selected learning rate and five held-out seeds. Bars are seed SD, faint points are individual seeds. Internal adapters occupy the last 1, 4, or 8 blocks. Downstream controls stack the same number of bottlenecks after the encoder. A fixed frozen final-CLS router is shared across placements and used in both training and testing. Every adapted arm in this initial study remains below the no-adapter baseline. The expanded search identifies positive DINOv2 configurations (Table 4).

explains how the aggregate rankings can differ without a metric-computation error. Per-task matrices show the underlying changes. Native logged class-block percentages are rounded to two decimals, and derived macro metrics inherit that precision. Only runs completing all five stages with the intended class blocks are included.

CODA-Prompt on a custom ImageNet-R split. Native ViT-B/16 uses ten tasks of twenty classes, batch 128, Adam with LR 0.001 and zero weight decay, cosine scheduling, 50 epochs per task, and training augmentation. The full prompt pool has 100 components, the reduced pool 10, and prompt length is eight. With ten tasks these allocate ten or one component per task, respectively. The no-prompt path retains the classifier on the frozen encoder. Compatibility guards enable that path without changing full/reduced prompt training. We use seed labels 0–4 and the implementation's class permutations. The 19,455/4,865 image lists are our exported split, not the published filename lists, so headline published accuracies are not directly comparable.

No-prompt failure and restart. A four-worker no-prompt process failed during the final task of its first seed because the data loader could not allocate shared memory. No complete seed was available. All five reported no-prompt seeds were restarted from fresh initialization with zero workers, preserving optimizer, LR, batch size, epochs, task count, and class-order seeds. Completed full/reduced outputs were unchanged. Worker count changes augmentation RNG streams, so matching seed labels does not create identical stochastic trajectories across those arms. This is a remaining confound for the no-prompt contrast, explicitly separated from the full/reduced comparison.

A no-prompt-only efficiency patch disables encoder gradients that its classifier-only optimizer never updates. Behavioral tests across task transitions and a CUDA ViT check establish identical logits, classifier gradients, and classifier updates under identical inputs and RNG state. This verifies that computational change, not equivalence of the distinct worker-induced data sequences. Only the five fully completed restarted seeds enter the no-prompt summary. Native CODA stage accuracy is the unweighted mean of the seen tasks' accuracies, with the same task-macro definition used for FP.

Table 22: All corrected placement test cells (five seeds 10–14). M is total stored adapter parameters in millions. Post stacks bottlenecks after the encoder, Internal places them in the last indicated number of blocks. FP is mean ± SD (%). Oracle gap is mean oracle-current-task-adapter FP minus routed FP (points), a diagnostic using task identity. Runtime excludes the shared frozen-reference cache pass. No cell reaches the prespecified FP $\leq 1\%$ collapse proxy. Five zero events still give a two-sided 95% exact binomial (Clopper–Pearson) upper bound of 52.18%, not a stability guarantee.

Encoder	Arm, location, modules	M	Selected LR	FP (%)	Oracle gap	Seconds
DINOv2-B	None, Post, 0	0.00	10^{-3}	73.67 ± 0.13	0.00	189.43
DINOv2-B	Pool, Post, 1	1.01	3×10^{-4}	55.65 ± 0.35	22.14	265.01
DINOv2-B	Pool, Post, 4	4.03	3×10^{-4}	53.27 ± 0.25	32.85	582.06
DINOv2-B	Pool, Post, 8	8.05	3×10^{-4}	50.71 ± 0.35	34.83	852.42
DINOv2-B	Pool, Internal, 1	1.01	3×10^{-4}	55.62 ± 0.23	24.41	5895.66
DINOv2-B	Pool, Internal, 4	4.03	10^{-3}	42.86 ± 0.86	12.91	6809.79
DINOv2-B	Pool, Internal, 8	8.05	3×10^{-3}	41.88 ± 3.56	7.43	11988.92
DINOv2-B	Shared, Post, 1	0.10	3×10^{-4}	31.82 ± 0.24	0.00	262.56
DINOv2-B	Shared, Internal, 4	0.40	3×10^{-3}	25.81 ± 0.84	0.00	6488.67
ViT-B/16	None, Post, 0	0.00	10^{-3}	48.92 ± 0.23	0.00	188.45
ViT-B/16	Pool, Post, 1	1.01	3×10^{-4}	30.68 ± 0.39	36.68	269.54
ViT-B/16	Pool, Post, 4	4.03	3×10^{-4}	25.70 ± 0.60	37.15	626.80
ViT-B/16	Pool, Post, 8	8.05	3×10^{-4}	23.98 ± 0.46	38.42	898.40
ViT-B/16	Pool, Internal, 1	1.01	10^{-3}	30.28 ± 0.33	24.31	4450.01
ViT-B/16	Pool, Internal, 4	4.03	10^{-3}	34.24 ± 0.50	7.36	5204.35
ViT-B/16	Pool, Internal, 8	8.05	3×10^{-3}	34.44 ± 2.99	18.73	7222.76
ViT-B/16	Shared, Post, 1	0.10	3×10^{-4}	12.61 ± 0.46	0.00	192.60
ViT-B/16	Shared, Internal, 4	0.40	10^{-3}	21.45 ± 1.52	0.00	2678.17

Table 23: Complete paired placement contrasts for DINOv2-B, in percentage points. Intervals are pointwise 95% paired t intervals. Equivalence is pointwise 90% CI containment within ±1 point, without multiplicity adjustment. Each comparison uses the same five held-out seed IDs and validation-selected rates.

Contrast	Mean [95% CI]	Equivalent
Pool post1 − None	-18.02 $[-18.54, -17.51]$	No
Pool post1 − Shared post1	$+23.83$ $[+23.25, +24.40]$	No
Pool post4 − None	-20.40 $[-20.80, -20.01]$	No
Pool post8 − None	-22.96 $[-23.52, -22.41]$	No
Pool internal1 − None	-18.05 $[-18.45, -17.64]$	No
Pool internal1 − Pool post1	-0.02 $[-0.27, +0.23]$	Yes
Pool internal4 − None	-30.81 $[-31.93, -29.69]$	No
Pool internal4 − Pool post4	-10.41 $[-11.33, -9.48]$	No
Pool internal4 − Shared internal4	$+17.05$ $[+15.84, +18.26]$	No
Pool internal8 − None	-31.79 $[-36.17, -27.41]$	No
Pool internal8 − Pool post8	-8.83 $[-13.37, -4.28]$	No
Shared post1 − None	-41.85 $[-42.19, -41.51]$	No
Shared internal4 − None	-47.86 $[-48.88, -46.84]$	No

A.6 Statistical Families and Resource Accounting

For a paired contrast, let d_s be the difference in percentage points between arms at seed s, n the number of complete seed pairs, $\bar{d}$ their mean, and s_d their sample standard deviation. The pointwise $100(1-\alpha)\%$ interval is $\bar{d} \pm t_{n-1,1-\alpha/2} s_d/\sqrt{n}$. The two-sided test uses $t = \bar{d}/(s_d/\sqrt{n})$ with $n-1$ degrees of freedom. For equivalence margin δ, the two one-sided tests reject means at or below $-\delta$ and at or above δ: their statistics are $(\bar{d}+\delta)/(s_d/\sqrt{n})$ and $(\bar{d}-\delta)/(s_d/\sqrt{n})$, respectively. The TOST p-value is the larger of the upper-tail probability for the first statistic and the lower-tail probability for the second. For multiplicity-adjusted equivalence claims, we require its Holm-adjusted value to be below 0.05. The initial placement tables explicitly report unadjusted pointwise equivalence. For sorted values $p_{(1)} \leq \cdots \leq p_{(m)}$, Holm-adjusted values are $\min\{1, \max_{j\leq i}(m-j+1)p_{(j)}\}$.

Table 24: Complete paired placement contrasts for ViT-B/16, in percentage points. Intervals are pointwise 95% paired t intervals. Equivalence is pointwise 90% CI containment within ± 1 point, without multiplicity adjustment. Each comparison uses the same five held-out seed IDs and validation-selected rates.

Contrast	Mean [95% CI]	Equivalent
Pool post1 − None	-18.24 $[-18.71, -17.76]$	No
Pool post1 − Shared post1	$+18.07$ $[+17.42, +18.72]$	No
Pool post4 − None	-23.22 $[-23.95, -22.49]$	No
Pool post8 − None	-24.94 $[-25.26, -24.61]$	No
Pool internal1 − None	-18.64 $[-19.09, -18.18]$	No
Pool internal1 − Pool post1	-0.40 $[-0.63, -0.18]$	Yes
Pool internal4 − None	-14.67 $[-15.45, -13.90]$	No
Pool internal4 − Pool post4	$+8.54$ $[+7.94, +9.15]$	No
Pool internal4 − Shared internal4	$+12.80$ $[+10.69, +14.90]$	No
Pool internal8 − None	-14.48 $[-18.02, -10.93]$	No
Pool internal8 − Pool post8	$+10.46$ $[+7.15, +13.77]$	No
Shared post1 − None	-36.31 $[-37.06, -35.55]$	No
Shared internal4 − None	-27.47 $[-29.42, -25.52]$	No

Table 25: Complete validation selection grid. Entries are final validation accuracy (%) averaged over seeds 0–1. Each cell has the same three-rate budget. Selection maximizes these scores before running held-out test seeds 10–14. Post controls with four/eight modules match the corresponding internal pool's parameter count.

Encoder	Arm, location, modules	3×10^{-4}	10^{-3}	3×10^{-3}	Chosen LR
DINOv2-B	None, Post, 0	71.32	72.68	72.34	10^{-3}
DINOv2-B	Pool, Post, 1	54.43	54.05	46.40	3×10^{-4}
DINOv2-B	Pool, Post, 4	52.43	47.97	37.88	3×10^{-4}
DINOv2-B	Pool, Post, 8	50.14	44.25	34.78	3×10^{-4}
DINOv2-B	Pool, Internal, 1	54.76	53.47	44.46	3×10^{-4}
DINOv2-B	Pool, Internal, 4	43.18	44.47	42.00	10^{-3}
DINOv2-B	Pool, Internal, 8	37.91	38.87	46.68	3×10^{-3}
DINOv2-B	Shared, Post, 1	30.81	29.78	25.14	3×10^{-4}
DINOv2-B	Shared, Internal, 4	16.02	23.67	23.98	3×10^{-3}
ViT-B/16	None, Post, 0	46.30	46.69	46.43	10^{-3}
ViT-B/16	Pool, Post, 1	29.33	26.74	21.12	3×10^{-4}
ViT-B/16	Pool, Post, 4	25.16	20.77	14.86	3×10^{-4}
ViT-B/16	Pool, Post, 8	23.22	18.86	13.21	3×10^{-4}
ViT-B/16	Pool, Internal, 1	27.11	28.91	26.12	10^{-3}
ViT-B/16	Pool, Internal, 4	32.93	33.10	29.14	10^{-3}
ViT-B/16	Pool, Internal, 8	32.96	34.04	34.05	3×10^{-3}
ViT-B/16	Shared, Post, 1	11.62	10.04	9.27	3×10^{-4}
ViT-B/16	Shared, Internal, 4	19.61	20.13	18.17	10^{-3}

A pointwise 90% interval alone does not establish multiplicity-adjusted equivalence. Zero-variance structural identities are reported descriptively without these divisions.

The corrected capacity analysis prespecifies eight contrasts per dataset, optimization regime, and metric: full minus shared-single, full minus bypass, retrained minus frozen reuse for full and shared-single, per-task minus shared-single, gate effect with and without the pool, and the factorial interaction. Two-sided tests and TOST use separate Holm families of eight. Analytic-baseline comparisons against full form separate families of three. Native methods use two comparisons against full per metric. Undefined identical-arm tests count toward family size, with $p = 1$ used only for adjustment bookkeeping, not as an inferential result. Reported 90% and 95% intervals are pointwise. Holm-adjusted equivalence decisions use adjusted TOST values. Contrasts are computed before rounding the displayed arm means. Placement's prespecified intervals are pointwise and its many contrasts share baselines. For the completed expanded placement study, the six pool-minus-none FP tests form one Holm family across encoders and routers. The gate follow-up adjusts three contrasts

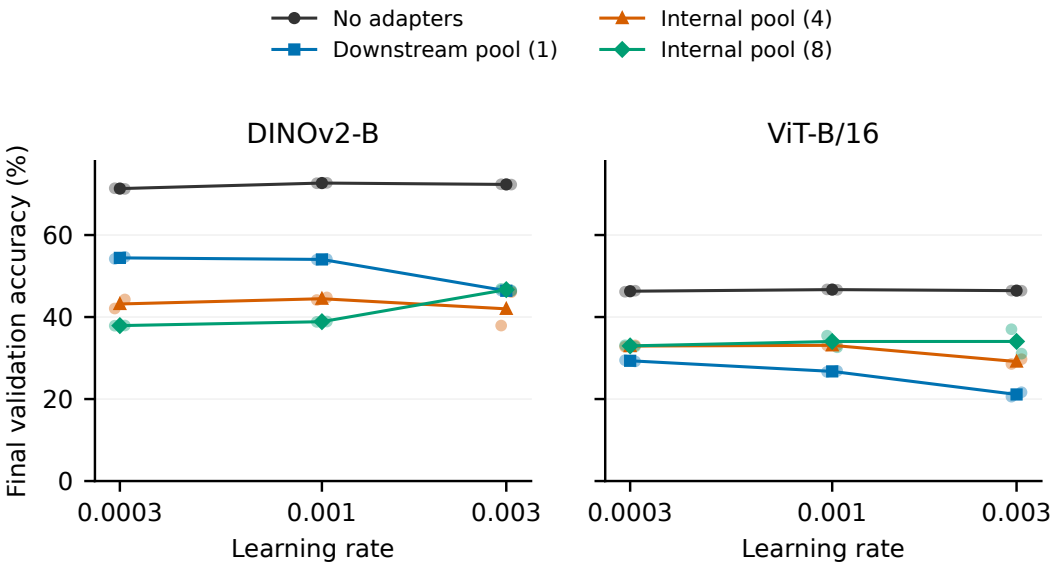


Figure 3: Validation learning-rate curves for the no-adapter, one-bottleneck downstream pool, and four/eight-block internal pools. Lines average validation seeds 0–1. Faint points are both seeds, not confidence intervals. Selection uses only these validation scores. Reported test results use seeds 10–14. Several adapted arms select a boundary of the finite grid, limiting conclusions about their best attainable performance. All 18 cells and all three rates are provided in Table 25.

Table 26: Additional native metrics (%, mean ± SD, five seeds). AP is mean diagonal accuracy. $\bar{A}$ averages seen-task accuracy over training stages. CODA counts prompt components (100 vs. 10 total). SEMA counts bottlenecks (12 initial in the expansion-disabled arm). These intervention sizes are method-specific.

Method	Intervention	AP	BWT	$\bar{A}$ (macro)	Components
CODA-Prompt	Full	79.77 ± 0.41	-4.75 ± 0.27	80.64 ± 0.81	100.00 ± 0.00
	1/task	76.68 ± 0.58	-3.03 ± 0.41	78.94 ± 0.86	10.00 ± 0.00
	No prompts	76.34 ± 0.66	-6.39 ± 0.47	75.91 ± 0.79	0.00 ± 0.00
SEMA (VTAB)	Full	92.48 ± 0.36	-5.19 ± 0.81	89.74 ± 0.79	20.60 ± 0.55
	Expansion off	92.10 ± 0.39	-3.74 ± 0.92	90.08 ± 0.87	12.00 ± 0.00
	No adapters	91.11 ± 0.28	-4.56 ± 0.95	88.01 ± 0.30	0.00 ± 0.00

within each encoder. The order follow-up adjusts four order contrasts, separately for two-sided tests and TOST. The fixed-rate routing follow-up has one primary FP test. Different metrics, repeated regimes, and related contrasts must not be counted as independent corroborations.

Persistent-state measurements count stored tensors, not only gradient-trained parameters. AURA pool bypass still allocates an unused adapter. Its zero forwarding capacity must not be reported as zero stored capacity. End-of-stream optimizer-eligible counts are measured after consolidation and are not the peak number trained during a task. Ridge state includes random projections, Gram/target statistics, and fitted coefficients. The 10,000-wide float64 Gram alone occupies 800 MB (decimal), before factorization workspace. Feature files, frozen encoder weights, and runtime allocation peaks are different quantities and are reported separately.

Jobs shared one NVIDIA GB10 node with concurrent training and, during recovery, a held controller. Some original CODA job durations include that held time. Other runs overlapped with different competing workloads. Those durations document execution, not comparative speed or energy efficiency. We draw no speed ratio from these timings.

Table 27: Sensitivity to each native implementation's official aggregation. Effects are intervention minus full in percentage points. Intervals are paired 95% CIs. p is the two-sided paired test with Holm correction within each method/metric's two interventions. These are not equivalence tests.

Method	Intervention	Δ official FP [95% CI]	p_{Holm}
CODA-Prompt	1/task	-1.54 $[-2.29, -0.79]$	0.005
CODA-Prompt	No prompts	-4.91 $[-5.30, -4.51]$	< 0.001
SEMA (VTAB)	Expansion off	-0.40 $[-0.87, +0.07]$	0.075
SEMA (VTAB)	No adapters	-1.42 $[-2.65, -0.18]$	0.067

Table 28: Analytic baseline resources and final accuracy. Persistent tensor memory is MiB (2^{20} bytes), including sufficient statistics and stored projections/coefficients. It excludes transient factorization workspace and frozen-feature storage. Runtime is mean seconds per complete run on the canonical compute host. Deterministic baselines omit SD. Projected ridge shows seed SD.

Encoder	Method	Persistent MiB	Seconds	FP (%)
T5-Large	NCM	0.55	6.56	62.44
T5-Large	Ridge, no projection	8.82	5.95	70.64
T5-Large	RanPAC-style ridge	810.01	999.86	74.82 ± 0.21
BGE-Large	NCM	0.55	3.96	56.76
BGE-Large	Ridge, no projection	8.82	10.40	64.53
BGE-Large	RanPAC-style ridge	810.01	1004.13	65.84 ± 0.29
Qwen3-1.7B	NCM	1.09	3.14	43.09
Qwen3-1.7B	Ridge, no projection	33.64	12.27	66.24
Qwen3-1.7B	RanPAC-style ridge	849.08	644.52	68.50 ± 0.17
DINOv2-L	NCM	1.56	1.06	83.64
DINOv2-L	Ridge, no projection	10.34	2.99	87.15
DINOv2-L	RanPAC-style ridge	824.89	451.75	88.84 ± 0.14

A.7 Data and Release Scope

The main corrected frozen-feature comparisons use the primary text stream and the 30,000-image ImageNet-R feature export (24,002 train and 5,998 test). DINOv2-L capacity and analytic baselines use identical feature-file hashes. This export differs from the capped 24,320-image collection used for native CODA and placement (19,455 train and 4,865 test). Placement additionally reserves its training-validation subset. The heterogeneous vision stream and two small semantic streams in the inventory belong to historical exploratory analyses. They do not support the corrected paper's central conclusions. The heterogeneous stream includes RESISC45, traffic signs, and textures (Cheng et al., 2017; Stallkamp et al., 2012; Cimpoi et al., 2014). Changing those domains also changes image difficulty and sample composition.

The anonymous source/results release includes experiment scripts, immutable runtime snapshots, exact configurations and split metadata, selection records, complete reported seed matrices, and analysis sources. A companion archive contains the exact cached feature CSVs. These artifacts support rerunning the experiments from the specified features and independently recomputing the reported statistics. They do not constitute complete raw-data provenance: original NLP sampling decisions and the earliest T5 extraction lineage are incompletely recoverable. Public raw images and pretrained encoder weights are obtained separately under their source licenses. Pretrained weights and large training checkpoints are not included in the source/results archive. Exact-feature reproduction therefore has a narrower scope than reconstruction starting with raw text.

A.8 Historical Baseline Adaptation Protocol

The nine historical baselines were modified to act on frozen precomputed embeddings and are denoted `-Flat`. Original methods inject adapters, low-rank updates, prompts, or fusion inside an encoder (Houlsby et al., 2019; Pfeiffer et al., 2021; Wang et al., 2023b; Madotto et al., 2021; Song et al., 2024; Wang et al., 2022b;a; Smith et al., 2023; Wang et al., 2025). Relocating those operations changes their mechanism and interaction with training. It is not merely a new input format.

Table 29: Task streams evaluated class-incrementally without task identity. The smaller text streams select different examples/classes from the primary embedding collection. Matching their counts does not match semantic difficulty or source distributions. The heterogeneous vision stream shares ImageNet-R's task/class counts and encoder but changes image domains and sample composition.

Stream	Modality	Tasks	Classes	Sampling	Encoders
Primary text	text	14	70	variable	T5-L, BGE-L, Qwen3
ImageNet-R (features)	vision	10	200	variable	DINOv2-L
ImageNet-R (capped)	vision	10	200	≤ 150	DINOv2-B, ViT-B/16
VTAB (native SEMA)	vision	5	50	variable	ViT-B/16
Heterogeneous	vision	10	200	$150^{\dagger}$	DINOv2-L
Collision (≈ 5 concepts)	text	7	14	98	T5-L
Disjoint (14 concepts)	text	7	14	98	T5-L

Primary text: 27,931 train and 11,984 test rows (39,915 total). Task/class counts are unequal. For example DBpedia has 1,988 training rows across 14 classes and 20News has 1,950 across 20. Corrected frozen-feature ImageNet-R uses 30,000 images (24,002 train, 5,998 test). The separate capped collection used by native CODA and placement contains 24,320 (19,455 train, 4,865 test before placement validation splitting). Native SEMA's VTAB split contains 1,796 train and 8,619 test images. †Three of the heterogeneous stream's four source domains supply 150 images per class. The textures component supplies 80, its split containing no more, so one task of ten is smaller than its ImageNet-R counterpart (§6).

For example, averaging selected prompt vectors into a downstream bias is not L2P's token-level prompting. Table 31 records the transformations for reproducibility. These variants are historical evidence about this harness, not reproductions of published accuracy or a common-protocol ranking of the original methods. The main native SEMA and CODA studies retain the authors' architectures and are reported under separate protocols.

A.9 Specification of AURA

Representations and residual adapter. The input $\mathbf{x}$ is a fixed cached feature. A learned shared LayerNorm produces $\mathbf{u} = \mathrm{LN}_{pre}(\mathbf{x})$. The fixed feature input does not imply that this normalization or the downstream classifier is fixed. Each adapter accepts $\mathbf{u}$, applies its own normalization, and computes

$$\mathbf{v} = \mathrm{softplus}(\boldsymbol{\gamma}) \odot \mathrm{LN}_1(\mathbf{u}), \tag{7}$$

$$\mathbf{h} = \mathrm{Dropout}(\mathrm{SiLU}(\mathrm{LN}_2(\mathbf{v}W_{down} + \mathbf{b}_{down}))), \tag{8}$$

$$\mathbf{g} = \mathrm{Sigmoid}(\mathrm{Mish}(\mathbf{u}W_{sq} + \mathbf{b}_{sq})W_{ex} + \mathbf{b}_{ex}), \tag{9}$$

$$A_m(\mathbf{u}) = \mathbf{u} + s_m\, \mathbf{g} \odot (\mathbf{h}W_{up} + \mathbf{b}_{up}). \tag{10}$$

Here $W_{down} \in \mathbb{R}^{d_h \times b}$, $W_{up} \in \mathbb{R}^{b \times d_h}$, $W_{sq} \in \mathbb{R}^{d_h \times r}$, $W_{ex} \in \mathbb{R}^{r \times d_h}$, and $r = \lfloor d_h/16 \rfloor$. The input gate starts at $\gamma_i = -1$. The down projection is orthogonally initialized and the up projection uses orthogonal gain 0.005 with zero bias. Residual scales, bottleneck widths, dropout, and EMA momentum use the per-feature settings in Table 17. A small initial residual limits the initial perturbation but does not guarantee later stability. Internal gates are components of the pool and are retained in the corrected pool-present factorial arms.

Allocation and routing. At task arrival, raw training-feature class means form $\mathcal{C}_k = \{\bar{\mathbf{x}}_{k,c}\}_c$. Allocation compares their cosine similarity to each existing adapter's stored raw centroid set:

$$\mathrm{sim}^* = \max_{m \leq N_a} \max_{\mathbf{c} \in \mathcal{C}_k,\, \mathbf{c}' \in \mathcal{C}_m^{(A)}} \cos(\mathbf{c}, \mathbf{c}').$$

A new adapter is created when the pool is empty or $\mathrm{sim}^* < \tau_{expand}$. Otherwise the maximizing adapter is reused with deterministic tie-breaking. New task centroids are appended to that adapter's set. Old stored centroids are not averaged away. Algorithm 1 gives the decision rule.

For prediction, affinity uses the pre-adapter representation $\mathbf{u}$ against those stored centroids: $q_m = \max_{\mathbf{c} \in \mathcal{C}_m^{(A)}} \cos(\mathbf{u}, \mathbf{c})$. Routing weights are $w_m = \exp(q_m/\tau_{adapt}) / \sum_j \exp(q_j/\tau_{adapt})$, and $\mathbf{z} = \sum_m w_m A_m(\mathbf{u})$. The shared normalization and routing temperature are trainable. Thus this AURA router is different from the explicitly fixed CLS router in the placement study.

Heads, prototypes, and optimization. Each task creates a cosine head: after dropout, normalized features are multiplied by normalized class weights and a learned scalar initialized at 16. The heads' independent scales may later differ. All seen-class logits are concatenated and receive the prototype correction in Eq. 1. Class prototypes are updated from the normalized training-batch class means by EMA. The correction uses $\epsilon = 10^{-9}$. Adapter and prototype temperatures are positive exponentiated log-parameters, initialized alike but learned separately. The prototype scale is learned directly. Cross-entropy is computed over all seen classes.

At a task boundary, existing adapters and heads are frozen, a new head is initialized, and the selected new or reused adapter is made trainable under the corrected policy. The shared pre-normalization, temperatures, and prototype scale continue training. After consolidation, task-specific parameters are frozen. Historical frozen-reuse arms leave the selected existing adapter frozen instead. Their new head and shared parameters still update. The explicit bypass ignores adapter outputs while retaining normalization, classifier heads, and, when enabled, prototype correction. Its unused allocated adapter is included in stored-state accounting but not forward-capacity counts.

A.10 Historical Measurements and Corrections

The historical experiments motivated the corrected study. They are not additional replications of its results. Their raw matrices and analysis code remain in the release under their original experiment identifiers. Three distinctions are essential. First, older AURA runs froze reused adapters after their first task, whereas the main capacity results retrain them and compare both policies explicitly. Second, early shared-protocol hyperparameters differed from later backbone-specific cosine-head configurations. Third, historical random-feature ridge selection used a normal-equation surrogate rather than sample-space GCV. The corrected analytic results in Table 6 supersede comparisons using that surrogate. Neither old ridge values nor untuned-versus-tuned comparisons justify margins against the corrected AURA models.

An additional AdapterFusion-Flat protocol error pooled previous tasks' training data during fusion. Reported corrected historical rows restrict fusion to the current task. Poor observed baseline performance was not evidence that its implementation was valid. More generally, the `-Flat` variants modify original architectures, and their internal contrasts cannot be substituted for native-method results (Appendix A.17).

A.11 Historical Class-dimension Normalization Defect

The original harness standardized each task head across its own class dimension before concatenating logits. For a zero-mean vector with unit population variance and c coordinates, the largest coordinate is bounded by $\sqrt{c-1}$ before learned affine scaling and ignoring numerical epsilon. This is an extremal bound, not an expected maximum. The measured mean untrained maxima were 0.97, 1.28, 1.81, and 1.95 for heads with 2, 4, 14, and 20 classes. Different head widths therefore entered the shared argmax with different score distributions.

In separately tuned historical T5 experiments, the full-minus-gate-floor FP gap was 53.75 pp with the class-dimension head, 14.21 pp with raw linear logits, and 12.38 pp with cosine heads. Here the gate-floor removes both the input-channel gate and prototype correction while retaining adapters (Appendix A.16). It is not an individual prototype-gate effect. The direct head replacement diagnoses a defect in this harness. Independent cosine-head temperatures can still diverge and do not guarantee cross-task calibration. The corrected main experiments use cosine heads and the explicit pool-by-prototype factorial instead of attributing a joint-removal gap to one component.

A historical regression relating gate contribution to task class count had slope -0.0409 and $R^2 = 0.71$, but treated repeated seeds of the same fourteen tasks as independent observations. We retain it only as descriptive context and do not use the invalid task-level p-value. The reported regression does not support an inferential claim about task class count. Historical transplant rankings also changed with the replacement head, so they do not establish carrier-independent sufficiency of the prototype term.

A.12 Historical Placement and Sparse-expansion Results

An earlier four-block DINOv2-B experiment at LR 3×10^{-3} gave pool-minus-none FP of $+7.59$ pp (95% CI $[7.33, 7.85]$). Its training used the active adapter set while inference mixed sets, and its routing representation changed with the first adapted layer. Increasing depth also increased parameter count. Learning-rate maxima were read from the test grid. Its zero-block pool contained no adapter parameters. That endpoint could not test a downstream adapter. The initial corrected restricted-grid experiment changes these jointly and yields negative pool-versus-none effects. The expanded search subsequently identifies positive DINOv2 configurations (Table 4). The old positive number is a measurement under a different implementation, not an independent confirmation of the corrected placement study. The difference cannot be attributed to one repair in isolation.

Historical native SEMA on two custom ImageNet-R class groupings added only 2.0 and 0.8 modules on average beyond twelve initial modules. Full versus expansion-disabled task-macro FP was equivalent within 0.5 pp, while no-adapter removal cost about 7.3 pp. Those observations concern sparse expansion under custom image splits. The original-split VTAB experiment instead adds 8.6 modules and exposes a difference between macro and example-weighted conclusions. We do not combine these protocols into one equivalence claim or infer that SEMA's published expansion benefits are invalid. The frozen-feature SEMA port had much greater forgetting and even changed the sign of pool deletion. Equal encoder identity alone did not make its intervention faithful to native SEMA.

A.13 Historical Gate Diagnostics and Exploratory Controls

In the earlier frozen-reuse protocol, prototype correction and routing both read the pre-adapter representation. Gate diagnostics found median competitive sets of three out of seventy text classes and one out of two hundred vision classes. Heads nevertheless overturned the prototype argmax on 27–32% of text inputs. Trained adapters moved their input by roughly 0.5–0.8% in relative norm, demonstrating that newly allocated adapters were not numerically inert, but not that reused adapters were trained. Prototype-temperature sweeps with small arm differences likewise conditioned on frozen reuse. The corrected retraining and fixed factorial controls in the main text provide the direct tests of that limitation.

The earlier class-dimension-head ledger contained thirteen testable AURA FP comparisons at a one-point equivalence margin. Eight established equivalence and five were inconclusive. These counts are not the count for the corrected cosine-head suite. A historical CODA-Prompt-Flat text comparison gave full-minus-no-pool FP of 0.9572 pp, with 90% CI $[0.3645, 1.5499]$ and 95% CI $[0.2258, 1.6887]$. Its mean below one point did not establish a one-point upper bound. Native CODA now supplies a materially different capacity result. The feature-space prompt surrogate cannot substitute for its original prompt mechanism.

Historical heterogeneous-vision, encoder-quality, semantic-stream, task-order, and training-budget comparisons remain available as exploratory artifacts. They jointly changed domain difficulty, source labels, sample composition, or optimization. In particular, disjoint versus repeated text label meanings did not hold dataset content constant, and a representation-quality ladder changed pretraining as well as separability. Their failure to support simple monotonic explanations did not eliminate heterogeneity, representation geometry, or semantic overlap as possible factors. The corrected paper makes no causal exclusion from those exploratory comparisons.

A.14 Allocation Properties

These elementary properties describe the centroid allocation rule, not learning performance.

Theorem 1 (Determinism conditional on arithmetic)**.** *For a fixed ordered input sequence, nonzero class centroids, fixed threshold and tie-breaking rule, and deterministic centroid/cosine arithmetic, adapter allocation is deterministic.*

Proof. The first task initializes one centroid set. Given a fixed current state, the next task's centroids, maximum similarities, and threshold decision are deterministic, as is the updated state. Induction completes the argument. Floating-point reductions must have fixed order for this implementation-level conclusion. Exact arithmetic is a sufficient alternative. □

Theorem 2 (Order dependence can occur)**.** *There exist tasks and a threshold for which permuting arrival order changes the final adapter count.*

Proof. Take three single-class unit centroids in $\mathbb{R}^2$ at angles 0, $\pi/4$, and $\pi/2$, and threshold $\tau = 1/2$. Adjacent similarities exceed τ and the outer pair's similarity is zero. Processing the middle centroid first lets both outer tasks join one adapter. Processing the outer centroids first creates two adapters. The middle task then joins one using the fixed tie-break. The final counts are one and two. This existential example makes no assertion for every threshold, including the endpoints. □

A.15 Dynamic Adapter Centroid Expansion Algorithm

Algorithm 1 gives the full procedure summarized in Appendix A.9.

Algorithm 1 Dynamic Adapter Centroid Expansion

Require: Task dataset $\mathcal{D}_k$, current adapter count N_a, adapter centroid sets $\{\mathcal{C}_1^{(A)}, \ldots, \mathcal{C}_{N_a}^{(A)}\}$, expansion threshold τ_{expand} and nonzero class centroids
Ensure: Assigned adapter index m^*, updated N_a, updated adapter centroids

// Step 1: per-class mean embedding for the new task
$\mathcal{C}_k \leftarrow \{\bar{\mathbf{x}}_{k,c}\}_{c \in \text{classes}(k)}$
$max_sim \leftarrow -\infty, \quad m^* \leftarrow 0$
// Step 2: find the most similar existing adapter
for $i = 1 \ldots N_a$ **do**
  $sim \leftarrow \max_{\mathbf{c}_{new} \in \mathcal{C}_k,\, \mathbf{c}_{old} \in \mathcal{C}_i^{(A)}} \text{CosineSim}(\mathbf{c}_{new}, \mathbf{c}_{old})$
  if $sim > max_sim$ **then**
    $max_sim \leftarrow sim, \quad m^* \leftarrow i$
  end if
end for
// Step 3: expand or reuse
if $N_a = 0$ or $max_sim < \tau_{expand}$ **then**
  $N_a \leftarrow N_a + 1$
  $\mathcal{C}_{N_a}^{(A)} \leftarrow \mathcal{C}_k$ *// new adapter A_{N_a}*
  $m^* \leftarrow N_a$
else
  $\mathcal{C}_{m^*}^{(A)} \leftarrow \mathcal{C}_{m^*}^{(A)} \cup \mathcal{C}_k$ *// reuse A_{m^*}*
end if
return $m^*, N_a, \{\mathcal{C}_1^{(A)}, \ldots, \mathcal{C}_{N_a}^{(A)}\}$

A.16 Distinct Joint-removal Controls

The historical name "floor" covered two different interventions. The *no-pool/no-prototype* control removes adapter prediction and prototype correction together. The *gate-floor* keeps adapters but removes their input-channel gate and the prototype logit correction. They must not be combined or treated as an individual prototype-gate ablation. Two historical gate-floor implementations gave

Table 30: Explicit component-removal definitions. The pool includes its internal channel gate when present unless the row says otherwise.

Condition	Pool	Channel gate	Prototype gate
Full	yes	yes	yes
No-pool	no	not applicable	yes
No-pool/no-prototype	no	not applicable	no
Channel gate removed	yes	no	yes
Prototype gate removed	yes	yes	no
Gate-floor	yes	no	no

AP 29.36% and 29.32%, a difference of 0.04 pp. This approximate agreement does not validate

unrelated pool interventions. The corrected fixed-configuration pool-by-prototype factorial now separates these effects (Table 3). Its four cells do not use the historical gate-floor as a substitute for pool bypass.

A.17 Baseline Adaptation Summary

Table 31 maps all nine historical `-Flat` adaptations to their native mechanisms (Appendix A.8). Published headline accuracies use different datasets, pretrained encoders, splits, and metrics, so we compare mechanisms here. Section 2 discusses the closest component and placement studies.

Table 31: Original mechanism vs. `-Flat` adaptation, all 9 baselines.

Model	**Original mechanism**	**`-Flat` adaptation**
L2P-Flat	Prepends a subset of a shared prompt pool as input tokens to a ViT sequence, selected by key-query similarity.	Averages the selected prompts into a single bias vector added directly to the frozen embedding, then a shared MLP projection.
DualPrompt-Flat	Injects a shared General-Prompt and task-specific Expert-Prompts into specific ViT layers.	Applies the General-Prompt and a softmax-weighted mixture of Expert-Prompts as MLPs directly on the embedding, no per-layer injection.
AdapterFusion-Flat	Fuses task-specific bottleneck adapters inside BERT via attention, trained on each target task's data.	Fuses one downstream adapter per task at the embedding level. Corrected fusion uses only current-task data. The earlier pooled-data defect is documented in Appendix A.10.
SEMA-Flat	Trains a per-layer autoencoder (last 3 ViT layers) whose reconstruction-error z-score triggers per-layer adapter expansion.	A single autoencoder (128-d bottleneck) trained continuously during each task's steps, triggering expansion for one adapter pool at the embedding level.
Houlsby-Flat	Two bottleneck adapters per transformer block and task: after the attention output projection and the feed-forward sublayer.	One `SimpleAdapter` per task applied once to the frozen embedding. Every adapter/head is evaluated and all seen-class logits are concatenated, with no inference task ID ($N_a{=}K$).
AdapterCL-Flat	One adapter per dialogue domain inside a seq2seq model. Entropy-based routing selects the lowest-entropy branch at inference.	One `SimpleAdapter` per task at the embedding level, with identical entropy-gated branch selection preserved.
O-LoRA-Flat	Low-rank $(\mathbf{A}, \mathbf{B})$ updates to attention weight matrices inside transformer layers, with an orthogonality penalty between successive tasks' subspaces.	The same low-rank update and orthogonality penalty applied to an external classification layer on the frozen embedding, not inside any attention matrix.
ConPET-Flat	A learned selector routes inputs to task-specific PET modules across layers. The source variant uses rehearsal.	A selector routes to downstream task adapters. The adaptation omits rehearsal by design. Fixed embeddings do not themselves preclude feature replay.
CODA-Prompt-Flat	Decomposed attention-weighted mixture over learned prompt components injected into ViT attention layers, with an orthogonality penalty between frozen and trainable components.	Prompt components, attention vectors, and the same orthogonality penalty applied as feature-space bias vectors mixed once at the embedding level.