DISSECTING PEFT-BASED CONTINUAL LEARNING: FROM EXPERT TRAINING TO INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Parameter-efficient fine-tuning (PEFT) has become a common strategy for class-incremental learning, with lightweight task-specific experts trained on a frozen pretrained backbone. However, existing methods often change several parts of the system simultaneously, making it difficult to determine whether reported gains arise from expert training or from choices such as the backbone, classification head, routing signal, and inference rule. We address this problem through a controlled component analysis of PEFT-based class-incremental learning. Holding expert training fixed, we vary these components across four benchmarks and two task granularities. Strong frozen-feature methods such as RanPAC provide a demanding reference point, yet a mechanism-free PEFT learner with a regularized least-squares readout and Mahalanobis-weighted expert ensembling achieves the highest mean across the eight settings. The value of an inference rule nevertheless depends on the head with which it is paired. Entropy remains a useful and simpler routing signal, particularly for probability ensembling, while Mahalanobis routing is more reliable when predictive confidence is poorly calibrated. Routing stress tests further show that retrieving the correct expert is not sufficient when unrelated experts dilute its prediction. Applying the same readout and combination choices to experts learned by established methods improves 19 of 24 settings and substantially changes their relative ranking, although the gains remain representation and dataset dependent. These findings show that the inference system surrounding expert learning is a major source of performance variation and should be evaluated separately from the learning mechanism itself.

1 INTRODUCTION

Most recent class-incremental learning (CIL) methods no longer update an entire network. They freeze a pretrained backbone and learn lightweight task-specific experts, such as prompts, adapters, or low-rank (LoRA) updates, one task at a time (Coleman et al., 2026). This parameter-efficient fine-tuning (PEFT) recipe is cheap, protects earlier knowledge, and has become the default starting point for new work, which advances through a steady series of refinements to how experts are trained: better prompt retrieval, hierarchical decompositions, orthogonality constraints, expandable subspaces, and shared or merged LoRA updates (Wang et al., 2022b;a; Smith et al., 2023; Wang et al., 2023; Liang & Li, 2024; Zhou et al., 2024b; Wu et al., 2025; He et al., 2025; Testa et al., 2025). Each new method typically reports an improvement of a few points over its predecessors.

A PEFT-based learner, however, is more than its expert-training mechanism. To predict without task identity, every method must also commit to a *pretrained backbone*, a *classification head* that maps expert features to scores over all classes seen so far, a *routing signal* that estimates which experts are relevant to a test image, and a *combiner* that turns the relevant experts into one prediction. These choices are usually treated as implementation details, yet new methods often change several of them at once and are evaluated only as complete systems. When a method gains two points over its predecessor, it is rarely clear whether the gain comes from how its experts were learned or from the machinery that turns them into predictions.

This paper measures that machinery directly. We deliberately remove every continual-learning mechanism from expert training: each task receives one LoRA expert, trained on its own data and then frozen, with no separation losses, replay, learned routers, or test-time updates. On top

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.

of these fixed experts we vary only the backbone, head, routing signal, and combiner, across four benchmarks and two stream lengths, and then transfer the resulting choices to experts trained by established methods.

The results are uncomfortable for current evaluation practice. With the experts held fixed, the choice of head and inference rule alone moves final accuracy by 19 to 38 points depending on the benchmark (Figure 2), a range comparable to or larger than that spanned by the twelve established methods on three of four benchmarks. Strong frozen-feature baselines raise the reference level substantially: RanPAC averages 76.9% across the eight settings. Nevertheless, the strongest controlled configuration, a regularized least-squares head with Mahalanobis-weighted expert ensembling, averages 78.1%, leads in two settings, and lies within one point of the best result in four more (Table 5). Conversely, replacing only the head and inference rule of three established expert-based methods improves 19 of 24 settings by 4.2 points on average and up to 15.5 points. On 20-task CUB-200, for example, InfLoRA trails native EASE by 14.9 points, but reaches within 0.5 points after replacing its readout (Table 6).

We do not claim that expert-training mechanisms are ineffective. Our claim is narrower and, we believe, more consequential: under current reporting practice their contribution cannot be separated from that of the surrounding inference system, and that system is at least as influential. Concretely, we contribute:

- **A controlled framework** that evaluates backbone, head, routing signal, and combiner on a fixed set of independently trained experts (Section 3).
- **A factorial analysis** showing that head and inference rule must be chosen jointly, that a least-squares head with probability-space ensembling is consistently competitive, and that Mahalanobis routing is preferable when predictions are unreliable (Section 5).
- **Transfer evidence** that these choices substantially change the accuracy of EASE, InfLoRA, and HiDe-LoRA without altering their experts (Section 5.5).
- **Cheap baselines and reporting recommendations** (Sections 5.6 and 6).

2 Related Work

PEFT-based continual learning. Pretrained models have shifted CIL toward adapting a largely frozen representation (Wu et al., 2022; Coleman et al., 2026). L2P, DualPrompt, and CODA-Prompt learn task-conditioned prompts and a mechanism to retrieve or compose them (Wang et al., 2022b;a; Smith et al., 2023); HiDe-Prompt decomposes task and class information, and Consistent Prompting identifies prompt-selection and classifier inconsistencies as major error sources (Wang et al., 2023; Gao et al., 2024). Adapter- and LoRA-based methods follow the same template: EASE expands task-specific subspaces, InfLoRA constrains low-rank updates to reduce interference, and HiDe-LoRA extends hierarchical decomposition to LoRA experts (Zhou et al., 2024b; Liang & Li, 2024; Wang et al., 2023); SD-LoRA, CL-LoRA, and GLAM address the scalability, sharing, or merging of accumulated updates (Wu et al., 2025; He et al., 2025; Testa et al., 2025). Each introduces a new training mechanism *together with* particular choices of classifier and inference, which is the bundling we disentangle.

Simple baselines and reality checks. Pretrained representations do much of the work in CIL: SimpleCIL classifies frozen features with prototypes, APER combines frozen and adapted representations, RanPAC uses random projections with an analytical ridge classifier, FeCAM models class-conditional distributions, and SLCA aligns the classifier (Zhou et al., 2024a; McDonnell et al., 2023; Goswami et al., 2023; Zhang et al., 2023); Feillet et al. (2025) question how much exemplar-free methods gain over their pretrained starting point. We extend this line of inquiry from frozen features to *expert-based* learners, where the readout also interacts with routing and composition, and show that the effect persists on experts learned by published methods.

Expert composition and model merging. Expert-based methods combine experts through retrieval, hard selection, output ensembling, feature expansion, or parameter merging. Model merging offers many operators for combining adapted weights, including averaging, task arithmetic, sign-consistent merging, and sparsification (Wortsman et al., 2022; Ilharco et al., 2023; Yadav et al., 2023; Yu et al., 2024; Davari & Belilovsky, 2024), but they are usually studied with a fixed classifier

and outside the task-agnostic setting, where a merged or wrongly selected expert may yield features the shared head cannot read. We therefore evaluate them jointly with the head.

3 Anatomy of a PEFT-Based Continual Learner

We describe PEFT-based CIL through four components surrounding expert training: the *backbone*, the shared *classification head*, and an *inference policy* made of a *routing signal* and a *combiner*. Experts are a fixed substrate, so any difference between configurations is attributable to these components.

3.1 Problem Definition and Fixed Expert Protocol

At task $t \in \{1, \dots, T\}$, the learner receives $\mathcal{D}_t = \{(x_i, y_i)\}_{i=1}^{n_t}$ with labels in $\mathcal{C}_t$, where class sets are disjoint, $\mathcal{C}_t \cap \mathcal{C}_{t'} = \varnothing$ for $t \neq t'$, and $\mathcal{C}_{1:t} = \bigcup_{j \leq t} \mathcal{C}_j$. Past raw examples are unavailable and task identity is unknown at test time, so predictions range over all of $\mathcal{C}_{1:t}$.

Let $f(\cdot\,; \theta_0)$ be a pretrained backbone with frozen parameters θ_0. For each task we train a LoRA update ΔW_t on $\mathcal{D}_t$ alone, using a temporary task-local classifier that is discarded afterwards, and then freeze ΔW_t as the task expert. For a test sample x, expert j yields a representation and class scores

$$h_j(x) = f(x; \theta_0 + \Delta W_j), \qquad s_j(x) = g\big(h_j(x)\big), \quad j \in \{1, \dots, t\}, \tag{1}$$

where g is the shared head. Experts never modify one another, so any gap between oracle and task-agnostic accuracy arises from the head, routing, or composition rather than from forgetting inside an expert. This protocol is intentionally naive: it is not a new method, but the simplest learner against which the surrounding components can be measured.

3.2 Backbone and Classification Head

Backbone. A frozen backbone cannot forget, but its architecture, pretraining data, and resolution still determine separability and robustness to shift. We vary it in one comparison and fix it elsewhere.

Head. The shared head g maps the features of any expert to scores over all observed classes, so it sets both class separation and cross-task calibration. After task t, it is updated with features of $\mathcal{D}_t$ extracted through the corresponding expert ΔW_t. Table 1 lists the five heads we evaluate. NCM and the analytical heads store class statistics rather than examples; `linear` stores features of all classes and serves as a latent-replay comparator, while `linmask` is trained on the current task only, like the masked classifiers common in prompt-based methods. For `ridge` and `lstsq`, per-task sufficient statistics are accumulated and the classifier is obtained in closed form,

$$\Theta = \left(\sum_t H_t^\top H_t + \lambda I \right)^{-1} \left(\sum_t H_t^\top Y_t \right), \tag{2}$$

where Y_t contains one-hot targets and H_t contains the backbone features (`lstsq`) or fixed random projections of them (`ridge`, following RanPAC). Since the rows of each H_t never change, this equals fitting on the union of all tasks without storing features.

3.3 Inference Policy: Routing Signal and Combiner

Routing signal. The router assigns each expert a relevance score $r_j(x)$, lower meaning more relevant. We compare the two signals in Table 2: predictive *entropy*, which measures confidence through the full expert–head pipeline, and *Mahalanobis* distance to each expert's training distribution, which does not depend on the head's predictions. Given either signal, the router keeps the k most relevant experts and normalizes their scores:

$$\mathcal{I}_k(x) = \arg\operatorname*{topk}_j \big[-r_j(x)\big], \qquad w_j(x) = \frac{\exp(-r_j(x)/\tau)}{\sum_{\ell \in \mathcal{I}_k(x)} \exp(-r_\ell(x)/\tau)}, \tag{3}$$

where $\tau > 0$ is a temperature; small values concentrate weight on the top-ranked expert, large values approach a uniform combination over $\mathcal{I}_k(x)$.

Table 1: Classification heads evaluated in the component analysis. For `ridge` and `lstsq`, the stored statistics are accumulated over random-projected and backbone features respectively, as noted in the prediction rule.

Head	Stored information	Prediction rule
NCM	Running class centroids	Assigns the class with the closest centroid in feature space.
`ridge`	Gram and cross-covariance	Applies regularized least-squares classification after a fixed random projection, following RanPAC.
`lstsq`	Gram and cross-covariance	Applies regularized least-squares classification directly in the backbone feature space.
`linear`	Stored features, all classes	Trains a linear classifier using SGD and cross-entropy; serves as a latent-replay comparator.
`linmask`	Classifier parameters only	Trains on the current task using cross-entropy restricted to its class outputs, without replay.

Table 2: Routing signals used to estimate expert relevance. Here $p_j = \mathrm{softmax}(s_j)$ and h_j is the feature produced by expert j. Lower scores indicate higher relevance.

Signal	Score	Interpretation
Entropy	$r_j^{\mathrm{ent}}(x) = -\sum_c p_{j,c}(x) \log p_{j,c}(x)$	Confidence through the full expert–head pipeline; a confidently wrong expert still ranks first.
Mahalanobis	$r_j^{\mathrm{maha}}(x) = \min_{c \in \mathcal{C}_j} (h_j - \mu_{j,c})^\top \Lambda_j (h_j - \mu_{j,c})$	Distance to expert j's training distribution, from stored class means and a regularized precision matrix.

Combiner. The combiner turns routed experts into a prediction (Table 3); oracle inference, which uses the correct expert, isolates expert and head quality from routing error. In the grids we abbreviate the task-agnostic combiners as `select`, `ens`, `ens_p`, `vote`, and `merge`, all driven by entropy, and denote selection and ensembling driven by the Mahalanobis signal as `maha_s` and `maha_e`.

4 Experimental Setup

Benchmarks and protocol. We evaluate on CIFAR-100, CUB-200, ImageNet-R, and ImageNet-A, split into 10- and 20-task streams with equal-sized, disjoint class increments. Task identity is withheld at inference; all configurations in a run share class order and split, and results average three runs. Unless stated otherwise, no previous-task images, separation objectives, learned routers, or test-time parameter updates are used.

Component analyses. Table 4 summarizes what each analysis varies and holds fixed. The primary grid crosses the five heads of Table 1 with the seven inference rules above on a fixed ImageNet-21K ViT-B/16 backbone with identical experts. At $T = 10$, a second grid crosses four heads (`lstsq`, `ridge`, Gaussian, `NCM`) with eight merging operators: linear, TIES, DARE, SLERP, task arithmetic, DARE-TIES, Breadcrumbs, and concatenation. Unless stated otherwise, routing uses $k = 3$ and $\tau = 1$, analytical heads use $\lambda = 100$, RanPAC uses $1{,}000$ random features, and covariance estimates use shrinkage 0.5. Further backbone, LoRA, optimization, and merging details are given in the appendix.

Routing stress test and transfer. The routing stress test fixes the head and combiner and varies only the candidate expert set: the oracle expert alone, the oracle expert plus $k-1$ random experts (oracle-containing), entropy top-k, Mahalanobis top-k, all experts, random-k, and worst-k. For transfer, we keep the representation learned by each method and replace only post-training decisions. For EASE, a shared `lstsq` head (each class fitted through its own adapter) replaces the prototype classifier, with probability ensembling over all adapters. For InfLoRA, a single `lstsq` head replaces the per-task classifier pool on the merged backbone, so no combiner is needed. For HiDe-LoRA, a shared `lstsq` head with probability ensembling replaces task-identity inference and

Table 3: Inference strategies evaluated in this work. Here $j^\star = \arg\min_j r_j(x)$, $p_j = \mathrm{softmax}(s_j)$, $\theta(x) = \theta_0 + \sum_j w_j(x)\Delta W_j$, w^* are mixing weights optimized at test time, and $j \in \mathcal{I}_k(x)$ for top-k methods.

Strategy	**Prediction**	**Property measured**
Oracle	$\arg\max_c s_{t(x),c}(x)$	Expert and head quality, free of routing error.
Selection	$\arg\max_c s_{j^\star,c}(x)$	Top-1 routing quality; one expert carries the prediction.
Logit ensemble	$\arg\max_c \sum_j w_j(x) s_{j,c}(x)$	Combines raw scores, so it remains sensitive to logit scale.
Prob. ensemble	$\arg\max_c \sum_j w_j(x) p_{j,c}(x)$	Combines normalized scores, removing that scale sensitivity.
Voting	$\mathrm{mode}_j[\arg\max_c s_{j,c}(x)]$	Uses decisions rather than score magnitudes; softmax mass breaks ties.
Param. merging	$\arg\max_c g_c(f(x;\theta(x)))$	Combines in parameter space before feature extraction, which is not equivalent to output averaging.

Table 4: Summary of the controlled component analyses.

Analysis	**Varied component**	**Fixed configuration and scope**
Backbone	ViT, CLIP, DINOv2	`lstsq` and `maha_e`; all four datasets at 224 pixels
Head–rule	Five heads $\times$ seven inference rules	Fixed backbone and experts; all four datasets
Head–merging	Four heads $\times$ eight merging rules	Fixed backbone and experts; all four datasets at $T = 10$
Routing	Candidate expert set	`lstsq`, probability ensemble, $k = 3$; ImageNet-R/A
Transfer	Head and inference rule	Experts learned by EASE, InfLoRA, or HiDe-LoRA
Efficiency	Method and stream length	Same hardware and evaluation scope; ImageNet-R

the Gaussian-aligned classifier. We exclude GLAM from this controlled transfer analysis because fitting a replacement head on its final merged model requires revisiting all previous-task training images and therefore changes the data-access protocol.

Metrics. We report final class-incremental accuracy over all classes and total training and inference time. For the balanced head–rule grid, we perform a fixed-effects variance decomposition over dataset, stream length, head, inference rule, and their interaction, treating seeds as replicates and reporting $\eta^2 = \mathrm{SS}_{\mathrm{term}}/\mathrm{SS}_{\mathrm{total}}$; cross-dataset ranking consistency is measured by mean pairwise Spearman correlation.

5 Results

5.1 A Mechanism-Free Learner Is Competitive with Recent Methods

Table 5 compares five configurations of our mechanism-free learner with twelve established methods under the same backbone and splits.

The simple learner remains competitive against the stronger reference set. The strongest variant, `lstsq+maha_e`, is best in two of eight settings and lies within one point of the best result in four more. It is effectively tied with RanPAC on CIFAR-100 when averaged over the two stream lengths, trails RanPAC on CUB-200 by 3.0–4.3 points, and performs better on both ImageNet-R and ImageNet-A settings. Its largest gain over the strongest established result in a column is 3.32 points on 10-task ImageNet-A.

The comparison is strongly benchmark dependent. RanPAC leads on CIFAR-100 and CUB-200, CL-LoRA and SD-LoRA are strongest on ImageNet-R, and GLAM is strongest among established methods on 20-task ImageNet-A. In contrast, `lstsq+maha_e` has a maximum deficit of 4.33 points from the best established result in any column. Averaged over all eight settings, it reaches 78.12%, compared with 76.88% for RanPAC, the strongest established method on average. The simpler entropy-based `lstsq+ens_p` reaches 76.62%, showing that the conclusion does not depend on Mahalanobis routing alone.

Table 5: Final class-incremental accuracy (%, mean ± standard deviation over three runs). All methods use the same ImageNet-21K ViT-B/16 backbone and identical data splits. The overall average is the unweighted mean of the eight setting means (standard deviations are not pooled). Bold indicates the highest mean in each column, including the overall average.

	CIFAR-100		CUB-200		ImageNet-R		ImageNet-A		Overall
Method	$T = 10$	$T = 20$	$T = 10$	$T = 20$	$T = 10$	$T = 20$	$T = 10$	$T = 20$	Avg. (%)
L2P	83.38±0.59	80.10±0.84	68.24±0.48	63.93±1.86	71.29±0.35	68.11±1.07	42.95±1.38	37.73±0.34	64.47
DualPrompt	84.39±0.56	79.76±0.66	74.37±0.71	70.70±1.30	65.82±0.20	61.58±0.37	40.34±0.47	35.53±0.72	64.06
EASE	88.04±0.56	85.68±0.43	86.40±0.11	86.68±0.08	75.94±0.36	73.96±0.20	56.10±0.88	45.48±1.76	74.79
InfLoRA	86.09±1.06	82.28±0.81	80.98±0.63	71.76±0.81	74.47±0.41	70.76±0.83	48.59±0.38	41.83±0.91	69.60
HiDe-LoRA	87.96±0.80	85.60±0.33	85.86±0.15	86.21±0.34	75.04±0.36	73.71±0.20	46.05±0.17	44.87±0.57	73.16
SD-LoRA	85.33±0.44	82.85±0.40	70.11±0.12	63.44±2.06	76.17±0.70	**76.76±0.28**	56.70±1.39	52.89±1.98	70.53
CL-LoRA	86.43±0.11	84.84±0.32	78.97±0.62	68.47±1.31	**79.58±0.34**	76.69±0.34	57.94±1.04	50.13±0.70	72.88
SimpleCIL	81.28±0.00	81.28±0.00	86.78±0.00	86.78±0.00	54.44±0.00	54.44±0.00	48.92±0.00	48.92±0.00	67.86
APER-Adapter	87.16±0.55	84.57±1.09	86.81±0.05	86.82±0.02	64.69±1.46	56.31±0.84	49.15±0.11	49.01±0.06	70.57
RanPAC	**91.47±0.20**	89.98±0.47	**89.01±0.23**	**88.22±0.51**	75.86±0.92	71.02±0.38	54.92±0.42	54.56±0.19	76.88
FeCAM	88.70±0.40	86.23±1.18	88.27±0.02	88.07±0.01	69.16±1.31	58.02±1.66	46.32±0.31	45.71±0.08	71.31
GLAM	86.77±0.22	87.43±0.50	77.18±0.82	75.67±1.39	75.02±0.62	73.78±0.18	56.81±0.09	**57.99±0.02**	73.83
Ours: `lstsq+maha_e`	91.39±0.09	**90.07±0.45**	86.04±0.35	83.89±0.23	78.66±0.08	76.22±0.26	**61.26±0.55**	57.44±0.24	**78.12**
Ours: `lstsq+ens_p`	90.74±0.14	89.28±0.27	84.43±0.16	81.67±0.26	77.32±0.18	74.55±0.15	59.40±0.28	55.60±0.52	76.62
Ours: `ridge+maha_e`	90.90±0.13	89.43±0.37	85.64±0.32	83.15±0.16	77.82±0.48	75.21±0.24	57.29±0.49	53.89±0.43	76.67
Ours: `ridge+ens_p`	89.81±0.12	87.82±0.11	84.35±0.23	81.23±0.06	75.52±0.16	72.53±0.37	54.03±0.47	50.97±0.73	74.53
Ours: `lstsq+ens`	90.68±0.20	89.00±0.28	84.37±0.16	81.58±0.35	77.28±0.19	74.53±0.22	59.13±0.21	55.64±0.53	76.53

Differences between our configurations follow the components, not the experts, since all five rows share the same experts. Replacing `ridge` with `lstsq` under probability ensembling adds up to 5.4 points (ImageNet-A), and replacing entropy with Mahalanobis weighting adds a further 0.7 points on CIFAR-100 up to 2.2 points on CUB-200, with the largest gains where predictive confidence is least reliable (Section 5.4). With identical experts, these differences match the gaps that separate published methods; the rest of this section explains where they come from.

5.2 The Backbone Is a Hidden Hyperparameter

With everything else fixed (Figure 1), DINOv2 improves over the ImageNet-21K ViT by 6.2 points on ImageNet-R and 12.3 points on ImageNet-A at $T = 10$, with similarly large gains at $T = 20$. On CIFAR-100 and CUB-200, however, ViT is slightly stronger than DINOv2, while CLIP trails ViT substantially. The ordering is therefore benchmark dependent, and the largest backbone effect exceeds most gaps between published methods. Cross-paper comparisons with different checkpoints conflate this effect with the method's contribution, which is why all rows in Table 5 use the same backbone.

5.3 Heads and Inference Rules Must Be Chosen Jointly

Crossing heads with inference rules on identical experts (Figure 2), accuracy varies from the weakest to the strongest cell by 27.7 points on CIFAR-100, 38.4 on CUB-200, 19.4 on ImageNet-R, and 22.7 on ImageNet-A. This exceeds the range spanned by the twelve established methods at $T = 10$ on three of the four benchmarks; ImageNet-R is the exception because the frozen-feature baselines themselves span a wide range.

The two analytical heads, `lstsq` and `ridge`, are strong under almost every rule, whereas NCM and `linmask` are highly sensitive to how experts are selected and combined. NCM with entropy-based selection reaches only 68.3% on CIFAR-100 and 41.4% on ImageNet-A; the same head with the same experts reaches 89.9% and 58.1% under Mahalanobis selection. The best rule also depends on the head: Mahalanobis ensembling is best for `lstsq`, `ridge`, and `linear`, while `linmask`, and NCM on three of four benchmarks, favour Mahalanobis selection. A combiner can look strong or weak depending on the head it is tested with.

The variance decomposition confirms this. After removing dataset and stream-length effects, the head accounts for 68.8% of the remaining variation. The inference-rule main effect and head–rule interaction account for 6.7% and 5.9%, respectively, so the interaction is approximately as important as the rule in isolation. Rule rankings are stable across datasets (mean pairwise Spearman correlation 0.94), but the best absolute result still depends on the head with which a rule is paired. An inference rule should therefore not be proposed or ablated with a single, arbitrary head.

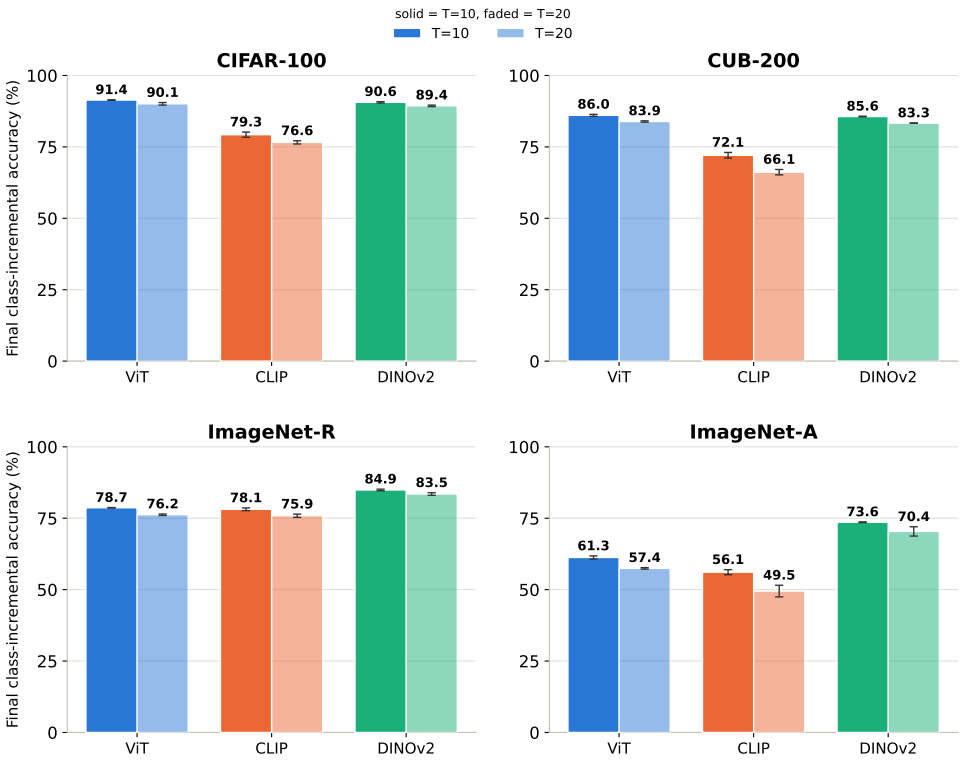


Figure 1: Backbone comparison on all four benchmarks. We fix the `lstsq` head and `maha_e` inference rule and evaluate ViT, CLIP, and DINOv2 at 224-pixel resolution. Solid and faded bars report $T = 10$ and $T = 20$, respectively; values are means over three runs and error bars show standard deviation.

Parameter-space composition provides a complementary result (Appendix D). With a fixed `lstsq` head at $T = 10$, direct linear merging is within 0.6 points of RanPAC on CIFAR-100 and CUB-200, and exceeds it by 1.8 points on ImageNet-R and 5.3 points on ImageNet-A. Several simple merging operators occupy a similar high-performing range, whereas concatenation fails sharply. This shows that task-specific updates can remain competitive when collapsed into one model, but their usefulness still depends on a compatible readout head.

5.4 Routing Is Informative, but Retrieval Is Not the Bottleneck

Figure 3b fixes the `lstsq` head and probability ensemble and varies only which experts are combined. Informed routing clearly beats uninformed selection: Mahalanobis top-k exceeds random-k by 3.0–4.0 points in every setting, entropy top-k by 1.1–3.6 points, and random-k and worst-k are the weakest sets throughout. Entropy thus carries useful information about which expert is relevant. Yet it is prediction-based, so a confidently wrong expert still ranks first. On 20-task ImageNet-A, where distribution shift makes predictions least reliable, entropy top-k falls 2.1 points below simply weighting all experts, whereas Mahalanobis top-k, which does not depend on the head's predictions, matches it. This is consistent with the larger gains of `maha_e` on CUB-200 and ImageNet-A in Table 5.

The larger gap lies elsewhere. Combining the correct expert with $k-1$ random experts remains 5.8–8.4 points below using the correct expert alone, and this gap grows with the number of tasks. The relevant knowledge is often retrieved but diluted by incompatible experts, so routing and combination form a coupled problem. The oracle expert is 10–17 points above every task-agnostic candidate set: much of the remaining headroom lies at inference, not in expert training.

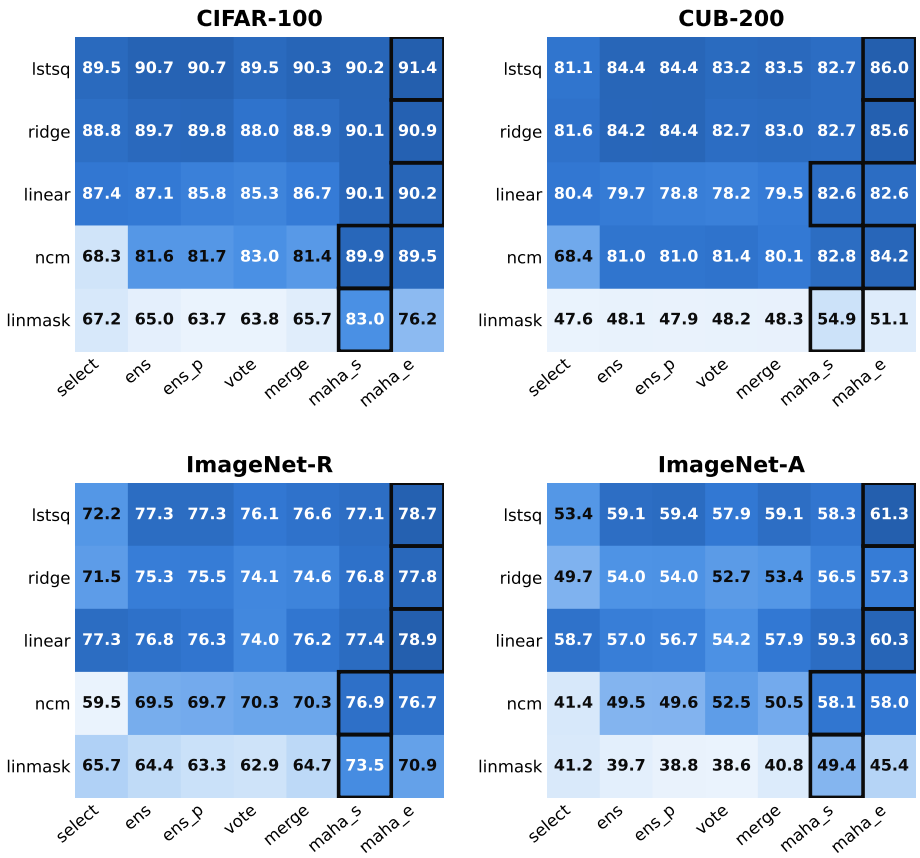


Figure 2: Interaction between the classification head and inference rule for 10 tasks over 3 separate runs. Outlined cells indicate the strongest inference rule for each head.

Table 6: Accuracy change after replacing only the head and inference rule. Native accuracies are reported in Table 5. Positive values indicate improvement; expert training is unchanged.

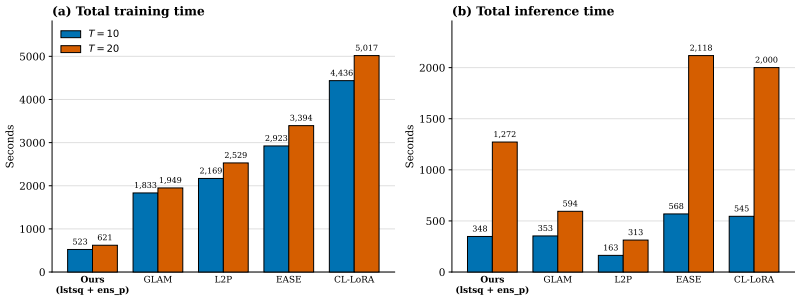
	CIFAR-100		CUB-200		ImageNet-R		ImageNet-A	
Modified method	$T = 10$	$T = 20$	$T = 10$	$T = 20$	$T = 10$	$T = 20$	$T = 10$	$T = 20$
EASE + `lstsq+ens_p`	+2.48	+3.88	+0.70	+0.50	−1.48	−2.24	+2.62	+8.70
InfLoRA + `lstsq`	+3.46	+4.54	+5.78	+14.47	+1.87	+5.23	+9.73	+15.52
HiDe-LoRA + `lstsq+ens_p`	+2.97	+4.14	−0.28	−1.59	−0.29	+0.16	+9.46	+10.81

5.5 Swapping the Readout Reorders Published Methods

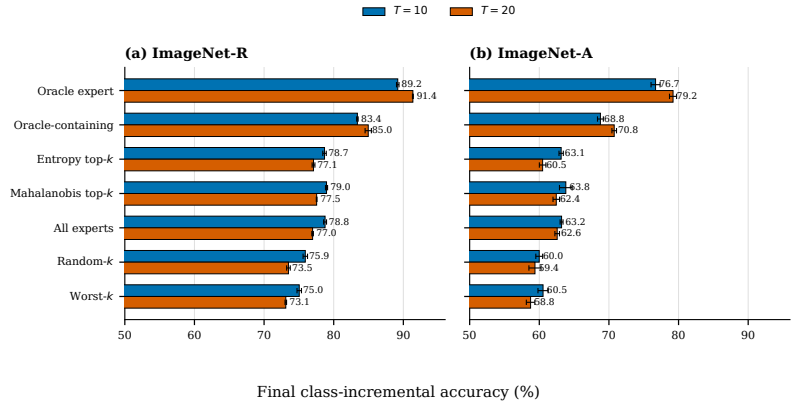
Table 6 replaces only the head and inference rule of three methods, keeping their experts as learned. The replacement improves 19 of the 24 settings, with a mean change of $+4.21$ points. InfLoRA improves everywhere, by 1.9–15.5 points; EASE and HiDe-LoRA gain 2.6–10.8 points on ImageNet-A.

This reorders the leaderboard. With its native classifier, InfLoRA trails EASE by 14.9 points on 20-task CUB-200; after replacing its readout, InfLoRA reaches 86.23%, only 0.45 points below native EASE. On 20-task ImageNet-A it rises from 41.83% to 57.35%, above RanPAC and close to GLAM. Much of InfLoRA's apparent weakness is therefore due to its readout rather than its training mechanism.

The negative cases, EASE on ImageNet-R and HiDe-LoRA on CUB-200 and 10-task ImageNet-R, show that a stronger head is not a universal plug-in: its value depends on the method's representation, which is why the two should be reported separately.



(a) Total training and inference time.



(b) Routing stress test.

Figure 3: Efficiency and routing analyses. (a) Total training and inference time for $T = 10$ and $T = 20$; lower is better, and all methods are measured on the same hardware and over the same evaluation scope. (b) Routing stress test with the `lstsq` head and entropy-weighted probability ensemble fixed at $k = 3$, varying only the candidate expert set. Bars report mean final accuracy and error bars show standard deviation over three runs.

5.6 Efficiency

Our `lstsq+ens_p` baseline trains 3.1×–8.5× faster than the evaluated methods (Figure 3a), without replay, separation losses, or joint optimization. Inference evaluates every expert per image, so cost grows with task count: our method is faster than EASE and CL-LoRA at both lengths, about as fast as GLAM at $T = 10$, but slower than GLAM at $T = 20$ and L2P.

6 Conclusion: What Should Be Reported

With expert training fixed, backbone, head, and inference choices shift accuracy by as much as gaps between published methods. Frozen-feature baselines remain strong, especially RanPAC on CUB-200, yet our simple learner leads in mean accuracy across eight settings. Readout changes also reorder established methods. We recommend reporting inference components separately from expert training; comparing matched frozen-feature and mechanism-free baselines; reporting oracle-expert and task-agnostic accuracy; and testing new heads and combiners with multiple partners.

Limitations. Most analyses use ViT-B/16; merging covers only $T = 10$, routing only ImageNet-R/A, and transfer only three methods. Per-image evaluation of every expert scales inference with task count; scalable accurate routing remains open.

GENERATIVE AI DECLARATION

Generative AI tools were used to assist with language editing, document organization, LaTeX formatting, and the preparation of visualization code. They were not used to generate experimental data or determine the reported results. All methodological choices, implementations, analyses, citations, and conclusions were reviewed and verified by the authors, who take full responsibility for the content of the paper.

REFERENCES

Eric Nuertey Coleman, Luigi Quarantiello, Ziyue Liu, Qinwen Yang, Samrat Mukherjee, Julio Hurtado, and Vincenzo Lomonaco. Parameter-efficient continual fine-tuning: A survey. *Neurocomputing*, 701:134502, 2026. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2026.134502. URL https://www.sciencedirect.com/science/article/pii/S0925231226019004.

MohammadReza Davari and Eugene Belilovsky. Model breadcrumbs: Scaling multi-task model merging with sparse masks. In *European Conference on Computer Vision*, pp. 270–287, 2024.

Eva Feillet, Adrian Popescu, and Céline Hudelot. A reality check on pre-training for exemplar-free class-incremental learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 7614–7625, 2025.

Zhanxin Gao, Jun Cen, and Xiaobin Chang. Consistent prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28463–28473, 2024.

Dipam Goswami, Yuyang Liu, Bartłomiej Twardowski, and Joost van de Weijer. FeCAM: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Jiangpeng He, Zhihao Duan, and Fengqing Zhu. CL-LoRA: Continual low-rank adaptation for rehearsal-free class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

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

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*, 2024.

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

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Irene Testa, Luigi Quarantiello, Eric Nuertey Coleman, Samrat Mukherjee, Julio Hurtado, and Vincenzo Lomonaco. GLAM: Efficient continual learning at scale via grouped LoRA adapter merging. *arXiv preprint arXiv:2509.13211*, 2025.

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

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. DualPrompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, 2022a.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022b.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, 2022.

Tz-Ying Wu, Gurumurthy Swaminathan, Zhizhong Li, Avinash Ravichandran, Nuno Vasconcelos, Rahul Bhotika, and Stefano Soatto. Class-incremental learning with strong pre-trained models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9601–9610, 2022.

Yichen Wu, Hongming Piao, Long-Kai Huang, Renzhen Wang, Wanhua Li, Hanspeter Pfister, Deyu Meng, Kede Ma, and Ying Wei. SD-LoRA: Scalable decoupled low-rank adaptation for class-incremental learning. In *International Conference on Learning Representations*, 2025.

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

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

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.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23554–23564, 2024b.

A Implementation Details and Hyperparameters

A.1 Compute Environment

Component	Configuration
Cluster	Leonardo (CINECA), SLURM `boost_usr_prod` partition
GPU	One NVIDIA A100 64 GB per job
CPU and memory	8 CPU cores and 40 GB RAM per job
Python	3.11.7
PyTorch	2.5.1 with CUDA 12.1
`timm`	1.0.27
NumPy	2.4.4
Precision	bfloat16 for feature extraction; float32 for head fitting

All experiments use a single GPU. Pretrained weights are cached locally, and the Hugging Face Hub is accessed in offline mode during execution.

A.2 Backbones

The main experiments use an ImageNet-21K pretrained ViT-B/16 at 224×224 resolution. The matched-resolution backbone diagnostic uses the following models.

Backbone	Implementation identifier	Patch	Tokens
ViT-B/16	`vit_base_patch16_224`	16	196
CLIP ViT-B/16	`vit_base_patch16_clip_224.openai`	16	196
DINOv2 ViT-B/14	`vit_base_patch14_dinov2.lvd142m`	14	256

DINOv2 is evaluated at 224 pixels using positional-embedding interpolation. It retains its patch size of 14 and therefore processes 256 tokens. This comparison controls input resolution, but not architecture or pretraining. The backbone remains frozen throughout training.

A.3 Task-Specific LoRA Experts

Hyperparameter	Value
Rank r	8
Scaling parameter α	32
Injection sites	Attention and MLP projection layers
SVD retained energy	0.90
Maximum stored rank after compression	8

Each task starts from the same frozen backbone and learns an independent LoRA update. Experts are not stacked. After training, the dense update is compressed by truncated SVD for storage and then frozen.

A.4 Expert Training

Hyperparameter	Value
Optimizer	AdamW with zero weight decay
Learning rate	5×10^{-4}
Batch size	128
Epochs	5 for CIFAR-100; 15 for the remaining benchmarks
Objective	Cross-entropy with a temporary task-local classifier

The temporary classifier is discarded after each expert is trained.

A.5 CLASSIFICATION HEADS

Head	Definition	Key setting
`ncm`	Cosine similarity to running class centroids	Sufficient statistics
`gauss`	Class-conditional diagonal Gaussian	Shrinkage 0.5
`ridge`	Ridge regression on fixed random-ReLU features	1,000 features; $\lambda = 100$
`lstsq`	Regularized least squares in the original feature space	$\lambda = 100$
`linear`	SGD-trained linear classifier using stored features	Feature-replay comparator
`linmask`	SGD-trained linear classifier using only current-task features and outputs	No feature buffer

The analytical heads accumulate sufficient statistics from each task and do not retain training images. The `linear` comparator retains features from previous tasks, whereas `linmask` updates only the outputs associated with the current task. The Gaussian head is included in complementary analyses but not in the primary five-head factorial grid.

A.6 ROUTING AND EXPERT COMBINATION

Unless stated otherwise, the inference rules use the top $k = 3$ experts and temperature $\tau = 1$. The temperature controls the sharpness of the softmax over negative routing scores: smaller values concentrate weight on the highest-ranked expert, while larger values produce a more uniform combination.

Rule	Description
`select`	Select the expert with minimum predictive entropy.
`ens`	Combine top-k logits using entropy-derived weights.
`ens_p`	Combine top-k probabilities using entropy-derived weights.
`vote`	Take the majority prediction over all experts.
`merge`	Combine selected LoRA updates in parameter space.
`maha_s`	Select one expert using Mahalanobis distance.
`maha_e`	Combine top-k logits using Mahalanobis-derived weights.

The strongest controlled configuration in the main comparison uses `lstsq` with `maha_e`. The transfer experiments use `lstsq` with probability ensembling when multiple expert representations are present.

A.7 EVALUATION PROTOCOL

We evaluate 10-task and 20-task class-incremental streams on CIFAR-100, CUB-200, ImageNet-R, and ImageNet-A. Classes are disjoint across tasks, task identity is unavailable at inference, and predictions cover all observed classes. Task orders are sampled with seeds 0, 1, and 2 and shared by all compared methods. We report final class-incremental accuracy as mean $\pm$ standard deviation over the three runs. Under the default protocol, previous-task images are not stored or revisited. Experiments that retain features or revisit training images are identified explicitly.

The baseline comparison uses the same ImageNet-21K pretrained ViT-B/16, dataset splits, task orders, hardware, and evaluation scope. Baseline training settings follow their official implementations; the runtime comparison therefore measures each complete method rather than enforcing equal epochs.

A.8 TEMPERATURE AND NUMBER OF SELECTED EXPERTS

Table 7 varies τ and k for `lstsq` with `maha_e` on CIFAR-100 with $T = 10$, seed 0, and the ImageNet-21K pretrained ViT-B/16.

Table 7: Final class-incremental accuracy (%) for different values of k and τ.

k / τ	0.25	0.5	1	2	5	10
1	89.32	89.32	89.32	89.32	89.32	89.32
2	90.50	90.58	90.78	90.77	90.69	90.58
3	90.60	90.87	90.97	90.86	90.82	90.78
5	90.61	90.90	**91.00**	90.86	90.66	90.61
7	90.61	90.91	90.91	90.74	90.50	90.36
10	90.61	90.91	90.92	90.67	90.33	90.11

The paper uses $k = 3$ and $\tau = 1$, which reaches 90.97%, within 0.03 points of the best cell. For $k \geq 2$ and $\tau \in [0.5, 2]$, the spread is only 0.42 points. The larger full-grid spread is primarily caused by hard top-1 selection.

B Per-Seed Backbone Comparison at 224 Pixels

This diagnostic fixes the `lstsq` head and `maha_e` inference rule and evaluates all backbones at 224×224 resolution, matching the configuration in Figure 1. The figure reports the three-run means, while Table 8 provides the seed-0 values for reproducibility.

Table 8: Final class-incremental accuracy (%) in the matched-resolution backbone diagnostic.

Dataset	Tasks	ViT-B/16	CLIP ViT-B/16	DINOv2 ViT-B/14
CIFAR-100	10	**91.28**	78.69	90.26
CIFAR-100	20	**90.23**	76.32	89.29
CUB-200	10	**85.54**	70.75	85.45
CUB-200	20	**83.72**	65.95	83.26
ImageNet-R	10	78.63	78.72	**84.56**
ImageNet-R	20	76.13	76.47	**83.85**
ImageNet-A	10	61.98	57.35	**73.76**
ImageNet-A	20	57.10	51.52	**69.96**

At matched resolution, no backbone dominates every benchmark. ViT-B/16 is strongest on CIFAR-100 and CUB-200, whereas DINOv2 provides large gains on ImageNet-R and ImageNet-A. This supports treating the backbone as part of the experimental configuration rather than as a neutral implementation choice.

C Head and Inference-Rule Factorial Analysis

The balanced factorial grid contains five heads, seven inference rules, four datasets, two stream lengths, and three seeds, for 840 observations. We fit the fixed-effects model

$$\mathrm{Acc} \sim \mathrm{Head} + \mathrm{Rule} + \mathrm{Head{:}Rule} + \mathrm{Dataset} + T,$$

treating seeds as replicates and reporting $\eta^2 = \mathrm{SS}_{\mathrm{term}}/\mathrm{SS}_{\mathrm{total}}$. The routing signal is not fitted separately because each rule uses one fixed signal; signal is therefore nested within rule and cannot be identified independently.

Table 9: Variance explained by each term. The right column removes benchmark difficulty (Dataset and T) and reports each term as a share of the remainder.

Term	η^2 (%)	Share after removing Dataset and T (%)
Dataset	51.1	–
T	2.8	–
Classification head	31.7	68.8
Inference rule	3.1	6.7
Head $\times$ inference rule	2.7	5.9
Residual	8.5	18.4

Within a dataset, the head accounts for most of the controllable variance. The rule main effect is smaller, but the head–rule interaction is of comparable size, so a rule should not be evaluated independently of its readout head.

C.1 Average Effects and Interactions

Table 10 reports marginal means relative to the grand mean of 68.5%. These effects describe average behavior across the entire grid, not the best conditional head–rule pair.

Table 10: Centered main effects in accuracy points.

Head	Effect	Inference rule	Effect
`lstsq`	$+7.1$	`maha_e`	$+4.1$
`ridge`	$+5.4$	`maha_s`	$+4.0$
`linear`	$+5.3$	`ens`	-0.7
`ncm`	-0.8	`merge`	-0.9
`linmask`	-17.0	`ens_p`	-0.9
		`vote`	-1.5
		`select`	-4.1

The largest deviations from an additive model are:

- `ncm` with `select`: -8.2 points;
- `linmask` with `maha_s`: $+5.8$ points;
- `linear` with `select`: $+4.4$ points;
- `lstsq` with `maha_s`: -4.3 points;
- `ncm` with `maha_s`: $+3.6$ points; and
- `ncm` with `maha_e`: $+3.6$ points.

C.2 Rank Stability Across Datasets

The mean pairwise Spearman correlation is 0.85 for head rankings and 0.94 for inference-rule rankings. The dataset-specific orderings are:

Dataset	Ranking
CIFAR-100 heads	`lstsq > ridge > linear > ncm > linmask`
CUB-200 heads	`lstsq > ridge > linear > ncm > linmask`
ImageNet-R heads	`linear > lstsq > ridge > ncm > linmask`
ImageNet-A heads	`lstsq > linear > ridge > ncm > linmask`
CIFAR-100 rules	`maha_s > maha_e > ens > merge > ens_p > vote > select`
CUB-200 rules	`maha_e > maha_s > ens > ens_p > merge > vote > select`
ImageNet-R rules	`maha_e > maha_s > ens > merge > ens_p > vote > select`
ImageNet-A rules	`maha_e > maha_s > merge > ens > ens_p > vote > select`

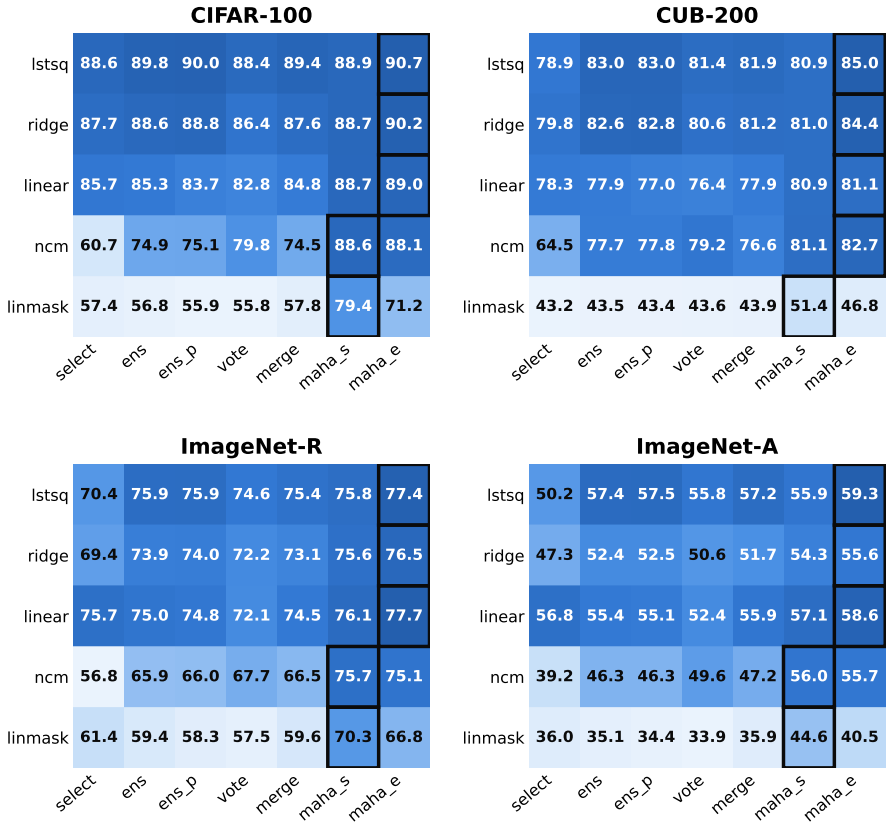


Figure 4: Final class-incremental accuracy for each head and inference rule using the ImageNet-21K pretrained ViT-B/16. Values are averaged over both stream lengths and three runs. Each outlined cell marks the strongest rule for that head.

D Classification Heads and Parameter Merging

After task t, the learned LoRA updates are combined with uniform coefficients using each parameter-merging rule and evaluated with a closed-form head. This $T = 10$ analysis uses no per-sample routing, learned merging coefficients, or previous-task images. Table 11 reports the average over the four benchmarks, while Figure 5 gives the complete results.

Table 11: Mean final class-incremental accuracy (%) across the four $T = 10$ benchmarks for each head and parameter-merging rule.

Head	Linear	TIES	DARE	SLERP	TA	DARE-TIES	Breadcrumbs	Concat.
`ncm`	**77.42**	75.49	77.41	76.51	72.33	71.51	77.08	39.06
`lstsq`	**79.44**	78.02	79.34	78.74	75.75	75.06	79.19	51.91
`ridge`	77.64	77.03	77.56	77.20	74.47	73.90	**78.10**	49.54
`gauss`	**78.44**	76.63	78.39	77.96	73.71	72.87	78.04	40.88

Direct linear merging is strongest for `ncm`, `lstsq`, and `gauss`, while Breadcrumbs is strongest for `ridge`. With the fixed `lstsq` head, linear merging reaches 91.4, 88.5, 77.7, and 60.2 on CIFAR-100, CUB-200, ImageNet-R, and ImageNet-A, respectively. These results are within 0.6 points of RanPAC on CIFAR-100 and CUB-200 and higher on ImageNet-R and ImageNet-A. Several merge rules remain close to this level, whereas concatenation is consistently poor. More elaborate merging therefore provides no uniform advantage, and the result depends on the readout head.

The generic `merge` rule has a centered effect of -0.9 in the full factorial grid (Table 10) because that quantity averages over every head, dataset, and stream length. It does not contradict the strong conditional configurations in this dedicated merging study.

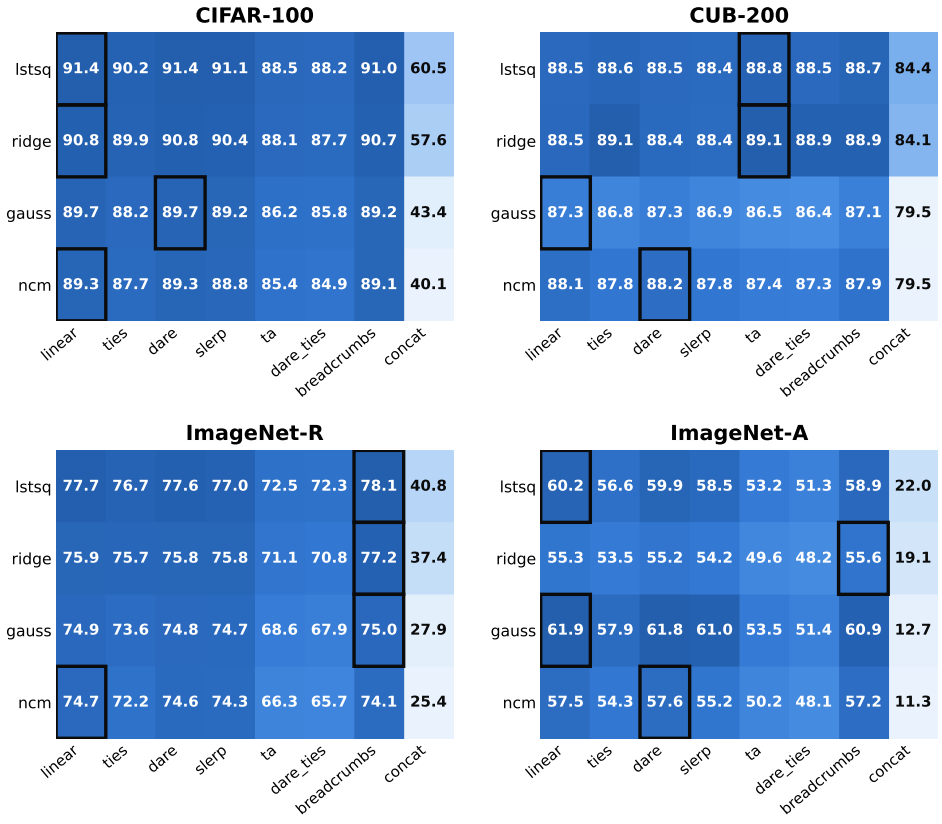


Figure 5: Final class-incremental accuracy for each classification head and parameter-merging strategy at $T = 10$. Values are means over three runs; outlined cells mark the strongest merging strategy for each head.

E Excluded GLAM Readout Diagnostic

For completeness, we retain GLAM's final merged model and replace its classifier-alignment stage with a regularized least-squares readout. The replacement head is fitted using all task training images passed through the final merged model. This intervention therefore revisits previous-task images and is not included in the controlled transfer analysis of Section 5.5. We report it only as a replay-based diagnostic.

Table 12: Final class-incremental accuracy (%, mean ± standard deviation) for native GLAM and the least-squares readout replacement.

Dataset	Tasks	GLAM	GLAM with `lstsq`	Δ
CIFAR-100	10	86.77±0.22	90.16±0.49	+3.39
CIFAR-100	20	87.43±0.50	91.02±0.07	+3.59
CUB-200	10	77.18±0.82	81.26±0.56	+4.08
CUB-200	20	75.67±1.39	80.90±0.59	+5.23
ImageNet-R	10	75.02±0.62	74.96±0.27	−0.06
ImageNet-R	20	73.78±0.18	74.32±0.09	+0.54
ImageNet-A	10	56.81±0.09	55.94±0.57	−0.87
ImageNet-A	20	57.99±0.02	57.10±0.39	−0.89

The replacement improves GLAM on CIFAR-100 and CUB-200 but is neutral or slightly worse on ImageNet-R and ImageNet-A. Because the diagnostic uses all task training images, it should not be compared directly with configurations that retain only sufficient statistics.

F Entropy- and Mahalanobis-Based Ensembling

This experiment compares entropy-weighted probability ensembling (`ens_p`) with Mahalanobis-weighted logit ensembling (`maha_e`) using the ImageNet-21K pretrained ViT-B/16. The difference is computed as `maha_e` minus `ens_p`.

Table 13: Comparison at $T = 10$. Values are final accuracy (%, mean ± standard deviation over three runs).

Configuration	CIFAR-100	CUB-200	ImageNet-R	ImageNet-A
`lstsq + ens_p`	90.74±0.14	84.43±0.16	77.32±0.18	59.40±0.28
`lstsq + maha_e`	91.39±0.09	86.04±0.35	78.66±0.08	61.26±0.55
`ridge + ens_p`	89.81±0.12	84.35±0.23	75.52±0.16	54.03±0.47
`ridge + maha_e`	90.90±0.13	85.64±0.32	77.82±0.48	57.29±0.49
Δ `lstsq`	+0.65	+1.61	+1.34	+1.86
Δ `ridge`	+1.09	+1.29	+2.30	+3.25

Table 14: Comparison at $T = 20$. Values are final accuracy (%, mean ± standard deviation over three runs).

Configuration	CIFAR-100	CUB-200	ImageNet-R	ImageNet-A
`lstsq + ens_p`	89.28±0.27	81.67±0.26	74.55±0.15	55.60±0.52
`lstsq + maha_e`	90.07±0.45	83.89±0.23	76.22±0.26	57.44±0.24
`ridge + ens_p`	87.82±0.11	81.23±0.06	72.53±0.37	50.97±0.73
`ridge + maha_e`	89.43±0.37	83.15±0.16	75.21±0.24	53.89±0.43
Δ `lstsq`	+0.79	+2.21	+1.67	+1.84
Δ `ridge`	+1.61	+1.92	+2.68	+2.92

Mahalanobis-weighted ensembling improves all eight `lstsq` settings by 1.50 points on average. The gain must be weighed against the need to store class means and precision estimates. Entropy-based probability ensembling is the simpler alternative used in the transfer experiments when multiple experts are present.

G Routing Stress Test

This experiment fixes the `lstsq` head, probability-space combination, entropy-derived combination weights, and $k = 3$. Only the candidate expert set changes. The oracle-containing condition always includes the ground-truth expert and samples the remaining $k - 1$ experts uniformly.

Table 15: Routing stress test. Values are final accuracy (%, mean ± standard deviation over three runs).

	ImageNet-R		ImageNet-A	
Candidate set	$T = 10$	$T = 20$	$T = 10$	$T = 20$
Oracle expert	89.20±0.18	91.37±0.06	76.70±0.65	79.21±0.51
Oracle-containing	83.40±0.11	84.96±0.45	68.78±0.42	70.76±0.33
Entropy top-k	78.68±0.25	77.10±0.20	63.14±0.31	60.48±0.47
Mahalanobis top-k	78.97±0.15	77.55±0.05	63.81±0.91	62.44±0.47
All experts	78.76±0.21	76.97±0.11	63.20±0.21	62.57±0.33
Random-k	75.92±0.32	73.50±0.26	59.99±0.49	59.38±0.87
Worst-k	75.05±0.32	73.12±0.11	60.54±0.73	58.75±0.61

Both learned routing signals outperform random selection, but the gap to the oracle remains substantial. Moreover, including the correct expert does not recover oracle-expert accuracy. Unrelated experts can still dilute the prediction, confirming that expert retrieval and composition must be considered jointly.

H Storage Accounting

Table 16 summarizes the state retained by each evaluated implementation. We report scaling rather than converting every parameter to a nominal float32 size, since stored dtypes and duplicated model instances differ across implementations. Exact parameter counts and runtime memory are reported with the efficiency results in the main paper.

Table 16: Stored state and its growth with the stream. Here T is the number of tasks, C the number of observed classes, d the feature dimension, and r the LoRA rank.

Method	Backbone state	Task-dependent state	Readout or routing state
Controlled configuration	One frozen backbone	One rank-r LoRA update per task; linear in T	Least-squares statistics, $O(d^2 + dC)$; routing statistics for `maha_e`
L2P	Two model instances in the evaluated implementation	Fixed prompt pool	Prompt keys and classifier parameters
DualPrompt	Two model instances in the evaluated implementation	Global and expert prompts	Prompt keys and classifier parameters
EASE	One frozen backbone	One adapter per task	Class prototypes over adapter-conditioned features
CL-LoRA	One frozen backbone	Shared and task-dependent LoRA state	Classifier and method-specific routing state
GLAM	One frozen backbone	Bounded set of merged groups	Class statistics used by classifier alignment
HiDe-LoRA	Two model instances in the evaluated implementation	Task-specific LoRA branches	Task-inference model and class statistics

For our method, the expert bank grows linearly with the number of tasks. The d^2 Gram term is independent of stream length, while the dC cross-covariance term grows with the number of observed classes. Mahalanobis routing additionally stores class means and regularized precision estimates. In the evaluated implementations, duplicated model instances dominate the stored parameter count of L2P, DualPrompt, and HiDe-LoRA, whereas GLAM bounds expert growth by maintaining a fixed number of merged groups.