STRONG BACKBONE, SIMPLE HEADS: THE CASE FOR GRADIENT-FREE CLASS-INCREMENTAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent class-incremental learning methods increasingly build on pre-trained vision transformers, adapting only a small set of parameters such as lightweight adapters. However, these methods are typically developed and benchmarked on older foundation models. We ask how they transfer to modern backbones, and whether parameter adaptation is needed at all. Comparing adapter-based methods with gradient-free, closed-form classifiers that keep only per-class statistics, under matched backbones, splits and seeds, we find that as backbones improve the gradient-free methods match or outperform gradient-based ones, even after extensive retuning of the latter. The gap is driven by forgetting rather than learning: the adapter method learns each task about as well as the closed-form heads, but forgets roughly three times as much, and its forgetting is higher on the stronger backbones while theirs falls. We also find that the composable, sufficient-statistic state of these heads enables two practical properties: federated merging across distributed instances and class-level machine unlearning. These results suggest that with strong foundation models, simple statistical decision rules can outperform increasingly elaborate adaptation mechanisms.

1 INTRODUCTION

Vision systems deployed in the real world must recognize new classes over time, yet standard supervised models are trained once on a fixed label set. Real-world settings rarely match that paradigm: new classes arise over time and the system must accommodate them without retraining. This is the framework for class-incremental learning (CIL) (Rebuffi et al., 2017; van de Ven et al., 2022), in which disjoint groups of classes arrive as a sequence of tasks, and the learner is evaluated on all classes seen so far, with no task identity given at inference. Naive training in this regime suffers from catastrophic forgetting (McCloskey & Cohen, 1989; French, 1999): gradient updates on later tasks overwrite parameters learned for earlier ones. Classical remedies, including replay buffers, parameter regularization (Kirkpatrick et al., 2017; Zenke et al., 2017), and dynamic architectures, trade memory or capacity for stability.

End-to-end fine-tuning (Rebuffi et al., 2017) gave way to prompt tuning (Wang et al., 2022b;a; Smith et al., 2023), then to small adapter modules and projection heads on a frozen backbone of a pre-trained model (PTM) (Zhou et al., 2024a;b; Sun et al., 2025; Jiang et al., 2025). All of these methods still share one liability: each task boundary triggers a new SGD optimization, which can interfere with earlier tasks and accumulate error over long streams. Gradient-free alternatives such as RanPAC (McDonnell et al., 2023) and FeCAM (Goswami et al., 2023) avoid this by fitting closed-form classifiers on the frozen features. Large self-supervised vision transformers such as DINOv2 or DINOv3 (Oquab et al., 2024; Siméoni et al., 2025) produce representations that are already close to linearly separable across many tasks, changing what a CIL system needs to do. In CIL built on such models, the open problem is no longer how to learn good features without forgetting, but rather how to attach and maintain a decision rule on top of a fixed feature space across an evolving stream of tasks.

Across CIL research in computer vision, ViT-B/16-IN21k has been the long-standing standard backbone for continual learning (Dosovitskiy et al., 2021; Zhou et al., 2024a; Jiang et al., 2025; Sun et al., 2025; McDonnell et al., 2023; Goswami et al., 2023). However, foundation models have since moved on. Self-supervised models such as DINOv2 and DINOv3 (Oquab et al., 2024; Siméoni et al.,

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.

2025) are larger, are trained on far more data, and produce features with different statistics; DINOv3's CLS features, for example, are five to seven times smaller in norm than those of the supervised ViT. Prompt-based CIL methods are already known to degrade under self-supervised pre-training (Wang et al., 2023), but adapter-based methods have not been tested this way. Whether the reported ranking of methods survives the change of backbone is therefore an open question.

We ask how much of this adaptation is necessary once the backbone is strong enough. To answer this, we pair frozen DINOv2 and DINOv3 backbones with a family of gradient-free streaming classifiers (nearest-class-mean, streaming LDA and QDA, FeCAM and RanPAC). We compare them with gradient-based adaptation under one protocol (the same features, splits, five seeds and metrics) across five backbones, from ViT-B/16-IN21k through DINOv2-B to DINOv3-B, -L and -H.

Our contributions are:

- **A matched comparison across backbone generations.** Under one protocol, on eight benchmarks including two under distribution shift, closed-form heads trail MiN on the older backbones. On DINOv3 they outperform MiN on most benchmarks and they lead EASE on every DINO backbone (Section 5.1).
- **Composability as the property that separates the two approaches.** We define composable heads, whose state after a stream is exactly the state of a joint fit; a class-separable head never rewrites the state of an earlier class. Our measurements are consistent with this: the closed-form heads learn each task as well as a joint fit would, the accuracy gap to adaptation is a gap in forgetting rather than learning, and the heads' forgetting falls as the backbone improves (Section 5.1). The most reliable way to stabilize MiN on DINOv3 is to stop its continual adaptation, which turns it into a composable classifier after the first task (Section 5.2).
- **Practical properties that follow from composability.** The same state gives exact federated merging, exact class deletion and invariance to task order, and a fit in seconds on a CPU, at the price of more memory than RanPAC's state (Sections 3.1 and 5.3).

One result marks the limit of these claims: under distribution shift the heads are only as good as the frozen representation (Section 5.1).

2 Related Work

PTM-based CIL with task-specific modules. Recent CIL methods increasingly favor freezing a strong pre-trained vision backbone and learning a lightweight task-specific module, including prompt-based methods such as L2P (Wang et al., 2022b), DualPrompt (Wang et al., 2022a) and CODA-Prompt (Smith et al., 2023), and adapter-based methods such as ADAM+Adapter (Zhou et al., 2024a), EASE (Zhou et al., 2024b), MOS (Sun et al., 2025), and MiN (Jiang et al., 2025). Although these approaches reduce the trainable footprint relative to end-to-end fine-tuning, they still require SGD-based adaptation at each task boundary and typically grow model state or inference complexity with the number of tasks.

Gradient-free and statistical CIL. A second line of work fits closed-form heads on frozen features and keeps per-class statistics instead of training parameters. SimpleCIL (Zhou et al., 2024a) is a cosine nearest-class-mean classifier and showed that such a head is already a strong baseline on frozen features. Streaming LDA (Hayes & Kanan, 2020) keeps class means and a shared covariance; Hayes & Kanan (2020) also evaluate a streaming QDA with per-class covariances. FeCAM (Goswami et al., 2023) classifies by Mahalanobis distance under a shrunk, normalized covariance per class. RanPAC (McDonnell et al., 2023) projects features through a fixed random nonlinear map and solves a ridge regression on the accumulated Gram matrix. In its published form RanPAC first adapts the backbone with a parameter-efficient module on the first task, following Zhou et al. (2024a); we use its frozen variant, without that step, throughout. We treat these heads as one family, the composable heads (Section 3.1). As its per-class-covariance member we use streaming QDA with each class covariance shrunk toward the pooled estimate (S-QDA). The property that organizes our comparison, composability, is the mergeability studied for data summaries in streaming algorithms (Agarwal et al., 2012).

Table 1: **Levels of composability.** State size for C classes at feature dimension d (M: RanPAC's random features). [a]Requires keeping the per-class scatters, not only SLDA's pooled one. [b]RanPAC's state composes, but its decision rule is not fixed: the ridge penalty is re-selected on held-out data, so the final classifier is not a function of the merged state alone.

Head	State	Composable	Invertible	Class-separable
NCM / SimpleCIL	class counts and sums, $O(Cd)$	✓	✓	✓
SLDA	class sums, second moments, $O(Cd + d^2)$	✓	✓	✓[a]
S-QDA	per-class counts, means, scatters, $O(Cd^2)$	✓	✓	✓
FeCAM	per-class second moments, $O(Cd^2)$	✓	✓	✓
RanPAC (frozen)	shared Gram, class cross-correlations, $O(M^2 + MC)$	✓[b]	✓	–
MiN, EASE (adaptation)	trained adapter weights	–	–	–

3 PRELIMINARIES

Class-Incremental Learning. Given a training set $\mathcal{D}$ spanning C classes, where each class c contributes n_c samples with n_c not assumed constant across c, $\mathcal{D}$ is partitioned into a sequence of T tasks $\mathcal{D} = (\mathcal{D}_1, \ldots, \mathcal{D}_T)$, where $\mathcal{D}_t$ denotes the data observed at task t. Each task t introduces a class set $\mathcal{C}_t$ of C/T new classes, pairwise-disjoint across tasks: $\mathcal{C}_s \cap \mathcal{C}_t = \emptyset$ for $s \neq t$. We write $\mathcal{C}_{1:t} = \bigcup_{s \leq t} \mathcal{C}_s$ for the set of all classes observed up to and including task t. The learner observes the tasks sequentially in a *single pass*: at task t it sees only $\mathcal{D}_t$ together with state carried over from earlier tasks. It is evaluated on $\mathcal{C}_{1:t}$, with no task identity provided at inference time. We use the standard seeded class ordering for each benchmark; splits are detailed in Sec. 4.3.

Composable state. Because the learner cannot retain or revisit all past data, its state must summarize what it has seen, and the two approaches we compare differ in what that summary is. A closed-form head keeps statistics of the frozen features that can be combined across datasets (Section 3.1); for the Gaussian heads they are also sufficient statistics of the class-conditional model (Appendix A). An adaptation method keeps trained weights (adapters, noise generators or prompts) and a classifier on the adapted features, and updates the weights by gradient descent on each new task's data. Its summary of an earlier class is therefore implicit in weights that every later task changes. Section 3.1 makes this difference precise.

3.1 COMPOSABLE HEADS

Definition. We call a classifier head *composable* if its entire state after seeing data $\mathcal{D}$ is a value $T(\mathcal{D})$ that respects combining datasets,

$$T(\mathcal{D}_1 \cup \mathcal{D}_2) = T(\mathcal{D}_1) \oplus T(\mathcal{D}_2) \tag{1}$$

for an associative and commutative operation $\oplus$, and if its predictions depend on the data only through T (and fixed hyperparameters), and the size of $T(\mathcal{D})$ does not grow with $|\mathcal{D}|$ (which excludes storing the data itself). The simplest way to obtain it is a state that is a sum, over training samples, of a fixed per-sample quantity. Every closed-form head we study is of this kind: counts and feature sums (nearest-class-mean), second moments (SLDA, S-QDA, FeCAM), or a random-feature Gram matrix and label cross-correlation (RanPAC). A gradient-trained adapter does not: its weights after training on $\mathcal{D}_1$ and then $\mathcal{D}_2$ depend on the order and path of optimization, not on $\mathcal{D}_1 \cup \mathcal{D}_2$ alone.

Levels of composability. Table 1 lists what each head supports. By equation 1 alone, a head is *streaming* ($T(\mathcal{D} \cup \{z\}) = T(\mathcal{D}) \oplus T(\{z\})$, independent of task order and split) and *mergeable* (shards combine into the joint state in one round). Two further properties hold only for some heads. A head is *invertible* if there is an operation $\ominus$ with $T(\mathcal{D} \setminus \mathcal{D}') = T(\mathcal{D}) \ominus T(\mathcal{D}')$ for every $\mathcal{D}' \subseteq \mathcal{D}$. It is *class-separable* if $T(\mathcal{D}) = (T_c(\mathcal{D}_c))_{c \in \mathcal{Y}}$, where $\mathcal{D}_c$ is class c's data, with $\oplus$ acting componentwise. Anything that combines classes, such as a pooled covariance, is then computed by the decision rule, and a class is deleted by resetting its component.

What composability implies for forgetting. Class separability gives these heads a useful property for continual learning. Once a class's task has passed, later tasks contain none of its data, so its component of the state never changes again. A class-separable head can still lose accuracy on an old class, as new classes compete for the prediction and as quantities the rule computes from several

classes, such as a pooled covariance, shift; but what it knows about the class itself is never overwritten. An adaptation method offers no such property, because every task updates parameters that all classes share.

4 EXPERIMENTS

4.1 DATASETS

We evaluate on six standard CIL benchmarks: **CIFAR-100** (100 classes, 50k train / 10k test images at 32×32) (Krizhevsky, 2009); **ImageNet-R** (200 renditions classes, 24k train / 6k test) (Hendrycks et al., 2021a); **ImageNet-A** (200 adversarial-natural classes, 5.9k train / 1.6k test) (Hendrycks et al., 2021b); **CUB-200-2011** (200 fine-grained bird species, 9.4k train / 2.4k test) (Wah et al., 2011); **OmniBenchmark** (300 classes drawn from 21 visual domains) (Zhang et al., 2022; Zhou et al., 2024b); and **VTAB** (50 classes from the Visual Task Adaptation Benchmark suite) (Zhai et al., 2019; Zhou et al., 2024b). We add two benchmarks under stronger distribution shift and report them alongside the six: **OpenEarthSensing** (RGB remote sensing; 189 classes in 9 tasks) and **MedMNIST** (three MedMNIST+ organ datasets merged into 33 classes, 11 organs in 3 imaging planes; 11 tasks); protocols are in Appendix C.

4.2 PRE-TRAINED MODEL BACKBONES

Experiments were run using feature embeddings generated by the following frozen pre-trained vision transformer models (Wolf et al., 2019): ViT-B/16-IN21k (86M), DINOv2-B/14 (86M), DINOv3-B/16 (86M), DINOv3-L/16 (300M), and DINOv3-H/16 (840M) (Dosovitskiy et al., 2021; Oquab et al., 2024; Siméoni et al., 2025). ViT-B/16-IN21k was chosen as a direct comparison to previous work, while DINOv2/3 models were selected for their high performance as generalized vision foundation models. Model backbones were loaded with their published checkpoints. We use each model's final-layer CLS token. Features use ImageNet normalization with a direct resize to 224×224 (Appendix C.1).

4.3 METHODS COMPARED, PROTOCOL, TUNING AND SCORING

Protocol: we follow the B0-IncN protocols of Zhou et al. (2024b): all classes arrive in one seeded sequence of equal-size tasks (Appendix C), and accuracy is measured on the full test set after each task. **Closed-form heads:** SimpleCIL, SLDA, S-QDA, FeCAM (best of its published α variants, chosen on the reported seeds, on the standard benchmarks, and its default $\alpha{=}(1, 1)$ on OES and MedMNIST) and RanPAC in its frozen variant, without the original first-task adaptation (published $M{=}10{,}000$), all fitted on the same cached features. **Adaptation methods:** MiN (Jiang et al., 2025) and EASE (Zhou et al., 2024b), run end to end from their reference implementations with our backbone wrappers. EASE keeps the backbone frozen and runs at its defaults, which on OES and MedMNIST we take from the closest standard benchmark (ImageNet-A and CIFAR-100). On OES and MedMNIST, MiN is the best of 22 one-factor changes to its defaults per cell (Appendix C.3), except in three cells where that configuration collapses and a fixed variant is used instead (marked in Table 2). **Tuning:** every method's hyperparameters are chosen per benchmark and backbone at seed 1993 on the test set, since our harness has no validation split; seed 1993 is also one of the five reported seeds, so absolute accuracies are optimistic for every method alike. The exception is RanPAC's ridge penalty, which it selects at each task on a held-out 20% of that task's training features, as in its reference implementation. Every reported number is then re-run at five seeds {1993, 42, 7, 2024, 31337}. Budgets differ and we report them: the median number of five-seed configurations per cell is 3 for FeCAM and 1 for S-QDA and RanPAC; MiN exposes nine coupled hyperparameters, of which we sweep two (Appendix C.3). **Scoring:** the closed-form heads, EASE and MiN on OES and MedMNIST are scored by exact-class accuracy. MiN on the six standard benchmarks uses its reference implementation's scoring, which counts a prediction as correct when it falls in the right task. This is lenient in MiN's favor: in MiN runs on the DINOv3 standard benchmarks that record both, it adds 1.0 pp on average and up to 3.7 pp (ImageNet-A), and about 2 pp on OES (Appendix E). Every comparison on the standard benchmarks is therefore conservative for the closed-form heads.

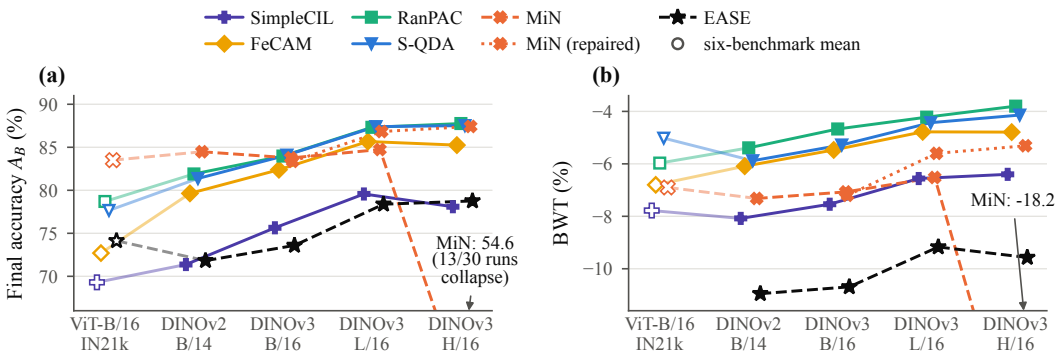


Figure 1: **The comparison with adaptation depends on backbone quality.** (a) Final accuracy A_B and (b) backward transfer against backbone. Filled markers average all eight benchmarks (Table 2); hollow markers average the six standard benchmarks at ViT-B/16, where the closed-form heads have no shift runs, so the lighter segment from ViT-B/16 to DINOv2-B/14 joins two different averages. Solid lines: closed-form heads; dashed: MiN at its tuned configuration (off the axis at DINOv3-H/16); dotted: MiN repaired per cell (Appendix E). Means of five-seed means. MiN's scoring on the standard benchmarks favors it (Section 4.3).

4.4 Metrics

We report three metrics (definitions in Appendix C.2): final accuracy A_B, the accuracy on all classes after the last task; average incremental accuracy $\bar{A}$, the accuracy on all classes seen so far, averaged over the points after each task; and backward transfer (BWT), where more negative values mean more forgetting.

5 Results

5.1 Closed-form heads keep up because they do not forget through updates

Accuracy. Figure 1a averages final accuracy over all eight benchmarks on the DINO backbones, and over the six standard benchmarks at ViT-B/16-IN21k, where the closed-form heads have no shift runs. At ViT-B/16, MiN leads the best closed-form head (RanPAC) by 4.8 pp, and at DINOv2-B/14 by 2.6 pp over all eight benchmarks. On DINOv3 the answer depends on which benchmarks are averaged. On the six standard benchmarks the two approaches are tied: tuned MiN leads the best closed-form head at DINOv3-B/16 (86.1 against 85.3%), and repaired MiN leads at DINOv3-L/16 and -H/16 by 0.2–0.3 pp (Appendix Table A6), less than the 1.0 pp its lenient scoring adds on these benchmarks. Including the two distribution-shift benchmarks, the closed-form heads lead at every DINOv3 backbone: tuned MiN trails by 0.3 and 2.7 pp at DINOv3-B and -L and collapses in 13 of its 30 standard-benchmark runs at -H, and repaired MiN stays 0.3–0.7 pp behind. The repaired variant supports our argument rather than weakening it. The repair that stabilizes MiN most reliably turns of its continual adaptation module by reducing LR to 0 (Section 5.2), converting it into only the closed-form heads after its first task. EASE trails the best closed-form head at every DINO backbone, by 9.0–10.5 pp over all eight benchmarks and 3.4–5.7 pp on the six standard benchmarks. On those six benchmarks, from ViT-B/16-IN21k to DINOv3-L/16, the best closed-form head gains 11.1 pp and MiN 3.8 pp (6.6 pp repaired): the adaptation that helps most on the weakest backbone helps least on the strong ones. Table 2 shows the same pattern on each benchmark (the same data across all four DINO backbones in Appendix Figure A1), with one exception. On ImageNet-A, EASE beats every closed-form head at every DINO backbone, by 2.4–5.8 pp, and MiN leads them there as well, though by less than the 3.7 pp its lenient scoring adds on this benchmark. ImageNet-A's images were collected because ImageNet classifiers misclassify them, and up to DINOv3-B/16 ImageNet-A is the benchmark on which every closed-form head does worst. We suspect that adapting the representation helps most where the pretrained features separate the classes least. Within the family, S-QDA beats FeCAM in 23 of 24 DINO cells on the standard benchmarks and on every shift cell, and trails RanPAC by 0.19 pp on average, ahead in 7 of 18 cells, on DINOv3 (Appendix D).

Table 2: **Final accuracy A_B and average incremental accuracy $\bar{A}$ (%) on all eight benchmarks, DINOv3 backbones.** Mean over five seeds; "Avg" is the mean over the six standard benchmarks. Bold: best in each column within a backbone. The first five rows of each block are closed-form heads. *MiN on the six standard benchmarks is, per cell, the better of its published configuration and one repair (L2-normalized features, γ=1), chosen on the reported seeds; the per-cell best over all repairs is in Figure 1 and scored by its own task-level metric, which is lenient by about 1 pp on average and up to 3.7 pp on ImageNet-A (Section 4.3); on OES and MedMNIST it is scored by exact class, and †marks the fixed variant. FeCAM is the best of its published variants on the six standard benchmarks and its default α=$(1, 1)$ on OES and MedMNIST. DINOv2-B/14, ViT-B/16, standard deviations and BWT are in Appendix Tables A5, A6 and A8.

	CIFAR		IN-R		IN-A		CUB		Omni		VTAB		Avg		OES		MedMNIST	
Method	A_B	$\bar{A}$	A_B	$\bar{A}$	A_B	$\bar{A}$	A_B	$\bar{A}$	A_B	$\bar{A}$	A_B	$\bar{A}$	A_B	$\bar{A}$	A_B	$\bar{A}$	A_B	$\bar{A}$
DINOv3-B/16																		
SimpleCIL	83.2	89.0	85.1	90.0	64.2	72.7	86.0	90.8	76.5	83.4	89.1	93.3	80.7	86.5	61.4	70.3	59.7	69.0
SLDA	85.3	90.7	84.7	89.7	64.7	70.1	88.7	92.5	79.7	86.3	94.3	96.3	82.9	87.6	73.0	80.8	78.9	85.7
FeCAM	88.0	92.4	87.0	90.8	62.9	70.8	89.4	92.7	82.9	88.6	94.0	96.5	84.0	88.6	77.9	83.3	77.0	83.5
RanPAC	**89.6**	**93.6**	**88.7**	92.3	65.8	73.4	89.8	91.4	83.1	89.0	**94.9**	**96.9**	**85.3**	89.4	79.3	85.3	80.6	83.2
S-QDA	88.4	92.6	88.4	92.0	66.4	73.2	89.5	93.0	**83.4**	89.0	94.8	**96.9**	85.2	89.4	**80.3**	85.7	**81.5**	**87.5**
MiN*	83.7	88.2	87.7	**92.4**	66.9	77.9	**90.7**	**94.2**	83.2	**90.1**	94.1	96.7	84.4	**89.9**	79.5	**85.9**	73.8†	81.1
EASE	80.0	87.6	74.5	82.4	**71.2**	**78.5**	86.9	91.2	78.0	84.8	87.3	93.2	79.7	86.3	66.4	75.9	44.5	56.7
DINOv3-L/16																		
SimpleCIL	89.7	93.6	93.0	95.8	80.1	85.5	87.9	92.4	79.4	86.1	89.2	93.2	86.6	91.1	63.9	72.3	53.4	63.0
SLDA	90.6	94.4	92.4	95.3	81.1	82.3	90.8	94.0	81.9	88.3	94.0	96.0	88.5	91.7	75.6	83.0	78.0	84.9
FeCAM	92.2	95.2	92.3	95.0	77.7	82.8	90.4	93.6	84.7	90.1	93.8	96.3	88.5	92.2	79.2	84.4	75.1	81.6
RanPAC	**93.1**	**95.9**	94.2	96.1	81.4	86.5	90.2	92.6	**85.1**	**90.6**	94.8	96.9	**89.8**	**93.1**	80.8	86.4	79.0	83.0
S-QDA	92.2	95.2	93.5	96.0	81.6	84.8	**91.1**	94.0	85.0	90.3	**94.9**	96.9	89.7	92.9	**81.5**	**86.7**	**79.3**	**85.6**
MiN*	76.5	83.5	**94.3**	**96.6**	82.5	88.6	90.9	**94.6**	74.3	83.5	94.8	**97.1**	85.5	90.6	80.2	**86.7**	73.7	81.5
EASE	88.7	93.5	86.8	91.3	**85.6**	**89.9**	87.9	92.2	80.7	87.3	88.9	94.0	86.4	91.4	70.1	79.5	38.4	53.0
DINOv3-H/16																		
SimpleCIL	91.4	94.6	94.5	96.8	87.0	90.3	85.5	91.1	78.2	85.7	88.4	92.8	87.5	91.9	64.9	73.0	34.8	46.5
SLDA	92.8	95.6	93.3	95.9	84.8	86.2	90.4	94.0	80.9	87.9	93.6	95.8	89.3	92.6	76.5	83.8	**77.5**	**84.6**
FeCAM	93.4	96.0	93.2	95.8	82.6	86.2	89.2	93.0	82.9	89.0	93.5	96.1	89.2	92.7	79.3	84.6	67.8	75.8
RanPAC	**94.3**	**96.6**	**95.2**	96.7	85.7	89.6	90.1	**94.1**	**84.6**	**90.3**	94.6	96.8	**90.8**	**94.0**	**81.4**	**86.9**	76.2	77.7
S-QDA	93.4	96.0	94.5	96.7	84.3	87.2	**90.8**	92.3	84.3	89.8	**95.1**	96.9	90.4	93.2	81.3	86.6	76.4	83.9
MiN*	73.8	78.2	**95.2**	**97.2**	88.2	**92.4**	88.3	92.9	62.1	78.1	94.8	**97.0**	83.7	89.3	80.8†	86.8	72.7†	81.7
EASE	89.1	93.9	90.4	93.6	**89.4**	**92.4**	81.4	88.4	78.8	86.5	88.6	93.7	86.3	91.4	71.8	80.7	40.7	51.8

Forgetting. Figure 1b shows where the difference comes from (values in Appendix Table A7). The closed-form heads forget less as the DINO backbone grows: S-QDA's backward transfer improves from -5.9 pp on DINOv2-B/14 to -4.1 pp on DINOv3-H/16, and the other heads follow, with FeCAM leveling off between DINOv3-L and -H. EASE forgets 4.7–5.4 pp more than S-QDA at every DINO backbone. MiN also forgets more than S-QDA at every DINOv3 backbone, even when repaired and scored leniently: repaired MiN's backward transfer is -7.2, -5.6 and -5.3 pp at DINOv3-B, -L and -H, against -5.3, -4.4 and -4.1 pp for S-QDA, and tuned MiN reaches -18.2 pp at DINOv3-H. Splitting accuracy into learning and forgetting on the 18 DINOv3 cells (Appendix Table A9), MiN learns each new task slightly better than the closed-form heads (93.4% against 90.5–92.2%), but this number is inflated by construction: under MiN's task-level scoring the task just learned counts every prediction inside that task as correct, so MiN's first point is 100% on every benchmark but forgets about three times as much (11.2 against 3.9–4.3 pp). MiN's forgetting is higher at DINOv3-L and -H than at -B (7.6, 14.4 and 11.6 pp), while S-QDA's falls (4.7, 3.6 and 3.4 pp). Where MiN can be scored by exact class, on MedMNIST at DINOv3-H/16, MiN learns new tasks no better than S-QDA (83.0% against 84.4%; Figure 2b), so MiN's learning advantage may come from the scoring alone. The closed-form heads lose nothing by learning in a stream: their accuracy on each task matches a joint fit on all data seen so far (intransigence is zero for SLDA and FeCAM, and within noise for S-QDA), as composability guarantees. The two approaches therefore differ by at most 3 pp in how well they learn a task but by 7 pp in how much they forget, and the forgetting gap widens in favor of the closed-form heads as features improve. If this trend holds, the next generation of backbones should widen the gap further rather than close it.

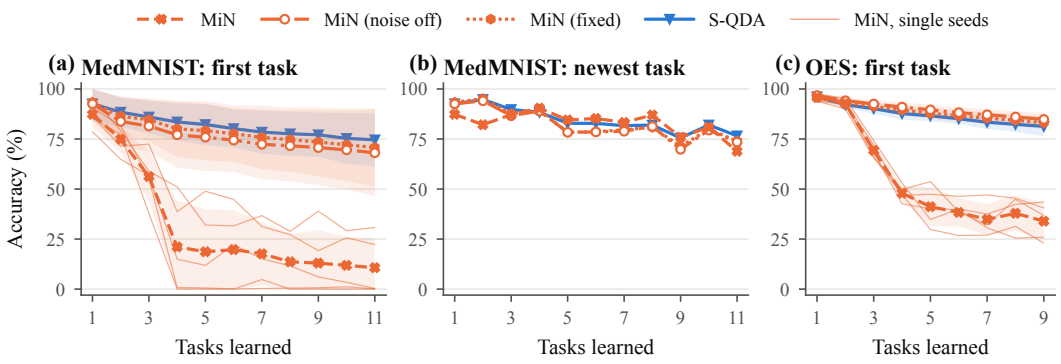


Figure 2: **Continual adaptation overwrites MiN's old classes; turning it off prevents this.** DINOv3-H/16, five seeds, exact-class scoring. (a, c) Accuracy on the first task after each later task, on MedMNIST and OES. (b) Accuracy on each new task when it is learned. Lines are seed means with ± 1 sd bands; thin lines are single seeds of standard MiN. MiN (noise off) removes its per-task noise generator, and MiN (fixed) trains it on the first task only; in both, the features are fixed after the first task.

Accuracy through the stream (Appendix Figure A2) shows the same difference in form. The closed-form heads lose accuracy gradually and in parallel as classes are added, which is what remains when no class's state is rewritten and new classes only compete for the prediction; at DINOv3-H/16, S-QDA goes from 99.0% after the second task of CIFAR-100 to 93.4% at the end. MiN's losses on CIFAR-100 and OmniBenchmark are larger and come early (97.1% to 81.6% between the second and fourth task of CIFAR-100), while on ImageNet-A, where the frozen features are weakest, it stays ahead throughout.

Distribution shift. The OES and MedMNIST columns of Table 2 test where the frozen regime breaks (numbers in Table A8). On OpenEarthSensing it holds: accuracy rises with backbone size up to DINOv3-L/16 and changes little beyond it, as on the standard benchmarks. On MedMNIST every closed-form head peaks at DINOv3-B/16 and falls at L and H. The heads fall by very different amounts: SLDA falls least (-1.4 pp), S-QDA and FeCAM fall further (-5.1 and -9.3 pp), and SimpleCIL, which uses no covariance at all, falls furthest (59.7 to 34.8%). Covariance estimation for MedMNIST's smallest classes can explain part of the extra loss, but not SimpleCIL's; its cosine prototypes appear to be hit hardest by how the larger backbones represent these images, and a shared covariance, as in SLDA, protects against it best. Adaptation does not rescue this regime: MiN trails the best closed-form head on MedMNIST at every DINO backbone, by 3.6–7.7 pp, and on OES leads only at DINOv2-B/14 and, with its noise off, at DINOv3-H/16. EASE, at its published defaults, trails every closed-form head on both benchmarks except SimpleCIL (on OES, and on MedMNIST at DINOv3-H/16), and on MedMNIST it reaches only 38–44% with backward transfer of -20 to -22 pp. A larger backbone is not automatically a better one for a closed-form head, which inherits whatever the representation does to the target domain. Fitting a closed-form head on a small labeled sample of a new domain, at two backbone sizes, is therefore a cheap way to check whether the larger backbone actually helps before relying on it.

5.2 Stabilizing adaptation on a new backbone moves it toward composability

Neither adaptation method transfers to DINO backbones at its published configuration. MiN is stable on the ViT-B/16 it was designed for (0 of 30 runs collapse across six benchmarks and five seeds), but it collapses in 7 of 30 runs on DINOv3-B/16 and 24 of 30 on DINOv3-H/16, and per-cell tuning of its learning rate and regularizer still leaves 13 of 30 collapsed at -H. We tried a series of repairs (Appendix E). Rescaling the features fixes the part of the problem caused by the smaller norm of DINO features, but the change that stabilized MiN most reliably combined that rescaling with switching the noise off, which together improve on the published configuration in 17 of 18 DINOv3 cells (by 31.8 pp on average). Figure 2 shows why, at DINOv3-H/16 and with every method scored by exact class. On MedMNIST, standard MiN learns each new task about as well as S-QDA (83.0%

against 84.4% on average), but its accuracy on the first task falls to 10.7% by the end, and to 1% or below by the fourth task in two of five seeds; on OES it falls to 34.0%. The noise it trains for each new task is applied to every input, so each update rewrites the features on which earlier classes are scored. Lowering the learning rate only delays the collapse. With the noise off, or with adaptation stopped after the first task, MiN keeps 68.1% and 70.8% of the first task on MedMNIST and 83–85% on OES, close to S-QDA (74.6% and 81.3%). With its noise off, MiN no longer updates anything that earlier classes depend on: after the first task it is a fixed feature map followed by a ridge head, a composable classifier, and in fact essentially RanPAC in its published form. It then matches our RanPAC, which skips the first-task adaptation, to within 0.05 pp on average over the 18 DINOv3 cells (with MiN under its lenient scoring). EASE shows the same from the other side. It does not collapse, but on DINOv3-B/16 its trained adapters finish below its own variant with frozen adapters (CIFAR-100, about 80 against 84.9%). On these backbones, the most reliable repairs are the ones that stop MiN's continual adaptation and so make it composable after the first task.

5.3 Algebraic composability: federated merging and class unlearning

Composability (Section 3.1) gives two operations that gradient-based methods cannot perform exactly: federated aggregation and class unlearning (Appendix A.1). Federated merging applies $\oplus$: each node trains independently and transmits its statistics to a server, which merges them in a single communication round, with no raw data or model parameters exchanged. Unlearning applies $\ominus$: removing class c reduces to deleting $(n_c, \boldsymbol{\mu}_c, \mathbf{S}_c)$ in closed form, with no replay or retraining. Gradient-based methods maintain learned adapter weights that admit neither operation (Appendix A.2); we therefore evaluate these properties only for the closed-form heads.

What is exact. Merging, class deletion and task reordering are exact for the S-QDA state, to float32 noise ($\leq 4.1 \times 10^{-7}$ relative; Table 1, Appendix A.3).

In practice. Because the S-QDA state is exact under merging and deletion, a model merged from any number of nodes, or with classes deleted, equals the centrally trained model; we verified this to float32 noise for a two-node merge and for class deletion (Appendix A.3). RanPAC shows why the level of composability matters: its accumulated statistics merge exactly, but its ridge penalty has to be re-chosen without data, and a two-node merge then loses up to 2.7 pp on ImageNet-A (Appendix A.4). The one part of the state that does not compose is where merging costs accuracy.

6 Discussion

Adaptation versus backbone quality. Our results say less about any one adaptation method than about how CIL methods are evaluated. Both adaptation methods we tested were designed and tuned on ViT-B/16-IN21k, and both lose their advantage on the backbones that have replaced it: even after retuning, MiN at best matches heads that train nothing, and EASE trails them. A method's ranking on one backbone is therefore weak evidence about its ranking on the next, and a benchmark that fixes the backbone at ViT-B/16 partly measures how well a method compensates for that backbone's weaknesses. We do not claim that adaptation is unnecessary: it wins on the weaker backbones and on OES at DINOv2-B/14 (Section 5.1). The value of adapter-based architectures should therefore be re-evaluated on modern backbones rather than carried over from results on ViT-B/16.

Forgetting is a property of optimization, not of classification. Across our experiments the closed-form heads, except the nearest-class-mean baseline SimpleCIL, forget less than both adaptation methods, and their forgetting falls as the backbone improves. This follows from how their state is updated, not from their decision rules. What remains, new classes competing for the prediction, shrinks as the backbone separates classes better. In this setting forgetting is a consequence of updating state that earlier classes depend on, not an inherent cost of learning from a sequence. For method design, this suggests asking of every new component whether its update can rewrite what earlier classes rely on, and preferring components whose state composes.

Suitability for edge compute. Once features are extracted, fitting a closed-form head is cheap: S-QDA fits OmniBenchmark on DINOv3-L/16 (300 classes) in 2.5 s on 8 CPU cores, whereas training MiN takes 2.19 GPU-hours (Appendix F). A device that already runs the backbone for inference can therefore learn new classes without a GPU or a training loop. Composability makes this collaborative:

each device computes its own class statistics and sends them once, and the merged model is exactly the one trained on the data of all devices. The backbone's forward pass still dominates computation on the device, and per-class covariances grow as Cd^2 (2.35 GB for S-QDA at 300 classes and d=1024), so on memory-limited devices the heads with smaller state (NCM, SLDA, RanPAC) are the practical choice.

Limitations. Our conclusions about adaptation rest on two methods, MiN and EASE, and other adaptation methods may transfer across backbones differently. Several aspects of the protocol favor those two methods, which makes the comparison conservative for the closed-form heads. MiN is scored by its own lenient task-level metric on the standard benchmarks, and it received the canonical DINO input preprocessing that the closed-form heads did not. Tuning budgets were not matched either: every method's hyperparameters were selected on test data at one of the reported seeds, the closed-form heads from small grids (at most 15 settings for S-QDA), MiN over two of its hyperparameters on the standard benchmarks and over 22 configurations on the shift benchmarks, while EASE ran at its published defaults, so a larger search might improve it. Beyond the protocol, three of our five backbones come from a single family, DINOv3, and we have not ruled out overlap between DINOv3's pretraining corpus and the standard benchmarks, which could inflate frozen-feature accuracy there. The remote-sensing and medical benchmarks are less likely to overlap with that corpus, and the conclusions on OES match those on the standard benchmarks.

7 Conclusion

We have shown that on modern foundation models, closed-form classifier heads that train no parameters match or exceed recent adapter-based CIL methods, and that the difference comes from forgetting rather than learning. Adapter methods developed on ViT-B/16-IN21k do not carry their advantage over to DINOv2 and DINOv3, and the most reliable way to stabilize MiN on these backbones was to switch off its continual adaptation, which turns it into a composable classifier. The same composable state gives exact federated merging and class unlearning, properties not shared by gradient-based alternatives.

Taken together, these findings suggest that as backbones improve, the key problem in class-incremental learning shifts from adapting the representation to maintaining the classifier. We hope this work encourages the community to develop and evaluate CIL methods on current foundation models rather than on the backbones they were first proposed for, and to treat composability as a first-class design criterion alongside accuracy. Where the frozen representation does not fit the data, as on MedMNIST, neither a better head nor the adaptation we tested closes the gap, which points to the representation itself as the next thing to improve.

AI use statement

We used large language model assistants in three ways. First, to write and debug the code for our experiments, and to launch and monitor those experiments on our compute cluster. Second, to draft and edit the text of this paper. Third, to a lesser extent, to help generate and discuss research ideas. Specifically, Claude Code was used for code generation, debugging and running experiments, and Claude and ChatGPT were used to help draft and review the text. The authors reviewed all code, results and text produced with this assistance, and take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

Ethics statement

Dual use. The methods we study add new classes to a recognition system cheaply, without a GPU or retraining, and let separate devices pool what they have learned. The same properties could be used in surveillance systems, or on autonomous platforms such as drones, to add new people, vehicles or other targets to what the system recognizes. Our efficiency and federated results lower the cost of such deployments. We do not study these applications and do not endorse them. **Privacy of per-class statistics.** Sharing per-class statistics instead of data is not, by itself, a privacy guarantee. When a class has few samples the statistics can be inverted: with one sample the class mean is that sample's feature vector, and with $n_c \leq d$ samples the mean and scatter matrix confine the samples to

a subspace of dimension at most $n_c - 1$ around the mean. Features from a known public backbone may in turn allow partial reconstruction of the input images. Federated merging or deletion of classes with few samples should therefore be combined with a formal protection such as differential privacy, or with a minimum class size. **Data.** All benchmarks we use are public research datasets. The medical images (MedMNIST) are de-identified and released for research.

Reproducibility statement

All datasets we use are publicly available (Section 4.1; protocols in Appendix C), and all backbones are public checkpoints (Section 4.2; preprocessing in Appendix C.1). Task splits and the five seeds are given in Section 4.3 and Appendix C, and the class order for each seed is generated by the released code; metrics are defined in Appendix C.2. Hyperparameter grids and the selected configurations are in Appendix C.3. For MiN and EASE we used the reference implementations; the changes we made to run them on DINO backbones are described in Appendices C.3 and E. The exactness checks for merging, deletion and task reordering are in Appendix A.3. All code, including feature extraction, the closed-form heads, the evaluation harness and the scripts that produce every table and figure, will be released publicly upon publication.

References

Pankaj K. Agarwal, Graham Cormode, Zengfeng Huang, Jeff M. Phillips, Zhewei Wei, and Ke Yi. Mergeable summaries. In *Proceedings of the 31st ACM Symposium on Principles of Database Systems (PODS)*, 2012.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. *CoRR*, abs/1912.03817, 2019. URL http://arxiv.org/abs/1912.03817.

George Casella and Roger L. Berger. *Statistical Inference*. Duxbury Press, Pacific Grove, CA, 2nd edition, 2002. ISBN 978-0534243128.

T. F. Chan, G. H. Golub, and R. J. LeVeque. Updating formulae and a pairwise algorithm for computing sample variances. Technical Report STAN-CS-79-773, Stanford University, 1979.

Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Robert M. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999. ISSN 1364-6613. doi: https://doi.org/10.1016/S1364-6613(99)01294-2. URL https://www.sciencedirect.com/science/article/pii/S1364661399012942.

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

Chuan Guo, Tom Goldstein, Awni Y. Hannun, and Laurens van der Maaten. Certified data removal from machine learning models. *CoRR*, abs/1911.03030, 2019. URL http://arxiv.org/abs/1911.03030.

Tyler L. Hayes and Christopher Kanan. Lifelong machine learning with deep streaming linear discriminant analysis. In *CVPR Workshops*, 2020.

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, October 2021a.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021b.

Kai Jiang, Zhengyan Shi, Dell Zhang, Hongyuan Zhang, and Xuelong Li. Mixture of noise for pre-trained model-based class-incremental learning, 2025. URL https://arxiv.org/abs/2509.16738.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, March 2017. ISSN 1091-6490. doi: 10.1073/pnas.1611835114. URL http://dx.doi.org/10.1073/pnas.1611835114.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pp. 109–165. Academic Press, 1989. doi: https://doi.org/10.1016/S0079-7421(08)60536-8. URL https://www.sciencedirect.com/science/article/pii/S0079742108605368.

Mark McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton van den Hengel. RanPAC: Random projections and pre-trained models for continual learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=aec58UfBzA.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. In *TMLR*, 2024.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning, 2017. URL https://arxiv.org/abs/1611.07725.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3, 2025. URL https://arxiv.org/abs/2508.10104.

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, 2023. URL https://arxiv.org/abs/2211.13218.

Hai-Long Sun, Da-Wei Zhou, Hanbin Zhao, Le Gan, De-Chuan Zhan, and Han-Jia Ye. Mos: Model surgery for pre-trained model-based class-incremental learning, 2025. URL https://arxiv.org/abs/2412.09441.

Gido M. van de Ven, Tinne Tuytelaars, and Andreas S. Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 4(12):1185–1197, Dec 2022. ISSN 2522-5839. doi: 10.1038/s42256-022-00568-3. URL https://doi.org/10.1038/s42256-022-00568-3.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Liyuan Wang, Jingyi Xie, Xingxing Zhang, Mingtian Huang, Hang Su, and Jun Zhu. Hierarchical decomposition of prompt-based continual learning: Rethinking obscured sub-optimality. In *NeurIPS*, 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, 2022a. URL https://arxiv.org/abs/2204.04799.

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, 2022b. URL https://arxiv.org/abs/2112.08654.

B. P. Welford. Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4(3):419–420, 1962.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771, 2019. URL http://arxiv.org/abs/1910.03771.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, 2017.

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019.

Yuanhan Zhang, Zhenfei Yin, Jing Shao, and Ziwei Liu. Benchmarking omni-vision representation through the lens of visual realms. In *European Conference on Computer Vision*, pp. 594–611. Springer, 2022.

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, 2024a. URL https://arxiv.org/abs/2303.07338.

Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. Expandable subspace ensemble for pre-trained model-based class-incremental learning, 2024b. URL https://arxiv.org/abs/2403.12030.

A COMPOSABLE STATE

This appendix gives the algebra behind the composable heads of Section 3.1 and the measurements that check it. Every closed-form head we study keeps sums over samples, so its state merges by addition; we state the merge for the per-class Gaussian triple used by SLDA, S-QDA and FeCAM, which is the least obvious case, and note where RanPAC differs. Throughout, the backbone is frozen, so each sample's feature vector is fixed, and hyperparameters are fixed within a (benchmark, backbone) cell.

A.1 SUFFICIENCY, MERGE AND INVERSE

For samples $x_1, \dots, x_n$ of one class, the triple $\mathcal{S} = (n, \mu, S)$ of count, mean and centered scatter $S = \sum_i (x_i - \mu)(x_i - \mu)^\top$ is a sufficient statistic for the mean and covariance of a Gaussian class-conditional model: the joint density depends on the data only through $\mathcal{S}$, by the factorization theorem (Casella & Berger, 2002). For the Gaussian heads, the state therefore loses nothing the model can use. Two disjoint sets of samples A and B merge exactly by the formulas of Welford (1962) and Chan et al. (1979):

$$n_{A\cup B} = n_A + n_B, \tag{2a}$$

$$\mu_{A\cup B} = \frac{n_A \mu_A + n_B \mu_B}{n_A + n_B}, \tag{2b}$$

$$S_{A\cup B} = S_A + S_B + \frac{n_A n_B}{n_A + n_B}(\mu_A - \mu_B)(\mu_A - \mu_B)^\top. \tag{2c}$$

The first two lines are sums; the third follows by expanding each scatter around the joint mean, which leaves only the between-set term. Writing $\mathcal{S}_A \oplus \mathcal{S}_B = \mathcal{S}_{A\cup B}$, the operation $\oplus$ is commutative (every term is symmetric in A and B) and associative (both groupings equal the statistic of $A \cup B \cup C$), with identity $(0, 0, 0)$. It can also be inverted: given $\mathcal{S}_{A\cup B}$ and $\mathcal{S}_B$ with $n_B < n_{A\cup B}$,

$$n_A = n_{A\cup B} - n_B, \quad \mu_A = \frac{n_{A\cup B}\,\mu_{A\cup B} - n_B\,\mu_B}{n_A}, \quad S_A = S_{A\cup B} - S_B - \frac{n_A n_B}{n_{A\cup B}}(\mu_A - \mu_B)(\mu_A - \mu_B)^\top, \tag{3}$$

which we write $\mathcal{S}_{A\cup B} \ominus \mathcal{S}_B$. Order invariance follows directly. Under any split of the stream into tasks, each class's streamed state is the $\oplus$-merge of its per-task states, which by associativity and commutativity equals the state computed from all of the class's data at once. A head whose decision depends on the data only through these states (NCM, SLDA, S-QDA, FeCAM) is therefore identical under every task order and task split. We implement $\oplus$ in the numerically stable parallel form of Chan et al. (1979).

A.2 FEDERATED MERGING AND CLASS DELETION

Merging. If the data of class c are split across W nodes, each node computes $\mathcal{S}_c^{(w)}$ from its own shard, and $\bigoplus_w \mathcal{S}_c^{(w)}$ equals the state a single node would compute from all the data. One round of communication therefore reproduces the centrally trained head, instead of approximating it over many rounds as parameter averaging does. **Deletion.** In a class-separable head (Table 1) the other classes' states never depended on class c, so deleting c means dropping $\mathcal{S}_c$; quantities pooled over classes, such as SLDA's and S-QDA's pooled scatter, are updated with $\ominus$. The result equals retraining without class c, at a cost independent of the data size. RanPAC's accumulated Gram matrix and cross-correlation are also sums and merge exactly, but its ridge penalty is re-selected on held-out data, so a merged RanPAC must recover the penalty without data (Appendix A.4).

Why gradient-trained methods do not share these properties. Prompt and adapter methods (L2P, DualPrompt, CODA-Prompt, ADAM+Adapter, EASE, MOS, MiN) accumulate their state through gradient steps whose every update depends on many classes at once, so no class's contribution can be separated from the rest. Federated training then needs either central replay or repeated parameter averaging, which only approximates the central solution, and removing a class needs retraining or an approximate unlearning procedure with bounds rather than equality (Bourtoule et al., 2019; Guo et al., 2019).

A.3 MEASURED EXACTNESS

For SLDA and S-QDA, whose state is exactly the per-class triples, merging, class deletion and task reordering are exact in exact arithmetic. Table A1 checks this on one cell. Two nodes are fitted either on disjoint blocks of tasks (task-disjoint) or on a random half of each task's samples (IID) and merged in one round; three classes are deleted; and the task order is reversed. In every case the resulting state matches the centrally trained one to float32 accumulation noise ($\leq 4.1 \times 10^{-7}$ relative), and final accuracy changes by at most 0.003 pp. FeCAM's per-class statistics also merge exactly and allow a class to be deleted in $O(1)$.

Table A1: **Exactness of the S-QDA state under composition.** Maximum absolute difference in the class scatters relative to the centrally trained magnitude / change in final accuracy (pp). One cell: MedMNIST × DINOv3-H/16, seed 1993; 2 nodes; 3 deleted classes. The 4.1×10^{-7} entries are float32 accumulation noise from a different summation order; where the order is preserved (task-disjoint, task order) the difference is exactly zero.

Head	merge, task-disjoint	merge, IID	delete	task order
S-QDA	0.0 / +0.000	4.1×10^{-7} / +0.003	3.8×10^{-7} / +0.000	0.0 / +0.000

A.4 RANPAC UNDER FEDERATED MERGING

RanPAC's accumulated statistics (the random-feature Gram matrix G and cross-correlation C) are sums and merge exactly in one round; only its ridge penalty, selected on held-out data at each session, is not determined by the merged state. Table A2 reports a two-node IID merge on DINOv3-B/16 in which the ridge is recovered without data as the geometric mean of the two nodes' ridges (the best of three data-free rules we tried: reusing one node's ridge, the geometric mean, and trace rescaling). The merge is stable except on ImageNet-A and CUB-200, where ridge recovery is noticeably less reliable.

Table A2: **RanPAC two-node IID federated merge**, DINOv3-B/16, five seeds. Gap to centralized A_B (pp) with the ridge set to the geometric mean of the two nodes' ridges. The centralized values come from this experiment's own runs, not from the main tables.

Benchmark	Centralized A_B	ΔA_B (geometric-mean ridge)
CIFAR-100	89.64 ±0.04	-0.02 ± 0.04
ImageNet-R	89.41 ±0.00	-0.46 ± 0.31
ImageNet-A	66.60 ±0.00	-2.69 ± 3.33
CUB-200	86.46 ±4.93	-0.99 ± 4.49
OmniBenchmark	83.26 ±0.00	$+0.00 \pm 0.00$
VTAB	94.55 ±0.74	-0.07 ± 0.46

B THE CLOSED-FORM HEADS

All five heads are fitted on the same frozen features and differ only in the state they keep and the rule that turns it into a prediction. Each rule below uses the state of all classes $\mathcal{C}_{1:t}$ seen so far; n_c, μ_c and S_c are class c's count, mean and scatter, and $N = \sum_c n_c$.

NCM (SimpleCIL). The state is each class's mean of L2-normalized features; a sample is assigned to the class whose mean has the highest cosine similarity (Zhou et al., 2024a). It has no hyperparameters.

SLDA. The state is each class's mean and the pooled within-class scatter $S_{\text{pool}} = \sum_c S_c$ (Hayes & Kanan, 2020). The pooled covariance $S_{\text{pool}}/(N - |\mathcal{C}_{1:t}|)$ is regularized toward a scaled identity, and a sample is assigned to the class mean with the smallest Mahalanobis distance under it.

S-QDA. The state is the per-class triple (n_c, μ_c, S_c). Each class covariance is shrunk toward the pooled one,

$$\hat{\mathbf{\Sigma}}_c = (1-\lambda)\,\frac{\mathbf{S}_c}{n_c - 1} + \lambda\,\frac{\mathbf{S}_{\text{pool}}}{N - |\mathcal{C}_{1:t}|}, \tag{4}$$

and then blended with an isotropic term, $\hat{\mathbf{\Sigma}}_c \leftarrow \eta\,\hat{\mathbf{\Sigma}}_c + (1-\eta)\,\bar{\sigma}_c^2\mathbf{I}$, where $\bar{\sigma}_c^2$ is the mean diagonal entry of $\hat{\mathbf{\Sigma}}_c$. The shrinkage $\lambda \in [0,1]$ moves the rule between a per-class-covariance discriminant ($\lambda = 0$) and a shared one ($\lambda = 1$), and $\eta = 1$ switches the blend off. A sample is assigned to $\arg\max_c \left[-\frac{1}{2}\big((\mathbf{x}-\boldsymbol{\mu}_c)^\top \hat{\mathbf{\Sigma}}_c^{-1}(\mathbf{x}-\boldsymbol{\mu}_c) + \log|\hat{\mathbf{\Sigma}}_c|\big) + \log(n_c/N)\right]$, the plug-in Gaussian discriminant with the empirical class prior. Both regularization steps are computed from the stored state at prediction time and do not change it.

FeCAM. The state is each class's mean and covariance of L2-normalized features (Goswami et al., 2023). Each covariance is shrunk as $\mathbf{\Sigma}_c + \alpha_1 \bar{d}_c \mathbf{I} + \alpha_2 \bar{o}_c(\mathbf{1}\mathbf{1}^\top - \mathbf{I})$, with $\bar{d}_c$ and $\bar{o}_c$ the mean diagonal and off-diagonal entries, normalized to a correlation matrix, and a sample is assigned to the class with the smallest Mahalanobis distance. The pair (α_1, α_2) is the α of our tables.

RanPAC (frozen). Features are projected by a fixed random matrix to $M = 10{,}000$ dimensions and passed through a ReLU; the state is the Gram matrix $G = \sum_i \mathbf{h}_i \mathbf{h}_i^\top$ and the class cross-correlation $Q = \sum_i \mathbf{h}_i \mathbf{y}_i^\top$ (McDonnell et al., 2023). Classes are scored by the ridge readout $\mathbf{h}^\top (G + \rho\mathbf{I})^{-1} Q$, with the ridge penalty ρ re-selected after each task on a held-out 20% of that task's features. We omit RanPAC's first-task adaptation of the backbone.

C Experimental protocol

Standard benchmarks. We follow the B0-IncN protocols of Zhou et al. (2024b): all classes arrive in one sequence partitioned into equal tasks. Tasks hold 5 classes on CIFAR-100 and ImageNet-R (20 and 40 tasks), 20 on ImageNet-A (10 tasks), 10 on CUB-200 and VTAB (20 and 5 tasks) and 30 on OmniBenchmark (10 tasks). MiN runs ImageNet-R in 20 tasks of 10 classes, its reference configuration. Train and test splits are the standard ones shipped with each benchmark, and accuracy is measured on the full test set of the classes seen so far after each task. Every reported number is a mean over five seeds {1993, 42, 7, 2024, 31337}.

Distribution-shift benchmarks. OpenEarthSensing (oes-rgb). Domain-1 RGB images, official ALL split: 189 classes, 63,146 training and 15,817 test images, 9 tasks of 21 classes ($189 = 9 \times 21$, no classes dropped; the closest clean split to the 10 sessions of the original OES protocol).

MedMNIST (organ3plane-224). We merge the three MedMNIST+ organ datasets at 224×224 into a single 33-class benchmark: OrganAMNIST (axial) gives classes 0–10, OrganCMNIST (coronal) 11–21 and OrganSMNIST (sagittal) 22–32, so the label set is 11 abdominal organs seen in 3 imaging planes. We merge MedMNIST's training and validation splits into our training split and use the test split unchanged, giving 72,803 training and 34,821 test images over 11 tasks of 3 classes. Classes are imbalanced: 2,206 samples on average, from 691 to 7,197.

Hyperparameter selection. Hyperparameters are chosen per benchmark and backbone at seed 1993 by final accuracy A_B on the test set; our harness has no validation split. Seed 1993 is also one of the five reported seeds, so absolute accuracies are optimistic for every method alike. RanPAC is the exception: at each task it selects its ridge penalty from $\{10^{-8}, \ldots, 10^{8}\}$ on a held-out 20% of that task's training features, as in its reference implementation, and then refits on the accumulated statistics. FeCAM runs its three published α settings at all five seeds, and we report the best per cell, chosen on the reported seeds. This applies to the six standard benchmarks; on OES and MedMNIST only FeCAM's default $\alpha{=}(1,1)$ was run. The MiN stability fixes on MedMNIST (DINOv3-B/16 and -H/16) and OES (DINOv3-H/16) were also chosen from test-set runs.

C.1 Feature preprocessing

All features are extracted once from the frozen backbone with the preprocessing specified by the checkpoint. For ViT-B/16-IN21k this is the timm transform of that checkpoint: per-channel normalization

with mean and standard deviation 0.5, bicubic resizing, and a center crop to 224×224. For the DINOv2 and DINOv3 checkpoints the features use ImageNet normalization (mean $[0.485, 0.456, 0.406]$, standard deviation $[0.229, 0.224, 0.225]$) with a direct bilinear resize to 224×224 and no center crop, and we take the final-layer-normalized CLS token (768, 1024 or 1280 dimensions). This differs from the published DINO evaluation transform (bicubic resize to 256, center crop to 224) in geometry only; we did not measure the effect of the difference. All closed-form heads in Table A6 read the same cached features for a given backbone. MiN trains end to end through its own reference pipeline, which uses the canonical evaluation geometry (bicubic resize to 256, center crop to 224) and, for DINO backbones, re-normalizes its inputs to ImageNet statistics (Appendix E). MiN therefore sees better-prepared inputs than the closed-form heads, which makes the comparison conservative for the heads. The stability fixes for MiN on MedMNIST (DINOv3-B/16 and -H/16) were, like all other selections, chosen from test-set runs.

C.2 Metrics

We report three standard CIL metrics. Let $a_{i,j}$ denote test accuracy on the classes of task j measured after the model has been trained through task i, and let B be the total number of tasks. **Final accuracy** $A_B = \frac{1}{B}\sum_{j=1}^{B} a_{B,j}$ measures accuracy on every class seen so far, evaluated only at the end of the stream. **Average incremental accuracy** $\bar{A} = \frac{1}{B}\sum_{i=1}^{B}\left(\frac{1}{i}\sum_{j=1}^{i} a_{i,j}\right)$ averages accuracy at every task step, rewarding methods that score well throughout the stream rather than just at the end. **Backward transfer** $\mathrm{BWT} = \frac{1}{B-1}\sum_{j=1}^{B-1}(a_{B,j} - a_{j,j})$ measures retroactive forgetting (more negative is more forgetting). Learning accuracy a_{kk} is accuracy on task k right after it is learned, and forgetting F is the average drop from the best to the final accuracy on each task (Chaudhry et al., 2018).

C.3 Hyperparameters

All closed-form heads are fitted on the same cached features. SimpleCIL has no hyperparameters and is run at the five seeds. RanPAC uses its published M=10,000 random features.

S-QDA and SLDA. S-QDA has two hyperparameters: the covariance shrinkage λ (Equation equation 4) and the isotropic blend η applied before the Cholesky factorization. Both come from the first stage of our sweep, $\lambda \in \{0.85, 0.93, 0.97, 0.99, 1.0\}$ and $\eta \in \{0.8, 0.9, 1.0\}$ at seed 1993; the selected values (Table A3) are then run at five seeds. SLDA uses the same values in each cell, so the two heads differ only in the covariance they keep.

We use the reference implementation of MiN (Jiang et al., 2025) and sweep its two principal hyperparameters across their ranges, selecting the best configuration per combination of backbone and benchmark cell at seed 1993: the noise learning rate and the buffer-projection scale, with hidden dimension 192 and 10 epochs per task held fixed. The same grid is applied to every benchmark and backbone pair, including the ViT-B/16-IN21k literature comparator. Each selected configuration is then re-run at the five seeds. The configurations are seed-stable on four of the five backbones (0 of 30 runs collapse); on DINOv3-H/16, 13 of 30 runs collapse, and we report the five-seed mean including them.

MiN. The reference MiN implementation exposes nine knobs. Following (Jiang et al., 2025), an exploratory pass on a subset of cells confirmed that the structural and buffer/batch defaults (`min_hidden_dim`, `min_buffer_size`, `min_buffer_batch`, `min_batch_size`, `min_init_batch_size`, `min_epochs`) transfer to our DINOv2/DINOv3 features without retuning. Two validation sweeps:

- *Hidden dimension.* We swept `min_hidden_dim` $\in \{96, 192, 384, 768\}$ crossed with three (lr,γ) settings on three benchmarks (36 cells), extended to ImageNet-A/OmniBenchmark/VTAB at $\{96, 192, 384\}$ (27 cells). The default 192 was best, or within noise of best, in all but one cell.
- *Buffer/batch.* An "all-classes" run at (`min_batch_size`, `min_init_batch_size`, `min_buffer_batch`) $= (256, 256, 1000)$ and a "paper-regime" run at $(128, 128, 16384)$ agreed within ± 1 pp on shared cells.

Table A3: S-QDA hyperparameters per benchmark and backbone.

Benchmark	Backbone	λ	η
CIFAR-100	DINOv2-B/14	0.99	1.0
CIFAR-100	DINOv3-B/16	0.93	1.0
CIFAR-100	DINOv3-L/16	0.97	0.9
CIFAR-100	DINOv3-H/16	0.99	1.0
CUB-200	DINOv2-B/14	1.0	1.0
CUB-200	DINOv3-B/16	0.97	0.9
CUB-200	DINOv3-L/16	0.97	1.0
CUB-200	DINOv3-H/16	0.99	1.0
ImageNet-A	DINOv2-B/14	0.99	0.9
ImageNet-A	DINOv3-B/16	0.99	0.8
ImageNet-A	DINOv3-L/16	1.0	0.8
ImageNet-A	DINOv3-H/16	1.0	0.8
ImageNet-R	DINOv2-B/14	0.97	1.0
ImageNet-R	DINOv3-B/16	0.97	0.9
ImageNet-R	DINOv3-L/16	0.99	0.8
ImageNet-R	DINOv3-H/16	0.99	0.8
OmniBench	DINOv2-B/14	0.85	1.0
OmniBench	DINOv3-B/16	0.85	1.0
OmniBench	DINOv3-L/16	0.85	1.0
OmniBench	DINOv3-H/16	0.85	1.0
VTAB	DINOv2-B/14	0.97	0.9
VTAB	DINOv3-B/16	0.97	0.8
VTAB	DINOv3-L/16	0.99	0.9
VTAB	DINOv3-H/16	0.99	0.8

We then fix $\texttt{min_hidden_dim} = 192$, $\texttt{min_buffer_size} = 8192$, $\texttt{min_init_lr} = 10^{-3}$, $\texttt{min_epochs} = 10$, cosine LR schedule, and $\texttt{min_init_batch_size} = \texttt{min_batch_size}$, with $(\texttt{min_batch_size}, \texttt{min_buffer_batch}) = (128, 500)$ for $\leq$ 20-task benchmarks and $(64, 256)$ for ImageNet-R at 40 tasks (OOM safety on DINOv3-H/16).

The main MiN grid sweeps the two sensitive knobs, $\texttt{min_lr} \in \{10^{-4}, 5{\cdot}10^{-4}, 10^{-3}\}$ and $\texttt{min_gamma} \in \{10, 50, 100, 500, 1000\}$ (15 runs/cell), on the standard six benchmarks and three DINOv3 backbones. After this initial sweep, several models were experiencing poor performance on several of the benchmarks so for those that had not yet converged we performed a second sweep with $\texttt{min_lr} \in \{10^{-6}, 10^{-5}, 5{\cdot}10^{-5}\}$ and $\texttt{min_gamma} \in \{10, 25, 50, 75, 100\}$ (15 runs/cell). A few selected configurations in Table A4 come from earlier exploratory runs outside these grids: ImageNet-A on DINOv3-L/16 uses $\texttt{min_lr}{=}3{\cdot}10^{-4}$, CIFAR-100 on DINOv3-H/16 uses $\texttt{min_init_lr}{=}10^{-4}$, and VTAB on DINOv2-B/14 was selected at seed 42.

On OpenEarthSensing and MedMNIST we instead searched MiN's configuration per cell over 22 one-factor changes to its defaults, each run at full length at seed 1993 (configurations that exceed memory at batch size 512 were dropped for DINOv3-L and -H), and re-ran the best at five seeds, scored exact-class.

EASE. EASE (Zhou et al., 2024b) runs from its reference implementation with the backbone frozen and its default hyperparameters. To run it on DINO backbones we made two changes to the reference code: the backbone stays frozen while the adapters are trained on the first task, as in EASE's design, where our first port had fine-tuned it; and the adapter subspace uses the backbone's own feature width (1024 for DINOv3-L/16, 1280 for -H/16) instead of the hard-coded 768. Runs use TF32 matrix multiplication. On OES and MedMNIST, EASE uses the default configuration of the closest standard benchmark (ImageNet-A: learning rate 0.05, batch 32, weight decay 0.005; CIFAR-100: learning rate 0.025, batch 48, weight decay $5{\cdot}10^{-4}$), changing only the dataset and task sizes, and its top-5 evaluation is clamped to the number of classes seen, which only matters when fewer than five classes have been seen. EASE and MiN load OES through a loader whose train/test deduplication places 7 of the 78,963 images differently from the closed-form heads' features (63,139/15,824 against 63,146/15,817); MedMNIST splits are identical. Its harness seeds only the class split, so repeated runs are not bit-identical.

Table A4: Best MiN hyperparameters per (benchmark, backbone) cell. The two sensitive knobs swept on the main grid are `min_lr` and `min_gamma`; remaining knobs (hidden dim, buffer/batch sizes, epochs) are pinned at the validated MiN-paper defaults (Section C.3). Daggered (†) cells come from our main sweep; un-daggered cells come from an earlier sweep with full grid coverage on that (backbone, benchmark) pair.

Benchmark	Backbone	`min_lr`	`min_gamma`
CIFAR-100	DINOv2-B/14	$5{\cdot}10^{-4}$	100
	DINOv3-B/16	$5{\cdot}10^{-5}$	75
	DINOv3-L/16†	10^{-4}	10
	DINOv3-H/16	10^{-6}	10
CUB-200	DINOv2-B/14	10^{-3}	100
	DINOv3-B/16†	10^{-3}	10
	DINOv3-L/16†	10^{-3}	10
	DINOv3-H/16	10^{-5}	25
ImageNet-A	DINOv2-B/14	10^{-3}	100
	DINOv3-B/16†	10^{-3}	50
	DINOv3-L/16†	$3{\cdot}10^{-4}$	10
	DINOv3-H/16	10^{-5}	50
ImageNet-R	DINOv2-B/14	10^{-3}	100
	DINOv3-B/16	$5{\cdot}10^{-5}$	75
	DINOv3-L/16	$5{\cdot}10^{-5}$	75
	DINOv3-H/16†	10^{-4}	50
OmniBenchmark	DINOv2-B/14	10^{-3}	100
	DINOv3-B/16	$5{\cdot}10^{-5}$	50
	DINOv3-L/16	$5{\cdot}10^{-5}$	75
	DINOv3-H/16	$5{\cdot}10^{-5}$	50
VTAB	DINOv2-B/14	10^{-3}	100
	DINOv3-B/16†	10^{-3}	10
	DINOv3-L/16†	10^{-4}	10
	DINOv3-H/16	$5{\cdot}10^{-5}$	25

C.4 COMPUTE

All GPU-side workload, the one-time feature-extraction pass for the closed-form heads and every MiN sweep cell, were run on an H100-based SuperPOD (NVIDIA H100 80 GB, 8 GPUs/node) for the final results reported in this paper.

Closed-form heads: CPU-bound after feature caching. Once features are cached, all classifier-side computation operates on cached d-dimensional embeddings and needs no GPU. Their sweeps therefore run CPU-only, each task using 8 cores and 32 GB of host RAM. The one-time feature-extraction pass that produces the cache runs on a single H100.

MiN: GPU-bound at every task boundary. MiN re-runs the backbone at every task increment to update its PiNoise adapters and recursive ridge-regression head, and also performs gradient updates against a replay buffer; both phases require the backbone resident on accelerator memory. Smaller cells run as single-H100 jobs (8 CPU cores, 48 GB RAM); cells with DINOv3-L/H run as full SuperPOD nodes ($8\times$H100, 64 CPU cores, 750 GB RAM). The full-node hosts are required because MiN simultaneously holds the frozen backbone, the PiNoise adapters, a $d \times C$ ridge-regression matrix, and a replay buffer of up to $16{,}384$ raw images re-encoded on each gradient step.

D FULL RESULTS

Table A5 gives every standard benchmark with all three metrics, Table A6 the six-benchmark means with MiN and EASE, and Table A8 the two distribution-shift benchmarks. Figures A3, A4, A5 and A2 show accuracy over the task stream on every benchmark, one figure per DINO backbone, and Table A9 splits final accuracy into learning and forgetting.

Table A5: **Per-benchmark results on the six standard benchmarks.** Final accuracy A_B, average incremental accuracy $\bar{A}$ and backward transfer BWT (%), means over five seeds; "Avg." is the mean over the six benchmarks. Bold: best A_B per benchmark within each backbone. S-QDA and SLDA use matched per-cell hyperparameters, FeCAM its best α variant and RanPAC M=10,000. ¶MiN is, per cell, the better of its published and repaired (L2-normalized features, γ=1) configuration at five seeds, chosen on those seeds, and is scored at the task level (Appendix E); ‡MiN at ViT-B/16-IN21k and DINOv2-B/14 is its paper per-cell hyperparameters (no repaired runs exist at DINOv2-B/14), also task-level. §At ViT-B/16-IN21k, SimpleCIL, FeCAM and RanPAC use the timm-corrected feature cache and S-QDA the normalization-corrected cache; FeCAM there is its α=$(10, 10)$ variant, the best of the three at ViT-B/16; EASE is an earlier five-seed reproduction whose BWT was not recorded (–). EASE rows: exact-class, default hyperparameters, five seeds.

	CIFAR-100			ImageNet-R			ImageNet-A			CUB-200			OmniBench			VTAB			Avg.		
Method	A_B	$\bar{A}$	BWT	A_B	$\bar{A}$	BWT	A_B	$\bar{A}$	BWT	A_B	$\bar{A}$	BWT	A_B	$\bar{A}$	BWT	A_B	$\bar{A}$	BWT	A_B	$\bar{A}$	BWT
ViT-B/16-IN21k																					
SimpleCIL§	78.1	84.4	−6.7	53.4	61.0	−8.4	41.7	52.5	−12.2	85.9	91.0	−5.4	73.2	80.3	−8.0	83.3	89.2	−6.1	69.3	76.4	−7.8
FeCAM(10,10)§	82.6	87.7	−5.5	60.7	67.8	−7.2	42.7	51.9	−10.2	86.3	91.3	−5.2	78.6	84.6	−7.1	85.3	90.6	−5.7	72.7	79.0	−6.8
RanPAC§	89.5	93.5	−4.1	68.4	75.6	−7.2	53.7	61.6	−9.2	88.9	92.7	−4.1	78.4	85.4	−8.0	93.2	95.9	−3.3	78.7	84.1	−6.0
S-QDA	89.5	93.4	−3.9	70.2	76.2	−6.2	47.2	50.8	−5.0	86.3	91.1	−5.0	78.9	85.1	−7.1	93.9	96.2	−2.9	77.7	82.1	−5.0
MiN‡	**92.1**	95.4	−3.4	**79.8**	86.2	−6.8	**62.0**	74.4	−13.6	**90.2**	93.9	−4.0	**81.9**	89.4	−10.4	**95.1**	97.2	−3.2	**83.5**	89.4	−6.9
EASE§	86.0	91.6	–	60.6	69.7	–	54.1	63.7	–	86.4	91.4	–	74.5	81.6	–	83.2	90.5	–	74.1	81.4	–
DINOv2-B/14																					
SimpleCIL	84.1	89.5	−5.6	76.7	82.6	−5.7	61.9	70.4	−9.4	85.5	90.8	−5.3	72.1	78.9	−7.9	83.1	89.0	−6.4	77.2	83.5	−6.7
SLDA	85.9	91.2	−5.4	77.8	84.0	−5.9	63.3	68.3	−5.8	88.2	92.2	−4.5	77.0	84.0	−7.9	93.1	95.4	−2.7	80.9	85.9	−5.4
FeCAM	88.2	92.4	−4.3	79.8	84.5	−4.8	61.3	69.3	−8.3	87.8	92.1	−4.7	80.6	86.4	−6.9	90.9	94.5	−4.1	81.4	86.5	−5.5
RanPAC	89.8	93.6	−4.0	83.1	87.5	−4.3	64.8	71.7	−7.4	90.1	93.1	−3.7	**81.3**	87.6	−7.5	93.0	95.9	−3.4	83.7	88.2	−5.0
S-QDA	87.6	92.2	−4.5	82.1	86.4	−4.3	64.0	70.7	−7.7	88.3	92.2	−4.4	80.7	86.7	−6.9	93.3	95.9	−3.1	82.7	87.4	−5.2
MiN‡	**90.8**	94.6	−4.5	**87.8**	91.9	−4.5	**75.4**	83.4	−9.8	**90.9**	94.3	−3.9	79.5	86.9	−17.3	**95.3**	97.4	−3.1	**86.6**	91.4	−7.2
EASE	86.5	91.9	−6.1	71.2	79.1	−9.3	70.6	77.8	−9.1	85.6	90.8	−6.7	74.4	81.0	−7.6	83.7	90.8	−10.6	78.7	85.2	−8.2
DINOv3-B/16																					
SimpleCIL	83.2	89.0	−6.1	85.1	90.0	−4.8	64.2	72.7	−8.9	86.0	90.8	−4.8	76.5	83.4	−8.2	89.1	93.3	−4.6	80.7	86.5	−6.2
SLDA	85.3	90.7	−5.5	84.7	89.7	−4.6	64.7	70.1	−5.7	88.7	92.5	−4.2	79.7	86.3	−7.5	94.3	96.3	−2.4	82.9	87.6	−5.0
FeCAM	88.0	92.4	−4.5	87.0	90.8	−3.7	62.9	70.8	−8.3	89.4	92.7	−3.7	82.9	88.6	−6.5	94.0	96.5	−2.9	84.0	88.6	−4.9
RanPAC	**89.6**	93.6	−4.1	**88.7**	92.3	−3.3	65.8	73.4	−8.3	89.8	91.4	−1.8	83.1	89.0	−6.7	**94.9**	96.9	−2.4	**85.3**	89.4	−4.4
S-QDA	88.4	92.6	−4.4	88.4	92.0	−3.5	66.4	73.2	−7.4	89.5	93.0	−3.9	**83.4**	89.0	−6.4	94.8	96.9	−2.5	85.2	89.4	−4.7
MiN¶	83.7	88.2	−10.8	87.7	92.4	−5.1	66.9	77.9	−12.7	**90.7**	94.2	−4.0	83.2	90.1	−8.4	94.1	96.7	−3.8	84.4	89.9	−7.5
EASE	80.0	87.6	−8.4	74.5	82.4	−9.2	**71.2**	78.5	−9.7	86.9	91.2	−5.6	78.0	84.8	−7.9	87.3	93.2	−9.0	79.7	86.3	−8.3
DINOv3-L/16																					
SimpleCIL	89.7	93.6	−3.9	93.0	95.8	−2.5	80.1	85.5	−5.9	87.9	92.4	−4.7	79.4	86.1	−7.7	89.2	93.2	−4.6	86.5	91.1	−4.9
SLDA	90.6	94.4	−3.7	92.4	95.3	−2.5	81.1	82.3	−1.7	90.8	94.0	−3.7	81.9	88.3	−7.0	94.0	96.0	−2.3	88.5	91.7	−3.5
FeCAM	92.2	95.2	−3.1	92.3	95.0	−2.4	77.7	82.8	−5.6	90.4	93.6	−3.6	84.7	90.1	−6.1	93.8	96.3	−2.9	88.5	92.2	−3.9
RanPAC	**93.1**	95.9	−2.8	94.2	96.1	−1.5	81.4	86.5	−5.6	90.2	92.6	−2.9	**85.1**	90.6	−6.4	94.8	96.9	−2.4	**89.8**	93.1	−3.6
S-QDA	92.2	95.2	−3.1	93.5	96.0	−2.1	81.6	84.8	−3.7	**91.1**	94.0	−3.4	85.0	90.3	−6.1	**94.9**	96.9	−2.4	89.7	92.9	−3.5
MiN¶	76.5	83.5	−14.9	**94.3**	96.6	−2.4	82.5	88.6	−7.1	90.9	94.6	−4.0	74.3	83.5	−20.4	94.8	97.1	−3.5	85.5	90.6	−8.7
EASE	88.7	93.5	−5.2	86.8	91.3	−5.0	**85.6**	89.9	−5.0	87.9	92.2	−5.7	80.7	87.3	−7.3	88.9	94.0	−6.9	86.4	91.4	−5.9
DINOv3-H/16																					
SimpleCIL	91.4	94.6	−3.1	94.5	96.8	−2.0	87.0	90.3	−3.6	85.5	91.1	−5.4	78.2	85.7	−8.1	88.4	92.8	−5.1	87.5	91.9	−4.5
SLDA	92.8	95.6	−2.7	93.3	95.9	−2.5	84.8	86.2	−1.7	90.4	94.0	−4.1	80.9	87.9	−7.7	93.6	95.8	−2.5	89.3	92.6	−3.5
FeCAM	93.4	96.0	−2.6	93.2	95.8	−2.2	82.6	86.2	−4.3	89.2	93.0	−4.0	82.9	89.0	−6.7	93.5	96.1	−3.1	89.2	92.7	−3.8
RanPAC	**94.3**	96.6	−2.3	**95.2**	96.7	−1.3	85.7	89.6	−4.5	90.1	94.1	−4.6	**84.6**	90.3	−6.3	94.6	96.8	−2.6	**90.8**	94.0	−3.6
S-QDA	93.4	96.0	−2.5	94.5	96.7	−1.9	84.3	87.2	−3.1	**90.8**	92.3	−1.6	84.3	89.8	−6.1	**95.1**	96.9	−2.4	90.4	93.2	−3.0
MiN¶	73.8	78.2	−16.9	**95.2**	97.2	−2.1	88.2	92.4	−4.7	88.3	92.9	−5.0	62.1	78.1	−37.5	94.8	97.0	−3.4	83.7	89.3	−11.6
EASE	89.1	93.9	−5.3	90.4	93.6	−3.5	**89.4**	92.4	−3.6	81.4	88.4	−12.8	78.8	86.5	−8.1	88.6	93.7	−6.5	86.3	91.4	−6.6

Within the family and across backbones. Over the 24 DINO cells, S-QDA has higher final accuracy than FeCAM in 23 (mean +1.2 pp). Against RanPAC at its published setting it is slightly behind on the 18 DINOv3 cells: the mean gap is −0.19 pp (standard error 0.15), with S-QDA ahead in 7 cells and the per-cell gap ranging from −1.38 to +0.93 pp; on DINOv2-B/14 RanPAC is ahead by 1.0 pp. S-QDA forgets less than MiN (best five-seed arm) in all 18 DINOv3 cells.

Per benchmark the trend is not monotone: CUB-200 and OmniBenchmark peak at DINOv3-L/16 for every closed-form head (MiN peaks at DINOv3-B/16 on OmniBenchmark), so which backbone is best

Table A6: **Final accuracy and forgetting, averaged over the six standard benchmarks.** A_B and BWT (%) are six-benchmark means over five seeds {1993, 42, 7, 2024, 31337}; ± is the mean over the six benchmarks of each benchmark's seed standard deviation. SimpleCIL, FeCAM, RanPAC and S-QDA are our runs on the same cached features (RanPAC at its published M=10,000; FeCAM at its best α variant). MiN is the reference implementation (Jiang et al., 2025) run at the per-cell hyperparameters selected in our original sweep (seed 1993), then re-run at the five seeds; ‡at DINOv3-H/16 13 of the 30 runs collapse. MiN values here use MiN's task-level scoring, which is lenient in its favor (Appendix E). **Repaired MiN**: on DINOv3, the best of the published configuration and the repair variants tried in each cell (Appendix E), chosen on the reported seeds and averaged over the seeds common to all six cells (n=3 at B and H, 2 at L), so it is optimistic for MiN; on ViT-B/16 and DINOv2-B/14 standard MiN never collapsed and no per-cell repair was selected (at ViT-B/16 the rescaling repair serves as a negative control, Appendix E). EASE is the reference implementation (Zhou et al., 2024b) with the backbone frozen, run at its default hyperparameters; values are means over benchmarks of five-seed means (no sd). ViT-B/16 EASE is an earlier reproduction of the same code; its BWT was not recorded and its raw runs are not available to us. S-QDA: six-benchmark mean over five seeds from the per-seed results; its small sd reflects that S-QDA's final state does not depend on the task split. At ViT-B/16-IN21k, SimpleCIL, FeCAM (α=(10, 10)) and RanPAC use the timm-corrected features and S-QDA the normalization-corrected features. MiN BWT is task-level like its accuracy. At ViT-B/16, MiN's BWT comes from the same run as its A_B (83.50), our tuned configuration; MiN's published configuration reaches 83.68 ±0.26 at ViT-B/16. Repaired-MiN BWT averages all seeds of each cell's winning variant, whereas its A_B uses only the seeds common to all six winners, so the two columns rest on slightly different seed sets. BWT values from the new extraction are six-benchmark means without sd. Bold: best in row. Per-benchmark values are in Table A5 (Appendix).

	Final accuracy A_B (%) ↑							Backward transfer BWT (%) ↑						
Backbone	SimpleCIL	FeCAM	RanPAC	S-QDA	MiN	Repaired MiN	EASE	SimpleCIL	FeCAM	RanPAC	S-QDA	MiN	Repaired MiN	EASE
ViT-B/16-IN21k	69.3 ±0.27	72.7 ±0.24	78.7 ±0.31	77.66 ±0.13	**83.50** ±0.35	–	74.1	−7.78	−6.80	−5.97	**−5.02** ±0.44	−6.90	–	–
DINOv2-B/14	77.23 ±0.30	81.43 ±0.17	83.68 ±0.33	82.67 ±0.02	**86.62** ±0.48	–	78.65	−6.72 ±0.80	−5.51 ±0.46	**−5.04** ±0.46	−5.16 ±0.36	−7.16	–	−8.23
DINOv3-B/16	80.68 ±0.20	84.04 ±0.13	85.32 ±0.40	85.16 ±0.01	**86.13** ±0.38	85.60 ±0.21	79.65	−6.23 ±0.62	−4.93 ±0.47	**−4.45** ±0.96	−4.66 ±0.27	−6.08	−6.25	−8.29
DINOv3-L/16	86.55 ±0.13	88.51 ±0.13	89.79 ±0.69	89.72 ±0.07	87.34 ±0.39	**90.13** ±0.07	86.42	−4.88 ±0.48	−3.95 ±0.38	−3.59 ±0.97	**−3.46** ±0.15	−5.66	−4.43	−5.86
DINOv3-H/16	87.50 ±0.15	89.16 ±0.11	90.75 ±0.74	90.40 ±0.04	47.23 ±6.97‡	**90.97** ±0.21	86.27	−4.55 ±0.42	−3.80 ±0.31	−3.61 ±0.82	**−2.96** ±0.21	−21.32	−4.14	−6.63

Table A7: **Eight-benchmark means plotted in Figure 1** (DINO backbones): final accuracy A_B and backward transfer BWT (%), each the mean over the six standard benchmarks and OES and MedMNIST of five-seed means. OES and MedMNIST BWT for the closed-form heads is computed from the per-task accuracies of the runs in Table A8. MiN is scored by its task-level metric on the standard benchmarks and by exact class on OES and MedMNIST; MiN (repaired) uses its six-benchmark mean with the MiN values on OES and MedMNIST.

	A_B				BWT		
Method	DINOv2-B/14	DINOv3-B/16	DINOv3-L/16	DINOv3-H/16	DINOv2-B/14	DINOv3-B/16	DINOv3-L/16
SimpleCIL	71.4	75.6	79.6	78.1	−8.1	−7.5	−6.6
FeCAM	79.6	82.4	85.7	85.3	−6.1	−5.5	−4.8
RanPAC	81.9	84.0	87.3	87.8	−5.4	−4.7	−4.2
S-QDA	81.4	84.1	87.4	87.5	−5.9	−5.3	−4.4
MiN	84.5	83.8	84.7	54.6	−7.3	−7.1	−6.5
MiN (repaired)	–	83.4	86.8	87.4	–	−7.2	−5.6
EASE	71.8	73.6	78.4	78.8	−11.0	−10.7	−9.2

is a property of the features rather than of the classifier. MiN at its original per-cell hyperparameters improves from DINOv3-B/16 to DINOv3-L/16 (86.13 to 87.34), but at DINOv3-H/16 13 of its 30 runs collapse (Table A6). This does not show that adaptation fails to scale; it suggests that gradient-based adapters need backbone-specific retuning where a closed-form head needs none.

E MiN ON DINO BACKBONES: TUNING, DIAGNOSIS AND REPAIR

We made an effort to give MiN (Jiang et al., 2025) its best chance on every backbone; this section records what we tried, what failed, and why. *Published* is the configuration of the MiN authors

Table A8: **Distribution shift.** Final accuracy A_B (%), mean ± sd over five seeds, exact-class. SimpleCIL is cosine nearest-class-mean; FeCAM is its default α=(1, 1). SLDA and S-QDA use matched hyperparameters within each backbone. RanPAC is pinned to its published M=10,000. Bold: best in row. MiN is standard MiN at the best of a 22-configuration search per cell (Section 4.3), except where marked † (fixed MiN: first-task learning rate 10^{-4}, and at DINOv3-H/16 also adapting once; Table A10), which is used where the searched configuration collapses. At DINOv3-H/16, standard MiN reaches 57.04 ±2.54 (OES) and 40.63 ±19.21 (MedMNIST), and noise-off MiN 84.42 ±0.19 and 72.05 ±2.00. At ViT-B/16-IN21k, for reference, MiN reaches 82.28 ±0.67 (OES) and 76.35 ±1.09 (MedMNIST); the closed-form heads were not run on that backbone. EASE runs at its published defaults, taken from ImageNet-A for OES and from CIFAR-100 for MedMNIST, without tuning.

Backbone	SimpleCIL	SLDA	S-QDA	RanPAC	FeCAM	MiN	EASE
OpenEarthSensing (RGB), 189 classes, 9 tasks							
DINOv2-B/14	54.84 ±0.09	69.70 ±0.06	76.96 ±0.30	76.00 ±0.08	74.33 ±0.08	**81.71** ±0.37	64.12 ±0.69
DINOv3-B/16	61.35 ±0.09	72.99 ±0.04	**80.25** ±0.18	79.28 ±0.06	77.87 ±0.05	79.46 ±0.72	66.38 ±0.30
DINOv3-L/16	63.91 ±0.10	75.57 ±0.06	**81.53** ±0.44	80.77 ±0.07	79.19 ±0.06	80.22 ±0.59	70.10 ±1.30
DINOv3-H/16	64.89 ±0.09	76.54 ±0.08	81.33 ±0.37	**81.36** ±0.61	79.27 ±0.05	80.83 ±0.69†	71.75 ±0.77
MedMNIST organ, 3 planes, 33 classes, 11 tasks							
DINOv2-B/14	52.69 ±1.03	75.09 ±0.64	**78.02** ±0.51	77.12 ±0.57	74.20 ±0.61	74.45 ±2.15	38.51 ±1.26
DINOv3-B/16	59.70 ±0.92	78.90 ±0.59	**81.50** ±0.52	80.62 ±0.60	77.03 ±0.57	73.82 ±2.27†	44.47 ±5.14
DINOv3-L/16	53.37 ±0.85	77.95 ±0.56	**79.32** ±0.48	79.01 ±0.55	75.10 ±0.52	73.68 ±1.45	38.41 ±4.65
DINOv3-H/16	34.84 ±0.46	**77.46** ±0.55	76.38 ±0.47	76.18 ±0.55	67.76 ±0.70	72.73 ±1.65†	40.74 ±4.78

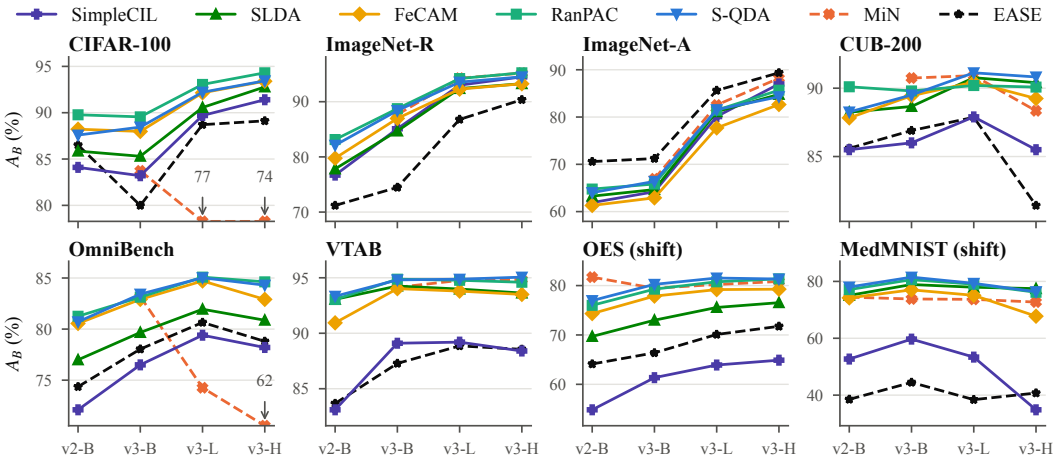


Figure A1: **Final accuracy on all eight benchmarks.** A_B (%) against backbone (DINOv2-B/14; DINOv3-B/16, -L/16, -H/16), mean over five seeds. The last two panels are the distribution-shift benchmarks. Solid lines: closed-form heads on frozen features . Dashed: adaptation methods at five seeds. EASE (black, star); MiN (orange) on the DINOv3 backbones. On the six standard benchmarks the MiN line is, per cell, the better of MiN's published and repaired (L2-normalized features, γ=1) configuration at five seeds, chosen on those seeds (Appendix E); on OES and MedMNIST it is standard MiN, or fixed MiN where the searched configuration collapses (Table A8). Values below a panel's axis are marked with an arrow.

($\gamma = 500$, noise injection on). *Paper HPs* are the per-cell hyperparameters from our own sweep (Appendix C.3), reported in Table A6. *Repaired* variants change MiN's feature scaling or noise, as described below.

Table A10 fixes the names we use for MiN from here on. The variants are not interchangeable: they differ in how much of the backbone is trained by gradient, and therefore in which composability level (Table 1) they reach.

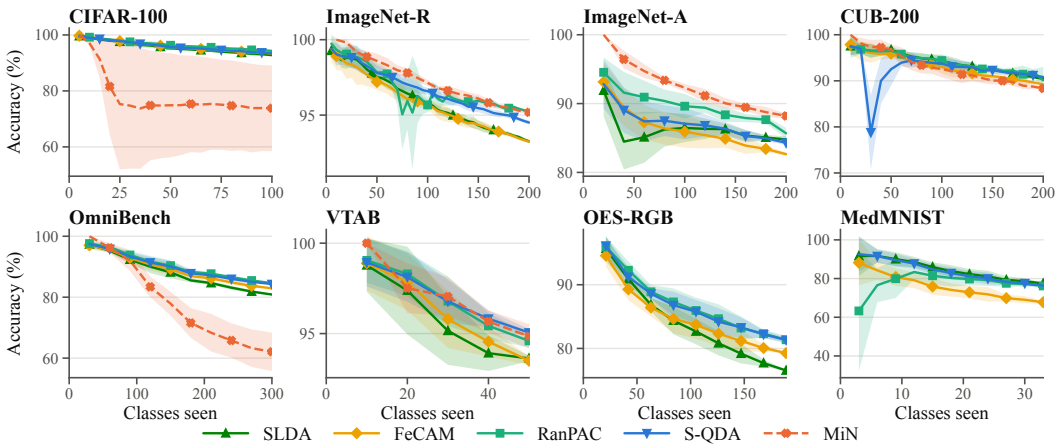


Figure A2: **Accuracy over the task stream at DINOv3-H/16, all eight benchmarks.** After each task, the accuracy on every task seen so far (the mean of the per-task accuracies), plotted against the number of classes seen. The last point of each curve is the final accuracy A_B. Lines are means over five seeds and bands ± 1 sd. On the standard benchmarks the four closed-form heads decline slowly and in parallel. MiN stays ahead on ImageNet-A, but on CIFAR-100 and OmniBench it collapses within the first few tasks. S-QDA dips briefly on CUB-200 after the third task and then recovers. On MedMNIST, FeCAM falls furthest over the stream. RanPAC's low first point there comes from 3 of 5 seeds landing near chance on the first 3-class task; it recovers by the second task. Configurations follow the main results tables: SLDA and S-QDA at matched hyperparameters, RanPAC at M=10000, FeCAM at its best α variant on the standard benchmarks and its default α=(1, 1) on OES and MedMNIST, and MiN at the rescaled-feature recipe, the better of its two five-seed configurations in every cell here. MiN's OES and MedMNIST runs are not included; MiN on MedMNIST is shown in Figure 2. MiN is scored by its own task-level metric, so its first point is 100% by construction. On ImageNet-R, MiN runs 20 tasks of 10 classes and the heads run 40 tasks of 5. MedMNIST uses 11 tasks of 3 classes.

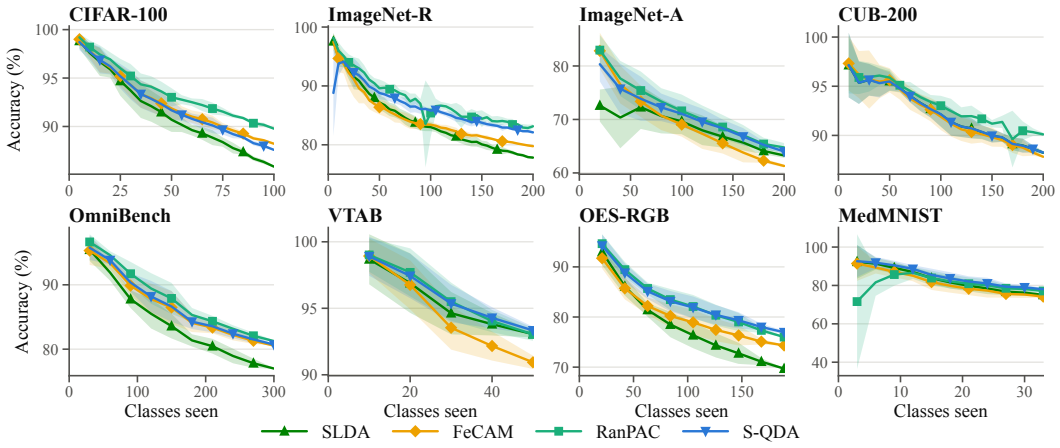


Figure A3: **Accuracy over the task stream at DINOv2-B/14, all eight benchmarks.** As in Figure A2: after each task, the accuracy on every task seen so far, against the number of classes seen; lines are means over five seeds and bands ± 1 sd, and the last point of each curve is A_B. MiN has no per-task runs at this backbone. On ImageNet-R, MiN runs 20 tasks of 10 classes and the heads run 40 tasks of 5. MedMNIST uses 11 tasks of 3 classes.

The collapse. On the ViT-B/16-IN21k backbone MiN was designed for, its published configuration is stable (0 of 30 runs collapse across the six benchmarks and five seeds). On DINOv3 backbones it is not: 7 of 30 runs collapse on DINOv3-B/16, 0 of 30 on DINOv3-L/16 and 24 of 30 on DINOv3-H/16.

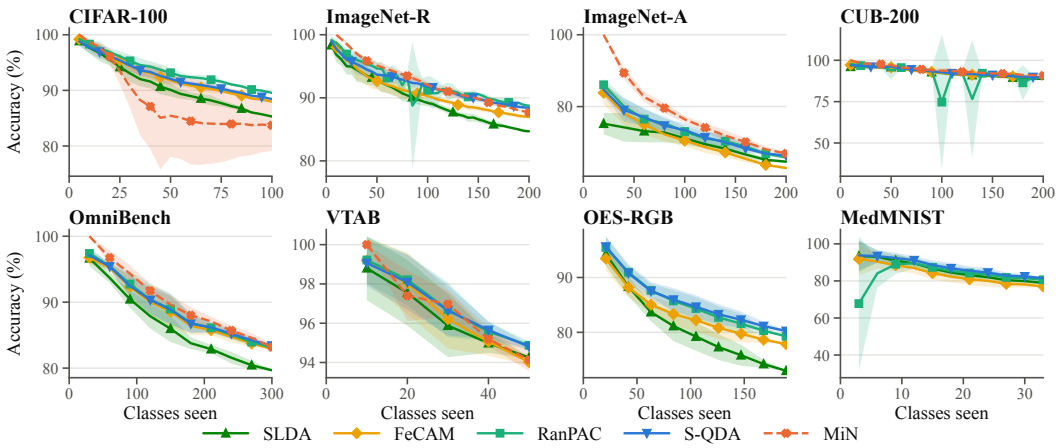


Figure A4: **Accuracy over the task stream at DINOv3-B/16, all eight benchmarks.** As in Figure A2: after each task, the accuracy on every task seen so far, against the number of classes seen; lines are means over five seeds and bands ± 1 sd, and the last point of each curve is A_B. MiN uses, per cell, the better of its published and rescaled-feature configurations at five seeds, and is scored by its own task-level metric, so its first point is 100% by construction; its OES and MedMNIST runs are not included. On ImageNet-R, MiN runs 20 tasks of 10 classes and the heads run 40 tasks of 5. MedMNIST uses 11 tasks of 3 classes.

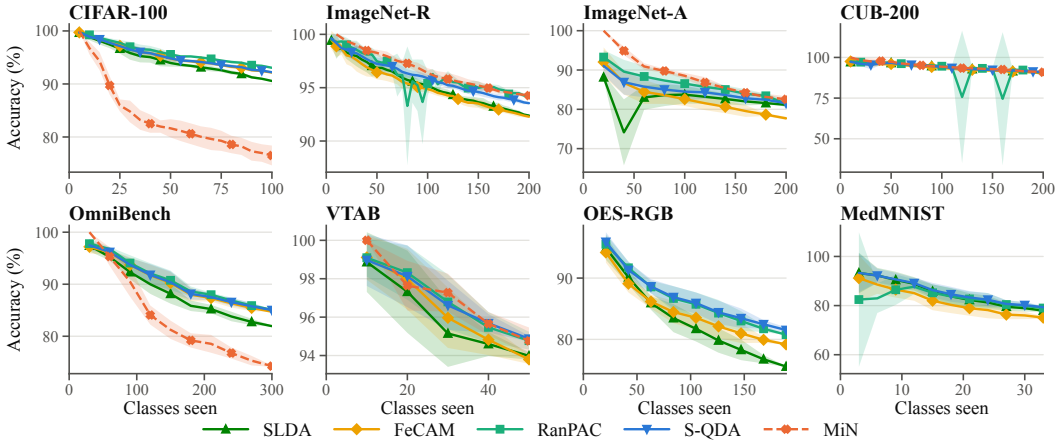


Figure A5: **Accuracy over the task stream at DINOv3-L/16, all eight benchmarks.** As in Figure A2: after each task, the accuracy on every task seen so far, against the number of classes seen; lines are means over five seeds and bands ± 1 sd, and the last point of each curve is A_B. MiN uses, per cell, the better of its published and rescaled-feature configurations at five seeds, and is scored by its own task-level metric, so its first point is 100% by construction; its OES and MedMNIST runs are not included. On ImageNet-R, MiN runs 20 tasks of 10 classes and the heads run 40 tasks of 5. MedMNIST uses 11 tasks of 3 classes.

Tuning MiN's hyperparameters per cell reduces but does not remove this: at our paper HPs, 13 of 30 DINOv3-H/16 runs still collapse. At DINOv3-H/16 the collapse is dominated by run-to-run nondeterminism at the first task, and byte-identical configurations at a fixed seed have given final accuracies of 10.5, 26.0 and 13.6%.

Not a preprocessing artifact. MiN's pipeline normalizes images with mean and standard deviation 0.5, which is correct for its original backbone but not for DINO checkpoints, which expect ImageNet statistics. Our MiN wrapper for DINOv2 and DINOv3 applies a fixed per-channel affine map that converts the pipeline's input into ImageNet-normalized input before the backbone. A runtime probe

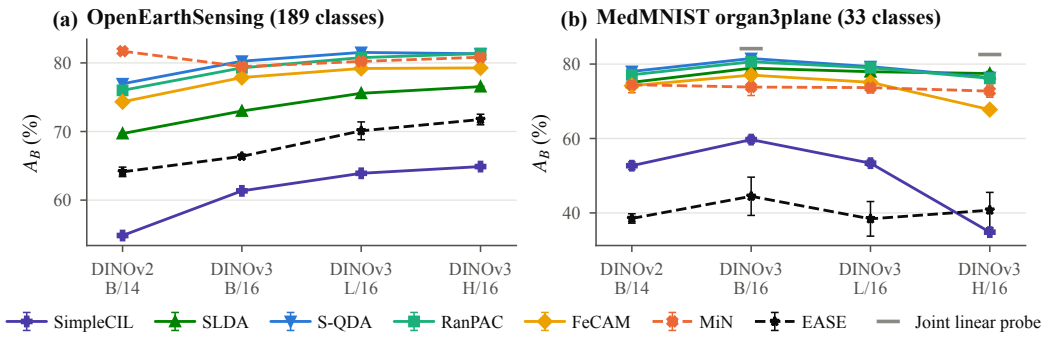


Figure A6: **Where the frozen regime breaks.** Final accuracy A_B (mean $\pm$ sd over 5 seeds) of four closed-form heads and MiN against backbone on two distribution-shifted benchmarks. **(a)** OpenEarthSensing (domain-1 RGB, 189 classes, 9 tasks): accuracy rises with backbone scale for every head. **(b)** MedMNIST organ3plane (33 classes, 11 tasks): accuracy peaks at DINOv3-B/16 and every head falls from there to DINOv3-H/16, the per-class-covariance heads most (S-QDA -5.12, FeCAM -9.27 pp versus SLDA -1.44). Gray dashes mark the joint linear-probe accuracy, measured only at DINOv3-B/16 (84.15) and DINOv3-H/16 (82.57); its 1.58 pp drop accounts for only part of the heads' decline. RanPAC uses M=10000; FeCAM uses its default α=$(1, 1)$; SimpleCIL is cosine nearest-class-mean; SLDA and S-QDA share hyperparameters within each backbone. Dashed: MiN (Table A8; fixed MiN at MedMNIST DINOv3-B/16 and -H/16 and OES DINOv3-H/16). EASE runs at its published defaults, taken from the closest standard benchmark.

Table A9: **Learning accuracy / forgetting** (%), six-benchmark means of five-seed means. Learning accuracy a_{kk} is accuracy on each task right after learning it; forgetting F is the average drop from the best to the final accuracy on each task (Chaudhry et al., 2018). "–": not measured. At ViT-B/16-IN21k, SLDA and S-QDA use the normalization-corrected features and SimpleCIL, RanPAC and FeCAM (α=$(10, 10)$) the timm-corrected features. ‡MiN at DINOv2-B/14 is its paper per-cell hyperparameters (no repaired runs there). MiN here uses the repaired recipe (L2-normalized features, $\gamma = 1$), not the paper hyperparameters, and is scored at the task level.

Method	ViT-B/16-IN21k	DINOv2-B/14	DINOv3-B/16	DINOv3-L/16	DINOv3-H/16
SLDA	80.9 / 5.8	85.8 / 5.5	87.5 / 5.1	91.6 / 3.7	92.5 / 3.7
S-QDA	82.3 / 5.5	87.4 / 5.3	89.4 / 4.7	92.9 / 3.6	93.1 / 3.4
RanPAC	84.2 / 6.1	88.3 / 5.2	89.4 / 5.0	93.0 / 4.1	94.0 / 3.8
SimpleCIL	76.4 / 7.8	–	–	–	–
FeCAM	78.9 / 6.8	86.5 / 5.5	88.5 / 4.9	92.1 / 3.9	92.6 / 3.8
MiN (repaired recipe)	85.5 / 8.6	93.1 / 7.2‡	91.1 / 7.6	94.8 / 14.4	94.3 / 11.6

in the execution environment confirmed that the backbone receives exactly the renormalized input, which differs from transform-level ImageNet normalization by at most 3.6×10^{-7}. Every DINO MiN number we report was produced with this correction, so the collapse is not caused by input normalization.

Diagnosis: feature scale. MiN's analytic head initializes its inverse-correlation matrix as $\mathbf{R} = \mathbf{I}/\gamma$, so γ competes against $\mathbf{X}^\top\mathbf{X}$, which grows with the square of the feature norm. The published $\gamma = 500$ was tuned for ViT-B/16-IN21k features of mean norm 85.3; DINOv3 features have mean norms of 15.5 (B), 15.1 (L) and 11.4 (H), so the same γ is roughly $30\times$ stronger relative to the data. This explains the gap between the two backbone families and the severity at DINOv3-H/16. It does not explain DINOv3-B versus -L, whose norms agree within 3% but whose collapse rates differ (7/30 vs. 0/30). Poor conditioning is ruled out: the worst-case condition numbers are 550 and 17, harmless in float64.

Repairs. We tested five variants: `fn-g1`, which L2-normalizes the features and sets $\gamma = 1$; `fn-g0.3`, which uses $\gamma = 0.3$; `fn-g1-lr1e4`, which also lowers the noise learning rate to 10^{-4}; and `noiseoff-fn-g1` and `noiseoff-nofn`, which disable MiN's noise injection with and

Table A10: **MiN variants.** All share MiN's fixed random ReLU projection and recursive-least-squares analytic head; they differ in what is trained by gradient.

Label	What is trained	Kind
Standard MiN	Per-task noise generators and adapter MLP in every block, plus LayerNorm on the first task; tuned per cell	continual adaptation
Fixed MiN	First-task learning rate 10^{-4}; in its adapt-once form, used at DINOv3-H/16, the adapter MLP and LayerNorm are trained on the first task only (later-task learning rate 0) and later tasks are closed-form	adapt once
Noise-off MiN	Noise and MLP removed; LayerNorm affine parameters still trained on the first task, then frozen	adapt once

without feature normalization. Applied uniformly, the fixed recipe `fn-g1` improves on the published configuration in 15 of 18 DINOv3 cells (mean $+26.28$ pp, 95% CI $[+13.46, +39.71]$; Wilcoxon $p = 0.0008$ over cells), and disabling the noise on top improves 17 of 18 (mean $+31.79$ pp, CI $[+19.14, +45.14]$). The same `fn-g1` recipe *hurts* on ViT-B/16-IN21k in all 6 cells (mean -5.97 pp, CI $[-9.22, -2.72]$). We read this negative control as evidence for the diagnosis: a change that helped everywhere would be a tuning win, whereas this one helps only where the feature statistics are mismatched. Table A11 gives the per-cell effect of the fixed recipe. The recipe does not fix everything. On CIFAR-100 with DINOv3-L/16 it reintroduces collapse (76.5 to 54.3), and only the lower noise learning rate repairs that cell. Picking the best variant per cell (on the same seeds, so optimistically biased) shows which knob matters where: disabling the noise is the best variant in 4 of 6 DINOv3-H/16 cells and 3 of 6 DINOv3-B/16 cells, and the lower noise learning rate rescues CIFAR-100 and OmniBenchmark on DINOv3-L/16. Averaged per backbone, the repaired per-cell best reaches 85.60 / 90.13 / 90.97 on DINOv3-B / L / H (the Repaired MiN column of Table A6).

Table A11: **MiN's published configuration vs. the fixed repair recipe** (`fn-g1`: L2-normalized features, $\gamma = 1$), final accuracy A_B (%) published $\rightarrow$ repaired, five seeds per cell. The recipe is applied uniformly, so the table is selection-blind. ViT-B/16-IN21k is the negative control.

Backbone	CIFAR-100	IN-R	IN-A	CUB-200	OmniBench	VTAB
ViT-B/16-IN21k	91.38 → 89.11	79.99 → 68.14	63.69 → 51.46	90.56 → 86.49	81.22 → 78.14	95.25 → 92.95
DINOv3-B/16	23.62 → 83.72	71.68 → 87.67	56.34 → 66.94	90.74 → 89.52	49.10 → 83.19	89.64 → 94.14
DINOv3-L/16	76.51 → 54.35	73.84 → 94.26	63.56 → 82.55	88.53 → 90.92	74.26 → 72.28	87.51 → 94.76
DINOv3-H/16	10.20 → 73.81	13.32 → 95.18	21.94 → 88.20	74.23 → 88.35	28.34 → 62.09	30.38 → 94.85

Two separate causes of instability. Our search on OpenEarthSensing and MedMNIST (Appendix C.3) separates two mechanisms behind its collapses on DINOv3. *Unstable first-task adaptation.* At a first-task learning rate of 10^{-3}, training the noise and LayerNorm parameters on task 0 sometimes ends in a poor state: task-0 accuracy reaches 65–85% instead of 95–99% and then decays to about zero as tasks arrive. On MedMNIST with DINOv3-B/16 this happened in 3 of 12 runs at 10^{-3} and in 0 of 17 at 10^{-4}. The failure is stochastic rather than tied to a seed: the same seed collapsed in one run and not in another. *Continual retraining overwrites old tasks.* MiN adds a noise generator per task in every block, trains it on the new task only and applies it to every input. Its analytic head has no replay, so the resulting drift in the features becomes forgetting. On MedMNIST with DINOv3-H/16, new tasks are still learned while first-task accuracy falls to $10.7 \pm 14.8\%$ by the end of the stream, and to at most 1% by the fourth task in 2 of 5 seeds (Figure 2, exact-class). Lowering the later-task learning rate (down to 10^{-5}) or raising γ only delays the crash, and fixing the mixture weights over the noise generators at uniform does not prevent it (7.7–62.9%). Adapting once (later-task learning rate 0) with a first-task learning rate of 10^{-4} removes it on every seed: first-task accuracy at the end is 70.8% for fixed MiN and 68.1% for noise-off MiN, against 74.6% for S-QDA. The second mechanism is the one the class-separability property of Section 3.1 rules out: an update for a new task rewrites the representation that old classes are scored in.

The repair does not transfer under distribution shift. Transferred unchanged to OpenEarthSensing, `fn-g1` changes no cell significantly (paired over seeds). On the one cell with real instability, DINOv3-H/16, the mean rises by 8.62 pp but every dispersion measure worsens (sd 17.18 to 23.28,

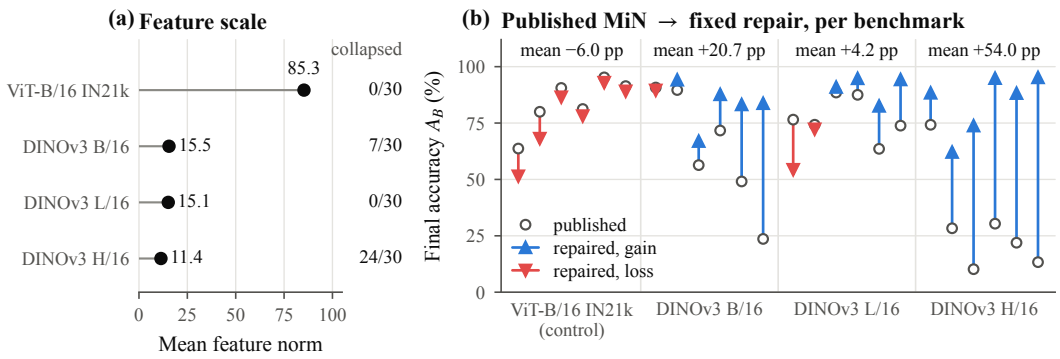


Figure A7: **MiN's published configuration is tuned to one backbone's feature scale.** (a) Mean feature norm of each backbone (measured on CIFAR-100, first task, seed 7), and the number of MiN runs that collapse (final accuracy below 50%) under the published configuration ($\gamma = 500$), out of 30 (six benchmarks $\times$ five seeds). (b) Final accuracy A_B of MiN in each of the 24 (benchmark, backbone) cells, from the published configuration (open circle) to one fixed repair applied uniformly (L2-normalized features, $\gamma = 1$), each the mean of five seeds. Cells are sorted by change within each backbone. Up-pointing triangles are gains and down-pointing triangles losses, and the group means are shown above. ViT-B/16-IN21k, the backbone MiN was tuned on, is the negative control: the repair hurts in all six of its cells, and it helps in 15 of 18 DINOv3 cells. Per-cell best variants (e.g. disabling noise injection) are not shown, because they are selected on the reported seeds.

one new collapse), so we do not count it as a repair. This is why the MiN column of Table A8 comes from a separate search on those benchmarks rather than from these runs.

A difference in accuracy definitions. MiN's evaluation code scores a prediction as correct when it falls anywhere in the correct *task* (both prediction and label are mapped to tasks before comparison), while all frozen heads are scored on the exact class. Task-level accuracy is never lower than exact-class accuracy; the difference is the within-task confusion. In later MiN runs that record both scores, the gap is 1.0 pp on average and up to 3.7 pp (ImageNet-A) on the DINOv3 standard benchmarks, and about 2 pp on OpenEarthSensing. MiN numbers on the six standard benchmarks use MiN's own (task-level) definition. The saved runs keep only scored accuracies, so they cannot be rescored; we report them as task-level and treat every comparison with them as conservative for the closed-form heads (tracked as a known limitation).

Why the closed-form heads are unaffected. The Gaussian heads are invariant to the scale of the features. If every feature is multiplied by a, every class mean and covariance, the pooled shrinkage target and the isotropic blend scale by a or a^2, all Mahalanobis distances are unchanged, and every class's log-determinant shifts by the same constant, so the decision is identical (up to a numerical fallback in our code).

F Compression and cost

Keeping only per-class statistics, rather than every feature, costs almost nothing for these decision rules: refitting each head jointly on all stored features gives almost the same accuracy as the streaming fit: streaming recovers 98.7–100.4% of the joint fit's accuracy in every cell (Table A13).

Figure A8 plots accuracy against stored state. Table A12 measures cost at our largest standard stress point (OmniBenchmark, C=300; DINOv3-L/16, d=1024) on 8 CPU cores. The per-class covariance heads (S-QDA, FeCAM) store $O(Cd^2)$ state: 2.35 GB for S-QDA, against 443 MB for RanPAC, whose M^2 Gram matrix is nearly independent of C and d. RanPAC is also cheaper at inference (37.6 vs. 974.1 ms per 1k samples). S-QDA is cheaper to fit (2.5 s including the Cholesky factorization, vs. 190.3 s), and every closed-form head is far cheaper to train than MiN (2.19 GPU-hours, measured with DINOv3-B/16). At these C and d, therefore, S-QDA does not win on memory or inference; its advantages are fit time and the structural properties of Section 5.3. State grows as Cd^2 (Table A14),

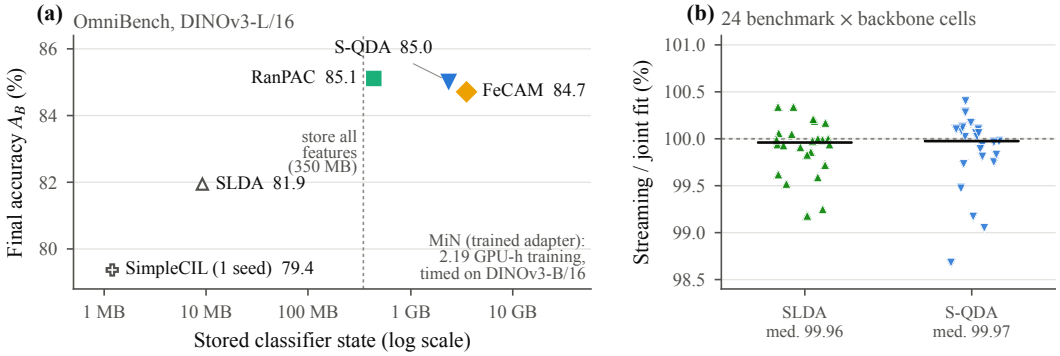


Figure A8: **Compression is nearly free in accuracy, not in memory. (a)** Final accuracy A_B against the classifier state each closed-form head stores at inference time, at the stress point (OmniBenchmark, C=300; DINOv3-L/16, d=1024; log x axis). State is the measured value from Table A12; accuracy is the 5-seed mean from the master results table (matched hyperparameters for SLDA and S-QDA; best variant for FeCAM; M=10^4 for RanPAC; sd ≤ 0.12 pp, too small to draw), except SimpleCIL, which is a single run. The dotted line marks the memory of storing every training feature instead (350 MB). MiN is trained rather than fitted, so it has no comparable state; its training cost (2.19 GPU-hours) is shown as a note and was timed on OmniBenchmark with DINOv3-B/16, since no DINOv3-L/16 timing exists. **(b)** Streaming accuracy as a percentage of the joint (batch) fit of the same head, for SLDA and S-QDA, over the 24 cells of six benchmarks × four backbones (DINOv2-B/14, DINOv3-B/L/H); the streaming-vs-joint table has no ViT-B/16 cells. Black bars are medians; the dashed line is 100 %. The lowest cell for every head is ImageNet-A with DINOv2-B/14 (98.7–99.2 %).

which becomes the binding constraint at large class counts. S-QDA keeps no features at all once a batch is folded into its statistics.

Table A12: **Efficiency and memory at the stress point.** OmniBenchmark (C=300), DINOv3-L/16 (d=1024), seed 1993, default increments. All classifier-side work on cached features, CPU-only on 8 cores / 32 GB, fp32; mean over 3 repeats after one warmup (sd in Table A15). *State* is measured by walking the fitted classifier's live tensors, not inferred from the formula, and is reported at inference time (i.e. including any derived factors the method caches). *Fit* accumulates sufficient statistics; *Derive* is the one-off solve that turns them into a classifier (Cholesky factors, FeCAM's per-class `pinv`, RanPAC's ridge system). Feature extraction is a shared one-time GPU pass and is excluded from every row. Best per column in bold, second underlined, among the five measured classifiers.

Method	State $f(C,d)$	State (measured)	Fit (s)	Derive (s)	Inference (ms / 1k)	Peak RAM
SimpleCIL	Cd	**1.2 MB**	**0.06**	**0.00**	**2.6**	**956 MB**
SLDA	d^2+Cd	<u>9.2 MB</u>	0.57	0.01	<u>8.7</u>	<u>995 MB</u>
Streaming-QDA	Cd^2	2.35 GB	<u>0.50</u>	1.95	974.1	3.78 GB
RanPAC	$M^2+dM+MC$	443 MB	190.29	<u>0.00</u>	37.6	1.62 GB
FeCAM	Cd^2	3.52 GB	0.87	66.70	1255.3	6.37 GB
Reference rows (not ranked)						
Store-all features	Nd	350 MB	–	–	–	–
MiN (PEFT, trained)	adapter	–	2.19 GPU-h	–	–	–

Table A13: **Joint vs. streaming fit**, final accuracy (%) joint / streaming, seed 1993. Recovery: streaming as % of joint.

Benchmark	Backbone	SLDA	recovery (%)	S-QDA	recovery (%)
CIFAR-100	v2-B/14	85.88 / 85.86	99.98	87.53 / 87.62	100.1
CIFAR-100	v3-B/16	85.25 / 85.29	100.05	88.48 / 88.5	100.02
CIFAR-100	v3-L/16	90.5 / 90.55	100.06	92.23 / 92.21	99.98
CIFAR-100	v3-H/16	92.8 / 92.74	99.94	93.46 / 93.43	99.97
CUB-200	v2-B/14	88.38 / 88.368	99.99	88.38 / 88.321	99.93
CUB-200	v3-B/16	88.719 / 88.718	100.0	89.567 / 89.465	99.89
CUB-200	v3-L/16	90.967 / 90.84	99.86	91.179 / 91.007	99.81
CUB-200	v3-H/16	90.416 / 90.416	100.0	90.628 / 90.886	100.28
ImageNet-A	v2-B/14	63.625 / 63.102	99.18	64.575 / 63.721	98.68
ImageNet-A	v3-B/16	64.892 / 64.854	99.94	66.857 / 66.224	99.05
ImageNet-A	v3-L/16	81.622 / 81.012	99.25	82.003 / 81.321	99.17
ImageNet-A	v3-H/16	84.981 / 84.971	99.99	84.664 / 84.212	99.47
ImageNet-R	v2-B/14	77.968 / 77.752	99.72	81.865 / 82.005	100.17
ImageNet-R	v3-B/16	84.89 / 84.57	99.62	88.622 / 88.405	99.75
ImageNet-R	v3-L/16	92.7 / 92.316	99.59	93.736 / 93.574	99.83
ImageNet-R	v3-H/16	93.637 / 93.189	99.52	94.689 / 94.429	99.73
OmniBench	v2-B/14	76.959 / 77.09	100.17	80.735 / 80.784	100.06
OmniBench	v3-B/16	79.699 / 79.564	99.83	83.409 / 83.475	100.08
OmniBench	v3-L/16	81.855 / 82.137	100.34	85.046 / 85.063	100.02
OmniBench	v3-H/16	80.969 / 80.9	99.91	84.194 / 84.294	100.12
VTAB	v2-B/14	93.039 / 93.352	100.34	93.363 / 93.741	100.4
VTAB	v3-B/16	94.35 / 94.287	99.93	94.965 / 94.925	99.96
VTAB	v3-L/16	93.967 / 94.141	100.19	94.883 / 94.976	100.1
VTAB	v3-H/16	93.48 / 93.676	100.21	95.127 / 95.18	100.06

Table A14: **Scaling of stored state and inference with C and d.** Measured, not extrapolated. (a) fixed DINOv3-L/16 (d=1024), sweeping the class count across benchmarks; (b) fixed OmniBenchmark (C=300), sweeping the backbone width. State is MB unless marked GB. The Cd^2 methods (FeCAM, Streaming-QDA) grow in both arguments; RanPAC's M^2 Gram dominates its footprint and is nearly independent of C and d; SimpleCIL and SLDA stay negligible.

(a) State (top) and inference ms/1k (bottom) vs. class count C, DINOv3-L/16.

Method	VTAB (C=50)	CIFAR-100 (C=100)	IN-R (C=200)	IN-A (C=200)	CUB-200 (C=200)	OmniBench (C=300)
SimpleCIL	0.2 MB	0.4 MB	0.8 MB	0.8 MB	0.8 MB	1.2 MB
SLDA	8.2 MB	8.4 MB	8.8 MB	8.8 MB	8.8 MB	9.2 MB
Streaming-QDA	404 MB	804 MB	1.57 GB	1.57 GB	1.57 GB	2.35 GB
RanPAC	424 MB	428 MB	436 MB	436 MB	436 MB	443 MB
FeCAM	601 MB	1.17 GB	2.35 GB	2.35 GB	2.35 GB	3.52 GB
SimpleCIL	1.0	1.3	1.8	1.9	1.9	2.6
SLDA	5.1	6.3	6.2	9.8	8.8	8.7
Streaming-QDA	197.1	446.2	728.6	742.9	983.6	974.1
RanPAC	34.0	42.6	36.3	39.4	40.7	37.6
FeCAM	241.0	424.4	845.6	896.7	993.2	1255.3

(b) State / fit (s) / inference (ms per 1k) vs. feature dimension d, OmniBenchmark (C=300).

Method	DINOv3-B/16 (d=768)	DINOv2-B/14 (d=768)	DINOv3-L/16 (d=1024)	DINOv3-H/16 (d=1280)
SimpleCIL	0.9 MB / 0.06 / 2.1	0.9 MB / 0.06 / 2.2	1.2 MB / 0.06 / 2.6	1.5 MB / 0.08 / 3.6
SLDA	5.4 MB / 0.35 / 6.0	5.4 MB / 0.35 / 5.7	9.2 MB / 0.57 / 8.7	14 MB / 0.80 / 11.4
Streaming-QDA	1.32 GB / 0.47 / 841.9	1.32 GB / 0.38 / 704.2	2.35 GB / 0.50 / 974.1	3.67 GB / 0.88 / 1689.3
RanPAC	434 MB / 184.91 / 32.2	434 MB / 191.41 / 31.9	443 MB / 190.29 / 37.6	453 MB / 190.93 / 44.1
FeCAM	1.98 GB / 0.53 / 742.0	1.98 GB / 0.60 / 833.5	3.52 GB / 0.87 / 1255.3	5.50 GB / 1.16 / 1773.3

Table A15: **Measured state vs. closed form** at OmniBenchmark (C=300), DINOv3-L/16. Ratio > 1 means the implementation retains more than the $f(C, d)$ accounting suggests. FeCAM keeps float64 sufficient statistics alongside its fp32 precisions; the Cd^2 QDA family caches one Cholesky factor per class at inference. Timing spread is the sd over 3 repeats.

Method	Formula	Measured	Ratio	Fit sd (s)	Inf. sd (ms/1k)
SimpleCIL	1.2 MB	1.2 MB	1.00	0.001	0.02
SLDA	9.2 MB	9.2 MB	1.00	0.051	1.07
Streaming-QDA	2.34 GB	2.35 GB	1.00	0.020	22.64
RanPAC	443 MB	443 MB	1.00	0.019	0.21
FeCAM	3.52 GB	3.52 GB	1.00	0.011	11.77